Audio-Visual Waypoints for Navigation

Changan Chen[1,2] Sagnik Majumder[1] Ziad Al-Halah[1] Ruohan Gao[1,2]
Santhosh K. Ramakrishnan[1,2] Kristen Grauman[1,2]
[1]UT Austin [2]Facebook AI Research

Abstract

In audio-visual navigation, an agent intelligently travels through a complex, unmapped 3D environment using both sights and sounds to find a sound source (e.g., a phone ringing in another room). Existing models learn to act at a fixed granularity of agent motion and rely on simple recurrent aggregations of the audio observations. We introduce a reinforcement learning approach to audio-visual navigation with two key novel elements 1) audio-visual waypoints that are dynamically set and learned end-to-end within the navigation policy, and 2) an acoustic memory that provides a structured, spatially grounded record of what the agent has heard as it moves. Both new ideas capitalize on the synergy of audio and visual data for revealing the geometry of an unmapped space. We demonstrate our approach on the challenging Replica environments of real-world 3D scenes. Our model improves the state of the art by a substantial margin, and our experiments reveal that learning the links between sights, sounds, and space is essential for audio-visual navigation.

1. Introduction

Intelligent robots must be able to move around efficiently in the physical world. Whereas traditional approaches rely on geometric maps and planning [50, 25, 16], recent work in embodied AI shows the promise of agents that instead *learn* to map and navigate. Sensing directly from egocentric images, they jointly learn a spatial memory and navigation policy in order to quickly reach target locations in novel, unmapped 3D environments [24, 23, 44, 35]. High quality simulators have accelerated this research direction to the point where policies learned in simulation can (in some cases) successfully translate to robotic agents deployed in the real world [23, 36, 7, 48].

Most current work centers around visual navigation by an agent that has been told where to find the target [23, 45, 35, 33, 7]. However, in the recently introduced AudioGoal task, the agent must use both visual and auditory sensing to travel through an unmapped 3D environment to find a sound-emitting object, without being told where it is [8, 17]. As a learning problem, AudioGoal not only has strong motivation from cognitive and neuroscience [21, 32, 43], it also has compelling real-world significance: a phone is ringing somewhere upstairs; a person is calling for help from another room; a dog is scratching at the door to go out.

What role should audio-visual inputs play in learning to navigate? There are two existing strategies. One employs deep reinforcement learning to learn a navigation policy that generates step-by-step actions based on both modalities (TurnRight, MoveForward, etc.) [8]. This has the advantage of unifying the sensing modalities, but can be inefficient when learning to make long sequences of individual local actions. The alternative approach separates the modalities—treating the audio stream as a beacon that signals the goal location, then planning a path to that location using a visual mapper [17]. This strategy has the advantage of modularity, but the disadvantage of restricting audio's role to localizing the target. Furthermore, both existing methods make strong assumptions about the granularity at which actions should be predicted, either myopically for each 0.5m step [8] or globally for the final goal location [17].

We introduce a new approach for AudioGoal navigation where the agent instead predicts non-myopic actions with self-adaptive granularity. Our key insight is to learn *audio-visual waypoints*: the agent dynamically sets intermediate goal locations based on its audio-visual observations and partial map. Intuitively, it is often hard to directly localize a distant sound source from afar, but it can be easier to identify the general direction (and hence navigable path) along which one could move closer to that source. See Figure 1.

Both the audio and visual modalities are critical to identifying waypoints in an unmapped environment. Audio input suggests the general goal direction; visual input reveals intermediate obstacles and free spaces; and their interplay indicates how the geometry of the 3D environment is warping the sounds received by the agent, such that it can learn to trace back to the hidden goal. In contrast, subgoals selected using only visual input are limited to mapped locations or clear line-of-sight paths [4, 5].

To realize our idea, our first contribution is a novel deep

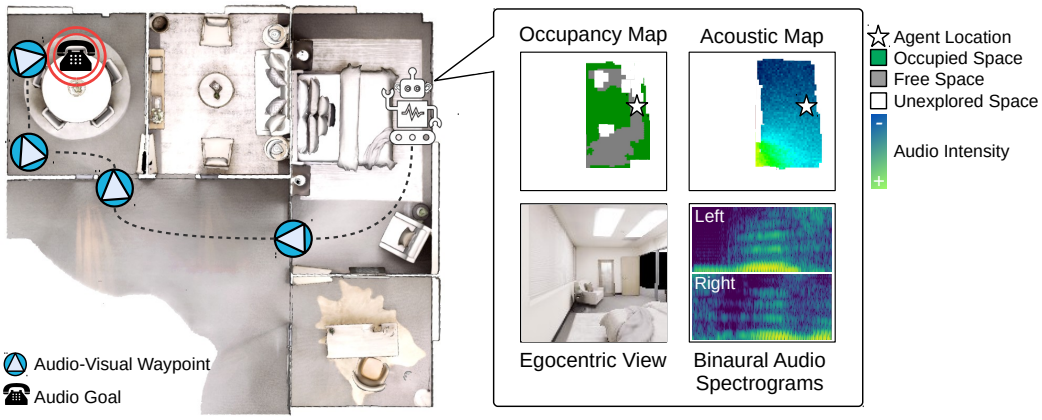


Figure 1: Audio-visual waypoints for navigation: Given egocentric audio-visual sensor inputs (bottom right), the proposed agent builds up both geometric and acoustic maps (top right) as it moves in the unmapped environment. The agent learns encodings for the multi-modal inputs together with a modular navigation policy to find the sounding goal (e.g., phone ringing in top left corner room) via a series of dynamically generated audio-visual waypoints. For example, the agent in the bedroom hears the phone ringing, identifies that it is in another room, and decides to first exit the bedroom. It then narrows down the phone location to the dining room, decides to enter it, and subsequently finds it. By intelligently identifying audio-visual waypoints, the agent efficiently reaches the goal.

reinforcement learning approach for AudioGoal navigation with audio-visual waypoints. The model is hierarchical, with an outer policy that generates waypoints and an inner module that plans to reach each waypoint. Whereas existing visual navigation methods employ classic heuristics to define subgoals (e.g., [48, 4, 5]), the proposed agent learns to set useful subgoals in an end-to-end fashion for the navigation task. This is a new idea for 3D navigation subgoals in general, not specific to audio-visual. As a second technical contribution, we introduce an *acoustic memory* to record what the agent hears as it moves, complementing its visual spatial memory. Whereas existing models aggregate audio evidence purely based on an unstructured memory (GRU), our proposed acoustic map is structured, interpretable, and integrates audio observations throughout the reinforcement learning pipeline.

We demonstrate our approach on the challenging 3D environments of Replica [49, 8]. Our approach outperforms the state of the art for AudioGoal navigation by a substantial margin (10 to 27 points in SPL on heard sounds), and generalizes much better to the challenging cases of unheard sounds and noisy audio. We show audio and vision are together powerful signals for learning to set intermediate goals, while the proposed acoustic memory helps the agent set goals and decide when to stop.

2. Related Work

Learning to navigate in 3D environments Traditionally, robots would navigate complex real-world environments by mapping the space with 3D reconstruction algorithms (i.e., SLAM) and then planning their movements [50, 16]. However, recent work shows the promise of *learning* map encodings and navigation policies directly from egocentric RGB-(D) observations [23, 24, 44, 35]. Current methods focus on the so-called PointGoal task: the agent is given a 2D displacement vector pointing to the goal location and must navigate through free space to get there; the agent relies on visual input and (typically) GPS odometry [23, 35, 33, 45, 7].

In contrast, the recently introduced AudioGoal task requires the agent to navigate to a *sound source* goal using vision and audio [8, 17]. Importantly, unlike PointGoal, AudioGoal does not provide a displacement vector indicating the goal. Existing AudioGoal methods either learn a policy to select the best immediate next action using the multi-modal inputs [8], or predict the final goal location from the audio input and then follow a planned path to it based on visual inputs [17]. Our ideas for audio-visual waypoints and an acoustic map are entirely novel, and have significant impact on results.

Navigation with intermediate goals The norm in the current literature is to learn policies that reward moving to the final goal location using a step-by-step action space (e.g., TurnRight, MoveForward, Stop) [23, 34, 35, 33]. Recent work explores ways to incorporate subgoals or waypoints for PointGoal navigation [48, 7, 4]. Taking inspiration from hierarchical learning [3, 53, 37, 14], the general idea is to select a subgoal, use planning (or a local policy) to navigate to the current subgoal, and repeat [48, 4, 7, 39, 55, 5]. For example, one approach applies a CNN to the RGB input to

infer the next waypoint, then applies model-based planning to find a collision-free path [4]. Active Neural SLAM plans a path to the point goal using a partial map of the environment and navigates to intermediate goals on that path using a local policy [7]. Outside of navigation, related modular approaches are also explored for agents that answer natural language questions [12, 20].

The modular nature of these methods resonates with the proposed model. However, there are several important differences. First, we tackle AudioGoal, not PointGoal, which means our top-level module is not given the goal location and must instead learn how to direct the agent based on the audio inputs. Second, we introduce audio-visual subgoals; whereas visual subgoals focus on visible obstacle avoidance, audio-visual waypoints benefit from the wide reach of audio. For example, a visual subgoal may consider either of two exit doors as equally good, whereas an audio-visual subgoal prefers the one from which greater sound appears to be emerging. Third, a key element of our approach is to learn to generate navigation subgoals in an end-to-end fashion. In contrast, prior work relies on heuristics like selecting frontiers [5, 48] or points along the shortest path [4, 7] to define subgoals. This is an important technical contribution independent of the audio-visual setting, as it frees the agent to dynamically identify subgoals driven by the ultimate navigation goal.

Visual semantic memory and mapping Learning-based visual mapping algorithms [27, 44, 24, 23] show exciting promise to overcome the limits of purely geometric maps (e.g., SLAM). A learned map can encode semantic information beyond 3D points while being trained with the agent's ultimate task (like navigation). Recent work explores memories that spatially index learned RGB-D features [52, 27, 23], build a topological memory with visually distinct nodes [44, 38, 6], or use attention models over stored visual embeddings [15]. Expanding this line of work, we introduce the first multi-modal spatial memory. It encodes both visual and acoustic observations registered with the agent's movement along the ground plane. We show that the multi-modal memory is essential for the agent to capture links between sounds, sights, and space, in order to produce good action sequences.

Sound localization Robotics systems localize sound sources with microphone arrays [41, 42], and active control can improve localization [40, 54]. The geometry of a room can be in part sensed by audio, as explored with ideas for echolocation [13, 10, 18]. In 2D video frames, methods learn to localize sounds based on their consistent audio-visual association [28, 51, 47, 2]. Unlike any of the above, we investigate the audio-visual navigation problem, where an agent learns to move efficiently towards a sound source in a 3D environment based on both audio and visual cues.

3. Approach

We consider the task of AudioGoal navigation [8, 17]. In this task the agent moves within a 3D environment and receives a sensor observation O_t at each time step t from its camera (RGB/depth) and binaural microphones. The environment is unmapped at the beginning of the navigation episode; the agent has to accumulate observations to understand the scene geometry while navigating. Unlike the common PointGoal task [1, 33, 19, 56, 31, 7], for AudioGoal the agent does not know the location of the goal (*i.e.* no GPS signal or displacement vector pointing to the goal is available). The agent must use the sound emitted by the audio source to locate and navigate successfully to the goal.

We introduce a novel navigation approach that predicts intermediate waypoints to reach the goal efficiently.

Our approach is composed of three main modules (Fig. 2). Given visual and audio inputs, our model 1) encodes these cues using a perception and mapping module, then 2) predicts a waypoint, and finally 3) plans and executes a sequence of actions that bring the agent to the predicted waypoint. The agent repeats this process until it predicts the goal has been reached and executes the *Stop* action.

3.1. 3D Environments and Audio-Visual Simulator

To explore our idea in a reproducible evaluation setting, we use the AI-Habitat simulator [33] with the publicly available Replica environments [49] together with the audio simulations and audio-visual (AV) navigation benchmark introduced by Chen and Jain et al. [8]. The Replica environments are meshes constructed from real-world scans of 18 apartments, offices, hotels, and rooms.[1] The agent can travel through the spaces while receiving real-time egocentric visual and audio observations. Using the room impulse responses (RIR) introduced by [8], we can place an audio source in the 3D environment, then simulate realistic audio renderings at each location in the scene at a spatial resolution of 0.5m. These state-of-the-art renderings capture how sound propagates and interacts with the surrounding geometry and surface materials, modeling all of the major features of the RIR: direct sound, early specular/diffuse reflections, reverberation, binaural spatialization, and frequency dependent effects from materials and air absorption (see [8] for details).

The simulator maintains a navigability graph of the environment (unknown to the agent). The agent can only move from one node to another if there is an edge connecting them and the agent is facing that direction. The action space $\mathcal{A}$

[1] The AI2-THOR simulation from [17] is not yet available, and it contains synthetic computer graphics imagery (vs. Replica's real-world scans).

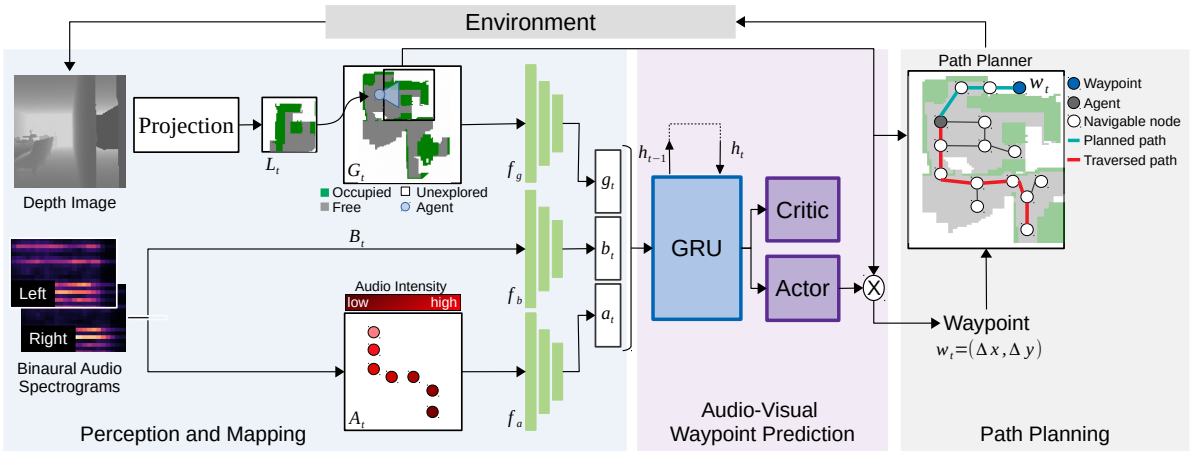


Figure 2: Model architecture. Our audio-visual navigation model uses the egocentric stream of depth images and binaural audio (B_t) to learn geometric (G_t) and acoustic (A_t) maps for the 3D environment. The multi-modal cues and partial maps (left) inform the RL policy's prediction of intermediate waypoints (center). For each waypoint, the agent plans the shortest navigable path (right). From this sequence of waypoints, the agent reaches the final AudioGoal efficiently.

has four actions: *MoveForward*, *TurnLeft*, *TurnRight* and *Stop*.

3.2. Perception and Mapping

Visual perception At each time step t, we extract visual cues from the agent's first-person depth view, which is more effective for map construction than RGB [7, 9]. First, we backproject the depth image into the world coordinates using the camera's intrinsic parameters to compute the local scene's 3D point cloud. Then, we project these points to a 2D top-down egocentric local occupancy map L_t of size 3×3 meters in front of the agent, corresponding to the typical distance at which the real-world sensor is reliable. The map has two channels, one for the occupied/free space and one for explored/unexplored areas. A map cell is deemed occupied if a 3D point is higher than 0.2m and lower than 1.5m, and it is deemed explored if any 3D point is projected into that cell. (Results are tolerant to noisy depth; see Supp.) We update an allocentric geometric map G_t by transforming L_t with respect to the agent's last pose change and then averaging it with the corresponding values of G_{t-1}.

Cells with a value above 0.5 are considered occupied or explored. See top branch in Figure 2.

Acoustic perception At each time step the agent receives binaural sound B_t represented by spectrograms for the right and left ear, a matrix representation of frequencies of audio signals as a function of time (second branch in Figure 2; see Supp for spectrogram details). Beyond encoding the current sounds, we also introduce an *acoustic memory*. The acoustic memory is a map A_t indexed on the ground plane like G_t that aggregates the audio intensity over time in a structured manner. It records a moving average of direct sound intensity solely at positions visited by the agent. See the third branch in Figure 2. The acoustic map and B_t provide spatially grounded information about both the environment and the goal: the walls and other major surfaces influence the sound received by the agent at any given location, while the sound source at the goal gives a coarse sense of direction when the agent is far away. This directional cue gets increasingly precise as the agent approaches the goal.

3.3. Audio-Visual Waypoint Predictor

Both the audio and visual inputs carry complementary information to set good waypoints en route to the audio goal. While the audio signals B_t (binaural inputs) and A_t (acoustic memory) inform the agent of the general direction of the goal and hint at the room geometry, the visual signal in the form of the occupancy map G_t allows spatial localization of the waypoint and helps to avoid obstacles. Recall Figure 1, where the agent in the bedroom needs to reach a phone ringing in another room.

We learn three encoders to represent the inputs: $g_t = f_g(G_t)$, $b_t = f_b(B_t)$ and $a_t = f_a(A_t)$. Functions f_g and f_a first transform the geometric and acoustic maps (G_t and A_t) such that the agent is located at the center of the map facing upwards and then crop them to size $s_g \times s_g$ and $s_a \times s_a$, respectively. Each function has a deep convolutional neural network (CNN) in the end to extract features (architecture details in Supp). We concatenate the three vectors g_t, b_t and a_t to obtain the full audio-visual feature, and pass it into a gated recurrent neural network (GRU) [11]. See Figure 2.

Our reinforcement learning waypoint predictor has an actor-critic architecture. It takes the hidden state h_t of the GRU and predicts a probability distribution $\pi(W_t|h_t)$ over possible waypoints. W_t is the action map of size $s_w \times s_w$ and represents the candidate waypoints in the area centered

around the agent.[2] We mask the output of the policy with the local occupancy map to ensure that the model selects waypoints that are in free spaces. We sample a waypoint $w_t = (\Delta x, \Delta y)$ from W_t according to the policy's predicted probability distribution. The waypoint is relative to the agent's current position and is passed to the planner (see Sec. 3.4).

This waypoint policy is an important element in our method design. It allows the agent to dynamically adjust its intermediate goals according to what it currently sees and hears. Unlike existing audio-visual navigation methods, our waypoints guide the agent at a variable granularity, as opposed to fixing its actions to myopic next steps [8] or a final goal prediction [17]. Unlike existing visual subgoal approaches, which rely on frontier-based heuristics or points along the shortest path [7, 48, 4, 5], our waypoints are inferred in tight integration with the navigation task.

3.4. Path Planner

Given the generated waypoint w_t, a shortest-path planner tries to generate a sequence of low-level actuation commands chosen from $\mathcal{A}$ to move the agent to that waypoint. The planner maintains a graph of the scene based on the geometric map G_t and estimates a path from the agent's current location to w_t using Dijkstra's algorithm. Based on the shortest path, a low-level actuation command is analytically computed. The agent executes the action, gets a new observation O_t, updates both G_t and A_t, and repeats the above procedure until it exits the planning loop.

The planning loop breaks under three conditions: 1) the agent reaches the waypoint, 2) the planner could not find a path to the waypoint, or 3) the agent reaches a planning step limit. The planning step limit is set to prevent bad waypoint prediction (due to noisy occupancy estimates) or hard-to-reach waypoints (like behind the wall of another room) from derailing the agent from the goal. If the model selects $w_t = (0, 0)$ (*i.e.* the agent's current location), this means that the agent believes it has reached the final goal; the *Stop* action is then executed and the episode terminates.

3.5. Reward and Training

Following typical navigation rewards [33, 8], we reward the agent with $+10$ if it succeeds in reaching the goal and executing the *Stop* action there, plus an additional reward of $+0.25$ for reducing the geodesic distance to the goal and an equivalent penalty for increasing it. Finally, we issue a time penalty of -0.01 per executed action to encourage efficiency. For each waypoint prediction step, the agent is rewarded with the cumulative reward value collected during the last round of planner execution. Altogether, the reward encourages the model to select waypoints that are reachable, far from the current agent position, and on the route to the goal, or to choose the goal itself if it is within reach. We train the model end-to-end with Proximal Policy Optimization (PPO) [46].

[2]Note that the Replica environment graphs have nodes only where the audio RIRs are available, and hence both actions and candidate waypoints are discrete sets. This disallows testing noisy actuation for any method.

4. Experiments

Environment We test with the Replica [49] environments in the Habitat simulator (Sec. 3.1). We follow the standard evaluation protocol [8] with train/val/test splits of 9/4/5 scenes. We stress that the test environments are disjoint from the train/val environments, requiring the agent to learn generalizable behaviors. Furthermore, for the same scene splits, we experiment with training and testing on disjoint sounds, requiring the agent to generalize to unheard sounds. Except where specified, the telephone ringing is the sound source. We follow the protocol of the benchmark [8], except we double the number of episodes tested for all models in order to reduce the variance in performance.

Implementation details We train our model with Adam [30] with a learning rate of 2.5×10^{-4}. The output of the three encoders g_t, b_t and a_t are all of dimension 512. We use a one-layer bidirectional GRU [11] with 512 hidden units that takes $[g_t, b_t, a_t]$ as input. The geometric map size s_g is 200 at a resolution of 0.1m, and the acoustic map size s_a is 20 at a resolution of 0.5m. The action map size s_w is 9 at a resolution of 0.5m, chosen based on validation, which corresponds to a physical size of 4.5×4.5m. We use an entropy loss on the policy distribution with coefficient 0.02. We train for 7.5 million policy prediction steps, and we set the upper limit of planning steps to 10.

Metrics We evaluate the following navigation metrics: 1) success rate (SR), the fraction of successful episodes; 2) success weighted by path length (SPL), the standard metric [1] that weighs successes by their adherence to the shortest path; 3) normalized distance to goal (NDG), the distance to the goal at the episode's conclusion relative to the shortest path, even for failure cases; 4) number of actions (NA), the average number of actions per episode, which penalizes rotation in place actions, which do not lead to path changes; 5) success weighted by number of actions (SNA). Please see Supp for more details on these comprehensive metrics and the baseline implementations.

Existing methods and baselines We compare the following methods (detailed in Supp):

- Random: a naive agent that randomly selects each action and signals *Stop* when it reaches the goal.
- Audio Direction Follower: predicts the audio direction of arrival (DoA), navigates in that direction for four steps, then repeats. We train a separate classifier based on audio input to predict when this agent should stop.

Table 1: AudioGoal navigation results. Our audio-visual waypoints navigation model (AV-WaN) reaches the goal faster (higher SPL) and it is more efficient (lower NA and higher SNA) compared to the state-of-the-art. SPL, SR, NDG and SNA are shown as percentages. The arrow direction shows whether higher ↑ or lower ↓ is better for each metric.

	Heard sound					*Unheard sounds*				
Model	SPL ↑	SR ↑	NDG ↓	NA ↓	SNA↑	SPL ↑	SR ↑	NDG ↓	NA ↓	SNA↑
Random Agent	4.7	18.5	89.7	448.9	1.7	4.7	18.5	89.7	448.9	1.7
Audio Direction Follower	54.7	72.0	19.5	173.0	41.1	11.1	17.2	55.4	415.2	8.4
Frontier Waypoints	44.0	63.9	47.5	218.5	35.2	6.5	14.8	96.3	434.2	5.1
Gan et al. [17]	57.6	83.1	17.6	130.5	47.9	7.5	15.7	76.3	336.6	5.7
Chen and Jain et al. [8]	75.0	95.7	1.3	84.4	43.4	28.6	41.2	16.0	223.1	14.9
AV-WaN (**Ours**)	**85.3**	**98.0**	**0.6**	**38.8**	**70.0**	**54.4**	**71.2**	**12.3**	**173.6**	**43.8**

Table 2: Ablation study for AV-WaN.

Model	SPL ↑	SR ↑	NDG ↓	NA ↓	SNA↑
AV-WaN w/o A_t and G_t	77.0	93.1	1.6	52.7	61.3
AV-WaN w/o G_t	82.0	97.2	0.7	41.2	66.1
AV-WaN w/o A_t	81.4	94.9	1.1	43.8	66.8
AV-WaN	**85.3**	**98.0**	**0.6**	**38.8**	**70.0**

– Frontier Waypoints: intersects the predicted DoA with the frontiers of the explored area and selects that point as the next waypoint. Frontier waypoints are commonly used in the visual navigation literature (e.g., [5, 48, 9, 7]), making this a broadly representative baseline for standard practice.

– Chen and Jain et al. [8]: a state-of-the-art end-to-end RL agent that selects actions using both visual and acoustic observations. It lacks any geometric or acoustic maps. We run the authors' code.

– Gan et al. [17]: a state-of-the-art agent that predicts the audio goal location from binaural spectrograms alone and then navigates with an analytical path planner on an occupancy map it progressively builds by projecting depth images. It uses a separate audio classifier to stop. We adapt the model to improve its performance on the Replica data, since the authors originally tested on a game engine simulator.

Navigation results We consider two settings: 1) *heard sound*—train and test on the telephone sound, following [8, 17], and 2) *unheard sounds*—train and test with disjoint sounds, following [8].[3] In both cases, the test environment is always unseen, hence both settings require generalization.

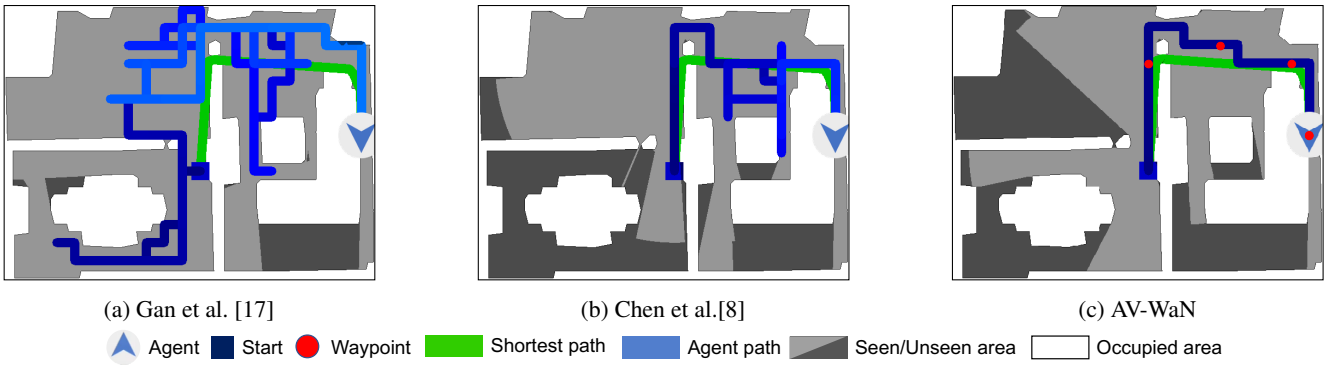
Table 1 shows the results. We refer to our model as AV-WaN (**A**udio-**V**isual **Wa**ypoint **N**avigation). Random does poorly due to the challenging nature of the AudioGoal task and the complex 3D environments. For the heard sound, AV-WaN strongly outperforms all the baselines and both existing methods—with 10.3% and 27.7% gains in SPL compared to [8] and [17], respectively. This result shows the advantage of our dynamic audio-visual waypoints and structured acoustic map, compared to the myopic action selection in [8] and the final-goal prediction in [17]. We find that the RL model of [8] fails when it oscillates around an obstacle. Meanwhile, we find that predicting the final audio goal location, as done by [17], is prone to errors and leads the agent to backtrack or change course often to redirect itself towards the goal. See Figure 3. The baselines that rely only on audio or frontiers to select waypoints significantly underperform our model, which demonstrates that learning to use both acoustic and visual cues is crucial to select good waypoints. Further, it highlights the value of directly *learning* waypoints, versus current heuristics. Additionally, our model increases the agent's efficiency substantially, as seen in the 22.1% gain in SNA compared to the best baseline; AV-WaN makes significantly fewer unnecessary actions (*e.g.* turning in place) and collisions.

In the unheard sounds setting (Table 1, right), our method again strongly outperforms all existing methods. Absolute performance declines for all methods, though, due to the unfamiliar audio spectrogram patterns. The acoustic memory is critical for this important setting; it successfully abstracts away the specific content of the training sounds to better generalize.

Ablations Table 2 shows ablations of the main components of our model. [4] Removing both the geometric and acoustic maps causes a big reduction in performance. This is expected since without A_t and G_t, the model has only the current audio observation B_t to predict the next waypoint. Notably, even this heavily ablated version of our model outperforms the best existing model [8] (see Table 1). This shows that our waypoint-based navigation framework itself is more ef-

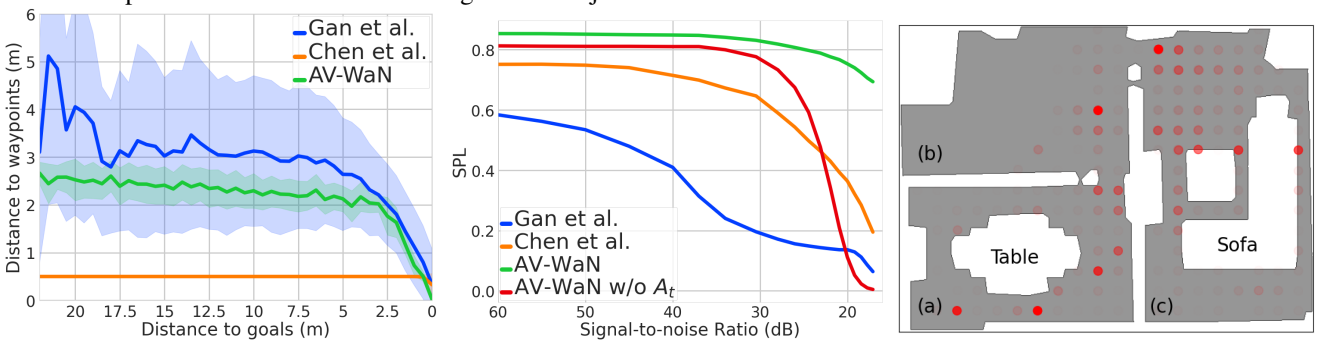
[3] Train on {telephone, fan, horn, engine 1, radio 1, music 1} and test on {engine 3, radio 3, music 3} from [8].

[4] We remove the masking operation when G_t is removed to ensure no geometric information is used.



(a) Gan et al. [17] (b) Chen et al.[8] (c) AV-WaN

Figure 3: Navigation trajectories on top-down maps. Agent path fades from dark blue to light blue as time goes by. Green is the shortest geodesic path in continuous space. All agents have reached the goal. Our waypoint model navigates to the goal more efficiently. Recall that the agent's inputs are egocentric views of the scanned real-world environment (Fig. 1); figures show the top-down view for ease of viewing the full trajectories.



(a) Waypoint distance distribution (b) Audio noise (c) Waypoint placement distribution

Figure 4: Analysis of selected waypoints (a,c) and accuracy vs. noise (b). See text.

fective than the simpler RL model [8]. We also see that the impact of A_t is higher than that of G_t. This demonstrates the importance of the proposed structured acoustic memory for efficient navigation. We find that A_t is especially helpful in selecting *Stop* when the agent has reached the goal and facilitates waypoint prediction (reduction in NDG > improvement in SR). However, both A_t and G_t are complementary and critical for our model to reach its best performance.

Dynamic waypoint selection To analyze the behavior of dynamic waypoint selection, Fig. 4a plots the distribution of euclidean distances to waypoints as a function of the agent's geodesic distance to the goal collected from all prediction steps across all episodes. The existing methods do not set waypoints. For Chen et al. [8], we say the "waypoint" distance is 0m if the agent chooses to stop and 0.5m otherwise. For Gan et al. [17], we say "waypoints" are the intersection of the predicted vector to the final goal and the explored area. We see that our agent selects waypoints that are further away when it is far from the goal, then predicts closer ones when converging on the goal. In contrast, the step-by-step model [8] effectively has a fixed radius waypoint, while the final-goal model [17] has a high variance even when close to the goal, indicative of the redirection and backtracking behavior described above. The large waypoint distances for [17] are a symptom of its backtracking due to misprediction and lack of temporal modeling; since depth projections are limited to 3m, an ideal agent should always pick a waypoint up to 3m away and push forward to the goal.

Noisy audio perception Fig. 4b shows the results of adding increasing Gaussian noise to the received audio waveforms, representing increasing audio noise, e.g., from the microphones. We see AV-WaN is quite robust to audio noise especially with A_t, while the existing methods suffer significantly.

Placement of waypoints To examine how waypoints are selected based on surrounding geometry, Fig. 4c plots the distribution of waypoints on a top-down map for a test environment. The waypoints are accumulated over trajectories with start or end points in room *a* or room *c*, and goal locations are excluded. We see waypoints are mostly selected around obstacles and doors, which enables the agent to traverse the space efficiently to the goal. The most frequent waypoints are usually 2-3m apart, which is near the maximum distance the agent can choose.

Time-varying sounds Finally, we consider a setting where the sound source (*music*) varies over time as the agent moves. Our method maintains its advantage in this challenging setting. See Supp.

5. Conclusion

We introduced a reinforcement learning framework for audio-visual navigation with waypoints and an acoustic memory. Our method improves the state of the art on this challenging problem, and our analysis shows the direct impact of the new technical contributions. In future work we plan to consider AV-navigation tasks of increasing complexity, such as distracting sound sources, moving sound sources, and real-world transfer.

References

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

[2] Relja Arandjelovic and Andrew Zisserman. Objects that sound. In *ECCV*, 2018.

[3] Pierre-Luc Bacon, Jean Harb, and Doina Precup. The option-critic architecture. In *AAAI*, 2017.

[4] Somil Bansal, Varun Tolani, Saurabh Gupta, Jitendra Malik, and Claire Tomlin. Combining optimal control and learning for visual navigation in novel environments. In *Conference on Robot Learning (CoRL)*, 2019.

[5] J. A. Caley, N. R. J. Lawrance, and G. A. Hollinger. Deep learning of structured environments for robot search. In *IROS*, 2016.

[6] Devendra Chaplot, Ruslan Salakhutdinov, Abhinav Gupta, and Saurabh Gupta. Neural topological slam for visual navigation. In *CVPR*, 2020.

[7] Devendra Singh Chaplot, Dhiraj Gandhi, Saurabh Gupta, Abhinav Gupta, and Ruslan Salakhutdinov. Learning to explore using active neural slam. In *ICLR*, 2020.

[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.

[9] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *International Conference on Learning Representations*, 2019.

[10] Jesper Christensen, Sascha Hornauer, and Stella Yu. Batvision: Learning to see 3d spatial layout with two ears. *arXiv:1912.07011*, 2020.

[11] Junyoung Chung, Kyle Kastner, Laurent Dinh, Kratarth Goel, Aaron C Courville, and Yoshua Bengio. A recurrent latent variable model for sequential data. In *NeurIPS*, 2015.

[12] Abhishek Das, Georgia Gkioxari, Stefan Lee, Devi Parikh, and Dhruv Batra. Neural modular control for embodied question answering. In *Conference on Robot Learning (CoRL)*, pages 53–62, 2018.

[13] I. Dokmanic, R. Parhizkar, A. Walther, Y. Lu, and M. Vetterli. Acoustic echoes reveal room shape. *PNAS*, 2013.

[14] Ben Eysenbach, Russ R Salakhutdinov, and Sergey Levine. Search on the replay buffer: Bridging planning and reinforcement learning. In *NeurIPS*, 2019.

[15] Kuan Fang, Alexander Toshev, Li Fei-Fei, and Silvio Savarese. Scene memory transformer for embodied agents in long-horizon tasks. In *CVPR*, 2019.

[16] Jorge Fuentes-Pacheco, José Ruíz Ascencio, and Juan M. Rendón-Mancha. Visual simultaneous localization and mapping: a survey. *Artificial Intelligence Review*, 43:55–81, 2012.

[17] Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In *ICRA*, 2020.

[18] Ruohan Gao, Changan Chen, Ziad Al-Halah, Carl Schissler, and Kristen Grauman. Visualechoes: Spatial image representation learning through echolocation. In *ECCV*, 2020.

[19] Daniel Gordon, Abhishek Kadian, Devi Parikh, Judy Hoffman, and Dhruv Batra. Splitnet: Sim2sim and task2task transfer for embodied visual navigation. *ICCV*, 2019.

[20] Daniel Gordon, Aniruddha Kembhavi, Mohammad Rastegari, Joseph Redmon, Dieter Fox, and Ali Farhadi. Iqa: Visual question answering in interactive environments. In *CVPR*, 2018.

[21] Frédéric Gougoux, Robert J Zatorre, Maryse Lassonde, Patrice Voss, and Franco Lepore. A functional neuroimaging study of sound localization: visual cortex activity predicts performance in early-blind individuals. *PLoS biology*, 2005.

[22] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1984.

[23] Saurabh Gupta, James Davidson, Sergey Levine, Rahul Sukthankar, and Jitendra Malik. Cognitive mapping and planning for visual navigation. In *CVPR*, 2017.

[24] Saurabh Gupta, David Fouhey, Sergey Levine, and Jitendra Malik. Unifying map and landmark based representations for visual navigation. *arXiv preprint arXiv:1712.08125*, 2017.

[25] R. I. Hartley and A. Zisserman. *Multiple View Geometry in Computer Vision*. Cambridge University Press, ISBN: 0521540518, second edition, 2004.

[26] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

[27] Joao F Henriques and Andrea Vedaldi. Mapnet: An allocentric spatial memory for mapping environments. In *CVPR*, 2018.

[28] John R Hershey and Javier R Movellan. Audio vision: Using audio-visual synchrony to locate sounds. In *NeurIPS*, 2000.

[29] 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.

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

[31] Noriyuki Kojima and Jia Deng. To learn or not to learn: Analyzing the role of learning for navigation in virtual environments. *arXiv preprint arXiv:1907.11770*, 2019.

[32] Nadia Lessard, Michael Paré, Franco Lepore, and Maryse Lassonde. Early-blind human subjects localize sound sources better than sighted subjects. *Nature*, 1998.

[33] 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 *ICCV*, 2019.

[34] Piotr Mirowski, Razvan Pascanu, Fabio Viola, Hubert Soyer, Andrew J Ballard, Andrea Banino, Misha Denil, Ross Goroshin, Laurent Sifre, Koray Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.

[35] Dmytro Mishkin, Alexey Dosovitskiy, and Vladlen Koltun. Benchmarking classic and learned navigation in complex 3d environments. *arXiv preprint arXiv:1901.10915*, 2019.

[36] Matthias Müller, Alexey Dosovitskiy, Bernard Ghanem, and Vladlen Koltun. Driving policy transfer via modularity and abstraction. *CoRR*, abs/1804.09364, 2018.

[37] Ofir Nachum, Shixiang Shane Gu, Honglak Lee, and Sergey Levine. Data-efficient hierarchical reinforcement learning. In *NeurIPS*, 2018.

[38] Tushar Nagarajan, Yanghao Li, Christoph Feichtenhofer, and Kristen Grauman. Ego-topo: Environment affordances from egocentric video. In *CVPR*, 2020.

[39] Suraj Nair and Chelsea Finn. Hierarchical foresight: Self-supervised learning of long-horizon tasks via visual subgoal generation. In *ICLR*, 2020.

[40] Kazuhiro Nakadai, Tino Lourens, Hiroshi G Okuno, and Hiroaki Kitano. Active audition for humanoid. In *AAAI*, 2000.

[41] Kazuhiro Nakadai and Keisuke Nakamura. Sound source localization and separation. *Wiley Encyclopedia of Electrical and Electronics Engineering*, 1999.

[42] Caleb Rascon and Ivan Meza. Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 2017.

[43] Brigitte RoÈder, Wolfgang Teder-SaÈlejaÈrvi, Anette Sterr, Frank RoÈsler, Steven A Hillyard, and Helen J Neville. Improved auditory spatial tuning in blind humans. *Nature*, 1999.

[44] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. *arXiv preprint arXiv:1803.00653*, 2018.

[45] Alexander Sax, Bradley Emi, Amir R Zamir, Leonidas Guibas, Silvio Savarese, and Jitendra Malik. Mid-level visual representations improve generalization and sample efficiency for learning visuomotor policies. *arXiv preprint arXiv:1812.11971*, 2018.

[46] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

[47] Arda Senocak, Tae-Hyun Oh, Junsik Kim, Ming-Hsuan Yang, and In So Kweon. Learning to localize sound source in visual scenes. In *CVPR*, 2018.

[48] Gregory J. Stein, Christopher Bradley, and Nicholas Roy. Learning over subgoals for efficient navigation of structured, unknown environments. In Aude Billard, Anca Dragan, Jan Peters, and Jun Morimoto, editors, *Proceedings of The 2nd Conference on Robot Learning*, volume 87 of *Proceedings of Machine Learning Research*, pages 213–222. PMLR, 29–31 Oct 2018.

[49] Julian Straub, Thomas Whelan, Lingni Ma, Yufan Chen, Erik Wijmans, Simon Green, Jakob J Engel, Raul Mur-Artal, Carl Ren, Shobhit Verma, et al. The replica dataset: A digital replica of indoor spaces. *arXiv preprint arXiv:1906.05797*, 2019.

[50] Sebastian Thrun. Probabilistic robotics. *Communications of the ACM*, 45(3):52–57, 2002.

[51] Yapeng Tian, Jing Shi, Bochen Li, Zhiyao Duan, and Chenliang Xu. Audio-visual event localization in unconstrained videos. In *ECCV*, 2018.

[52] Hsiao-Yu Fish Tung, Ricson Cheng, and Katerina Fragkiadaki. Learning spatial common sense with geometry-aware recurrent networks. In *CVPR*, 2019.

[53] Alexander Sasha Vezhnevets, Simon Osindero, Tom Schaul, Nicolas Heess, Max Jaderberg, David Silver, and Koray Kavukcuoglu. Feudal networks for hierarchical reinforcement learning. In *ICML*, 2017.

[54] Y. Wang, M. Kapadia, P. Huang, L. Kavan, and N. Badler. Sound localization and multi-modal steering for autonomous virtual agents. In *Interactive 3D Graphics and Games (I3D)*, 2014.

[55] Jimmy Wu, Xingyuan Sun, Andy Zeng, Shuran Song, Johnny Lee, Szymon Rusinkiewicz, and Thomas Funkhouser. Spatial action maps for mobile manipulation. *arXiv preprint arXiv:2004.09141*, 2020.

[56] Fei Xia, William B Shen, Chengshu Li, Priya Kasimbeg, Micael Tchapmi, Alexander Toshev, Roberto Martín-Martín, and Silvio Savarese. Interactive gibson: A benchmark for interactive navigation in cluttered environments. *arXiv preprint arXiv:1910.14442*, 2019.

6. Supplementary Material

In this supplementary material we provide additional details about:

- Video (with audio) for qualitative assessment of our agent's performance.
- Noisy depth experiment (Sec. 6.2), as noted in Sec. 3.2 of the main paper.
- Time-varying sound as audio goal experiment (Sec. 6.3), as referenced in Sec. 4 of the main paper.
- Effect of the masking operation (Sec. 6.4).
- Binaural spectrogram calculation details (Sec. 6.5).
- Details of the CNN encoders from the perception and mapping component of our AV-WaN (Sec. 6.6).
- Details on the navigation metric definitions (Sec. 6.7).
- Baselines' implementation details (Sec. 6.8).

6.1. Qualitative Video

The supplementary video demonstrates the audio simulation platform that we use and shows the comparison between our proposed model and the baselines as well as qualitative results from the unheard and time-varying sound experiments. Please listen with headphones to hear the binaural audio correctly. The video is available at `http://vision.cs.utexas.edu/projects/audio_visual_waypoints`.

6.2. Noisy Depth Perception

Similar to our experiments on the model robustness against noisy audio perception, we test here the impact of noise on depth, the other input modality used by the agents. We use the standard Redwood depth noise model from AI-Habitat [33]. To imitate a Sim2Real scenario, we do not retrain the models with the noisy depth sensor; we use noisy depth at inference time only. We find that all models are robust to this type of noise, with SPL performance varying by less than 1% and SR by less than 0.5%. These mild reductions for all methods are smaller than the margins separating the different baselines, meaning the conclusions from the main results hold with noisy depth at test time.

6.3. Time-varying Sound

Next we give details for the time-varying sound experiment summarized in Sec. 4 of the main paper. Here we consider a variant of the evaluation setting where the audio goal is a time-varying sound (*e.g.* people talking, music). While periodic sounds (*e.g.* phone ringing, baby crying, dog barking, alarm sounding) are also challenging and common in real-world settings, time-varying sounds present added technical challenges for the learning agent. In this setting, there are variations in the audio properties (like intensity) that are not only a function of distance to the goal and spatial configuration, but also time.

To study this case, we use a *music* clip (a piano playing) as the target audio goal instead of a ringing telephone (see the heard sound experiment in main paper). In this new setup, whenever the agent is spawned at a random location in the environment in the beginning of an episode $t = 0$ we also select a random starting temporal point of the audio source t_a. Then, at each time step t the agent hears a new 1 second clip of the audio source at $t_a + t$. We loop from the beginning if the end of the audio source is reached.

In Table 3, we see that AV-WaN outperforms all the baselines and existing methods by a substantial margin— 12.8% and 35.4% gains in SPL compared to [8] and [17], respectively. Whereas our model shows robustness and maintains a similar level of performance as observed with periodic sounds, both [17] and [8] struggle to adapt to this new type of audio goal (compare to the heard sound results in Table 1 in the main paper). Furthermore, an ablation study of our model (Table 3 bottom) shows again that our idea of the structured acoustic memory (A_t) plays an essential role in our model's ability to navigate fast and to handle audio variations efficiently.

6.4. Effect of the Masking Operation

To understand the effect of the masking operation (Sec. 3.3 in main paper), Table 4 shows an ablation experiment in the heard sound setting. We remove the masking operation for our AV-WaN model at test time (row 1). Doing so harms SPL by 0.7% compared to our full model because of some invalid predictions, *i.e.* waypoints that are predicted in an occupied area.

6.5. Spectrogram Details

We resample the audio to 22.05 kHZ, and obtain the audio spectrograms for both binaural channels using the Short-Time Fourier Transform (STFT) [22] with a Hann window of length 2048 and hop length of 512. We downsample the computed spectrogram frequencies by a factor of 25 and concatenate the left and right audio spectrograms as two channels in a tensor B_t of size $41 \times 40 \times 2$, following [8]. A room impulse response (RIR) is characterized by three stages: direct sound, early reflections, and reverberation. Direct sound is a strong signal of agent's distance to goal. We compute the intensity of the direct sound part of the audio signal by taking the root-mean-square (RMS) value of the first 3ms non-zero audio waveform averaged between the left and right channels.

Table 3: AudioGoal navigation results with a time-varying sound.

Model	SPL ↑	SR ↑	NDG ↓	NA ↓	SNA ↑
Random Agent	4.7	18.5	89.7	448.9	1.7
Audio Direction Follower	56.2	76.7	11.3	141.2	42.1
Frontier Waypoints	43.6	63.2	46.4	210.7	35.1
Gan et al. [17]	49.6	79.5	13.2	123.2	40.6
Chen et al. [8]	72.2	91.6	2.3	90.7	40.6
AV-WaN w/o A_t and G_t	74.8	97.1	0.3	45.5	59.3
AV-WaN w/o G_t	83.5	98.4	0.3	38.7	68.2
AV-WaN w/o A_t	77.2	95.3	1.3	43.6	61.9
AV-WaN	**85.0**	**98.8**	**0.2**	**36.4**	**69.2**

Table 4: Ablation experiments on the masking operation.

Model	SPL ↑	SR ↑	NDG ↓	NA ↓	SNA ↑
AV-WaN w/o masking operation	84.6	**98.1**	**0.2**	**36.2**	68.8
AV-WaN	**85.3**	98.0	0.6	38.8	**70.0**

6.6. CNN Architecture Details

The CNN component of the f_g and f_b encoders has three convolution layers each, with kernel sizes of [8, 4, 3] and strides of [4, 2, 1] respectively. Similarly, the CNN component of f_a has three convolution layers with kernel sizes of [5, 3, 3] and strides of [2, 1, 1]. For all CNNs, the channel size doubles after each convolution layer starting from 32 and each convolution layer is followed by a ReLU activation function. We use a fully connected layer at the end of each CNN to transform the CNN features into an embedding of size 512.

6.7. Metric Definitions

Next we elaborate on the navigation metrics defined in Sec. 4 of the main paper.

1. Success Rate (SR): the fraction of successfully completed episodes, *i.e*. the agent reaches the goal within the time limit of 500 steps and selects the stop action exactly at the goal location.

2. Success weighted by path length (SPL) [1]: weighs the successful episodes with the ratio of the shortest path l_i to the executed path p_i, $\text{SPL} = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i}{\max(p_i, l_i)}$.

3. Normalized distance to goal (NDG): the distance of the agent to the goal d_i at the end of the episode relative to the shortest path length l_i, $\text{NDG} = \frac{1}{N}\sum_{i=1}^{N} \frac{d_i}{l_i}$. While SR and SPL measure only the performance for successful episodes, NDG shows us how close the agent came to the goal for failure cases as well.

4. Number of actions (NA): the average number of actions executed by the agent across all episodes. This metric more stringently captures the agent's efficiency in navigating to the goal. Two agents my have different NA but the same SPL score for an episode since NA also penalizes actions that do not lead to path changes, like rotation in place.

5. Success weighted by number of actions (SNA): weighs the successful episodes by the ratio of the number of actions taken for the shortest path l_i^a to the number of executed actions by the agent's p_i^a, $\text{SNA} = \frac{1}{N}\sum_{i=1}^{N} S_i \frac{l_i^a}{\max(p_i^a, l_i^a)}$.

6.8. Baseline Implementation Details

6.8.1 Gan et al. [17]

Since code for [17] was not available at the time of our submission, we implemented this method ourselves. We followed instructions given by the authors, and also implemented our own enhancements to improve its performance on the Replica dataset.

We use a VGG-like CNN to predict the relative location $(\Delta x, \Delta y)$ of the audio goal given the binaural audio spectrograms as input. The CNN has 5 convolutional (conv.) layers interleaved with 5 max pooling layers and followed by 3 fully-connected (FC) layers. Each of the conv. and FC layers has a batch-normalization [29] of 10^{-5} and a ReLU activation function except for the last FC layer which outputs the prediction. The conv. layers have the following configuration - a square kernel of size 3, a stride of 1 in both directions, and a symmetric zero-padding of 1. The number

of output channels of the 5 conv. layers are {64, 128, 256, 512, 512} in order. The max pooling layers have a square kernel of size 2 with a stride of 2 in each direction. The FC layers have sizes {128, 128, 2} in order. We train the network until convergence to lower the minimum squared error (MSE) loss using Adam [30] with an initial learning rate of 3×10^{-3} and a batch size of 128. The model from [17] has a separate audio classifier for stopping. This classifier has the same architecture as the goal prediction model except for last FC layer which just has 1 output unit and a sigmoid activation. The stopping classifier is trained to minimize the binary cross entropy (BCE) loss with an initial learning rate of 3×10^{-5}.

During navigation, the agent predicts the AudioGoal location after every N time steps if the predicted location is not reached before that. The agent stops if the episode times out after 500 time steps or the stopping classifier predicts the stop action. The original paper sets $N = 1$ but we found that 1-step predictions are very reactive in nature, *i.e.* the agent keeps going back and forth or keeps turning while standing at the same location. This leads to a very low performance in the realistic Replica test environments [8]. We improve the prediction stability and the navigation performance by predicting after every $N = 30$ steps for the periodic heard and unheard sound experiments and $N = 20$ steps for the time-varying sound experiment, where N is chosen through a hyperparameter search on the validation split.

6.8.2 Audio Direction Follower

For this baseline, we train a CNN model to predict the direction of arrival (DoA) of the sound with the binaural spectrograms as input. We collect the ground truth DoAs by using the ambisonic room impulse responses (RIR) sampled at 44.1 kHz from [8]. The first sound samples from the RIRs that correspond to the direct sound are used to build a circular intensity map around the agent at the height of the agent's ears. The circular map is discretized into 36 bins where each bin is equal to $360°/36 = 10°$. We select the bin with the maximum intensity in this map to approximate the DoA of the direct sound.

Our CNN for classifying the DoAs from binaural spectrograms has an architecture similar to that of ResNet-18 [26]. The feature extractor of the CNN has 2 convolutional (conv.) layers at the start, each with a square kernel of size 3, a symmetric zero-padding of 1 and a symmetric stride value of 1 and 2 respectively. These starting conv. layers are followed by 4 residual blocks each of which downsamples its input by a factor of 2. The residual blocks use the configuration from [26] but makes the first 2 convolutions in the block strided with a symmetric stride value of 2. Every conv. layer in the network has a batch-normalization of 10^{-5} and a ReLU activation function. The number of

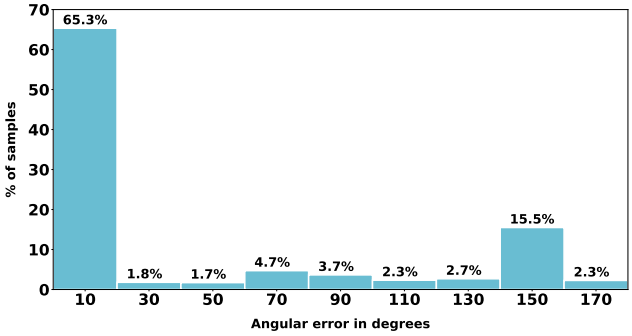


Figure 5: The DoA model predicts the audio direction based on binaural audio spectrograms with an error of less than 10 degrees for 65% of the samples.

output channels of the first 2 conv. layers and the 4 residual blocks are {64, 64, 64, 64, 128, 256} in order. The CNN ends with a single fully-connected layer with 36 output units for classifying a binaural spectogram pair into one of the 36 classes where each class corresponds to a $10°$ DoA bin. The network is trained until convergence to lower the negative log-likelihood (NLL) loss with Adam [30], a batch size of 128 and an initial learning rate of 3×10^{-3} for the periodic heard sound experiments and 3×10^{-4} for the unheard and the time-varying sounds experiments.

On a test split, the DoA model achieves a classification accuracy of 63.37% on the heard sound, 54.64% on the unheard sounds and 61.23% on the time-varying sound setting. Figure 5 shows a histogram of the percentage of samples for different angular errors between the predicted DoAs and the ground truth in the heard sound setting. We see that for around 65% of the samples the model has an absolute angular error lower than $10°$.

During navigation, the agent predicts the DoA using the previous model and moves to an intermediate goal that is 4 steps away (2 meters) in that direction. The intermediate goal is recomputed after every 4 time steps as long as the agent has not reached the audio goal. If the predicted intermediate goal does not lie at a navigable location in the agent's geometric map (G_t), it executes a random action. The agent uses the same audio classifier as the Gan et al. [17] baseline for stopping.

6.8.3 Frontier Waypoints

Similar to the Audio Direction Follower baseline, this agent also predicts the DoA of the direct sound but moves to the nearest frontier [5, 48] in that direction instead of an intermediate goal that is four steps away. To improve this baseline, we enforce an additional constraint so that the frontier point is always at least 3 steps (1.5 meters) away to ensure that the agent does not make reactive predictions and keep going

back and forth between the same two points in the environment. If there is no frontier point along the predicted DoA (a common case when the agent starts off), then the agent simply moves 3 steps in that direction. If the agent finds the next frontier to be N steps away, then the agent does not predict another frontier waypoint until it reaches the current one or $2N$ time steps have passed. For stopping, the agent uses the same stopping classifier as the previous baselines.