Measuring Acoustics with Collaborative Multiple Agents*

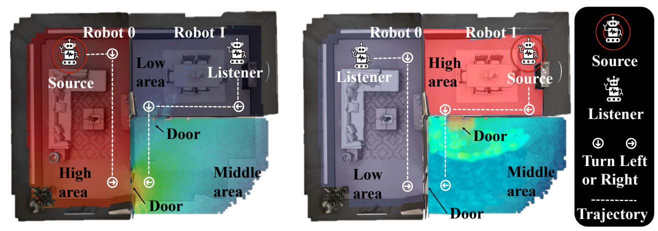
Yinfeng Yu[1,6] , **Changan Chen**[2] , **Lele Cao**[1,3] , **Fangkai Yang**[4] and **Fuchun Sun**[1,5,†]
[1]Beijing National Research Center for Information Science and Technology, State Key Lab on Intelligent Technology and Systems, Department of Computer Science and Technology, Tsinghua University
[2]University of Texas at Austin
[3]Motherbrain, EQT Group
[4]Microsoft Research
[5]THU-Bosch JCML Center
[6]College of Information Science and Engineering, Xinjiang University
yyf17@mails.tsinghua.edu.cn, changan@cs.utexas.edu, lele.cao@eqtpartners.com, fangkai.yang@microsoft.com, fcsun@mail.tsinghua.edu.cn

Abstract

As humans, we hear sound every second of our life. The sound we hear is often affected by the acoustics of the environment surrounding us. For example, a spacious hall leads to more reverberation. Room Impulse Responses (RIR) are commonly used to characterize environment acoustics as a function of the scene geometry, materials, and source/receiver locations. Traditionally, RIRs are measured by setting up a loudspeaker and microphone in the environment for all source/receiver locations, which is time-consuming and inefficient. We propose to let two robots measure the environment's acoustics by actively moving and emitting/receiving sweep signals. We also devise a collaborative multi-agent policy where these two robots are trained to explore the environment's acoustics while being rewarded for wide exploration and accurate prediction. We show that the robots learn to collaborate and move to explore environment acoustics while minimizing the prediction error. To the best of our knowledge, we present the very first problem formulation and solution to the task of collaborative environment acoustics measurements with multiple agents.

1 Introduction

Sound is critical for humans to perceive and interact with the environment. Before reaching our ears, sound travels via different physical transformations in space, such as reflection, transmission and diffraction. These transformations are characterized and measured by a Room Impulse Response (RIR) function [Välimäki *et al.*, 2016]. RIR is the transfer function between the sound source and the listener (microphone). Convolving the anechoic sound with RIR will get the sound with reverberation [Cao *et al.*, 2016]. RIR is utilized in many applications such as sound rendering [Schissler *et al.*, 2014], sound source localization [Tang *et al.*, 2020], audio-visual matching [Chen *et al.*, 2022], and audio-visual navigation [Chen *et al.*, 2020; Chen *et al.*, 2021b; Chen *et al.*, 2021a; Yu *et al.*, 2022b]. For example, to achieve clear speech in a concert hall, one might call for a sound rendering that drives more acoustic reverberation while keeping auditoriums with fewer reverberation [Mildenhall *et al.*, 2022]. The key is to measure RIR at different locations in the hall. However, RIR measuring is time-consuming due to the large number of samples to traverse. To illustrate, in a 5×5 m^2 room with a spatial resolution of 0.5m, the number of measurable points is 11×11=121. The source location (omnidirectional) can sample one of these 121 points. Assuming a listener with four orientations (0, 90, 180, 270), this listener can choose from 121 points with four directions for each chosen point. So, the number of source-listener pairs becomes 121×121×4=58,564. Assuming the sampling rate, duration and precision of binaural RIR is 16K, 1 second and float32 respectively, one RIR sample requires $2 \times 16000 \times 4$ Bytes = 128KB from computer storage (memory). The entire room would take up to $58,564 \times 128$ KB ≈ 7.5 GB. Moreover, it also means that one has to move the source/listener devices 58,564 times and performs data sending/receiving for each point.



(a) Phase One (b) Phase Two

Figure 1: Learn to measure environment acoustics with two collaborative robots. The background color indicates sound intensity ("High", "Middle" and "Low" areas). Each step (one step per second) embodies three steps: 1) robot 0 emits a sound, and robot 1 receives the sound; 2) robot 1 emits the sound, and robot 0 receives the sound; 3) two robots make a movement following their learned policies. This process repeats until reaching the maximum number of time steps.

*The full paper with appendix together with source code can be found at https://yyf17.github.io/MACMA.

†Corresponding author: Fuchun Sun.

There are some attempts to solve the challenge of storage: FAST-RIR [Ratnarajah *et al.*, 2022b] relies on handcrafted features, source (emitter) and listener (receiver) locations to generate RIR while being fidelity agnostic; MESH2IR [Ratnarajah *et al.*, 2022a] uses the scene's mesh and source/listener locations to generate RIR while ignoring the measurement cost; Neural Acoustic Field (NAF) [Luo *et al.*, 2022] tries to learn the parameters of the acoustic field, but its training time and model storage cost grow linearly with the number of environments [Majumder *et al.*, 2022]. Some work [Singh *et al.*, 2021] suggests that storing the original RIR data of the sampled points is optional, and only the acoustic field parameters must be stored. However, given a limited number of action steps, it is challenging to model the acoustic field.

To overcome the aforementioned challenges, we propose MACMA (Measuring Acoustics with Collaborative Multiple Agents) which is illustrated in Figure 1). Both agents, one source (emitter) and one listener (receiver), learn a motion policy to perform significance sampling of RIR within any given 3D scene. The trained agents can move (according to the learned motion policy) in any new 3D scene to predict the RIR of that new scene. To achieve that, we design two policy learning modules: the RIR prediction module and the dynamic allocation module of environment reward. In Appx. B, we explore the design of environmental reward and based on this, and we further propose a reward distribution module to learn how to efficiently distribute the reward obtained at the current step, thereby incentivizing the two agents to learn to cooperate and move. To facilitate the convergence of optimization, we design loss functions separately for the policy learning module, the RIR prediction module, and the reward allocation module. Comparative experiments and ablation experiments are performed on two datasets Replica [Straub *et al.*, 2019] and Matterport3D [Chang *et al.*, 2017], verifying the effectiveness of the proposed solution. To the best of our knowledge, this work is the first RIR measurement method using two collaborative agents. The main contributions of this work are:

- we propose a new setting for planning RIR measuring under finite time steps and a solution to measure the RIR with two-agent cooperation in low resource situations;
- we design a novel reward function for the multi-agent decomposition to encourage coverage of environment acoustics;
- we design evaluation metrics for the collaborative measurement of RIR, and we experimentally verify the effectiveness of our model.

2 Related Work

RIR generation. Measuring the RIR has been of long-standing interest to researchers [Cao *et al.*, 2016; Savioja and Svensson, 2015]. Traditional methods for generating RIR include statistical based methods [Schissler *et al.*, 2014; Tang *et al.*, 2022] and physics-based methods [Mehra *et al.*, 2013; Taylor *et al.*, 2012]. However, they are computationally prohibitive. Recent methods estimate the acoustics of RIR by parameters to generate RIR indirectly [Masztalski *et al.*, 2020; Diaz-Guerra *et al.*, 2021; Ratnarajah *et al.*, 2021]. Although these methods are flexible in extracting different acoustic cues, their predictions are independent of the source and receiver's exact locations, making them unsuitable for scenarios where the mapping between RIR and locations of source and receiver is important (e.g. sound source localization and audiovisual navigation). FAST-RIR [Ratnarajah *et al.*, 2022b] is a GAN-based RIR generator that generates a large-scale RIR dataset capable of accurately modeling source and receiver locations, but for efficiency, they rely on handcrafted features instead of learning them, which affects generative fidelity [Luo *et al.*, 2022]. Neural Acoustic Field (NAF) [Luo *et al.*, 2022] addresses the issues of efficiency and fidelity by learning an implicit representation of RIR [Mildenhall *et al.*, 2022], and by introducing global features and local embeddings. However, NAF cannot generalize to new environments, and its training time and model storage cost grows linearly with the number of environments [Majumder *et al.*, 2022]. The recently proposed MESH2IR [Ratnarajah *et al.*, 2022a] is an indoor 3D scene IR generator that takes the scene's mesh, listener positions, and source locations as input. MESH2IR [Ratnarajah *et al.*, 2022a] and FAST-RIR [Ratnarajah *et al.*, 2022b] assume that the environment and reverberation characteristics have been given, hence they only consider the fitting for the existing dataset, and ignore the measurement cost. However, our model considers how two moving agents collaborate to optimize RIR measuring from the aspects of time consumption, coverage, accuracy, etc. It is worth mentioning that our model addresses multiple scenarios, so that it is more suitable for generalizing to unseen environments.

Audio spatialization. Binaural audio generation methods comprise converting mono audio to binaural audio using visual information in video [Garg *et al.*, 2021], utilizing spherical harmonics to generate binaural audio from mono audio for training [Xu *et al.*, 2021], and generate binaural audio from video [Ruohan and Kristen, 2019]. Using 360 videos from YouTube to generate 360-degree ambisonic sound [Morgado *et al.*, 2018] is a higher-dimensional audio spatialization. Alternatively, [Rachavarapu *et al.*, 2021] directly synthesize spatial audio. Audio spatialization has a wide range of practical applications, such as object/speaker localization [Jiang *et al.*, 2022], speech enhancement [Michelsanti *et al.*, 2021], speech recognition [Shao *et al.*, 2022], etc. Although all the above works use deep learning, our work is fundamentally different in that we propose to model binaural RIR measuring as a decision process (of two moving agents that learns to plan measurement) using time-series states as input.

Audio-visual learning. [Majumder *et al.*, 2022] harnesses the synergy of egocentric visual and echogenic responses to infer ambient acoustics to predict RIR. The advancement of audiovisual learning has good applications in many tasks, such as audiovisual matching [Chen *et al.*, 2022], audiovisual source separation [Majumder and Grauman, 2022] and audiovisual navigation [Chen *et al.*, 2020; Dean *et al.*, 2020; Gan *et al.*, 2020; Gan *et al.*, 2022; Yu *et al.*, 2022a; Yu *et al.*, 2023]. There are also work to use echo responses with vision to learn better spatial representations [Gao *et al.*, 2020], infer depth [Christensen *et al.*, 2020], or predict floor plans of 3D

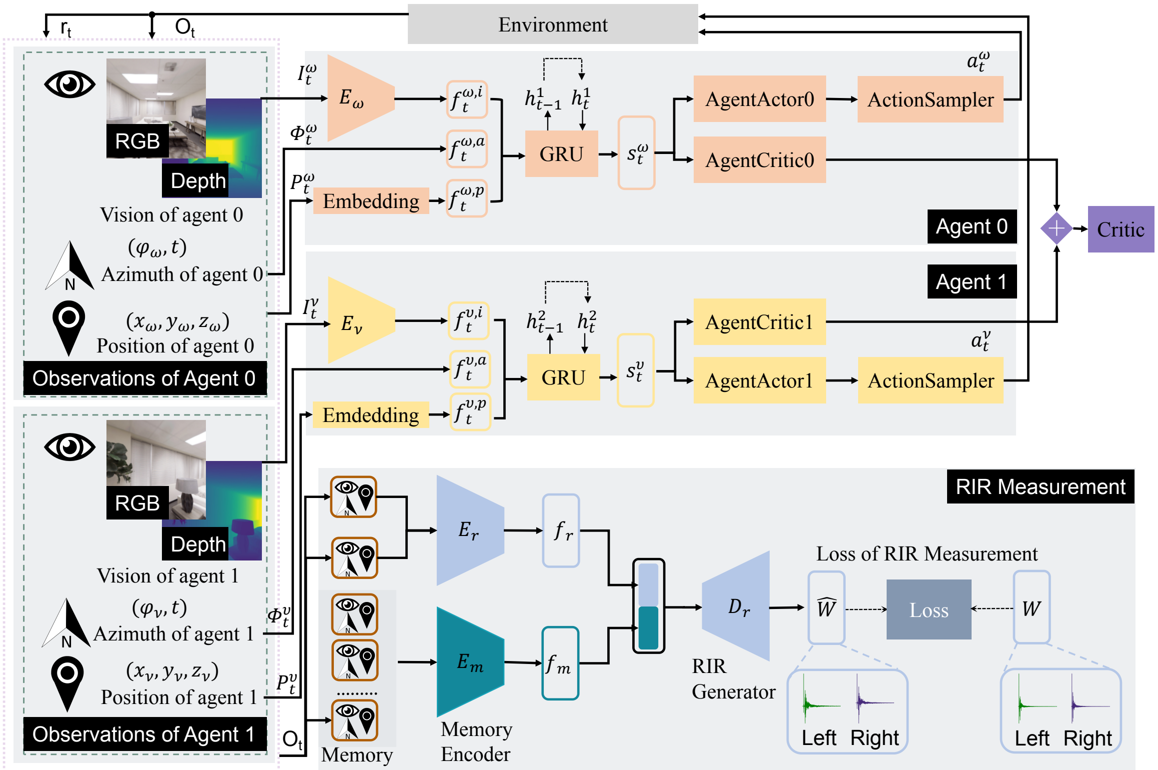


Figure 2: The MACMA architecture: the agent 0 and the agent 1 first learn to encode observations as s_t^ω and s_t^ν respectively using encoder E_ω and E_ν, which are fed to actor-critic networks to predict the next action a_t^ω and a_t^ν. The RIR Measurement learns how to predict room impulse response $\hat{W}_t$ guided by ground truth W_t.

environments [Purushwalkam *et al.*, 2021]. The closest work to ours is audio-visual navigation, but audio-visual navigation has navigation goals, but the agents in our setup have no clear navigation destinations.

Multi-agent learning. There are two types of collaborative multi-agents: value decomposition based methods [Rashid *et al.*, 2018; Rashid *et al.*, 2020] and actor-critic [Foerster *et al.*, 2018; Lowe *et al.*, 2017] based methods. The centralized training with decentralized execution (CTDE) paradigm [Wang *et al.*, 2021] has recently attracted attention for its ability to address non-stationarity while maintaining decentralized execution. Learning a centralized critic with decentralized actors (CCDA) is an efficient approach that exploits the CTDE paradigm. Multi-agent deep deterministic policy gradient (MADDPG) and counterfactual multi-agent (COMA) are two representative examples. In our design, we have a centralized critic (named Critic) with decentralized actors (named AgentActor0 and AgentActor1). But our task differs from all the above multi-agent learning methods in that the agents in our scenario are working on a collaborative task, while previous multi-agent learning research has mainly focused on competitive tasks.

3 The Proposed Approach

Our model MACMA has two collaborative agents moving in a 3D environment in Figure 1, using vision, position, and azimuth to measure the RIR. The proposed model mainly consists of three parts: agent 0, agent 1, and RIR measurement (see Figure 2). Given egocentric vision, azimuth, and position inputs, our model encodes these multi-modal cues to 1) determine the action for agents and evaluate the action taken by the agents for policy optimization, 2) measure the room impulse response and evaluate the regression accuracy for the RIR generator, and 3) evaluate the trade-off between the agents' exploration and RIR measurement. The two agents repeat this process until the maximal steps have been reached.

Specifically, at each step t (cf. Figure 2), the robots receive the current observation of their own O_t^ω and O_t^ν respectively, where $O_t^\omega = (I_t^\omega, \Phi_t^\omega, P_t^\omega)$, $O_t^\nu = (I_t^\nu, \Phi_t^\nu, P_t^\nu)$, $I_t^\omega = (I_t^{\omega,rgb}, I_t^{\omega,depth})$ and $I_t^\nu = (I_t^{\nu,rgb}, I_t^{\nu,depth})$ are egocentric visions for robot 0 and robot 1 respectively. $\Phi_t^\omega = (\varphi_t^\omega, t)$ and $\Phi_t^\nu = (\varphi_\nu, t)$ are azimuths for robot 0 and robot 1 respectively. $P_t^\omega = (x_\omega, y_\omega, z_\omega)$ and $P_t^\nu = (x_v, y_v, z_v)$ are positions for robot 0 and robot 1 respectively. I_t^ω or I_t^ν denotes the current visual input that can be RGB ($128\times128\times3$ pixels) and/or depth (with a dimension of $128\times128\times1$) image[1], Φ_t^ω and Φ_t^ν are 2D vector with time. P_t^ω and P_t^ν are 3D vector. Although there exists a *navigability graph* (with nodes and edges) of the environment, this graph is hidden from the robot, hence the robots must learn from the accumulated observations O_t^ω and O_t^ν to understand the geome-

[1]Both RGB and depth images capture the 90-degree field of view in front of the navigating robot.

try of the scene. At each step, the robot at a certain node A can only move to another node B in the navigability graph if 1) an edge connects both nodes, and 2) the robot is facing node B. The viable robotic action space is defined as $\mathcal{A}$ ={`MoveForward`, `TurnLeft`, `TurnRight`, `Stop`}, where the `Stop` action should be executed when the robot completes the task or the number of the robot's actions reach the maximum number of steps. The overall goal is to **predict RIR in new scene accurately and explore widely**.

3.1 Problem Formulation

We denote agent 0 and agent 1 with superscript ω and ν, respectively. The game $\mathcal{M} = (\mathcal{S}, (\mathcal{A}^\omega, \mathcal{A}^\nu), \mathcal{P}, (\mathcal{R}^\omega, \mathcal{R}^\nu))$ consists of state set $\mathcal{S}$, action sets ($\mathcal{A}^\omega$, $\mathcal{A}^\nu$), a joint state transition function $\mathcal{P}: \mathcal{S} \times \mathcal{A}^\omega \times \mathcal{A}^\nu \rightarrow \mathcal{S}$, and the reward functions $\mathcal{R}^\omega : \mathcal{S} \times \mathcal{A}^\omega \times \mathcal{A}^\nu \times \mathcal{S} \rightarrow \mathbb{R}$ for agent 0 and $\mathcal{R}^\nu : \mathcal{S} \times \mathcal{A}^\omega \times \mathcal{A}^\nu \times \mathcal{S} \rightarrow \mathbb{R}$ for agent 1. Each player wishes to maximize their discounted sum of rewards. r is the reward given by the environment at every time step in an episode. MACMA is modeled as a multi-agent [Sunehag *et al.*, 2018; Rashid *et al.*, 2018] problem involving two collaborating players sharing the same goal:

$$\begin{aligned}
&\min.\ \mathcal{L} \quad \text{s.t. } \pi^\star = (\pi^{\star,\omega}, \pi^{\star,\nu}) = \underset{\pi^\omega \in \Pi^\omega,\ \pi^\nu \in \Pi^\nu}{\arg\max}\ G(\pi^\omega, \pi^\nu, r)\\
&\text{where} \quad G(\pi^\omega, \pi^\nu, r) = w^\omega G(\pi^\omega, r) + w^\nu G(\pi^\nu, r),\\
&\qquad G(\pi^\omega, r) = \textstyle\sum_{t=0}^{T-1} \gamma^t r_t \rho^\omega, \quad G(\pi^\nu, r) = \textstyle\sum_{t=0}^{T-1} \gamma^t r_t \rho^\nu,\\
&\qquad \rho^\omega = (1-\rho)/2, \quad \rho^\nu = (1-\rho)/2,\\
&\qquad w^\omega > 0, \quad w^\nu > 0, \quad 0 \le \rho \le 1 \quad \text{or} \quad \rho = -1.0,
\end{aligned} \tag{1}$$

where the loss $\mathcal{L}$ is defined in Equation 5. $G(\pi^\omega, \pi^\nu, r)$ is the expected joint rewards for agent 0 and agent 1 as a whole. $G(\pi^\omega, r)$ and $G(\pi^\nu, r)$ are the discounted and cumulative rewards for agent 0 and agent 1, respectively. w^ω and w^ν denote the constant cumulative rewards balance factors for agent 0 and agent 1, respectively. ρ^ω and ρ^ν are immediate reward contributions for agent 0 and agent 1, respectively. ρ is a constant (throughout the training) reward allocation parameter. Inspired by *Value Decomposition Networks* (VDNs) [Sunehag *et al.*, 2018] and QMIX [Rashid *et al.*, 2018], we construct the objective function $G(\pi^\omega, \pi^\nu, r)$ in Equation 1 by combining the non-negative partial reciprocal constraint respond to $G(\pi^\omega, r)$ and $G(\pi^\nu, r)$ (see theoretical details in Appx. A.1).

Agent 0, agent 1 and their optimization. The agent 0 and agent 1 receive the current observation $O_t^\omega = (I_t^\omega, \Phi_t^\omega, P_t^\omega)$ and $O_t^\nu = (I_t^\nu, \Phi_t^\nu, P_t^\nu)$ at the t-th step. The visual (I_t^ω and I_t^ν) part is encoded into a visual feature vector using a CNN encoder: $f_t^{\omega, i}$ and $f_t^{\nu, i}$ (E_ω for agent 0 and E_ν for agent 1). Visual CNN encoders E_ω and E_ν are constructed in the same way (from the input to output layer): `Conv8x8`, `Conv4x4`, `Conv3x3` and a 256-dim linear layer; ReLU activations are added between any two neighboring layers. P_t^ω and P_t^ν are embedded by an embedding layer and encoded into feature vectors $f_t^{\omega, p}$ and $f_t^{\nu, p}$, respectively. Then, we concatenate the two vectors together with $f_t^{\omega, a}$ (Φ_t^ω) and $f_t^{\nu, a}$ (Φ_t^ν) to obtain the global observation embedding $e_t^\omega = [f_t^{\omega, i}, f_t^{\omega, a}, f_t^{\omega, p}]$ and $e_t^\nu = [f_t^{\nu, i}, f_t^{\nu, a}, f_t^{\nu, p}]$. We transform the observation embeddings to state representations using a gated recurrent unit (GRU), $s_t^\omega = \text{GRU}(e_t^\omega, h_{t-1}^1)$. We adopt a similar procedure to obtain $s_t^\nu = \text{GRU}(e_t^\nu, h_{t-1}^2)$. The state vectors ($s_t^\omega$ for agent 0 and s_t^ν for agent 1) are then fed to an actor-critic network to 1) predict the conditioned action probability distribution $\pi_{\theta_1^\omega}(a_t^\omega | s_t^\omega)$ for agent 0 and $\pi_{\theta_1^\nu}(a_t^\nu | s_t^\nu)$ for agent 1, and 2) estimate the state value $V_{\theta_2^\omega}(s_t^\omega, r_t^\omega)$ for agent 0 and $V_{\theta_2^\omega}(s_t^\nu, r_t^\nu)$ for agent 1. The actor and critic are implemented with a single linear layer parameterized by θ_1^ω, θ_1^ν, θ_2^ω, and θ_2^ν, respectively. For the sake of conciseness, we use $\boldsymbol{\theta}$ to denote the compound of θ_1^ω, θ_1^ν, θ_2^ω, and θ_2^ν hereafter. The action samplers in Figure 2 sample the actual action (i.e. a_t^ω for agent 0 and a_t^ν for agent 1) to execute from $\pi_{\theta_1^\omega}(a_t^\omega | s_t^\omega)$ for agent 0 and $\pi_{\theta_1^\nu}(a_t^\nu | s_t^\nu)$ for agent 1, respectively. Both agent 0 and agent 1 optimize their policy by maximizing the expected cumulative rewards $\text{G}(\pi^\omega, r)$ and $\text{G}(\pi^\nu, r)$ respectively in a discounted form. The Critic module evaluates the actions taken by agent 0 and agent 1 to guide them to take an improved action at the next time step. The loss of $\mathcal{L}^m$ is formulated as Equation 2.

$$\mathcal{L}^m = w_m^\omega \cdot \mathcal{L}_m^\omega + w_m^\nu \cdot \mathcal{L}_m^\nu, \tag{2}$$

where $\mathcal{L}_m^\omega$ and $\mathcal{L}_m^\nu$ are motion loss for agent 0 and agent 1 respectively, w_m^ω, w_m^ν are hyperparameters. The loss $\mathcal{L}_m^j$ is defined as

$$\begin{aligned}
\mathcal{L}_m^j = &\sum 0.5 \left(\hat{V}_{\theta^j}(s) - V^j(s) \right)^2 \\
&- \sum \left[\hat{A}^j \log(\pi_{\theta^j}(a|s)) + \beta \cdot H(\pi_{\theta^j}(a|s)) \right],
\end{aligned} \tag{3}$$

where $j \in \{\omega, \nu\}$, and the estimated state value of the target network for j is denoted as $\hat{V}_{\theta^j}(s)$. $\text{V}^j(s) = \max_{a \in \mathbb{A}^j} \mathbb{E}[r_t + \gamma \cdot \text{V}^j(s_{t+1}) \mid s_t = s]$. The advantage for a given length-T trajectory is: $\hat{\text{A}}_t^j = \sum_{i=t}^{T-1} \gamma^{i+2-t} \cdot \delta_i^j$, where $\delta_t^j = r_t + \gamma \cdot \text{V}^j(s_{t+1}) - \text{V}^j(s_t)$. $H(\pi_{\theta^j}(a \mid s))$ is entropy of $\pi_{\theta^j}(a \mid s)$. We collectively denote all the weights in Figure 2 except the above actor-critic network for agent 0 and agent 1 as $\boldsymbol{\Omega}$ hereafter for simplicity.

RIR measurement and its regression. We encode the observations O_t^ω and O_t^ν with encoder E_r, and the output of the encoder E_r is f_r. The historical observations $O_{t+1-\kappa}^\omega, O_{t+1-\kappa}^\nu, \cdots, O_{t-1}^\omega, O_{t-1}^\nu, O_t^\omega, O_t^\nu$ are sorted in the memory, and are encoded by E_m outputting f_m, where κ is the length of the memory bank. Then, f_r and f_m are concatenated. The predicted RIR $\hat{W}_t$ is obtained using RIR generator D_r. For more details on the structure of E_r, E_m and D_r, please refer to Appx. A.2. RIR measurement is learned with the ground truth RIR W_t. $\mathcal{L}^\xi$ denote the loss of RIR measurement. $\mathcal{L}^\xi$ are formulated as

$$\begin{aligned}
&\mathcal{L}^\xi = (1 - w^{\text{MSE}}) \cdot 10 \cdot \mathcal{L}^{\text{STFT}} + w^{\text{MSE}} \cdot 4464.2 \cdot \mathcal{L}^{\text{MSE}},\\
&\mathcal{L}^{\text{STFT}} = \sum \Delta(W_t, \hat{W}_t), \quad \mathcal{L}^{\text{MSE}} = \sum \text{MSE}(W_t, \hat{W}_t),
\end{aligned} \tag{4}$$

where $\hat{W}_t$ is the predicted RIR from the RIR Measurement module. W_t is the ground truth RIR. $\Delta(W_t, \hat{W}_t)$ is STFT (Short-time Fourier transform) distance. It is calculated by Equation 8. 10 and 4464.2 are experimental parameters from grid search.

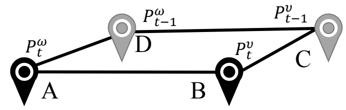


Figure 3: Demonstration of the current (A and B) and previous (C and D) positions of two robots. The above four coplanar points are denoted as Γ_{ABCD}.

The total evaluation. The Critic module is implemented with a linear layer. The total loss of our model is formulated as Equation 5. We minimize $\mathcal{L}$ following Proximal Policy Optimization (PPO) [Schulman *et al.*, 2017].

$$\mathcal{L} = w_m \cdot \mathcal{L}^m + w_\xi \cdot \mathcal{L}^\xi, \tag{5}$$

where $\mathcal{L}^m$ is the loss component of motion for two agents, $\mathcal{L}^\xi$ is the loss component of room impulse response prediction, w_m and w_ξ are hyperparameters. The losses $\mathcal{L}^m$ and $\mathcal{L}^\xi$ are formulated in Equation 2 and Equation 4, respectively.

The design of environmental reward. It can be seen from Figure 3 that the Γ_{ABCD} is formed by the positions of the two agents at the current time step and the previous time step. r_t is the current step reward, which is calculated by the following Equation 6.

$$r_t = r_t^\xi + r_t^\zeta + r_t^\psi + r_t^\phi, \tag{6}$$

ξ denotes the prediction of room impulse response. ζ denotes coverage rate. ψ denotes the length of the perimeter of the convex hull. ϕ denotes the area of the perimeter of the convex hull.

Among them, r_t^ξ is the reward component in terms of measurement accuracy, which evaluates the improvement of the reward of the measurement accuracy of the current step and the reward of the measurement accuracy of the previous step. r_t^ξ is calculated by

$$r_t^\xi = \alpha^\xi \cdot (\xi_t - \xi_{t-1}), \quad \xi_t = -\Delta(W_t, \hat{W}_t), \tag{7}$$

where ξ_t is the measurement accuracy of the current step. As briefly explained before, $\Delta(W_t, \hat{W}_t)$ is the STFT distance that can be calculated by

$$\Delta(W_t, \hat{W}_t) = 0.5 \cdot \Theta(z, \hat{z}) + 0.5 \cdot \Xi(z, \hat{z}), \tag{8}$$

where z is the magnitude spectrogram of ground truth RIR W_t for the current time step, while $\hat{z}$ is the corresponding predicted variant. $\Theta(z, \hat{z})$ is the average loss of spectral convergence for z and $\hat{z}$; and $\Xi(z, \hat{z})$ is the log STFT magnitude loss. $\Theta(z, \hat{z})$ and $\Xi(z, \hat{z})$ are computed with

$$\Theta(z, \hat{z}) = \frac{\|z - \hat{z}\|_F}{\|z\|_F} \quad \text{and} \quad \Xi(z, \hat{z}) = \sum \left|\log(\frac{z}{\hat{z}})\right|, \tag{9}$$

where $\|\cdot\|_F$ is Frobenius Norm. $z = \Lambda(W_t) = \sqrt{y_r^2 + y_i^2}$, where y_r is real part of STFT transform[2] of W_t, y_i is an imaginary part of the result of the STFT transform of W_t. $\hat{z} = \Lambda(\hat{W}_t)$ is defined similarly to z, and the calculation process of both z and $\hat{z}$ are the same.

[2]The parameters for STFT transform are #FFT=1024, #shift =120, #window=600, window="Hamming window".

Algorithm 1 MACMA (Measuring Acoustics with Collaborative Multiple Agents)

Input: Environment $\mathcal{E}$, # updates M, # episode N, max time steps T.
Parameter: Stochastic policies π, initial actor-critic weights $\boldsymbol{\theta}_0$, initial other weights except for actor-critic weights $\boldsymbol{\Omega}_0$.
Output:Trained weights, $\boldsymbol{\theta}_M$ and $\boldsymbol{\Omega}_M$.

for i=1, 2, ... M **do**
  // Run policy $\pi_{\theta_{i-1}}$ for N episodes T time steps
  $\{(o_{t,i},\, h_{t-1,i},\, a_{t,i},\, r_{t,i})\} \leftarrow$ roll$(\mathcal{E}, \pi_{\boldsymbol{\theta}_{i-1}}, T)$
  Compute advantage estimates
  RIR prediction and environmental reward assignment
  // Optimize w.r.t. $\boldsymbol{\theta}$ and $\boldsymbol{\Omega}$
  $\boldsymbol{\theta}_i, \boldsymbol{\Omega}_i \leftarrow$ new $\boldsymbol{\theta}$ and $\boldsymbol{\Omega}$ from PPO algorithm w.r.t. minimizing Equation 5
end for

ζ_t is the coverage of the current step, which is the ratio of visited nodes (only one duplicate node is counted) to all nodes in the scene at time step t. We calculate r_t^ζ by

$$r_t^\zeta = \alpha^\zeta \cdot (\zeta_t - \zeta_{t-1}). \tag{10}$$

ψ_t and ϕ_t are respectively the perimeter and area of Γ_{ABCD} in Figure 3 at time step t. We calculate r_t^ψ and r_t^ϕ with

$$r_t^\psi = \alpha^\psi \cdot (\psi_t - \psi_{t-1}) \quad \text{and} \quad r_t^\phi = \alpha^\phi \cdot (\phi_t - \phi_{t-1}), \tag{11}$$

where $\alpha^\xi = 1.0$, $\alpha^\zeta = 1.0$, $\alpha^\psi = -1.0$ and $\alpha^\phi = 1.0$ are hyperparameters (see Appx. A.9).

Overall algorithm. The entire procedure of MACMA is presented as pseudo-code in Algorithm 1.

4 Experiments

We adopt the commonly used 3D environments collected using the SoundSpaces platform [Chen *et al.*, 2020] and Habitat simulator [Savva *et al.*, 2019]. They are publicly available as several datasets: Replica [Straub *et al.*, 2019], Matterport3D [Chang *et al.*, 2017] and SoundSpaces (audio) [Chen *et al.*, 2020]. Replica contains 18 environments in the form of grids (with a resolution of 0.5 meters) constructed from accurate scans of apartments, offices, and hotels. Matterport3D has 85 scanned grids (1-meter resolution) of indoor environments like personal homes. To measure the RIR in Replica [Straub *et al.*, 2019] and Matterport3D [Chang *et al.*, 2017], we let two agents move a certain number of steps (250 and 300 steps for Replica and Matterport3D, respectively) throughout the scene and plan a measuring path. At every time step, the two agents measure the RIR while moving. The experimental procedure contains several phases: a) we pretrain generator D_r under the setting $\mathcal{L} = \mathcal{L}^m$ (w_m = 1.0 and w_ξ = 0.0) with random policy for both agent 0 and agent 1 in the training split, b) we train and validate every baseline with the generator D_r fine-tune together in the training and validation split, c) we test every baseline in the test split. MACMA is benchmarked towards several baselines: **Random**, **Nearest neighbor**, **Occupancy** [Ramakrishnan *et al.*, 2020] and **Curiosity** [Pathak *et al.*, 2017]. **Random** uniformly samples one of three actions and executes

Stop when it reaches maximum steps. **Nearest neighbor** predict from closest experience (Appx. A.3). **Occupancy** orient to occupy more area, making the area of Γ_{ABCD} in Figure 3 larger. **Curiosity** strives to visit nodes that have not been visited already in the current episode. To evaluate different methods, we adopt the evaluation metrics **CR** (coverage rate), **PE** (prediction error),**WCR** (weighted coverage rate),**RTE** (RT60 Error) and **SiSDR** (scale-invariant signal-to-distortion ratio), among which **WCR** is the most important evaluation metric since it is a trade-off between encouraging high prediction accuracy and more exploration. **CR** is the ratio of the number of visited nodes by agents and the number of nodes in the current episode. $CR = {}^{N_v}/_{N_e}$, where N_v is the total number of unique nodes that two agents have visited together, and N_e is the total number of all individual nodes in the current episode. PE $= \Delta(W_t, \hat{W}_t)$, where W_t (Equation 8) is the ground truth RIR, $\hat{W}_t$ is the predicted RIR. WCR $= (1.0 - \lambda) * \text{CR} + \lambda * (1.0 - \text{PES})$, where PES stands for Scaled Prediction Error. PES $= 2/(1 + \exp(-\text{PE})) - 1.0$, where $0 \leq \lambda \leq 1.0$ is a hyper-parameter. **RTE** describes the difference between the ground truth RT60 value and the predicted one. SiSDR = $10 \log_{10} {}^{\|X_T\|^2}/_{\|X_E\|^2}$, where $\|X_E\|^2$ is the error vector and $\|X_T\|^2$ is the ground truth vector. We select the hyperparameters in grid search and the details are in Appx. A.4.

4.1 Experimental Results

Results are averages of 5 tests with different random seeds.

Quantitative comparison of the two datasets. Results are runs on two datasets under the experimental settings: α^{ξ}=1.0, α^{ζ}=1.0, α^{ψ}=-1.0, α^{ϕ}=1.0, κ=2, λ=0.1, ρ=-1.0. As seen from Table 1, on the Replica dataset, MACMA achieves the best results on the metrics WCR, PE and CR. But the Curiosity model has the best results on the metrics RTE and SiSDR. The Curiosity model encourages agents to visit more new nodes near them, which drive up the robots' exploration ability (it improves performance on metrics RTE and SiSDR) while reducing their team's performance (it reduces performance on the metric CR). The Occupancy model (ranks as the second over CR) motivates the exploratory ability of the entire group (the group of agent 0 and agent 1) but ignores their individual exploration performance (the ranks over the metrics of RTE and SiSDR lower than that of CR). MACMA combines the group exploration ability and the individual exploration ability, achieving a good trade-off between the two abilities, so that the group exploration ability of MACMA has increased by a large margin (e.g. over the CR metric) and finally won the championship on the WCR metric. On the Matterport3D dataset, MACMA achieves the best results on all metrics. As a result, we can conclude that MACMA quantitatively outperforms baselines over both datasets.

Qualitative comparison on exploration capability. Figure 4 shows the navigation trajectories of agent 0 and agent 1 for different algorithms by the end of a particular episode from the Replica (top row) and Matterport3D (bottom row) dataset. The light-gray areas in Figure 4 indicate the exploration field of the robots. We observe that MACMA tends to explore the most extensively compared to the other baselines. Particularly, there are three rooms in the entire scene in Replica, and MACMA is the only method that managed to traverse all three rooms using the same number of time steps as baselines.

Qualitative comparison on RIR prediction. We show the spectrograms generated by these models and from the ground truth in Figure 5. These binaural spectrograms with channel 0 and channel 1 last for one second (the x-axis is the time axis). The spectrogram of the RIR from both Random's and Occupancy's generation have fewer color blocks than the ground truth between 0.2 seconds and 0.4 seconds and more color blocks than the ground truth between 0.8 seconds and 1 second. The spectrogram of the RIR from the Nearest neighbor's generation has more colored regions than the ground truth spectrogram. The spectrogram of the RIR from Curiosity's prediction has fewer color blocks than the ground truth between 0.2 seconds and 0.4 seconds. At the same time, the spectrogram of the RIR from Curiosity's generation has more color blocks than the ground truth between 0.8 seconds and 1 second in the Replica dataset. And the spectrogram of the RIR from Curiosity's prediction has more colored regions than the ground truth spectrogram in the Matterport3D dataset. The spectrogram of the generated RIR from MACMA (Ours) is the closest to the ground truth spectrogram. In conclusion, from a qualitative human visual point of view, the spectral quality of the RIRs generated by our model is the best. Additionally, in Appx. A.5, we show that the RIR's quality in the waveform of the RIRs generated by our model is also superior.

4.2 Ablation Studies

Ablation on modality. Results are run on dataset Replica under the experimental settings of α^{ξ}=1.0, α^{ζ}=1.0, α^{ψ}=-1.0, α^{ϕ}=1.0, κ=2, λ=0.1, ρ=-1.0. As shown in Table 2, RGBD (vision with RGB images and Depth input) seems to be the best choice.

More ablations. We explore the relationship between modality importance, action selection, and RIR measurement accuracy in Appx. A.6 and Appx. A.7. We present the extension model MACMARA (MACMA with a dynamic Reward Assignment module) in Appx. B.5. More ablation studies on memory size κ and the reward component can be found in Appx. A.4.

5 Conclusion

In this work, we propose a novel task where two collaborative agents learn to measure room impulse responses of an environment by moving and emitting/receiving signals in the environment within a given time budget. To tackle this task, we design a collaborative navigation and exploration policy. Our approach outperforms several other baselines on the environment's coverage and prediction error. A known limitation is that we only explored the most basic setting, one listener (receiver), and one source (emitter), and did not study the settings with two or more listeners or sources. Another limitation of our work is that our current assessments are conducted in a virtual environment. It would be more meaningful to evaluate our method on real-world cases, such as a robot moving in a real house and learning to measure environmental acoustics collaboratively. Lastly, we have not considered

Model	Replica					Matterport3D				
	WCR (↑)	PE (↓)	CR (↑)	RTE (↓)	SiSDR (↑)	WCR (↑)	PE (↓)	CR (↑)	RTE (↓)	SiSDR (↑)
Random	0.3103	5.4925	0.3439	14.7427	20.3534	0.2036	5.5552	0.2254	23.5281	12.3042
Nearest neighbor	0.3444	5.4533	0.3817	14.0269	22.0135	0.2099	5.3342	0.2321	28.8765	15.2351
Occupancy	0.4464	3.7224	0.4907	12.5532	23.0666	0.2225	4.5327	0.2449	20.3399	18.3848
Curiosity	0.4327	3.4883	0.4742	**10.9565**	**23.8669**	0.2111	4.4255	0.2319	29.5572	20.0031
MACMA (Ours)	**0.6977**	**3.2509**	**0.7669**	13.8896	23.6501	**0.3030**	**4.0113**	**0.3327**	**15.9338**	**21.3187**

Table 1: The results of quantitative comparison between our proposed method (MACMA) and baselines.

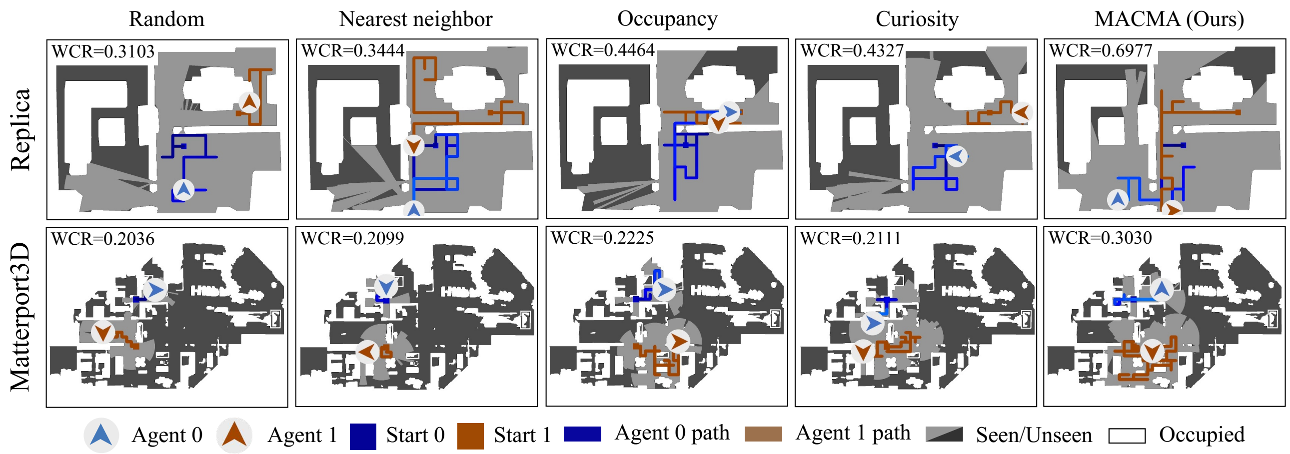


Figure 4: Visualization of the navigation trajectories by the end of a particular episode from Replica (top row) and Matterport3D (bottom row) dataset. Higher WCR values and bigger "seen" areas (colored in light-grey) indicate better performances.

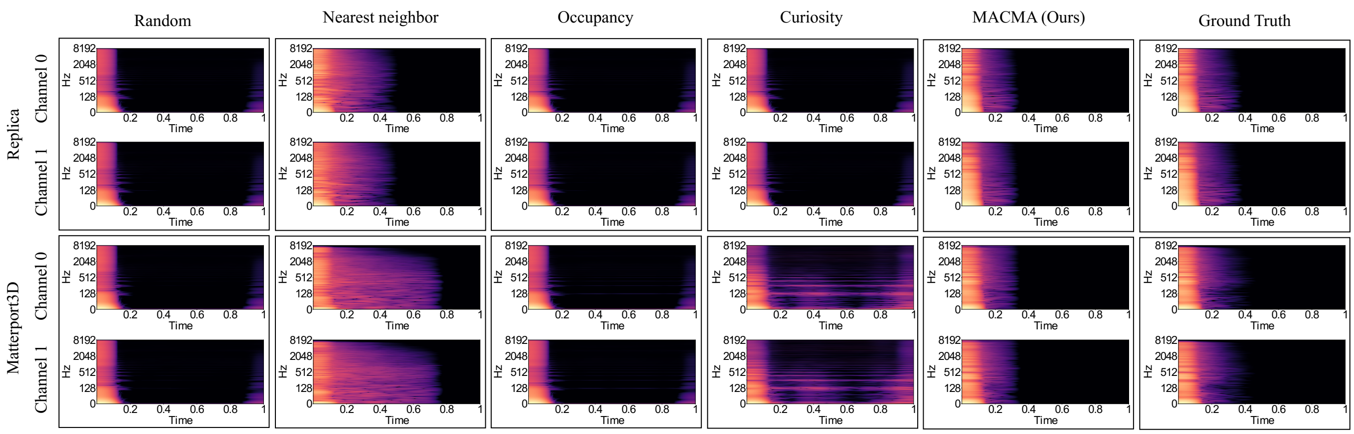


Figure 5: Qualitative comparison of RIR prediction (Binaural RIR with channel 0 and channel 1) in spectrogram from (a) Replica and (b) Matterport3D dataset. Every row is the result of one model except last one. The last row is the ground truth of RIR.

Vision	WCR (↑)	PE (↓)	CR (↑)	RTE (↓)	SiSDR (↑)
Blind	0.5020	3.4966	0.5512	14.2049	23.0903
RGB	0.5930	3.8204	0.6541	15.5897	**23.7713**
Depth	0.5068	3.4927	0.5566	29.6905	23.5089
RGBD	**0.6977**	**3.2509**	**0.7669**	**13.8896**	23.6501

Table 2: Ablation on modality.

semantic information about the scene in policy learning. Incorporating semantic information about the scene into policy learning would be more meaningful. The above three are left for future exploration.

Ethical Statement

This research follows IJCAI's ethics guidelines and does not involve human subjects or privacy issues with the dataset. The dataset is publicly available.

Acknowledgments

We thank the IJCAI 2023 reviewers for their insightful feedback. This work is supported by multiple projects, including the Sino-German Collaborative Research Project *Cross-modal Learning* (NSFC62061136001/DFG SFB/TRR169), the National Natural Science Foundation of China (No. 61961039), the National Key R&D Program of China (No. 2022ZD0115803), the Xinjiang Natural Science Foundation

(No. 2022D01C432, No. 2022D01C58, No. 2020D01C026, and No. 2015211C288), and partial funding from the THU-Bosch JCML Center.

References

[Cao *et al.*, 2016] Chunxiao Cao, Zhong Ren, Carl Schissler, Dinesh Manocha, and Kun Zhou. Interactive sound propagation with bidirectional path tracing. *ACM Trans. Graph.*, 35(6):180:1–180:11, 2016.

[Chang *et al.*, 2017] Angel X. Chang, Angela Dai, Thomas A. Funkhouser, Maciej Halber, Matthias Nießner, et al. Matterport3d: Learning from RGB-D data in indoor environments. In *International Conference on 3D Vision*, pages 667–676, 2017.

[Chen *et al.*, 2020] 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*, pages 17–36, 2020.

[Chen *et al.*, 2021a] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *CVPR*, pages 15516–15525, 2021.

[Chen *et al.*, 2021b] Changan Chen, Sagnik Majumder, Ziad Al-Halah, Ruohan Gao, Santhosh Kumar Ramakrishnan, and Kristen Grauman. Learning to set waypoints for audio-visual navigation. In *ICLR*, 2021.

[Chen *et al.*, 2022] Changan Chen, Ruohan Gao, Paul Calamia, and Kristen Grauman. Visual acoustic matching. In *CVPR*, pages 18836–18846, 2022.

[Christensen *et al.*, 2020] Jesper Haahr Christensen, Sascha Hornauer, and Stella X. Yu. Batvision: Learning to see 3d spatial layout with two ears. In *ICRA*, pages 1581–1587, 2020.

[Dean *et al.*, 2020] Victoria Dean, Shubham Tulsiani, and Abhinav Gupta. See, hear, explore: Curiosity via audio-visual association. In *NeurIPS*, 2020.

[Diaz-Guerra *et al.*, 2021] David Diaz-Guerra, Antonio Miguel, and José Ramón Beltrán. gpurir: A python library for room impulse response simulation with GPU acceleration. *Multim. Tools Appl.*, 80(4):5653–5671, 2021.

[Foerster *et al.*, 2018] Jakob N. Foerster, Gregory Farquhar, Triantafyllos Afouras, Nantas Nardelli, and Shimon Whiteson. Counterfactual multi-agent policy gradients. In *AAAI*, pages 2974–2982, 2018.

[Gan *et al.*, 2020] Chuang Gan, Yiwei Zhang, Jiajun Wu, Boqing Gong, and Joshua B. Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. In *ICRA*, pages 9701–9707, 2020.

[Gan *et al.*, 2022] Chuang Gan, Yi Gu, Siyuan Zhou, Jeremy Schwartz, Seth Alter, James Traer, Dan Gutfreund, Joshua B Tenenbaum, Josh H McDermott, and Antonio Torralba. Finding fallen objects via asynchronous audio-visual integration. In *CVPR*, pages 10513–10523, 2022.

[Gao *et al.*, 2020] Ruohan Gao, Changan Chen, Ziad Al-Halah, Carl Schissler, and Kristen Grauman. Visualechoes: Spatial image representation learning through echolocation. In *ECCV*, pages 658–676, 2020.

[Garg *et al.*, 2021] Rishabh Garg, Ruohan Gao, and Kristen Grauman. Geometry-aware multi-task learning for binaural audio generation from video. In *BMVC*, page 1, 2021.

[Jiang *et al.*, 2022] Hao Jiang, Calvin Murdock, and Vamsi Krishna Ithapu. Egocentric deep multi-channel audio-visual active speaker localization. In *CVPR*, pages 10534–10542, 2022.

[Lowe *et al.*, 2017] Ryan Lowe, Yi Wu, Aviv Tamar, Jean Harb, Pieter Abbeel, and Igor Mordatch. Multi-agent actor-critic for mixed cooperative-competitive environments. In *NeurIPS*, pages 6379–6390, 2017.

[Luo *et al.*, 2022] Andrew Luo, Yilun Du, Michael Tarr, Josh Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. *CoRR*, abs/2204.00628, 2022.

[Majumder and Grauman, 2022] Sagnik Majumder and Kristen Grauman. Active audio-visual separation of dynamic sound sources. In *ECCV*, pages 551–569, 2022.

[Majumder *et al.*, 2022] Sagnik Majumder, Changan Chen, Ziad Al-Halah, and Kristen Grauman. Few-shot audio-visual learning of environment acoustics. *CoRR*, abs/2206.04006, 2022.

[Masztalski *et al.*, 2020] Piotr Masztalski, Mateusz Matuszewski, Karol Piaskowski, and Michal Romaniuk. Storir: Stochastic room impulse response generation for audio data augmentation. In *Interspeech*, pages 2857–2861, 2020.

[Mehra *et al.*, 2013] Ravish Mehra, Nikunj Raghuvanshi, Lakulish Antani, Anish Chandak, Sean Curtis, and Dinesh Manocha. Wave-based sound propagation in large open scenes using an equivalent source formulation. *ACM Trans. Graph.*, 32(2):19:1–19:13, 2013.

[Michelsanti *et al.*, 2021] Daniel Michelsanti, Zheng-Hua Tan, Shi-Xiong Zhang, Yong Xu, Meng Yu, Dong Yu, and Jesper Jensen. An overview of deep-learning-based audio-visual speech enhancement and separation. *IEEE ACM Trans. Audio Speech Lang. Process.*, 29:1368–1396, 2021.

[Mildenhall *et al.*, 2022] Ben Mildenhall, Pratul P Srinivasan, Matthew Tancik, Jonathan T Barron, Ravi Ramamoorthi, and Ren Ng. Nerf: representing scenes as neural radiance fields for view synthesis. *Commun. ACM*, 65(1):99–106, 2022.

[Morgado *et al.*, 2018] Pedro Morgado, Nuno Vasconcelos, Timothy R. Langlois, and Oliver Wang. Self-supervised generation of spatial audio for 360° video. In *NeurIPS*, pages 360–370, 2018.

[Pathak *et al.*, 2017] Deepak Pathak, Pulkit Agrawal, Alexei A Efros, and Trevor Darrell. Curiosity-driven exploration by self-supervised prediction. In *ICML*, pages 2778–2787, 2017.

[Purushwalkam *et al.*, 2021] Senthil Purushwalkam, Sebastia Vicenc Amengual Gari, Vamsi Krishna Ithapu, Carl

Schissler, Philip Robinson, Abhinav Gupta, and Kristen Grauman. Audio-visual floorplan reconstruction. In *ICCV*, pages 1163–1172, 2021.

[Rachavarapu *et al.*, 2021] Kranthi Kumar Rachavarapu, Aakanksha, Vignesh Sundaresha, and A. N. Rajagopalan. Localize to binauralize: Audio spatialization from visual sound source localization. In *ICCV*, pages 1910–1919, 2021.

[Ramakrishnan *et al.*, 2020] Santhosh K. Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *ECCV*, pages 400–418, 2020.

[Rashid *et al.*, 2018] Tabish Rashid, Mikayel Samvelyan, Christian Schröder de Witt, Gregory Farquhar, Jakob N. Foerster, and Shimon Whiteson. QMIX: monotonic value function factorisation for deep multi-agent reinforcement learning. In *ICML*, pages 4292–4301, 2018.

[Rashid *et al.*, 2020] Tabish Rashid, Gregory Farquhar, Bei Peng, and Shimon Whiteson. Weighted QMIX: expanding monotonic value function factorisation for deep multi-agent reinforcement learning. In *NeurIPS*, 2020.

[Ratnarajah *et al.*, 2021] Anton Ratnarajah, Zhenyu Tang, and Dinesh Manocha. IR-GAN: room impulse response generator for far-field speech recognition. In *Interspeech*, pages 286–290, 2021.

[Ratnarajah *et al.*, 2022a] Anton Ratnarajah, Zhenyu Tang, Rohith Aralikatti, and Dinesh Manocha. MESH2IR: neural acoustic impulse response generator for complex 3d scenes. In *MM '22: The 30th ACM International Conference on Multimedia, Lisboa, Portugal, October 10 - 14, 2022*, pages 924–933. ACM, 2022.

[Ratnarajah *et al.*, 2022b] Anton Ratnarajah, Shi-Xiong Zhang, Meng Yu, Zhenyu Tang, Dinesh Manocha, and Dong Yu. Fast-rir: Fast neural diffuse room impulse response generator. In *ICASSP*, pages 571–575, 2022.

[Ruohan and Kristen, 2019] Gao Ruohan and Grauman Kristen. 2.5 d visual sound. In *CVPR*, pages 324–333, 2019.

[Savioja and Svensson, 2015] Lauri Savioja and U Peter Svensson. Overview of geometrical room acoustic modeling techniques. *The Journal of the Acoustical Society of America*, 138(2):708–730, 2015.

[Savva *et al.*, 2019] Manolis Savva, Jitendra Malik, Devi Parikh, Dhruv Batra, Abhishek Kadian, Oleksandr Maksymets, Yili Zhao, Erik Wijmans, Bhavana Jain, Julian Straub, Jia Liu, and Vladlen Koltun. Habitat: A platform for embodied AI research. In *ICCV*, pages 9338–9346, 2019.

[Schissler *et al.*, 2014] Carl Schissler, Ravish Mehra, and Dinesh Manocha. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *TOG*, 33(4):1–12, 2014.

[Schulman *et al.*, 2017] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *CoRR*, abs/1707.06347, 2017.

[Shao *et al.*, 2022] Yiwen Shao, S. Z., and Dong Yu. Multi-channel multi-speaker ASR using 3d spatial feature. In *ICASSP*, pages 6067–6071, 2022.

[Singh *et al.*, 2021] Nikhil Singh, Jeff Mentch, Jerry Ng, Matthew Beveridge, and Iddo Drori. Image2reverb: Cross-modal reverb impulse response synthesis. In *ICCV*, pages 286–295, 2021.

[Straub *et al.*, 2019] 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. *CoRR*, abs/1906.05797, 2019.

[Sunehag *et al.*, 2018] Peter Sunehag, Guy Lever, Audrunas Gruslys, Wojciech Marian Czarnecki, Vinicius Zambaldi, Max Jaderberg, Marc Lanctot, Nicolas Sonnerat, Joel Z Leibo, Karl Tuyls, et al. Value-decomposition networks for cooperative multi-agent learning based on team reward. In *AAMAS*, pages 2085–2087, 2018.

[Tang *et al.*, 2020] Zhenyu Tang, Lianwu Chen, Bo Wu, Dong Yu, and Dinesh Manocha. Improving reverberant speech training using diffuse acoustic simulation. In *ICASSP*, pages 6969–6973, 2020.

[Tang *et al.*, 2022] Zhenyu Tang, Rohith Aralikatti, Anton Jeran Ratnarajah, and Dinesh Manocha. GWA: A large high-quality acoustic dataset for audio processing. In *SIGGRAPH*, pages 36:1–36:9, 2022.

[Taylor *et al.*, 2012] Micah Taylor, Anish Chandak, Qi Mo, Christian Lauterbach, Carl Schissler, and Dinesh Manocha. Guided multiview ray tracing for fast auralization. *IEEE Trans. Vis. Comput. Graph.*, 18(11):1797–1810, 2012.

[Välimäki *et al.*, 2016] Vesa Välimäki, Julian Parker, Lauri Savioja, Julius O Smith, and Jonathan Abel. More than 50 years of artificial reverberation. In *Audio engineering society conference*, 2016.

[Wang *et al.*, 2021] Yihan Wang, Beining Han, Tonghan Wang, Heng Dong, and Chongjie Zhang. DOP: off-policy multi-agent decomposed policy gradients. In *ICLR*, 2021.

[Xu *et al.*, 2021] Xudong Xu, Hang Zhou, Ziwei Liu, Bo Dai, Xiaogang Wang, and Dahua Lin. Visually informed binaural audio generation without binaural audios. In *CVPR*, pages 15485–15494, 2021.

[Yu *et al.*, 2022a] Yinfeng Yu, Lele Cao, Fuchun Sun, Xiaohong Liu, and Liejun Wang. Pay self-attention to audio-visual navigation. In *BMVC*, 2022.

[Yu *et al.*, 2022b] Yinfeng Yu, Wenbing Huang, Fuchun Sun, Changan Chen, Yikai Wang, and Xiaohong Liu. Sound adversarial audio-visual navigation. In *ICLR*, 2022.

[Yu *et al.*, 2023] Yinfeng Yu, Lele Cao, Fuchun Sun, Chao Yang, Huicheng Lai, and Wenbing Huang. Echo-enhanced embodied visual navigation. *Neural Computation*, 35(5):958–976, 2023.