The Dissertation Committee for Changan Chen
certifies that this is the approved version of the following dissertation:

4D Audio-Visual Learning: A Visual Perspective of Sound Propagation and Production

Committee:

Dr. Kristen Grauman, Supervisor

Dr. David Harwath

Dr. Yuke Zhu

Dr. Andrea Vedaldi

Dr. Dinesh Manocha

4D Audio-Visual Learning: A Visual Perspective of Sound Propagation and Production

by

Changan Chen

Dissertation

Presented to the Faculty of the Graduate School of

The University of Texas at Austin

in Partial Fulfillment

of the Requirements

for the Degree of

Doctor of Philosophy

The University of Texas at Austin

August 2024

Acknowledgments

This thesis could not have been completed without the invaluable support and guidance of many mentors, colleagues, friends, and family members, each of whom has contributed immensely to my journey.

First and foremost, my profound gratitude goes to my advisor, Kristen Grauman. Her passion for research, deep engagement with every project, extensive knowledge, meticulous attention to detail, and genuine care for her students have been truly inspiring. Kristen is undoubtedly the ideal advisor, and under her guidance, I have honed my ability to conduct rigorous and innovative research.

I extend heartfelt thanks to my committee members and collaborators. A special mention goes to Ziad Al-Halah, whose mentoring and help was crucial during the initial stages of my PhD. I am also grateful to David Harwath for his insightful contributions to my understanding of audio and speech processing. My time as a visiting researcher at Facebook AI Research (FAIR) and my internship at FAIR London allowed me to work with exceptional professionals like Carl Schissler, Paul Calamia, Andrea Vedaldi, Natalia Neverova, Alexander Richard, and Roman Shapovalov, all of whom greatly enriched my research experience. My thesis committee consists of David Harwath, Yuke Zhu, Andrea Vedaldi, and Dinesh Manocha, and they have provided very insightful feedback on my thesis and presentations, which greatly enhanced the quality of my thesis.

I also want to share my gratitude to my fellow students at UT Austin—Ruohan Gao, Santhosh K. Ramakrishnan, Wei Sun, Sagnik Majumder, Arjun Somayazulu, Kumar Ashutosh, Jordi Ramos, Anshul Tomar, Puyuan Peng, Zihui Xue, and Ami Baid. Their collaboration was invaluable in advancing our shared projects.

My undergraduate mentors—Greg Mori, Alexander Alahi, Manolis Savva, Fred Tung—played pivotal roles in sparking my interest in research. Their guidance

laid the foundation for my academic pursuits.

Throughout my postdoc search, I benefited from the advice and support of Jiajun Wu, Ehsan Adeli, Fei-Fei Li, Gordon Wetzstein, Angjoo Kanazawa, Alyosha Efros, Jim Glass, Antonio Torralba, Josh McDermott, Abhinav Gupta, and Andrew Owens. Their insights into career choices were immensely helpful. I also want to thank Georgios Pavlakos, David Harwath, Yuke Zhu, and Qixing Huang at UT Austin for their career advice.

Finally, I must express my deepest appreciation to my parents, Chunbiao Chen and Chunling Li, for their unwavering support and love throughout this journey. I am also eternally grateful to Jasmin Zhang, whose companionship has been my greatest comfort during challenging times.

Abstract

4D Audio-Visual Learning: A Visual Perspective of Sound Propagation and Production

Changan Chen
The University of Texas at Austin, 2024

SUPERVISOR: Dr. Kristen Grauman

Humans use multiple modalities to perceive the world, including vision, sound, touch, and smell. Among them, vision and sound are two of the most important modalities that naturally co-occur. For example, we see and hear dogs barking, people having conversations, or cars honking on roads in our daily lives.

Recent work has been exploring this natural correspondence between sight and sound, which are, however, mainly object-centric, i.e., the semantic relations between objects and the sounds they make. While exciting, the correspondence with the surrounding 3D space is often overlooked. For example, we hear the same sound differently in different environments or even different locations in the same environment. In this thesis, I present *4D audio-visual learning*, which learns the *correspondence between sight and sounds in spaces, providing a visual perspective of sound propagation and sound production*. More specifically, I focus on four topics in this direction: *simulating* sounds in spaces, *navigating* with sounds in spaces, *synthesizing* sounds in spaces and *learning action sounds*. Throughout these topics, I use vision as the main bridge to connect audio and scene understanding. Below, I will detail the work on each of these topics.

Simulating sounds in spaces: Collecting visual-acoustic measurements is costly in

the real world. To enable machine learning models, I begin with building a first-of-its-kind simulation platform named SoundSpaces. Given any arbitrary source sound, source/receiver locations, and the mesh of the 3D environment, SoundSpaces produces realistic audio renderings simulating how sounds propagate in space as a function of the 3D environments and materials of different surfaces. Coupled with a modern visual rendering pipeline called Habitat, SoundSpaces produces 3D consistent visual and audio renderings. It is also continuous, configurable, and generalizable to novel environments. This platform has unlocked many research opportunities, enabling multimodal embodied AI and beyond.

Navigating with sounds in spaces: In robotics, navigating to localize a sound is an important application, for example, rescue robots searching for people or home service robots locating speech commands. However, existing robots mainly perceive the environment with vision sensors alone. To empower robots to see and hear, I introduce the *audio-visual navigation* task, where an embodied agent must navigate to the sounding object in an unknown environment by seeing and hearing. I train an end-to-end navigation policy based on reinforcement learning that predicts an action at every time step. This policy not only navigates to find the sounding object but also generalizes to unheard sounds and unseen environments. In a follow-up work, I introduce a hierarchical navigation policy that learns to set waypoints in an end-to-end fashion which further improves the navigation efficiency of the previous work. I also investigate the *semantic audio-visual navigation* problem, where sounds always come from semantically meaningful and visible objects, and I show that my proposed policy can learn to associate how objects sound to how they look without explicit annotations. Lastly, I show that we can also transfer the policy trained in simulation to the real world with frequency-adaptive prediction and demonstrate that with a physical robot platform.

Synthesizing sounds in spaces: While it is important to study sight and sound in an embodied setting, isolating perception from decision-making is also valuable for applications in augmented reality or virtual reality, such as generating matching

audio-visual streams for immersive experiences. I first propose the *audio-visual dereverberation* task, the goal of which is to remove reverberation from audio by utilizing visual cues. I show that the proposed model does well on downstream tasks such as speech recognition and speaker identification. In other applications, it is also desirable to *add* reverberation to audio to match the environment acoustics. I then investigate the inverse task: *visual acoustic matching*, where we transform audio to match the acoustics of a scene. Coupled with a self-supervised acoustic alteration strategy, the model learns to inject the proper amount of reverberation into the audio corresponding to the acoustics of the space. Lastly, to model the fine-grained acoustic changes within a scene, I propose the novel-view acoustic synthesis task, which requires the model to further reason about the nuanced change of audio in the same space at novel viewpoints.

Learning action sounds: Vision not only provides cues about how *sound propagates* in the space as a function of the environment configurations but also captures *how sounds are produced*. Learning or generating sounds from silent videos is important for applications such as creating sound effects for films or virtual reality games. To understand how our physical activities produce sound, I propose to learn *how human actions sound from narrated in-the-wild egocentric videos* with a novel multimodal consensus embedding approach. I show that our model successfully discovers sounding actions from in-the-wild videos and learns embeddings for cross-modality retrieval. I then investigate how to *generate temporally and semantically matching action sounds from silent videos*. I propose a novel ambient-aware audio generation model that learns to disentangle foreground action sounds from the ambient background sounds in in-the-wild training videos, which also enables controllable generation of the ambient sound.

Overall, my thesis covers promising directions in 4D audio-visual learning, that is, building fundamental simulation platforms, enabling multimodal embodied perception, providing faithful multimodal synthesis in 3D environments, and learning action sounds from in-the-wild videos. I show results on real videos and real-world

environments, as well as simulation. In the last chapter of my thesis, I outline the potential research that remains to be explored in the future for 4D audio-visual learning.

Table of Contents

List of Tables 14
List of Figures 17
Chapter 1: Introduction 27
Chapter 2: Related Work 38
2.1 Audio-Visual Learning 38
2.1.1 Audio-Visual Localization 38
2.1.2 Action Sounds 39
2.1.3 Egocentric Video Understanding with Audio 39
2.1.4 Multimodal Fusion 40
2.2 Embodied AI and Robotics 40
2.2.1 Visual Navigation 41
2.2.2 Sound Localization in Robotics 41
2.2.3 Audio-based Navigation 42
2.2.4 Hierarchical Navigation Policies 42
2.2.5 Visual Semantic Memory and Mapping for 3D Environments . 43
2.3 3D Scenes and Acoustics 44
2.3.1 3D Environments 44
2.3.2 Acoustic Simulation 44
2.3.3 Novel-View Synthesis (NVS) 45
2.4 Audio Signal Processing and Sound Synthesis 45
2.4.1 Audio Dereverberation and Speech Enhancement 46
2.4.2 Acoustic Matching 46
2.4.3 Audio Spatialization 47
Chapter 3: The SoundSpaces Platform 48
3.1 SoundSpaces: Simulating Sounds in 3D Environments 50
3.1.1 Audio Simulation Details 52
3.1.2 Visualizing Audio Simulations 55
3.2 SoundSpaces 2.0: A Simulation Platform for Visual-Acoustic Learning 56
3.2.1 Rendering Pipeline 58
3.2.2 Evaluation and Benchmarks 64
3.3 Conclusions 68

Chapter 4: Physical Audio-Visual Navigation . . . 70
4.1 Audio-Visual Navigation Benchmark in SoundSpaces . . . 72
4.1.1 Navigation Network and Training . . . 75
4.1.2 Experiments . . . 78
4.2 Learning to Set Waypoints for Audio-Visual Navigation . . . 85
4.2.1 Approach . . . 87
4.2.2 Experiments . . . 91
4.3 Continuous Audio-Visual Navigation in SoundSpaces 2.0 . . . 97
4.4 Sim2Real Transfer with Frequency-Adaptive Acoustic Field Prediction 99
4.4.1 Approach . . . 102
4.4.2 Data Curation . . . 108
4.4.3 Robot Platform . . . 110
4.4.4 Experiments . . . 110
4.5 Conclusions . . . 115
Chapter 5: Semantic Audio-Visual Navigation . . . 117
5.1 Semantic Audio-Visual Navigation . . . 119
5.2 Approach . . . 121
5.2.1 Observation Encoder . . . 123
5.2.2 Goal Descriptor Network . . . 123
5.2.3 Policy Network . . . 124
5.2.4 Training . . . 124
5.3 Experiments . . . 126
5.4 Conclusions . . . 133
Chapter 6: Learning Audio-Visual Dereverberation . . . 135
6.1 The Audio-Visual Dereverberation Task . . . 138
6.2 Dataset Curation . . . 140
6.3 Approach . . . 142
6.4 Experiments . . . 146
6.5 Conclusions . . . 150
Chapter 7: Visual Acoustic Matching . . . 153
7.1 The Visual Acoustic Matching Task . . . 156
7.2 Datasets . . . 156
7.2.1 SoundSpaces-Speech Dataset . . . 157
7.2.2 Acoustic AVSpeech Web Videos . . . 158
7.3 Approach . . . 158

7.3.1 Audio-Visual Feature Sequence Generation 159
7.3.2 Crossmodal Encoder 160
7.3.3 Waveform Generation and Loss 160
7.3.4 Acoustics Alteration for Self-Supervision 162
7.4 Experiments 163
7.4.1 Results on SoundSpaces-Speech 166
7.4.2 Results on Acoustic AVSpeech 168
7.5 Conclusions 170
Chapter 8: Novel-View Acoustic Synthesis 172
8.1 The Novel-View Acoustic Synthesis Task 175
8.2 Datasets 176
8.2.1 The Replay-NVAS Dataset 177
8.2.2 The SoundSpaces-NVAS Dataset 178
8.3 Visually-Guided Acoustic Synthesis 179
8.3.1 Ambient Sound Separation 180
8.3.2 Active Speaker Localization 181
8.3.3 Visual Acoustic Network and Fusion 182
8.3.4 Acoustic Synthesis 183
8.3.5 Temporal Alignment 184
8.3.6 Loss 184
8.4 Experiments 185
8.4.1 Results on SoundSpaces-NVAS 186
8.4.2 Results on Replay-NVAS 189
8.5 Conclusion 191
Chapter 9: SoundingActions: Learning How Actions Sound from Narrated Egocentric Videos 192
9.1 Task Formulation 196
9.2 Multimodal Contrastive-Consensus Coding 197
9.2.1 Align-Refine Two-stage Training 197
9.2.2 Multimodal Contrastive Coding 198
9.2.3 Multimodal Consensus Coding 199
9.2.4 Implementation Details 200
9.3 Training and Eval Data for Sounding Actions 201
9.4 Experiments 203
9.4.1 Sounding Action Discovery 205

9.4.2 Sounding Action Retrieval 207
9.4.3 Audio Classification on EPIC-Sounds 209
9.5 Conclusion 210
Chapter 10: Action2Sound: Ambient-Aware Generation of Action Sounds from Egocentric Videos 212
10.1 Ambient-aware Action Sound Generation 215
10.1.1 Action-to-Sound Generation 215
10.1.2 Disentangling Action and Ambient Sounds 216
10.1.3 Retrieval Augmented Generation and Controllable Generation . 217
10.1.4 Audio-Visual Latent Diffusion Model 218
10.1.5 Audio-Visual Representation Learning 220
10.1.6 Implementation Details 220
10.2 The Ego4D-Sounds Dataset 222
10.3 Experiments 222
10.3.1 Evaluation 222
10.3.2 Results on Ego4D-Sounds 224
10.3.3 Ambient Sound Control 225
10.3.4 Human Evaluation 227
10.3.5 Results on EPIC-KITCHENS 228
10.3.6 Demo on VR Cooking Game 228
10.4 Conclusion 229
Chapter 11: Conclusions and Future Work 231
References 234
Vita 274

List of Tables

3.1 Comparison with existing non-commercial datasets/simulation platforms. *Geometric* refers to acoustic simulation that is based on geometry of the objects and the space. *Configurable* means ability to alter simulation parameters, material and microphone properties. *Arbitrary Env* refers to the ability to render for an arbitrary new mesh environment, including point clouds generated in the wild. 58

3.2 Simulation speed vs. quality tradeoff. We report mean and standard deviation over 5 runs. 65

3.3 Far-field automatic speech recognition benchmark. 67

4.1 Summary of SoundSpaces dataset properties 78

4.2 Adding sound to sight and GPS sensing improves navigation performance significantly. Values are success rate normalized by path length (SPL); higher is better. 80

4.3 Navigation performance (SPL) when generalizing to unheard sounds. Higher is better. Results are averaged over 7 test runs; all standard deviations are ≤ 0.01. 81

4.4 AudioGoal navigation results. Our audio-visual waypoints navigation model (AV-WaN) reaches the goal faster (higher SPL) and it is more efficient (higher SNA) compared to the state-of-the-art. SPL, SR, SNA are shown as percentages. For all metrics, higher is better. (H) denotes a hierarchical model. 93

4.5 Ablation study for AV-WaN. Results are averaged over 5 test runs; all standard deviations are ≤ 0.5. 95

4.6 Continuous audio-visual navigation benchmark. DTG stands for distance to goal. We report the mean and standard deviation by training on 1 random seed, and evaluating on 3 random seeds. 98

4.7 Results of the AudioGoal navigation experiment and our model strongly outperforms existing methods. 110

4.8 Results for testing on real acoustic field data. 114

5.1 Navigation performance on the SoundSpaces Matterport3D dataset [35]. Our SAVi model has higher success rates and follows a shorter trajectory (SPL) to the goal compared to the state-of-the-art. Equipped with its explicit goal descriptor and having learned semantically grounded object sounds from training environments, our model is able to reach the goal more efficiently—even after it stops sounding—at a significantly higher rate than the closest competitor (see the SWS metric). All metrics are the higher the better except for DTG. 126

5.2 Navigation performance on *unheard sounds* in the presence of unheard distractor sounds. 131
5.3 Ablation experiment results. 132

6.1 Results on multiple speech analysis tasks, evaluated on the LibriSpeech test-clean set that is reverberated with our environmental simulator (with the exception of the "Anechoic (Upper bound)" setting, which is evaluated on the original audio). FT refers to tests where the models are finetuned with the audio-enhanced data. The relative improvement compared to Reverberant is included in parentheses. 147
6.2 Results on real data demonstrating sim2real transfer. 149
6.3 Breakdown of word error rate (WER) for VIDA without and with VAN on real test data. 149

7.1 Results on the SoundSpaces-Speech and Acoustic AVSpeech datasets for Seen and Unseen environments. All input audio at test time is novel (unheard during training). Note that the STFT metric is applicable only for SoundSpaces, where we can access the ground truth A_T's spectrogram. For all metrics, lower values are better. Standard errors for STFT, RTE and MOSE are all less than 0.04, 0.013s and 0.01 on SoundSpaces-Speech. Standard errors for RTE and MOSE are all less than 0.005s and 0.01 on Acoustic AVSpeech. 164
7.2 Ablations on model design and data. 168
7.3 Ablations on acoustics alteration. RTE is reported. 170
7.4 User study results. X%/Y% indicates among all paired examples for this baseline and AViTAR, X% of participants prefer this baseline while **Y%** prefer AViTAR. 170

8.1 **Results on SoundSpaces-NVAS and Replay-NVAS.** We report the magnitude spectrogram distance (Mag), left-right energy ratio error (LRE), and RT60 error (RTE). Replay-NVAS does not have novel environment setup due to data being collected in a single environment. For all metrics, lower is better. In addition to baselines, we also evaluate ViGAS w/o visual by removing the active speaker localization and visual features. Note that reverberation time is mostly invariant of the receiver location in the same room and thus input audio has low RTE. A good model should preserve this property while synthesizing the desired acoustics for the target viewpoint. 185
8.2 **Ablations of the model on both datasets**. 189
8.3 **Speech enhancement on Replay-NVAS**. 190
8.4 **Human Study**. Participants favor our approach over the two most realistic sounding baselines, (1) copying the input signal, and (2) a digital signal processing baseline. 190

9.1 Example verb groups and how frequently they sound 202

9.2 Sounding action discovery. Area-under-curve (AUC) values are reported for both ROC and precision-recall (PR) curves, for audio-vision (AV) and audio-language (AL). Both are the higher the better. We train our model five times with different seeds; the standard deviation is always within 0.01. 205

9.3 Sounding action retrieval. We report *Recall @5 and @10* for different query-retrieval modalities. 208

9.4 Results of classification on EPIC-Sounds. L: Linear-Probe; F: Fine-tuning. * denotes pretraining with supervised audio classification while the rest are pretrained in a self-supervised fashion. 210

10.1 Comparison with other audio-visual action datasets. Ego4D-Sounds not only has one order of magnitude more clips, but it is also coupled with language descriptions, supporting evaluation of sound generation based on semantics. 221

10.2 Results on Ego4D-Sounds test set. We also report the performance of the ground truth audio, which gives the upper bound value for each metric. 224

10.3 Survey results showing user preferences. Higher is better. Our model in the action-ambient joint generation setting scores highest for action sound quality, showing its ability to produce action-relevant sounds despite training with in-the-wild data. Ours in the action-focused generation setting scores highest for the least ambient sound, at a slight drop in action sound quality score, showing the ability to eliminate background sounds when requested by the user. 228

10.4 Results on Epic-Kitchens. GT stands for Ground Truth. 228

List of Figures

1.1 While the status quo of most audio-visual learning systems focuses on object-level correspondences, my thesis considers the correspondence between sight and sound in spaces. 28

1.2 Research summary. My research approaches 4D audio-visual learning by first simulating sounds in spaces (column 1) and then exploring active perception with audio-visual navigation (column 2), synthesizing sounds in spaces (column 3), and lastly learning action sounds from in-the-wild videos (column 4). 29

3.1 **Acoustic simulation in SoundSpaces 1.0.** We capture room impulse responses between each location pair within the illustrated grid (here for the 'frl_apartment_0' scene in Replica). In our platform, agents can experience binaural audio at densely sampled locations $\mathcal{L}$ marked with black dots—hearing the sound's intensity, direction, and frequency texture. Heatmaps display audio pressure fields, decreasing from red to blue. **Left**: When a sound source in $\mathcal{S}$ is placed in the center. **Right**: When a source is placed on the stairs. Notice how the sound received by the agent at different positions changes when the sound source moves, and how 3D structures influence the sound propagation. 49

3.2 **Pressure field of audio simulation** overlaid on the top-down map of apartment_2 from Replica [265]. Our audio-enabled agent gets rich directional information about the goal, since the pressure field variation is correlated with the shortest distance. Notice the discontinuities across walls and the gradient of the field along the *geodesic* path an agent must use to reach the goal (different from shortest Euclidean path). As a result, to an agent standing in the top right or bottom rooms, the audio reveals the door as a good intermediate goal. In other words, the audio stream signals to the agent that it must leave the current room to get to the target. In contrast, the GPS displacement vector would point through the wall and to the goal, which is a path the agent would discover it cannot traverse. Note that the visual stream is essential to couple with the audio stream in order to navigate around obstacles. 55

3.3 Illustration of SoundSpaces 2.0 rendering in a multi-room multi-floor HM3D [228] environment. In this scenario, a boy is watching TV in the living room while his mom calls him to have dinner from the kitchen downstairs. We model various frequency-dependent acoustic phenomena for sound propagation from all sources (TV and mom) to him, including direct sound, reflection, reverb, transmission, diffraction and air absorption. The sound propagation is based on a bidirectional path-tracing algorithm that takes the geometry of the scene as well as materials of objects in the space as input. The received sound is spatialized to binaural with the head-related transfer function (HRTF). As a result, SoundSpaces 2.0 renders the visual and audio observations with *spatial and acoustic correspondence*. For example, the TV being situated more towards the right results in right-ear signals that are stronger than those in the left ear. 57

3.4 **(a)** In this example, a person's phone rings in the dining room while she is in the living room and she asks the robot to bring her the phone. Upon receiving the audio signal with the binaural microphone, the robot needs to figure out two things: 1) what she is saying (far-field automatic speech recognition) and 2) how to navigate to her and the phone (audio-visual navigation). Note that far-field ASR is not limited to robotics; it has various applications such as video captioning. **(b)** Comparing real measurements and simulations in the Replica apartment [265] for 7 measurement positions and the 250Hz to 4000Hz frequency band. SoundSpaces 2.0 has a much lower error for the direct-to-reverberant ratio (DRR) compared to SoundSpaces. **(c)** Energy decay curve comparisons. The energy decay curve of SoundSpaces 2.0 is much closer to the real measurements than SoundSpaces. 66

4.1 **Audio source in an unmapped** 3D environment, where an autonomous agent must navigate to the goal. The top-down map is overlaid with the acoustic pressure field heatmap. Our audio-enabled agent gets rich directional information about the goal, since the audio intensity variation is correlated with the shortest path distance. The acoustics also reveal the room's geometry, major structures, and materials. Notice the gradient of the field along the *geodesic* path an agent must use to reach the goal (different from the shortest Euclidean path, which would cut through the inner wall). As a result, the proposed agent enjoys the synergy of both modalities: audio reveals the door as a good intermediate goal, while vision reveals the physical obstacles along the path, such as the furniture in the lefthand room. 71

4.2 **Audio-visual** navigation network. Our model uses both acoustic and visual cues from the 3D environment for effective navigation of complex scenes. . 76

4.3 **Navigation trajectories on top-down maps.** Agent path color fades from dark blue to light blue as time goes by. Green path indicates the shortest geodesic path. **Top:** Replica - The PointGoal agent bumps into the wall several times trying to move towards the target, unable to figure out the target is actually located in another room. In contrast, the AudioGoal and AudioPointGoal agents better sense the target: the sound travels through the door and the agent leaves the starting room immediately. **Bottom:** Matterport - the AudioGoal agent best avoids backtracking to efficiently reach the target in a large multi-room home. 82

4.4 **Audio as a learned spatial sensor.** (a) Navigation accuracy with increasing GPS noise. Unlike existing PointGoal agents, our AudioGoal agent does not rely on GPS, and hence is immune to GPS noise. (b) t-SNE projection of audio features, color coded to reveal their correlation with the goal location (left) and direction (right), *i.e.*, , source is far (red) or near (violet), and to the left (blue) or right (red) of the agent. 82

4.5 **Impact of each modality on action selection** for two AudioGoal episodes. We show one episode per row, and three sampled timesteps each. See Fig. 4.3 for legend. Blue and green bars display the importance of vision and audio, respectively. **Top:** Initially, the agent relies on audio to tell that the goal is on its left and decides to turn left. Later, it uses vision to recognize obstacles in front of it and decides to turn right. Finally, the agent decides to stop because the sound intensity has peaked. **Bottom:** Initially, the agent decides to turn left, following the audio source. Then the agent uses vision to identify the free space and decides to move forward. Later, the agent relies more on audio to decide to turn right as it hears the target from the right. 83

4.6 Waypoints for audio-visual navigation: Given egocentric audio-visual sensor inputs (depth and binaural sound), 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 may hear the phone ringing, identify that it is in another room, and decide to first exit the bedroom. It may then narrow down the phone location to the dining room, decide to enter it, and subsequently find it. Whereas existing hierarchical navigation methods rely on heuristics to determine subgoals, our model learns a policy to set waypoints jointly with the navigation task. 86

4.7 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. 88

4.8 Navigation trajectories on top-down maps vs. all existing AudioGoal methods. 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. The agent's inputs are egocentric views (Fig. 1); figures show the top-down view for ease of viewing the full trajectories. . . . 94

4.9 Analysis of selected waypoints (a,c) and accuracy vs. microphone noise (b). See text. . . . 96

4.10 Our robot predicts an acoustic field with a frequency-adaptive model and navigates to locate the sound source. . . . 100

4.11 Acoustic field prediction model. The model first extracts audio and visual features, and then tiles and concatenates both features to predict the acoustic field. . . . 103

4.12 Navigation pipeline. The model first predicts the acoustic field, samples the peak as the long-term goal, and navigates toward the goal with a path planner. . . . 105

4.13 Sim2real error as a function of frequencies. We report the mean and standard deviation of distance errors between the predicted and the ground truth peak locations. . . . 107

4.14 Navigation trajectory comparison. Our model successfully navigates to the source while other baselines fail due to either getting stuck or navigating in the wrong direction. . . . 113

4.15 Visualization of acoustic field prediction within the same episode. Top row: when the robot is still far from the goal. Bottom row: when the robot is right next to the goal. Our model predicts accurately in both cases. . . . 114

4.16 Acoustic field predictions on real data. The real data is measured with a lower resolution. We show the prediction and measurement for multiple sounds and directions. Our model predicts all of these cases accurately. . . . 115

5.1 Semantic audio-visual navigation in 3D environments: an agent must navigate to a sounding object. Since the sound may stop while the agent searches for the object, the agent is incentivized to learn the association between how objects look and sound, and to build contextual models for where different semantic sounds are more likely to occur (e.g., water dripping in the bathroom). . . . 118

5.2 In our model, the agent first encodes input observations and stores their features in memory M. Then our goal descriptor network leverages the acoustic cues to dynamically infer and update a *goal descriptor* D_t of the target object, which contains both location L_t and object category C_t information about the goal. By conditioning the agent's scene memory on the goal descriptor, the learned state representation s_t preserves information most relevant to the goal. Our transformer-based policy network attends to the encoded observations in M with self-attention to reason about the 3D environment seen so far, and it attends to M_e with D_t to capture possible associations between the hypothesized goal and the visual and acoustic observations to predict the state s_t. Then, s_t is fed to an actor-critic network, which predicts the next action a_t. The agent receives its reward from the environment based on how close to the goal it moves and whether it succeeds in reaching it. 122

5.3 Example SAVi navigation trajectories. In the first episode (top/magenta) the agent hears a water dripping sound and in the second episode (bottom/orange) a sound of opening and closing a door. For each episode, we show three egocentric visual views (right) sampled from the agent's trajectory at the start location ①, when the sound stops ②, and at the end location ③. In the top episode, the acoustic event lasts for two thirds of the trajectory and when the sound stops the agent has an accurate estimate of the object location that helps it find the sounding object (the sink). The second episode (bottom) has a much shorter acoustic event. The agent's estimate of the object location is inaccurate when the sound stops but still helps the agent as a general directional cue. The agent leverages this spatial cue *and* the semantic cue from its estimate of the object category, a cabinet, to attend to its multimodal memory to find the object in the kitchen and end the episode successfully. 128

5.4 Cumulative success rate vs. silence percentage. 133

6.1 The goal of audio-visual dereverberation is to leverage the visual observation of the environment to improve speech enhancement. 136

6.2 Visual cues reveal key factors influencing reverb effects on human speech audio. For example, these audio speech samples (depicted as waveforms and spectrograms) are identical lexically, but have very different reverberation properties owing to their differing environments. In the church, reverb is strong, in the classroom it is less, and when the speaker is distant from the camera it is again more evident. . . . 139

6.3 Audio-visual rendering for a Matterport environment. Left: bird's-eye view of the 3D environment. Right: panorama image rendered at the camera location and the corresponding received spectrogram. 140

6.4 VIDA model architecture. We convert the input speech to a spectrogram and use overlapping sliding windows to obtain 2.56 second segments. For visual inputs, we use separate ResNet18 networks to extract features e_r and e_d, which are fused to obtain e_c. We feed the spectrogram segment S_r^i to a UNet encoder, tile and concatenate e_c with the encoder's output, then use the UNet decoder to predict the clean spectrogram $\hat{S}_s^i$. During inference, we stitch the predicted spectrogams back into a full spectrogram and use Griffin-Lim [106] to reconstruct the output dereverberated waveform. 143

6.5 t-SNE of audio and visual features colored by the distance to the speaker (c) and RT60 (d). 150

6.6 Example input images, clean spectrograms, reverberant spectograms and spectrograms dereverberated by VIDA (top is from a scan, bottom is a real pano). The speaker is out of view in the first case and distant in the second case (back of the classroom). Though both received audio inputs are quite reverberant, our model successfully removes the reverb and restores the clean source speech. 151

7.1 Goal of visual acoustic matching: transform the sound recorded in one space to another space depicted in the target visual scene. For example, given source audio recorded in a studio, re-synthesize that audio to match the room acoustics of a concert hall. 154

7.2 Example images in (a) SoundSpaces and (b) AVSpeech. 157

7.3 AViTAR model illustration. We extract visual feature sequence V_i from input image I_T with a ResNet-18 [115], and audio feature sequence A_i from input audio A_S with 1D convolutions. V_i and A_i are passed into crossmodal encoders for crossmodal reasoning. The output feature sequence M_i is processed and upsampled with 1D convolutions to recover the output of the same temporal length. Finally, we use a multi-resolution speech GAN loss to guide the audio synthesis to be high fidelity. The acoustics alteration process is applied to the target audio during training if and only if there is no mismatched audio, e.g., on the Acoustic AVSpeech dataset. 159

7.4 Acoustics alteration process. Spectrograms of the resulting audio after each step are shown. We first dereverberate the target audio A_T to obtain cleaner audio A_C, randomize its acoustics by applying an impulse response of another environment to obtain A_R, and finally, add Gaussian noise to A_R to create A_S. Notice how the spectral pattern changes in this process. 162

7.5 Qualitative predicted audio. For all audio clips, we compute the magnitude spectrogram, convert the magnitude to dB, and plot the spectrogram with x-axis spanning from 0 to 1.28 s (left to right) and y-axis from 0 to 3000 Hz (bottom to top). Row 1: SoundSpaces-Speech example where the target space is a large empty room with a lot of reverberation. Our model predicts the audio closest to the target clip. AV U-Net's spectrogram is too smoothed compared to ours and misses some fine reverb details, which leads to perceptual distortion. Row 2: examples on Acoustic AVSpeech (unseen images). We feed one clean audio clip to match three different scenarios (office, garage, auditorium). From left to right, the audio spectrogram becomes more reverberant as phoneme patterns get extended and blurred on the temporal axis (est. RT60 times shown). NB: AViTAR processes waveforms, not spectrograms; here they are for visualization. 169

8.1 **Novel-view acoustic synthesis task.** Given audio-visual observations from one viewpoint and the relative target viewpoint pose, render the sound received at the target viewpoint. Note that the target is expressed as the desired pose of the microphones; the image at that pose (right) is neither observed nor synthesized. 173

8.2 Example source and target views for the two introduced datasets: Replay-NVAS (left) and SoundSpaces-NVAS (right). 177

8.3 **Visually Guided Acoustic Synthesis (ViGAS)**. Given the input audio A_S, we first separate out the ambient sound to focus on the sound of interest. We take the source audio and source visual to localize the active speaker on the 2D image. We also extract the visual acoustic features of the environment by running an encoder on the source visual. We concatenate the active speaker feature, source visual features, and the target pose, and fuse these features with a MLP. We feed both the audio stream A_C and fused visual feature V_C into the acoustic synthesis network, which has M stacked audio-visual fusion blocks. In each block, the audio sequence is processed by dilated conv1d layers and the visual features are processed by conv1d layers. Lastly, the previously separated ambient sound is added back to the waveform. During training, our temporal alignment module shifts the prediction by the amount of delay estimated between the source and the target audio to align the prediction well with the target. 180

8.4 **Qualitative examples**. For all binaural audio, we show the left-channel and the right-channel waveforms side-by-side. Row 1: SoundSpaces-NVAS example where given the source viewpoint and input audio, the model synthesizes audio for three different target viewpoints (target views are for reference only). In this case, the active speaker is the male speaker as indicated by the bounding box. For target viewpoint 1, the view rotates about 90 degrees and the male speaker is on the left side and the predicted left channel is louder than the right channel. Viewpoint 2 moves away from the speaker and thus yields lower amplitude compared to the first prediction. For target viewpoint 3, it is completely located outside of the living room, in which case, the sound could only come from the door open on the right (louder right channel) and the reverberation also greatly increases due to the vanishing direct sound. Row 2: Replay-NVAS example where the speaker is located on the left in the source viewpoint which becomes the right and further from the camera in target viewpoint 2, the model also predicts lower amplitude and louder right channel. On the right side, we show an example of the audio-visual speech enhancement for the active speaker. The model enhances the speech to largely match with the near-range audio (target). 188

9.1 We aim to distinguish sounds that are directly caused by human actions (bottom) from those that are not (top). Given egocentric training videos with language descriptions of the camera wearer's ("C") current action, we learn an embedding where the audio and visual features of any given clip are best aligned only when both are also consistent with the language. This allows discerning clips where the audio and vision may be *correlated* (e.g., the cutting machine running making loud noise in top row) versus those where the sounds are *driven by human action* (digging in bottom row)—importantly, without language at inference time. 193

9.2 **Main idea**. On the left, the Venn diagram illustrates different ways audio (A), video (V) and language (L) modalities can overlap in the content they capture. C refers to the camera wearer. Regions II,III,IV are information that is only shared between two modalities but not the third, e.g., the racing game in ① where the game sounds correlate with the vision, yet are not about the camera wearer's described action (using hands on laptop), the lifting action in ③, where the visuals and language agree but the action is inaudible, and the off-screen talking action in ④, where talking is heard and described, but the camera wearer cannot be seen speaking. Region I is the information that corresponds to all modalities agreeing, e.g., the visible and audible plastering action in ②. Our model's "align" phase detects any such (dis)agreements via pairwise contrastive learning on the modalities. In the "refine" phase, we use the intersection of that agreement (region I) to refine the embedding. For example, on the right, we show what the three modality embeddings should look like after the "align" stage for examples 1 and 2. Embeddings of instances where all modalities agree will be closer in the embedding space and apart otherwise. In other words, for example 1, yellow (video) cannot be close to blue (audio) unless green is too (language). 194

9.3 **Multimodal contrastive-consensus loss**. (a): Given three modality embeddings e_i^t, e_j^t, e_k^t, multimodal contrastive coding pulls each pair of modalities closer while pushing modality pairs from another sample further away. (b): However, not all modalities agree on how close they should be depending on the instance. Thus we set the furthest distance a feature has with respect to the anchor feature as the consensus and push the remaining embeddings away to meet this consensus. 198

9.4 Long-tail distribution of sounding actions. 204

9.5 Sounding action discovery accuracy 204

9.6 Example visual embedding cluster from our model 207

9.7 Qualitative examples for retrieval. The first row is video-to-audio retrieval, motivated by adding audio effects for silent videos. The second row is audio-to-text retrieval, motivated by audio captioning applications. For each row, we show three correct retrieval examples along with their text (gray indicates the text is not observed by the model). For the retrieved item, we show the ground truth rank as the superscript. All examples are long-tail sounding actions, showing how our model learns to capture the features of how actions sound. 208

10.1 Real-world audio consists of both foreground action sounds (whose causes are visible in the FoV) and background ambient sounds that are generated by sources offscreen. Whereas prior work is agnostic to this division when performing generation, our method is ambient-aware and disentangles action sound from ambient sound. Our key technical insight is how to train with in-the-wild videos exhibiting natural ambient sounds, while still learning to factor out their effects on generation. The green arrows reference how we condition generation on sound from a related, but time-distinct, video clip to achieve this. 213

10.2 Illustration of the harm of ambient sound in video-to-audio generation. In this example, this person is closing a packet of ginger powder, which makes some rustling sound (red circled in the middle). There is also some buzzing sound semantically irrelevant to the visual scene in the background, which dominates the energy of the spectrogram. On the right-hand side, we show a prediction made by a vanilla model that misses the action sound but predicts the ambient sound. 215

10.3 Audio condition selection and the model architecture. **Left**: During training, we randomly sample a neighbor audio clip as the audio condition. For inference, we query the training set audio with the (silent) input video and retrieve an audio clip that has the highest audio-visual similarity with the input video using our trained AV-Sim model (Sec. 10.1.5). **Right**: We represent audio waveforms as spectrograms and use a latent diffusion model to generate the spectrogram conditioned on both the input video and the audio condition. At test time, we use a trained vocoder network to transform the spectrogram to a waveform. 217

10.4 Two inference settings: "action-ambient joint generation" and "action-focused generation". In the first setting, we condition on audio retrieved from the training set and aim to generate both plausible action and ambient sounds. In the second setting, we specify an audio file with low ambient sound and the model focuses on generating plausible action sounds while minimizing the ambient sounds. 218

10.5 Example clips in Ego4D-Sounds. We show one video frame, the action description, and the sound for each example. Note how these actions are subtle and long-tail, usually not present in typical video datasets. 221

10.6 Qualitative example. We show the frames of each video followed by the waveform/spectrogram of various baseline methods. Our model generates the most synchronized sounds. 225

10.7 The achieved ambient level and accuracy of the prediction as a function of the input ambient levels. (a): we show the ambient level of our model changes according to the ambient level in the audio condition while the ambient level of "Ours w/o cond" and the original audio stay constant, illustrating the controllability of our model. (b) FAD is low for most input ambient levels unless it goes too extreme (too low or too high), showing our model generates high-quality action sounds even when varying output ambient levels. 226

10.8 Visualization of action-focused generation. For both examples, Diff-Foley [174], Ours w/o cond or Ours (action-ambient generation) generate plausible action sounds along with ambient sounds. In contrast, our model conditioned on a low ambient sound generates plausible action sounds (see green boxes) with minimal ambient sound. 227

10.9 We apply our model on a VR cooking game clip where the person cuts a sushi roll three times. Our model successfully predicts the 3 cutting sounds. 229

Chapter 1: Introduction

Humans use multiple modalities to perceive the world, including vision, sound, touch, and smell. Among them, vision and sound are two important modalities that provide complementary information about each other. For example, when we talk to other people, we not only hear the speech sounds they make but also see their lip movements, which are also indicative of the speech content. Or when we hear footsteps approaching from behind, we also have the mental image of someone walking toward us. Motivated by the correspondence between sight and sound, audio-visual learning has gained popularity in the past few years. Various tasks have been introduced, for example, audio-visual classification, where the goal is to classify the foreground object based on both the audio and visual streams [11, 211, 46], audio-visual localization, the goal of which is to localize sounding objects on the 2D plane [280, 255, 12], and audio-visual separation, which separates sounds based on the association between sounds and visual objects [322, 89, 211].

While audio-visual learning has achieved many breakthroughs, most of the existing tasks focus on the *semantic* correspondence between single objects [11, 322, 211, 12]. While object-level correspondence has been extensively studied, the correspondence between sound and *space* is often overlooked. When objects vibrate and produce sound waves, the sound waves propagate and attenuate in the air, reflect off, get absorbed or transmit through surfaces, and then reach our ears. Our ear canals then shape the sound uniquely, allowing us to sense the direction of the sound without looking at the objects. The sound we hear is thus a function of the geometry of the space, materials of different objects, and the source/receiver locations. For example, here are some phenomena we observe in our daily lives: as we move closer to the sound source, the volume increases; if we speak in an empty room, there tends to be a lot of reverberation; if we talk in a carpeted room, we can hear each other more clearly; if we hear the alarm going off, we know it most likely comes from the

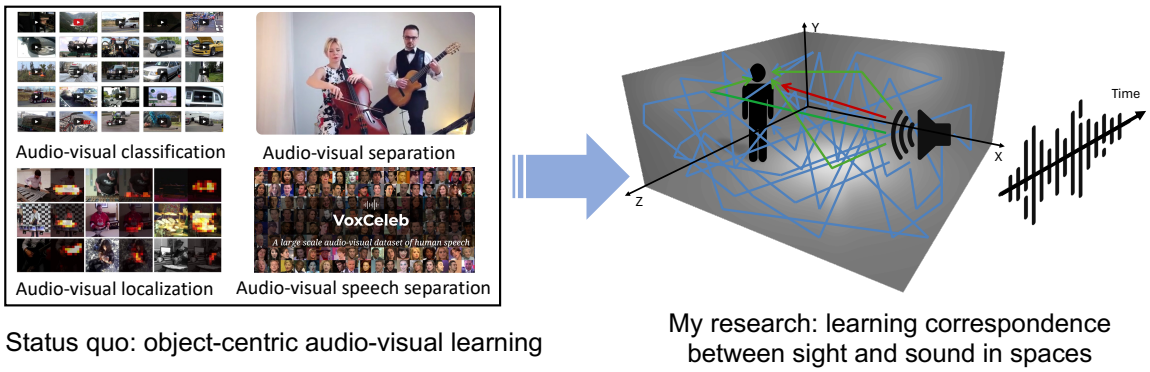


Figure 1.1: While the status quo of most audio-visual learning systems focuses on object-level correspondences, my thesis considers the correspondence between sight and sound in spaces.

kitchen. There is rich physical and semantic correspondence between sounds and spaces. While it is impossible to get the full measurement of spaces in the real world, vision captures important information about the surrounding environment, e.g., 3D geometry, materials, and source/receiver locations, and connects sounds and spaces.

Studying the link between sounds and spaces is important to many rising real-world applications in robotics, augmented reality (AR), and virtual reality (VR). For example, for a home assistance robot, when it hears a sound, either humans' speech commands or glass shattering (emergency), it needs to navigate to find where the sound comes from and act accordingly. This requires the robot to reason about how sounds change as it moves and in which room the sounding object is likely to be located. When we wear AR glasses, we would like the glass to enhance our experiences. For example, when we have conversations with people in a spacious environment, we would like the glass to remove the reverberation in the sounds based on visual observations of the space. When we talk with friends in virtual reality, we also like to hear sounds consistent with what we see to have immersive experiences.

Motivated by these applications, my research aims to go beyond object-centric audio-visual learning and study the correspondence between sight and sound in spaces, thus audio-visual learning in 4D (3 spatial dimensions plus the time dimension).

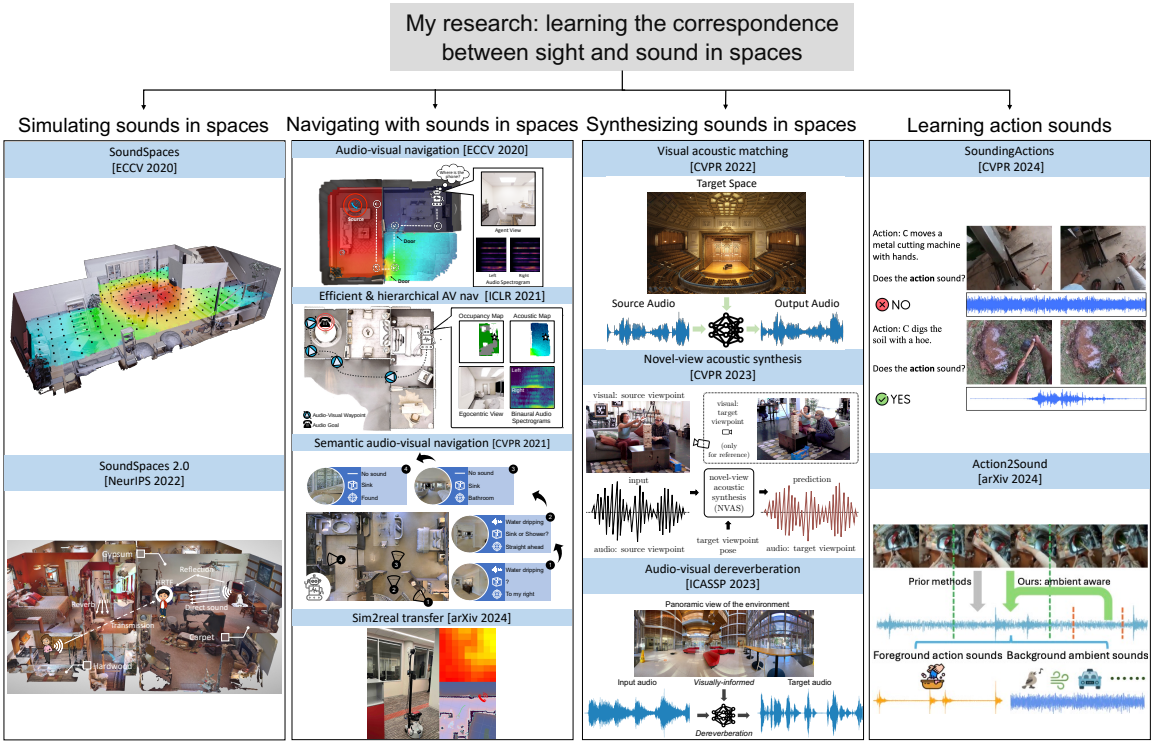


Figure 1.2: Research summary. My research approaches 4D audio-visual learning by first simulating sounds in spaces (column 1) and then exploring active perception with audio-visual navigation (column 2), synthesizing sounds in spaces (column 3), and lastly learning action sounds from in-the-wild videos (column 4).

See Fig. 1.1. I investigate how sounds are produced and propagated in spaces from a visual perspective, which is an underexplored area with many open challenges. For sound propagation, I aim to answer a few important questions: 1) How do we get scalable 3D audio-visual data since data is the key to all modern machine learning algorithms? 2) In the context of mobile robot navigation, both the environment and sound change as the robot moves around. How do we learn policies that make the optimal movement decision while actively perceiving audio-visual inputs? 3) For AR/VR applications, producing 3D consistent audio-visual streams is the key to providing an immersive experience. How do we reason about the scene geometry's effect on the transformation of the sound from visuals? 4) For sound production, how do we learn how human actions make sounds from in-the-wild videos that have

coupled background sounds and no annotations?

To answer these questions, I approach 4D audio-visual learning by studying the following topics (Fig. 1.2).

1. Simulating sounds in spaces: To address the data challenge, I take a simulation-driven approach to allow clean and controllable generation of audio-visual data. I build the first-of-its-kind audio-visual simulator SoundSpaces [35] that provides pre-rendered impulse responses for 3D environments. In follow-up work, I further introduce SoundSpaces 2.0 [40] that renders on-the-fly and is both configurable and generalizable to the real world. See this work in Chapter 3.

2. Navigating with sounds in spaces: Navigation is one of the essential abilities of robots operating in the real world. For robots to actively perceive audio-visual inputs, I start with tackling the navigation problem. I present the first audio-visual navigation task and benchmark [35], an efficient hierarchical navigation policy that learns to set waypoints [38], the semantic audio-visual navigation task and model that reasons about the semantic relation between sounds and the space [36], the continuous audio-visual navigation benchmark in SoundSpaces 2.0 [40], and lastly, a sim-to-real transfer technique for enabling audio-visual navigation on real robots [45]. See this work in Chapter 4 and Chapter 5

3. Synthesizing sounds in spaces: For AR/VR applications, it is vital for device wearers to perceive 3D consistent content (audio-visual streams) for an immersive experience. Toward that goal, I study synthesizing acoustically correct sounds given visuals of the scene. I introduce the audio-visual dereverberation task that learns to remove reverberation with visual cues [42], the visual-acoustic matching task that transforms an audio clip to match the acoustics of a space specified in the image [39], and lastly the novel-view acoustic synthesis task that given audio-visual observations from a reference viewpoint, synthesizes the audio at a target viewpoint [41]. See this work in Chapter 6, Chapter 7 and Chapter 8.

4. Learning action sounds: Humans interact with surrounding objects every moment of our lives, which often produce sounds and are captured in videos. To learn how sounds are produced, I focus on learning action sounds from in-the-wild videos by answering two questions: whatactions make sounds, and how to generate action sounds given a silent video? I first propose a novel self-supervised embedding to learn what actions sound from narrated in-the-wild egocentric videos [43]. I then devise a novel audio-conditioning mechanism to generate action sounds given silent video by learning to disentangle foreground action sounds and the ambient background sounds from in-the-wild videos [44]. See this work in Chapter 9 and Chapter 10.

Having overviewed the main thrusts of this thesis, I next provide more context for these problem areas and summarize my insights and results. One of the main challenges in studying the acoustic correspondence between sight and sound is the lack of data. Different from object-centric audio-visual learning, where recording videos of audio-visual events is sufficient for studying the correspondence, understanding the link between sound and spaces not only requires recording the sound but also measuring the 3D spaces, which is very expensive. Even if we collected these 4D data, they would be passive recordings, not meeting the requirement for robotics applications. To address the data limitation, I propose to take a simulation-driven approach that allows curating clean and controllable audio-visual data that are scalable with machine learning models. Specifically, I start by building the first *audio-visual simulation platform*: SoundSpaces, which renders audio realistically based on the visual scans of real-world environments. SoundSpaces precomputes room impulse responses (RIRs), which is a transfer function that describes how sounds transform from the source location to the receiver location based on the 3D environment. The precomputed RIRs allow fast iteration of embodied tasks, especially reinforcement learning-based models. However, SoundSpaces does not generalize to new environments or continuous spaces. Thus, in follow-up work, we introduce SoundSpaces 2.0, which allows

continuous spatial sampling, generalization to novel environments, and configurable microphone and material properties. This geometry-based acoustic simulation offers both high fidelity and realism while also being fast enough to be used for embodied learning. We showcase the simulator's properties and benchmark its performance against real-world audio measurements. In Chapter 3, we discuss both SoundSpaces versions and what they have enabled in research.

Enabled by the simulation, I then look into embodied settings where an agent makes active decisions for movement while perceiving the environment. I focus on navigation, which is an essential ability of robots operating in the real world. Following in the steps of visual navigation from egocentric observations [109, 188, 247], I proposed the first *audio-visual navigation* task, where the goal is to find a sounding object in an unknown environment. This has many applications in the real world. For example, a rescue robot needs to find the source of the sound of someone yelling for help when there is a fire, or a home robot needs to locate where the burglar is when he breaks into the home. In Chapter 4, I introduce the definition of task, a baseline navigation algorithm, and the benchmark. This model learns to act at a fixed granularity of agent motion and relies on simple recurrent aggregations of the audio observations. We further introduce an efficient hierarchical policy to audio-visual navigation with two key novel elements: 1) 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 to reveal the geometry of an unmapped space. Lastly, we introduce a frequency-adaptive method for transferring the policy from simulation to the real world and build a physical robot that can navigate to sounding objects in the real world without being trained on any real audio data.

In the previously defined audio-visual navigation, the task assumes a constantly sounding target and restricts the role of audio to signal the target's position (an invisible point). This is a simplification of the real-world scenario where the

sound-making object is visible and may only emit sounds for a short period of time. In Chapter 5, we introduce the *semantic audio-visual navigation* task, where objects in the environment make sounds consistent with their semantic meaning (e.g., toilet flushing, door creaking) and acoustic events are sporadic or short in duration. We propose a transformer-based model to tackle this new semantic AudioGoal task, incorporating an inferred goal descriptor that captures both spatial and semantic properties of the target. Our model's persistent multimodal memory enables it to reach the goal even long after the acoustic event stops. In support of the new task, we also expand the SoundSpaces audio simulations to provide semantically grounded sounds for an array of objects in Matterport3D. Our method strongly outperforms existing audio-visual navigation methods by learning to associate semantic, acoustic, and visual cues.

When an intelligent agent moves around in the environment, understanding the content of the sound is just as important as finding where the sound comes from. For example, someone might ask a home robot to "bring me a coffee from the kitchen" from a distance. The agent needs to perform both automatic speech recognition (ASR) and audio-visual navigation to bring the coffee to that person. However, when receiving sounds from a distance, there is reverberation in the received sound, which severely impacts the accuracy of automatic speech recognition. This is not only harmful for robotic applications but also for human perception, where too much reverberation in a video recording degrades the quality of speech. Prior work attempts to remove reverberation based on the audio modality only. My idea is to learn to *dereverberate speech from audio-visual observations*. The visual environment surrounding a human speaker reveals important cues about the room geometry, materials, and speaker location, all of which influence the precise reverberation effects. In Chapter 6, I introduce Visually-Informed Dereverberation of Audio (VIDA), an end-to-end approach that learns to remove reverberation based on both the observed monaural sound and visual scene. In support of this new task, I develop a large-scale dataset SoundSpaces-Speech that uses realistic acoustic renderings of speech in

real-world 3D scans of homes offering a variety of room acoustics. Demonstrating my approach on both simulated and real imagery for speech enhancement, speech recognition, and speaker identification, I show it achieves state-of-the-art performance and substantially improves over audio-only methods.

In audio-visual dereverberation, the goal is to leverage the visual knowledge of the 3D environment to help remove the reverberation in the speech. However, in some other applications, synthesizing the acoustics of environments might be of vital importance. For example, in virtual reality, we would like a person's voice to sound like it was produced in the virtual world, i.e., matching the acoustics of the environment that we visually observe. For this reason, in Chapter 7, I introduce the *visual acoustic matching* task, in which an audio clip is transformed to sound like it was recorded in a target environment. Given an image of the target environment and a waveform for the source audio, the goal is to re-synthesize the audio to match the target room acoustics as suggested by its visible geometry and materials. To address this novel task, I propose a crossmodal transformer model that uses audio-visual attention to inject visual properties into the audio and generate realistic audio output. In addition, I devise a self-supervised training objective that can learn acoustic matching from in-the-wild Web videos, despite their lack of acoustically mismatched audio. I demonstrate that my approach successfully translates human speech to a variety of real-world environments depicted in images, outperforming both traditional acoustic matching and more heavily supervised baselines.

In visual acoustic matching, the goal is to match the acoustics of some target environments. However, since it only targets single-channel audio, it does not account for the fine-grained acoustic changes, e.g., how the acoustics change from one viewpoint in space to another in the same environment. This has applications in augmented reality, where we would like to replay real-world videos from different viewpoints. In Chapter 8, I introduce the *novel-view acoustic synthesis* (NVAS) task: Given the sight and sound observed at a source viewpoint and the camera pose of an

unseen target viewpoint, can we synthesize the sound of that scene from that viewpoint? I propose a neural rendering approach: Visually-Guided Acoustic Synthesis (ViGAS) network that learns to synthesize the sound at an arbitrary point in space by analyzing the input audio-visual cues. To benchmark this task, I collect two first-of-their-kind large-scale multi-view audio-visual datasets, one synthetic and one real. I show that our model successfully reasons about the spatial cues and synthesizes faithful audio on both datasets. To our knowledge, this work represents the very first formulation, dataset, and approach to solving the novel-view acoustic synthesis task, which has exciting potential applications ranging from AR/VR to art and design.

In the final thrust of my thesis, I explore the relationship between human actions and sounds. Our vision not only provides cues about sound waves propagating in spaces, but also captures how these sound waves are produced by object collisions or vibrations. We interact with objects around us at every moment of our lives, for example, when we close a door, chop vegetables, or type on keyboards. These physical activities produce sounds and are strongly associated with the subjects of our activity and how we perform it. Understanding the link between sounds and actions is valuable for a number of applications, such as multimodal activity recognition, cross-modal retrieval, or forecasting the physical effects of a person's actions. In SoundingActions (Chapter 9), we propose a novel self-supervised embedding to learn *how actions sound* from narrated in-the-wild egocentric videos. Whereas existing methods rely on curated data with known audio-visual correspondence, our multimodal contrastive-consensus coding (MC3) embedding reinforces the associations between audio, language, and vision when all modality pairs agree while diminishing those associations when any one pair does not. We show our approach can successfully discover how the long-tail of human actions sound from egocentric video, outperforming an array of recent multimodal embedding techniques on two datasets (Ego4D [105] and EPIC-Sounds [121]) and multiple crossmodal tasks.

Building on the ideas of SoundingActions, I expand the scope to go from *discovering* action-sound associations, to actually *generating* the sounds that could go

with a given visual action in video. The task offers a complementary way to study the fundamental problem of audio-visual actions, and it also has various possible applications, such as creating sound effects for films or virtual reality games. Existing approaches implicitly assume total correspondence between the video and audio during training, yet many sounds happen off-screen and have weak to no correspondence with the visuals—resulting in uncontrolled ambient sounds or hallucinations at test time. In Chapter 10, we propose a novel *ambient-aware* audio generation model. We devise a novel audio-conditioning mechanism to learn to disentangle foreground action sounds from the ambient background sounds in in-the-wild training videos. Given a novel silent video, our model uses retrieval-augmented generation to create audio that matches the visual content both semantically and temporally. We train and evaluate our model on two in-the-wild egocentric video datasets Ego4D [105] and EPIC-KITCHENS [57]. Our model outperforms an array of existing methods, allows controllable generation of the ambient sound, and even shows promise for generalizing to computer graphics game clips. Overall, our work is the first to focus video-to-audio generation faithfully on the observed visual content despite training from uncurated clips with natural background sounds.

To summarize, my thesis focuses on studying the correspondence between sight and sound in 3D spaces. This includes developing platforms, addressing active perception challenges in robotics, tackling generation problems in augmented and virtual reality, and studying how human actions produce sounds in real-world videos. I have made the following contributions:

1. Built and supported a first-of-its-kind simulation platform SoundSpaces that unlocks many research opportunities (Chapter 3).

2. Empowered embodied agents to see, hear, and move in 3D scenes and the ability to actively locate sound sources in 3D environments (Chapter 4, Chapter 5).

3. Extended audio-visual generation to model 3D scene acoustics by reasoning

about the acoustic properties of an environment from visuals to guide audio generation (Chapter 6, Chapter 7, Chapter 8).

4. Enabled learning and generating long-tail human action sounds from in-the-wild videos (Chapter 9, Chapter 10).

Next, I examine the significant related work pertinent to the research discussed in this thesis. I present the methods discussed earlier in Chapter 4 to Chapter 10 in detail. The concluding chapter summarizes the research conducted for my thesis and outlines potential future directions that extend toward my long-term research goals beyond this thesis.

Chapter 2: Related Work

In this chapter, I review prior work relevant to my thesis, including audio-visual learning (Sec. 2.1), embodied AI (Sec. 2.2), 3D environments and acoustics (Sec. 2.3), and audio processing and sound synthesis (Sec. 2.4). The material in this chapter serves both to understand the literature in the research explored in this thesis and to introduce the difference between existing work and my proposed models.

2.1 Audio-Visual Learning

In this section, I review previous work on audio-visual localization, action sounds, egocentric video understanding with audio, and the existing strategies for fusing the audio and video streams together.

2.1.1 Audio-Visual Localization

The goal of audio-visual localization is to localize the object that makes sound in video frames. The audio-visual correspondence (AVC) framework aims to maximize the similarity between audio and visual features [12, 211]. This work typically deals with single-source scenarios, but it is not capable of extending to multi-source scenarios. Other methods propose to combine audio-visual separation with audio-visual localization and solve them jointly with a mix-and-separate strategy [322, 87].

Different from this work, my research focuses on localizing sounds in 3D. For example, in audio-visual navigation (Chapter 4 and Chapter 5), the agent is tasked to navigate in 3D spaces to find the goal. In addition, the agent can actively choose its action while the existing work deals with passively collected videos. In Chapter 8, the model also needs to reason the 3D location of the active speaker in order to transform sounds from one location to another.

2.1.2 Action Sounds

Some work [137, 199, 91] leverages audio to improve activity recognition on video datasets such as UCF101 [261] and ActivityNet [73], which have visual labels but no audio labels. Existing audio datasets such as AudioSet [93] and VGG-Sound [46] target general sound classes such as music, speech, and sports. EPIC-Sounds [121] provides an audio classification benchmark for actions in kitchen environments, but it has no labels for the correspondence between the visual action and the sound. The Greatest Hits dataset [212] contains videos where people hit and scratch object surfaces with a drumstick, which enables audio synthesis from videos. Interaction sound has also been studied in robotics, e.g., using a robotic platform to collect sounds and study the synergy between action and sounds [85, 53]. Impact sounds are modeled in a physics-based simulator [84].

Throughout, the existing work assumes a fixed, given taxonomy of action classes or audio labels of interest. Different from this work, in Chapter 9, we learn how actions make sounds from in-the-wild narrated egocentric videos, and in Chapter 10, we study the action2sound generation problem from in-the-wild egocentric videos, both without relying on a taxonomy of discrete labels for the audio events.

2.1.3 Egocentric Video Understanding with Audio

Understanding human activities in videos has long been a core challenge of computer vision. Early research studies activity recognition from exocentric videos such as UCF101 [261], Kinetics [136], or ActivityNet [73]. Recent work explores the egocentric setting and introduces large egocentric datasets such as Ego4D [105] or EPIC-KITCHENS [57]. Leveraging both the video and audio streams in egocentric videos, many interesting tasks are enhanced, such as action recognition [137], localization [230], active speaker localization [132], sounding object localization [120], and state-aware visual representations from audible interactions [189].

Existing audio-visual learning work for egocentric video focuses on perception,

i.e., understanding what happens in the video. In contrast, in Chapter 10, we target the video-to-audio generation problem by learning to disentangle the action sound from ambient sounds.

2.1.4 Multimodal Fusion

One standard solution for audio-visual feature fusion is to represent audio as spectrograms, a matrix representation of the spectrum of frequencies of a signal as it varies with time, process them with a convolutional neural network (CNN), and concatenate with visual features from another CNN[211, 86, 70, 88, 35]. This fusion strategy is limited by using one global feature to represent the scene and thus supports only coarse-grained reasoning. The transformer [290] has proven to be a power tool in vision [149, 95]. Its self-attention operation provides a natural mechanism to fuse high-dimensional signals of different sensory modalities, and it has been used in various tasks such as action recognition [21], self-supervised learning [6, 2, 216], and language modeling [107]. Audio-visual attention [285, 284, 166] has been recently studied to capture the correlation between visual features and audio features. In Chapter 7, I use crossmodal attention to learn how different regions of the image contribute to reverberation. We show that compared with the conventional concatenation-based fusion, the proposed model predicts acoustics from images more accurately.

2.2 Embodied AI and Robotics

In this section, I review work that is related to my thesis in recent embodied AI and robotics literature. More specifically, I will cover visual navigation, sound source localization in robotics, audio-based navigation, hierarchical navigation, and memory and mapping for 3D environments.

2.2.1 Visual Navigation

To navigate autonomously, traditionally, a robot builds a map via 3D reconstruction (i.e., SLAM) and then plans a path using the map [79]. Recent work instead learns navigation policies directly from egocentric observations [109, 244, 188]. A popular task is PointGoal navigation, where the goal position is given to the agent [109, 188, 247, 302]. Alternatively, in the ObjectGoal setting, the agent is given an object label rather than the goal location, and must navigate to the nearest instance of that category (e.g., go to a table) [327, 18, 31]. Visual navigation can be tied to other tasks to attain intelligent behavior, such as question answering [101, 59, 60], active visual recognition [128], and instruction following [9, 47].

Previously, embodied agents were all deaf and did not have hearing ability. In this thesis, I propose the AudioGoal task in Chapter 4 and the semantic AudioGoal task in Chapter 5, which enable the embodied agent to both see and hear. In contrast to both PointGoal and ObjectGoal, in both AudioGoal settings, the agent is not given specific goal information. Instead, it needs to react to an acoustic event to determine what kind of object is sounding and navigate to it. Furthermore, unlike ObjectGoal, the agent needs to navigate to the specific object instance that emitted the sound rather than any instance of that category. Our task represents real-world scenarios where dynamic objects draw the attention of an agent and call it to action (e.g., the sound of a heavy object falling upstairs).

2.2.2 Sound Localization in Robotics

In robotics, microphone arrays are often used for sound source localization [201, 232, 202, 203]. Past studies fuse AV cues for surveillance [309, 226], speech recognition [317], human robot interaction [3, 292], and robotic manipulation tasks [239]. None attempt audio-visual navigation in unmapped environments. Concurrent work explores AV navigation in computer graphics environments [83]. In contrast to our end-to-end RL agent, their model decouples the task into predicting the goal location

from audio and then planning a path to it. Our simulation platform SoundSpaces (in Chapter 3) is more realistic for both visuals (real-world images in ours vs. computer graphics in [83]) and acoustics (ray tracing/sound penetration/full occlusion model in ours vs. low-cost game audio in [83]).

2.2.3 Audio-based Navigation

Cognitive science also confirms that audio is a strong navigational signal [278, 185]. Blind and sighted people show comparable skill on spatial navigation [76] and sound localization [103, 162, 238, 294] tasks. Consequently, audio-based AR/VR equipment has been devised for auditory sensory substitution for human users for obstacle avoidance and navigation [183, 108]. Additionally, cartoon-like virtual 2D and 3D AV environments can help evaluate human learning of audio cues [55, 303, 184]. Unlike our proposed SoundSpaces platform in Chapter 3, these environments are non-photorealistic and they are for *human* navigators; they do not support AI agents or training. Prior studies with autonomous agents in simulated environments are restricted to human-constructed game boards, do not use acoustically correct sound models, and train and test on the same environment [298, 305].

2.2.4 Hierarchical Navigation Policies

Current methods often learn policies that reward moving to the final goal location using a step-by-step action space (e.g., TurnRight, MoveForward, Stop) [109, 187, 188, 247]. However, recent work explores ways to incorporate subgoals or waypoints for PointGoal navigation. Taking inspiration from hierarchical learning [14, 198], the general idea is to select a subgoal, use planning (or a local policy) to navigate to the current subgoal, and repeat [262, 15, 32, 200, 308, 26]. For example, [15] apply a CNN to the RGB input to predict the next waypoint—the ground truth of which is collected using trajectory optimization—then apply model-based planning. Active Neural SLAM (ANS) [32] plans a path to the point goal (or a predicted long-term

exploration goal) using a partial map of the environment, generating each subgoal to be within 0.25 m of the agent using an analytic shortest path planner.

I present a hierarchical navigation policy AV-WaN in Sec. 4.2.1.2. Differing from the existing work on hierarchical navigation, AV-WaN not only tackles Audio-Goal navigation but also learns to generate navigation subgoals in an end-to-end fashion, whereas prior work relies on heuristics like selecting frontiers [26, 262] or points along the shortest collision-free path [15, 32] to define subgoals.

2.2.5 Visual Semantic Memory and Mapping for 3D Environments

Learning-based visual mapping algorithms [116, 244, 110, 109] show exciting promise to overcome the limits of purely geometric maps. Some methods to use an implicit memory representation in navigation to aggregate observations, e.g., a recurrent network [187, 247, 35, 154, 9, 193], and other methods leverage explicit map-based memories to record occupancy [109, 48, 227, 32, 38, 225] or object locations [31, 28].

Prior work typically only stores visual observations in the memory and is limited in the audio-visual navigation task. I introduce the first multimodal spatial memory in Chapter 4, which encodes both visual and acoustic observations registered with the agent's movement along the ground plane. I show that multimodal memory is essential for the agent to produce good action sequences.

In Chapter 5, the model is tasked to reason about the semantic relation of the sporadic sounds and the environment. To capture long-term dependencies, another promising direction is to use a transformer architecture [290] to record observations and poses [74]. We build in this direction and introduce a scene memory transformer that, unlike prior work, 1) is multimodal and 2) leverages an explicit learned *goal descriptor* to attend to the memory. Our memory model learns audio-visual associations between the goal and the observations from the scene, a crucial functionality as we demonstrate in experiments.

2.3 3D Scenes and Acoustics

My thesis focuses on understanding the correspondence between sounds and spaces. In this section, I will first review prior work that targets creating 3D environments, and then related work that simulates sounds, and lastly the 3D vision problem of novel-view synthesis.

2.3.1 3D Environments

Recent research in embodied perception is greatly facilitated by new 3D environments and simulation platforms. Compared to artificial environments like video games [139, 161, 133, 311, 270], photorealistic environments portray 3D scenes in which real people and mobile robots would interact. Their realistic meshes can be rendered from agent-selected viewpoints to train and test RL policies for navigation in a reproducible manner [7, 29, 312, 150, 13, 265, 25, 313, 247]. Many are captured with 3D scanners and real 360 photos, meaning that the views are indeed the perceptual inputs a robot would receive in the real world [29, 265, 7]. None of the commonly used environments and simulators provide audio rendering. We present the first audio-visual simulator for AI agent training and the first study of audio-visual embodied agents in realistic 3D environments in Chapter 3.

2.3.2 Acoustic Simulation

Sounds are first produced by vibrating objects and then propagate in space before reaching human ears. Modeling sound propagation has a long history in the literature, the goal of which is to simulate realistic high-fidelity audio that is consistent with the given environment specification. Interactive acoustic simulation systems have been extensively used in games and AR/VR applications. Sound propagation algorithms typically fall into two main categories: wave-based [4, 134, 196] and geometric [80, 155, 250]. Wave-based methods aim to solve the wave equation numerically, resulting in high computation expense. In the geometric method family, the

Image-Source Methods [5] solve the specular reflection of sounds deterministically but have low accuracy for late reverb, while path-tracing based approaches offer both high accuracy and efficiency [245]. Aside from sound propagation, some simulators like TDW [84] model impact sounds between objects.

Both SoundSpaces versions use a bidirectional path-tracing algorithm for rendering audio. However, SoundSpaces 2.0 overcomes SoundSpaces' core limitations by enabling on-the-fly rendering, and we also augment the propagation algorithm by adding diffraction and improving reverberation level accuracy. Compared to existing public platforms, SoundSpaces 2.0 adds significant generality and flexibility—accepting arbitrary scene geometry, generalizing to new 3D meshes on the fly, rendering in real-time, and allowing configuration of materials and microphones—all of which we demonstrate.

2.3.3 Novel-View Synthesis (NVS)

Kickstarted by advances in neural rendering [258, 186], recent work considers variants of the NVS problem. Most approaches assume dozens of calibrated images for reconstructing a single static scene. Closer to monocular video NVS, authors have considered reducing the number of input views [208, 125, 234, 319, 156] and modeling dynamic scenes [164, 223, 215, 168, 269, 283]. However, none of this work tackles audio. In Chapter 8, we introduce the first principled treatment of novel-view acoustic synthesis (NVAS).

2.4 Audio Signal Processing and Sound Synthesis

A core problem throughout many of my papers is how to process or generate audio signals. In this section, I will review the literature on audio dereverberation and speech enhancement, as well as acoustic matching and spatialization.

2.4.1 Audio Dereverberation and Speech Enhancement

Audio dereverberation and speech enhancement have a long and rich literature [207, 190, 206, 144, 19]. While dereverberation can be done with microphone arrays, we focus on single audio channel approaches, which require fewer assumptions about the input data. Recent deep learning methods achieve promising results to dereverberate [113, 306, 325, 71, 326, 267], denoise [315, 77, 267], or separate [118, 264] the audio stream using audio input alone, and such enhancements can improve downstream speech recognition [147, 144] and speaker recognition [259]. Acoustic simulations can provide data augmentation during training [144, 113, 147, 326]. Accounting for environmental effects on reverb, some work targets "room-aware" deep audio features capturing reverberation properties (e.g., RT60) [97], or injects reverberation effects from a different room via acoustic matching [266]. To our knowledge, the only prior work drawing on the *visual* stream to infer dereverberated audio is limited to using lip regions on near-field faces to first separate out distractor sounds [274], and does not model anything about the visual scene for dereverberation purposes. In contrast, our dereverberation model in Chapter 6 accounts for the full visual scene, far-field speech sources, and even out-of-view speakers. Our approach is the first to learn visual room acoustics for dereverberation, and it yields state-of-the-art results with direct benefits for multiple downstream tasks.

2.4.2 Acoustic Matching

The goal of *acoustic matching* is to transform an audio recording made in one environment to sound as if it were recorded in a target environment. The audio community deals with this task with various approaches depending on what information about the target environment is accessible. If audio recorded in the target environment is provided, blind estimation of two acoustic parameters, direct-to-reverberant ratio (DRR), which describes the energy ratio of direct arrival sound and reflected sound, and reverberation time (RT60), the time it takes for a sound

to decay 60dB, is sufficient to create simple RIRs that yield plausibly matched audio [65, 82, 145, 176, 194, 314]. Blind estimation of the room impulse response from reverberant speech has also been explored [263, 296]. In music production, acoustic matching is applied to change the reverberation to emulate that of a target space or processing algorithm [152, 243]. Recent work conditions the target-audio generation on a low-dimensional audio embedding [266]. Unlike any of the above audio-only work, in Chapter 7, we introduce and tackle the *visual* acoustic matching problem, where the target environment is expressed via an input image.

2.4.3 Audio Spatialization

While the goal of acoustic matching is to transform the audio to match the acoustics of the environment, the goal of audio spatialization is to match the microphone configuration from single-channel audio, e.g., binaural or ambisonics. Recently, the vision community has explored spatializing sounds with video, which provides cues of the sounding object locations. This work typically spatializes monaural sounds by upmixing them to multiple channels conditioned on the video, where the sound emitters are static [86, 191]. The monaural sounds used as input are obtained by downmixing the target audio. These problems typically assume static emitter and receiver locations, which simplifies the learning problem because acoustics is the same between input and output. In Chapter 8, the novel-view acoustic synthesis is much more complicated because it requires generating audio that both matches the acoustics of different locations as well as the binaural microphone configuration.

Chapter 3: The SoundSpaces Platform

What we see and hear dominates our perceptual experience, and there is often a strong relationship between the two modalities. At the object level, we can anticipate the sounds an object makes based on how it looks, and vice versa (a dog barks, a door slams, a baby cries). At the environment level, materials and geometry of the surrounding 3D space that we see transform the sounds that reach our ears. For example, a person speaking in a marble-floored, high-ceiling museum sounds distinct from one speaking in a cozy carpeted bookshop.

Modeling the correspondence between visuals and acoustics in 3D spaces is of vital importance for many applications in embodied AI and augmented/virtual reality (AR/VR). For instance, a rescue robot needs to localize the person who is calling for help; a service robot needs to look and listen to know if the espresso machine is running properly; an AR system needs to generate sounds that are consistent with the user's acoustical environment for an immersive experience.

Realistic simulations of the first-person perceptual experience are a valuable resource for AI research. They allow training and evaluating models at scale and in a replicable manner. On the visual side, fast visual simulators [247, 273] coupled with 3D assets from scanned real-world environments [29, 312, 265, 228] have facilitated substantial work in visual navigation and related tasks in recent years [302, 36, 9, 126, 231, 131], enabling rigorous benchmarks [210] and even successful "sim2real" transfer to agents that move in the real world [299, 282, 135]. On the audio side, acoustic simulation has been traditionally pursued for physical models [23], gaming [169] and auralization for architectural design [293], typically restricted to simple parametric geometries and in isolation from visual context.

In this chapter, I detail our efforts in building audio-visual simulation platforms

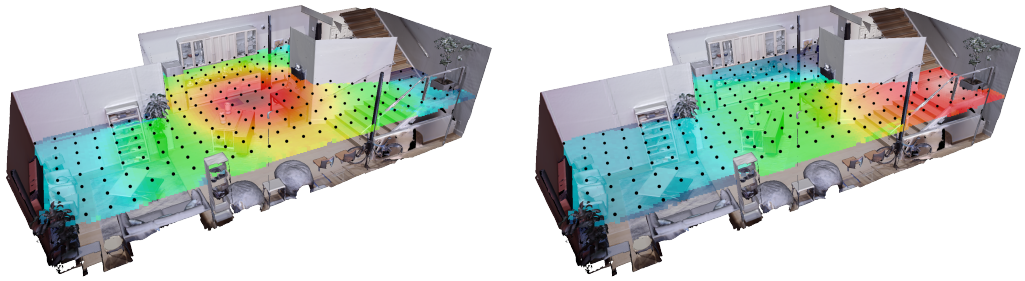


Figure 3.1: **Acoustic simulation in SoundSpaces 1.0.** We capture room impulse responses between each location pair within the illustrated grid (here for the 'frl_apartment_0' scene in Replica). In our platform, agents can experience binaural audio at densely sampled locations $\mathcal{L}$ marked with black dots—hearing the sound's intensity, direction, and frequency texture. Heatmaps display audio pressure fields, decreasing from red to blue. **Left**: When a sound source in $\mathcal{S}$ is placed in the center. **Right**: When a source is placed on the stairs. Notice how the sound received by the agent at different positions changes when the sound source moves, and how 3D structures influence the sound propagation.

to enable embodied audio-visual learning as well as visual-acoustic learning. I will introduce the first SoundSpaces platform [35] that was published in ECCV 2020 in Sec. 3.1 and the enhanced version SoundSpaces 2.0 [40] that was published in NeurIPS 2022 in Sec. 3.2. Both papers introduced both the simulator and the audio-visual navigation benchmark, which I will defer to Sec. 4.1 and Sec. 4.3.

In Sec. 3.1, I present the basics of our acoustic rendering pipeline and how we pre-render impulse responses for discrete locations. Due to the pre-rendered nature, SoundSpaces 1.0 is very fast but limited to discrete locations and environments. In Sec. 3.2, I demonstrate how we build SoundSpaces 2.0 for real-time rendering and customizing configurations as well as environments. SoundSpaces 1.0 is still one order of magnitude faster (500 fps) than SoundSpaces 2.0 (30 fps). Both platforms are valuable to the research community depending on the specific application.

3.1 SoundSpaces: Simulating Sounds in 3D Environments

Our audio platform augments the Habitat simulator [247], particularly the Matterport3D [29] and Replica [265] datasets hosted within it. Habitat is an open-source 3D simulator with a user-friendly API that supports RGB, depth, and semantic rendering. The API offers fast (over 10K fps) rendering and support for multiple datasets [265, 312, 29]. This has incentivized embodied AI community to embrace it as the 3D simulator for training navigation and question answering agents [247, 33, 148].

Matterport3D [29] is a dataset of 85 real-world homes and other indoor environments with 3D meshes and image scans. The environments are large, with on average 517 m^2 of floor space. Replica [265] is a dataset of 18 apartment, hotel, office, and room scenes with 3D meshes. By extending these Habitat-compatible 3D assets with our audio simulator, we enable users to take advantage of the efficient Habitat API and easily adopt the audio modality for AI agent training. Our audio platform and data are publicly available at `https://github.com/facebookresearch/sound-spaces`.

Our high-fidelity audio simulator takes into account important factors for a realistic sound rendering in a 3D environment. We use a state-of-the-art algorithm for room acoustics modeling [27] and a bidirectional path tracing algorithm to model sound reflections in the room geometry [291]. Since materials also influence the sounds received in an environment (e.g., walking across marble floors versus a shaggy carpet), we set the acoustic material properties of major surfaces by mapping the meshes' semantic labels to materials in an existing database [66]. Each material has different absorption, scattering, and transmission coefficients that affect our sound propagation. This enables our simulator to model fine-grained acoustic properties like sound propagation through walls.

For each scene, we simulate the acoustics of the environment by pre-computing room impulse responses (RIR). The RIR is the 1D transfer function between a sound source and microphone, which varies as a function of the room geometry, materials, and the sound source location [158]. RIRs can be convolved with an arbitrary source

audio signal to generate the audio signals received by the microphone [22, 27, 81, 245, 246]

Let $\mathcal{S} = \{(x_i^s, y_i^s, z_i^s)\}_{i=1}^N$ denote the set of N possible sound source positions, and let $\mathcal{L} = \{(x_i^r, y_i^r, z_i^r)\}_{i=1}^N$ denote the set of possible listener positions (i.e., agent microphones). We densely sample a grid of N locations with a spatial resolution of 0.5m (Replica) or 1m (Matterport). The Replica scenes range in area from 9.5 to 141.5 m^2 and thus yield $N \in [38, 566]$; for Matterport the range is 53.1 to 2921.3 m^2, with $N \in [20, 2103]$. Points are placed at a vertical height of 1.5m, reflecting the fixed height of a robotic agent. Then we simulate the RIR for each possible source and listener placement at these locations, $\mathcal{S} \times \mathcal{L}$. Having done so, we can look up any source-listener pair on the fly and render the sound by convolving the desired waveform with the selected RIR. See Figure 3.1.

Given our simulations, for any audio source placed in a location $\mathcal{S}_i$ we can generate the ambisonic audio (roughly speaking, the audio equivalent of a 360° image) heard at a particular listener location $\mathcal{L}_j$. We convert the ambisonics to binaural audio [321] in order to represent an agent with two human-like ears, for whom perceived sound depends on the body's relative orientation in the scene.* Our platform also permits the rendering of multiple simultaneous sounds.

Since an agent might not be able to stand at each location in $\mathcal{L}$ due to embodiment constraints (e.g., no climbing on the sofa), we create a graph capturing the reachability and connectivity of these locations. First we remove nodes that are non-navigable, then for each node pair (i, j), we consider the edge $e(i, j)$ as valid if and only if the Euclidean distance between i and j is 0.5m for Replica or 1m for Matterport (i.e., nodes i and j are immediate neighbors) and the geodesic and Euclidean distances between them are equal (i.e., no obstacle in between). This navigability graph is constructed due to RIRs being rendered on discrete grid. I will show later

*While algorithms could also run with ambisonic inputs, using binaural sound has the advantage of allowing human listeners to interpret our video results.

in Sec. 3.2 that with continuous simulation, no graph construction is required.

3.1.1 Audio Simulation Details

Grid construction. We use an automatic point placement algorithm to determine the locations where the simulated sound sources and listeners are placed in a two-step procedure: adding points on a regular grid and then pruning. For adding points on a regular grid, first, we compute an axis-aligned 3D bounding box of a scene. Within this box we sample points from a regular 2D square grid with resolution 0.5m (Replica) or 1m (Matterport) that slices the bounding box in the horizontal plane at a distance of 1.5m from the floor (representing the height of a humanoid robot).

The second step prunes grid points in inaccessible locations. To prune, we compute how *closed* the region surrounding a particular point is. This entails tracing R uniformly distributed random rays in all directions from the point, then letting them diffusely reflect through the scene up to B bounces using a path tracing algorithm. Simultaneously, we compute the total number of "hits" H: the number of rays that intersect the scene. After all rays are traced, the *closed-ness* $C \in [0, 1]$ of a point is given by $C = \frac{H}{R \cdot B}$. A point is declared outside the scene if $C < C_{min}$. the value of C for a particular point is below a threshold C_{min}. Finally, we remove points that are within a certain distance d_{min} from the nearest geometry, as identified using the shortest length of the initial rays traced from the point in the previous pruning step.

For all scenes we use $R = 1000$, $B = 10$ and $d_{min} = 5$cm. This value of d_{min} was chosen to avoid the placement of points inside walls or in small inaccessible areas. We find $C_{min} = 0.5$ works for most scenes. The exceptions are scenes with open patio areas, where we found $C_{min} = 0.1$ works best to provide a sufficient number of points on the patio.

Materials and transmission model. In addition to its geometry, a room's *materials* affect the RIR. To capture this aspect, we use the semantic labels provided

in Replica to determine the acoustic material properties of the geometry. For each semantic class that was deemed to be acoustically relevant, we provide a mapping to an equivalent acoustic material from an existing material database [66]. For the *floor*, *wall*, and *ceiling* classes, we assume acoustic materials of carpet, gypsum board, and acoustic tile, respectively. This helps simulate more realistic sounds than if a single material were assumed for all surfaces. In addition, we add a ceiling to those Replica scenes that lack one, which is necessary to simulate the acoustics accurately.

The simulation also includes a path-tracing simulation through walls according to their material properties. Each material has absorption, scattering, and transmission coefficients. We use a transmission model similar to that used in graphics rendering. While this is modeled to ensure the precision of the simulation, the impact of transmission is generally small compared to the propagation of sound through open doors [171].

Acoustic simulation technique. During the simulations, we compute the room impulse responses between all pairs of points, producing N^2 RIRs. The simulation technique stems from the theory of geometric acoustics (GA), which supposes sound can be treated as a particle or ray rather than a wave [245]. This class of simulation methods is capable of accurately predicting the behavior of sound at high frequencies, but requires special modeling of wave phenomena (e.g., diffraction) that occur at lower frequencies.Specifically, our acoustic simulation is based on a bidirectional path tracing algorithm [291] modified for room acoustics applications [27]. Additionally, it uses a recursive formulation of multiple importance sampling (MIS) to improve the convergence of the simulation [94].

The simulation begins by tracing rays from each source location in $\mathcal{S}$. These source rays are propagated through the scene up to a maximum number of bounces (200). At each ray-scene intersection of a source path, information about the intersected geometry, incoming and outgoing ray directions, and probabilities are cached.

After all source rays are traced, the simulation traces rays from a listener location in $\mathcal{L}$. These rays are again propagated through the scene up to a maximum number of bounces. At each ray-scene intersection of a listener path, rays are traced to connect the current path vertex to the path vertices previously generated from all sources. If a connection ray is not blocked by scene geometry, a path from the source to the listener has been found. The energy throughput along that path is multiplied by a MIS weight and is accumulated to the impulse response for that source-listener pair. After all rays have been traced, the simulation is finished.

We perform the simulation in parallel for four logarithmically-distributed frequency bands.[†] These bands cover the human hearing range and are uniform in their distribution from a perceptual standpoint. For each band, the simulation output is a histogram of sound energy with respect to propagation delay time at audio sample rate (44.1kHz for Replica and 16kHz for Matterport). Spatial information is also accumulated in the form of low-order spherical harmonics for each histogram bin. After ray tracing, these energy histograms are converted to pressure IR envelopes by applying the square root, and the envelopes are multiplied by bandpass-filtered white noise and summed to generate the frequency-dependent reverberant part of the monaural room impulse response [159].

Ambisonic signals (roughly speaking, the audio equivalent of a 360° image) are generated by decomposing a sound field into a set of spherical harmonic basis. We generate ambisonics by multiplying the monaural RIR by the spherical harmonic coefficients for each time sample. Early reflections (ER, paths of order ≤ 2) are handled specially to ensure they are properly reproduced. ER are not accumulated to the main energy histogram, but are instead clustered together based on the plane equation of the geometry involved in the reflection(s). Then, each ER cluster is added to the final pressure IR with frequency-dependent filtering corresponding to the ER energy and its spherical harmonic coefficients.

[†][0Hz,176Hz], [176Hz,775Hz], [775Hz,3409Hz], [3409Hz,20kHz]

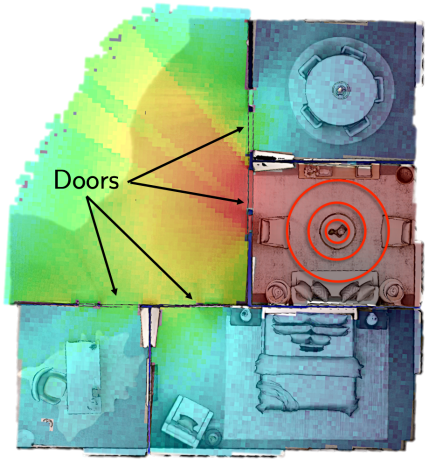


Figure 3.2: **Pressure field of audio simulation** overlaid on the top-down map of apartment_2 from Replica [265]. Our audio-enabled agent gets rich directional information about the goal, since the pressure field variation is correlated with the shortest distance. Notice the discontinuities across walls and the gradient of the field along the *geodesic* path an agent must use to reach the goal (different from shortest Euclidean path). As a result, to an agent standing in the top right or bottom rooms, the audio reveals the door as a good intermediate goal. In other words, the audio stream signals to the agent that it must leave the current room to get to the target. In contrast, the GPS displacement vector would point through the wall and to the goal, which is a path the agent would discover it cannot traverse. Note that the visual stream is essential to couple with the audio stream in order to navigate around obstacles.

The result of this process is second-order ambisonic pressure impulse responses that can be convolved with arbitrary new monaural source audios to generate the ambisonic audio heard at a particular listener location. We convert the ambisonics to binaural audio [321] in order to represent an agent with two human-like ears, for whom perceived sound depends on the body's relative orientation in the scene.

3.1.2 Visualizing Audio Simulations

Next, we illustrate the pressure field visualization of two other scenes in the Replica dataset. In Fig. 3.2, we display another big scene (apartment_2) with four rooms, with the audio source inside one of the rooms. Notice how the pressure

decreases from the source along geodesic paths, which leads to doors serving as secondary sources or intermediate goals that lead the agent in the right direction.

3.2 SoundSpaces 2.0: A Simulation Platform for Visual-Acoustic Learning

SoundSpaces's [35] foremost limitation is its pre-computed, discretized nature. The provided RIRs are pre-computed for all source and receiver pairs on a 0.5m grid, and for a fixed list of 100 total environments. While this has the advantage of fast rendering, it prevents sampling data at new locations. This in turn means that 1) an agent in the simulator can only move or hop between discrete grid points in the space, which abstracts away some difficult parts of the navigation task; 2) the simulations do not generalize to novel environments—just the 100 provided; and 3) the pre-computed data itself is on the order of TBs, impeding the ability to change configurations, e.g., of the microphone types or materials. ThreeDWorld [84] offers continuous-space rendering, yet it only supports audio rendering for simple 3D environment geometry, namely an oversimplified "shoebox" (rectangular parallelepiped) model, and thus is not applicable to real-scan datasets [312, 228]. In sum, existing audio-visual rendering platforms fall short in accuracy, speed, and flexibility, which in turn constrains the scope of research tasks they can support within audio-visual embodied learning [36, 62, 320] and visual-acoustic learning [257, 173, 39].

In this section, I introduce SoundSpaces 2.0, which performs on-the-fly geometry-based audio rendering for arbitrary environments. Like SoundSpaces 1.0, it accounts for all major real-world acoustic factors: direct sounds, early specular/diffuse reflections, reverberation, binaural spatialization, and frequency-dependent effects from materials and air absorption. Unlike the original SoundSpaces, it allows highly realistic rendering of arbitrary camera views and arbitrary microphone placements for waveforms of the user's choosing, accounting for. Furthermore, unlike SoundSpaces 1.0, SoundSpaces 2.0 generalizes audio simulation to *any* input mesh, making it pos-

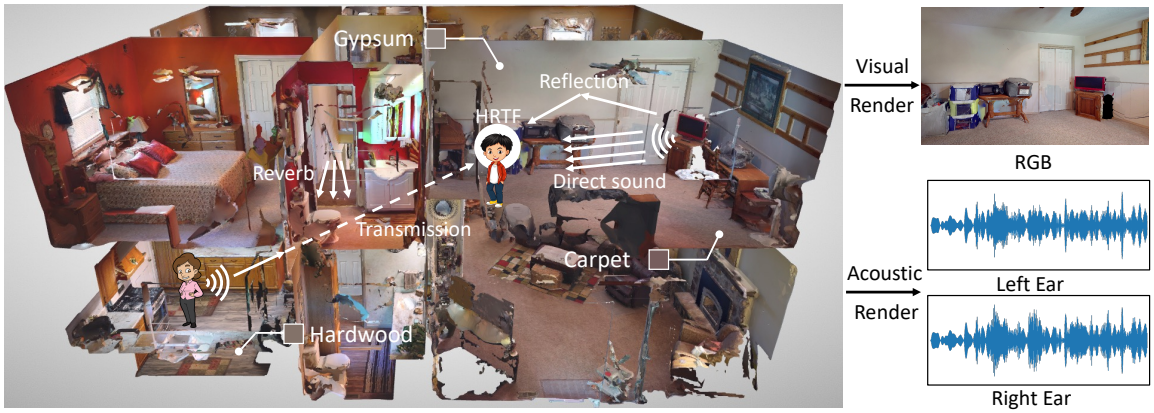


Figure 3.3: Illustration of SoundSpaces 2.0 rendering in a multi-room multi-floor HM3D [228] environment. In this scenario, a boy is watching TV in the living room while his mom calls him to have dinner from the kitchen downstairs. We model various frequency-dependent acoustic phenomena for sound propagation from all sources (TV and mom) to him, including direct sound, reflection, reverb, transmission, diffraction and air absorption. The sound propagation is based on a bidirectional path-tracing algorithm that takes the geometry of the scene as well as materials of objects in the space as input. The received sound is spatialized to binaural with the head-related transfer function (HRTF). As a result, SoundSpaces 2.0 renders the visual and audio observations with *spatial and acoustic correspondence*. For example, the TV being situated more towards the right results in right-ear signals that are stronger than those in the left ear.

sible for the first time to import sound into well-used environment assets like Gibson [312], HM3D [228], and Matterport3D [29], as well as any future or emerging one like Ego4D [105]. In addition, SoundSpaces 2.0 allows users to configure various properties of the simulation, such as source-receiver locations, simulation parameters, material properties, and microphone configuration. The rendering platform (illustrated in Fig. 3.3.) and associated research codebase are publicly available. [‡]

I will describe the new platform and its functionality, and I illustrate its flexibility with various concrete examples. In addition, I perform systematic experiments to answer two questions: 1) How accurate are the audio-visual simulations? and 2) How well can machine learning models trained in SoundSpaces 2.0 generalize to

[‡]`https://github.com/facebookresearch/sound-spaces`

Platform	Audio-Visual	Geometric	Configurable	Arbitrary Env
SoundSpaces [35]	✓	✓	✗	✗
GWA [275]	✗	✓	✓	✗
ThreeDWorld [84]	✓	✗	✓	✗
Pyroomacoustics [248]	✗	✗	✓	✗
SoundSpaces 2.0 (Ours)	✓	✓	✓	✓

Table 3.1: Comparison with existing non-commercial datasets/simulation platforms. *Geometric* refers to acoustic simulation that is based on geometry of the objects and the space. *Configurable* means ability to alter simulation parameters, material and microphone properties. *Arbitrary Env* refers to the ability to render for an arbitrary new mesh environment, including point clouds generated in the wild.

real-world data? For this purpose, we collect real-world audio RIR measurements for a public scene dataset Replica [265] and benchmark the simulation accuracy. We also benchmark two downstream tasks: continuous audio-visual navigation (discussed in Sec. 4.3) and far-field speech recognition. For speech recognition, we show that machine-learning models trained on our synthetic data can generalize when tested on real data. We also propose an acoustic randomization technique that models the real-world distribution of materials' acoustic properties, and we show that this strategy leads to better sim2real generalization.

3.2.1 Rendering Pipeline

In this section, I will detail the features of SoundSpaces 2.0, including the simulation enhancement, continuity, configurability, generalizability and the rendering modes.

3.2.1.1 Rendering Pipeline and Simulation Enhancements

The core of SoundSpaces 2.0 is the audio propagation engine (RLR-Audio-Propagation) we are releasing for research purposes.[§] We integrate this engine into

[§]`https://github.com/facebookresearch/rlr-audio-propagation`

the existing visual simulator Habitat-Sim [247], which offers fast visual rendering.[¶] In addition, we provide high-level APIs for various downstream tasks (e.g., navigation) and training scripts at the SoundSpaces repo.[‖]

Fig. 3.3 illustrates the propagation pipeline. Similar to SoundSpaces 1.0, SoundSpaces 2.0 takes the scene mesh data processed by Habitat, together with source and receiver locations specified by the user, and computes a room impulse response (RIR) using a bidirectional path-tracing algorithm [27]. This module models various acoustic phenomena, including reflection, transmission, and diffraction, as well as spatialization. The simulation operates in M logarithmically-spaced frequency bands (configurable), where it computes an energy-time histogram at the audio sampling rate. This histogram incorporates spatial information using spherical harmonics for each time sample that represents the directional distribution of arriving sound energy. This representation is then spatialized to either an ambisonic or binaural pressure impulse response [252], which can be convolved with the source audio signals to generate the sound at the receiver position.

Compared to SoundSpaces 1.0 in Sec. 3.1, we have improved the simulation in a few ways. SoundSpaces did not include any simulation of acoustic diffraction, and thus exhibited abrupt occlusion of sources. We have removed this limitation using the fast diffraction approach from [253], which is able to efficiently compute smooth diffraction effects for occluded sources. We also improved the accuracy of the direct-to-reverberant ratio (DRR), the ratio of the sound pressure level of a direct sound from a directional source to the reverberant sound pressure level, by fixing a bias of $\sqrt{4\pi}$ that was present in the indirect sound pressure of the original SoundSpaces.

In the following, we overview modeling advances in SoundSpaces 2.0 that promote continuity, configurability, generalizability, and performance.

[¶] https://github.com/facebookresearch/habitat-sim/blob/main/docs/AUDIO.md
[‖] https://github.com/facebookresearch/sound-spaces

3.2.1.2 Continuity

Spatial continuity. Humans move around in the real world continuously while hearing. Given an arbitrary source location s, receiver location r, and receiver's heading direction θ in a given mesh environment, we render the impulse response between the source and receiver as $R(s, r, \theta)$. The sound received by the receiver is computed as $A^r = A^s * R(s, r, \theta)$, where A^s is the sound emitted from the source and $*$ denotes convolution. Whereas SoundSpaces [35] restricts the s and r locations to a 0.5m discrete grid due to its pre-computed approach and hefty storage requirements, SoundSpaces 2.0 allows arbitrary placements.

Acoustic continuity. While an agent moves in the environment, it moves smoothly from point A to point B (even with a small step size). With the spatial continuity property, we can render $R(s, r_A, \theta_A)$ and $R(s, r_B, \theta_B)$ for these two locations respectively. However, the original SoundSpaces takes the rendered IR for each location and convolves it with the source sound directly as the audio observation. This calculation implicitly assumes the source does not emit sound continuously, i.e., it starts to emit when the agent moves to a new location, stops after one second, and resumes at the agent's next location.

In SoundSpaces 2.0, we introduce acoustic continuity for both the source sound and listener. More specifically, given a sampling rate F and the time between two steps Δt, the number of received audio samples is $N = F\Delta t$ per step. Assuming a listener is at location x_i at time t_i, the audio signal received by the listener at time t_i emitted from the source at time t_p is $t_i - R(s, x_i, \theta_{x_i}) + 1$. We take the corresponding source sound segment $A^s[t_p : t_p + N]$ and convolve it with $R(s, x_i, \theta_{x_i})$ without zero padding to compute $A^{x_i}_{t_i}$. Following the common practice [197], we apply linear crossfading between $A^{x_{i-1}}_{t_i}$ and $A^{x_i}_{t_i}$ to smooth out the transition from x_{i-1} to x_i with an overlap time window of T seconds.

3.2.1.3 Configurability

Due to its pre-computed nature, it is impossible to change any simulation setup (parameters, microphones, or materials) for the original SoundSpaces. All are configurable in SoundSpaces 2.0, as summarized below.

Simulation parameters. We expose many useful parameters for users to configure, including the sampling rate, the number of frequency bands, number of rays for direct/indirect sounds, whether reflection, transmission or diffraction is enabled, etc.

Microphone types. We provide several types of built-in microphone configurations, including monaural single-channel audio, binaural (modeling a human listener), and ambisonics (full sphere surround sound). In addition, users are also able to configure their own microphone array by specifying an array of monaural microphone locations.

Custom HRTFs. We allow users to load their own head-related transfer functions (HRTFs), which incorporate customized human perception in the acoustic rendering simulation.

Material modeling. Materials of objects/surfaces have a big impact on how humans perceive the sound in an environment. Consider the difference between sound in a recording studio versus a living room of the same size. Due to the absorptive materials in the recording studio, the sound will consist primarily of direct sound without reverberation, whereas in the living room, the sound will consist of a mixture of direct sound and reverberation.

Existing real-scan datasets have semantic annotations at the level of object categories, e.g., chair, table, couch and floor, while lacking material annotations of what these objects are made of, e.g., wood or steel for tables. SoundSpaces coped

with this issue by defining a fixed mapping from object categories to acoustic materials, e.g., floors are always mapped to the carpet material, which is very absorptive. However, this fixed mapping fails to reflect the fact in the real world, different instances of the same object category could have very different acoustic properties, e.g., a floor could be carpet or wood or concrete materials depending on the home type.

To account for this variation, we expose an API to let users define their own acoustic material configurations. We provide 29 built-in acoustic materials, e.g., wood, concrete, curtain, soil, water. Every acoustic material has a list of candidate object categories to be mapped from. It also has a set of coefficients for absorption, scattering, and transmission in the following format: $[f_1, c_1, f_2, c_2, ..., f_n, c_n]$, where f_i is a frequency and c_i is the coefficient for a certain acoustic phenomenon at frequency f_i. This allows modeling the frequency-dependent acoustic properties of different acoustic materials. For example, high-frequency waves are absorbed more compared to low frequencies when reflecting from carpets.

We also model distance-dependent damping of the sound propagation media. This includes air absorption as well as transmission losses through materials. Air absorption is calculated using an analytical model [16]. Users can specify the frequency-dependent damping coefficients for each material, expressed as dB per meter, in a similar format to the other material properties.

3.2.1.4 Generalizability

Generalization to scene datasets. Our new simulator accommodates arbitrary 3D meshes as input. This makes it compatible with all available scene datasets (e.g., Gibson [312], HM3D [228], Ego4D [105], Matterport3D [29], Replica [265]), as well as any future assets that become available, such as if a user scans their own lab or home environment. This is an important advance over SoundSpaces, which was restricted to Replica and Matterport3D alone.

Generalization to shoebox rooms. We expose APIs for creating shoebox rooms with different materials for walls, which simulate simpler setups as in Pyroomacoustics [248] and TDW [84].

Generalization to the real world. The fidelity and flexibility of our simulation platform also supports generalization to the real world. In Sec. 3.2.2, we score the simulator output against real-world RIRs and show how machine learning models trained on SoundSpaces 2.0 can generalize to real data.

3.2.1.5 Rendering Modes and Rendering Performance

Our simulation generates high-quality audio rendering based on mesh and materials, and this fidelity can be instrumental for certain research areas. On the other hand, in tasks like embodied navigation with reinforcement learning, which typically require millions (or even billions [302]) of training iterations, rendering speed is of vital importance. Thus, we offer two built-in rendering modes: *high-speed* and *high-quality*.

In high-speed mode, we reduce the number of rays and improve the accuracy by leveraging previously computed impulse responses [249], under the assumption that movements are spatially continuous. Our algorithms use information computed on previous simulation frames, such as sound propagation paths and RIRs, to reduce the number of rays and ray bounces that are needed on each frame for sufficient sound quality (see Sec.3.2.2.1). In high-quality mode, we set all rendering parameters to max and turn off the temporal coherence feature to ensure that every impulse response is accurate without temporal blurring. Our engine is multi-threaded and users can set the number of threads when using either mode. See Sec. 3.2.2.1 for analysis of the simulation performance in terms of speed and accuracy.

Despite that the high-speed mode can reach real-time performance (30 fps) in SoundSpaces 2.0, the rendering is still one magnitude slower than SoundSpaces

1.0 (500+ fps). As a result, for applications that require high rendering speed, e.g., online RL training, SoundSpaces 1.0 is still very useful.

3.2.2 Evaluation and Benchmarks

Next we evaluate both the simulation quality and its value for downstream tasks with two machine learning benchmarks. Fig. 3.4a illustrates these two tasks.

3.2.2.1 Simulation Speed vs. Quality Tradeoff

To understand the tradeoff between the quality versus speed of rendering, we report the accuracy and speed of different modes by rendering RIRs along random trajectories with an average length of 15m across 20 Matterport3D environments. We profile the speed on a Xeon(R) Gold 6230 CPU with 2.10GHz. See Table 3.2. For accuracy, we measure the relative RT60 error of RIRs generated in high-speed mode compared to RIRs generated in high-quality mode. RT60 is a standard acoustic measurement that is defined as the time it takes for the sound pressure level to reduce by 60 dB [124]. We see high-speed greatly improves efficiency over the high-quality mode, by 8× with single thread and 33× with 5 threads, while only losing 9.5% accuracy despite RT60 calculation being noisy. When coupled with distributed training, it meets the requirements of today's RL agent training. In addition, we test the navigation model trained in high-speed mode on high-quality mode; the performance difference is smaller than 1% compared to the test performance in high-speed mode in Table 4.6. In comparison, TDW [84] runs at 60 FPS and SoundSpaces runs at 500+ FPS (bottleneck on I/O) at the cost of simplified room models or not being configurable, respectively, c.f. Table 3.1. We treat high-quality mode as the gold standard and benchmark its quality against real-world IRs next.

3.2.2.2 Validating Simulation Accuracy with Real IRs

How realistic are our audio simulations? To quantify this, we collect real acoustic measurements of the FRL apartment from the Replica dataset [265] and compare

Table 3.2: Simulation speed vs. quality tradeoff. We report mean and standard deviation over 5 runs.

	Relative RT60 Error (%)	1 Thread (FPS)	5 Threads (FPS)
High-quality	0.0 ± 0.0	0.9 ± 0.0	4.0 ± 0.1
High-speed	9.5 ± 0.2	7.7 ± 0.2	33.5 ± 0.4

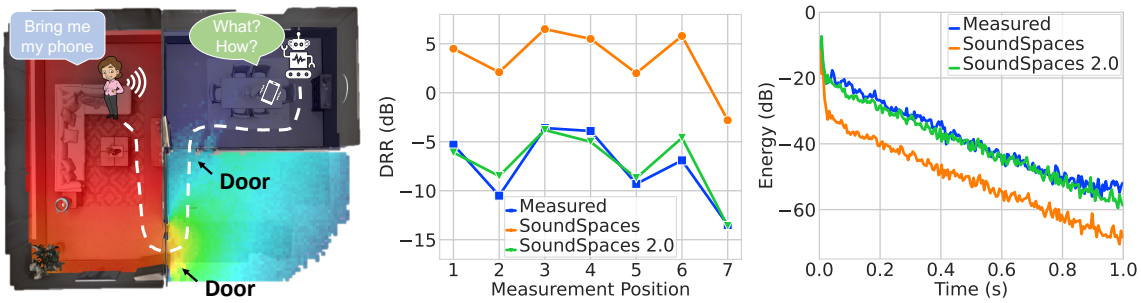
them to SoundSpaces 2.0 outputs. IR measurements were captured at seven different source/receiver positions throughout the real-world apartment using an omnidirectional B&K Type 4295 speaker (100Hz to 8kHz frequency response) and Earthworks M30 microphone with the exponential sine sweep method. These measurements are publicly available to assist future research.

Figure 3.4b compares the measurements to the corresponding simulations at the same source/receiver positions, for both the original SoundSpaces and the proposed SoundSpaces 2.0 (high-quality mode). ** We report the direct-to-reverberant ratio (DRR) acoustic parameters derived from the impulse responses [124] in Figure 3.4b. SoundSpaces 2.0 has a better match of direct-to-reverberant ratio, where the error compared to measurements is reduced from 11.0 dB to 0.98 dB on average, while preserving the same relative RT60 error of 12.4%. Figure 3.4c reinforces that advantage, plotting the energy-time curves of the simulations versus the real measurements from 250Hz to 4000Hz. Overall, the proposed new features and improvements lead to higher realism for the acoustic simulation.

3.2.2.3 Far-Field Automatic Speech Recognition

Speech recognition is critical for many applications, including far-field scenarios where the speaker is far from the microphone (e.g., speaking to a smart home assistant device). When speech recognition models are trained on a clean speech corpus, such as LibriSpeech [214], they generalize poorly to far-field cases with unanticipated

**The measurements were scaled to match the direct sound level of the simulations. The acoustic material properties of the mesh were optimized to match the measurements following [251].



(a) Task illustration (b) DRR comparison (c) Energy decay comparison

Figure 3.4: **(a)** In this example, a person's phone rings in the dining room while she is in the living room and she asks the robot to bring her the phone. Upon receiving the audio signal with the binaural microphone, the robot needs to figure out two things: 1) what she is saying (far-field automatic speech recognition) and 2) how to navigate to her and the phone (audio-visual navigation). Note that far-field ASR is not limited to robotics; it has various applications such as video captioning. **(b)** Comparing real measurements and simulations in the Replica apartment [265] for 7 measurement positions and the 250Hz to 4000Hz frequency band. SoundSpaces 2.0 has a much lower error for the direct-to-reverberant ratio (DRR) compared to SoundSpaces. **(c)** Energy decay curve comparisons. The energy decay curve of SoundSpaces 2.0 is much closer to the real measurements than SoundSpaces.

reverberation. Due to the high expense of collecting real IRs, synthetic impulse responses are thus often used to augment speech for far-field ASR [147, 179, 275]. Here, we propose to benchmark far-field ASR systems augmented by our generated impulse responses.

We take the pretrained transformer-based ASR system from SpeechBrain [233], an open-sourced speech toolkit, as the base model. For finetuning, we augment speech in the train-clean-100 split of LibriSpeech [214] with IRs generated in different systems and finetune 60 epochs. For testing, we augment speech from a real RIR dataset [272], where IRs are recorded in real environments, e.g., home, conference rooms, auditoriums. In this way, we test the sim2real generalization for models trained on the synthetic data. We compare the pretrained model with the ASR model finetuned on IRs generated with Pyroomacoustics, SoundSpaces 1.0, and SoundSpaces 2.0 (high-quality mode). We ensured the simulated RIRs have matching RT60 distributions.

Table 3.3: Far-field automatic speech recognition benchmark.

	Word Error Rate (%)
Pretrained	29.10
Finetuned on real IRs [147]	13.32
Finetuned on Pyroomaoustics [248]	16.24
Finetuned on SoundSpaces 1.0 [35]	18.48
Finetuned on SoundSpaces 2.0	**12.48**

In addition, we compare with the ASR model finetuned on real IRs [147] from the RWCP sound scene database [204], the 2014 REVERB challenge database [144], and the Aachen impulse response database (AIR) [129].

Table 3.3 shows the results. As we can see, the pretrained model generalizes poorly to far-field speech with word error rate (WER) of 29.1%, compared to 2.4% WER on a clean test set lacking any reverberation. Finetuning with synthetic IRs leads to a dramatic improvement. Comparing Pyroomacoustics and SoundSpaces 1.0 to SoundSpaces 2.0, our generated IRs lead to much lower WER. Finetuning on real IRs also reduces the error substantially, but still not as much as our simulated data, which can be generated at scale across a wide variety of environments. Our simulation generates realistic IRs that help machine learning models generalize better to reality.

Acoustic randomization. In the real world, instances of a given object category need not share identical material profiles. While existing simulations do not model such nuances, in SoundSpaces 2.0 we can manipulate the materials in a more subtle way. Inspired by domain randomization techniques [282, 286] that randomize simulation parameters for better sim2real generalization, we explore if acoustic randomization offers similar benefits. Specifically, we define a set of possible acoustic materials for each object category. When rendering, a random material is picked for a category to simulate the category-level variation. In addition, to model the differences of acoustic materials, we add $\mathcal{N}(0, 0.1)$ Gaussian noise to each coefficient. Altogether, this strategy models both the category-level and instance-level material

nuances.

When we use the proposed acoustic randomization technique to generate the same amount of data for finetuning, the ASR model has even lower WER on the test set, reduced from 12.48% to 12.04%, while uniform randomization, i.e., uniformly sampling coefficients between 0 and 1, leads to a higher WER of 12.58%. This not only shows the benefit of acoustic randomization but also how SoundSpaces 2.0's configurability facilitates research on acoustic sim2real.

3.3 Conclusions

We believe SoundSpaces can facilitate significant new work in embodied AI, multimodal perception, and audio research. The platform is general and accessible, and our experiments offer concrete examples of its potential. This platform has been used widely for various research, for example, learning neural acoustic fields [173], audio-visual floorplan reconstruction [225], active audio-visual separation [177], and audio localization from motion [49]. Some of the following chapters also utilize this platform, including embodied audio-visual navigation (Chapter 4 and Chapter 5), audio-visual dereverberation (Chapter 6), visual acoustic matching (Chapter 7 and novel-view acoustic synthesis (Chapter 8). Beyond the vision community, the audio community also sees benefits of this platform, e.g., the L3DAS23 Challenge was organized in ICASSP 2023 based on data rendered with SoundSpaces.

Like any research tool, there are certain limitations and assumptions that are important to recognize. Our simulation platform supports audio rendering for arbitrary environments with a state-of-the-art path-tracing algorithm. For this algorithm to render accurately, the scene meshes need to have high quality, i.e., no large open holes on the mesh, otherwise the rays will leak from the holes, resulting in inaccurate simulation. To aid users in checking the mesh quality for audio rendering, we expose an API to let users check the percentage of rays leaked from the mesh; users can repair the mesh accordingly if the ray efficiency is low. Path tracing is also vulnerable

to the standard shortcomings of geometrical-acoustics techniques, e.g., room modes, though our implementation takes care to eliminate the typical lack of diffraction as described in Sec. 3.1.

Materials have an impact on audio simulation, and one of the open challenges of material modeling is that it is infeasible to accurately know the acoustic material properties only given the environment meshes, e.g., we cannot estimate how much energy the floor absorbs purely based on the mesh or rendered visuals. Currently, we tackle that by assigning common material properties to objects (Sec. 3.2.1.3), which allows our simulator to operate with fairly lightweight assumptions about the incoming mesh. For more in-depth treatment of materials, one could perform acoustic measurements into the environment scanning pipeline when creating a digital replica of a real-world environment.

In this work, we validate the simulation accuracy with real IRs collected in the apartment from the Replica dataset. To improve the simulation accuracy and further understand its difference from the real world, future work could collect acoustic measurements in diverse environments with varying geometry and materials, which is supported by our simulation platform (Sec. 3.2.1.4).

Chapter 4: Physical Audio-Visual Navigation

In Chapter 3, I introduced the SoundSpaces simulation platform, which simulates sounds as a function of the spatial configuration. This platform has enabled various different applications, one of them being the active perception of sight and sound in the environment. For robots with multimodal perception, the ability to move around and reach a goal is essential. In this chapter, I will present the *physical* audio-visual navigation task, where an agent navigates to a single point that emits sounds. Later in Chapter 5, I will present the *semantic* audio-visual navigation task, where the sound is emitted from a semantic object.

Embodied agents perceive and act in the world around them, with a constant loop between their sensed surroundings and their selected movements. Both sights and sounds constantly drive our activity: the laundry machine buzzes to indicate it is done, a crying child draws our attention, and the sound of breaking glass may require urgent help.

In embodied AI, the *navigation* task is of particular importance, with applications in search and rescue or service robotics, among many others. Navigation has a long history in robotics, where a premium is placed on rigorous geometric maps [279, 114]. More recently, researchers in computer vision are exploring models that loosen the metricity of maps in favor of end-to-end policy learning and learned spatial memories that can generalize to visual cues in novel environments [327, 110, 109, 244, 8, 188, 247].

However, while current navigation models tightly integrate seeing and moving, they are deaf to the world around them. This poses a significant sensory hardship: sound is key to (1) understanding a physical space and (2) localizing sound-emitting targets. As leveraged by blind people and animals who perform sonic navigation, acoustic feedback partially reveals the geometry of a space, the presence of occluding

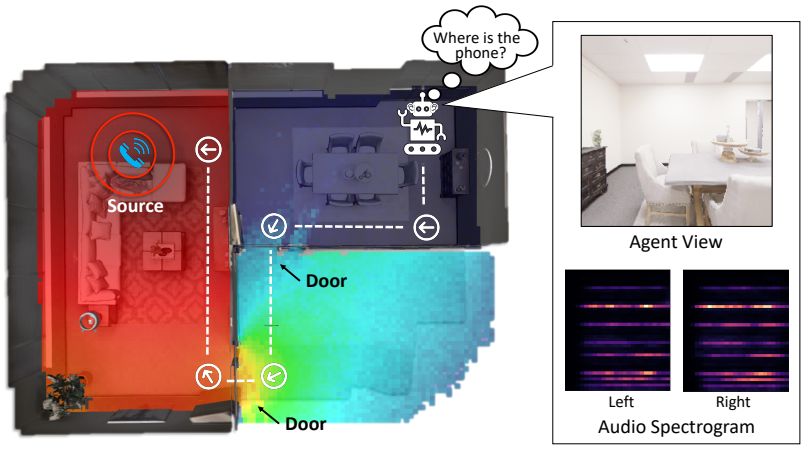


Figure 4.1: **Audio source in an unmapped** 3D environment, where an autonomous agent must navigate to the goal. The top-down map is overlaid with the acoustic pressure field heatmap. Our audio-enabled agent gets rich directional information about the goal, since the audio intensity variation is correlated with the shortest path distance. The acoustics also reveal the room's geometry, major structures, and materials. Notice the gradient of the field along the *geodesic* path an agent must use to reach the goal (different from the shortest Euclidean path, which would cut through the inner wall). As a result, the proposed agent enjoys the synergy of both modalities: audio reveals the door as a good intermediate goal, while vision reveals the physical obstacles along the path, such as the furniture in the lefthand room.

objects, and the materials of major surfaces [219, 72]—all of which can complement the visual stream. Meanwhile, targets currently outside the visual range may be detectable *only* by their sound (e.g., a person calling from upstairs, the ringing phone occluded by the sofa, footsteps approaching from behind). Finally, aural cues become critical when visual cues are unreliable (e.g., the lights flicker off) or orthogonal to the agent's task (e.g., a rescue site with rubble that breaks prior visual context).

Motivated by these factors, I first introduce the audio-visual navigation benchmark presented in the original SoundSpaces paper in Sec. 4.1. I then improve the navigation performance with a hierarchical policy in Sec. 4.2, which was published in ICLR 2021 [38]. I also cover the continuous audio-visual navigation benchmark presented in SoundSpaces 2.0 [40] in Sec. 4.3. Lastly, I introduce a frequency-adaptive sim2real model that transfers audio-visual navigation policies trained in the simula-

tion to the real world in Sec. 4.4.

Besides publications, I have organized the SoundSpaces Challenge * at the Embodied AI Workshop † at CVPR 2021, CVPR 2022 and CVPR 2023, and more than 12 teams have participated in the challenge. There have been other follow-up work that presents variations of audio-visual navigation, for example, adversarial audio-visual navigation [320], audio-visual navigation with dynamic sound sources [318] and audio-visual-language navigation [217].

4.1 Audio-Visual Navigation Benchmark in SoundSpaces

In this section, I introduce *audio-visual navigation* for complex, visually realistic 3D environments. The autonomous agent can both see and hear while attempting to reach its target. We consider two variants of the navigation task: (1) *AudioGoal*, where the target is indicated by the sound it emits, and (2) *AudioPointGoal*, where the agent is additionally directed towards the goal location at the onset. The former captures scenarios where a target initially out of view makes itself known aurally (e.g., phone ringing). The latter augments the popular PointGoal navigation task [8] and captures scenarios where the agent has a GPS pointer towards the target, but should leverage audio-visual cues to navigate the unfamiliar environment and reach it faster.

We propose a multi-modal deep reinforcement learning (RL) approach to train navigation policies end-to-end from a stream of audio-visual observations. Importantly, audio observations must be generated with respect to both the agent's current position and orientation as well as the physical properties of the 3D environment. With the previously introduced SoundSpaces platform (Sec. 3.1), the proposed embodied AI agent learns a policy to choose motions in a novel, unmapped environment that will bring it efficiently to the target while discovering relevant aspects of the latent environment map. See Figure 4.1.

*https://soundspaces.org/challenge

†https://embodied-ai.org/

Our results show the powerful synergy between audio and vision for navigation. The agent learns to blend both modalities to map novel environments, and doing so yields faster learning at training time and faster, more accurate navigation at inference time. Furthermore—in one of our most exciting results—we demonstrate that for an audio goal, the audio stream competes well with the goal displacement vectors upon which current navigation methods often depend [8, 247, 102, 148, 33], while having the advantage of not assuming perfect GPS odometry. Finally, we explore the agent's ability to generalize to not only unseen environments, but also unheard sounds.

4.1.0.1 The audio-visual navigation task

We propose two novel navigation tasks: AudioGoal Navigation and Audio-PointGoal Navigation. In AudioGoal, the agent hears an audio source located at the goal—such as a phone ringing—but receives no direct position information about the goal. AudioPointGoal is an audio extension of the PointGoal task studied often in the literature [8, 247, 102, 313, 148, 33] where the agent hears the source and is told its displacement from the starting position. In all three tasks, to navigate and avoid obstacles, the agent needs to reach the target using sensory inputs alone. That is, no map of the scene is provided to the agent.

Task definitions. For PointGoal [8, 247, 302], a randomly initialized agent is tasked with navigating to a point goal defined by a displacement vector (Δ_x^0, Δ_y^0) relative to the starting position of the agent. For AudioGoal, the agent instead receives audio from the sounding target; the AudioGoal agent does not receive a displacement vector pointing to the target. The observed audio is updated as a function of the location of the agent, the location of the goal, and the structure and materials of the room. In AudioPointGoal, the agent receives the union of information received in the PointGoal and AudioGoal tasks, *i.e.*, , audio as well as a point vector. Note that physical obstacles (walls, furniture) typically exist along the displacement vector, which the agent must sense while navigating.

Agent and goal embodiment. We adopt the standard cylinder embodiment used in Habitat. A target has diameter 0.2m and height 1.5m, and, consistent with prior PointGoal work, has no visual presence. While the goal itself does not have a visible embodiment (currently unsupported in Habitat), vision—particularly in the abstraction of depth—is essential to detect and avoid obstacles to move towards the target. Hence, all the tasks have a crucial vision component.

Action space. The action space is: *MoveForward*, *TurnLeft*, *TurnRight*, and *Stop*. The last three actions are always valid. The *MoveForward* action is invalid when the agent attempts to traverse from one node to another without an edge connecting them (as per the graph defined in Sec. 3.1). If valid, *MoveForward* takes the agent forward by 0.5m (Replica) or 1m (Matterport). For all models, there is no actuation noise, *i.e.*, , a step executes perfectly or does not execute at all.

Sensors. The sensory inputs are binaural sound (absent in PointGoal), GPS (absent in AudioGoal), RGB, and depth. To capture binaural spatial sound, the agent emulates two microphones placed at human height. We assume an idealized GPS sensor, following prior work [247, 33, 102, 148]. However, as we will demonstrate in results, our audio-based learning provides a steady navigation signal that makes it feasible to disable the GPS sensor for the proposed AudioGoal task.

Episode specification. An episode of PointGoal is defined by an arbitrary 1) scene, 2) agent start location, 3) agent start rotation, and 4) goal location. In each episode the agent can reach the target if it navigates successfully. An episode for AudioGoal and AudioPointGoal additionally includes a source audio waveform. The waveform is convolved with the RIR corresponding to the specific scene, goal, agent location and orientation to generate dynamic audio for the agent. We consider a variety of audio sources, both familiar and unfamiliar to the agent (detailed below). An episode is successful if the agent executes the *Stop* action while being exactly at the location

of the goal. Agents are allowed a time horizon of 500 actions for all tasks, similar to [247, 127, 33, 102, 148].

4.1.1 Navigation Network and Training

To navigate autonomously, the agent must be able to enter a new yet-unmapped space, accumulate partial observations of the environment over time, and efficiently transport itself to a goal location. Building on recent embodied visual navigation work [327, 110, 109, 8, 188, 247], we take a deep reinforcement learning approach, and we introduce audio to the observation. During training, the agent is rewarded for correctly and efficiently navigating to the target. This yields a policy that maps new multisensory egocentric observations to agent actions.

Sensory inputs. The audio inputs are spectrograms, following literature in audio learning [213, 322, 86]. Specifically, to represent the agent's binaural audio input (corresponding to the left and right ear), we first compute the Short-Time Fourier Transform (STFT) with a hop length of 160 samples and a windowed signal length of 512 samples, which corresponds to a physical duration of 12 and 32 milliseconds at a sample rate of 44100Hz (Replica) and 16000Hz (Matterport). By using the first 1000 milliseconds of audio as input, STFT gives a 257×257 and a 257×101 complex-valued matrix, respectively; we take its magnitude and downsample both axes by a factor of 4. For better contrast we take its logarithm. Finally, we stack the left and right audio channel matrices to obtain a $65 \times 65 \times 2$ and a $65 \times 26 \times 2$ tensor, denoted A. The visual input V is the RGB and/or depth image, $128 \times 128 \times 3$ and $128 \times 128 \times 1$ tensors, respectively, where 128 is the image resolution for the agent's 90° field of view. The relative displacement vector $\Delta = (\Delta_x, \Delta_y)$ points from the agent to the goal in the 2D ground plane of the scene.

Which specific subset of these three inputs (audio, visual, vector) the agent receives depends on the the agent's sensors and the goal's characterization (cf. Sec. 4.1).

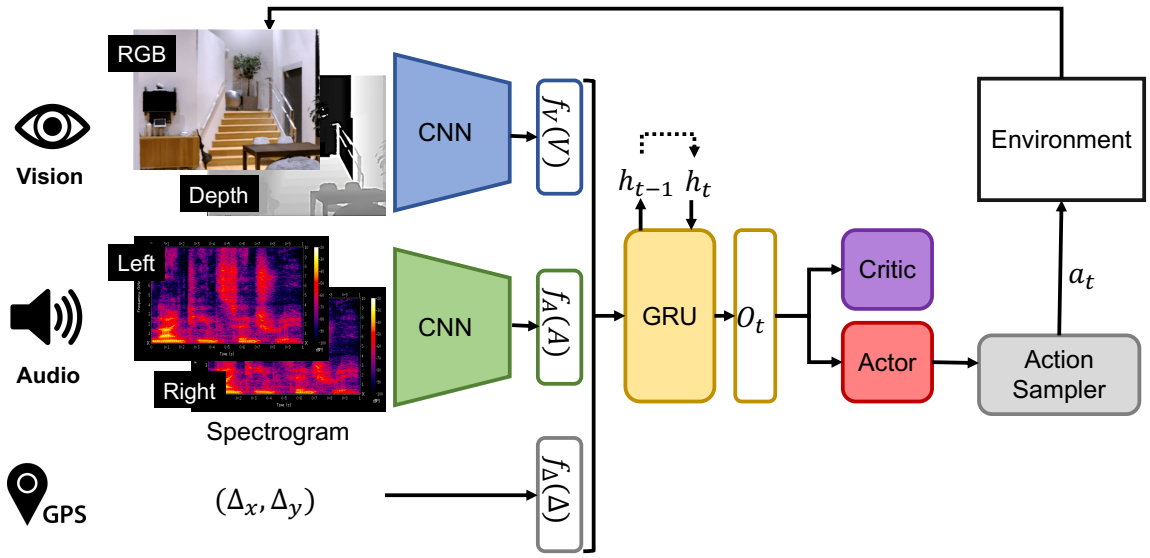


Figure 4.2: **Audio-visual** navigation network. Our model uses both acoustic and visual cues from the 3D environment for effective navigation of complex scenes.

The sensory inputs are transformed to a probability distribution over the action space by the policy network, as we describe next.

Network architecture. Next we define the parameterization of the agent's policy $\pi_\theta(a_t|o_t, h_{t-1})$, which selects action a_t given the current observation o_t and aggregated past states h_{t-1}, and the value function $V_\theta(o_t, h_{t-1})$, which scores how good the current state is. Here θ refers to all trainable weights of the network.

Our network architecture is inspired by current RL models in the visual navigation literature [247, 304, 58, 127]. We expand the traditional vision-only navigation model to enable acoustic perception for audio-visual navigation. As highlighted in Fig. 4.2, we transform A and V by corresponding CNNs $f_A(\cdot)$ and $f_V(\cdot)$. The CNNs have separate weights but the same architecture of conv 8×8, conv 4×4, conv 3×3 and a linear layer, with ReLU activations between each layer. The outputs of the CNNs are vectors $f_A(A)$ and $f_V(V)$ of length L_A and L_V, respectively. These are concatenated to the relative displacement vector Δ and transformed by a gated recurrent unit (GRU) [52]. The GRU operates on the current step's input as well as the

accumulated history of states h_{t-1}. The GRU updates the history to h_t and outputs the representation of the agent's state o_t. Finally, the value of the state $V_\theta(o_t, h_{t-1})$ and the policy distribution $\pi_\theta(a_t|o_t, h_{t-1})$ are estimated using the critic and actor heads of the model. Both are linear layers.

Training. We train the network with Proximal Policy Optimization (PPO) [254]. The agent is rewarded for reaching the goal quickly. Specifically, it receives a reward of $+10$ for executing *Stop* at the goal location, a negative reward of -0.01 per time step, $+1$ for reducing the geodesic distance to the goal, and the equivalent penalty for increasing it. We add an entropy maximization term to the cumulative reward optimization, for better action space exploration [112, 254].

Synergy of audio for navigation. Because our agent can both hear and see, it has the potential to not only better localize the target (which emits sound), but also better plan its movements in the environment (whose major structures, walls, furniture, etc. all affect how the sound is perceived). See Figure 3.1. The optimal policy would trace a path $\mathcal{P}^*$ corresponding to monotonically decreasing geodesic distance to the goal. Notably, the displacement Δ does not specify the optimal policy: moving along $\mathcal{P}^*$ decreases the geodesic distance but may decrease or increase the Euclidean distance to the goal at each time step. For example, if the goal is behind the sofa, the agent must move around the sofa to reach it. Importantly, the audio stream A has complementary and potentially stronger information than Δ in this regard. Not only does the intensity of the audio source reflect the Euclidean distance to the target, but also the geometry of the room captured in the acoustics reveals geodesic distances. As we show in results, the visual and aural inputs are synergistic; neither fares as well on its own.

Implementation details. The lengths of audio, visual, point vector, and final state, *i.e.*, , L_A, L_V, L_Δ, and L_S are 512, 512, 2, and 1026, respectively. We use

Table 4.1: Summary of SoundSpaces dataset properties

Dataset	# Scenes	Resolution	Sampling Rate	Avg. # Node	Avg. Area	# Training Episodes	# Test Episodes
Replica	18	0.5m	44100Hz	97	47.24 m^2	0.1M	1000
Matterport3D	85	1m	16000Hz	243	517.34 m^2	2M	1000

a single bidirectional GRU with input size 512, hidden size 512, and we use one recurrent layer. We optimize the model using Adam [143] with PyTorch defaults for coefficients for momentum and a learning rate of $2.5e-4$. We discount rewards with a decay of 0.99. We train the network for $30M$ agent steps on Replica and $60M$ on Matterport3D, which amounts to 105 and 210 GPU hours respectively.

4.1.2 Experiments

Our main objectives are to show:

1. Tackling navigation with both sight and sound (*i.e.*, , the proposed AudioPointGoal) leads to better navigation and faster learning. This demonstrates that audio has complementary information beyond merely goal coordinates that facilitates navigation.

2. Listening for an audio target in a 3D environment serves as a viable alternative to GPS-based cues. Not only does the proposed AudioGoal agent navigate better than the PointGoal agent, it does so without PointGoal's assumption of perfect odometry and even with noisy audio sensors. The AudioGoal task has the important advantage of realism: the agent autonomously senses the target in AudioGoal, whereas the target is directly given to the agent via Δ in PointGoal—a rare scenario in real applications.

3. Audio-visual navigation can generalize to both new environments and new sound sources. In particular, audio-visual agents can navigate better with audio even when the sound sources are unfamiliar.

Datasets. Table 4.1 summarizes SoundSpaces, which includes audio renderings for the Replica and Matterport3D datasets. Each episode consists of a tuple: ⟨scene, agent start location, agent start rotation, goal location, audio waveform⟩. We generate episodes by choosing a scene and a random start and goal location. To eliminate easier episodes, we prune those that are either too short (geodesic distance less than 4) or can be completed by moving mostly in a straight line (ratio of geodesic to Euclidean distance less than 1.1). We ensure that at the onset of each episode the agent can hear the sound, since in some large environments the audio might be inaudible when the agent is very far from the sound source.

Sound sources. Recall that the RIRs can be convolved with an arbitrary input waveform, which allows us to vary the sounds across episodes. We use 102 copyright-free natural sounds of telephones, music, fans, and others (`http://www.freesound.org`). Unless otherwise specified, the sound source is the telephone ringing. We stress that in all experiments, the environment (scene) at test time is unmapped and has never been seen previously in training. It is valid for sounds heard in training to also be heard at test time, e.g., a phone ringing in multiple environments will sound different depending on both the 3D space and the goal and agent positions. Experiments for objective 3 examine the impact of varied train/test sounds.

Metrics. We use the success rate normalized by inverse path length (SPL), the standard metric for navigation [8]. We consider an episode successful only if the agent reaches the goal *and* executes the *Stop* action.

Baselines. We consider three non-learning baselines adapted from previous work [247, 48]: RANDOM chooses an action randomly among {*MoveForward*, *TurnLeft*, *TurnRight*}. FORWARD always calls *MoveForward* and if it hits an obstacle, it calls *TurnRight* then resumes going forward and repeats. GOAL FOLLOWER always first orients itself towards the goal and then calls *MoveForward*. All three issue the *Stop* action upon reaching the goal.

Table 4.2: Adding sound to sight and GPS sensing improves navigation performance significantly. Values are success rate normalized by path length (SPL); higher is better.

		Replica		Matterport3D	
		PointGoal	AudioPointGoal	PointGoal	AudioPointGoal
Baselines	RANDOM	0.044	0.044	0.021	0.021
	FORWARD	0.063	0.063	0.025	0.025
	GOAL FOLLOWER	0.124	0.124	0.197	0.197
Varying visual sensor	Blind	0.480	**0.681**	0.426	**0.473**
	RGB	0.521	**0.632**	0.466	**0.521**
	Depth	0.601	**0.709**	0.541	**0.581**

1: Does audio help navigation? First we evaluate the impact of adding audio sensing to visual navigation by comparing PointGoal and AudioPointGoal agents. Table 4.2 compares the navigation performance (in SPL) for both agents and the baselines on the test environments. We consider three visual sensing capabilities: no visual input (Blind), raw RGB images, or depth images. (We found RGB+D was no better than depth alone.)

Audio improves accuracy significantly, showing the clear value in multi-modal perception for navigation. Both learned agents do better with stronger visual inputs (depth being the strongest), though the margin between RGB and depth is a bit smaller for AudioPointGoal. This is interesting because it suggests that audio-visual learning captures geometric structure (like depth) from the raw RGB images more easily than a model equipped with vision alone. As expected, the simple baselines perform poorly because they do not utilize any sensory inputs (and hence perform the same on both tasks).

To see how audio influences navigation behavior, Fig. 4.3 shows example trajectories.

2: Can audio supplant GPS for an audio target? Next we explore the extent to which audio supplies the spatial cues available from GPS sensing during (audio-) visual navigation. This test requires comparing PointGoal to AudioGoal. Recall that

Table 4.3: Navigation performance (SPL) when generalizing to unheard sounds. Higher is better. Results are averaged over 7 test runs; all standard deviations are ≤ 0.01.

Dataset			*Same sound*		*Varied heard sounds*		*Varied unheard sounds*	
		PG	*AG*	*APG*	*AG*	*APG*	*AG*	*APG*
Replica	Blind	0.480	0.673	0.681	0.449	0.633	0.277	0.649
	RGB	0.521	0.626	0.632	0.624	0.606	0.339	0.562
	Depth	0.601	0.756	0.709	0.645	0.724	0.454	0.707
Matterport3D	Blind	0.426	0.438	0.473	0.352	0.500	0.278	0.497
	RGB	0.466	0.479	0.521	0.422	0.480	0.314	0.448
	Depth	0.541	0.552	0.581	0.448	0.570	0.338	0.538

unlike (Audio)PointGoal, AudioGoal receives *no* displacement vector pointing to the goal; it can only hear and see.

Fig. 4.4a reports the navigation accuracy as a function of GPS quality. The leftmost point uses perfect GPS that tells the PointGoal agents (but not the AudioGoal agent) the exact direction of the goal; for subsequent points, Gaussian noise of increasing variance is added, up to $\sigma = 1.5$m. All agents use depth. While AudioGoal's accuracy is by definition independent of GPS failures, the others suffer noticeably.[‡]. This may be why AG is better than PG and APG on Replica. Furthermore, AudioPointGoal (APG) degrades much more gracefully than PointGoal (PG) in the face of GPS noise. This is evidence that *the audio signal gives similar or even better spatial cues than the PointGoal displacements*—which are likely overly optimistic given the unreliability of GPS in practice and especially indoors. T-SNE [289] visualizations (Fig. 4.4b) reinforce this finding: our learned audio features for AudioGoal naturally encode the distance and angle to the goal. Note that these findings stand even with microphone noise: with 40dB SNR (bad microphone), SPL only drops marginally from 0.756 to 0.753 and from 0.552 to 0.550 on Replica and Matterport, respectively.

Next we explore whether our AudioGoal agent learned *more* than a pointer

[‡]Replica has more multi-room trajectories, where audio gives clear cues of room entrances/exits (vs. open floor plans in Matterport)

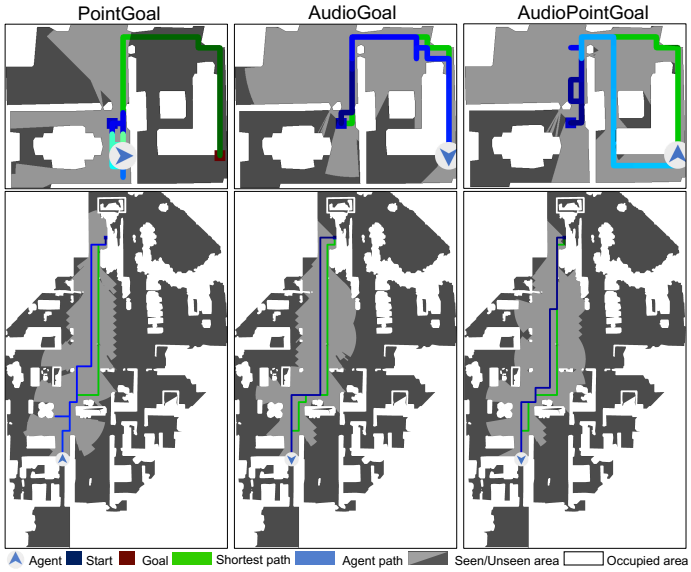
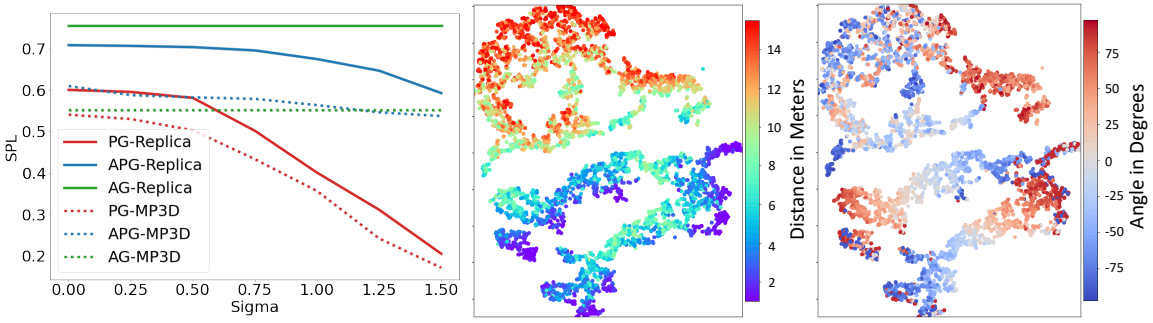


Figure 4.3: **Navigation trajectories on top-down maps.** Agent path color fades from dark blue to light blue as time goes by. Green path indicates the shortest geodesic path. **Top:** Replica - The PointGoal agent bumps into the wall several times trying to move towards the target, unable to figure out the target is actually located in another room. In contrast, the AudioGoal and AudioPointGoal agents better sense the target: the sound travels through the door and the agent leaves the starting room immediately. **Bottom:** Matterport - the AudioGoal agent best avoids backtracking to efficiently reach the target in a large multi-room home.



(a) From perfect to noisy GPS (b) t-SNE of AudioGoal audio feature

Figure 4.4: **Audio as a learned spatial sensor.** (a) Navigation accuracy with increasing GPS noise. Unlike existing PointGoal agents, our AudioGoal agent does not rely on GPS, and hence is immune to GPS noise. (b) t-SNE projection of audio features, color coded to reveal their correlation with the goal location (left) and direction (right), *i.e.*, , source is far (red) or near (violet), and to the left (blue) or right (red) of the agent.

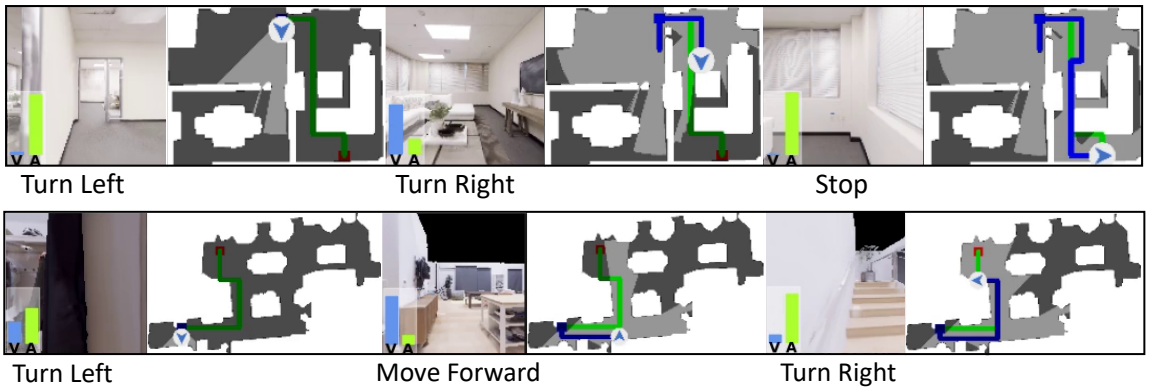


Figure 4.5: **Impact of each modality on action selection** for two AudioGoal episodes. We show one episode per row, and three sampled timesteps each. See Fig. 4.3 for legend. Blue and green bars display the importance of vision and audio, respectively. **Top:** Initially, the agent relies on audio to tell that the goal is on its left and decides to turn left. Later, it uses vision to recognize obstacles in front of it and decides to turn right. Finally, the agent decides to stop because the sound intensity has peaked. **Bottom:** Initially, the agent decides to turn left, following the audio source. Then the agent uses vision to identify the free space and decides to move forward. Later, the agent relies more on audio to decide to turn right as it hears the target from the right.

to the goal based on the sound intensity. We run a variant of our model in which the audio input consists of only the intensity of the left and right waveforms; the audio CNN is removed, and the rest of the network in Fig 4.2 remains the same. This simplified audio input allows the agent to readily learn to follow the intensity gradient. The performance of the AudioGoal-Depth agent drops to an SPL of 0.291 and 0.014 showing that our model (SPL of 0.756 and 0.552 in Fig 4.4a) does indeed learn additional environment information from the full spectrograms to navigate more accurately.

We expect that the audio and visual input vary in their relative impact on the agent's decision making at any given time point, based on the environment context and goal placement. To compute their impact, we ablate each modality in turn by replacing it with its average training sample value, and compare the resulting action probability under our model to that of the action chosen with both modalities. We calculate the importance of each input modality using the absolute difference

of logarithmic action probability, normalized by the sum of the two ablations. The greater the change in the selected action, the more impact that modality had on the learned agent's actual choice. Fig. 4.5 shows examples of the AV impact scores alongside the egocentric view of the agent at different stages in the trajectory. We see the agent draws dynamically on either or both modalities to inform its motions in the environment.

3: What is the effect of different sound sources? Next, we analyze the impact of the sound source. First, we explore generalization to novel sounds. We divide the 102 sound clips into 73/11/18 splits for train/val/test, respectively. We train for AudioGoal (AG) and AudioPointGoal (APG), then validate and test on disjoint val and test sounds. In all cases, the test environments are unseen.

Table 4.3 shows the results. As we move left to right in the table, the sound generalization task gets harder: from a single heard sound, to variable heard sounds, to variable unheard sounds. Note, the non-learning baselines are unaffected by changes to the audio and hence are omitted here. Our APG agents almost always outperform the PointGoal agent, even for unheard test sounds, strengthening the conclusions from Table 4.2. APG performs fairly similarly on heard and unheard sounds, showing it has learned to balance all three modalities. On the other hand, AG's accuracy declines with varied heard sounds and unheard sounds. While it makes sense that the task of following an unfamiliar sound is harder, we also expect that larger training repositories of more sounds will resolve much of this decline.

4.2 Learning to Set Waypoints for Audio-Visual Navigation

In the previous section, I introduced the audio-visual navigation benchmark, where an embodied agents navigate in an unknown environment with audio-visual inputs. What role should audio-visual inputs play in learning to navigate? There are two existing strategies. The navigation policy previously introduced in Sec. 4.1 learns to generate step-by-step actions (TurnRight, MoveForward, etc.) based on both modalities [35]. 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 [83]. 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 step (0.5 to 1 m) [35] or globally for the final goal location [83].

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 to set audio-visual waypoints*: the agent dynamically sets intermediate goal locations based on its audio-visual observations and partial map—and does so in an end-to-end manner with learning the navigation task. 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 4.6.

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

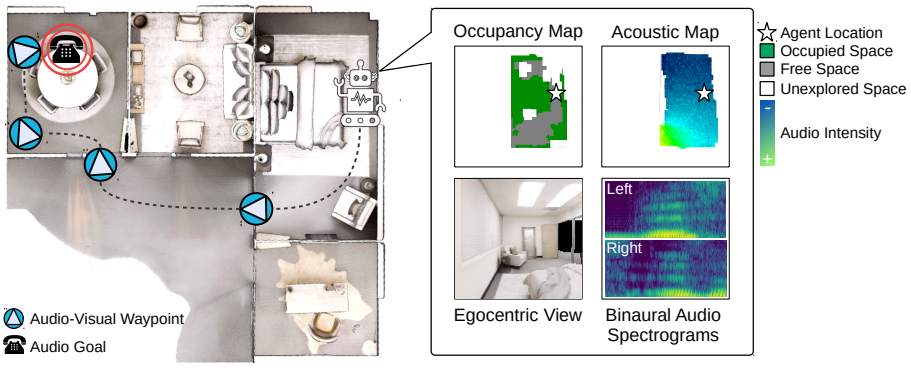


Figure 4.6: Waypoints for audio-visual navigation: Given egocentric audio-visual sensor inputs (depth and binaural sound), the proposed agent builds up both geometric and acoustic maps (top right) as it moves in the unmapped environment. The agent learns encodings for the multi-modal inputs together 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 may hear the phone ringing, identify that it is in another room, and decide to first exit the bedroom. It may then narrow down the phone location to the dining room, decide to enter it, and subsequently find it. Whereas existing hierarchical navigation methods rely on heuristics to determine subgoals, our model learns a policy to set waypoints jointly with the navigation task.

only visual input are limited to mapped locations or clear line-of-sight paths.

To realize our idea, our first contribution is a novel deep 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. Hierarchical policies for 3D navigation are not new, e.g., [32, 262, 15, 26]. However, whereas existing visual navigation methods employ heuristics to define subgoals, 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 visual navigation subgoals in general, not specific to audio goals. 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 complex 3D environments of Replica and Matterport3D using the SoundSpaces platform. It outperforms the AudioGoal navigation policy introduced in Sec. 4.1.1 by a substantial margin (8 to 49 points in SPL on heard sounds), and generalizes much better to the challenging cases of unheard sounds, noisy audio, and distractor sounds. Our results show learning to set waypoints in an end-to-end fashion outperforms current subgoal approaches, while the proposed acoustic memory helps the agent set goals more intelligently.

4.2.1 Approach

We consider the previously introduced AudioGoal navigation task (Sec. 4.1). 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 (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, 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. 4.7). 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.

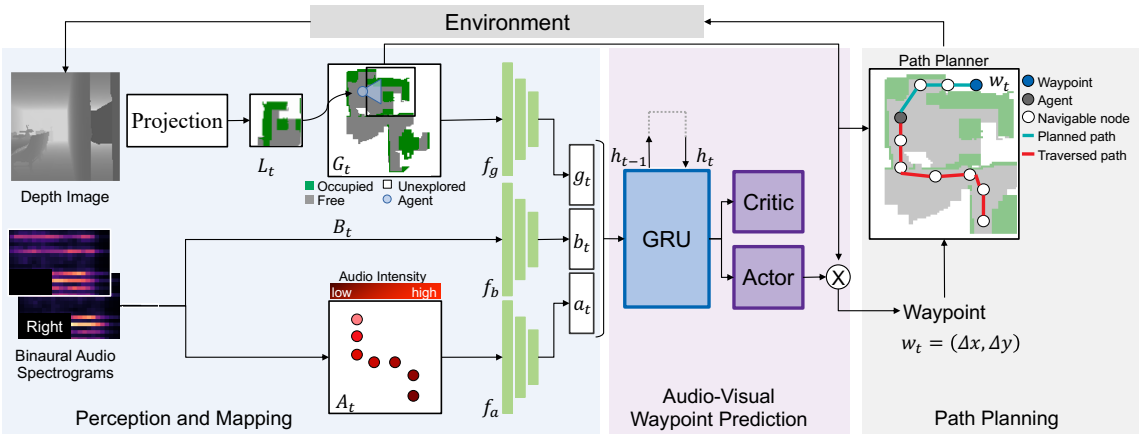


Figure 4.7: Model architecture. Our audio-visual navigation model uses the egocentric stream of depth images and binaural audio (B_t) to learn geometric (G_t) and acoustic (A_t) maps for the 3D environment. The multi-modal cues and partial maps (left) inform the RL policy's prediction of intermediate 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.

4.2.1.1 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 [32, 48]. 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 it has a 3D point that 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. 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 4.7.

Acoustic perception At each time step the agent receives binaural sound B_t rep-

resented 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 4.7). 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 4.7. Note that a map of audio intensities reveals both distance and directional information about the sound source, since the gradient in audio intensity helps indicate the goal direction. 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.

4.2.1.2 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 4.6, 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 convolutional neural network (CNN) in the end to extract features.

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) [52]. See Figure 4.7.

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.[§] off the RIR grid. 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. 4.2.1.3).

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 AV navigation methods, our waypoints guide the agent at a variable granularity, as opposed to fixing its actions to myopic next steps [35] or a final goal prediction [83]. Unlike existing visual subgoal approaches, which rely on frontier-based heuristics or points along the shortest path [32, 262, 15, 26], our waypoints are inferred in tight integration with the navigation task. Our results demonstrate the advantages.

4.2.1.3 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. Unexplored areas in the map are considered free space during planning [32]. 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 environment graphs have nodes only where the SoundSpaces audio RIRs are available, and hence both actions and candidate waypoints are discrete sets. Note: this disallows testing noisy actuation for any method because audio observations are not available at positions

The planning loop breaks under three conditions: 1) the agent reaches the waypoint, 2) the planner could not find a path to the waypoint (in this case the agent executes a random action before breaking the loop), or 3) the agent reaches a planning step limit. The planning step limit is set to mitigate 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.

4.2.1.4 Reward and Training

Following typical navigation rewards [247, 35], 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.

All learnable modules are jointly trained and updated every 150 waypoint prediction steps with Proximal Policy Optimization (PPO) [254]. The PPO loss consists of a value network loss, policy network loss, and an entropy loss to encourage exploration.

4.2.2 Experiments

Environments Following the protocol in Sec. 4.1, we use both Replica and Matterport environments, and train/test on disjoint environments to evaluate generalization.

Metrics We evaluate the following navigation metrics: 1) success rate (SR), the frac-

tion of successful episodes, i.e., episodes in which the agent stops exactly at the audio goal location on the grid; 2) success weighted by path length (SPL), the standard metric [8] that weighs successes by their adherence to the shortest path; 3) success weighted by number of actions (SNA), which penalizes rotation in place actions, which do not lead to path changes.

Existing methods and baselines We compare the following methods:

- **Random**: an agent that randomly selects each action and signals *Stop* when it reaches the goal.
- **Direction Follower**: a hierarchical model that sets intermediate goals K meters away in the audio's predicted direction of arrival (DoA), and repeats. K is estimated through a hyperparameter search on the validation split, which yields $K = 2$ in Replica and $K = 4$ in Matterport. We train a separate classifier based on audio input to predict when this agent should stop.
- **Frontier Waypoints**: a hierarchical model that 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., [26, 262, 32], making this a broadly representative baseline for standard practice.
- **Supervised Waypoints**: a hierarchical model that uses the RGB frame and audio spectrogram to predict waypoints in its field of view (FoV) with supervised (non-end-to-end) learning. This model is inspired by Bansal et al. [15], which learns to predict waypoints in a supervised fashion.
- **AV-Nav (see Sec. 4.1.1)**: a state-of-the-art end-to-end AudioGoal RL agent that selects actions using audio-visual observations. It lacks any geometric or acoustic maps.

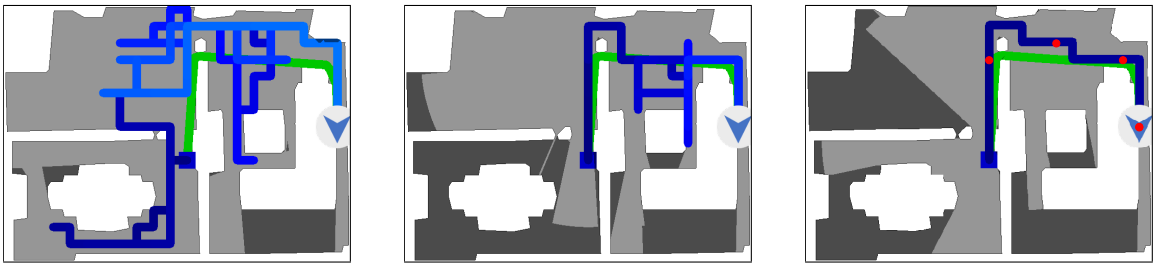
Table 4.4: AudioGoal navigation results. Our audio-visual waypoints navigation model (AV-WaN) reaches the goal faster (higher SPL) and it is more efficient (higher SNA) compared to the state-of-the-art. SPL, SR, SNA are shown as percentages. For all metrics, higher is better. (H) denotes a hierarchical model.

	Replica						*Matterport3D*					
	Heard			*Unheard*			*Heard*			*Unheard*		
Model	SPL	SR	SNA	SPL	SR	SNA	SPL	SR	SNA	SPL	SR	SNA
Random Agent	4.9	18.5	1.8	4.9	18.5	1.8	2.1	9.1	0.8	2.1	9.1	0.8
Direction Follower (H)	54.7	72.0	41.1	11.1	17.2	8.4	32.3	41.2	23.8	13.9	18.0	10.7
Frontier Waypoints (H)	44.0	63.9	35.2	6.5	14.8	5.1	30.6	42.8	22.2	10.9	16.4	8.1
Supervised Waypoints (H)	59.1	88.1	48.5	14.1	43.1	10.1	21.0	36.2	16.2	4.1	8.8	2.9
Gan et al.	57.6	83.1	47.9	7.5	15.7	5.7	22.8	37.9	17.1	5.0	10.2	3.6
AV-Nav	78.2	94.5	52.7	34.7	50.9	16.7	55.1	71.3	32.6	25.9	40.1	12.8
AV-WaN (**Ours**) (H)	**86.6**	**98.7**	**70.7**	**34.7**	**52.8**	**27.1**	**72.3**	**93.6**	**54.8**	**40.9**	**56.7**	**30.6**

- **Gan et al. [83]**: a state-of-the-art AudioGoal 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 Replica and Matterport, 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 [35, 83], and 2) *unheard sounds*—train and test with disjoint sounds, following [35]. In both cases, the test environment is always unseen, hence both settings require generalization.

Table 4.4 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 other methods—with 8.4% and 29% SPL gains on Replica compared to AV-Nav and Gan et al., and 17.2% and 49.5% gains on Matterport. This result shows the advantage of our dynamic audio-visual waypoints and structured acoustic map, compared to the myopic action selection in AV-Nav and the final-goal



(a) Gan et al. [83] (b) Chen et al. [35] (c) AV-WaN

Figure 4.8: Navigation trajectories on top-down maps vs. all existing AudioGoal methods. 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. The agent's inputs are egocentric views (Fig. 1); figures show the top-down view for ease of viewing the full trajectories.

prediction in Gan et al. We find that the RL model of AV-Nav fails when it oscillates around an obstacle. Meanwhile, predicting the final audio goal location, as done by Gan et al., is prone to errors and leads the agent to backtrack or change course often to redirect itself towards the goal. This result emphasizes the difficulty of the audio-visual navigation task itself; simply reducing the task to PointGoal after predicting the goal location from audio (as done in Gan et al.) is much less effective than the proposed model. See Figure 4.8.

Our method also surpasses all three other hierarchical models. This highlights our advantage of directly *learning* to set waypoints, versus the heuristics used in current hierarchical visual navigation models. Even the Supervised Waypoints model does not generalize as well to unseen environments as AV-WaN. We expect this is due to the narrow definition of the optimal waypoint posed by supervision compared to our model, which learns from its own experience what is the best waypoint for the given navigation task in an end-to-end fashion.

In the unheard sounds setting covering 102 sounds (Table 4.4, right), our method again strongly outperforms all existing methods on both datasets and in almost every metric. The only exception is our 2.8% lower SPL vs. AV-Nav on

Table 4.5: Ablation study for AV-WaN. Results are averaged over 5 test runs; all standard deviations are ≤ 0.5.

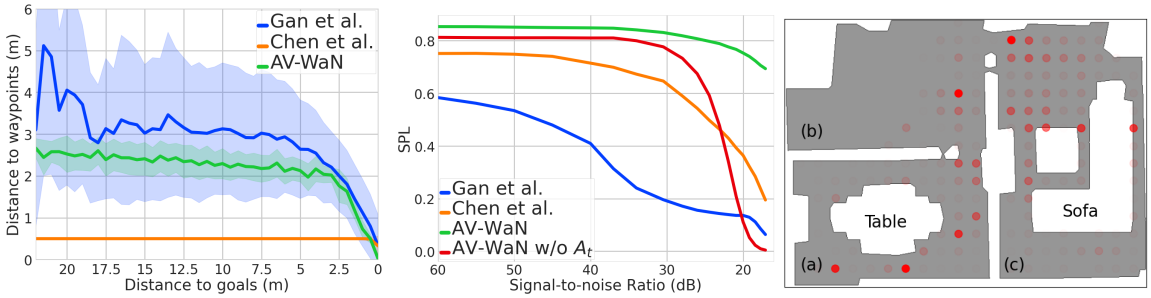
	Replica						*Matterport3D*					
	Heard			*Unheard*			*Heard*			*Unheard*		
Model	SPL	SR	SNA	SPL	SR	SNA	SPL	SR	SNA	SPL	SR	SNA
AV-WaN w/o A_t and G_t	84.3	97.8	69.1	34.0	48.6	25.4	68.8	92.1	52.1	20.5	30.4	15.5
AV-WaN w/o G_t	85.1	97.5	69.0	27.0	45.6	20.3	70.2	94.0	52.4	25.4	45.0	19.2
AV-WaN w/o A_t	85.7	98.7	70.2	34.5	**63.3**	24.8	70.2	93.6	53.2	36.7	53.8	28.6
AV-WaN w/o waypoints	79.8	95.5	48.4	25.5	38.2	10.6	44.3	63.2	20.3	25.5	40.0	11.0
AV-WaN	**86.6**	**98.7**	**70.7**	**34.7**	52.8	**27.1**	**72.3**	**93.6**	**54.8**	**40.9**	**56.7**	**30.6**

Replica, though our model still surpasses AV-Nav in terms of SNA on that dataset, meaning we have better accuracy when normalizing for total action count. 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 4.5 shows ablations of the input modalities and the audio-visual waypoint component of our model.[¶] Removing both the geometric and acoustic maps causes a 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 [35] (see Table 4.4). This shows that our waypoint-based navigation framework itself is more effective than the simpler RL model [35], as well as the existing subgoal approaches. Removing just A_t also leads to a drop in performance, which demonstrates the importance of the proposed structured acoustic memory for efficient navigation. Both A_t and G_t are complementary and critical for our model to reach its best performance. Finally, we evaluate the impact of our idea of audio-visual waypoint prediction.

We replace the actor network in our model (see Fig. 4.7 middle) with a linear

[¶]When G_t is removed, we remove the masking operation to ensure no geometric information is used as input, but we keep the geometric map for the planner.



(a) Waypoint distance distribution

(b) Audio noise

(c) Waypoint placement distribution

Figure 4.9: Analysis of selected waypoints (a,c) and accuracy vs. microphone noise (b). See text.

layer that outputs the action distribution over the four primitive actions in $\mathcal{A}$. An action sampler directly samples an action from this distribution and executes it in the environment. In this case, there is no need for a planner. Our gains over that ablation confirm the value of the waypoints to our model, even when all other components are fixed.

Failure cases We next analyze the unsuccessful episodes for our model. We identify two repeating types of failures among these episodes. The first is where the audio goal is cornered among obstacles or lies right next to a wall. In this case, while AV-WaN reaches the goal quickly, it keeps oscillating around it and fails to pinpoint the location of the goal due to strong audio reflections from the obstacles around the goal or due to mapping errors. In the second case, we notice that sometimes the agent prematurely executes a stop action next to the audio goal. We expect that the differences in the audio intensity in the immediate neighborhood of the goal where the sound is the loudest are harder to detect, which may lead to this behavior.

Noisy audio and distractor sounds To understand the robustness of our model under noisy audio perception, we consider two sources of audio noise: environment noise and microphone noise. For environment noise, we add distractor sounds (e.g. human speaking, fan spinning) to interfere with the agent's audio perception. The agent is always tasked to find the telephone, and the distractor is an unheard sound

placed at a random location. At each time step, the agent receives the combined waveforms of two sounds and needs to pick up on the telephone signal and find its source location. We use the same episodes from the Heard experiment (Table 4.4) to train and evaluate the agent. With distractors, the best performing baseline, AV-Nav, obtains 71.7% and 53.3% test SPL on Replica and Matterport respectively, while our model achieves 83.1% and 70.9%. For microphone noise, we add increasing Gaussian noise to the received audio waveforms. Fig. 4.9b shows the results. AV-WaN is quite robust to audio noise, especially with A_t, while the existing AudioGoal methods suffer significantly. Hence our model's advantages persist in noisier settings common in the real world, and the acoustic memory is essential in this noisy setting.

Dynamic waypoint selection Fig. 4.9a plots the distribution of euclidean distances to waypoints as a function of the agent's geodesic distance to the goal. 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.

Placement of waypoints To examine how waypoints are selected based on surrounding geometry, Fig. 4.9c plots the distribution of waypoints on a top-down map for a test Replica 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 are the decision states that lie at critical junctions in the state spaces from which the agent can gather the most new information and transition to new, potentially unexplored regions [104]. The most frequent waypoints are usually 2-3m apart, close to the maximum distance the agent can choose.

4.3 Continuous Audio-Visual Navigation in SoundSpaces 2.0

In previous sections, I presented both the original audio-visual navigation task formulation and an efficient hierarchical policy to improve the performance. However, due to its reliance on SoundSpaces, audio-visual navigation thus far must assume the

Table 4.6: Continuous audio-visual navigation benchmark. DTG stands for distance to goal. We report the mean and standard deviation by training on 1 random seed, and evaluating on 3 random seeds.

Train	Test	Success (%)	SPL (%)	DTG (m)
SoundSpaces [35]	Continuous space	64.2 ± 0.8	27.5 ± 0.4	5.6 ± 0.2
SoundSpaces [35]	Continuous space & continuous sound	0.9 ± 0.2	0.3 ± 0.1	12.9 ± 0.1
SoundSpaces 2.0	Continuous space & continuous sound	64.7 ± 3.9	49.3 ± 3.0	5.9 ± 0.5

agent travels along the discrete grid. The navigation task is thus easier due to the lack of collisions and implied perfect localization.

Here we introduce the *continuous* audio-visual navigation task, enabled by SoundSpaces 2.0 simulation. In this task, the agent can choose to either move forward 0.15 m per step at a speed of 1m/s or turn left/right 10 degrees. However, these actions might fail or be partially executed due to collisions, while previously, the agent always teleports to the next location. If the agent issues a stop action within 1m radius of the goal, the episode is regarded as successful. Importantly, the agent not only moves in continuous space but also receives acoustically continuous audio signals (cf. Sec. 3.2.1.2). We use the high-speed rendering mode.

We generalize the existing audio-visual navigation policy AV-Nav (introduced in Sec. 4.1.1) to a distributed audio-visual navigation (DAV-Nav) agent equipped with DD-PPO [302] to speed up the training process. We train and test on the AudioGoal navigation dataset [35]. To ablate the simulation improvement as detailed in Sec. 3.2.1.1, for the SoundSpaces baseline, we train DAV-Nav on SoundSpaces' discrete setup (agent only moving between grid points) with data rendered from the enhanced simulation; the action space is either moving forward 1 m, turning left/right 90 degrees or issuing a stop action.

Table 4.6 shows the results using the standard metrics of success rate, success rate normalized by path length (SPL), and distance to goal. If only the space is continuous, the DAV-Nav agent trained on SoundSpaces has 64.2% success rate and 27.5% SPL on average compared to 64.7% success rate and 49.3% SPL of the agent

trained on SoundSpaces 2.0. This shows spatial continuity mostly harms the agent's efficiency rather than its success rate; the agent can still navigate to the source despite having more collisions. However, when the sound is acoustically continuous, the baseline's performance drops. This is likely because the agent relies on the direct-sound cue that is (inaccurately) always present in the audio, while in the continuous-sound rendering, direct sound is always mixed with the reverberation in the environment, making navigation more difficult. In comparison, the agent trained on SoundSpaces 2.0 achieves a much higher success rate and is much closer to the goal location on average. This shows it is essential to model both spatial and acoustic continuity for audio-visual navigation, which SoundSpaces 2.0 enables. Furthermore, recall that SoundSpaces 2.0 opens up any other 3D scene dataset for exploring audio-visual navigation, whereas previously only Replica or Matterport3D were applicable.

4.4 Sim2Real Transfer with Frequency-Adaptive Acoustic Field Prediction

In previous sections, I introduced multiple navigation policies that enable audio-visual navigation on both SoundSpaces 1.0 and SoundSpaces 2.0. However, the question of how to transfer the policy from the simulation to the real world remains unresolved. In this section, I introduce a frequency-adaptive method for sim2real transfer.

With the success of learning-based navigation systems in photorealistic simulation environments, some work explores transferring the learned policy to the real world by bridging the gap between the simulation and the real world [324, 10, 218, 135]. Recent work [92] does sim2real transfer for audio-visual navigation with data augmentation however without further investigating the acoustic gap. The sound differs from light in that it spans a wide range of frequencies, which is one of the main barriers to sim2real transfer. In this section, we perform a systematic evaluation of the acoustic gap and propose a solution to bridge that gap.

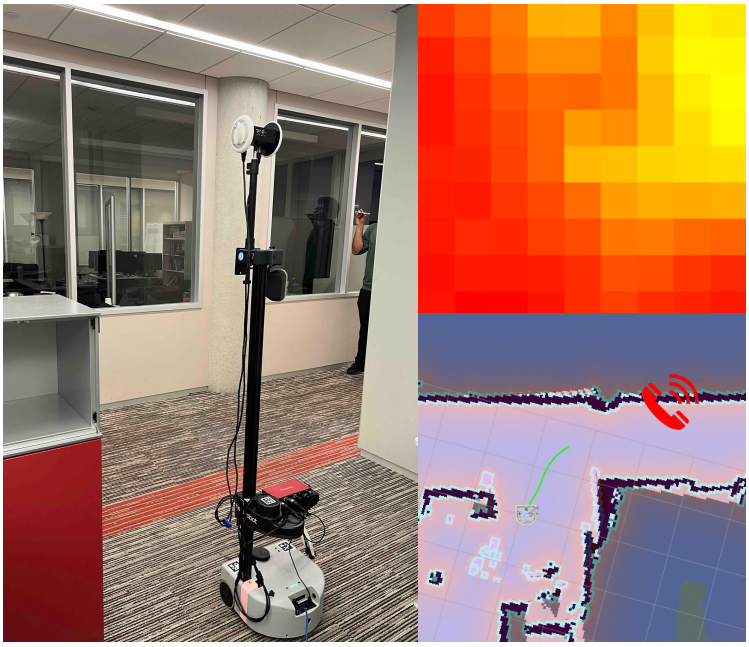


Figure 4.10: Our robot predicts an acoustic field with a frequency-adaptive model and navigates to locate the sound source.

State-of-the-art approaches in audio-visual navigation rely on reinforcement learning to train the navigation policy end-to-end [34, 36], which is not only hard to interpret but also impractical to generalize to the real world directly due to various sim2real gaps. Recent visual navigation work has shown success in sim2real transfer with hierarchical models [10, 229], which typically consist of a high-level path planner and low-level motion planner. This design helps abstract away some of the low-level physical discrepancies.

Inspired by such methods, we design a modular approach to ease the transfer from the simulation to the real world. To achieve this, we confront a key question: what is the proper high-level planning task that can survive sim2real transfer for audio-visual navigation? To this end, we propose a novel prediction task: *acoustic field prediction*—predicting the local sound pressure field around the agent. The gradient of this field reflects the direction of the sound. Measuring acoustic fields is expensive in the real world since it requires simultaneously capturing the sound

pressure of all points in the field due to the dynamic nature of sound. However, they are free to compute in simulation. We first build an audio-visual model as the acoustic field predictor (AFP) and curate a large-scale acoustic field dataset on SoundSpaces 2.0 (see Chapter 9). We show that this approach outperforms existing methods on the Continuous AudioGoal navigation benchmark.

After validating the proposed approach in simulation, we then investigate where acoustic discrepancy arises. It is known that ray-tracing-based acoustic simulation algorithms introduce more errors with lower frequencies due to wave effects [245]. Given this observation, we focus on evaluating how the sim2real error changes as a function of frequencies. We first collect real acoustic field data with the source sound being white noise, whose audio energy uniformly spans across all frequencies. We then train acoustic field prediction models that only take the sub-frequency band of the input audio and test it on the real white noise data. By computing the errors across multiple samples, we show that the errors do not strictly go down as the frequency goes higher, and using the best frequency band yields errors smaller than using all frequencies for the white noise sound.

However, simply taking the best frequency band does not work for all sounds since different sounds have different spectral distributions. To address this issue and make the model aware of the spectral difference, we propose a novel frequency-adaptive prediction strategy, that intelligently selects the best frequency sub-band based on measured errors as well as the received spectral distribution to predict the acoustic field. To validate this approach, we collect more acoustic field data with various sounds and show that the frequency-adaptive model leads to the lowest error on the real data compared to other strategies.

Lastly, we build a robot platform that equips the Hello Robot with a 3Dio binaural microphone and then deploy our trained policy on this robot. We show that our robot can successfully navigate to various sounds with our trained frequency-adaptive acoustic field prediction model. See Fig. 4.10.

In summary, we propose a novel acoustic field prediction approach that learns to navigate without interaction with the environment. This approach improves the SOTA methods on the challenging Continuous AudioGoal navigation benchmark. We perform a systematic evaluation of the sim2real and propose a frequency-adaptive strategy as the treatment for sim2real. We show this strategy works on both collected real data and our robot platform. To the best of our knowledge, this is the first work to investigate and propose a principal solution to the sim2real transfer problem for audio-visual navigation.

4.4.1 Approach

In this work, we use the SoundSpaces 2.0 platform introduced in Chapter 3 and target the continuous AudioGoal navigation benchmark introduced in Sec. 4.3.

4.4.1.1 A Modular Design for Sim2real Transfer

Transferring a navigation policy trained in simulation to the real world is not trivial due to many domain gaps between the simulation environment and the real world, which include the visual discrepancy, the physical dynamics discrepancy, the robot actuation discrepancy and—specifically in this task—the acoustic discrepancy.

We focus on investigating the acoustic discrepancy and to bridge other domain gaps (e.g., visuals and physics), we take a hierarchical approach that disentangles navigation into high-level path planning and low-level motion planning. This has a few benefits: 1) disentangling the policy makes it possible to utilize existing SLAM algorithms on the real robot to abstract away domain gaps other than the audio. 2) disentangling the policy makes the intermediate output more interpretable and easier to debug 3) specifically in this work, posing the high-level planning as a supervised prediction task makes it easier to measure the sim2real difference because we can evaluate the performance by collecting real measurements without repeatedly running robots.

The key challenge here is to formulate the proper waypoint prediction task

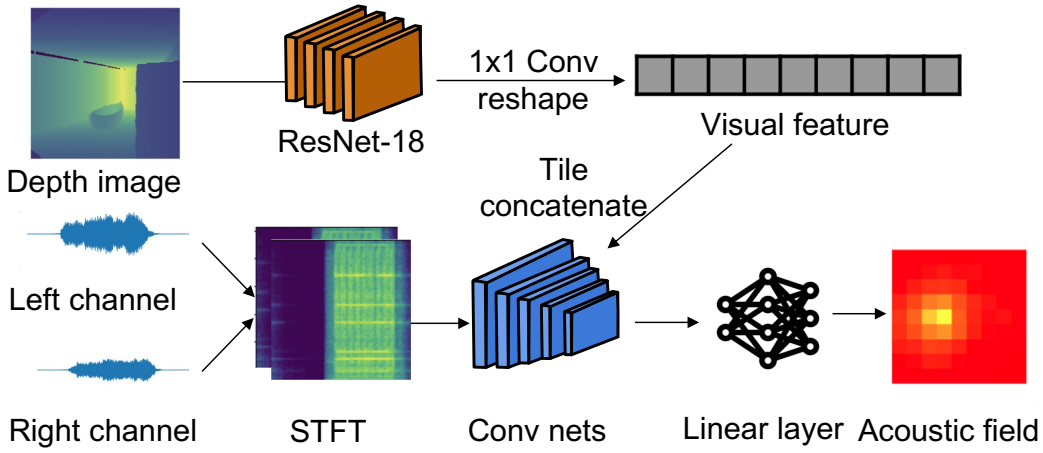


Figure 4.11: Acoustic field prediction model. The model first extracts audio and visual features, and then tiles and concatenates both features to predict the acoustic field.

that could survive the sim2real transfer. One existing approach [83] predicts the exact location of the audio goal directly, which is however an ill-posed problem since the environment geometry is unknown. For example, when the audio goal is in another room, the received audio reveals the direction to the door rather than the exact direction of the goal. We propose to predict the local acoustic field (sound pressure field) centered around the agent. Collecting acoustic field data for training is very expensive in the real world since it requires a complicated microphone array setup. However, rendering them in simulation is nearly free.

The hierarchical model alone however does not address the audio discrepancy directly. While SoundSpaces 2.0 produces realistic audio renderings, there is some unavoidable difference between simulation and the real world. It is known that ray-tracing-based algorithms yield worse performance with lower frequencies due to wave effects [245]. This implies the model needs to be aware of this spectral difference for sim2real. Thus we introduce a frequency-adaptive prediction strategy to help the model better transfer to the real world in Sec. 4.4.1.4.

4.4.1.2 Acoustic Field Prediction

To tackle the acoustic field prediction problem, we first present a model that uses both audio and visual observations (see Fig. 4.11). The motivation behind using the visual sensor is that the visual observation of the surrounding environment can be useful in inferring the geometry of the environment, which affects the acoustic field. For example, walls often act as the boundary of the acoustic field.

More specifically, the model takes in a depth image of 128×128 pixels. We use depth image instead of RGB image because it contains 3D information of the scene and tends to yield better performance [117, 24, 75]. The input audio to the model is a one-second segment of binaural audio, following prior work [34].

We use a pre-trained ResNet [115] to extract features from the input image and reshape it with a 1×1 Conv layer into a 1d vector of size 512. For the input binaural audio, we first process the waveforms with Short-Time Fourier Transform (STFT) to convert the time-domain signal into the frequency domain. We then use a 2D Conv net to encode the features and then tile and concatenate with the visual feature. Lastly, we feed the final output through one linear layer and reshape the prediction into the size of the target acoustic field $L \times L$.

4.4.1.3 Hierarchical Navigation

With this acoustic field prediction model, we then construct a hierarchical navigation pipeline (see Fig. 4.12) to perform audio-visual navigation, which executes the following steps: 1) sampling a long-term goal; 2) navigating to the long-term goal; and 3) making the stopping decision. Different from the hierarchical policy AV-WaN I introduced earlier in 4.2, the long-term goal in this pipeline is produced by the acoustic field predictor model while AV-WaN produces the long-term waypoint with a reinforcement learning policy.

Sampling a Long-Term Goal At each time step, the agent predicts the acoustic field based on audio-visual inputs and then identifies the maximum value of the field.

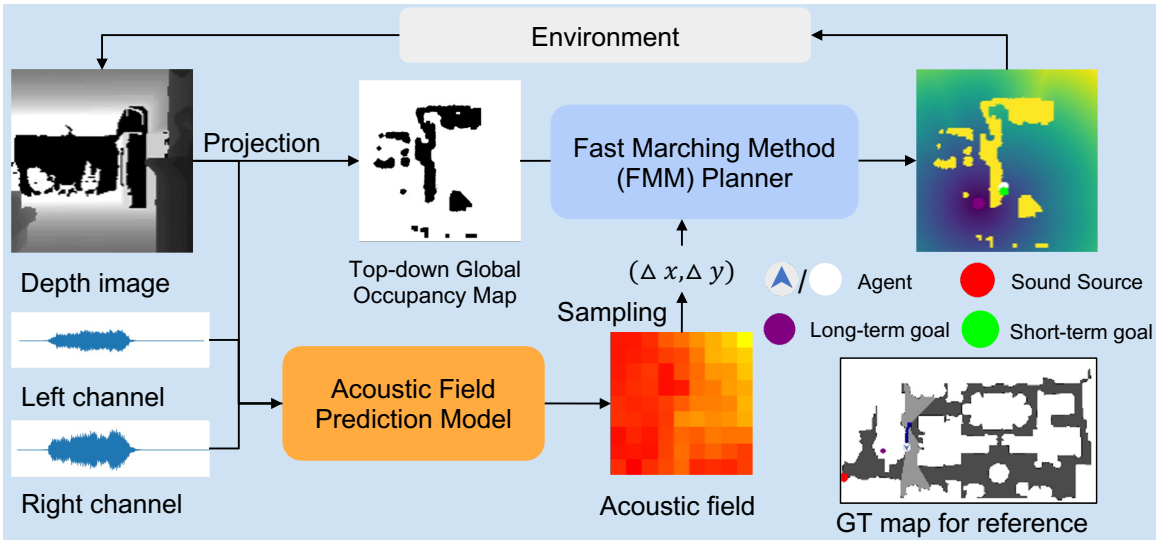


Figure 4.12: Navigation pipeline. The model first predicts the acoustic field, samples the peak as the long-term goal, and navigates toward the goal with a path planner.

We set the peak location as the long-term goal either when there is no existing long-term goal or the new peak value surpasses the value of the existing long-term goal since as the agent gets closer to the goal, the sound usually gets louder.

Navigating to the Long-Term Goal After sampling the long-term goal, for path planning, we use the Fast Marching Method (FMM) [256] to determine the best route to the goal in simulation. FMM takes the occupancy map, the agent's current position, and the long-term goal as inputs. The occupancy map is computed by calculating the point cloud observed at each timestep using the depth camera. Next, FMM calculates the distance between each navigable point in the map to the long-term goal. The algorithm then selects the adjacent point on the map with the lowest value as its short-term goal and the agent then moves towards that point. When the long-term goal is sampled at a non-navigable location, we use breadth-first search (BFS) to find the closest available point to navigate to.

Stopping Criteria The stopping condition is evaluated each time after the agent reaches a long-term goal or the closest navigable point to the long-term goal. When the agent samples a new long-term goal, if the peak value of the predicted acoustic field is at the center of the field, the agent issues the stop action.

4.4.1.4 Frequency-adaptive Prediction

Existing audio-visual navigation models use all frequencies in the input audio. However, as discussed earlier, the acoustic gap with the real world is a function of frequencies. Thus models trained with all frequencies assuming them equally reliable would have lower performance when deployed on a real robot.

Given this observation, we first systematically examine how the gap changes as a function of the frequency. The idea is simple: with a given frequency band $[F_1, F_2]$, we first train an acoustic field prediction (AFP) model in simulation using only that band, then test it on real-world data of the same sound and same band, and calculate the prediction error. We equally divide all frequencies into 5 subbands and we show the distribution of errors over the frequency bands in Fig. 4.13. As expected, the lower frequencies tend to yield larger prediction errors. However, the error does not monotonically decrease as frequency increases. We also trained a model that uses all frequencies, which has a distance error of $0.86m$, underperforming the best frequency band.

With this measurement, the most intuitive idea would be just to take the frequency band that has the least sim2real error and train a model with that band. However, this will not work for real-world scenarios where some sounds span across many frequency bands while others only occupy a very narrow range of frequencies. To take that into account, we propose a frequency-adaptive prediction strategy that uses the best frequency band based on both the measured error and the energy distribution of the received audio.

Assume we divide all frequencies linearly into N bands. Given a received audio

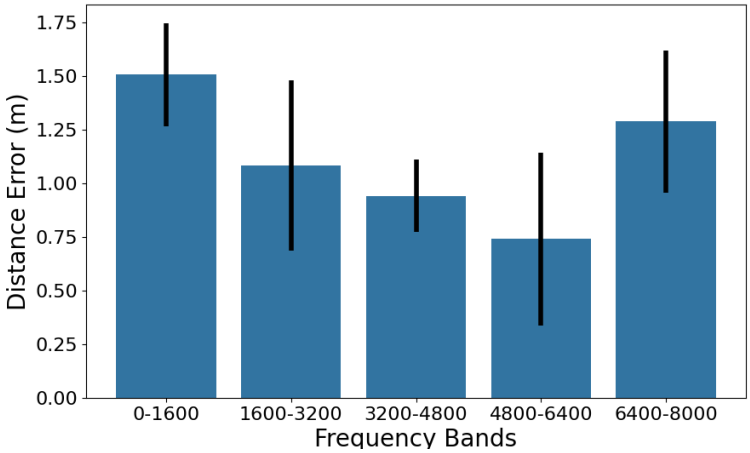


Figure 4.13: Sim2real error as a function of frequencies. We report the mean and standard deviation of distance errors between the predicted and the ground truth peak locations.

A_r, we first convert it into the frequency domain and divide it into these N bands. Based on the measured errors, we have a weighting function that assigns weights to these bands based on their sim2real errors:

$$p(i) = (\frac{1}{e_i})^{\alpha}, i \in [1, ..., N], \tag{4.1}$$

where e_i is the error in Fig. 4.13. For each subband i of the input, we then compute another weight based on the energy of the band normalized with respect to the highest energy:

$$q(i) = (\frac{r_i}{r_m})^{\beta}, i \in [1, ..., N], \tag{4.2}$$

where r_i is the energy of that band and $r_m = \max_i r_i$. We basically assign higher weights to frequency bands that have more energy. Lastly, we take the product of these weights:

$$w(i) = p(i) \times q(i), i \in [1, ..., N] \tag{4.3}$$

We take band i with the highest $w(i)$ to produce the final prediction. Both α and β are hyper-parameters, and we perform a grid search to find the best values on validation.

Intuitively, what the weighting function does in eq. (4.3) is: if the input sound has a fairly equal distribution of energies over all subbands, it will take the best band

from $p(i)$ that has the lowest sim2real error. If the input sound has a very skewed energy distribution, it will prioritize taking the band where the audio has the most energy. In this way, we factor into both the measured difference and the spectral distribution of individual sounds.

4.4.1.5 Implementation and Training Details

For the size of the acoustic field, we set L to 9 with a grid resolution of $0.5m$, i.e., $4.5m \times 4.5m$ centered around the agent based on our ablations. α and β are set to 5 and 0.8 respectively based on the validation performance.

We train the predictor with Mean Squared Error (MSE) loss till convergence. For optimization, we use the Adam optimizer [143] with a learning rate set of 0.001.

4.4.2 Data Curation

Due to the expense of measuring real acoustic field data, we choose to utilize simulation to collect large-scale training data. We also collect real data for measuring the sim2real gap and validating our frequency-adaptive prediction model.

4.4.2.1 SoundSpaces Acoustic Field Dataset

SoundSpaces 2.0 (see Chapter 3) supports computing the impulse response $I(s, r)$ between the source location s and the receiver location r as a function of the 3D environment but does not have direct API support for rendering the acoustic field. To compute the field, given a (s, r) pair, we first sample a grid centered at the receiver location of size $L \times L$, for each grid point p, we compute $I(s, p)$ which results in L^2 number of RIRs per receiver location. However, these RIRs are represented in the form of waveforms instead of single numbers. To best represent the sound pressure at each single point, we take the maximum amplitude of the waveform.

For sampling the S/R locations and environments, we utilize the existing dataset for audio-visual navigation, which provides configurations of the environ-

ment and source/receiver locations. This dataset uses scenes from the Matterport3D dataset [29], which contain scans of real-world environments such as apartments, offices, and even churches. We sample 500 episodes per environment for the 57 training environments in the navigation dataset. We also perform a similar operation to curate the validation and test set. In total, we collect 1.1M/52K/52K samples for train/val/test. Along with these acoustic fields, we also render the RGB-D images at the corresponding locations. See examples in Fig. 4.15.

4.4.2.2 Real Measurements Collection

To measure the sim2real error, we collect real audio measurements to evaluate the trained model's performance. For that, we use a 3Dio microphone to capture the binaural audio with a smartphone serving as the speaker output. We aligned the real-world parameters closely with those in our simulator, such as the height of the speaker and receiver. Since the simulator employs a mono receiver, the two-channel audio data we gathered is transformed into mono format by averaging the amplitude values across both channels. This process was repeated for ten distinct speaker positions (8 different directions w.r.t the agent and two data points for when the speaker is near the agent). We also downsample the acoustic field resolution from 9×9 to 3×3 so that we could collect more data in more environments.

For the source of the sounds, we use two types of sounds: white noise and normal sounds. To compute the sim2real errors in Fig. 4.13, it is important for the sound to have uniform distribution across all frequencies, and we use white noise for that. For evaluating the final frequency-adaptive acoustic-field prediction model, we choose 7 unheard sounds that have varying spectral distributions and play them as the source. For each sound, we collect 10 data points. We split them equally into validation and test for hyperparameter searching.

	SR ↑	SPL ↑	Soft SPL ↑
Random	0.01	0.07	0.12
DDPPO [302]	0.82	0.63	0.66
Direction Follower [37]	0.67	0.50	0.48
Beamforming [160]	0.02	0.01	0.24
Gan et al. [83]	0.63	0.53	0.68
AFP w/ predicting max	0.54	0.34	0.38
AFP w/o vision	0.84	0.71	0.72
AFP (Ours)	**0.91**	**0.76**	**0.75**

Table 4.7: Results of the AudioGoal navigation experiment and our model strongly outperforms existing methods.

4.4.3 Robot Platform

To deploy our sim2real policy on a real robot, we build our own audio-visual robot by equipping a HelloRobot with a 3Dio binaural microphone as shown in Fig. 4.10. We use Focusrite Scarlett Solo as the audio interface to amplify the audio signals from the binaural microphone.

To start the navigation, we first sample the current audio from the microphone and predict the long-term goal from the acoustic field. We then pass this goal to the robot and use HelloRobot's navigation stack to move the robot towards the goal. Once the robot reaches the long-term goal, it comes to a complete stop for a second to sample the audio again. This process is repeated until the predicted goal location is in the center of the acoustic field. If the sampled long-term goal is in an inaccessible region, we have a time limit of 5 seconds after which the robot stops and samples a new goal.

4.4.4 Experiments

For experiments, I first show the results on the continuous AudioGoal navigation benchmark in Sec. 4.4.4.1, then the acoustic field prediction on real measurements in Sec. 4.4.4.2, and lastly the real navigation results in Sec. 4.4.4.3.

4.4.4.1 Results on Continuous AudioGoal Navigation Benchmark

We first demonstrate the effectiveness of our navigation system on the challenging Continuous AudioGoal navigation benchmark (see Sec. 4.3), where the agent moves in a continuous unseen environment to find the location of a ringing telephone sound. For metrics, we use the common Success Rate (SR), success weighted by inverse path length (SPL), and soft SPL. An episode is considered successful when the agent issues the stop action within 1 meter of the goal. SPL [17] is defined as $\text{SPL}_\text{i} = S_i \cdot l_i / \max(p_i, l_i)$, where i denotes the index of the episode, $S = 1$ when the episode is successful and $S = 0$ otherwise, l denotes the length of the shortest path between the agent and the audio goal, and p denotes the length of the actual path taken by the agent in the episode. Soft SPL is a variation of SPL where $S_i = 1$ for all i.

We compare with the following models: **DDPPO** [302]: an end-to-end reinforcement learning policy trained with distributed proximal policy optimization. **Direction Follower** [37]: this model predicts the direction of the audio goal and navigates with the same waypoint planner. We stop the agent automatically when it is within a 1-meter radius of the goal. **Gan et al.** [83]: this model predicts the (x,y) location of the audio source and navigates using a waypoint planner. The agent stops whenever it reaches its predicted location or the closest navigable point. **Beamforming** [160]: classical beamforming method that calculates the direction of arrival of the sound and navigates with the same waypoint planner.

To further justify our model design choice, we also compare with the following ablations of our own model. **AFP w/ predicting max**: this model does not predict the whole acoustic field. Instead, it predicts a single point that represents the highest point of the local acoustic field. **AFP w/ audio-only**: this model only takes in the audio input, which tests whether the full model uses the visual information when predicting the acoustic field.

Results are shown in Tab. 4.7. Our model strongly outperforms all baselines

and ablations. Compared to DDPPO, our model is more efficient due to its hierarchical nature since the DDPPO model often gets stuck with obstacles and corners. Direction Follower and Gan et al. predict the goal direction/location directly, which is however ill-posed when goals are in some other room at a distance from the robot. As a result, their navigation performance is also pretty poor. For the Beamforming baseline, similar to ours, it also predicts the local direction of arrival of the sound, however, since it is not robust to reverberation and noise, it performs quite badly. Lastly, the two ablations perform comparably to baselines but also underperform the full model, showing it is beneficial to predict the full acoustic field and our model uses visual sensors to understand the environment.

We show the comparison of trajectories with these baselines in the same episode in Fig. 4.14, where our model is more efficient in reaching the goal. We also visualize the acoustic fields of both the ground truth and prediction in Fig. 4.15. Initially, the model predicts high values at the corner (in the direction of the goal), and as the agent gets close to the goal, it predicts high values at the center of the field.

4.4.4.2 Experiment 2: Acoustic Field Prediction on Real Data

Here we evaluate our frequency-adaptive acoustic field prediction (FA-AFP) model on the collected real acoustic field data. We compare our method to the random baseline and ablations of our approach. We consider three ablations: "All-freq AFP" uses all frequencies for prediction. "Best-freq AFP" uses the best frequency band shown in Fig. 4.13 and "Highest-energy AFP" uses the band where the received audio has the highest energy. We measure the performance of different prediction errors with the angle and distance of the predicted max location on the acoustic field. We train our models on 73 sounds and test on 7 unheard sounds.

The results are shown in Tab. 4.8. We show that compared to the random prediction, our All-freq AFP model reduces the prediction error drastically. If we

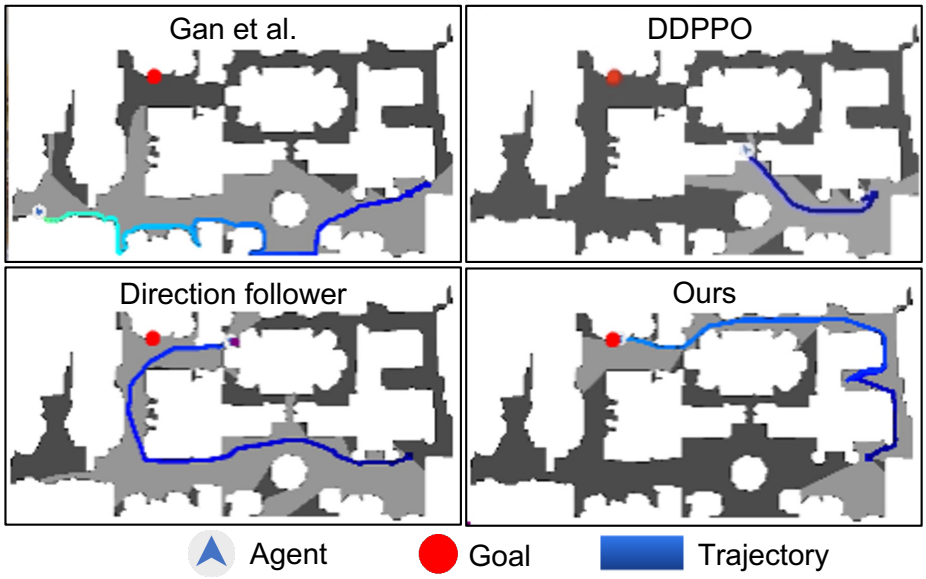


Figure 4.14: Navigation trajectory comparison. Our model successfully navigates to the source while other baselines fail due to either getting stuck or navigating in the wrong direction.

always use the best frequency for prediction, it helps lower the angle error a bit but not the distance error. Using the frequency band with the highest energy brings down the prediction error more. Our frequency-adaptive prediction model (FA-AFP) improves the performance even further, showing the importance of intelligently selecting a frequency band for prediction.

In Fig. 4.16, we show examples of the collected acoustic field and the predicted acoustic field for multiple directions and sounds. Note that the acoustic field is only sampled at a 3×3 grid centered at the robot to reduce the cost of collection. Our predictions are consistently accurate across examples.

4.4.4.3 Experiment 3: Real Robot Navigation

Finally, to validate the whole navigation pipeline, we deploy our navigation policy on the real robot platform (described in Sec. 4.4.3). When deploying on the real robot, one thing that differs from the previously collected real data is that the robot also makes some low-frequency noise while running. To address this issue, we

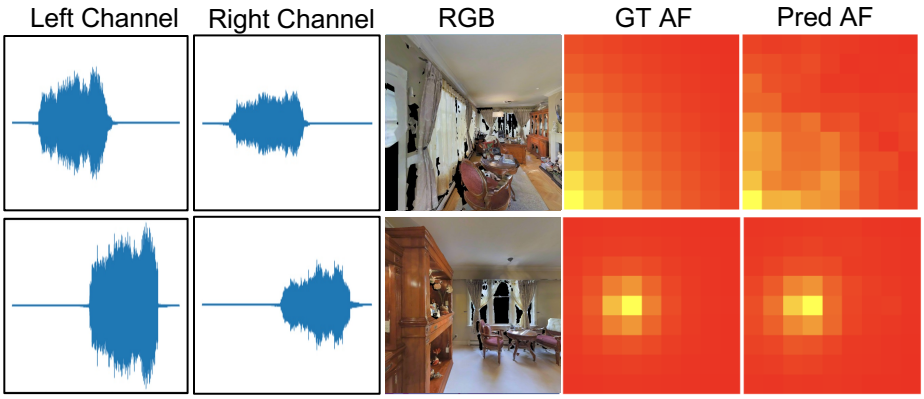


Figure 4.15: Visualization of acoustic field prediction within the same episode. Top row: when the robot is still far from the goal. Bottom row: when the robot is right next to the goal. Our model predicts accurately in both cases.

	Angle ↓	Distance ↓
Random	1.57	1.45
All-freq AFP	0.22	0.74
Best-freq AFP	0.20	0.74
Highest-energy AFP	0.04	0.70
FA-AFP (Ours)	**0.04**	**0.63**

Table 4.8: Results for testing on real acoustic field data.

collect recordings of the robot noise and perform data augmentation by adding the noise to the received sound during training to improve the model performance.

We conduct 20 navigation examples with various source/receiver distances and directions and show that our robot can navigate the sounding object with a 75% success rate. We also tried to deploy the best-performing baseline DDPPO, which however failed all the test scenarios, which is likely due to the significant physical sim2real gap since that model trains with RL end-to-end. We show one navigation step example in Fig. 4.10, where the model predicts the acoustic field correctly.

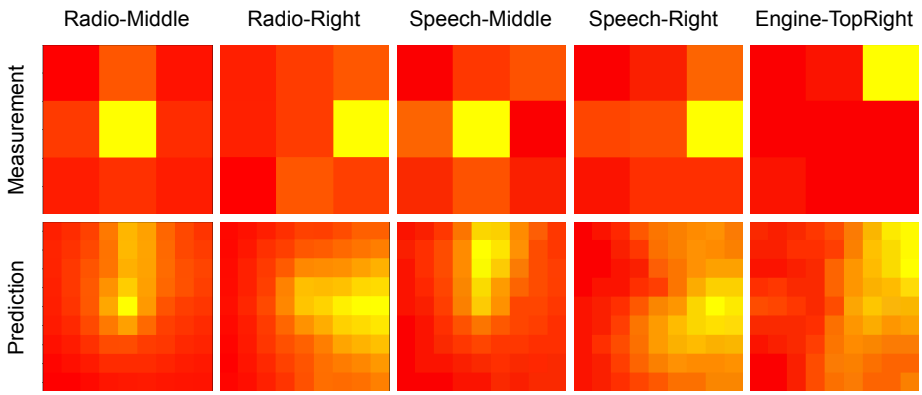


Figure 4.16: Acoustic field predictions on real data. The real data is measured with a lower resolution. We show the prediction and measurement for multiple sounds and directions. Our model predicts all of these cases accurately.

4.5 Conclusions

In this chapter, I showed two audio-visual navigation benchmarks and multiple audio-visual navigation policies that can successfully navigate to find sounding objects in unknown environments. These policies learn to generalize to novel environments as well as novel unheard sounds. In addition, I introduced an approach for transferring the policy from simulation to the real world. While exciting first steps in embodied audio-visual learning, there are some limitations of this work as well as promising future directions.

First of all, in the audio-visual navigation task's assumption, the sounding object is a randomly sampled point in space (not visible) and it emits sounds throughout the whole episode. This is not a realistic assumption because real-world objects are visible and often only produce sounds for a short period of time. I address this issue by introducing the semantic audio-visual navigation task later in Chapter 5.

Secondly, despite the robot being able to navigate successfully to the goal in both simulation and the real world, the current task setting assumes one single sound source present in the environment and the agent predicts the direction of the sound (gradient of the sound pressure field) and moves in that direction. This is

not the case in real-world environments, for example, in coffee shops or restaurants, where all sounds are mixed together, including the sound we want robots to react to. The existing solution does not differentiate different sounds and would fail in such a complex scenario. One possible solution to deal with this is to use language to provide specification of the sound; for example, we could ask the robot to navigate to a specific sound in the environment and reward the agent for reaching that sound only. By doing so, the model will learn to implicitly separate the sound of interest and navigate to find that sound.

Chapter 5: Semantic Audio-Visual Navigation

In Chapter 3, I introduced the simulation platform SoundSpaces, and then in Chapter 4, I introduced further the audio-visual navigation task along with some models that could navigate successfully to find sounding objects. While exciting first steps, existing audio-visual navigation work has two key limitations. First, prior work assumes the target object constantly makes a steady repeating sound (e.g., alarm chirping, phone ringing). While important, this corresponds to a narrow set of targets; in real-world scenarios, an object may emit a sound only briefly or start and stop dynamically. Second, in current models explored in realistic 3D environment simulators, the sound-emitting target has neither a visual embodiment nor any semantic context. Rather, target sound sources are placed arbitrarily in the environment and without relation to the semantics of the scene and objects. As a result, the role of audio is limited to providing a beacon of sound announcing where the object is.

In light of these limitations, we introduce a novel task: *semantic audio-visual navigation*. In this task, the agent must navigate to an object situated contextually in an environment that only makes sound for a certain period of time. Semantic audio-visual navigation widens the set of real-world scenarios to include acoustic events of short temporal duration that are semantically grounded in the environment. It offers new learning challenges. The agent must learn not only how to associate sounds with visual objects, but also how to leverage the semantic priors of objects (along with any acoustic cues) to reason about where the object is likely located in the scene. For example, hearing the dishwasher stop running and issue its end of cycle chime should suggest both what visual object to search for as well as the likely paths for finding it, i.e., towards the kitchen rather than the bedroom. Notably, in the proposed task, the agent is not given any external information about the goal (such as a displacement vector or name of the object to search for). Hence the agent must learn to leverage

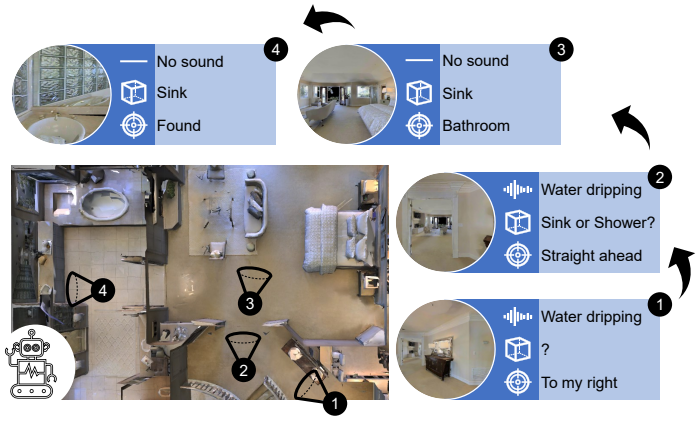


Figure 5.1: Semantic audio-visual navigation in 3D environments: an agent must navigate to a sounding object. Since the sound may stop while the agent searches for the object, the agent is incentivized to learn the association between how objects look and sound, and to build contextual models for where different semantic sounds are more likely to occur (e.g., water dripping in the bathroom).

sporadic acoustic cues that may stop at any time as it searches for the source, inferring what visual object likely emitted the sound even after it is silent. See Figure 5.1.

To tackle semantic AudioGoal, we introduce a deep reinforcement learning model that learns the association between how objects look and how they sound. We develop a *goal descriptor* module that allows the agent to hypothesize the goal properties (i.e., location and object category) from the received acoustic cues before seeing the target object. Coupled with a transformer, it learns to attend to the previous visual and acoustic observations in its memory—conditioned on the predicted goal descriptor—to navigate to the audio source. Furthermore, to support this line of research, we instrument audio-visual simulations for real scanned environments such that semantically relevant sounds are attached to semantically relevant objects.

We evaluate our model on 85 large-scale real-world environments with a variety of semantic objects and their sounds. Our approach outperforms state-of-the-art models in audio-visual navigation with up to an absolute 8.9% improvement in SPL. Furthermore, our model is robust in handling short acoustic signals emitted by the goal with varying temporal duration, and compared to the competitors, it more often

reaches the goal after the acoustic observations end. In addition, our model maintains good performance in the presence of environment noise (distractor sounds) compared to baseline models. Overall, this chapter shows the potential for embodied agents to learn about how objects look and sound through interactions with a 3D environment.

In Sec. 5.1, I define the semantic audio-visual navigation task. In Sec. 5.2 and Sec. 5.3, I present the approach and the experiments respectively. This work was published in CVPR 2021 [36].

5.1 Semantic Audio-Visual Navigation

We introduce the novel task of semantic audio-visual navigation. In this task, the agent is required to navigate in a complex, unmapped environment to find a semantic sounding object—"semantic AudioGoal" for short. Different from AudioGoal [35, 83], the goal sound *need not be periodic*, has *variable duration*, and is associated with a meaningful *semantic object* (e.g., the door creaking is associated with the apartment's door). This setting represents common real world events, and as discussed above, poses new challenges for embodied learning. Relying on audio perception solely to produce step-by-step actions is not sufficient, since the audio event is relatively short. Instead, the agent needs to reason about the category of the sounding object and use both visual and audio perception to predict its location.

3D environments and simulator. Consistent with the active body of computer vision work on embodied AI done in simulation, and to facilitate reproducibility of our work, we rely on a visually and acoustically realistic simulation platform to model an agent moving in complex 3D environments. We use SoundSpaces [35], which enables realistic audio rendering of arbitrary sounds for the real-world environment scans in Replica [265] and Matterport3D [29]. We use the Matterport environments due to their greater scale and complexity. As discussed above, SoundSpaces is Habitat-compatible [247] and allows rendering arbitrary sounds at any pair of source and

receiver (agent) locations on a uniform grid of nodes spaced by 1 m. Next we explain how we extend this audio data to provide semantically meaningful sounds.

Semantic sounds data collection. We use the 21 object categories defined in the ObjectGoal navigation challenge [18] for Matterport3D environments: chair, table, picture, cabinet, cushion, sofa, bed, chest of drawers, plant, sink, toilet, stool, towel, tv monitor, shower, bathtub, counter, fireplace, gym equipment, seating, and clothes. All of these categories have objects that are visually present in Matterport environments. By rendering object-specific sounds at the locations of the Matterport objects, we obtain semantically meaningful and contextual sounds. For example, the water flush sound will be associated with the toilet in the bathroom, and the crackling fire sound with the fireplace in the living room or the bedroom. We filter out object instances that are not reachable by the navigability graph. The number of object instances for train/val/test is 303/46/80 on average for each object category.

We consider two types of sound events: object-emitted and object-related. Object-emitted sounds are generated by the object, e.g., a toilet flushing, whereas object-related sounds are caused by people's interactions with the object, e.g., food being chopped on the counter. To provide a variety of sounds, we search a public database `freesound.org` by the 21 object names to get long copyright-free audio clips per object. We split the original clips (average length 81s) evenly into train/val/test clips. These splits allow the characteristics of the unheard sounds (i.e., waveforms not heard during training) to be related to those in the training set, while still preserving natural variations.* The duration of the acoustic phase in each episode is randomly sampled from a Gaussian of mean 15s and deviation 9s, clipped for a minimum 5s and maximum 500s. If the sampled duration is longer than the length of the audio clip, we replay the clip.

*Note that even the same waveform will sound different when rendered in a new environment; the sound received by the agent is a function of not only that waveform but also the environment geometry and the agent's position relative to the source.

Action space and sensors. The agent's action space is *MoveForward*, *TurnLeft*, *TurnRight*, and *Stop*. The last three actions are always valid, while *MoveForward* only takes effect when the node in front of the agent is reachable from that position (no collision). The sensory inputs are egocentric binaural sound (two-channel audio waveforms), RGB, depth, and the agent's current pose.

Episode specification and success criterion. An episode of semantic AudioGoal is defined by 1) the scene, 2) the agent start location and rotation, 3) the goal location, 4) the goal (object) category and 5) the duration of the audio event. In each episode in a given scene, we choose a random object category and a random instance of that category as the goal. The agent's start pose is also randomly positioned in the scene. In semantic AudioGoal, the agent has to stop near the particular sounding object instance, not simply any instance of the class. This is a stricter success criterion than ObjectGoal [18], which judges an episode as successful if the agent stops near any instance of that category. We define a set of viewpoints around each object within 1 m of the object's boundary; issuing the *Stop* action at any of these viewpoints is considered a successful termination of the episode.

5.2 Approach

We propose SAVi, a novel model for the **s**emantic **a**udio-**v**isual **n**av**i**gation task. SAVi uses a persistent multimodal memory along with a transformer model, which, unlike RNN-based architectures (e.g., [35]) or reactive ones (e.g., [83]), can directly attend to observations with various temporal distances from the current step to locate the goal efficiently. Furthermore, our model learns to capture goal information from acoustic events in an explicit descriptor and uses it to attend to its memory, thus enabling the agent to discover any spatial and semantic cues that may help it reach the target faster.

Our approach has three main components (Figure 5.2): 1) an Observation Encoder that maps the egocentric visual and acoustic observations received by the

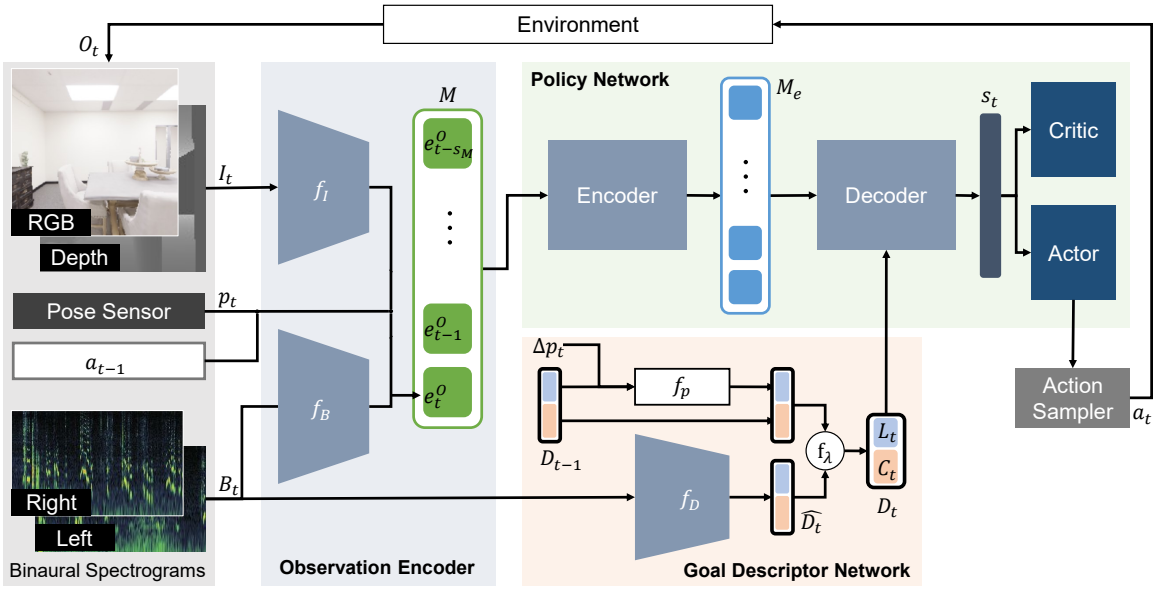


Figure 5.2: In our model, the agent first encodes input observations and stores their features in memory M. Then our goal descriptor network leverages the acoustic cues to dynamically infer and update a *goal descriptor* D_t of the target object, which contains both location L_t and object category C_t information about the goal. By conditioning the agent's scene memory on the goal descriptor, the learned state representation s_t preserves information most relevant to the goal. Our transformer-based policy network attends to the encoded observations in M with self-attention to reason about the 3D environment seen so far, and it attends to M_e with D_t to capture possible associations between the hypothesized goal and the visual and acoustic observations to predict the state s_t. Then, s_t is fed to an actor-critic network, which predicts the next action a_t. The agent receives its reward from the environment based on how close to the goal it moves and whether it succeeds in reaching it.

agent at each step to an embedding space; 2) a Goal Descriptor Network that produces a goal descriptor based on the encoded observations; and 3) a Policy Network that given the encoded observations and the predicted goal descriptor, extracts a descriptor-conditioned state representation and outputs the action distribution. Next, we describe each module. We defer CNN architecture details to Sec. 5.2.4.

5.2.1 Observation Encoder

At each time step t, the agent receives an observation $O_t = (I_t, B_t, p_t, a_{t-1})$, where I is the egocentric visual observation consisting of an RGB and depth image; B is the received binaural audio waveform represented as a two-channel spectrogram; p is the agent pose defined by its location and orientation (x, y, θ) with respect to its starting pose p_0 in the current episode; and a_{t-1} is the action taken at the previous time step.

Our model encodes each visual and audio observation with a CNN, $e_t^I = f_I(I_t)$ and $e_t^B = f_B(B_t)$. Then, the observation O_t encoding is $e_t^O = [e_t^I, e_t^B, p_t, a_{t-1}]$. The model stores the encoding of the observations up to time t in memory $M = \{e_i^O : i = \max\{0, t - s_M\}, \ldots, t\}$ (see Figure 5.2 second column), where s_M is the memory size.

5.2.2 Goal Descriptor Network

As described in Sec. 5.1, the agent does not receive direct information about the goal; rather, it needs to rely solely on its observations to set its own target. Audio carries rich cues about the target—not only its relative direction and distance from the agent, but also the type of object that may have produced the acoustic event. Hence, we leverage the acoustic signal to predict the goal properties, namely its location (spatial) and object category (semantics). Both properties are crucial for successful navigation. The estimated goal location gives the agent an idea of where to find the goal. However, since the acoustic event may be short-lived, and the estimate may be inaccurate, the agent cannot solely rely on this initial estimate. Our model thus aims to also leverage the goal semantics in terms of both the object's likely appearance and the scene's visual context.

The goal descriptor network is a CNN f_D such that $\hat{D}_t = f_D(B_t)$, where $\hat{D}_t$ is the step-wise estimate of the descriptor and it consists of two parts: the current estimate of the goal location $\hat{L}_t = (\Delta x, \Delta y)$ relative to the agent's current pose

p_t, and its predicted object label $\hat{C}_t$. To reduce the impact of noise from a single prediction, the agent aggregates the current estimate with the previous goal descriptor $D_t = f_\lambda(\hat{D}_t, D_{t-1}, \Delta p_t) = (1-\lambda)\hat{D}_t + \lambda f_p(D_{t-1}, \Delta p_t)$, where $f_p(\cdot)$ transforms the previous goal location $\hat{L}_{t-1}$ based on the last pose change Δp_t (the goal label is unaffected by this transformation), and λ is the weighting factor, which is set to 0.5 based on validation. When sound stops (i.e., the sound intensity becomes zero), the agent maintains its latest estimate D_t by simply transforming the previous descriptor based on the pose change Δp_t to obtain the current descriptor $D_t = f_p(D_{t-1}, \Delta p_t)$.

5.2.3 Policy Network

Our reinforcement learning policy network is based on a transformer architecture. Using the memory M collected so far in the episode, the transformer proceeds by encoding these observation embeddings with a self-attention mechanism to capture any possible relations among the inputs, yielding the encoded memory $M_e = \text{Encoder}(M)$. Then, using the predicted goal descriptor D_t, a decoder network attends to all cells in the encoded memory M_e to calculate the state representation $s_t = \text{Decoder}(M_e, D_t)$.

An actor-critic network uses s_t to predict the action distribution and value of the state. The actor and the critic are each modelled by single linear layer neural network. Finally, an action sampler samples the next action a_t from this action distribution, determining the agent's next motion in the 3D scene.

5.2.4 Training

To train the goal descriptor network, we generate pairs of ground truth locations and categories from the simulator for the array of training sounds, and train the prediction network in a supervised fashion. For the category prediction portion, we find off-policy training gives good accuracy; hence we pre-train the classifier on 3.5M collected spectrogram-category pairs at a variety of positions in the training environ-

ments and freeze it during policy training. In contrast, location prediction is learned better on-policy. Training the L_t predictor on-policy has the benefit of matching the training data distribution with policy behavior, leading to higher accuracy. We use the same experience collected for policy training to train the location predictor and update them at the same frequency. We use the mean squared error loss for the location predictor and the cross entropy loss for the goal object label predictor.

For policy training, we follow a two-stage training paradigm (as shown to be effective for transformer-based models [74]) using decentralized distributed proximal policy optimization (DD-PPO) [302]. In the first stage, we set the memory size $s_M = 1$ (the most recent observation) to train the observation encoder without attention. Then, in the second stage, we freeze the observation encoder and train the rest of the model with the full memory size ($s_M = 150$). In both stages, the loss consists of a value network loss to reduce the error of state-value prediction, a policy network loss to produce better action distributions, and an entropy loss to encourage exploration. We refer readers to PPO [254] for more details. To train the policy, we reward the agent with +10 if it reaches the goal successfully and issue an intermediate reward of +1 for reducing the geodesic distance to the goal, and an equivalent penalty for increasing it. We also issue a time penalty of -0.01 per time step to encourage efficiency.

To avoid sampling easy episodes (e.g., short or straight-line paths), we require the geodesic distance from the start pose to the goal to be greater than 4 m and the ratio of Euclidean distance to geodesic distance to be greater than 1.1. We collect 0.5M/500/1000 episodes for train/val/test splits for all 85 Matterport3D SoundSpaces environments.

We train our model with Adam [143] with a learning rate of 2.5×10^{-4} for the policy network and 1×10^{-3} for the descriptor network. We roll out policies for 150 steps, update them with collected experiences for two epochs, and repeat this procedure until convergence. We train all methods, both ours and the baselines, for

	Heard Sounds					*Unheard Sounds*				
	Success	SPL	SNA	DTG	SWS	Success	SPL	SNA	DTG	SWS
Random	1.4	3.5	1.2	17.0	1.4	1.4	3.5	1.2	17.0	1.4
ObjectGoal RL	1.5	0.8	0.6	16.7	1.1	1.5	0.8	0.6	16.7	1.1
Gan et al. [83]	29.3	23.7	**23.0**	11.3	14.4	15.9	12.3	11.6	12.7	8.0
AV-Nav [35]	21.6	15.1	12.1	11.2	10.7	18.0	13.4	12.9	12.9	6.9
AV-WaN [38]	20.9	16.8	16.2	10.3	8.3	17.2	13.2	12.7	11.0	6.9
SMT [74] + Audio	22.0	16.8	16.0	12.4	8.7	16.7	11.9	10.0	12.1	8.5
SAVi (**Ours**)	**33.9**	**24.0**	18.3	**8.8**	**21.5**	**24.8**	**17.2**	**13.2**	**9.9**	**14.7**

Table 5.1: Navigation performance on the SoundSpaces Matterport3D dataset [35]. Our SAVi model has higher success rates and follows a shorter trajectory (SPL) to the goal compared to the state-of-the-art. Equipped with its explicit goal descriptor and having learned semantically grounded object sounds from training environments, our model is able to reach the goal more efficiently—even after it stops sounding—at a significantly higher rate than the closest competitor (see the SWS metric). All metrics are the higher the better except for DTG.

300M steps for them to fully converge.

At each time step, the agent receives a binaural audio clip of 1s, represented as two 65×26 spectrograms. The audio is computed by convolving the appropriate impulse response from SoundSpaces with the source audio waveform, thereby generating the sound the agent would hear in that environment at its current position relative to the source. The RGB and depth images are center cropped to 64×64. Both the observation encoder CNNs f_B and f_I and the descriptor network f_D use a simplified ResNet-18 [115] that is trained from scratch. For the transformer model, we use one encoder layer and one decoder layer, which employ multi-attention with 8 heads. The hidden state size is 256 and the memory size s_M is 150, matching the frequency of policy updates.

5.3 Experiments

Baselines. We compare our model to the following baselines and existing work:

1. **Random**: A random baseline that uniformly samples one of three actions and executes *Stop* automatically when it reaches the goal (perfect stopping).

2. **ObjectGoal RL**: An end-to-end RL policy with a GRU encoder and RGB-D inputs (no audio). It is given the one-hot encoding of the true category label as an additional input to search for the goal object instance. This baseline is widely used in ObjectGoal tasks [109, 31, 193, 30]. We train this method for 800M steps with perfect stopping.

3. **Gan et al. [83]**: A modular audio-visual model that trains a goal location predictor offline and uses a geometric planner for planning. Since the original model can not handle sporadic audio events, we improve its goal location predictor with our update operation f_λ.

4. **AV-Nav [35] (see Sec. 4.1.1)**: An end-to-end RL policy that encodes past memory with a GRU RNN and is trained to reach the goal using audio and visual observations.

5. **AV-WaN [38] (see Sec. 4.2)**: A hierarchical RL model that records acoustic observations on the ground plane, predicts waypoints, and uses a path planner to move towards these waypoints using a sequence of navigation actions.

6. **SMT [74] + Audio**: We adapt the scene memory transformer (SMT) model [74] to our task by also encoding the audio observation in its memory. Unlike our model, it does not explicitly predict the goal description and relies only on the cues available in memory to reach the goal. The latest observation embedding is used as decoder input to decode M_e and predict the state.

All models use the same reward function and inputs. For all methods, there is no actuation noise since audio rendering is only available at grid points (see [35] for details).

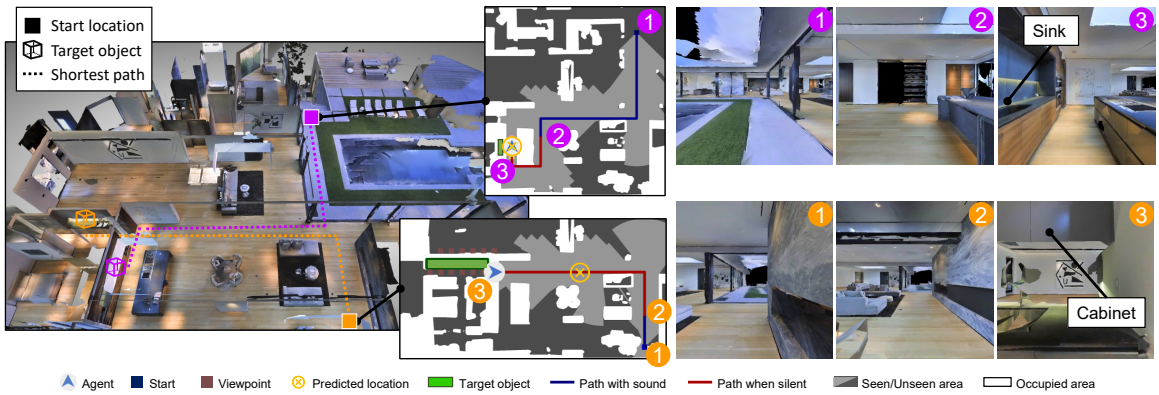


Figure 5.3: Example SAVi navigation trajectories. In the first episode (top/magenta) the agent hears a water dripping sound and in the second episode (bottom/orange) a sound of opening and closing a door. For each episode, we show three egocentric visual views (right) sampled from the agent's trajectory at the start location ①, when the sound stops ②, and at the end location ③. In the top episode, the acoustic event lasts for two thirds of the trajectory and when the sound stops the agent has an accurate estimate of the object location that helps it find the sounding object (the sink). The second episode (bottom) has a much shorter acoustic event. The agent's estimate of the object location is inaccurate when the sound stops but still helps the agent as a general directional cue. The agent leverages this spatial cue *and* the semantic cue from its estimate of the object category, a cabinet, to attend to its multimodal memory to find the object in the kitchen and end the episode successfully.

Metrics. We evaluate the following navigation metrics: 1) success rate: the fraction of successful episodes; 2) success weighted by inverse path length (SPL): the standard metric [8] that weighs successes by their adherence to the shortest path; 3) success weighted by inverse number of actions (SNA) [38]: this penalizes collisions and in-place rotations by counting number of actions instead of path lengths; 4) average distance to goal (DTG): the agent's distance to the goal when episodes are finished; 5) success when silent (SWS): the fraction of successful episodes when the agent reaches the goal after the end of the acoustic event.

Navigation results. Following standard protocol [35] we evaluate all models in two settings: 1) *heard sounds*—train and test on the same sound 2) *unheard sounds*—train and test on disjoint sounds. In both cases, the test environments are always

unseen, hence both require generalization. All results are averaged over 1,000 test episodes.

Table 5.1 shows the results. Our SAVi approach outperforms all other models by a large margin on all metrics—with 0.3%, 8.9%, 7.2%, 7.2% absolute gains in SPL on *heard sounds* and 4.9%, 3.8%, 4%, 5.3% absolute SPL gains on *unheard sounds* compared to Gan et al. [83], AV-Nav [35], AV-WaN [38], and SMT [74], respectively. This shows our model leverages audio-visual cues intelligently and navigates to goals more efficiently. AV-WaN represents the state-of-the-art for AudioGoal audio-visual navigation. Our SAVi model's gains over AV-WaN show both 1) the distinct new challenges offered by the semantic AudioGoal task, and 2) our model's design effectively handles them.[†]

In addition, our model improves the success-when-silent (SWS) metric by a large margin compared to the closest competitor. This emphasizes the advantage of our goal descriptor module. The explicit and persistent descriptor for the goal in our model helps to maintain the agent's focus on the target even after it stops emitting a sound. Although the SMT+Audio [74] model also has access to a large memory pool and can leverage implicit goal information from old observations, lacking our goal descriptor and the accompanying goal-driven attention, it underperforms our model by a sizeable margin.

As expected, Random does poorly on this task due to the challenging complex environments. Although ObjectGoal RL has the goal's ground truth category label as input, it fails in most cases. This shows that knowing the category label by itself is insufficient to succeed in this task; the agent needs to locate the specific instance of that category, which is difficult without the acoustic cues.

[†]While AV-WaN [38] reports large performance improvements over AV-Nav [35] on the standard AudioGoal task, we do not observe similar margins between the two models here. We attribute this to temporal gaps in the memory caused by AV-WaN's waypoint formulation—which are not damaging for constantly sounding targets, but do cause problems for semantic AudioGoal.

Navigation trajectories. Figure 5.3 shows test episodes for our SAVi model. The agent uses its acoustic-visual perception and memory along with the spatial and semantic cues from the acoustic event, whether from a long event (water dripping sound) or a short one (opening and closing a door sound), to successfully find the target objects (the sink and the cabinet).

Common failure cases are when: 1) the sound stops too early in the episode, and the agent has not accumulated enough spatial or semantic cues about the goal. In this case the agent might either search for the wrong object (noisy semantics) or search for the object in the wrong place (noisy location); 2) the agent issues a premature stop action near the target object but not exactly at the right location.

Distractor sounds. In our tests so far, there is a single acoustic event per episode, whether comprised of a heard or unheard sound (Table 5.1). Next, we generalize the setting further to include *unheard distractor sounds*—sounds happening simultaneously with the target object. This corresponds to real-world scenarios, for example, where the door slams shut while the AC is humming. For this setting to be well-defined, the agent must know which sound is its target; hence, we input the one-hot encoding of the target object to all models and concatenate it with their state features. For our model, in addition to replacing C_t with this one-hot encoding, we also use it as input to the location prediction network along with B_t. This allows the location prediction network to learn to identify which of the sounds mixed in the input needs to be localized. We use the 102 periodic sounds from SoundSpaces [35] as the set of possible distractor sounds, which are disjoint from the target object sounds curated for this work. We divide these 102 sounds into non-overlapping 73/11/18 splits for train/val/test, and hence the distractor sound at test time is unheard. In each episode, we randomly position one distractor sound in the environment at a location different from the goal.

Table 5.2 shows the results. While the performance of the baselines suffers from the distracting environment noise, our agent is still able to reach a success rate

	Success ↑	SPL ↑	SNA ↑	DTG ↓	SWS ↑
AV-Nav [35]	4.0	2.4	2.0	14.7	2.3
AV-WaN [38]	3.0	2.0	1.8	14.0	1.6
SMT [74]+Audio	4.2	2.9	2.1	14.9	2.8
SAVi (**Ours**)	**11.8**	**7.4**	**5.0**	**13.1**	**8.4**

Table 5.2: Navigation performance on *unheard sounds* in the presence of unheard distractor sounds.

of 11.8% and SPL of 7.4%, which is 7.6% and 4.5% higher than the best-performing baseline. This shows the proposed inferred goal descriptor helps the agent attend to important observations to capture semantic and spatial cues, making our model more robust to the environment noise. That said, the absolute performance declines for all methods in this hard setting. We plan to investigate ways to explicitly separate the "clutter" sounds in future work.

Analyzing the goal descriptor. Next we ablate the two main components in the goal descriptor, location and category, to study their relative impact for the *unheard sounds* experiment from Table 5.1. Table 5.3 shows that ablating any component results in a performance drop. L_t has a comparatively larger impact on our model's performance.

Next we analyze the *successful* episodes in the context of L_t and C_t. For 56% of them, our model ends the episode by stopping at its own estimate of the goal location in its descriptor, suggesting that the agent has successfully used its directional sound prediction to guide its movements. On the other hand, for the other 44%, the agent stops at a (correct) location *different* than L_t, suggesting that the agent has relied more on the visual context cues leading to the anticipated object C_t. In fact, if we inject a random category label instead of C_t at the start of the episode, success rates and SPL drop up to 8%. The learned associations between the spatial and semantic cues are important for success; breaking these associations with random category labels forces the agent to attend to contradictory cues about the goal in its memory,

	Success ↑	SPL ↑	SNA ↑	DTG ↓	SWS ↑
C_t-only	20.5	13.5	11.6	9.8	11.0
L_t-only	23.9	16.2	**13.5**	**9.3**	13.8
w/o aggregation	21.9	14.3	11.1	9.7	13.4
Full model	**24.8**	**17.2**	13.2	9.9	**14.7**

Table 5.3: Ablation experiment results.

thus increasing the chance of failure.

To understand if the performance gain comes from our goal descriptor or the transformer, we further ablate our model by replacing the transformer with an RNN. We find that our goal descriptor network also provides significant improvements when combined with RNNs.

Goal descriptor accuracy and aggregation. The goal descriptor network has two main modules: 1) $f_D(\cdot)$, which produces the current descriptor estimate and 2) an aggregation function $f_\lambda(\cdot)$, which aggregates the current estimate with the previous goal descriptor. Next we evaluate goal prediction accuracy with and without aggregation, as well as how aggregation impacts the navigation performance.

The average location prediction error is 8.1 m and the average category prediction accuracy is 64.5% with aggregation, and 8.4 m, 53.6% without aggregation. Aggregation is important because the source sound is divided into 1s clips for each step, and the characteristics of the sound in some seconds are harder to identify, e.g., the silent moment between pulling and pushing a chest of drawers. Essentially, aggregation stabilizes the goal descriptor prediction. Navigation performance is affected as well: success rate and SPL drop about 3 points without aggregation ("w/o aggregation" ablation in Table 5.3).

Robustness to silence duration. Figure 5.4 analyzes how the models perform after the goal sound stops. We plot the cumulative success rate vs. silence ratio, where the latter is the ratio of the minimum number of actions required to reach

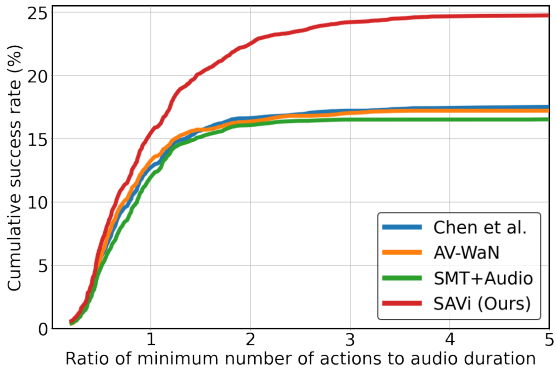


Figure 5.4: Cumulative success rate vs. silence percentage.

the goal to the duration of audio. A point (x, y) on this plot means the fraction of successful episodes with ratios up to x among all episodes is y. When this ratio is greater than 1, no agent can reach the goal before the audio stops. The greater this ratio is, the longer the fraction of silence, and hence the harder the episode. Indeed, we see for all models the success rate accumulates more slowly as the ratio becomes bigger. However, while the success rates of AV-Nav [35], AV-WaN [38], and SMT [74] increase only marginally for ratios greater than 1, our model shows a noticeable increase after the ratios surpass 1 and even 2. This indicates our model is able to cope with long silence to reach goals, thanks to the guidance of our predicted goal descriptor and its attention on the memory.

5.4 Conclusions

In this chapter, we introduce the task of semantic audio-visual navigation in complex 3D environments. To support this task, we expand an existing audio simulation platform to provide semantically grounded object sounds. We introduce a transformer-based model that learns to predict a goal descriptor capturing both spatial and semantic properties of the target. By encoding the observations conditioned on this goal descriptor, our model learns to associate acoustic events with visual observations. We show that our approach outperforms existing state-of-the-art models.

We provide an in-depth analysis of the impact of the goal descriptor and its components and show that our model is more robust to long silence duration and acoustic distractors.

There are however some limitations to this work. One of them is that this work relies on the curation of the semantic sounds and the assignment of these sounds to objects to learn the correspondence. This process of creating the dataset is time-consuming and prevents the model from scaling to more diverse scenarios in the real world. One possible solution is to automatically create the dataset by leveraging existing video datasets such as AudioSet [93] or VGG-Sound [46], where visual objects and the sounds co-occur.

Another limitation of this work is that the generalization is evaluated on unheard audio clips rather than unheard categories. To generalize to sounds of unheard categories requires building an external knowledge base, i.e., the correspondence between visual objects and the sounds, and the semantic correlation between the object and the space.

I demonstrated my efforts in building embodied audio-visual agents in both this chapter and the previous chapter. To achieve the goal of deploying these policies on real robots, besides solving the sim2real problem discussed earlier, it is also important to interact with humans and take humans' speech commands as input, e.g., when someone asks a robot "Bring me a coffee", the robot needs to both understand the command and execute it. However, understanding (recognizing) speeches from a distance is not trivial because there is usually a strong presence of reverberation. In the next chapter, I will present how I approach this problem by leveraging the visual cues.

Chapter 6: Learning Audio-Visual Dereverberation

In Chapter 4 and Chapter 5, I presented several approaches for embodied audio-visual learning, where the agent needs to perceive the environment while actively making decisions. When the agent hears sounds in the environment, not only it is important to sense the direction and distance cues from the audio, but also it needs to understand the semantics of the sound, e.g., recognizing what objects might produce the sound in Chapter 5 or understanding humans' speech commands like "bring me a coffee". Beyond robotics, perceiving sounds and understanding sounds have a wide range of applications, such as audio/video processing and machine translation. Motivated by this application, in this chapter, instead of studying joint perception and decision-making, I will focus on just the perception part. More specifically, I will demonstrate how to enhance sounds received in spaces by leveraging visual cues. This work was published in ICASSP 2023 [42].

Audio reverberation occurs when multiple reflections from surfaces and objects in the environment build up then decay, altering the original audio signal. While reverberation bestows a realistic sense of spatial context, it also can degrade a listener's experience. In particular, the quality of human speech is greatly affected by reverberant environments—as illustrated by how difficult it can be to parse the words of a family member speaking loudly from another room in the house, a tour guide describing the artwork down the hall of a magnificent cavernous cathedral, or a colleague participating in a Zoom call from a cafe. Consistent with the human perceptual experience, automatic speech recognition (ASR) systems noticeably suffer when given reverberant speech input [144, 326, 271, 113, 307, 71]. Thus there is great need for intelligent *dereverberation* algorithms that can strip away reverb effects for speech enhancement, recognition, and other downstream tasks, which could in turn benefit many applications in teleconferencing, assistive hearing devices, augmented reality, and video indexing.

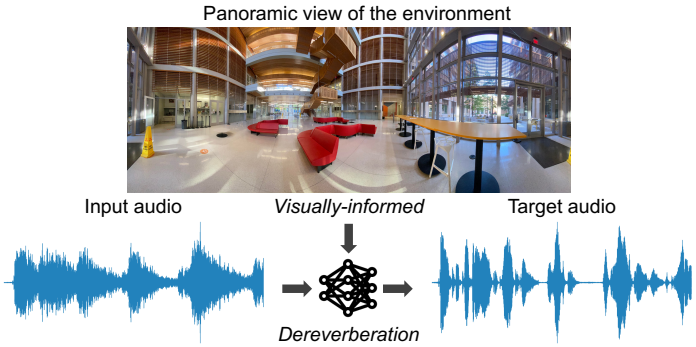


Figure 6.1: The goal of audio-visual dereverberation is to leverage the visual observation of the environment to improve speech enhancement.

The audio community has made steady progress devising machine learning solutions to tackle speech dereverberation [71, 97, 259, 113, 306, 325, 267]. The general approach is to take a reverberant speech signal, usually represented with a Short-Time Fourier Transform (STFT) spectrogram, and feed it as input to a model that estimates a clean version of the signal with the reverberation removed. Past approaches have tackled this problem with signal processing and statistical techniques [205, 206], while many modern approaches are based on neural networks that learn a mapping from reverberant to clean spectrograms [113, 71, 77]. To our knowledge, all existing models for dereverberation rely purely on audio. Unfortunately this often underconstrains the dereverberation task since the latent parameters of the recording space are not discernible from the audio alone.

However, we observe that in many practical settings of interest—video conferencing, augmented reality, Web video indexing—reverberant audio is naturally accompanied by a visual (video) stream. Importantly, the visual stream offers valuable cues about the room acoustics affecting reverberation: where are the walls, how are they shaped, where is the human speaker, what is the layout of major furniture, what are the room's dominant materials (which affect absorption), and even what is the facial appearance and/or body shape of the person speaking (since body shape determines the acoustic properties of a person's speech, and reverb time is frequency dependent). For example, reverb is typically stronger when the speaker is

further away; speech is more reverberant in a large church or hallway; heavy carpet absorbs more sound. See Figure 6.2. While some recent work explores acoustic modeling using images [178, 39, 173, 142], no prior work has investigated how to leverage visual-acoustic cues for dereverberation.

Our idea is to learn to dereverberate speech from audio-visual observations (Fig. 6.1) In this task, the input is reverberant speech and visual observations of the environment surrounding the human speaker, and the output is a prediction of the clean source audio. To tackle this problem, there are two key technical challenges. First, how to model the multi-modal dereverberation process in order to infer the latent clean audio. Second, how to secure appropriate training data spanning a variety of physical environments for which we can sample speech with known ground truth (non-reverberant, anechoic) audio. The latter is also non-trivial because ordinary audio/video recordings are themselves corrupted by reverberation but lack the ground truth source signal we wish to recover.

For the modeling challenge, we introduce an end-to-end approach called Visually-Informed Dereverberation of Audio (VIDA). VIDA consists of a Visual Acoustics Network (VAN) that learns reverberation properties of the room geometry, object locations, and speaker position. Coupled with a multi-modal UNet dereverberation module, it learns to remove the reverberations from a single-channel audio stream. In addition, we propose an audio-visual (AV) matching loss to enforce consistency between the visually-inferred reverberation features and those inferred from the audio signal. We leverage the outputs of our model for multiple downstream tasks: speech enhancement, speech recognition, and speaker identification.

Next, to address the training data challenge, we develop SoundSpaces-Speech, a new large-scale audio-visual dataset based on SoundSpaces (presented in Chapter 3). Our data approach inserts "clean" audio voices together with a 3D humanoid model at various positions within an array of indoor environments, then samples the images and properly reverberating audio when placing the receiver microphone and camera

at other positions in the same house. This strategy allows sampling realistic audio-visual instances coupled with ground truth raw audio to train our model, and it has the added benefit of allowing controlled studies that vary the parameters of the capture setting. As we will show, the data also supports sim2real transfer for applying our model to real audio-visual observations.

Our main contributions are to 1) present the task of audio-visual dereverberation, 2) address it with a new multi-modal modeling approach and a novel reverb-visual matching loss, 3) provide a benchmark evaluation framework built on both SoundSpaces-Speech and real data, and 4) demonstrate the utility of AV dereverberation for multiple practical tasks. We first train and evaluate our model on 82 large-scale real-world environments—each a multi-room home containing a variety of objects—coupled with speech samples from the LibriSpeech dataset [214]. We consider both near-field and far-field settings where the human speaker is in-view or quite far from the camera, respectively. The proposed model outperforms methods restricted to the audio stream, and improves the state of the art for multiple tasks with speech enhancement. We also show that our model trained in simulation can transfer directly to real-world data. Overall, our work shows the potential for speech enhancement models to benefit from seeing the 3D environment.

I first define the task in Sec. 6.1 and then introduce the dataset, approach and experimental results in Sec. 6.2, Sec. 6.3 and Sec. 6.4 respectively.

6.1 The Audio-Visual Dereverberation Task

We introduce the novel task of *audio-visual dereverberation*. In this task, a speaker (or other sound source) and a listener are situated in a 3D environment, such as the interior of a house. The speaker—whose location is unknown to the listener—produces a speech waveform A_s. A superposition of the direct sound and the reverb is captured by the listener, denoted A_r. The reverberant speech A_r can be modeled as the convolution of the anechoic source waveform A_s with the room impulse response

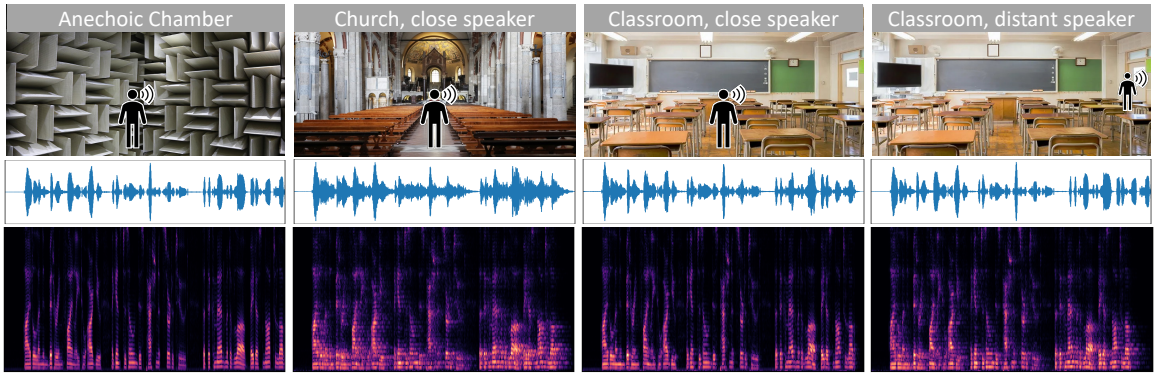


Figure 6.2: Visual cues reveal key factors influencing reverb effects on human speech audio. For example, these audio speech samples (depicted as waveforms and spectrograms) are identical lexically, but have very different reverberation properties owing to their differing environments. In the church, reverb is strong, in the classroom it is less, and when the speaker is distant from the camera it is again more evident.

R, i.e. $A_r(t) = A_s(t) * R(t)$ [207]. It is possible in principle to measure the RIR R for a real-world environment, but doing so can be impractical when the source and listener are able to move around or must cope with different environments. Furthermore, in the common scenario where we want to process video captured in environments to which we have no physical access, measuring the RIR is simply impossible.

Crucially to our task, we consider an alternative source of information about the environment: vision. We assume the listener has an RGB-D observation of its surroundings, obtained from a RGB-D camera or an RGB camera coupled with single-image depth estimation [67, 98]. Intuitively, we should be able to leverage the information about the environment's geometry and material composition that is implicit in the visual stream—as well as the location of the speaker (if visible)—to estimate its reverberant characteristics. We anticipate that these cues can inform an estimate of the room acoustics, and thus the clean source waveform. Given the RGB I_r and depth image I_d captured by the listener from its current vantage point, the task is to predict the source waveform A_s from the images and reverberant audio: $\hat{A}_s(t) = f_p([I_r, I_d, A_r(t)])$. This setting represents common real-world scenarios previously discussed, and poses new challenges for speech enhancement and recognition.

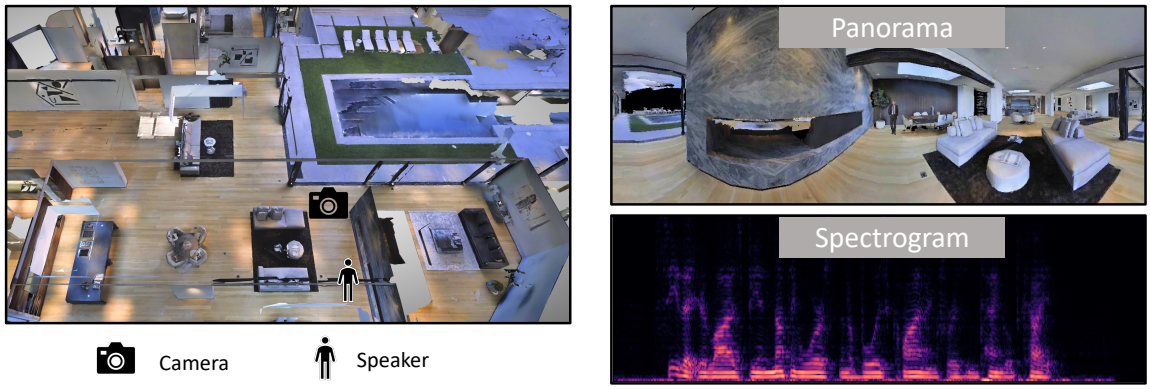


Figure 6.3: Audio-visual rendering for a Matterport environment. Left: bird's-eye view of the 3D environment. Right: panorama image rendered at the camera location and the corresponding received spectrogram.

6.2 Dataset Curation

For the proposed task, obtaining the right training data is itself a challenge. Existing video data contains reverberant audio but lacks the ground truth anechoic audio signal, and existing RIR datasets [181, 130, 195] do not have images paired with the microphone position. We introduce both real and simulated datasets to enable reproducible research on audio-visual deverberation.

3D environments and acoustic simulator. First we introduce a large-scale dataset in which we couple real-world visual environments with state-of-the-art audio simulations accurately capturing the environments' spatial effects on real samples of recorded speech. We want our dataset to allow control of a variety of physical environments, the positions of the listener/camera and sources, and the speech content of the sources—all while maintaining both the observed reverberant $A_r(t)$ and ground truth anechoic $A_s(t)$ sounds. To this end, we leverage the audio-visual simulator SoundSpaces [35], which provides precomputed RIRs $R(t)$ on a uniform grid of resolution 1 m for the real-world environment scans in Replica [265] and Matterport3D [29]. We use 82 Matterport environments due to their greater scale and complexity; each environment has multiple rooms spanning on average 517 m^2.

SoundSpaces-Speech. We extend SoundSpaces to construct reverberant speech. As the source speech corpus we use LibriSpeech [214], which contains 1,000 hours of 16kHz read English speech from audio books, and is widely used in the speech recognition literature. We train our models with the train-clean-360 split, and use the dev-clean and test-clean sets for validation and test splits, respectively. Note that these splits have non-overlapping speaker identities. Similarly, we use the standard disjoint train/val/test splits for the Matterport 3D visual environments [35]. Thus, neither the houses nor speaker voices observed at test time are ever observed during training.

For each source utterance, we randomly sample a source-receiver location pair in a random environment, then convolve the speech waveform $A_s(t)$ with the associated SoundSpaces RIR $R(t)$ to obtain the reverberant $A_r(t)$. To augment the visual scene, we insert a 3D humanoid of the same gender as the real speaker at the speaker location and render RGB and depth images at the listener location. We consider two types of image: panorama and normal field of view (FoV). For the panorama image, we stitch 18 images each having a horizontal FoV of 20 degrees, for a full image resolution of 192×756. For the normal FoV, we render images with a 80 degree FoV, at a resolution of 384×256. While the panorama gives a fuller view of the environment and thus should allow the model to better estimate the room acoustics, the normal FoV is more common in existing video and thus will facilitate our model's transfer to real data. See Fig. 6.3. We generate 49,430/2,700/2,600 such samples for the train/val/test splits, respectively.

Real data collection. To explore whether models trained in simulation can also work in the real world, we also collect a set of real images and speech recordings while preserving the ground truth anechoic audio.

To collect image data, we use an iPhone 11 camera to capture a panoramic RGB image and a monocular depth estimation algorithm [98] to generate the corresponding depth image.

To record the audio, we use a ZYLIA ZM-1 microphone. We place both the camera and microphone at the same height (1.5m) as the SoundSpaces RIRs. For the source speech, we play utterances from the LibriSpeech test set through a loudspeaker held by a person facing the camera. We collect data from varying environments, including auditoriums, meeting rooms, atriums, corridors, and classrooms. For each environment, we vary the speaker location from near-field to mid-field to far-field. For each location, we play around 10 utterances. During data collection, the microphone also records ambient sounds like people chatting, door opening, AC humming, etc. In total, we collect 200 recordings. Code and data are available at `https://github.com/facebookresearch/learning-audio-visual-dereverberation`.

6.3 Approach

We propose the Visually-Informed Dereverberation of Audio (VIDA) model, which leverages visual cues to learn representations of the environmental acoustics and sound source locations to dereverberate audio. While our model is agnostic to the audio source type, we focus on speech due to the importance of dereverberating speech for downstream analysis. VIDA consists of two main components (Figure 6.4): 1) a Visual Acoustics Network (VAN), which learns to map RGB-D images of the environment to features useful for dereverberation, and 2) the dereverberation module itself, which is based on a UNet encoder-decoder architecture. The UNet encoder takes as input a reverberant spectrogram, while the decoder takes the encoder's output along with the visual dereverberation features produced by the VAN and reconstructs a dereverberated version of the audio.

Visual Acoustics Network. Visual observations of a scene reveal information about room acoustics, including room geometry, materials, object locations, and the speaker position. We devise the VAN to capture all these cues into a latent embedding vector, which is subsequently used to remove reverb. This network takes as its input an RGB image I_r and a depth image I_d, captured from the listener's current position

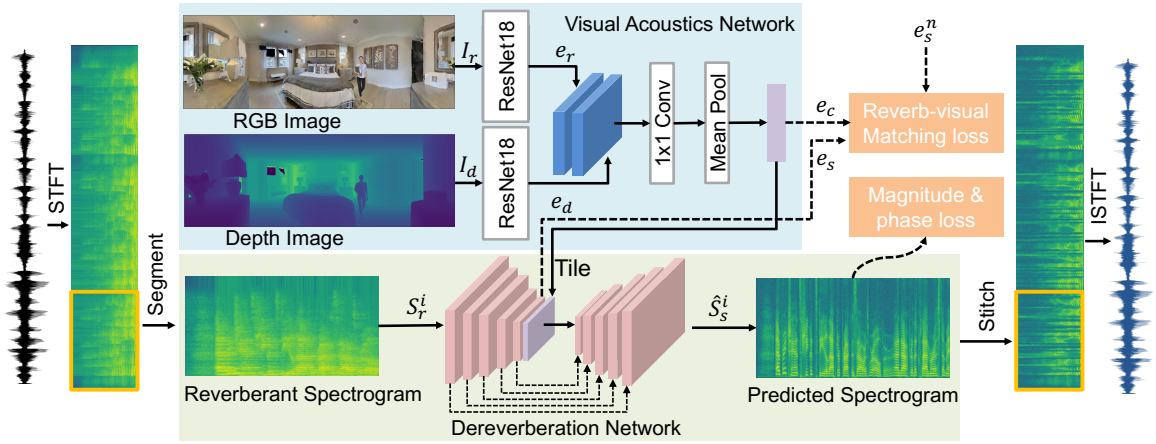


Figure 6.4: VIDA model architecture. We convert the input speech to a spectrogram and use overlapping sliding windows to obtain 2.56 second segments. For visual inputs, we use separate ResNet18 networks to extract features e_r and e_d, which are fused to obtain e_c. We feed the spectrogram segment S_r^i to a UNet encoder, tile and concatenate e_c with the encoder's output, then use the UNet decoder to predict the clean spectrogram $\hat{S}_s^i$. During inference, we stitch the predicted spectrogams back into a full spectrogram and use Griffin-Lim [106] to reconstruct the output dereverberated waveform.

within the environment. The depth image contains information about the geometry of the environment and arrangement of objects, while the RGB image contains more information about their material composition. To better model these different information sources, we use two separate ResNet18 [115] networks to extract their features, i.e. $e_r = f_r(I_r)$ and $e_d = f_d(I_d)$. We concatenate e_r and e_d channel-wise and feed the result to a 1x1 convolution layer $f_c(\cdot)$ to reduce the number of total channels to 512 followed by a subsequent pooling layer $f_l(\cdot)$ to reduce the spatial dimension, resulting in the output vector $e_c = f_l(f_c([e_r; e_d]))$.

Dereverberation Network. To recover the clean speech audio, we use the UNet [240] architecture, a fully convolutional network often used for image segmentation. We first use the Short-Time Fourier Transform (STFT) to convert the reverberant input audio A_r to a complex spectrogram S_r. We treat S_r as a 2-dimensional, 2-channel image, where the horizontal dimension represents time, the vertical dimension repre-

sents frequency, and the two channels represent the log-magnitude and phase angle. Our UNet takes spectrograms of a fixed size of 256×256 as input, but in general the duration of the speech audio we wish to dereverberate will be variable. Therefore, the model processes the full input spectrogram using a series of overlapping, sliding windows. Specifically, we segment the spectrogram along the time dimension into a sequence of fixed-size chunks $S_r^{seg} = \{S_r^1, S_r^2, ..., S_r^n\}$ using a sliding window of length s frames and 50% overlap between consecutive windows to avoid boundary artifacts. To derive the ground-truth target spectrograms used in training, we perform the exact same segmentation operation on the clean source audio A_s to obtain $S_s^{seg} = \{S_s^1, S_s^2, ..., S_s^n\}$.

During training, when a particular waveform S_r is selected for inclusion in a data batch, we randomly sample one of its segments S_r^i to be the input to the model, and choose the corresponding S_s^i as the target. We first compute the output of the VAN, e_c, for the environment image associated with S_r. Next, S_r^i is fed to the UNet's encoder to extract the intermediate feature map $e_s = f_{enc}(S_r^i)$. We then spatially tile and concatenate e_c channel-wise with e_s, and feed the fused features to the UNet decoder, which predicts the source spectrogram segment $\hat{S}_s^i = f_{dec}([e_s, e_c])$.

Spectrogram prediction loss. The primary loss function we use to train our model is the Mean-Squared Error (MSE) between the predicted and ground-truth spectrograms, treating the magnitude and phase separately. For a given predicted spectrogram segment $\hat{S}_s^i$, let $\hat{M}_s^i$ denote the predicted log-magnitude spectrogram, $\hat{P}_s^i$ denote the predicted phase spectrogram, and M_s^i and P_s^i denote the respective ground-truth magnitude and phase spectrograms. We define the magnitude loss as:

$$L_{magnitude} = ||M_s^i - \hat{M}_s^i||_2.$$

To address the issue of phase wraparound, we map the phase angle to its corresponding rectangular coordinates on the unit circle and then compute the loss for the phase:

$$L_{phase} = ||\sin(P_s^i) - \sin(\hat{P}_s^i)||_2 + ||\cos(P_s^i) - \cos(\hat{P}_s^i)||_2.$$

Reverb-visual matching loss. To reinforce the consistency between the visually-inferred room acoustics and the reverberation characteristics learned by the UNet encoder, we also employ a contrastive reverb-visual matching loss:

$$L_{matching}(e_c, e_s, e_s^n) = \max\{d(f_n(e_c), f_n(e_s)) - d(f_n(e_c), f_n(e_s^n)) + m, 0\}.$$

Here, $d(x, y)$ represents L2 distance, $f_n(\cdot)$ applies L2 normalization, m is a margin, and e_s^n is a different speech embedding sampled from the same data batch. This loss forces the embeddings output by the VAN and the UNet encoder to be consistent, which we empirically show to be beneficial.

Training. Our overall training objective (for a single training example) is as follows:

$$L_{total} = L_{magnitude} + \lambda_1 L_{phase} + \lambda_2 L_{matching},$$

where λ_1 and λ_2 are weighting factors for the phase and matching losses. To augment the data, we further choose to rotate the image view for a random angle for each input during training. This is possible because our audio recording is omni-directional and is independent of camera pose. This data augmentation strategy prevents the model from overfitting; without it our model fails to converge. It creates a one-to-many mapping between reverb and views, forcing the model to learn a viewpoint-invariant representation of the room acoustics.

Testing. At test time, we wish to re-synthesize the entire clean waveform instead of a single fixed-length segment. In this case, we feed all of the segments for a waveform S_r into the model and temporally concatenate all of the output segments. Because consecutive segments overlap by 50%, during the concatenation step we only retain the middle 50% of the frames from each segment and discard the rest. Finally, to re-synthesize the waveform we use the Griffin-Lim algorithm [106] to iteratively improve the predicted phase for 30 iterations, which we find works better than directly using the predicted phase or using Griffin-Lim with a randomly initialized phase.

6.4 Experiments

We evaluate our model by dereverberating speech for three downstream tasks: speech enhancement (SE), automatic speech recognition (ASR), and speaker verification (SV). We evaluate using both real scanned Matterport3D environments with simulated audio as well as real-world data collected with a camera and mic.

Evaluation tasks and metrics. We report the standard metrics Perceptual Evaluation of Speech Quality (PESQ) [237], Word Error Rate (WER), and Equal Error Rate (EER) for the three tasks, respectively. For ASR and SV, we use pretrained models from the SpeechBrain [233] toolkit. We evaluate these models off-the-shelf on our (de)reverberated version of the LibriSpeech test-clean set, and also explore fine-tuning the model on the (de)reverberated LibriSpeech train-clean-360 data to ensure all models have exposure to reverberant speech when training. For speaker verification, we construct a set of 80k sampled utterance pairs consisting of different rooms, mic placements, and genders to account for session variability, similar to [236].

Baseline models. In addition to evaluating the the clean and reverberant audio (with no enhancement), we compare against multiple baseline dereverberation models:

1. MetricGAN+ [78]: a recently proposed state-of-the-art model for speech enhancement; we use the public implementation from SpeechBrain [233], trained on our dataset. Following the original paper, we optimize for PESQ during training, then choose the best-performing model snapshot (on the validation data) specific to each of our downstream tasks.

2. HiFi-GAN [267]: a recent model for denoising and dereverberation. We use the public implementation. *

3. WPE [205]: A statistical speech dereverberation model that is commonly used for comparison.

*https://github.com/rishikksh20/hifigan-denoiser

	SE	*ASR*		*SV*	
	PESQ	WER	FT	EER	FT
Anechoic (Upper bound)	4.64	2.50	2.50	1.62	1.62
Reverberant	1.54	8.86	4.62	4.69	4.57
MetricGAN+ [78]	2.33	7.49	4.86	4.67	2.75
HiFi-GAN [267]	1.83	9.31	5.59	4.30	2.49
WPE [205]	1.63	8.18	4.30	5.19	4.48
VIDA w/o VAN	2.32	4.92	3.76	4.67	2.61
VIDA w/ normal FoV	2.33	4.85	3.73	4.53	2.79
VIDA w/o matching loss	**2.38**	4.59	3.72	4.02	2.62
VIDA w/o human mesh	2.31	4.57	3.72	4.00	2.52
VIDA w/ random image	2.34	4.94	3.82	4.70	2.48
VIDA	2.37	**4.44**	**3.66**	**3.97**	**2.40**

Table 6.1: Results on multiple speech analysis tasks, evaluated on the LibriSpeech test-clean set that is reverberated with our environmental simulator (with the exception of the "Anechoic (Upper bound)" setting, which is evaluated on the original audio). FT refers to tests where the models are finetuned with the audio-enhanced data. The relative improvement compared to Reverberant is included in parentheses.

We emphasize that all baselines are *audio-only* models, as opposed to our proposed *audio-visual* model. Our multimodal dereverberation technique could extend to work in conjunction with other newly-proposed audio-only models, i.e., ongoing architecture advances are orthogonal to our idea.

Results on SoundSpaces-Speech. Table 6.1 shows the results for all models on SE, ASR, and SV. First, since existing methods report results on anechoic audio, we note the pretrained SpeechBrain model applied to anechoic audio (first row) yields errors competitive with the SoTA [107], meaning we have a solid experimental testbed. Comparing the results on anechoic vs. reverberated speech, we see that reverberation significantly degrades performance on all tasks. Our VIDA model outperforms all other models, and by a large margin on the ASR and SV tasks.without finetuning, we achieve absolute improvements of 0.04 PESQ (1.71% relative improvement), 0.48% WER (9.75% relative improvement), and 0.68% EER (14.56% relative improvement)

over the *best baseline* in each case (which happens to be the audio-only version of VIDA for both the ASR and SV tasks). The results are statistically significant according to a paired t-test. After finetuning the ASR model, the gain is still largely preserved at 0.64% WER (14.88% relative), although it is important to note that finetuning downstream models on enhanced speech is not always feasible, e.g., if using off-the-shelf ASR. Our results demonstrate that learning the acoustic propertiesof an environment from visual signals is very helpful for dereverberating speech, enabling the model to leverage information unavailable in the audio alone.

Ablations. To study how much VIDA leverages visual signals, we ablate the visual network VAN (audio-only). Table 6.1 shows the results. All performance degrades significantly, showing that visual acoustic features are helpful for dereveberation. To understand how well VIDA works with a normal field-of-view (FoV) camera, we replace the panorama image input with a FoV of 80 degrees randomly sampled from the current view. All metrics drop compared to using a panorama, as expected. This is expected, because the model is limited in what it can see with a narrower field of view; the inferred room acoustics are impaired by not seeing the full environment or missing where the speaker is. Compared to the audio-only ablation, however, VIDA still performs better; even a partial view of the environment helps the model understand the scene and dereverberate the audio. Next, we ablate the proposed reverb-visual matching loss ("w/o matching loss"). Without it, VIDA's performance declines on all metrics. This shows by forcing the visual feature to agree with the reverberation feature, our model learns a better representation of room acoustics. To examine how much the model leverages the human speaker cues and uses the visual scene, we evaluate VIDA on the same test data but with the 3D humanoid removed ("w/o human mesh") or train VIDA with random images ("w/ random image") and re-evaluate. All three metrics become worse. This shows our model pays attention to both the presence of the human speaker and the scene geometry to better anticipate reverberation.

Results on real data. Next, we deploy our model in the real world. We use

	SE PESQ	*ASR* WER	*SV* EER
Anechoic (Upper bound)	4.64	2.52	1.42
Reverberant	1.22	18.39	3.91
MetricGAN+ [78]	**1.62**	21.42	5.70
HiFi-GAN [267]	1.33	24.05	5.21
VIDA w/o VAN	1.41	15.18	4.24
VIDA w/ normal FoV	1.44	14.71	3.79
VIDA	1.49	**13.02**	**3.75**

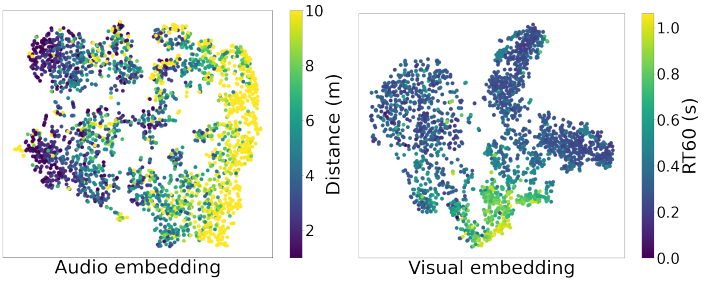
Table 6.2: Results on real data demonstrating sim2real transfer.

	Atrium	Conf. Room	Classroom	Corridor
Near-field	14.1 / **9.0**	**5.0** / 6.5	6.1 / **5.3**	2.2 / **1.8**
Mid-field	21.8 / **18.9**	**7.7** / **7.7**	2.6 / **1.5**	7.3 / **4.4**
Far-field	52.4 / **50.5**	22.0 / **6.7**	**5.9** / 6.8	25.2 / **21.1**

Table 6.3: Breakdown of word error rate (WER) for VIDA without and with VAN on real test data.

all models trained in simulation to dereverberate the real-world dataset (cf. Sec. 6.2) before using the finetuned ASR/SV models to evaluate the enhanced speech. Table 6.2 shows the results of all models on real data. Reverberation does more damage to the WER compared to in simulation. Although MetricGAN+ [78] has better PESQ, it has a weak WER score. Our VIDA model again outperforms all baselines on ASR and SV. This demonstrates the realism of the simulation and the capability of our model to transfer to real-world data, a promising step for VIDA's wider applicability.

Table 6.3 breaks down the ASR performance for VIDA without and with VAN by environment type and speaker distance. The atrium is quite reverberant due to the large space.Although the auditorium is similarly large, the space is designed to reduce reverberation and thus both models have lower WER. The conference room and the classroom have smaller sizes and are comparatively less reverberant. The corridor only becomes reverberant when the speaker is far away. VIDA outperforms



(a) Audio t-SNE (b) Visual t-SNE

Figure 6.5: t-SNE of audio and visual features colored by the distance to the speaker (c) and RT60 (d).

VIDA w/o VAN in most cases, especially in highly reverberant ones.

Analyzing learned features. Figure 6.5a and 6.5b analyze our model's learned audio and visual features via 2D t-SNE projections [289]. For each sample, we color the point according to either (c) the ground truth distance between the camera/microphone and the human speaker or (d) the reverberation time for the audio signal to decay by 60 dB (known as the RT60). Neither of these variables are available to our model during training, yet when learning to perform deverberation, our model exposes these high-level properties relevant to the audio-visual task. Consistent with the quantitative results above, this analysis shows how our model captures elements of the visual scene, room geometry, and speaker location that are valuable to proper dereverberation.

Qualitative examples. Figure 6.6 shows a simulated and real-world example. As we can see, the reverberant spectrogram is much blurrier compared to the clean spectrogram, while our predicted spectrogram removes those reverberations by leveraging the visual cues of room acoustics.

6.5 Conclusions

In this chapter, we introduced the novel task of audio-visual dereverberation. The proposed VIDA approach learns to remove reverb by attending to both the audio

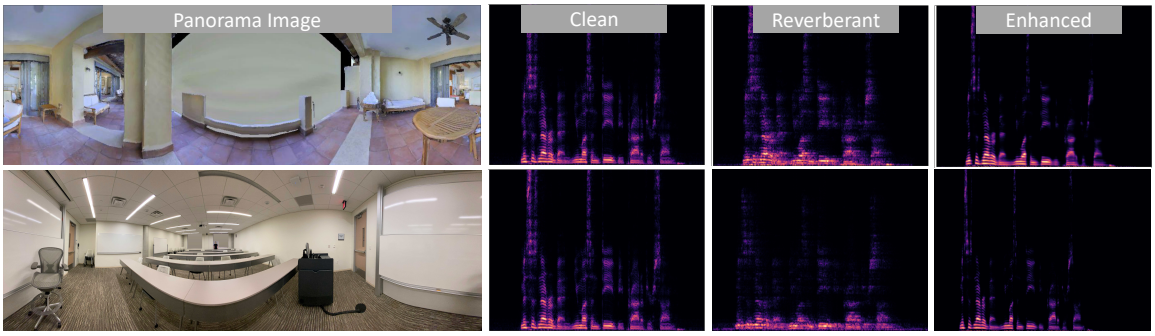


Figure 6.6: Example input images, clean spectrograms, reverberant spectograms and spectrograms dereverberated by VIDA (top is from a scan, bottom is a real pano). The speaker is out of view in the first case and distant in the second case (back of the classroom). Though both received audio inputs are quite reverberant, our model successfully removes the reverb and restores the clean source speech.

and visual streams, recovering valuable signals about room geometry, materials, and speaker locations from visual encodings of the environment. In support of this task, we develop a large-scale dataset providing realistic, spatially registered observations of speech and 3D environments. VIDA successfully dereverberates novel voices in novel environments more accurately than an array of baselines, improving multiple downstream tasks.

One of the limitations of this work is the reliance on the large amount of paired clean/reverberated audio along with images. This could be difficult to collect in the real world because usually cameras only record the sound as the receiver, while the source audio is often not captured. In this chapter, we rely on simulating the room acoustics and corresponding images to avoid this problem. If we were to train on real-world data for better performance on the real data, one possible solution is pre-train the visual encoder with audio in a self-supervised way so that the visual encoder already captures room acoustics features.

While VIDA outperforms audio-only baselines on an array of tasks, especially ASR, perceptually, the difference is not significant. If the input audio has lots of reverberation, the output audio often contains some distortion potentially because

the input and output audios are very different, and L2 loss alone is not enough to remove the excessive amount of reverberation. Both observations indicate more research is needed to improve the perceptual quality in addition to improving machine perception on various tasks by, for example, incorporating generative losses.

In this chapter, I showed how to infer room acoustics from visual observations of the scene and use it to remove reverberation in the audio. One question someone might ask is: is the opposite of this task possible, i.e., adding reverberation to audio based on the visual information? Yes. It is possible, and in the next chapter, I will show how to transform an audio clip to match the acoustics of an environment specified in an image.

Chapter 7: Visual Acoustic Matching

In Chapter 6, I demonstrated how to leverage the visual knowledge to help remove reverberation in speech for better perception. However, sometimes the reverse process is more desired, i.e., adding proper reverberation to the audio that corresponds to the space. In this chapter, I will present a work that deals with that, which was published in CVPR 2022 [39].

As discussed throughout this thesis, the audio we hear is always transformed by the space we are in, as a function of the physical environment's geometry, the materials of surfaces and objects in it, and the locations of sound sources around us. This means that we perceive the same sound differently depending on where we hear it. For example, imagine a person singing a song while standing on the hardwood stage in a spacious auditorium versus in a cozy living room with shaggy carpet. The underlying song content would be identical, but we would experience it in two very different ways.

For this reason, it is important to model room acoustics to deliver a realistic and immersive experience for many applications in augmented reality (AR) and virtual reality (VR). Hearing sounds with acoustics *inconsistent* with the scene is disruptive for human perception. In AR/VR, when the real space and virtually reproduced space have different acoustic properties, it causes a cognitive mismatch, and the "room divergence effect" damages the user experience [300].

Creating audio signals that are consistent with an environment has a long history in the audio community. If the geometry (often in the form of a 3D mesh) and material properties of the space are known, simulation techniques can be applied to generate a room impulse response (see Chapter 3 for more details). In the absence of geometry and material information, the acoustical properties can be estimated blindly from audio captured in that room (e.g., reverberant speech), then used to auralize

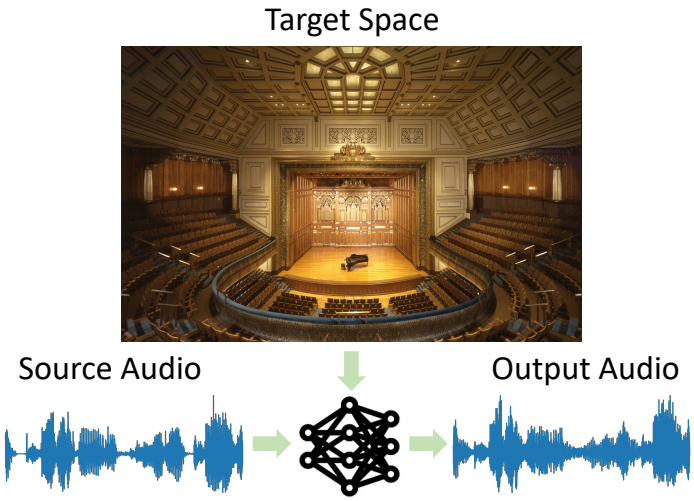


Figure 7.1: Goal of visual acoustic matching: transform the sound recorded in one space to another space depicted in the target visual scene. For example, given source audio recorded in a studio, re-synthesize that audio to match the room acoustics of a concert hall.

a signal [145, 194, 266]. However, both approaches have practical limitations: the former requires access to the full mesh and material properties of the target space, while the latter gets only limited acoustic information about the target space from the reverberation in the audio sample. Neither uses imagery of the target scene to perform acoustic matching.

We propose a novel task: *visual acoustic matching*. Given an image of the target environment and a source audio clip, the goal is to re-synthesize the audio as if it were recorded in the target environment (see Figure 7.1). The idea is to transform sounds from one space to another space by altering their scene-driven acoustic signatures. Visual acoustic matching has many potential applications, including smart video editing where a user can inject sounding objects into new backgrounds, film dubbing to make a different actor's voice sound appropriate for the movie scene, audio enhancement for video conference calls, and audio synthesis for AR/VR to make users feel immersed in the visual space displayed to them.

To address visual acoustic matching, we introduce a crossmodal transformer

model together with a novel self-supervised training objective that accommodates in-the-wild Web videos having unknown room acoustics.

Our approach accounts for two key challenges: how to faithfully model the complex crossmodal interactions, and how to achieve scalable training data. Regarding the first challenge, different regions of a room affect the acoustics in different ways. For example, reflective glass leads to longer reverberation in high frequencies while absorptive ceilings reduce the reverberation more quickly. Our model provides fine-grained audio-visual reasoning by attending to regions of the image and how they affect the acoustics. Furthermore, to capture the fine details of reverberation effects—which are typically much smaller in magnitude than the direct signal—we use 1D convolutions to generate time-domain signals directly and apply a multi-resolution generative adversarial audio loss.

Regarding the second key challenge, one would ideally have *paired* training data consisting of a sound sample not recorded in the target space plus its proper acoustic rendering for the scene shown in the target image, i.e., a source and target audio for each visual scene in the training set. However, such a strategy requires either physical access to the pictured environments, or knowledge of their room impulse response functions—either of which severely limits the source of viable training data. Meanwhile, though a Web video does exhibit strong correspondence between its visual scene and the scene acoustics, it offers only the audio recorded in the target space. Accounting for these tradeoffs, we propose a self-supervised objective that automatically creates acoustically mismatched audio for training with Web videos. The key insight is to use dereverberation and acoustic randomization to alter the original audio's acoustics while preserving its content.

We demonstrate our approach on challenging real-world sounds and environments, as well as controlled experiments with realistic acoustic simulations in scanned scenes. Our quantitative results and subjective evaluations via human studies show that our model generates audio that matches the target environment with high per-

ceptual quality, outperforming a state-of-the-art model that has heavier supervision requirements [257] as well as traditional acoustic matching models.

I first introduce the visual acoustic matching task in Sec. 7.1, the dataset in Sec. 7.2, our approach in Sec. 7.3 and lastly the results in Sec. 7.4.

7.1 The Visual Acoustic Matching Task

We introduce a novel task, *visual acoustic matching*. In this task, an audio recording A_S recorded in space S and an image I_T of a different target space T are provided as input. The goal is to predict A_T, which has the same audio content as A_S but sounds as if it were recorded in space T with a microphone co-located with I_T's camera. Our goal is thus to learn a function f such that $f(A_S, I_T) = A_T$. The microphone co-location is important because acoustic properties vary as the listener location changes; inconsistent camera locations would lead to a perceived mismatch between the visuals and acoustics. The space S can have arbitrary acoustic characteristics, from an anechoic recording studio to a concert hall with significant reverberation. We assume there is one sounding object, leaving the handling of background sounds or interference as future work.

Importantly, our task formulation does *not* assume access to the impulse response, nor does it require the input audio to be anechoic. In comparison, the Image2Reverb [257] task requires access to both the impulse response and clean input audio, and does not account for the co-location of the camera and microphone.

7.2 Datasets

We consider two datasets: simulated audio in scanned real-world environments (Sec. 7.2.1), and in-the-wild Web videos with their recorded audio (Sec. 7.2.2). The former has the advantage of clean paired training data for A_T and A_S as well as precise ground truth for evaluating the output audio, but necessarily has a realism

(a)  (b) 

Figure 7.2: Example images in (a) SoundSpaces and (b) AVSpeech.

gap. The latter has the advantage of total realism, but makes quantitative evaluation more complex.

For both, we focus on human speech in indoor settings given its relevance to many of the applications cited above, and due to the fact that human listeners have strong prior knowledge about how reverberation should affect speech. However, our model design is not specific to speech.

7.2.1 SoundSpaces-Speech Dataset

With the SoundSpaces platform [35], acoustics can be accurately simulated based on 3D scans of real-world environments [29, 265, 312]. This allows highly realistic rendering of arbitrary camera views and arbitrary microphone placements for waveforms of the user's choosing, accounting for all major real-world audio factors: direct sounds, early specular/diffuse reflections, reverberation, binaural spatialization, and effects from materials and air absorption.

We adopt a SoundSpaces-Speech dataset created in [42] consisting of paired clean (anechoic) and reverberant audio samples together with camera views.* The RIRs for 82 Matterport3D [29] environments are convolved with non-overlapping speech clips from LibriSpeech [214]. A 3D humanoid of the same gender as the real speaker is inserted at the speaker location and panorama RGB-D images are rendered at the listener location. See Figure 7.2a. Excluding those samples where the speaker is

*Note that [42] uses the data for dereverberation, not acoustic matching.

very distant or out-of-view (for which the visual input does not capture the geometry of the source location), there are 28,853/1,441/1,489 samples for the train/val/test splits.

7.2.2 Acoustic AVSpeech Web Videos

Web videos offer rich and natural supervision for the association between visuals and acoustics. We adopt a subset of the AVSpeech [70] dataset, which contains 3-10 second YouTube clips from 290k videos of single (visible) human speakers without interfering background noises. We automatically filter the full dataset down to those clips likely to meet our problem formulation criteria: 1) microphone and camera should be co-located and at a position different than the sound source (so that the audio contains not only the source speech but also the reverberation caused by the environment), and 2) audio recording should be reverberant (so that the physical space has influenced the audio). Cameras in this dataset are typically static, and thus we use single frames and their corresponding audio for this task. This yields 113k/3k/3k video clips for train/val/test splits. We refer to this filtered dataset as Acoustic AVSpeech. See Figure 7.2b.

7.3 Approach

We present the **A**udio-**Vi**sual **T**ransformer for **A**udio Gene**r**ation model (**AViTAR**) (Figure 7.3). AViTAR learns to perform crossmodal attention based on sequences of convolutional features of audio and images and then synthesizes the desired waveform $\hat{A_T}$. We first define the audio-visual features (Sec. 7.3.1) and their crossmodal attention (Sec. 7.3.2), followed by our approach to waveform generation (Sec. 7.3.3). Finally, we present our acoustics alteration idea to enable learning from in-the-wild video (Sec. 7.3.4).

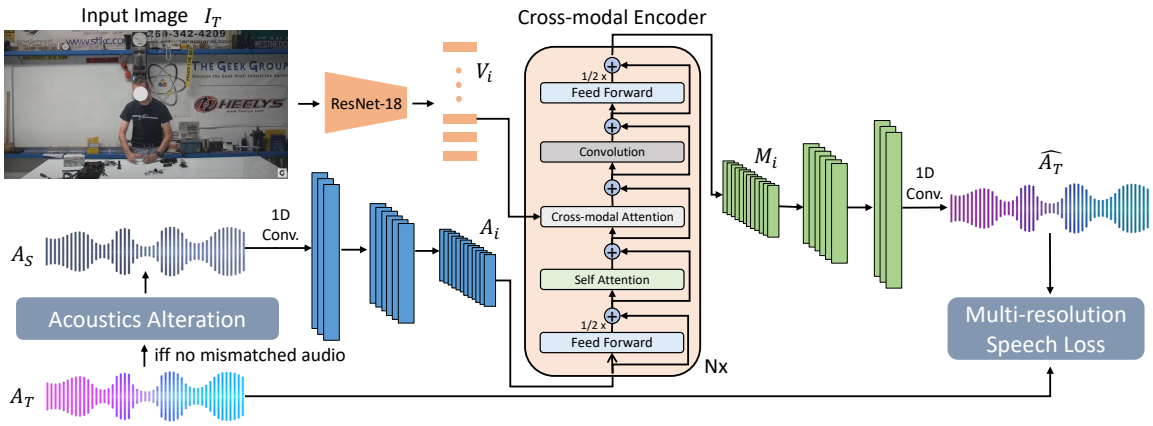


Figure 7.3: AViTAR model illustration. We extract visual feature sequence V_i from input image I_T with a ResNet-18 [115], and audio feature sequence A_i from input audio A_S with 1D convolutions. V_i and A_i are passed into crossmodal encoders for crossmodal reasoning. The output feature sequence M_i is processed and upsampled with 1D convolutions to recover the output of the same temporal length. Finally, we use a multi-resolution speech GAN loss to guide the audio synthesis to be high fidelity. The acoustics alteration process is applied to the target audio during training if and only if there is no mismatched audio, e.g., on the Acoustic AVSpeech dataset.

7.3.1 Audio-Visual Feature Sequence Generation

To apply crossmodal attention, we first need to generate sequences of audio and visual features, where each element in the sequence represents features of a part of the input space. For visual sequence generation from image I_T, we use ResNet18 [115] and flatten the last feature map before the pooling layer, yielding the visual feature sequence V_i.

For audio feature sequence generation from source audio A_S, we generate audio features A_i from the waveform directly with stacked 1D convolutions. We first use one 1D conv layer to embed the input waveform into a latent space. We then apply a sequence of strided 1D convolutions, each doubling the channel size while downsampling the input sequence. The output audio features are a sequence of vectors of size S, with length downsampled D times from the input. Weight normalization is applied to 1D conv layers. We rather than STFT spectrograms so that the audio features are not limited to one resolution and can be optimized end-to-end to learn

the most important features for the visual acoustic matching task.

7.3.2 Crossmodal Encoder

Prior work often models audio-visual inputs in a simplistic manner by representing the image feature with one single vector and concatenating it with the audio feature [211, 86, 70, 323, 88, 42, 35]. However, for visual acoustic matching, it is important to reason how different regions of the space contribute to the acoustics differently. For example, a highly reflective glass door leads to longer reverberation time for high frequencies, while absorptive ceilings diminish that quickly. Thus, we propose to attend to image regions to reason how different image patches contribute to the acoustics, leveraging recent advances on the transformer architecture [290, 149, 107].

For crossmodal attention, we first adopt the conformer variant [107] of encoder blocks, which adds one convolution layer inside the block for modeling local interaction for speech features. Based on this block, we insert one crossmodal attention layer $\mathcal{A}_{cm}$ after the first feed-forward layer, described as follows:

$$\mathcal{A}_{cm}(A_i, V_i) = \text{softmax}(\frac{A_i V_i^T}{\sqrt{S}})V_i, \tag{7.1}$$

where the attention scores between the two sequences of features A_i and V_i are first calculated by dot-product, then normalized by softmax, scaled by $\frac{1}{\sqrt{S}}$, and finally used to weight the visual features V_i. This crossmodal attention allows the model to attend to different image region features and reason about how they affect the reverberation. Absolute positional encoding is added to the visual encoding. After passing V_i and A_i through N encoder blocks, we obtain the fused audio-visual feature sequence M_i, which has the same length as A_i.

7.3.3 Waveform Generation and Loss

Recent audio-visual work generates audio outputs by inferring spectrograms then using ISTFT reconstruction to obtain a waveform (e.g., [316, 88, 70, 86, 323,

322]). While sensible for source separation, where the target signal is a subset of the source signal, ratio mask prediction is inadequate for our task, because reverberation might occupy periods of silence in the input audio and the ratio will be unbounded (as we verify in results). Futhermore, generating audio based on spectrograms is limiting because 1) predicting the coherent phase component remains challenging [1, 51], and 2) the spectrogram has one fixed resolution (one FFT size, hop length, and window size).

Instead, we aim to synthesize time-domain signals directly, skipping the intermediate spectrogram generation step and allowing more flexibility for what losses can be imposed, inspired by recent advances on time-domain speech synthesis [287, 222, 151, 157]. Specifically, with the fused audio-visual feature sequence M_i, we apply a sequence of transposed strided 1D convolutions, each halving the channel size while upsampling the input sequence, which is exactly the reverse operation of the audio encoding. Altogether, we upsample the audio sequence D times and obtain a waveform of the same length as the input.

Next we incorporate a multi-resolution generative loss. We found directly minimizing a Euclidean distance based loss between the target ground truth audio A_T and the inferred audio $\hat{A_T}$ leads to distortion in the generated audio on this task (cf. Figure 7.5 and Tab. 5.3). Therefore, to let the model learn how to reverberate the input speech properly, we employ a generative adversarial loss where a set of discriminators operating at different resolutions are trained to identify reverberation patterns and guide the generated audio to sound like real examples. Specifically, we apply an adversarial loss [151] comprised of the generator and discriminator losses:

$$\mathcal{L}_G = \sum_{k=1}^{K} (\mathcal{L}_{Adv}(G; D_k) + \lambda_1 \mathcal{L}_{FM}(G; D_k)) + \lambda_2 \mathcal{L}_{Mel}(G),$$

$$\mathcal{L}_D = \sum_{k=1}^{K} \mathcal{L}_{Adv}(D_k; G),$$

where each D_k is a sub-discriminator that operates at one of K different scales and periods for distinguishing the fake and real examples. $\mathcal{L}_{Adv}$ is the LS-GAN [180]

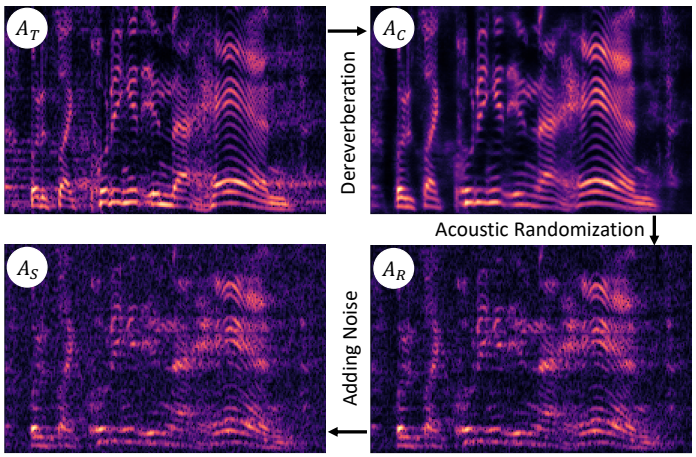


Figure 7.4: Acoustics alteration process. Spectrograms of the resulting audio after each step are shown. We first dereverberate the target audio A_T to obtain cleaner audio A_C, randomize its acoustics by applying an impulse response of another environment to obtain A_R, and finally, add Gaussian noise to A_R to create A_S. Notice how the spectral pattern changes in this process.

training objective, which trains the generator to fake the discriminator and trains the discriminator to distinguish real examples from fake ones. For the generator G, a feature matching loss [157] $\mathcal{L}_{FM}$ is used, which is a learned similarity metric measured by the difference in features of the discriminator between a ground truth sample and a generated sample. An additional mel-spectrogram loss $\mathcal{L}_{Mel}$ is imposed on the generator for improving the training efficiency and fidelity of the generated audio. λ_1 and λ_2 are two weighting factors for these two losses. The generator loss $\mathcal{L}_G$ and discriminator loss $\mathcal{L}_D$ are trained alternatively competing against each other. For more details, refer to [151].

7.3.4 Acoustics Alteration for Self-Supervision

The training paradigm differs in one important way depending on the source of training data (cf. Sec. 7.2). For the simulated SoundSpaces data, we have access to an anechoic audio sample A_S as well as the ground truth reverberated sample A_T as it should be rendered in the target environment for a camera seeing view I_T. This

means we can train to (implicitly) discover the mapping that takes the target image to an RIR which, when convolved with A_S, yields A_T.

For the in-the-wild video data (AVSpeech), however, we have only A_T and I_T to train, i.e., we only observe sounds that *do* match their respective views. Thus, to leverage unannotated Web video, we need to create an audio clip that preserves the target audio content but has *mismatched* acoustics. Figure 7.4 illustrates the steps for this process. First we strip away the original acoustics of the target environment by performing dereverberation on the audio A_T alone with the pretrained model from [42]. Since dereverberation is imperfect, there is residual acoustic information in the dereverberated output A_C, meaning that the resulting "clean" audio is still predictive of the target environment.

Thus, we subsequently randomize the acoustics by convolving that audio with an impulse response of another environment, yielding A_R; that IR is randomly chosen from the corresponding train/val/test split of SoundSpaces-Speech. The idea is to transform the semi-clean intermediate sound into another space to create more acoustic confusion, thereby forcing the model to learn from the target image. Finally, to further suppress the residual acoustics from the training environment, we add Gaussian noise with SNR randomly sampled from 2-10 dB to A_R and obtain the training source audio A_S. In short, with this strategy, we are able to leverage readily available Web videos for our proposed task, despite its lack of ground truth paired audio.

7.4 Experiments

We validate our model on two datasets using comprehensive metrics and baselines.

Evaluation metrics. We measure the quality of the generated audio from three aspects: 1) the closeness to the ground truth (if ground truth audio is available), as measured by **STFT Distance**, i.e., the MSE between the generated and true

	SoundSpaces-Speech						*Acoustic AVSpeech*			
	Seen			*Unseen*			*Seen*		*Unseen*	
	STFT	RTE	MOSE	STFT	RTE	MOSE	RTE	MOSE	RTE	MOSE
Input audio	1.192	0.331	0.617	1.206	0.356	0.611	0.387	0.658	0.392	0.634
Blind Reverb. [295]	1.338	0.044	0.312	-	-	-	-	-	-	-
Image2Reverb [257]	2.538	0.293	0.508	2.318	0.317	0.518	-	-	-	-
AV U-Net [86]	**0.638**	0.095	0.353	**0.658**	0.118	0.367	0.156	0.570	0.188	0.540
AViTAR w/o visual	0.862	0.140	0.217	0.902	0.186	0.236	0.194	0.504	0.207	0.478
AViTAR	0.665	**0.034**	**0.161**	0.822	**0.062**	**0.195**	**0.144**	**0.481**	**0.183**	**0.453**

Table 7.1: Results on the SoundSpaces-Speech and Acoustic AVSpeech datasets for Seen and Unseen environments. All input audio at test time is novel (unheard during training). Note that the STFT metric is applicable only for SoundSpaces, where we can access the ground truth A_T's spectrogram. For all metrics, lower values are better. Standard errors for STFT, RTE and MOSE are all less than 0.04, 0.013s and 0.01 on SoundSpaces-Speech. Standard errors for RTE and MOSE are all less than 0.005s and 0.01 on Acoustic AVSpeech.

target audio's magnitude spectrograms; 2) the correctness of the room acoustics, as measured by the **RT60 Error (RTE)** between the true and inferred A_T's RT60 values. RT60 indicates the reverberation time in seconds for the audio signal to decay by 60 dB, a standard metric to characterize room acoustics. We estimate the RT60 directly from magnitude spectrograms of the output audio, using a model trained with disjoint SoundSpaces data, since impulse responses are not available for the target environments; and 3) the speech quality preserved in the synthesized speech, measured by the **Mean Opinion Score Error (MOSE)**, which is the difference in speech quality between the true target audio and generated audio, as assessed by a deep learning based objective model MOSNet [170].[†]

Both the RTE and MOSE metrics are content-invariant and thus useful for evaluation when only audio with correct acoustics and mismatched content is available as ground truth, i.e., Web videos.

[†]By taking the difference with the true target audio's MOS score (rather than simply the output's score), we account for the fact that properly reverberated speech need not have high speech quality.

In addition, we conduct user studies to evaluate whether a given audio is perceived as matching the room acoustics of the reference image.

Seen and unseen environments. On both datasets, we evaluate by pairing the source audio A_S with a target image I_T coming from either the training set (*Seen*) or test set (*Unseen*). The audio is always unobserved in training. The Seen case is useful to match the audio to scenes where we have video recordings (e.g., the film dubbing case). The Unseen case is important for injecting room acoustics depicted in novel images (e.g., to match sounds for a random Web photo being used as a Zoom call background).

Baselines. We consider the following baselines:

1. **Input audio**. This is the naive baseline that does nothing, simply returning the input A_S as output.
2. **Blind Reverberator**. This is a traditional acoustic matching approach [295] using audio recorded in the target space T as reference with content different from A_T. It first estimates RT60 and DRR from the reference audio (estimators are trained using simulated IRs), and then synthesizes the target IR by shaping an exponentially decaying white noise based on those two parameters. Unlike our model, this method requires reference audio at test time and IRs at training time. It is therefore inapplicable for the Unseen case (no reference audio) and AVSpeech (no training IRs).
3. **Image2Reverb [257]**. This is a recent approach that trains an IR predictor from images, then convolves the predicted IRs with A_S to obtain the target audio. This model requires access to the IR during training and thus is not applicable to the Acoustic AVSpeech dataset. We use the authors' code and convert the SoundSpaces-Speech data into the format of their dataset. We replace their depth prediction model with the ground truth depth image, to improve this baseline's performance.

4. **AV U-Net [86]**. This is an audio-visual model originally proposed for visually guided spatial sound generation based on a U-Net network for processing audio spectrograms. We adapt it for visual acoustic matching by removing the ratio mask prediction (which we find does not work well). Instead, we feed in a magnitude spectrogram, predict the target magnitude spectrograms, and generate the time-domain signals with Griffin Lim [106]. This baseline helps isolate the impact of our proposed crossmodal attention architecture compared to the common U-Net approach [86, 211, 88, 51, 322].
5. **AViTAR w/o visual**. This model is solely audio-based and is the same as our proposed model except that it does not have visual inputs or the crossmodal attention layer.

7.4.1 Results on SoundSpaces-Speech

For the SoundSpaces data, we have access to clean anechoic speech, which we use as the input A_S. The simulations offer a clean testbed for this task, showing the potential of each model when it is noise-free and the visuals reveal the full geometry via the panoramic RGB-D images.

Table 7.1 (left) shows the results. As expected, the clean input audio baseline does poorly because it does not account for the target environment. Our AViTAR model has the lowest RT60 error and MOS error, indicating that it best predicts the correct acoustics from images, injects them into the speech, and synthesizes high-quality audio. The AV U-Net baseline has slightly lower STFT distance than ours, likely because its training objective is to minimize STFT distance. However it has higher perceptual errors (RTE and MOSE). Image2Reverb's [257] high errors reveal the difficulty of our task and data, and its inapplicability to AVSpeech highlights our model's self-supervised training advantage. Despite having the estimated RT60 as input (and thus having low RT60 error), Blind Reverberator's STFT and MOS errors are much higher than AViTAR's, showing that images are a promising way to

characterize room acoustics beyond the traditional RT60. Plus, its inapplicability for the other scenarios highlights fundamental advantages of AViTAR. Without access to visual information ("w/o visual"), AViTAR can only learn to add an average amount of reverberation to the input audio; this confirms that our model successfully learns the acoustics from the visual scene. Although this variant has higher RT60 error than AV U-Net, its MOS error is lower because the audio quality is better.

Ablations. Table 7.2 shows results for ablations on unseen images. For the model architecture, to understand if attending to different image regions with crossmodal attention is helpful, we train the full model with the length of visual feature sequence reduced to one by mean pooling the final ResNet feature map ("w/ pooled visual feature"). This model underperforms the full model on both STFT and RT60 metrics, showing that the audio-visual attention leads to a better visual understanding of room acoustics. Next we ablate the generative loss and replace it with the non-generative multi-resolution STFT loss [157] ("w/o generative loss"), which slightly improves the STFT error but leads to a large drop on the acoustics recovery and speech quality. Despite being multi-resolution, without learnable discriminators to learn to model those fine reverberation details, the audio quality gets worse.

The synthetic dataset provides access to meta information useful to evaluate whether and how much AViTAR reasons about different visual properties. The location of the sound source matters for acoustics because it directly influences acoustic characteristics like the direct-to-reverberant ratio (DRR). When we remove the 3D humanoid from the scene ("w/o human") in all test images, all error metrics increase, which indicates that our model reasons about the location of the sound source in the image for accurate acoustic matching. To understand if the model learns meaningful information from the visuals, we replace the target image with a random image ("w/ random image"); this significantly harms our model's performance.

AViTAR	STFT	RTE (s)	MOSE
Full model	0.822	**0.062**	0.195
w/ pooled visual feature	0.850	0.067	**0.193**
w/o generative loss	**0.777**	0.081	0.314
w/o human	0.884	0.139	0.218
w/ random image	0.940	0.236	0.250

Table 7.2: Ablations on model design and data.

7.4.2 Results on Acoustic AVSpeech

Next, we train our model on the in-the-wild AVSpeech videos, and test it on novel clean speech clips from LibriSpeech [214] (A_S) paired with target images (I_T) from AVSpeech. Here we do not have ground truth for the target speech, so we evaluate with RTE and MOSE.

Table 7.1 (right) shows the results. Our proposed AViTAR model achieves the lowest RT60 error compared to all baselines. This shows our model trained in its self-supervised fashion successfully generalizes to novel images and novel audio, and demonstrates we can do acoustic matching even for non-anechoic inputs. AViTAR's MOS error is also the lowest compared to all baselines, showing that it is able to synthesize high-fidelity audio while injecting the proper amount of reverberation into the speech. The absolute errors on AVSpeech are higher than on SoundSpaces, which makes sense because the YouTube imagery is more variable, and it has a narrower field of view and no depth, making the geometry and materials of the scene only partly visible.

Ablations on acoustic alteration. Table 7.3 shows ablations on the proposed acoustics alteration strategy. In short, all three steps are necessary to create an acoustic mismatch with the image, thereby forcing the model to recover the correct acoustics based on the image and allowing better generalization to novel sounds.

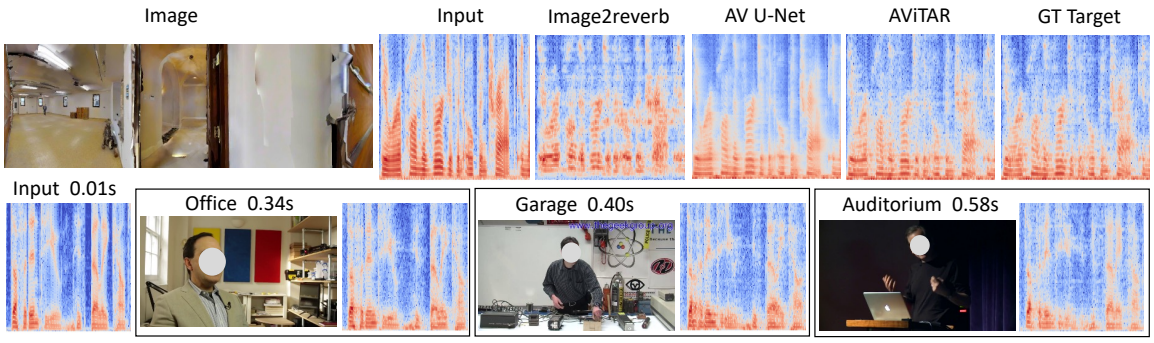


Figure 7.5: Qualitative predicted audio. For all audio clips, we compute the magnitude spectrogram, convert the magnitude to dB, and plot the spectrogram with x-axis spanning from 0 to 1.28 s (left to right) and y-axis from 0 to 3000 Hz (bottom to top). Row 1: SoundSpaces-Speech example where the target space is a large empty room with a lot of reverberation. Our model predicts the audio closest to the target clip. AV U-Net's spectrogram is too smoothed compared to ours and misses some fine reverb details, which leads to perceptual distortion. Row 2: examples on Acoustic AVSpeech (unseen images). We feed one clean audio clip to match three different scenarios (office, garage, auditorium). From left to right, the audio spectrogram becomes more reverberant as phoneme patterns get extended and blurred on the temporal axis (est. RT60 times shown). NB: AViTAR processes waveforms, not spectrograms; here they are for visualization.

User study. To supplement the quantitative metrics and directly capture the perceptual quality of the generated samples, we next conduct a user study. We show participants the image of the target environment I_T, the accompanying ground truth audio clip A_T as reference, and paired audio clips $\hat{A_T}$ generated by AViTAR and each baseline. We ask participants to select the clip that most sounds as if it were recorded in the target environment and best matches the reverberation in the given clip. We select 30 reverberant examples from SoundSpaces-Speech and AVSpeech and ask 30 participants to complete the assignment on MTurk.

Table 7.4 shows the resulting preference scores. Compared to each baseline, AViTAR is always preferred. Note that no participant has a background in acoustics, and some might simply pick the one that sounds "clean" rather than having the correct room acoustics. This may be the reason even the anechoic input has a

Acoustics Alteration	Seen	Unseen
Dereverb. + Randomization + Noise	**0.144**	**0.183**
Dereverb. + Randomization	0.178	0.197
Dereverb. + Noise	0.170	0.208
Dereverb.	0.230	0.250
A_T + Randomization + Noise	0.236	0.249

Table 7.3: Ablations on acoustics alteration. RTE is reported.

	SoundSpaces	AVSpeech
Input Speech	42.1% / **57.9**%	40.1% / **59.9**%
Image2Reverb [257]	25.9% / **74.1**%	- / -
AV U-Net [86]	29.8% / **70.2**%	27.2% / **72.8**%
AViTAR w/o visual	39.6% / **60.4**%	46.3% / **53.9**%

Table 7.4: User study results. X%/Y% indicates among all paired examples for this baseline and AViTAR, X% of participants prefer this baseline while **Y%** prefer AViTAR.

higher preference score than the U-Net model. Despite the lack of domain knowledge, participants still consistently favor our model over other baselines.

Qualitative examples. Figure 7.5 shows example outputs.

7.5 Conclusions

We proposed the visual acoustic matching task and introduced the first model to address it. Given an image and audio clip, our method injects realistic room acoustics to match the target environment. Our results validate their realism with both objective and perceptual measures. Importantly, the proposed model is trainable with unannotated, in-the-wild Web videos.

To leverage Web videos, I proposed the acoustic alteration strategy to create self-supervision. The first step in this process is dereverberation with an off-the-shelf model that is not perfect. To overcome the residual acoustics, I further randomize

the acoustics and add noise. However, the last two steps inject biases in their own ways and affect the generalization. One possible solution to this is to train the dereverberation model and acoustic matching model at the same time so that they can jointly optimize [260].

Another limitation of this work is that the proposed AViTAR model does audio processing, audio-visual crossmodal reasoning, and audio generation together. While this end-to-end approach makes modeling easier, it requires more computation resources and a longer time to train the model. One way to decouple the crossmodal reasoning and generation is to break down the pipeline into two stages. In the first stage, one model generates intermediate audio representation without caring about the quality, and in the next stage, another model generates audio with high fidelity from this representation. This approach has its own tradeoff as well, that is the potential risk in accumulating errors from the multi-stage learning.

For this task, evaluation is also non-trivial. Measuring the acoustic properties of in-the-wild data (YouTube speech videos) is an open challenge due to the lack of robust solutions. To deal with that, I trained an RT60 predictor in simulation and evaluated it on the real data, which unavoidably suffered from the sim2real gap. Human evaluation is a more desired measurement, but it is not scalable. In the future, metrics for approximating human perception are needed to improve the evaluation.

In this chapter, I demonstrated how to transform audio clips to match the acoustics of spaces from images, assuming the visuals of the target environment or location are given. In some applications, that might be the case. For example, sometimes, we might desire to generate sounds from a reference image plus a pose offset without having access to the target visuals directly. In the next chapter, I will show how to transform the sound from one viewpoint to another in the same space.

Chapter 8: Novel-View Acoustic Synthesis

In Chapter 6 and Chapter 7, I discussed how we could either remove reverberation or add reverberation based on visuals, where input visuals characterize views at the input viewpoint. In some applications, the target locations might not be given directly, instead, we might have visuals from a source viewpoint and the relative change in camera poses as inputs. This is often the case for reconstructing novel views from a reference view. In this chapter, I will present the acoustic version of this task, that is, given the audio-visual observations of a source viewpoint, how do we synthesize the sound of another viewpoint in the same space? This work was published at CVPR 2023 [41].

Replaying a video recording from a new viewpoint* has many applications in cinematography, video enhancement, and virtual reality. For example, it can be used to edit a video, simulate a virtual camera, or, given a video of a personal memory, even enable users to experience a treasured moment again—not just on a 2D screen, but in 3D in a virtual or augmented reality, thus 'reliving' the moment.

While the applications are exciting, there are still many unsolved technical challenges. Recent advances in 3D reconstruction and novel-view synthesis (NVS) address the problem of synthesizing new *images* of a given scene [186, 182, 224]. However, thus far, the view synthesis problem is concerned with creating visuals alone; the output is silent or at best naively adopts the sounds of the original video (from the "wrong" viewpoint). Without sound, the emotional and cognitive significance of the replay is severely diminished.

In this work, we address this gap and introduce the new task of *novel-view acoustic synthesis* (NVAS). The goal of this task is to synthesize the sound in a scene

*We use "viewpoint" to mean a camera or microphone pose.

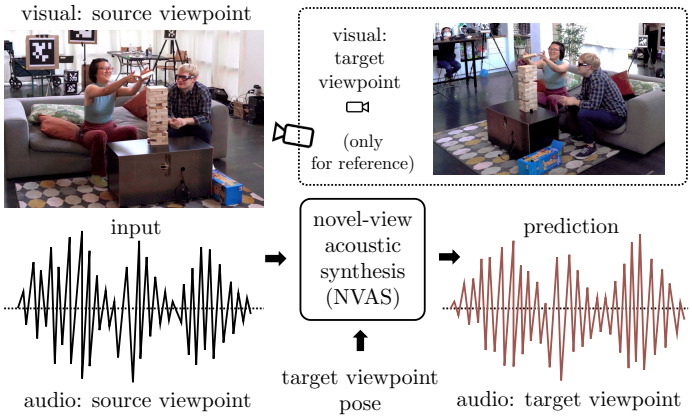


Figure 8.1: **Novel-view acoustic synthesis task.** Given audio-visual observations from one viewpoint and the relative target viewpoint pose, render the sound received at the target viewpoint. Note that the target is expressed as the desired pose of the microphones; the image at that pose (right) is neither observed nor synthesized.

from a new acoustic viewpoint, given only the visual and acoustic input from another source viewpoint in the same scene (Fig. 8.1).

NVAS is very different from the existing NVS task, where the goal is to reconstruct images instead of sounds, and these differences present new challenges. First, the 3D geometry of most real-life scenes changes in a limited manner during the recording. On the contrary, sound changes substantially over time, so the reconstruction target is highly dynamic. Secondly, visual and audio sensors are very different. A camera matrix captures the light in a highly-directional manner, and a single image comprises a large 2D array of pixels. In contrast, sounds are recorded with one or two microphones which are at best weakly-directional, providing only a coarse sampling of the sound field. Thirdly, the frequency of light waves is much higher than that of sound waves; the length of audio waves is thus larger to the point of being comparable to the size of geometric features of the scene, meaning that effects such as diffraction are often dominant, and spatial resolution is low. As a result, techniques that require spatial precision, such as triangulation and segmentation, are not applicable to audio. Lastly, sounds mix together, making it difficult to segment them, and they are affected by environmental effects such as reverberation that are distributed and

largely unobservable.

While the NVS and NVAS tasks are indeed very different, we hypothesize that NVAS is an inherently multimodal task. In fact, vision can play an important role in achieving accurate sound synthesis. First, establishing correspondences between sounds and their sources as they appear in images can provide essential cues for resynthesizing the sounds realistically. For instance, human speech is highly directional and sounds very differently if one faces the speaker or their back, which can only be inferred from visual cues. In addition, the environment acoustics also affect the sound one hears as a function of the scene geometry, materials, and emitter/receiver locations. The same source sounds very differently if it is located in the center of a room, at the corner, or in a corridor, for example. In short, vision provides cues about space and geometry that affect sound, and are difficult to estimate from the sound alone.

In order to validate our hypothesis, we propose a novel *visually-guided acoustic synthesis network* that analyzes audio and visual features and synthesizes the audio at a target location. More specifically, the network first takes as input the image observed at the source viewpoint in order to infer global acoustic and geometric properties of the environment along with the bounding box of the active speaker. The network then reasons how the speaker and scene geometry change in 3D based on the relative target pose with a fusion network. We inject the fused features into audio with a gated multi-modal fusion network and model the acoustic changes between viewpoints with a time-domain model.

In order to conduct our experiments on the new NVAS task, we require suitable training and benchmarking data, of which currently there is none available. To address that, we contribute two new datasets: one real (Replay-NVAS) and one synthetic (SoundSpaces-NVAS). The key feature of these datasets is to record the sight and sound of different scenes from multiple cameras/viewpoints. Replay-NVAS contains video recordings of groups of people performing social activities (e.g., chatting,

watching TV, doing yoga, playing instruments) from 8 surrounding viewpoints simultaneously. It contains 37 hours of highly realistic everyday conversation and social interactions in one home-like environment. To our knowledge, Replay-NVAS represents the first large-scale real-world dataset enabling NVAS. This dataset would also greatly benefit many other existing tasks including NVS, active speaker localization, etc. For SoundSpaces-NVAS, we render 1.3K hours of audio-visual data based on the SoundSpaces [40] platform. Using this simulator, one can easily change the scene geometry and the positions of speakers, cameras, and microphones. This data serves as a powerful test bed with clean ground truth for a large collection of home environments, offering a good complement to Replay-NVAS. For both datasets, we capture binaural audio, which is what humans perceive with two ears. Together the datasets contain 1,337 hours of audio-visual capture, with 1,032 speakers across 121 3D scenes. Datasets are publicly available for future research. †

We show that our model outperforms traditional signal processing approaches as well as learning-based baselines, often by a substantial margin, in a quantitative evaluation and a human study. We show qualitative examples where the model predicts acoustic changes according to the viewpoint changes, e.g., left channel becomes louder when the viewpoint changes from left to right. In a nutshell, we present the first work that deals with novel-view acoustic synthesis, and contribute two large-scale datasets along with a novel neural rendering approach for solving the task.

I first introduce the novel-view acoustic synthesis task in Sec. 8.1, the dataset in Sec. 8.2, our approach in Sec. 8.3 and lastly the results in Sec. 8.4.

8.1 The Novel-View Acoustic Synthesis Task

We introduce a new task, *novel-view acoustic synthesis* (NVAS). Assuming there are N sound emitters in the scene (emitter i emits sound C^i from location L^i),

†`https://replay-dataset.github.io`

given the audio A_S and video V_S observed at the source viewpoint S, the goal is to synthesize the audio A_T at the target viewpoint T, as it would sound from the target location, specified by the relative pose P_T of the target microphone (translation and orientation) with respect to the source view (Fig. 8.1). Furthermore, we assume that the active sound emitters in the environment are visible in the source camera, but we make no assumptions about the camera at the target location.

The sound at any point R is a function of the space:

$$A_R = \mathcal{F}(L^{1,\ldots,N}, C^{1,\ldots,N}, R \mid E), \tag{8.1}$$

where R is the receiver location (S or T) and E is the environment. The emitted sounds C^i are not restricted to speech but can be ambient noise, sounding objects, etc. Our goal here is to learn a transfer function $\mathcal{T}(\cdot)$ defined as $A_T = \mathcal{T}(A_S, V_S, P_T)$, where $S, T, L^{1,\ldots,N}, C^{1,\ldots,N}, E$ are not directly given and need to be inferred from V_S and P_T, which makes the task inherently multi-modal.

This task is challenging because the goal is to model the sound field of a dynamic scene and capture acoustic changes between viewpoints given one pair of audio-visual measurements. While traditional signal processing methods can be applied, we show in Sec. 8.4 that they perform poorly. In this work, we present a learning-based rendering approach.

8.2 Datasets

We introduce two datasets for the NVAS task: live recordings (Sec. 8.2.1), and simulated audio in scanned real-world environments (Sec. 8.2.2) (see Fig. 8.2). The former is real and covers various social scenarios, but offers limited diversity of sound sources, viewpoints and environments, and is noisy. The latter has a realism gap, but allows perfect control over these aforementioned elements.

Both datasets focus on human speech given its relevance in applications. However, our model design is not specific to speech. For both datasets, we capture binau-

Figure 8.2: Example source and target views for the two introduced datasets: Replay-NVAS (left) and SoundSpaces-NVAS (right).

ral audio, which best aligns with human perception. Note that for both datasets, we collect multiple multi-modal views for training and evaluation; during inference the target viewpoint(s) (and in some cases target environment) are withheld.

8.2.1 The Replay-NVAS Dataset

Replay-NVAS contains multi-view captures of acted scenes in apartments. We capture 46 different scenarios (e.g., having a conversation, having dinner, or doing yoga) from 8 different viewpoints. In total, we collect 37 hours of video data, involving 32 participants across all scenarios.

In each scenario, we invite 2–4 participants to act on a given topic. Each participant wears a near-range microphone, providing a clean recording of their own speech. The scene is captured by 8 DLSR cameras, each augmented with a 3Dio binaural microphone. In this way, the data captures video and audio simultaneously from multiple cameras, resulting in 56 possible source/target viewpoint combinations for each scene. The videos are recorded at 30 FPS and the audio is recorded with a 48k sampling rate. We use a clapper at the beginning of the recording for temporal synchronization. Each scenario lasts 3–8 min. We use off-the-shelf software for multi-view camera calibration.

To construct the dataset, we extract one-second long clips from each video with overlapping windows. We automatically remove silent and noisy clips based on the energy of near-range microphones, which results in 77K/12K/2K clips in total for train/val/test. During training, for one sample, we randomly select two out of eight viewpoints, one as the source and one as the target.

This dataset is very challenging. It covers a wide range of social activities. It is harrowed by ambient sound, room reverberation, overlapping speech and non-verbal sounds such as clapping and instruments. Participants can move freely in the environment. We believe that this data will be useful to the community beyond the NVAS task as it can be used for benchmarking many other problems, including active speaker localization, source separation, and NVS.

8.2.2 The SoundSpaces-NVAS Dataset

In this dataset, we synthesize multi-view audio-visual data of two people having conversations in 3D scenes. In total, we construct 1.3K hours of audio-visual data for a total of 1,000 speakers, 120 3D scenes and 200K viewpoints.

Our goal is to construct audio-visual data with strong spatial and acoustic correspondences across multiple viewpoints, meaning that the visual information should indicate what the audio should sound like, e.g., observing speaker on the left should indicate the left ear is louder and observing speaker at a distance should indicate there is higher reverberation. We use the SoundSpaces 2.0 platform [40], which allows highly realistic audio and visual rendering for arbitrary camera and microphone locations in 3D scans of real-world environments [29, 265, 312]. It accounts for all major real-world acoustics phenomena: direct sounds, early specular/diffuse reflections, reverberation, binaural spatialization, and effects from materials and air absorption.

We use the Gibson dataset [312] for scene meshes and LibriSpeech [214] for speech samples. As we are simulating two people having conversations, for a given environment, we randomly sample two speaker locations within 3 m and insert two

copyright-free mannequins (one male and one female) at these two locations.[‡] We then randomly sample four nearby viewpoints facing the center of the two speakers at a height of 1.5 m (Fig. 8.2, right). For each speaker, we select a speech sample from LibriSpeech with matching gender. We render images at all locations as well as binaural impulse response for all pairs of points between speakers and viewpoints. The received sound is obtained by convolving the binaural impulse response with the speech sample.

During training, for one sample, we randomly sample two out of four rendered viewpoints, one as the source and one as the target. We also randomly choose one speaker to be active, simulating what we observe on the real data (i.e., usually only one person speaks at a time).

8.3 Visually-Guided Acoustic Synthesis

We introduce a new method, **Vi**sually-**G**uided **A**coustic **S**ynthesis (ViGAS), to address the NVAS problem, taking as input sound and an image and outputting the sound from a different target microphone pose.

ViGAS consists of five components: ambient sound separation, active speaker localization, visual acoustic network, acoustic synthesis, and temporal alignment. The high-level idea is to separate the observed sound into primary and ambient, extract useful visual information (active speaker and acoustic features), and use this information to guide acoustic synthesis for the primary sound. Temporal alignment is performed during training for better optimization. ViGAS is discussed in detail next and summarised in Fig. 8.3.

[‡]`https://renderpeople.com/free-3d-people`

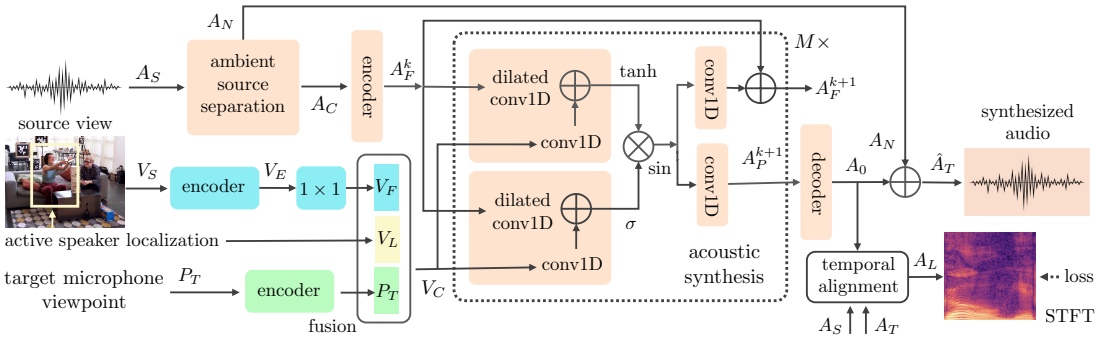


Figure 8.3: **Visually Guided Acoustic Synthesis (ViGAS)**. Given the input audio A_S, we first separate out the ambient sound to focus on the sound of interest. We take the source audio and source visual to localize the active speaker on the 2D image. We also extract the visual acoustic features of the environment by running an encoder on the source visual. We concatenate the active speaker feature, source visual features, and the target pose, and fuse these features with a MLP. We feed both the audio stream A_C and fused visual feature V_C into the acoustic synthesis network, which has M stacked audio-visual fusion blocks. In each block, the audio sequence is processed by dilated conv1d layers and the visual features are processed by conv1d layers. Lastly, the previously separated ambient sound is added back to the waveform. During training, our temporal alignment module shifts the prediction by the amount of delay estimated between the source and the target audio to align the prediction well with the target.

8.3.1 Ambient Sound Separation

ViGAS starts by decomposing the input sound into primary and ambient (traffic, electric noise from a fridge or the A/C, etc.). Ambient sound is important for realism, but it also interferes with learning the model because it can carry significant energy, making the model focus on it rather than on the primary sounds, and its spatial distribution is very different from the primary sounds.

By explicitly separating primary and ambient sounds, ViGAS: (1) accounts for the fact that the transfer functions of primary and ambient sounds are very different and thus difficult to model together; (2) avoids wasting representational power on modelling ambient sounds that might be difficult to reconstruct accurately and depend less on the viewpoint; and (3) prevents ambient sounds, which are noise-like and

high-energy, from dominating learning and reconstruction. In practice, as we show in Sec. 8.4, without the ambient sound separation, the model performs poorly.

The goal of ambient sound separation is thus to construct a function $(A_C, A_N) = \mathcal{P}(A_S)$ that separates the input sound A_S into primary sound A_C and ambient sound A_N. Existing approaches to this problem are based on signal processing [69, 20] or learning [63, 77]. We find that pretrained speech enhancement models such as Denoiser [63] tend to aggressively remove the noise including the primary sound, which hinders re-synthesis. We thus opt for band-pass filtering, passing frequencies within a certain range and rejecting/attenuating frequencies outside of it, which we found to work well. We cut frequencies below 80 Hz for SoundSpaces-NVAS and 150 Hz for Replay-NVAS.

8.3.2 Active Speaker Localization

Knowing where the emitters of different primary sounds are located in the environment can help to solve the NVAS task. In this chapter, we focus on localizing the active speaker, although there can be other important primary sound events like instruments playing, speakers interacting with objects, etc. The goal of active speaker localization is to predict the bounding box of the active speaker in each frame of the video (examples in Fig. 8.4). The bounding box is in the format of $(y_{\min}, y_{\max}, x_{\min}, x_{\max})$ and x, y are normalized to $[0, 1]$ by the image width and height, respectively.

On SoundSpaces-NVAS, this task is relatively easy because of the strong correspondence between the appearance of the speaker and the gender of the speech sample, which enables to easily train a classifier for active speakers. However, this is much harder on Replay-NVAS because cameras record speakers from a distance and from diverse angles, meaning that lip motion, the main cue used by speaker localization methods [132, 276, 241], is often not visible. Hence, the model has to rely on other cues to identify the speaker (such as body motion, gender or identity).

Furthermore, sometimes people speak or laugh over each other.

Since our focus is not speaker localization, for the Replay-NVAS we assume that this problem is solved by an external module that does audio-visual active speaker localization. To approximate the output of such a module automatically, we rely on the near-range audio recordings. Specifically, we first run an off-the-shelf detection and tracker [56] on the video at 5 FPS and obtain, with some manual refinement, bounding boxes B_t^i for $i = 1, \ldots, N$ at each frame t. We manually assign the near-range microphone audio A_N^i to each tracked person. We select the active speaker D based on the maximum energy of each near-range microphone, i.e., $D = \text{argmax}_i \left\{ \sum A_N^i[t : t + \Delta t]^2 \right\}$, where Δt is the time interval we use to calculate the audio energy. We output bounding box B^D as the localization feature V_L.

8.3.3 Visual Acoustic Network and Fusion

The active speaker bounding box B^D only disambiguates the active speaker from all visible humans on 2D, which is not enough to indicate where the speaker is in 3D. To infer that, the visual information is also needed. Since there is usually not much movement in one second (the length of the input video clip), the video clip does not provide much extra information compared to a single frame. Thus, we choose the middle frame to represent the clip and extract the visual acoustic features V_E from the input RGB image with a pretrained ResNet18 [115] before the average pooling layer to preserve spatial information. To reduce the feature size, we feed V_E into a 1D convolution with kernel size 1 and output channel size 8. We then flatten the visual features to obtain feature V_F.

The target pose is specified as the translation along x, y, z axes plus difference between orientations of the source "view" and the target "view" expressed via rotation angles: $+y$ (roll), $+x$ (pitch) and $+z$ (yaw). We encode each angle α as its sinusoidal value: $(\sin(\alpha), \cos(\alpha))$.

Similarly, the target pose is not enough by itself to indicate where the target

viewpoint T is in the 3D space; to infer that, the source view V_S is again needed. For example, in top row of Fig 8.4, for target viewpoint 3, "two meters to the right and one meter forward" is not enough to indicate the target location is in the corridor, while the model can reason that based on the source view.

We use a fusion network to predict a latent representation of the scene variables S, T, L^D, E (*cf.* Sec. 8.1) by first concatenating $[V_L, P_T, V_F]$ and then feeding it through a multilayer perceptron (MLP). See Fig. 8.3 for the network.

8.3.4 Acoustic Synthesis

With the separated primary sound A_C and the visual acoustic feature V_C as input, the goal of the acoustic synthesis module is to transform A_C guided by V_C. We design the acoustic synthesis network to learn a non-linear transfer function (implicitly) that captures these major acoustic phenomena, including the attenuation of sound in space, the directivity of sound sources (human speech is directional), the reverberation level, the head-related transfer function, as well as the frequency-dependent acoustic phenomena. Training end-to-end makes it possible to capture these subtle and complicated changes in the audio.

Inspired by recent advances in time-domain signal modeling [209, 235], we design the network as M stacked synthesis blocks, where each block consists of multiple conv1D layers. We first encode the input audio A_C into a latent space, which is then fed into the synthesis block. The key of the synthesis block is a gated multimodal fusion network that injects the visual information into the audio as follows:

$$z = \tanh(p_A^k(A_F^k) + p_V^k(V_C)) \odot \sigma(q_A^k(A_F^k) + q_V^k(V_C)), \tag{8.2}$$

where $\odot$ indicates element-wise multiplication, σ is a logistic sigmoid function, $k = 1, \ldots, M$ is the layer index and p, q are both learnable 1D convolutions.

After passing z through a sinusoidal activation function, the network uses two separate conv1D layers to process the feature, one producing the residual connection

A_F^{k+1} and one producing the skip connection A_P^{k+1}. All skip connections A_P^{k+1} are mean pooled and fed into a decoder to produce the output A_O. We add back the separated ambient sound A_N as the target audio estimate: $\hat{A}_T = A_O + A_N$.

8.3.5 Temporal Alignment

In order for the model to learn well, it is important that input and output sounds are temporally aligned. While the Replay-NVAS data is already synchronised based on the clapper sound, due to the finite speed of sound, the sounds emitted from different locations may still arrive at microphones with a delay slightly different from the one of the clapper, causing misalignments that affect training.

To align source and target audio for training, we find the delay τ that maximizes the generalized cross-correlation:

$$\mathcal{R}_{A_S,A_T}(\tau) = \mathbb{E}_t[h_S(t) \cdot h_T(t-\tau)], \tag{8.3}$$

where h_S and h_T are the feature embedding for A_S and A_T respectively at time t. We use the feature extractor h from the generalized cross-correlation phase transform (GCC-PHAT) algorithm [146], which whitens the audio by dividing by the magnitude of the cross-power spectral density. After computing τ, we shift the prediction A_O by τ samples to align with the A_T and obtain A_L. Note that alignment is already exact for SoundSpaces-NVAS.

8.3.6 Loss

To compute the loss, we first encode the audio with the short-time Fourier transform (STFT), a complex-valued matrix representation of the audio where the y axis represents frequency and the x axis is time. We then compute the magnitude of the STFT, and optimize the L1 loss between the the predicted and ground truth magnitudes as follows:

$$L = \big|||STFT(A_L)||_2 - ||STFT(A'_T)||_2\big|, \tag{8.4}$$

	SoundSpaces-NVAS						Replay-NVAS		
	Single Environment			*Novel Environment*			*Single Environment*		
	Mag	LRE	RTE	Mag	LRE	RTE	Mag	LRE	RTE
Input audio	0.225	1.473	0.032	0.216	1.408	0.039	0.159	1.477	0.046
TF Estimator [301]	0.359	2.596	0.059	0.440	3.261	0.092	0.327	2.861	0.147
DSP [50]	0.302	3.644	0.044	0.300	3.689	0.047	0.463	1.300	0.067
VAM [39]	0.220	1.198	0.041	0.235	1.131	0.051	0.161	0.924	0.070
ViGAS w/o visual	0.173	0.973	0.031	0.181	1.007	0.036	0.146	0.877	**0.046**
ViGAS	**0.159**	**0.782**	**0.029**	**0.175**	**0.971**	**0.034**	**0.142**	**0.716**	0.048

Table 8.1: **Results on SoundSpaces-NVAS and Replay-NVAS.** We report the magnitude spectrogram distance (Mag), left-right energy ratio error (LRE), and RT60 error (RTE). Replay-NVAS does not have novel environment setup due to data being collected in a single environment. For all metrics, lower is better. In addition to baselines, we also evaluate ViGAS w/o visual by removing the active speaker localization and visual features. Note that reverberation time is mostly invariant of the receiver location in the same room and thus input audio has low RTE. A good model should preserve this property while synthesizing the desired acoustics for the target viewpoint.

where A'_T is the primary sound separated from A_T with $\mathcal{P}(\cdot)$. By taking the magnitude, we do not model the exact phase values, which we find hinders learning if being included in the loss.

8.4 Experiments

We compare with several traditional and learning-based baselines and show that ViGAS outperforms them in both a quantitative evaluation and a human subject study.

Evaluation. We measure performance from three aspects: 1. closeness to GT as measured by the **magnitude spectrogram distance (Mag)**. 2. correctness of the spatial sound as measured by the **left-right energy ratio error (LRE)**, i.e., the difference of ratio of energy between left and right channels and 3. correctness of

the acoustic properties measured by **RT60 error (RTE)** [257, 39], i.e., the error in reverberation time decaying by 60dB (RT60). We use a pretrained model [39] to estimate RT60 directly from speech.

Baselines. We consider the following baselines: 1. **Input audio**. Copying the input to the output. 2. **TF Estimator [301] + Nearest Neighbor**, i.e. storing the transfer function estimated during training and retrieving the nearest neighbor during test time. We estimate transfer functions with a Wiener filter [301] and index them with the ground-truth locations of the speaker, source viewpoint, and target viewpoint for the single environment setup and their relative pose for the novel environment setup. At test time, this method searches the database to find the nearest transfer function and applies it on the input audio. 3. **Digital Signal Processing (DSP) [50]** approach that takes the distance, azimuth, and elevation of the sound source, applies an inverse a head-related transfer function (HRTF) to estimate the speech spoken by the speaker and then applies another HRTF to estimate the audio at the target microphone location. This baseline adjusts the loudness of the left and right channels based on where the speaker is in the target view. We supply GT coordinates for SoundSpaces-NVAS and speakers' head positions estimated with triangulation on Replay-NVAS. 4. **Visual Acoustic Matching (VAM) [39]**, recently proposed for a related task of matching acoustics of input audio with a target image. This task only deals with single viewpoint and single-channel audio. We adapt their model with minimal modification by feeding in the image from the *source* viewpoint and concatenating the position offset of the target microphone at the multimodal fusion step.

8.4.1 Results on SoundSpaces-NVAS

Table 8.1 shows the results. For synthetic data, we consider two evaluation setups: 1. single environment: train and test on the same environment and 2.

novel environment: train and test on multiple non-overlapping Gibson environments (90/10/20 for train/val/test).

In the single environment setup, our model largely outperforms all baselines as well as our audio-only ablation on all metrics. TF Estimator performs poorly despite being indexed by the ground truth location values because estimating a transfer function directly from two audio clips is non-trivial and noisy for low-energy parts of the signal. DSP also performs badly despite having the ground truth 3D coordinates of the sound source. This is because head related transfer functions are typically recorded in anechoic chambers, which does not account for acoustics of different environments, e.g., reverberation. Both traditional approaches perform worse than simply copying the input audio, indicating that learning-based models are needed for this challenging task. The recent model VAM [39] performs much better compared to the traditional approaches but still underperforms our model. There is a significant difference between ViGAS w/o visual and the full model; this shows that the visual knowledge about the speaker location and the environment is important for this task.

Fig. 8.4 shows an example where given the same input source viewpoint, our model synthesizes audio for three different target viewpoints. The model reasons about how the geometry and speaker locations changes based on the source view and the target pose, and predicts the acoustic difference accordingly.

For the novel environment setup, our model again outperforms all baselines. Compared to ViGAS in the single environment setup, both the magnitude spectrogram distance and the left-right energy ratio error increase. This is expected because for novel (unseen) environments, single images capture limited geometry and acoustic information. The model fails sometime when there is a drastic viewpoint change, e.g., target viewpoint 3 in Fig. 8.4. This setup requires the model to reason or "imagine" the environment based on single audio-visual observation, which poses great challenge for NVAS as well as NVS, where typically synthesis is performed in a fully observed environment.

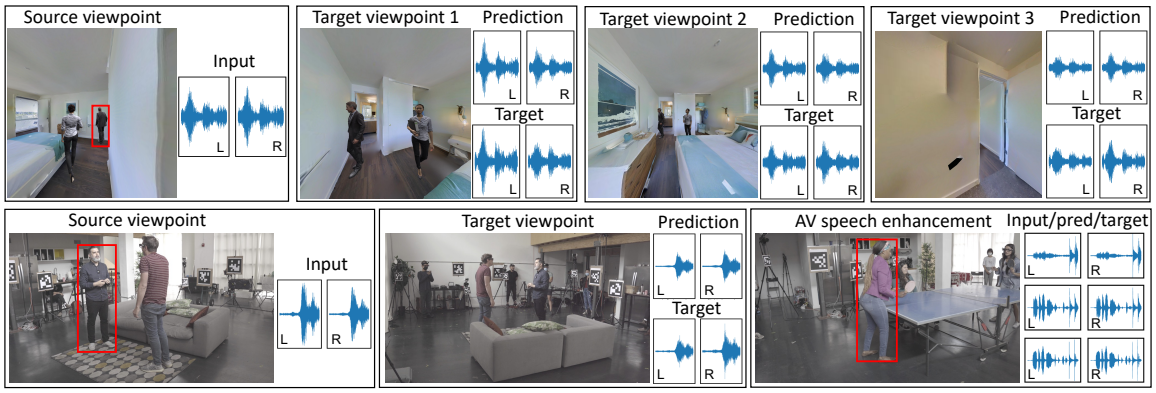


Figure 8.4: **Qualitative examples**. For all binaural audio, we show the left-channel and the right-channel waveforms side-by-side. Row 1: SoundSpaces-NVAS example where given the source viewpoint and input audio, the model synthesizes audio for three different target viewpoints (target views are for reference only). In this case, the active speaker is the male speaker as indicated by the bounding box. For target viewpoint 1, the view rotates about 90 degrees and the male speaker is on the left side and the predicted left channel is louder than the right channel. Viewpoint 2 moves away from the speaker and thus yields lower amplitude compared to the first prediction. For target viewpoint 3, it is completely located outside of the living room, in which case, the sound could only come from the door open on the right (louder right channel) and the reverberation also greatly increases due to the vanishing direct sound. Row 2: Replay-NVAS example where the speaker is located on the left in the source viewpoint which becomes the right and further from the camera in target viewpoint 2, the model also predicts lower amplitude and louder right channel. On the right side, we show an example of the audio-visual speech enhancement for the active speaker. The model enhances the speech to largely match with the near-range audio (target).

Ablations. Table 8.2 shows ablations on the model design. To understand if the model uses visual information, we ablate the visual features V_F and the active speaker feature V_L. Removing the active speaker feature leads to less damage on the model performance, because without the explicitly localized active speaker, the model can still implicitly reason about the active speaker location based on the image and audio. If both are removed ("ViGAS w/o visual" in Table 8.1), the performance suffers most.

To study the effectiveness of the temporal alignment and ambient sound separation modules, we ablate them separately. Removing the temporal alignment leads

ViGAS	SS-NVAS		Replay-NVAS	
	Mag	LRE	Mag	LRE
full model	**0.159**	0.782	**0.142**	0.716
w/o visual features	0.171	0.897	0.146	0.920
w/o ASL	0.161	0.814	0.143	0.757
w/o alignment	0.176	**0.771**	0.144	**0.706**
w/o separation	0.165	0.840	0.182	0.859

Table 8.2: **Ablations of the model on both datasets**.

to higher Mag error and slightly lower LRE. As for ambient sound separation, the results show that optimizing for the high-energy noise-like ambient sound degrades the performance.

8.4.2 Results on Replay-NVAS

Table 7.1 (right) shows the Replay-NVAS results. Compared to SoundSpaces-NVAS, the magnitudes of all errors are smaller because there are less drastic acoustic changes between viewpoints (8 DLSR cameras form a circle around the participants). Traditional approaches like TF Estimator and DSP still perform poorly despite using the 3D coordinates of the camera and the speaker (triangulated from multiple cameras). VAM performs better due to end-to-end learning; however, our model outperforms it. Compared to ViGAS w/o visual, the full model has much lower left-right energy ratio error and slightly higher reverberation time error, showing that the model takes into account the speaker position and viewpoint change for synthesizing the audio.

Fig. 8.4 (row 2, left) shows a qualitative example. In the source viewpoint, the active speaker is on the left, while in the target viewpoint, he is further from the camera and on the right. The model synthesizes an audio waveform that captures the corresponding acoustic change, showing that our model successfully learns from real videos.

	Mag	RTE
Input	0.279	0.376
ViGAS (ours)	**0.234**	**0.122**

Table 8.3: **Speech enhancement on Replay-NVAS**.

Dataset	Input	DSP	ViGAS
SoundSpaces-NVAS	24%	2%	**74%**
Replay-NVAS	43%	6%	**51%**

Table 8.4: **Human Study**. Participants favor our approach over the two most realistic sounding baselines, (1) copying the input signal, and (2) a digital signal processing baseline.

Audio-visual speech enhancement. In some real-world applications, e.g., hearing aid devices, the goal is to obtain the enhanced clean speech of the active speaker. This can be seen as a special case of NVAS, where the target viewpoint is the active speaker. Our model is capable of performing audio-visual speech enhancement without any modification. We simply set the target audio to the near-range audio recording for the active speaker. We show the results in Table 8.3. Our model obtains cleaner audio compared to the input audio (example in Fig. 8.4, row 2, right).

Human subject study. To supplement the quantitative metrics and evaluate how well our synthesized audio captures the acoustic change between viewpoints, we conduct a human subject study. We show participants the image of the target viewpoint V_T as well as the audio A_T as reference. We provide three audio samples: the input, the prediction of ViGAS, and the prediction of DSP (the most naturally sounding baseline) and ask them to select a clip that sounds closest to the target audio. We select 20 examples from SoundSpaces-NVAS and 20 examples from Replay-NVAS and invite 10 participants to perform the study.

See Table 8.4 for the results. On the synthetic dataset SoundSpaces-NVAS, our approach is preferred over the baselines by a large margin. This margin is lower

on the real-world Replay-NVAS dataset but is still significant.

8.5 Conclusion

We introduce the challenging novel-view acoustic synthesis task and a related benchmark in the form of both real and synthetic datasets. We propose a neural rendering model that learns to transform the sound from the source viewpoint to the target viewpoint by reasoning about the observed audio and visual stream. Our model surpasses all baselines on both datasets. We believe this research unlocks many potential applications and research in multimodal novel-view synthesis.

While this is very exciting, we acknowledge that we are feeding the ground truth active-speaker localization bounding boxes into the model, which has simplified the task. In the future, we plan to incorporate active-speaker localization models and let the model jointly learn to localize and synthesize. In addition to this, the proposed model heavily relies on the training data to learn to synthesize sounds for novel viewpoints, which might be the reason why the model does not generalize to novel environments very well. Incorporating prior knowledge of human head models could improve the data efficiency as well as generalization.

Chapter 9: SoundingActions: Learning How Actions Sound from Narrated Egocentric Videos

In previous chapters, I covered simulation platforms that support audio-visual rendering in Chapter 3, learning navigation policies in Chapter 4 and Chapter 5 as well as learning acoustic properties of 3D environment in Chapter 6, Chapter 7 and Chapter 8. The key to all these problems is understanding how sound propagates in space as a function of the environment. Before sound waves propagate, they are produced from object vibrations due to certain external forces such as human actions. To understand the association between human actions and the sounds they make, I propose to learn action sounds from egocentric videos, which provide rich information about how human actions produce sounds. More specifically, I aim to answer two questions to approach the problem: 1. What actions sound? and 2. How to generate action sounds? I will investigate the first problem in this chapter and the second problem in the next chapter. This work was accepted at CVPR 2024.

Human activity often produces sounds. Closing a door, chopping vegetables, typing on a keyboard, talking with a friend—our interactions with the objects and people around us generate audio that reveals our physical behaviors. These sounds can be strongly associated with the subjects of our activity and how we perform it. For example, opening a water bottle sounds different than opening a cabinet; chopping sweet potatoes sounds different than chopping onions; chopping onions sounds different than mincing onions (the same object). Understanding the link between sounds and actions is valuable for a number of applications, such as multimodal activity recognition, crossmodal retrieval, content generation, or forecasting the physical effects of a person's actions.

How should AI learn about *sounding actions*? Existing work typically curates annotated datasets for supervised learning [93, 121, 46, 220], taking care to select

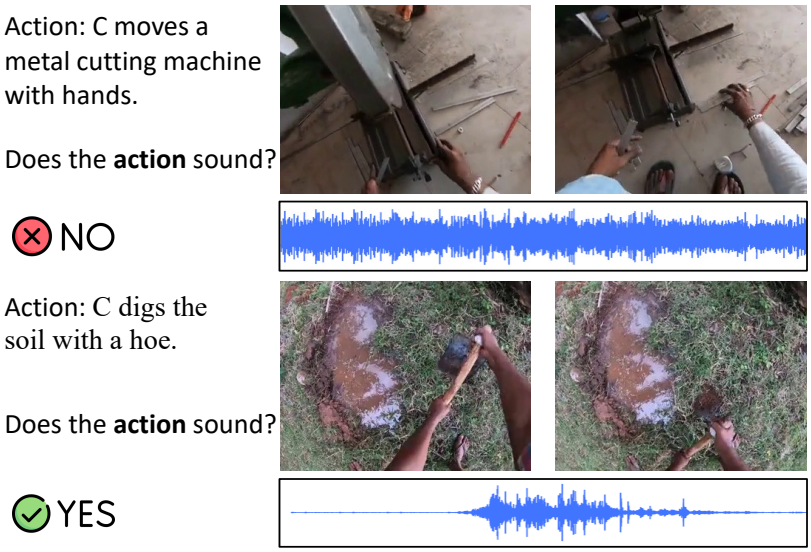


Figure 9.1: We aim to distinguish sounds that are directly caused by human actions (bottom) from those that are not (top). Given egocentric training videos with language descriptions of the camera wearer's ("C") current action, we learn an embedding where the audio and visual features of any given clip are best aligned only when both are also consistent with the language. This allows discerning clips where the audio and vision may be *correlated* (e.g., the cutting machine running making loud noise in top row) versus those where the sounds are *driven by human action* (digging in bottom row)—importantly, without language at inference time.

events or actions that have associated sounds (e.g., lawnmowing, chopping), while others deliberately collect videos of object collisions (e.g., striking objects with a drumstick [212] or crashing into them with a robot [85, 53]), or develop physics-based simulations [84]. On the one hand, these approaches are appealing for their ability to focus on meaningful audio-visual correspondences. On the other hand, their curated nature risks limiting the scope of sounding actions that can be learned.

Instead, we aim to learn how human actions sound from narrated in-the-wild egocentric videos. See Figure 9.1. Given a pool of videos of everyday human activity, the goal is to learn a crossmodal representation where sounding actions cluster together based on how they look and sound. By sampling the videos freely, we can broaden the scope to discover the breadth of sounding actions without having to

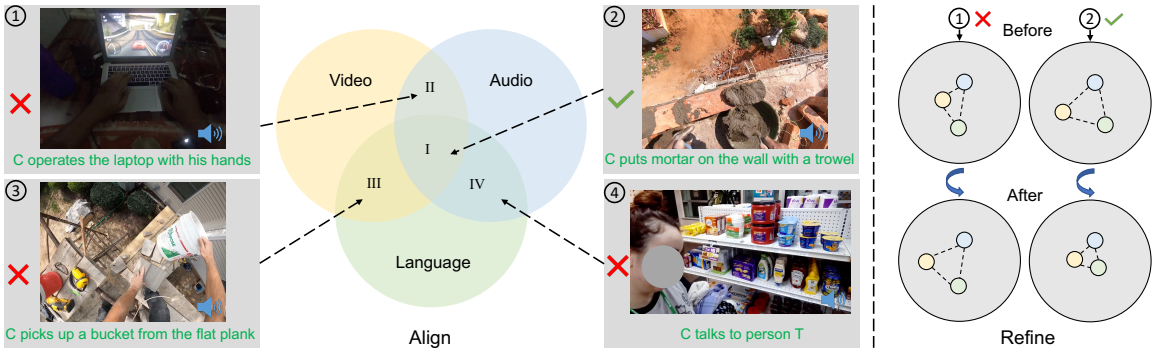


Figure 9.2: **Main idea**. On the left, the Venn diagram illustrates different ways audio (A), video (V) and language (L) modalities can overlap in the content they capture. C refers to the camera wearer. Regions II,III,IV are information that is only shared between two modalities but not the third, e.g., the racing game in ① where the game sounds correlate with the vision, yet are not about the camera wearer's described action (using hands on laptop), the lifting action in ③, where the visuals and language agree but the action is inaudible, and the off-screen talking action in ④, where talking is heard and described, but the camera wearer cannot be seen speaking. Region I is the information that corresponds to all modalities agreeing, e.g., the visible and audible plastering action in ②. Our model's "align" phase detects any such (dis)agreements via pairwise contrastive learning on the modalities. In the "refine" phase, we use the intersection of that agreement (region I) to refine the embedding. For example, on the right, we show what the three modality embeddings should look like after the "align" stage for examples 1 and 2. Embeddings of instances where all modalities agree will be closer in the embedding space and apart otherwise. In other words, for example 1, yellow (video) cannot be close to blue (audio) unless green is too (language).

rely on a closed, pre-defined set of action categories. In particular, by focusing on *unscripted egocentric* video from wearable cameras in daily-life settings [105, 57], we aim to include subtle and long-tail scenarios unavailable in curated datasets, such as sounds of keys jangling when unlocking a door, scissors snipping when cutting the dog's fur, or fingernails scratching on one's own arm. Egocentric video is a particularly attractive source here because 1) human interaction sounds are more audible in near-field egocentric recordings and 2) passively captured long-form ego-video simply covers more everyday sounds, including the rare ones.

However, the learning task is challenging because some visible actions do not make any sound, and some sounds are the result of off-screen actions. Finally, other

sounds may be *correlated* with on-screen objects (such as traffic noise and a city street), but are not directly related to the video by a salient and visible camera wearer action. For this reason, although existing self-supervised audio-visual methods [11, 12, 153, 211, 175, 90, 96, 6] are good at detecting audio-visual correspondences, they tend to capture general correlations rather than the action-specific correspondence.

To address this challenge, we propose a novel *multimodal consensus embedding* approach. Importantly, we suppose the in-the-wild egocentric training videos are accompanied by free-form natural language descriptions describing the actions of the camera wearer, as provided in the "narrations" of existing large-scale ego-video datasets [105, 57]. The main idea is to seek video samples where there is semantic agreement between all three modalities—the audio, visual, and language—while distancing those that do not. This *intersection* of the modalities with language assures that correspondences in the audio and visual streams stem from alignment on the sounding action.

To achieve this, the proposed model first aligns a preliminary embedding from contrastive losses imposed per instance on each pair of modalities. Next, we refine those embeddings with a consensus objective that targets a minimum (bottleneck) pairwise similarity. The latter pushes all pairs of inter-modality agreement towards this consensus—or lack thereof—while jointly continuing to optimize the paired-modalities' contrastive losses. In this way, we overcome the simplifying assumption made by existing multimodal embeddings that require all modalities to agree [281, 96, 2]. See Figure 9.2.

We demonstrate our approach by training with in-the-wild data from Ego4D [105] without audio labels and testing on both Ego4D and EPIC-Sounds [121]. To allow a formal large-scale evaluation of sounding actions, we introduce a dataset of professional annotations on 33K video clips spanning Ego4D. Our model successfully discovers sounding actions that agree with ground truth labels on both datasets. Compared to existing multimodal embedding paradigms [68, 175, 281, 96], our model

not only better discovers sounding actions and learns embeddings for crossmodality retrieval, but also generalizes better to the audio classification benchmark on EPIC-Sounds. To our knowledge, this is the first result of its kind to show sounding actions discovered organically from narrated in-the-wild video. We will release the data and code for our models.

I first formulate the task in Sec. 9.1, introduce the approach in Sec. 9.2, cover the training and evaluation data in Sec. 9.3 and lastly discuss the experimental results in Sec. 9.4.

9.1 Task Formulation

We define a **sounding action** as a human-initiated action that produces sound during its execution due to interactions with the surrounding environment. We are particularly interested in learning how subtle and long-tail daily human actions sound. If hypothetically we were given a clip with audio a, video v, and label y indicating whether the clip contains a sounding action, our objective would be to minimize the distance between audio-visual embeddings if $y = 1$ and maximize the distance between them if $y = 0$, i.e., minimizing $(-1)^y \mathcal{D}(e_a, e_v)$, where $\mathcal{D}$ measures the distance and $e_{a,v}$ are their embeddings. However, we do not assume access to any such direct supervision; labeling sounding actions is expensive, both because many actions do not produce sounds, and because many clips do not contain actions. Instead, we aim to discover sounding actions in a weakly supervised fashion, while simultaneously learning multimodal embeddings that capture them well.

To this end, we leverage "narrations", a form of language description that is collected in recent egocentric video datasets such as Ego4D [105] and EPIC-Kitchens [57]. These narrations are timestamped free-form sentences describing the current action being performed by the camera-wearer. See Figure 9.2 for examples. Note that there may be other events in the video, too (e.g., a TV is playing), but these are *not* narrated. This is significant: the language specifically addresses near-

field human interactions with objects, people, and the environment. The narrations offer two key benefits: 1) the timestamps provide *temporal* grounding of actions that occur in the video, indicating where potentially interesting clips are and 2) the language provides *semantic* grounding of actions—which our multimodal consensus idea will exploit to learn action-specific audio-visual correspondence.

Formally, given a video with frames $v \in \mathcal{R}^{T\times H\times W\times C}$, audio $a \in \mathcal{R}^{S}$, and language narration l, where T and S are the number of frames for video and audio respectively, the goal is to learn embeddings e_v and e_a that are close in the embedding space if both a and v capture the same human action described in l, and distant otherwise. If we plot how the three modalities overlap in a Venn diagram (Figure 9.2), we can see that what we are interested in learning is exactly region I, i.e., a camera-wearer action that sounds. From an information-theory perspective, this is equivalent to learning modality-invariant embeddings.

9.2 Multimodal Contrastive-Consensus Coding

Next we present our solution MC3 (Multimodal Contrastive-Consensus Coding) for learning modality-invariant embeddings, which consists of an *inter-sample contrastive loss* and an *intra-sample consensus loss*. See Fig. 9.3. We first present the two-stage training framework in Sec. 9.2.1 and then discuss the two losses in Sec. 9.2.2 and Sec. 9.2.3. For simplicity, we denote the n input modalities as $M_i, i \in [1, n]$.

9.2.1 Align-Refine Two-stage Training

We design a two-stage training paradigm. The high-level idea is to first optimize the pairwise agreement in an "align" stage, and then refine these embeddings with global consensus in the "refine" stage. See Fig. 9.2.

In the first stage, we train modality encoders with a contrastive loss $\mathcal{L}_{\text{contrastive}}$, which guides modality embeddings to have a good initial alignment that captures the pairwise similarity between modalities that capture the same underlying action, as

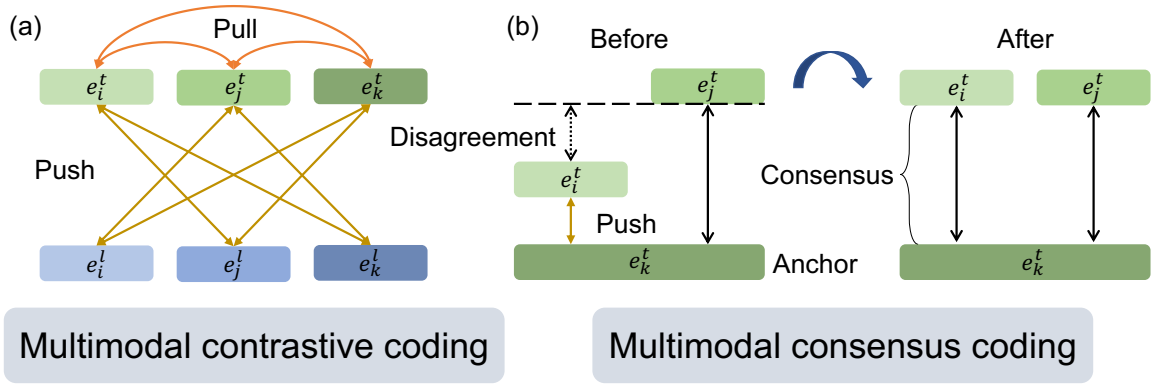


Figure 9.3: **Multimodal contrastive-consensus loss**. (a): Given three modality embeddings e_i^t, e_j^t, e_k^t, multimodal contrastive coding pulls each pair of modalities closer while pushing modality pairs from another sample further away. (b): However, not all modalities agree on how close they should be depending on the instance. Thus we set the furthest distance a feature has with respect to the anchor feature as the consensus and push the remaining embeddings away to meet this consensus.

opposed to random initialization.

In the second stage, we refine the pairwise-aligned embeddings with a globally established consensus. Specifically, we train the model with a consensus loss $\mathcal{L}_{\text{consensus}}$ that pushes all intra-sample modality agreement towards this consensus, while jointly optimizing the contrastive loss $\mathcal{L}_{\text{contrastive}}$, to maximally capture the shared information across modalities. The MC3 loss $\mathcal{L}_{\text{MC3}}$ combines the contrastive and consensus losses, and will be detailed below. We confirm experimentally that it is important to keep the contrastive loss in the second stage, although the main purpose of this stage is to refine embeddings with consensus.

9.2.2 Multimodal Contrastive Coding

crossmodal contrastive learning has been shown to discover representations where modalities are informative of each other [192]. Prior work [288, 221] shows that minimizing the contrastive loss between M_i and M_j maximizes the lower bound on the mutual information $I(M_i; M_j)$. Inspired by this, we first use contrastive learning to optimize the pairwise similarities $\mathcal{S}(e_i, e_j) = e_i e_j$, where $e_{i,j}$ is the latent embedding

normalized on the unit sphere for modality pair i, j. We use the InfoNCE [288] loss to optimize each individual $\mathcal{S}(e_i, e_j)$ as follows:

$$\mathcal{L}_{i,j} = -\frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \log \frac{\exp(e_i^t e_j^t / \tau)}{\sum_{l \in \mathcal{B}} \exp(e_i^t e_j^l / \tau)}, \tag{9.1}$$

where $\mathcal{B}$ is the batch and τ is the temperature. This loss treats modalities from the same sample as positive pairs and pulls them closer and it treats modalities from different samples as negative pairs and pushes them apart. See Fig. 9.3 (a). The total loss is the sum of losses enumerated over all pairs of modalities, i.e., $\mathcal{L}_{\text{contrastive}} = \sum_{i,j} \mathcal{L}_{i,j}$.

9.2.3 Multimodal Consensus Coding

The contrastive loss above attempts to bring all temporally co-occurring modalities closer assuming there are strong correspondences among them in the input space. However, naively doing so would be problematic for instances where not all modalities agree (cf. Figure 9.2). To tackle this issue, we propose a novel objective that leverages the consensus of inter-sample modalities discovered from the contrastive coding as additional supervision.

First of all, we choose an anchor modality M_a, which serves as the point of comparison for other modalities $M_i, i \in [1, n], i \neq a$. With the normalized embedding e_i^t of modality i and sample t, we then compute the cosine similarity score between each non-anchor modality and the anchor modality. Now, these similarity scores may or may not agree with each other. To only learn embeddings shared across all modalities, we set the consensus score as the minimum (bottleneck) score:

$$c^t = \mathcal{K}^{-1}(\min_{i, i \neq a}(\mathcal{K}_1(e_1^t e_a^t), ..., \mathcal{K}_n(e_n^t e_a^t))), \tag{9.2}$$

where $\mathcal{K}_i(x) = ((x+1)/2)^{\alpha_i}, x \in [-1, 1]$ is a modality-specific scaling function that first maps scores to [0, 1] and then adjusts the distribution with a tunable parameter α_i. $\mathcal{K}^{-1}$ is the inverse function that maps the scaled score back to the original space.

The intuition behind $\mathcal{K}_i(x)$ is that different modalities carry different amounts of information and we want to normalize the score distributions among the different modality pairs, making them comparable.

The consensus score c^t is high if and only if all pairwise scores are high, and it is low if there exists at least one modality that does not agree with the anchor modality. After obtaining the consensus score, we design a loss that forces all modalities to follow this consensus, as follows:

$$\mathcal{L}_{\text{consensus}} = \frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \sum_{i, i \neq a} ||e_i^t e_a^t - c^t||_2 \tag{9.3}$$

The total loss $\mathcal{L}_{\text{MC3}}$ is the sum of the contrastive loss (Eq. 9.1) and the consensus loss (Eq. 9.3):

$$\begin{aligned} \mathcal{L}_{\text{MC3}} = -\frac{1}{|\mathcal{B}|}(& \underbrace{\sum_{t \in \mathcal{B}} \sum_{i,j} \log \frac{\exp(e_i^t e_j^t / \tau)}{\sum_{l \in \mathcal{B}} \exp(e_i^t e_j^l / \tau)}}_{\text{Inter-sample}} \\ & \underbrace{- \sum_{i, i \neq a} ||e_i^t e_a^t - c^t||_2}_{\text{Intra-sample}}). \end{aligned} \tag{9.4}$$

This loss pushes embeddings with a low consensus score apart while pulling together embeddings with a high consensus score, and thus aligns embeddings better in the joint embedding space. See Fig. 9.3.

Optimizing this loss is not trivial since it has both contrastive and reconstruction objectives. Indeed, directly optimizing the loss does not work well as shown in the ablation study (Sec. 9.4.1). The proposed two-stage training paradigm (Sec. 9.2.1) helps train the model stably.

9.2.4 Implementation Details

Our modalities of interest are $M_1 = A$ (audio), $M_2 = V$ (vision), and $M_3 = L$ (language). There are six pairwise contrastive losses for three modalities. When

computing the modality consensus, we empirically find using audio as the anchor leads to the best results in our task (cf. Sec. 4.2.2). We set the scaling parameters α_l and α_v to 1 and 0.5 respectively, based on a hyperparameter search on the validation set.

For extracting the feature representations, we use TimeSformer [21] as our video encoder, DistillBERT [242] as our text encoder, and AST [99] as our audio encoder. We initialize the video and language encoders with embeddings from [165], and the audio encoder with embeddings pretrained on ImageNet [64]. We train all encoders. We choose these initial encoders due to their good results in the literature; however, our MC3 loss is not specific to the choice of these encoders and others could be swapped in.

We train all models on 8 A40 GPUs with a learning rate of 3e$-$5 and batch size of 256 for 5 epochs for both stages, and take the final checkpoint for evaluation. We use the Adam optimizer [143]. Our implementation is based on the codebase from [165].

9.3 Training and Eval Data for Sounding Actions

Dataset. Ego4D [105] is a large-scale egocentric video dataset that has more than 3,600 hours of video recordings depicting hundreds of daily activities—and 2,113 of those hours have audio available. As discussed, it also has time-stamped narrations that are free-form sentences describing the current activity performed by the camera-wearer. However, Ego4D has no annotation of whether an action makes sounds, what sounds an action makes, or whether there exists other (non-action) sounds. It is thus non-trivial to detect if an action in the clip makes sound based on simple heuristics, e.g., the burst of sound energy, since many actions could produce continuous sounds with ambient-sound characteristics, e.g., wiping tables or sawing wood.

We construct the training dataset by extracting clips from each Ego4D video based on the narration timestamps. These clips cover a wide range of daily activities

Wash	Close	Cut	Drop	Stir	Wipe	Rub	Touch	Lift	Hold
0.90	0.82	0.77	0.64	0.64	0.53	0.39	0.27	0.19	0.09

Table 9.1: Example verb groups and how frequently they sound

and environments, including construction sites, cooking, arts and crafts, shopping, farming, and many others. Since the timestamp is only an approximate point for where an action occurs, we sample the clip from 0.5 s before to 1 s after the timestamp (1.5 s duration) so that the clip is likely long enough to capture the action sound, if there is any, without introducing visuals that stray from the narrated action. We sample a training set of 250K clips from 1,876 hours of video. From their narrations, we find there are 6,114 unique nouns (objects) and 2,819 unique verbs (actions).

Ground truth annotations for evaluation. Today's egocentric video datasets lack annotations for sounding actions. Thus, to determine how well our model learns long-tail sounding actions and facilitate future research, we collect a large ground truth evaluation set for Ego4D using professional annotators trained for the task. It consists of 33K clips manually labeled as to whether or not the camera wearer's action sounds, i.e., indicating whether the action described in the narration is both visible and audible in the clip.

To ensure annotation quality, in addition to providing concrete examples and annotation guidelines and iterating with quality control feedback to the professional annotators, we assign three annotators per clip and take the majority vote as the correct answer. We split the 33K obtained annotations into 3K for validation and 30K for test. We stress that this is an eval set only; our training data (above) has no manual labels about sound, only free-form language narrations.

Action type analysis. In total, among the 33,000 resulting ground truth clips, 17,693 are positive and 15,307 are negative. The fact that only half of this in-the-wild clip distribution consists of sounding actions underscores the need for models

that can tell the difference between audio that *co-occurs* with human action and *actions that sound*. To gain insight into the annotations, we group them by semantic similarity and analyze them at a group level. While narrations provide semantic descriptions of actions, using them for grouping would be too noisy since the same action could be described in different ways. To reduce the influence of narration variance, we utilize the taxonomy defined in Ego4D (for analysis only, not training). For example, "check", "examine", and "inspect" should belong to the same group (taxon). We first group these clips by verb alone, i.e., extracting verbs from narrations and then applying the taxonomy, which results in 106 unique groups. We then compute the percentage of clips in each group that make sounds. Tab. 9.1 shows 10 examples. We see that actions involving more significant human motions (wash, close, cut) are more often sounding, whereas more subtle movements (lift, hold) are often not. Importantly, there is not a one-to-one mapping between an action verb and its sounding label—how actions sound is scenario-dependent and hence must be mined from the data.

While grouping by verbs provides some insights, how actions make sounds also depends on the object that they interact with, e.g., cutting a carrot sounds different from cutting bread. To this end, we further group the 17K sounding clips by both verbs and nouns, which results in 2,388 unique action groups. We plot the long-tail distribution of them in Fig. 9.4 and show examples sampled from this distribution. This plot shows the diverse and long-tail nature of sounding actions and our test set annotations, which is not present in existing action datasets [121, 212, 85, 53, 261, 73].

9.4 Experiments

We compare our model with several baselines and ablations on three tasks: sounding action discovery (on Ego4D), sounding action retrieval (on Ego4D), and audio event classification (on EPIC-Sounds). We show our model outperforms an array of existing learning methods.

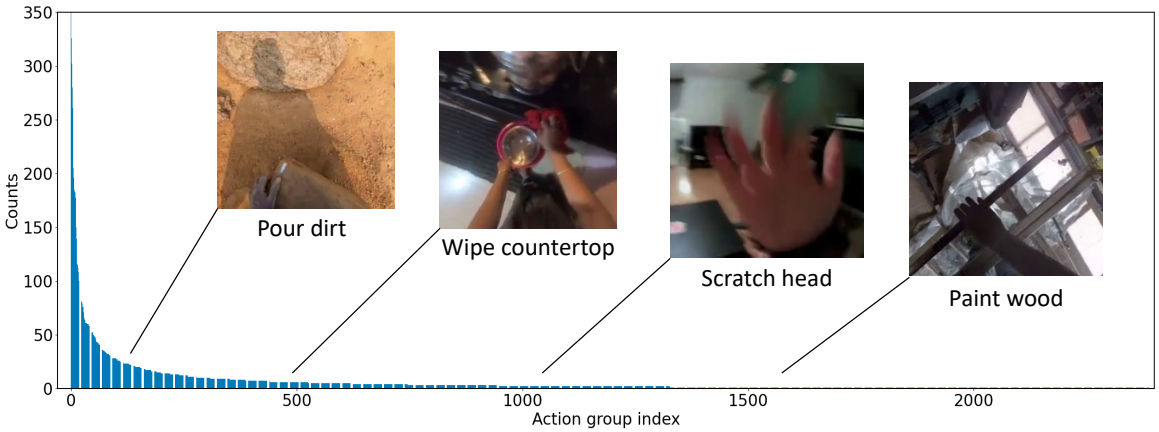


Figure 9.4: Long-tail distribution of sounding actions.

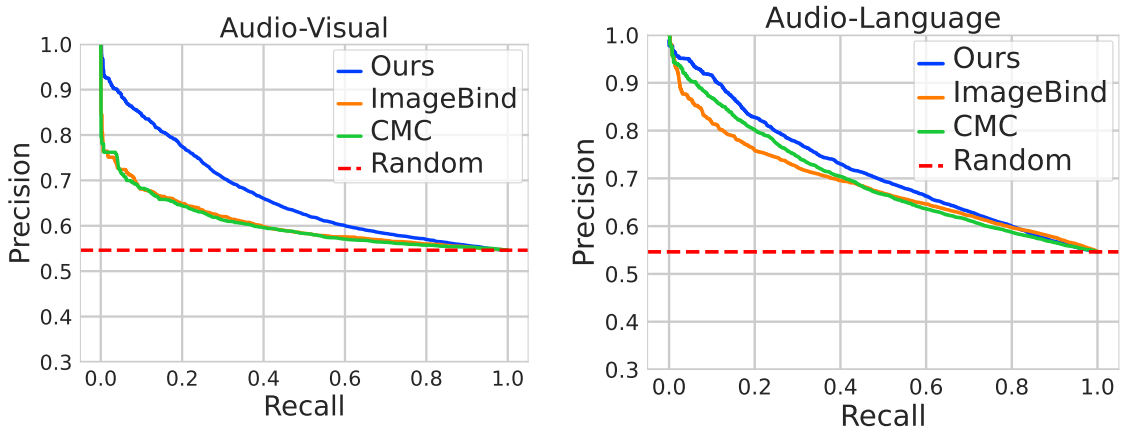


Figure 9.5: Sounding action discovery accuracy

SotA Baselines. We consider two baselines that only use a contrastive loss for two modalities: CLAP [68] for audio-language and CM-ACC [175] for audio-video. For more than two modalities, we consider two more baselines: CMC [281] uses contrastive objectives between all pairs of viewpoints (modalities in our case), representing the joint training paradigm; ImageBind [96] learns the joint embedding by first performing vision-language pretraining and then freezing the vision encoder and training the vision-audio modality pair. This represents strategies that align modalities sequentially. For a fair comparison, we equip all baselines with the same encoder and the same initialization as ours (see Sec. 9.2.4) while keeping their original losses.

				AV		*AL*	
	🔊	🎥	📄	ROC	PR	ROC	PR
Random	✗	✗	✗	0.500	0.559	0.500	0.559
CLAP [68]	✓	✗	✓	-	-	0.637	0.695
CM-ACC [175]	✓	✓	✗	0.540	0.590	-	-
CMC [281]	✓	✓	✓	0.550	0.601	0.635	0.693
ImageBind [96]	✓	✓	✓	0.554	0.605	0.642	0.685
w/o $\mathcal{L}_{\text{consensus}}$	✓	✓	✓	0.563	0.615	0.635	0.694
w/o $\mathcal{L}_{\text{contrastive}}$	✓	✓	✓	0.436	0.493	0.584	0.620
w/o align-stage	✓	✓	✓	0.448	0.507	0.464	0.521
MC3	✓	✓	✓	**0.598**	**0.666**	**0.658**	**0.715**

Table 9.2: Sounding action discovery. Area-under-curve (AUC) values are reported for both ROC and precision-recall (PR) curves, for audio-vision (AV) and audio-language (AL). Both are the higher the better. We train our model five times with different seeds; the standard deviation is always within 0.01.

9.4.1 Sounding Action Discovery

Human interactions with objects in our daily lives are complex and subtle. Due to many incidental background sounds, recognizing whether actions make sound is not trivial but can be useful for applications like multimodal video generation, e.g., verifying the generated action video and audio match. Towards this goal, we answer the question "what actions sound?" by performing sounding action discovery. In this experiment, we take the per-modality encoders learned on the narrated 250K Ego4D clips and apply them to the 30K test clips. Given a test clip, we feed the video and audio through their corresponding modality encoders, and compute the cosine similarity between the output embeddings. That score indicates how likely it is that the action in the video sounds. For completeness, instead of defining a hard threshold for positives, we plot the ROC and precision-recall (PR) curves by varying the positive threshold, and calculate the area-under-curve (AUC) values for them—common metrics for classification [61, 277] that are invariant to the absolute score values. For both metrics, higher values are better, indicating the model learns meaningful embeddings of sounding actions. Similarly, we can also evaluate discovery

for audio-language, if narrations are available.

Results. Table 9.2 shows the results for sounding action discovery. We first look at discovery with audio-visual modalities alone at test time ("*AV*" columns). CM-ACC [175] discovers sounding actions much better than random chance, showing that audio-visual contrastive learning captures both visual action embeddings and action sound embeddings. CMC [281] and ImageBind [96] do better—benefiting (like us) from the language modality at training time. However, neither the joint nor sequential training paradigm exploits modality agreement, resulting in weak crossmodal constraints, and thus only marginal performance improvement. In comparison, our model MC3 explicitly models the modality consensus and improves the discovery result substantially by learning embeddings most relevant to sounding actions.

We also report the discovery result from using audio-language modalities ("*AL*" columns). Since narrations provide action specifications, the discovery performance is better than *AV*, e.g., CLAP [68] vs CM-ACC [175]. While CMC's [281] and ImageBind's [96] joint training results are not much better than CLAP [68], our model improves the "*AL*" discovery by leveraging the video modality and imposing the trimodal consensus constraint.

Fig. 9.5 plots the precision-recall curves. For the audio-visual curve, our model always has higher precision compared to baselines, especially when recall is low. This is strong evidence of our model learning features of sounding actions, whereas baselines are limited to capturing general audio-visual correspondence—whether action-based or not. We observe a similar trend for audio-language discovery.

Ablations. To study the importance of each loss and the two-stage training, we first ablate the consensus loss in the second stage ("w/o $\mathcal{L}_{\text{consensus}}$" in Table 9.2), which trains the model contrastively for both stages. The model performance drops significantly, showing that exploring the modality consensus is key to learning how actions sound. We then ablate the contrastive loss in the second stage ("w/o $\mathcal{L}_{\text{contrastive}}$"), which harms performance even more. This suggests that $\mathcal{L}_{\text{consensus}}$ functions like a

Figure 9.6: Example visual embedding cluster from our model

regularization term that forces the $\mathcal{L}_{\text{contrastive}}$ to learn sounding action embeddings. Lastly, we ablate the two-stage training strategy by removing the align stage ("w/o align-stage"), which optimizes $\mathcal{L}_{\text{MC3}}$ directly; this model fails badly. Aligning embeddings first is critical to making MC3's training stable.

Clustering. To visualize the learned embeddings, we group video embeddings in the test set with agglomerative clustering into 20 clusters. Fig. 9.6 shows the top 8 examples of one cluster. This cluster clearly captures the sound of water running. Not only does it group videos with similar actions that make this sound, but also it shows the learned embeddings are agnostic of the background (the bathroom example), and unbiased by the head/hand movement since the cluster has varying degrees of movement.

9.4.2 Sounding Action Retrieval

Retrieving a different modality given an action video, audio, or description is another useful application, such as adding sound effects to silent videos or retrieving captions for action sounds. To explore this setting, we answer the question "how do different actions sound?" by evaluating the crossmodal retrieval performance of long-tail sounding actions that are in the same category. Different from the binary

	$V \rightarrow A$		$A \rightarrow V$		$L \rightarrow A$		$A \rightarrow L$	
	@5	@10	@5	@10	@5	@10	@5	@10
Random	0.1	0.1	0.1	0.1	0.1	0.1	0.1	0.1
CLAP [68]	-	-	-	-	**49.8**	87.6	34.0	67.1
CM-ACC [175]	34.6	63.5	30.9	57.7	-	-	-	-
CMC [281]	36.5	67.9	33.8	63.7	44.1	81.8	32.8	64.3
ImageBind [96]	32.8	61.5	29.7	57.9	42.6	76.5	30.6	60.5
w/o $\mathcal{L}_{\text{consensus}}$	33.9	63.0	30.0	56.1	45.0	84.7	32.9	65.8
w/o $\mathcal{L}_{\text{contrastive}}$	3.3	3.7	6.4	12.5	3.1	4.7	3.3	8.0
w/o align-stage	10.0	19.4	5.9	11.8	11.6	20.9	6.5	12.6
MC3	**38.4**	**72.8**	**34.4**	**66.3**	46.2	**88.5**	**37.5**	**73.8**

Table 9.3: Sounding action retrieval. We report *Recall @5 and @10* for different query-retrieval modalities.

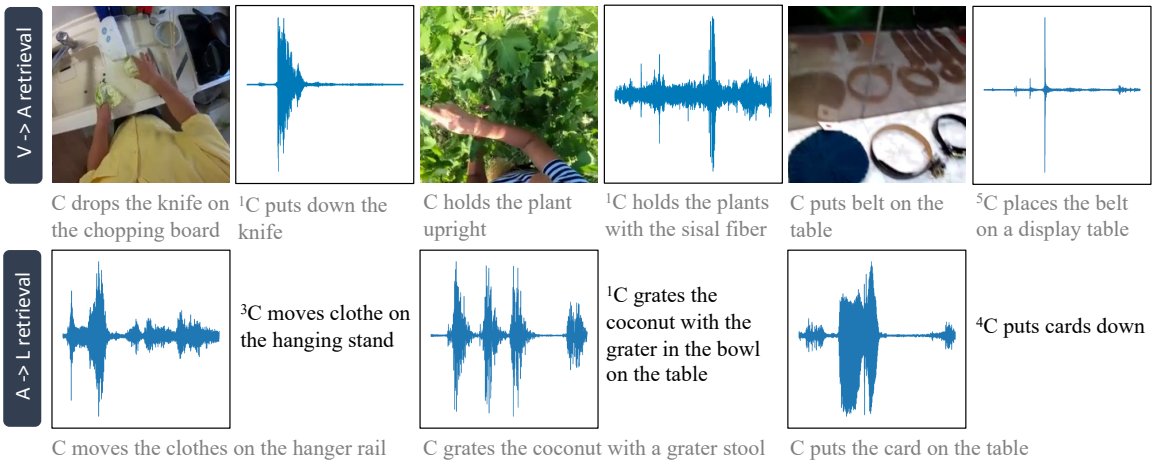


Figure 9.7: Qualitative examples for retrieval. The first row is video-to-audio retrieval, motivated by adding audio effects for silent videos. The second row is audio-to-text retrieval, motivated by audio captioning applications. For each row, we show three correct retrieval examples along with their text (gray indicates the text is not observed by the model). For the retrieved item, we show the ground truth rank as the superscript. All examples are long-tail sounding actions, showing how our model learns to capture the features of how actions sound.

classification task above, here we aim to retrieve other examples of the same action.

To do this, we utilize the action groups constructed in Sec. 9.3 based on verbs and nouns, and only keep groups that have more than two instances of sounding

actions (such that there will be at least one true positive to retrieve for each query). We then divide each action group equally into a query pool of 7,559 examples and a retrieval pool of 7,032 examples. Given a query modality M_i of instance A, we compute its distance to other modalities M_j of all instances in the retrieval pool. A retrieval is correct if the retrieved instance B and A belong to the same action group.

Results. Table 9.3 shows the results for four different query-retrieval modality settings. For audio-visual retrieval, we observe that all models can retrieve video with audio (or audio with video) for similar actions with much higher recall than random chance. Our model strongly outperforms the baselines and ablations, benefiting from modeling the modality consensus explicitly. We also observe that retrieving audio with video is easier than the opposite, likely because audio can be vague sometimes, e.g., a collision sound might occur due to various actions while seeing a cutting action indicates the likely sound. For audio-language retrieval, our model similarly outperforms the baselines by large margins.

Qualitative examples. In Fig 9.7, we show examples for video-to-audio and audio-to-language retrieval. Even though these actions are subtle, our model retrieves audio or captions that are very relevant.

9.4.3 Audio Classification on EPIC-Sounds

Finally, we evaluate our learned representation on a standard audio benchmark. To assess the impact of our model's action sounds representation, we consider EPIC-Sounds [121], a challenging audio classification benchmark for sounds in kitchen environments. To our knowledge, EPIC-Sounds represents the only large-scale benchmark for audio in egocentric video. Note, this classification task is different from the sounding action discovery task in Sec. 9.4.1 in that here the model only takes an audio clip as input.

We consider both linear-probe and fine-tuning settings. In the linear-probe setup, we freeze the model weights and only train the last classification layer, which

		Top-1	Top-5	mCA	MAP	mAUC
Random	-	7.71	30.95	2.29	0.023	0.500
ASF [138]*	L	45.53	79.33	13.48	0.172	0.789
SSAST [100]	L	28.74	64.84	7.14	0.079	0.755
MC3	L	42.44	78.76	12.79	0.153	0.818
ASF [138]*	F	53.75	84.54	20.11	**0.254**	0.873
SSAST [100]	F	53.47	84.56	20.22	0.235	0.879
MC3	F	**55.97**	**85.86**	**21.65**	0.242	**0.885**

Table 9.4: Results of classification on EPIC-Sounds. L: Linear-Probe; F: Fine-tuning. * denotes pretraining with supervised audio classification while the rest are pretrained in a self-supervised fashion.

evaluates the quality of the pre-trained representations. In the fine-tuning setup, we fine-tune both the encoder and the last layer.

Table 9.4 shows the results. We compare with two SotA methods reported in EPIC-Sounds: SSAST [100] and ASF [138]. SSAST is pretrained on LibriSpeech [214] and shares the *same network architecture* as ours, while ASF is trained on VGG-Sound with supervised learning. With linear-probe, our model strongly outperforms SSAST [100], which, like us, is also pretrained in a self-supervised fashion with no audio labels. ASF [138] does better than both, likely due to its advantage of supervised audio classification pretraining. When fine-tuning, our model outperforms both prior methods in all but one metric when following the same fine-tuning and evaluation protocol. This shows our MC3 audio encoder—trained for sounding action discovery—learns generalizable action sound embeddings, improving the state of the art. The margins are naturally smaller in the fine-tuning regime, as is typical, since all models have time to adapt to the new domain.

9.5 Conclusion

In this chapter, we explored the problem of learning how first-person actions sound from in-the-wild, narrated egocentric videos—without audio labels. Training

with 250K clips from Ego4D, we show the promise of our novel multimodal consensus framework for accurately aligning representations to capture the long-tail of sounding actions in novel (unnarrated) videos, with clear impact on sounding action discovery, retrieval, and pre-training for audio classification.

While this is very exciting, there are also some limitations. First of all, the most notable limitation is that we rely on synchronized narration data. Although both Ego4D and EPIC-KITCHENS have such data, they do not come for free. This limits our method from being applied to a wider range of data. One possible way to address this might be using some foundational vision-language model to provide captions thus the semantic grounding of correspondence.

Another limitation of this work is that contrastive learning preserves the semantic information of the sound but not the temporal information. Thus for video-to-audio generation, while the retrieved sound might be semantically relevant, the sound is not guaranteed to have temporal consistency with the video. To lift this limitation, we introduce a generative method to generate both semantically and temporally matching sounds in the next chapter.

Chapter 10: Action2Sound: Ambient-Aware Generation of Action Sounds from Egocentric Videos

In Chapter 9, I introduced a self-supervised approach for learning the embeddings of actions that make sounds. Building on the ideas of SoundingActions, I expand the scope to go from discovering action-sound associations, to actually generating the sounds that could go with a given visual action in video. The task offers a complementary way to study the fundamental problem of audio-visual actions and understand the link from action to sound, and it also has various possible applications, such as creating sound effects for films or virtual reality games. To tackle the generation problem for egocentric action videos, in this chapter, I introduce a generative method that takes a silent video and generates temporally and semantically meaningful sounds for the video.

As discussed in the previous chapter, generating impact/action sounds is important for many real-world applications, such as text-to-video generation, generating sound effects for films (Foley), or sound effect generation for virtual reality (VR) and video games. Some prior work studies impact sound synthesis from videos [212, 268] while others target more general video-to-audio generation [123, 174]. All these methods *implicitly assume total correspondence between the video and audio* and aim to generate the whole target audio from the video. However, this strategy falls short for in-the-wild training videos, which are rife with off-screen ambient sounds, e.g., traffic noise, people talking, or A/C running. While some of these ambient sounds are weakly correlated with the visual scene, such as the wind blowing in an outdoor environment, many of them have no visual correspondence, such as off-screen speech or a stationary buzzing noise from the fridge. Existing methods are not able to disentangle action sounds from ambient sounds and treat them as a whole, leading to uncontrolled generation of ambient sounds at test time and sometimes even halluci-

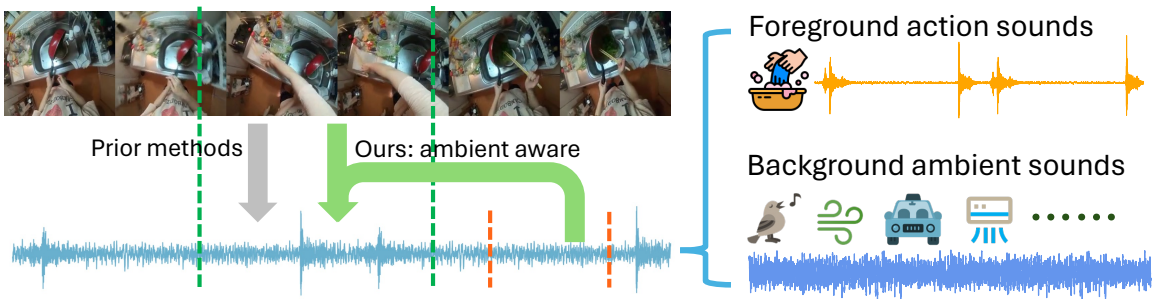


Figure 10.1: Real-world audio consists of both foreground action sounds (whose causes are visible in the FoV) and background ambient sounds that are generated by sources offscreen. Whereas prior work is agnostic to this division when performing generation, our method is ambient-aware and disentangles action sound from ambient sound. Our key technical insight is how to train with in-the-wild videos exhibiting natural ambient sounds, while still learning to factor out their effects on generation. The green arrows reference how we condition generation on sound from a related, but time-distinct, video clip to achieve this.

nation, e.g., random action or ambient sounds. This is particularly problematic for generating action sounds because they are often subtle and transient compared to the ambient sounds. For example, trained in the traditional way, a model given a scene that looks like a noisy restaurant risks generating "restaurant-like" ambient sounds, while ignoring the actual movements and activities of the foreground actions, such as a person stirring their coffee with a metal spoon.

How can we disentangle the foreground action sounds from background ambient sounds for in-the-wild video data *without* ground truth separated streams? Simply applying a noise removal algorithm on the target audio does not work well since in-the-wild blind source separation of general sounds from a single microphone is still an open challenge [297]. The key observation we have is that while action sounds are highly localized in time, ambient sounds tend to persist across time. Given this observation, we propose a simple but effective solution to disentangle ambient and action sounds: during training, in addition to the input video clip, we also condition the generation model on an audio clip from the same long video as the input video clip but from different timestamps. See Fig. 10.1. By doing so, we lift the burden of

generating energy-dominating ambient sounds and encourage the model to focus on learning action cues from the visual frames to generate action sounds. At test time, we do not assume access to (even other clips of) the ground truth video/audio. Instead, we propose to retrieve an audio segment from the training set with an audio-visual similarity scoring model, inspired by recent ideas in retrieval-augmented generation (RAG) [140, 111, 163]. This benefits examples where the visual scene has a weak correlation with the ambient sound that is appealing to capture, e.g., outdoor environments.

Existing action sound generation work relies on either clean, manually-collected data that has a limited number of action categories [212, 268, 53], or videos crawled from YouTube based on predefined taxonomies [93, 46, 123]. To expand the boundary of action sound generation to in-the-wild human actions, we take advantage of recent large-scale egocentric video datasets [105, 57]. Though our model is not tailored to egocentric video in any way, there are two main benefits of using these datasets: 1) egocentric videos provide a close view of human actions compared to exocentric videos, where hand-object interactions are much smaller from a distance and often occluded, and 2) these datasets have timestamped narrations describing atomic actions. We design an automatic pipeline to extract and process clips from Ego4D, and curate Ego4D-Sounds with 1.2 million audio-visual action clips.

Our idea of disentangling action and ambient sounds implicitly in training is model-agnostic. In this chapter, we instantiate it by designing an audio-visual latent diffusion model (AV-LDM) that conditions on both modality streams for audio generation. We evaluate our AV-LDM against recent methods on a wide variety of metrics and show that our model outperforms the existing methods significantly on both Ego4D-Sounds and EPIC-KITCHENS. We conduct a human evaluation study that shows our model synthesizes plausible action sounds according to the video. We also show promising preliminary results on virtual reality game clips. To the best of our knowledge, this is the first work that demonstrates the disentanglement of

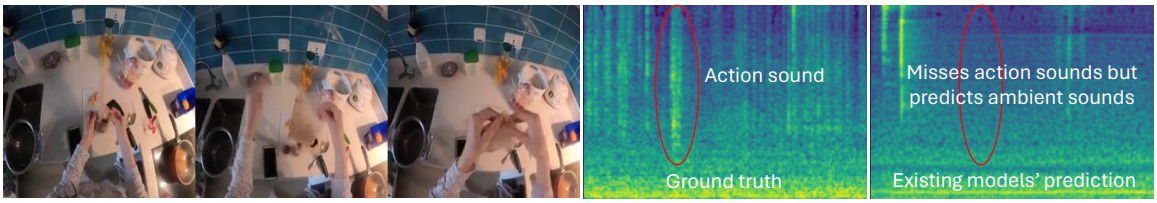


Figure 10.2: Illustration of the harm of ambient sound in video-to-audio generation. In this example, this person is closing a packet of ginger powder, which makes some rustling sound (red circled in the middle). There is also some buzzing sound semantically irrelevant to the visual scene in the background, which dominates the energy of the spectrogram. On the right-hand side, we show a prediction made by a vanilla model that misses the action sound but predicts the ambient sound.

foreground action sounds from background sounds for action-to-sound generation on in-the-wild videos.

I first introduce our proposed approach in Sec. 10.1, then present the dataset in Sec. 10.2, and lastly discuss the experiments in Sec. 10.3.

10.1 Ambient-aware Action Sound Generation

We first discuss our high-level idea of how to guide the generation model to disentangle action sounds from ambient sounds. We then extend the latent diffusion models (LDM) to accommodate both audio and video conditions, which we name AV-LDM. We also discuss our pretraining stage.

10.1.1 Action-to-Sound Generation

Given a video $V \in \mathbb{R}^{(T*S_V)\times H\times W\times 3}$, where T is the duration of the video and S_V is the video sample rate, and the accompanying audio waveform $A \in \mathbb{R}^{1\times(T*S_A)}$, where S_A is the audio sample rate, our goal is to model the conditional distribution $p(A|V)$ for video-to-audio generation. During training we observe natural video coupled with its audio, whereas at inference time we have only a silent video—e.g., could be an output from text-to-video generation, or a VR/video game clip, or simply a real-world video for which we want to generate new plausible sounds.

10.1.2 Disentangling Action and Ambient Sounds

Learning a video-to-audio generation model using in-the-wild egocentric videos is challenging because of entangled foreground action and background ambient sounds, as illustrated in Fig. 10.2. More specifically, the reasons are two-fold: 1) while action sounds are usually of very short duration, ambient sounds can last the entire clip, and therefore dominate the loss, leading to low-quality action sound generation; 2) while some ambient sounds might be semantically related to the visual scene such as bird chirping in the woods, in many cases, ambient sounds are difficult to infer from the visual scene because they are the results of the use of certain microphones, recording conditions, people speaking, off-screen actions, etc. Forcing a generation model to learn those background sounds from video results in hallucinations during inference (see examples in Fig. 10.6).

Therefore, it is beneficial to proactively disentangle action sounds and ambient sounds during training. However, separating in-the-wild ambient sounds is still an open challenge as recent models rely on supervised training on artificially mixed sounds, for which the ground truth complex masks can be obtained [297]. Simply applying off-the-shelf noise reduction methods to training data leads to poor performance, as we will show in Sec. 10.3.

While it is difficult to *explicitly* separate the ambient and action sound in the target audio, our key observation is that ambient sounds are usually fairly stationary across time. Given this observation, we propose a simple but effective method to achieve the disentanglement. During training, in addition to video clip V, we also provide the model an audio clip A_n that comes from the same training video but a different timestamp as the input video clip (see Fig. 10.3). Therefore, instead of modeling $p(A|V)$, we model $p(A|V, A_n)$. Given the hypothesis that A_n is likely to share ambient sound characteristics with A, it can take away the burden of learning weakly correlated or even uncorrelated ambient sounds from visual input alone, and encourages the model to focus on learning action features from the visual input. For

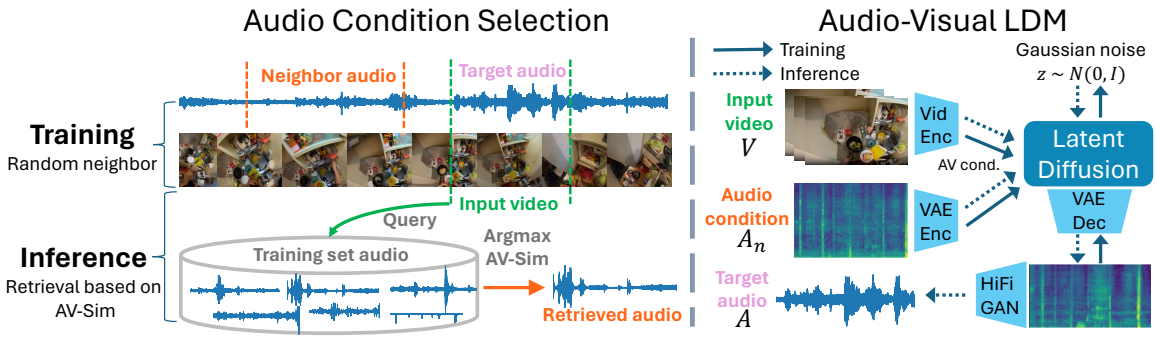


Figure 10.3: Audio condition selection and the model architecture. **Left**: During training, we randomly sample a neighbor audio clip as the audio condition. For inference, we query the training set audio with the (silent) input video and retrieve an audio clip that has the highest audio-visual similarity with the input video using our trained AV-Sim model (Sec. 10.1.5). **Right**: We represent audio waveforms as spectrograms and use a latent diffusion model to generate the spectrogram conditioned on both the input video and the audio condition. At test time, we use a trained vocoder network to transform the spectrogram to a waveform.

the selection of A_n, we randomly sample one audio clip from the nearest X clips in time. While there is no guarantee that the sampled audio shares exactly the same ambient sound with the target audio, their ambient sounds should largely overlap since they are close in time, which provides a consistent learning signal to help the model learn the disentanglement.

10.1.3 Retrieval Augmented Generation and Controllable Generation

While during training we have access to the clips in the same long video as the input clip, we of course cannot access that information at test time. How we select A_n at test time depends on the purpose of the generation. We consider two use cases: *action-ambient joint* generation and *action-focused* generation. In the first scenario, we would like the model to generate both the action sound and the ambient sound that is plausible for the visual environment. This is, for example, useful for generating sound effects for videos. In the latter scenario, we would like the model to focus the generation on action sounds and *minimize* ambient sounds, which is useful, for example, for generating sounds for games. Fig. 10.4 depicts the two scenarios.

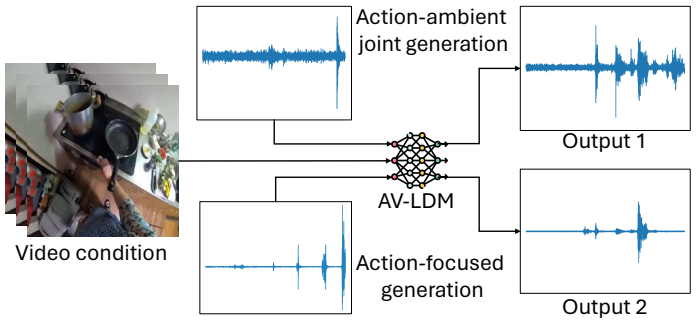


Figure 10.4: Two inference settings: "action-ambient joint generation" and "action-focused generation". In the first setting, we condition on audio retrieved from the training set and aim to generate both plausible action and ambient sounds. In the second setting, we specify an audio file with low ambient sound and the model focuses on generating plausible action sounds while minimizing the ambient sounds.

For action-ambient joint generation, we want A_n to be semantically relevant to the visual scene. Inspired by recent work in retrieval augmented regeneration, we propose to retrieve audio such that:

$$A_n = \arg\max_{A_i \in \mathcal{D}} \text{AV-Sim}(A_i, V), \tag{10.1}$$

where $\mathcal{D}$ is the dataset of all training audio clips and V is the (silent) input video. AV-Sim(A, V) is a similarity scoring function that measures the similarity between A and V, which we will cover in Sec. 10.1.5.

For action-focused generation, we want A_n to have minimal ambient level. We find simply filling A_n with all zeros results in poor performance, likely because it is too far out of the training distribution. Instead, we find conditioning the generation on a low-ambient sound will hint the model to focus on action sound generation and generate minimal ambient sound. See Sec. 10.3.3.

10.1.4 Audio-Visual Latent Diffusion Model

While the above idea of disentanglement is universal and not specific to any model architecture, here we instantiate this idea on diffusion models due to their success in audio generation [167, 174]. We extend the latent diffusion model to ac-

commodate our audio-visual conditions, thus yielding an audio-visual latent diffusion model (AV-LDM).

Fig. 10.3 (right) shows the architecture of our model. During training, given audio waveform target A, we first compute the mel-spectrogram $x_0 \in \mathbb{R}^{T \times D_{\text{mel}}}$, where D_{mel} is the number of mel bins. We then use a pretrained Variational Autoencoder (VAE) to compress the mel-spectrogram x_0 to a latent representation $z_0 \in \mathbb{R}^{C' \times H' \times W'}$, where z_0 is the generation target of the LDM. We condition the generation on both the video feature $c_v \in \mathbb{R}^{T_v, D_c}$ and audio feature $c_a \in \mathbb{R}^{T_a, D_c}$. We extract the video feature with a pretrained video encoder (see Sec. 10.1.5) from V. We extract the audio feature from the audio condition A_n with the same VAE encoder and then transform the feature into 1-d vector with a multilayer perceptron (MLP).

Following [174], we use cross attention where the query is produced by z_t, which is the sample diffusion step t, and key and value are produced by $\text{concat}([\text{Pos}_v + c_v; \text{Pos}_a + c_a])$, where Pos denotes learnable positional embeddings. The model is trained with the denoising objective:

$$\mathcal{L} = \mathbb{E}_{t \sim \text{uniform}(1,T), z_0, \epsilon_t} \|\epsilon_t - \epsilon_\theta(\mathbf{x}_t, t, c_v, c_a)\|^2,$$

where ϵ_t is the standard Gaussian noise sampled for diffusion step t, and $\epsilon_\theta(\mathbf{x}_t, t, c_v, c_a)$ is the model estimation of it (θ represents model parameters).

The reverse process can be parameterized as:

$$p(z_T) = \mathcal{N}(0, I),$$
$$p_\theta(z_{t-1}|z_t) = \mathcal{N}(z_{t-1}; \frac{1}{\sqrt{\alpha_t}}\Big(z_t - \frac{1-\alpha_t}{\sqrt{1-\bar{\alpha}_t}}\epsilon_\theta(z_t, t, c_v, c_a)\Big), \sigma_t^2 I),$$

where α_t and σ_t are determined by noise schedule of the diffusion process. To generate audio during inference, we first sample standard Gaussian noise z_T, and then apply classifier free guidance [119] to estimate $\tilde{\epsilon}_\theta$ as

$$\tilde{\epsilon}_t(z_t, t, c_v, c_a) = \omega\epsilon_\theta(z_t, t, c_v, c_a) + (1-\omega)\epsilon_\theta(z_t, t, \emptyset, \emptyset),$$

where $\emptyset$ denotes zero tensor. For the above estimation to be more precise, during training, we randomly replace c_v with $\emptyset$ with probability 0.2. As for c_a, we found dropping it even with even a small probability harms the performance, and therefore we always condition the LDM with c_a.

During inference, we use DPM-Solver [172] on LDM to sample a latent representation, which is then upsampled into a mel-spectrogram by the decoder of VAE. Lastly, we use a vocoder (HiFi-GAN [151]) model to generate waveform from the mel-spectrogram.

10.1.5 Audio-Visual Representation Learning

Generating semantically and temporally synchronized action sounds from video requires the video encoder to capture these relevant features. In addition, we would like to train a video model and an audio model whose representations align in the embedding space to support retrieval-augmented generation discussed in Sec. 10.1.3. For this purpose, we train a video encoder and audio encoder contrastively to optimize the following objective:

$$\text{AV-Sim}(A, V) = -\frac{1}{|\mathcal{B}|} \sum_{t \in \mathcal{B}} \log \frac{\exp(e_A^t e_V^t / \tau)}{\sum_{l \in \mathcal{B}} \exp(e_A^t e_V^l / \tau)},$$

where $\mathcal{B}$ is the current batch of data, e_A^t and e_V^t are normalized embeddings of the audio and video features, τ is a temperature parameter. To leverage the full power of narrations on Ego4D, we initialize the video encoder weights from models pre-trained on video and language from [165].

10.1.6 Implementation Details

We use Ego4D-Sounds (see Sec. 10.2) to train our AV-LDM. Video is sampled at 5FPS and audio is sampled at 16kHz. Video is passed through the pre-trained video encoder to produce condition features $c_v \in \mathbb{R}^{16 \times 768}$. The audio waveform is transformed into a mel-spectrogram with a hop size of 256 and 128 mel bins. The mel-spectrogram is then passed to the VAE encoder with padding in the temporal

Figure 10.5: Example clips in Ego4D-Sounds. We show one video frame, the action description, and the sound for each example. Note how these actions are subtle and long-tail, usually not present in typical video datasets.

Datasets	Clips	Language	Action Types
The Greatest Hits [212]	46.6K	✗	Hit, scratch, prod
VGG-Sound [46]	200K	Video tags	Not action-specific
EPIC-SOUNDS [122]	117.6K	Audio labels	Kitchen actions
Ego4D-Sounds	1.2M	Action narrations	In-the-wild actions

Table 10.1: Comparison with other audio-visual action datasets. Ego4D-Sounds not only has one order of magnitude more clips, but it is also coupled with language descriptions, supporting evaluation of sound generation based on semantics.

dimension to produce target $z_0 \in \mathbb{R}^{4\times16\times24}$. The audio condition is processed the same way except that we use an additional MLP to process VAE's output to produce $c_a \in \mathbb{R}^{24\times768}$. We load the weights of VAE and LDM from the pretrained Stable Diffusion to speed up training, similar to [174], and VAE is kept frozen during training. LDM is trained for 8 epochs with batch size 720 on Ego4D-Sounds with the AdamW optimizer with learning rate $1e-4$. During inference, we use 25 sampling steps with classifier-free guidance scale $\omega = 6.5$. For HiFi-GAN, we train it on a combination of 0.5s segments from Ego4D[105], Epic-Kitchens [122], and AudioSet [93]. We use AdamW to train HiFi-GAN with a learning rate of $2e-4$ and batch size of 64 for 120k steps. We set the number of random nearby audio samples $X = 6$.

10.2 The Ego4D-Sounds Dataset

Next we describe our efforts to curate Ego4D-Sounds, an audio-video dataset for human action sound generation. Our goal is to curate a high-quality dataset for action-audio correspondence for action-to-sound generation, addressing the issue of limited action types in the existing impact sound datasets [212, 54].

Ego4D [105] is an existing large-scale egocentric video dataset that has more than 3,600 hours of video recordings depicting hundreds of daily activities; 2,113 of those hours have audio available. It also has time-stamped narrations that are free-form sentences describing the current activity performed by the camera-wearer. We first utilize the narration timestamps in Ego4D to extract clips. However, not all clips have meaningful action sounds and there are many actions like "talk with someone", "look around", "turn around" that have low audio-visual correspondence. We then use an automatic pipeline to process all extracted clips to create the Ego4D-Sounds dataset, which has 1.2 million audio-visual action clips. Similarly, for the test set, we curate 11k clips for evaluation. We show examples in Fig. 10.5 and comparison with other datasets in Tab. 10.1.

For all resulting clips, we extract them as 3s clips with 224×224 image resolution at 30 FPS. For audio, we extract them as a single channel with a 16000 sample rate.

10.3 Experiments

In this section, we first present the evaluation metrics, and then the results on Ego4D-Sounds along with human evaluation. We also discuss results on EPIC-KITCHENS, and qualitative results on VR games.

10.3.1 Evaluation

To evaluate the performance of our model, we use the following metrics:

1. Fréchet Audio Distance (FAD) [141]: evaluates the quality of generated audio

clips against ground truth audio clips by measuring the similarity between their distributions. We use the public pytorch implementation. *

2. Audio-visual synchronization (AV-Sync) [174]: a binary classification model that classifies whether the video and generated audio streams are synchronized. Following [174], we create negative examples by either shift audio temporally or sample audio from a different video clip.

3. Contrastive language-audio contrastive (CLAP) scores [310]: evaluates the semantic similarity between the generated audio and the action description. We finetune the CLAP model † on the Ego4D-Sounds data and compute scores for the generated audio and the narration at test time.

These metrics measure different aspects of generation collectively, including the distribution of generated samples compared to the ground truth clips, synchronization with the video, and the semantic alignment with the action description.

We compare with the following baseline methods:

1. Retrieval: we retrieve the audio from the training set using the AV-Sim model introduced in Sec. 10.1.5. This method represents retrieval-based generation models such as ImageBind [96].

2. Spec-VQGAN [123]: a video-to-audio model that generates audio based on a codebook of spectrograms. We run their pre-trained model on our test set.

3. Diff-Foley [174]: a recent LDM-based model. We follow their fine-tuning steps on egocentric videos to train on our dataset.

Neither learning-based model has the ability to tackle the ambient sound, whereas our model disentangles it from the action sound.

*https://github.com/gudgud96/frechet-audio-distance
†https://github.com/LAION-AI/CLAP

	FAD ↓	AV-Sync (%)↑	CLAP↑
Ground Truth (Upper Bound)	0.0000	77.69	0.2698
Retrieval	1.8353	11.84	0.0335
Spec-VQGAN [123]	3.9017	7.12	0.0140
Diff-Foley [174]	3.5608	5.98	0.0346
Ours w/o vocoder	4.9282	29.60	0.1319
Ours w/o cond + denoiser	1.4676	1.09	0.0009
Ours w/o cond	1.4681	39.63	0.1418
Ours w/ random test cond	1.0635	28.74	0.1278
AV-LDM (Ours)	**0.9999**	**45.74**	**0.1435**

Table 10.2: Results on Ego4D-Sounds test set. We also report the performance of the ground truth audio, which gives the upper bound value for each metric.

In addition, we also evaluate the following ablations: "w/o vocoder": we replace the trained HiFi-GAN vocoder with Griffin-Lim; "w/o cond": we remove the audio condition at training time; "w/o cond + denoiser": we use an off-the-shelf model to denoise the target audio [‡]; "w/ random test cond": we use random audio from the training set as the condition instead of retrieving audio with the highest AV-Sim score.

10.3.2 Results on Ego4D-Sounds

In this section, we evaluate the ambient-sound joint generation setting with retrieval augmented generation. The results are shown in Tab. 10.2. Compared to all three baselines, we outperform them on all three metrics by a large margin. While the Retrieval baseline retrieves natural sounds from the training set and has a low FAD score compared to Spec-VQGAN and Diff-Foley, both its AV-Sync accuracy and CLAP scores are very low. Diff-Foley has a higher performance than Spec-VQGAN since it has been trained on this task, but it still largely underperforms our model w/o cond, likely because their video features do not generalize to the egocentric setting well.

[‡] `https://github.com/timsainb/noisereduce`

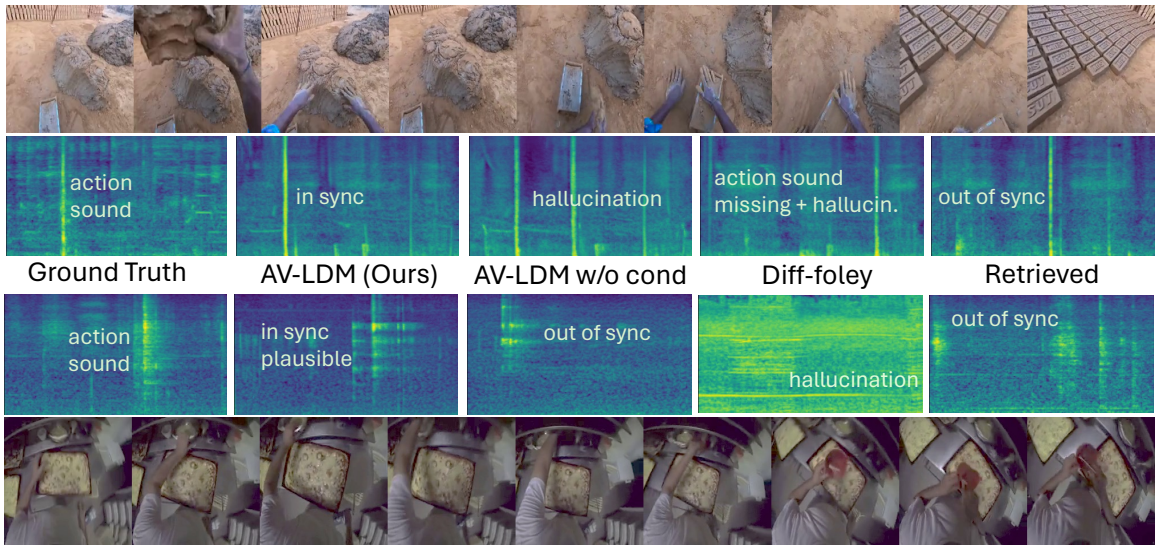


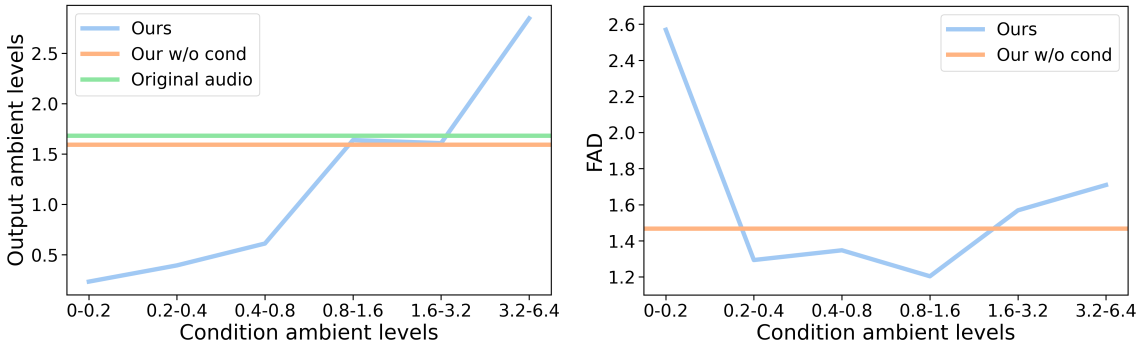
Figure 10.6: Qualitative example. We show the frames of each video followed by the waveform/spectrogram of various baseline methods. Our model generates the most synchronized sounds.

For ablations, "Ours w/o cond" has a much worse FAD score compared to the full model, showing the importance of our ambient-aware training. As expected, "Ours w/o cond + denoiser" has very low scores on AV-Sync and CLAP since existing noise reduction algorithms are far from perfect. We also test our model by conditioning it on a random audio segment at test time instead of the one retrieved with the highest audio-visual similarity and its performance also gets worse, verifying the effectiveness of our retrieval-based solution.

We show two qualitative examples in Fig. 10.6 comparing our model with several baselines and we show that our model synthesizes both more synchronized and more plausible sounds.

10.3.3 Ambient Sound Control

By disentangling action sounds from ambient sounds, our model allows taking any given sound as the condition at test time. To examine whether our model truly relies on the audio condition to learn the ambient sound information, we test the



(a) Varying ambient level condition (b) Audio generation accuracy (FAD)

Figure 10.7: The achieved ambient level and accuracy of the prediction as a function of the input ambient levels. (a): we show the ambient level of our model changes according to the ambient level in the audio condition while the ambient level of "Ours w/o cond" and the original audio stay constant, illustrating the controllability of our model. (b) FAD is low for most input ambient levels unless it goes too extreme (too low or too high), showing our model generates high-quality action sounds even when varying output ambient levels.

model by providing audio conditions of various ambient levels and then calculate the ambient level in the generated audio. The ambient level is defined as the lowest energy of any 0.5s audio segment in a 3s audio.

The results are shown in Fig. 10.7, where we also plot the ambient levels of "Ours w/o cond" and the original audio. Our model changes the ambient sound level according to the input ambient (shown in Fig. 10.7a) while still synthesizing plausible action sounds (shown in Fig. 10.7b). FAD spikes when the condition ambient is too low or too high, most likely because the generated ambient sound is out of distribution since the original audio always has some ambient sounds.

Fig. 10.8 shows example outputs from our model and several baselines. The examples show how our model generates plausible action sounds when conditioned on a low-ambient sound for action-focused generation. We can see that the action-focused setting generates similar action sounds as the action-ambient setting while having a minimal ambient level. While by definition we lack a good evaluation of this setting (there is no ground truth audio source separation for the data), our model

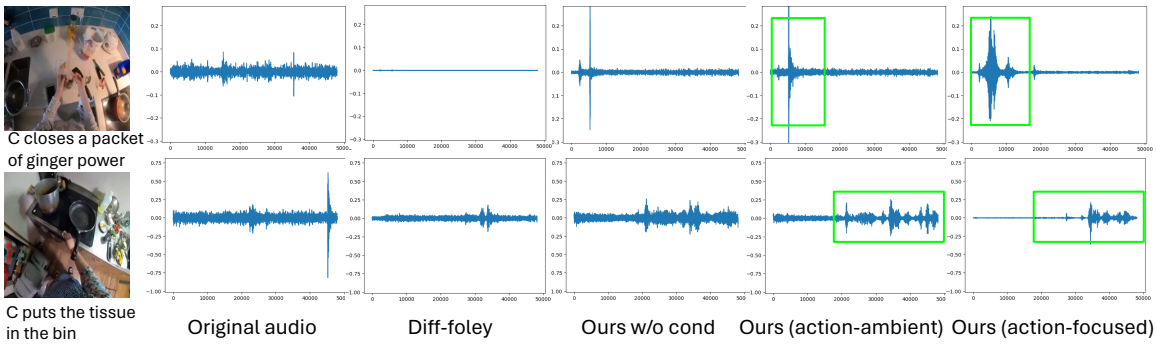


Figure 10.8: Visualization of action-focused generation. For both examples, Diff-Foley [174], Ours w/o cond or Ours (action-ambient generation) generate plausible action sounds along with ambient sounds. In contrast, our model conditioned on a low ambient sound generates plausible action sounds (see green boxes) with minimal ambient sound.

shows an emerging capability of generating clean action sounds although it has never been explicitly trained to do so.

10.3.4 Human Evaluation

To further validate the performance of various models, we conduct a subjective human evaluation. In each survey, we provide 30 questions and each question has 5 videos with the same visuals but different audio samples. For each video, we ask the participant to select the video(s) whose audio 1) is most semantically plausible and temporally synchronized with the video and 2) has the least ambient sounds. We invite 20 participants to complete the survey and compute the average voting for all 30 examples.

Tab. 10.3 shows the results. Overall, all learning-based methods generate reasonable action sounds while our model (action-ambient) has the highest score for action-sound quality compared to other methods. Although ours (action-focused) has a slightly lower action-sound score, it has significantly less ambient sound. This is likely because sometimes the low-ambient condition can lead the model to suppress some minor action sounds.

	Action sound quality	Least ambient sound
Retrieval	12.5%	12.5%
Diff-Foley [174]	47.5%	12.5%
AV-LDM w/o cond	55.0%	17.5%
AV-LDM (action-focused)	60.0%	**97.5%**
AV-LDM (action-ambient)	**72.5%**	22.5%

Table 10.3: Survey results showing user preferences. Higher is better. Our model in the action-ambient joint generation setting scores highest for action sound quality, showing its ability to produce action-relevant sounds despite training with in-the-wild data. Ours in the action-focused generation setting scores highest for the least ambient sound, at a slight drop in action sound quality score, showing the ability to eliminate background sounds when requested by the user.

	GT	Retrieval	Diff-Foley	Ours w/o cond	AV-LDM (Ours)
FAD ↓	0.0000	1.9618	3.4649	1.4731	**1.3200**
AV-Sync (%) ↑	73.94	13.84	14.19	50.42	**59.26**

Table 10.4: Results on Epic-Kitchens. GT stands for Ground Truth.

10.3.5 Results on EPIC-KITCHENS

To evaluate whether our model generalizes to other datasets, we also test our model on the EPIC-KITCHENS dataset. We first sample 1000 3s clips on EPIC-KITCHENS and then evaluate the retrieval baseline, Diff-Foley, Ours w/o cond, and our full model on these data and then compute the FAD and AV-Sync scores for them.

Results are shown in Tab. 10.4. Similar to what we observe on Ego4D-Sounds, our model outperforms other models on FAD and AV-Sync by a large margin, showing ours learns better to generate action sounds from visuals, which also transfer to other datasets.

10.3.6 Demo on VR Cooking Game

One compelling application of action-to-sound generation is to generate sound effects for games in virtual reality, where simulating complex hand-object interactions

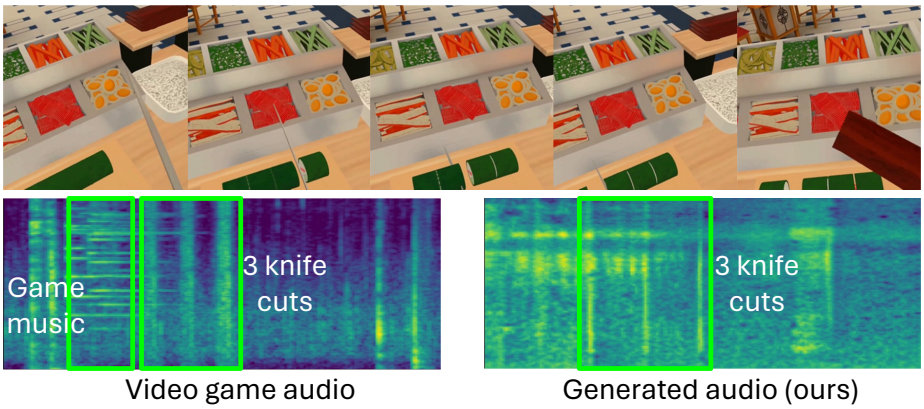


Figure 10.9: We apply our model on a VR cooking game clip where the person cuts a sushi roll three times. Our model successfully predicts the 3 cutting sounds.

is non-trivial. To examine whether our learned model generalizes to VR games, we collect game videos of a cooking VR game "Clash Of Chefs" from YouTube and test our model without fine-tuning. Preliminary results suggest our model can generate synced action sounds (see Fig. 10.9). This suggests a promising future in learning action-to-sound models from real-world egocentric videos and applying them to VR games to give a game user an immersive audio-visual experience that dynamically adjusts to their own actions.

10.4 Conclusion

We investigate the problem of generating sounds for human actions in egocentric videos. We propose an ambient-aware approach that disentangles the action sound from the ambient sound, allowing successful generation after training with diverse in-the-wild data, as well as controllable conditioning on ambient sound levels. We show that our model outperforms existing methods and baselines—both quantitatively and through human subject studies. Overall, it significantly broadens the scope of relevant training sources for achieving action-precise sound generation.

While we showed some working demos on VR games as a proof of concept, the model does not generalize to game videos robustly due to visual discrepancy. To transfer a generative model trained on the real data to the simulation data in a

zero-shot manner requires further investigation and research, e.g., by incorporating domain transfer techniques.

By conditioning the generation on a nearby audio segment, we disentangle ambient sounds from action sounds, assuming the conditioned audio and the target audio share ambient sound characteristics. In this chapter, a random neighbor works in most cases, but there are scenarios where a random neighbor would pick an audio segment that does not share the background sound. It is potentially helpful to use a learning-based method to intelligently pick the audio condition to further improve the performance.

Chapter 11: Conclusions and Future Work

In the preceding chapters, I presented my thesis research on 4D audio-visual learning. I covered topics on simulating realistic sounds in 3D environments, enabling embodied agents to move to find sounding objects while seeing and hearing, synthesizing audio with acoustics corresponding to the visual observations, and learning action sounds in egocentric videos. The key element in my research is the link between sounds and 3D scenes, i.e., how sounds are transformed by the environment and how 3D scenes are perceived with vision. Understanding this correspondence is essential to both the robotics and AR/VR applications I investigated.

While studying this correspondence in various applications, I have made the following important technical contributions that are applicable beyond my task settings and applications:

1. I devise a hierarchical reinforcement learning policy to improve navigation efficiency in Chapter 4. This policy shows the importance of decoupling high-level planning with low-level navigation for decision-making in active audio-visual perception, which is also applicable to other robotics tasks.

2. I design a transformer-based navigation policy in Chapter 5 that can locate a sounding object even after the sound stops. This demonstrates the potential of using transformers to model long-range context and serve as a memory for embodied agents, especially when dealing with highly dynamic audio signals.

3. I propose frequency-adaptive prediction that does better sim2real transfer for audio-visual navigation in Chapter 4. This shows the importance of investigating the spectral discrepancy for acoustic sim2real transfer and treating it accordingly rather than regarding all frequencies as the same.

4. I design the acoustic alteration strategy to create self-supervision to learn from YouTube videos in Chapter 7. This strategy addresses the data challenge for visual-acoustic learning and enables learning acoustic correspondence from Internet videos that do not have annotations or paired clean audio.

5. I use language to provide grounding for audio-visual correspondence learning in Chapter 9. This approach extends audio-visual learning from learning from curated datasets to learning from in-the-wild videos where there is no annotation indicating how visuals and sounds correspond.

6. I propose to disentangle ambient/action sounds for better video-to-audio generation in Chapter 10. This disentanglement allows controllable generation and also shows promising direction in learning from in-the-wild video data where sounds of interest and backgrounds are coupled.

While exciting first steps, there remain many open research problems that need further investigation. More specifically, I am interested in the following three problems: 1) guiding audio-visual generation with perceptual metrics, 2) collecting large-scale visual-acoustic datasets, and 3) the inverse rendering problem in acoustics. I will detail these problems below.

Learning with perpetual metrics. I introduced multiple generation algorithms in this thesis, including generating acoustics from images, novel-view acoustic synthesis, or action sound generation. These algorithms are typically optimized with objective loss functions such as Euclidean distance or some generative losses. These losses do not necessarily align with human perception of the sound, especially for binaural sounds. It is thus important to take into account human perception while learning the generation model.

Collecting large-scale real-world visual-acoustic data. One of the main motivations behind introducing SoundSpaces was to make it easier to study the correspondence between sight and sound in spaces in a clean and controllable setting because collecting such data in the real world is very expensive. SoundSpaces has enabled much research work to explore this correspondence and showed the potential of building machine-learning models that perceive the 3D space both visually and acoustically. However, there is some unavoidable domain gap when applying models trained on simulated data to real-world applications. To improve models' performance on these tasks, it is important to collect large-scale and high-quality data to train machine learning models in domain.

Inverse rendering. Generating sounds given visuals (either an image of the 3D environment or an egocentric video of human actions) is challenging but what about the inverse process? Can we infer the material properties of objects or the geometry properties of spaces from both audio and visual observations? Compared to generation, this is more difficult due to the lack of intermediate annotations or data. This poses a new challenge in both building learning models as well as collecting the right data for these tasks.

References

[1] Yang Ai and Zhen-Hua Ling. A neural vocoder with hierarchical generation of amplitude and phase spectra for statistical parametric speech synthesis. In *IEEE Transactions on Audio, Speech and Language Processing*, 2019.

[2] Hassan Akbari, Liangzhe Yuan, Rui Qian, Wei-Hong Chuang, Shih-Fu Chang, Yin Cui, and Boqing Gong. Vatt: Transformers for multimodal self-supervised learning from raw video, audio and text. In *NeurIPS*, 2021.

[3] Xavier Alameda-Pineda and Radu Horaud. Vision-guided robot hearing. *The International Journal of Robotics Research*, 2015.

[4] Andrew Allen and Nikunj Raghuvanshi. Aerophones in flatland: Interactive wave simulation of wind instruments. *ACM Transactions on Graphics (TOG) 34, 4 (2015), 1–11*, 2015.

[5] Jont B. Allen and David A. Berkley. Image method for efficiently simulating small-room acoustics. *J. Acoust. Soc. Am., vol. 65, no. 4, pp. 943–950,*, 1979.

[6] Humam Alwassel, Dhruv Mahajan, Bruno Korbar, Lorenzo Torresani, Bernard Ghanem, and Du Tran. Self-supervised learning by cross-modal audio-video clustering. In *NeurIPS*, 2020.

[7] P. Ammirato, P. Poirson, E. Park, J. Kosecka, and A. Berg. A dataset for developing and benchmarking active vision. In *ICRA*, 2016.

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

[9] Peter Anderson, Qi Wu, Damien Teney, Jake Bruce, Mark Johnson, Niko Sunderhauf, Ian Reid, Stephen Gould, and Anton Van Den Hengel. Vision-and-language navigation: Interpreting visually-grounded navigation instructions in real environments. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 3674–3683, 2018.

[10] Peter Anderson, Ayush Shrivastava, Joanne Truong, Arjun Majumdar, Devi Parikh, Dhruv Batra, and Stefan Lee. Sim-to-real transfer for vision-and-language navigation. In *CoRL*, 2020.

[11] Relja Arandjelović and Andrew Zisserman. Look, listen and learn. In *ICCV*, 2017.

[12] Relja Arandjelović and Andrew Zisserman. Objects that sound. In *ECCV*, 2018.

[13] I. Armeni, A. Sax, A. R. Zamir, and S. Savarese. Joint 2D-3D-Semantic Data for Indoor Scene Understanding. *ArXiv e-prints*, February 2017.

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

[15] 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.

[16] HE Bass, LC Sutherland, and AJ Zuckerwar. Atmospheric absorption of sound: Update. *The Journal of the Acoustical Society of America*, 88(4):2019–2021, 1990.

[17] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects, 2020.

[18] Dhruv Batra, Aaron Gokaslan, Aniruddha Kembhavi, Oleksandr Maksymets, Roozbeh Mottaghi, Manolis Savva, Alexander Toshev, and Erik Wijmans. Objectnav revisited: On evaluation of embodied agents navigating to objects. *arXiv preprint arXiv:2006.13171*, 2020.

[19] Jacob Benesty, Shoji Makino, and Jingdong Chen. *Speech Enhancement*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[20] M. Berouti, Richard Schwartz, and John Makhoul. Enhancement of speech corrupted by acoustic noise. In *IEEE International Conference on Acoustics, Speech, and Signal Processing*, 1979.

[21] Gedas Bertasius, Heng Wang, and Lorenzo Torresani. Is space-time attention all you need for video understanding? In *Proceedings of the International Conference on Machine Learning (ICML)*, July 2021.

[22] Stefan Bilbao. Modeling of complex geometries and boundary conditions in finite difference/finite volume time domain room acoustics simulation. *IEEE Transactions on Audio, Speech, and Language Processing*, 21(7):1524–1533, 2013.

[23] Stefan Bilbao, Charlotte Desvages, Michele Ducceschi, Brian Hamilton, Reginald Harrison-Harsley, Alberto Torin, and Craig Webb. Physical Modeling, Algorithms, and Sound Synthesis: The NESS Project. *MIT Press*, 2020.

[24] Joydeep Biswas and Manuela Veloso. Depth camera based indoor mobile robot localization and navigation. In *2012 IEEE International Conference on Robotics and Automation*, pages 1697–1702. IEEE, 2012.

[25] S. Brodeur, E. Perez, A. Anand, F. Golemo, L. Celotti, F. Strub, J. Rouat, H. Larochelle, and A. Courville. HoME: a Household Multimodal Environment. In *https://arxiv.org/abs/1711.11017*, 2017.

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

[27] Chunxiao Cao, Zhong Ren, Carl Schissler, Dinesh Manocha, and Kun Zhou. Interactive sound propagation with bidirectional path tracing. *ACM Transactions on Graphics (TOG)*, 35(6):1–11, 2016.

[28] Vincent Cartillier, Zhile Ren, Neha Jain, Stefan Lee, Irfan Essa, and Dhruv Batra. Semantic mapnet: Building allocentric semanticmaps and representations from egocentric views. *arXiv preprint arXiv:2010.01191*, 2020.

[29] Angel Chang, Angela Dai, Thomas Funkhouser, Maciej Halber, Matthias Niebner, Manolis Savva, Shuran Song, Andy Zeng, and Yinda Zhang. Matterport3d: Learning from rgb-d data in indoor environments. In *2017 International Conference on 3D Vision (3DV)*, pages 667–676, 2017.

[30] Matthew Chang, Arjun Gupta, and Saurabh Gupta. Semantic visual navigation by watching youtube videos. In *NeurIPS*, 2020.

[31] Devendra Singh Chaplot, Dhiraj Gandhi, Abhinav Gupta, and Ruslan Salakhutdinov. Object goal navigation using goal-oriented semantic exploration. In *NeurIPS*, 2020.

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

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

[34] 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.

[35] 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.

[36] Changan Chen, Ziad Al-Halah, and Kristen Grauman. Semantic audio-visual navigation. In *CVPR*, 2021.

[37] 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.

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

[39] Changan Chen, Ruohan Gao, Paul Calamia, and Kristen Grauman. Visual acoustic matching. In *CVPR*, 2022.

[40] Changan Chen, Carl Schissler, Sanchit Garg, Philip Kobernik, Alexander Clegg, Paul Calamia, Dhruv Batra, Philip W Robinson, and Kristen Grauman. Soundspaces 2.0: A simulation platform for visual-acoustic learning. In *NeurIPS*, 2022.

[41] Changan Chen, Alexander Richard, Roman Shapovalov, Vamsi Krishna Ithapu, Natalia Neverova, Kristen Grauman, and Andrea Vedaldi. Novel-view acoustic synthesis. In *CVPR*, 2023.

[42] Changan Chen, Wei Sun, David Harwath, and Kristen Grauman. Learning audio-visual dereverberation. In *ICASSP*, 2023.

[43] Changan Chen, Kumar Ashutosh, Rohit Girdhar, David Harwath, and Kristen Grauman. Soundingactions: Learning how actions sound from narrated egocentric videos. In *CVPR*, 2024.

[44] Changan Chen*, Puyuan Peng*, Ami Baid, Sherry Xue, Wei-Ning Hsu, David Harwath, and Kristen Grauman. Action2sound: Ambient-aware generation of action sounds from egocentric videos. In *arXiv*, 2024.

[45] Changan Chen*, Jordi Ramos*, Anshul Tomar*, and Kristen Grauman. Sim2real transfer for audio-visual navigation with frequency-adaptive acoustic field prediction. In *arXiv*, 2024.

[46] Honglie Chen, Weidi Xie, Andrea Vedaldi, and Andrew Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP*, 2020.

[47] Howard Chen, Alane Suhr, Dipendra Misra, Noah Snavely, and Yoav Artzi. Touchdown: Natural language navigation and spatial reasoning in visual street environments. In *CVPR*, 2019.

[48] Tao Chen, Saurabh Gupta, and Abhinav Gupta. Learning exploration policies for navigation. In *ICLR*, 2019.

[49] Ziyang Chen, Shengyi Qian, and Andrew Owens. Sound localization from motion: Jointly learning sound direction and camera rotation. In *ICCV*, 2023.

[50] Corey I. Cheng and Gregory H. Wakefield. Introduction to head-related transfer functions (hrtfs): representations of hrtfs in time, frequency, and space. *journal of the audio engineering society*, 49(4):231–249, april 2001.

[51] Hyeong-Seok Choi, Jang-Hyun Kim, Jaesung Huh, Adrian Kim, Jung-Woo Ha, and Kyogu Lee. Phase-aware speech enhancement with deep complex u-net. In *ICLR*, 2019.

[52] 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.

[53] Samuel Clarke, Negin Heravi, Mark Rau, Ruohan Gao, Jiajun Wu, Doug James, and Jeannette Bohg. Diffimpact: Differentiable rendering and identification of impact sounds. In *5th Annual Conference on Robot Learning*, 2021.

[54] Samuel Clarke, Ruohan Gao, Mason Wang, Mark Rau, Julia Xu, Jui-Hsien Wang, Doug L. James, and Jiajun Wu. Realimpact: A dataset of impact sound fields for real objects. In *Proceedings of the IEEE International Conference on Computer Vision and Pattern Recognition*, 2023.

[55] Erin C Connors, Lindsay A Yazzolino, Jaime Sánchez, and Lotfi B Merabet. Development of an audio-based virtual gaming environment to assist with navigation skills in the blind. *JoVE (Journal of Visualized Experiments)*, 2013.

[56] MMTracking Contributors. MMTracking: OpenMMLab video perception toolbox and benchmark. `https://github.com/open-mmlab/mmtracking`, 2020.

[57] Dima Damen, Hazel Doughty, Giovanni Maria Farinella, Sanja Fidler, Antonino Furnari, Evangelos Kazakos, Davide Moltisanti, Jonathan Munro, and Toby Perrett. Scaling egocentric vision: The epic-kitchens dataset. In *ECCV*, 2018.

[58] A. Das, S. Datta, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Embodied Question Answering. In *CVPR*, 2018.

[59] A. Das, G. Gkioxari, S. Lee, D. Parikh, and D. Batra. Neural Modular Control for Embodied Question Answering. In *ECCV*, 2018.

[60] Abhishek Das, Federico Carnevale, Hamza Merzic, Laura Rimell, Rosalia Schneider, Josh Abramson, Alden Hung, Arun Ahuja, Stephen Clark, Gregory Wayne, et al. Probing emergent semantics in predictive agents via question answering. In *ICML*, 2020.

[61] Jesse Davis and Mark Goadrich. The relationship between precision-recall and roc curves. In *ICML*, 2006.

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

[63] Alexandre Defossez, Gabriel Synnaeve, and Yossi Adi. Real time speech enhancement in the waveform domain. In *Interspeech*, 2020.

[64] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, pages 248–255. Ieee, 2009.

[65] James Eaton, Nikolay Gaubitch, Allistair Moore, and Patrick Naylor. Estimation of room acoustic parameters: The ACE challenge. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 24(10), 2016.

[66] M David Egan, JD Quirt, and MZ Rousseau. Architectural acoustics, 1989.

[67] David Eigen, Christian Puhrsch, and Rob Fergus. Depth map prediction from a single image using a multi-scale deep network. In *NIPS*, 2014.

[68] Benjamin Elizalde, Soham Deshmukh, Mahmoud Al Ismail, and Huaming Wang. Clap: Learning audio concepts from natural language supervision. *arXiv preprint arXiv:2206.04769*, 2022.

[69] Yariv Ephraim and Harry L. Van Trees. A signal subspace approach for speech enhancement. *IEEE Transactions on Speech and Audio Processing*, 1995.

[70] Ariel Ephrat, Inbar Mosseri, Oran Lang, Tali Dekel, Kevin Wilson, Avinatan Hassidim, William T Freeman, and Michael Rubinstein. Looking to listen at the cocktail party: A speaker-independent audio-visual model for speech separation. In *SIGGRAPH*, 2018.

[71] Ori Ernst, Shlomo E. Chazan, Sharon Gannot, and Jacob Goldberger. Speech dereverberation using fully convolutional networks. In *2018 26th European Signal Processing Conference (EUSIPCO)*, 201o.

[72] C. Evers and P. Naylor. Acoustic slam. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2018.

[73] Bernard Ghanem Fabian Caba Heilbron, Victor Escorcia and Juan Carlos Niebles. Activitynet: A large-scale video benchmark for human activity understanding. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 961–970, 2015.

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

[75] Maksim Filipenko and Ilya Afanasyev. Comparison of various slam systems for mobile robot in an indoor environment. In *2018 International Conference on Intelligent Systems (IS)*, pages 400–407. IEEE, 2018.

[76] Madeleine Fortin, Patrice Voss, Catherine Lord, Maryse Lassonde, Jens Pruessner, Dave Saint-Amour, Constant Rainville, and Franco Lepore. Wayfinding in the blind: larger hippocampal volume and supranormal spatial navigation. *Brain*, 2008.

[77] Szu-Wei Fu, Chien-Feng Liao, Yu Tsao, and Shou-De Lin. Metricgan: Generative adversarial networks based black-box metric scores optimization for speech enhancement. In *International Conference on Machine Learning (ICML)*, 2019.

[78] Szu-Wei Fu, Cheng Yu, Tsun-An Hsieh, Peter Plantinga, Mirco Ravanelli, Xugang Lu, and Yu Tsao. MetricGAN+: An improved version of MetricGAN for speech enhancement. In *Interspeech*, 2021.

[79] 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.

[80] Thomas Funkhouser, Ingrid Carlbom, Gary Elko, Gopal Pingali, Mohan Sondhi, and Jim West. A beam tracing approach to acoustic modeling for interactive virtual environments. In *Proceedings of the 25th annual conference on Computer graphics and interactive techniques*, 1998.

[81] Thomas Funkhouser, Nicolas Tsingos, Ingrid Carlbom, Gary Elko, Mohan Sondhi, James E West, Gopal Pingali, Patrick Min, and Addy Ngan. A beam tracing method for interactive architectural acoustics. *The Journal of the acoustical society of America*, 115(2):739–756, 2004.

[82] Hannes Gamper and Ivan J Tashev. Blind reverberation time estimation using a convolutional neural network. In *2018 16th International Workshop on Acoustic Signal Enhancement (IWAENC)*, pages 136–140, 2018.

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

[84] Chuang Gan, Jeremy Schwartz, Seth Alter, Martin Schrimpf, James Traer, Julian De Freitas, Jonas Kubilius, Abhishek Bhandwaldar, Nick Haber, Megumi Sano, Kuno Kim, Elias Wang, Damian Mrowca, Michael Lingelbach, Aidan Curtis, Kevin T. Feigelis, Daniel M. Bear, Dan Gutfreund, David D. Cox, James J. DiCarlo, Josh H. McDermott, Joshua B. Tenenbaum, and Daniel L. K. Yamins. Threedworld: A platform for interactive multi-modal physical simulation. In *NeurIPS Track on Datasets and Benchmarks*, 2021.

[85] Dhiraj Gandhi, Abhinav Gupta, and Lerrel Pinto. Swoosh! rattle! thump! - actions that sound. In *RSS*, 2022.

[86] Ruohan Gao and Kristen Grauman. 2.5d visual sound. In *CVPR*, 2019.

[87] Ruohan Gao and Kristen Grauman. Co-separating sounds of visual objects. In *ICCV*, 2019.

[88] Ruohan Gao and Kristen Grauman. VisualVoice: Audio-visual speech separation with cross-modal consistency. In *CVPR*, 2021.

[89] Ruohan Gao, Rogerio Feris, and Kristen Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018.

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

[91] Ruohan Gao, Tae-Hyun Oh, Kristen Grauman, and Lorenzo Torresani. Listen to look: Action recognition by previewing audio. In *CVPR*, 2020.

[92] Ruohan Gao, Hao Li, Gokul Dharan, Zhuzhu Wang, Chengshu Li, Fei Xia, Silvio Savarese, Li Fei-Fei, and Jiajun Wu. Sonicverse: A multisensory simulation platform for embodied household agents that see and hear. In *ICRA*, 2023.

[93] Jort F. Gemmeke, Daniel P. W. Ellis, Dylan Freedman, Aren Jansen, Wade Lawrence, R. Channing Moore, Manoj Plakal, and Marvin Ritter. Audio set: An ontology and human-labeled dataset for audio events. In *Proc. IEEE ICASSP 2017*, New Orleans, LA, 2017.

[94] Iliyan Georgiev. Implementing vertex connection and merging. *Technical Re-port. Saarland University.*, 2012.

[95] Rohit Girdhar and Kristen Grauman. Anticipative Video Transformer. In *ICCV*, 2021.

[96] Rohit Girdhar, Alaaeldin El-Nouby, Zhuang Liu, Mannat Singh, Kalyan Vasudev Alwala, Armand Joulin, and Ishan Misra. Imagebind: One embedding space to bind them all. In *CVPR*, 2023.

[97] Ritwik Giri, Michael L. Seltzer, Jasha Droppo, and Dong Yu. Improving speech recognition in reverberation using a room-aware deep neural network and multi-task learning. In *2015 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2015.

[98] Clément Godard, Oisin Mac Aodha, Michael Firman, and Gabriel J. Brostow. Digging into self-supervised monocular depth prediction. In *ICCV*, 2019.

[99] Yuan Gong, Yu-An Chung, and James Glass. AST: Audio Spectrogram Transformer. In *Proc. Interspeech 2021*, pages 571–575, 2021.

[100] Yuan Gong, Cheng-I Lai, Yu-An Chung, and James Glass. Ssast: Self-supervised audio spectrogram transformer. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2022.

[101] D. Gordon, A. Kembhavi, M. Rastegari, J. Redmon, D. Fox, and A. Farhadi. IQA: Visual Question Answering in Interactive Environments. In *CVPR*, 2018.

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

[103] 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.

[104] Anirudh Goyal, Riashat Islam, Daniel Strouse, Zafarali Ahmed, Matthew Botvinick, Hugo Larochelle, Yoshua Bengio, and Sergey Levine. Infobot: Transfer and exploration via the information bottleneck. In *ICLR*, 2019.

[105] Kristen Grauman, Andrew Westbury, Eugene Byrne, Zachary Chavis, Antonino Furnari, Rohit Girdhar, Jackson Hamburger, Hao Jiang, Miao Liu, Xingyu Liu, Miguel Martin, Tushar Nagarajan, Ilija Radosavovic, Santhosh Kumar

Ramakrishnan, Fiona Ryan, Jayant Sharma, Michael Wray, Mengmeng Xu, Eric Zhongcong Xu, Chen Zhao, Siddhant Bansal, Dhruv Batra, Vincent Cartillier, Sean Crane, Tien Do, Morrie Doulaty, Akshay Erapalli, Christoph Feichtenhofer, Adriano Fragomeni, Qichen Fu, Abrham Gebreselasie, Cristina Gonzalez, James Hillis, Xuhua Huang, Yifei Huang, Wenqi Jia, Weslie Khoo, Jachym Kolar, Satwik Kottur, Anurag Kumar, Federico Landini, Chao Li, Yanghao Li, Zhenqiang Li, Karttikeya Mangalam, Raghava Modhugu, Jonathan Munro, Tullie Murrell, Takumi Nishiyasu, Will Price, Paola Ruiz Puentes, Merey Ramazanova, Leda Sari, Kiran Somasundaram, Audrey Southerland, Yusuke Sugano, Ruijie Tao, Minh Vo, Yuchen Wang, Xindi Wu, Takuma Yagi, Ziwei Zhao, Yunyi Zhu, Pablo Arbelaez, David Crandall, Dima Damen, Giovanni Maria Farinella, Christian Fuegen, Bernard Ghanem, Vamsi Krishna Ithapu, C. V. Jawahar, Hanbyul Joo, Kris Kitani, Haizhou Li, Richard Newcombe, Aude Oliva, Hyun Soo Park, James M. Rehg, Yoichi Sato, Jianbo Shi, Mike Zheng Shou, Antonio Torralba, Lorenzo Torresani, Mingfei Yan, and Jitendra Malik. Ego4d: Around the world in 3,000 hours of egocentric video. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18973–18990, 2022.

[106] Daniel Griffin and Jae Lim. Signal estimation from modified short-time fourier transform. *IEEE Transactions on acoustics, speech, and signal processing*, 32 (2):236–243, 1984.

[107] Anmol Gulati, Chung-Cheng Chiu, James Qin, Jiahui Yu, Niki Parmar, Ruoming Pang, Shibo Wang, Wei Han, Yonghui Wu, Yu Zhang, and Zhengdong Zhang. Conformer: Convolution-augmented transformer for speech recognition. In *Interspeech*, 2020.

[108] Ryan Gunther, Rick Kazman, and Carolyn MacGregor. Using 3d sound as a navigational aid in virtual environments. *Behaviour & Information Technology*, pages 435–446, 2004.

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

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

[111] Kelvin Guu, Kenton Lee, Zora Tung, Panupong Pasupat, and Ming-Wei Chang. Realm: Retrieval-augmented language model pre-training. *ArXiv*, abs/2002.08909, 2020.

[112] Tuomas Haarnoja, Aurick Zhou, Pieter Abbeel, and Sergey Levine. Soft actor-critic: Off-policy maximum entropy deep reinforcement learning with a stochastic actor. In *ICML*, 2018.

[113] Kun Han, Yuxuan Wang, DeLiang Wang, William S. Woods, Ivo Merks, and Tao Zhang. Learning spectral mapping for speech dereverberation and denoising. In *ICASSP*, 2015.

[114] R. Hartley and A. Zisserman. *Multiple view geometry in computer vision*. Cambridge University Press, 2004.

[115] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *2016 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 770–778, 2016.

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

[117] Peter Henry, Michael Krainin, Evan Herbst, Xiaofeng Ren, and Dieter Fox. Rgb-d mapping: Using kinect-style depth cameras for dense 3d modeling of indoor environments. *The international journal of Robotics Research*, 31(5): 647–663, 2012.

[118] John R Hershey, Zhuo Chen, Jonathan Le Roux, and Shinji Watanabe. Deep clustering: Discriminative embeddings for segmentation and separation. In *ICASSP*, 2016.

[119] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022.

[120] Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Egocentric audio-visual object localization. In *CVPR*, 2023.

[121] Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, and Andrew Zisserman. EPIC-SOUNDS: A Large-Scale Dataset of Actions that Sound. In *IEEE International Conference on Acoustics, Speech, Signal Processing (ICASSP)*, 2023.

[122] Jaesung Huh, Jacob Chalk, Evangelos Kazakos, Dima Damen, and Andrew Zisserman. Epic-sounds: A large-scale dataset of actions that sound. In *ICASSP 2023 - 2023 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 1–5, 2023.

[123] Vladimir Iashin and Esa Rahtu. Taming visually guided sound generation. In *BMVC*, 2021.

[124] ISO. ISO 3382, Acoustics—Measurement of room acoustic parameters. *International Standards Organisation*, 2012.

[125] Ajay Jain, Matthew Tancik, and Pieter Abbeel. Putting nerf on a diet: Semantically consistent few-shot view synthesis. In *iccv*, 2021.

[126] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G. Schwing, and Aniruddha Kembhavi. Two body problem: Collaborative visual task completion. In *CVPR*, 2019. first two authors contributed equally.

[127] Unnat Jain, Luca Weihs, Eric Kolve, Mohammad Rastegari, Svetlana Lazebnik, Ali Farhadi, Alexander G Schwing, and Aniruddha Kembhavi. Two body problem: Collaborative visual task completion. In *CVPR*, 2019. first two authors contributed equally.

[128] Dinesh Jayaraman and Kristen Grauman. End-to-end policy learning for active visual categorization. *TPAMI*, 2018.

[129] Marco Jeub, Magnus Schafer, and Peter Vary. A binaural room impulse response database for the evaluation of dereverberation algorithms. In *2009 16th International Conference on Digital Signal Processing*, pages 1–5, 2009. doi: 10.1109/ICDSP.2009.5201259.

[130] Marco Jeub, Magnus Schäfer, Hauke Krüger, Christoph Matthias Nelke, Christophe Beaugeant, and Peter Vary. Do we need dereverberation for hand-held telephony? In *International Congress on Acoustics*, 2010.

[131] Mohit Bansal Jialu Li, Hao Tan. Envedit: Environment editing for vision-and-language navigation. In *CVPR*, 2022.

[132] Hao Jiang, Calvin Murdock, and Vamsi Krishna Ithapu. Egocentric deep multi-channel audio-visual active speaker localization. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 10544–10552, 2022.

[133] M. Johnson, K. Hofmann, T. Hutton, and D. Bignell. The malmo platform for artificial intelligence experimentation. In *Intl. Joint Conference on AI*, 2016.

[134] III Julius O. Smith. Physical modeling using digital waveguides. *Computer Music Journal Vol. 16, No. 4 (Winter, 1992), pp. 74-91 (18 pages)*, 1992.

[135] Abhishek Kadian, Joanne Truong, Aaron Gokaslan, Alexander Clegg, Erik Wijmans, Stefan Lee, Manolis Savva, Sonia Chernova, and Dhruv Batra. Sim2real

predictivity: Does evaluation in simulation predict real-world performance? In *RA-L*, 2020.

[136] Will Kay, João Carreira, Karen Simonyan, Brian Zhang, Chloe Hillier, Sudheendra Vijayanarasimhan, Fabio Viola, Tim Green, Trevor Back, Paul Natsev, Mustafa Suleyman, and Andrew Zisserman. The kinetics human action video dataset. *CoRR*, abs/1705.06950, 2017.

[137] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Epic-fusion: Audio-visual temporal binding for egocentric action recognition. In *2019 IEEE/CVF International Conference on Computer Vision (ICCV)*, pages 5491–5500, 2019.

[138] Evangelos Kazakos, Arsha Nagrani, Andrew Zisserman, and Dima Damen. Slow-fast auditory streams for audio recognition. In *IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 855–859, 2021.

[139] M. Kempka, M. Wydmuch, G. Runc, J. Toczek, and W. Jakowski. Vizdoom: A doom-based ai research platform for visual reinforce- ment learning. In *Proc. IEEE Conf. on Computational Intelligence and Games*, 2016.

[140] Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. Generalization through memorization: Nearest neighbor language models. *ArXiv*, abs/1911.00172, 2019.

[141] Kevin Kilgour, Mauricio Zuluaga, Dominik Roblek, and Matthew Sharifi. Fréchet audio distance: A metric for evaluating music enhancement algorithms. *arxiv*, 2018.

[142] Hansung Kim, Luca Remaggi, Sam Fowler, Philip JB Jackson, and Adrian Hilton. Acoustic room modelling using 360 stereo cameras. *IEEE Transactions on Multimedia*, 2020.

[143] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[144] Keisuke Kinoshita, Marc Delcroix, Sharon Gannot, Emanuël Habets, Reinhold Haeb-Umbach, Walter Kellermann, Volker Leutnant, Roland Maas, Tomohiro Nakatani, Bhiksha Raj, Armin Sehr, and Takuya Yoshioka. A summary of the reverb challenge: State-of-the-art and remaining challenges in reverberant speech processing research. *EURASIP Journal on Advances in Signal Processing*, 2016.

[145] Florian Klein, Annika Neidhardt, and Marius Seipel. Real-time estimation of reverberation time for selection of suitable binaural room impulse responses. In *Audio for Virtual, Augmented and Mixed Realities: Proceedings of 5th International Conference on Spatial Audio (ICSA 2019)*, pages 145–150, 2019.

[146] C. Knapp and G. Carter. The generalized correlation method for estimation of time delay. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1976.

[147] Tom Ko, Vijayaditya Peddinti, Daniel Povey, Michael L. Seltzer, and Sanjeev Khudanpur. A study on data augmentation of reverberant speech for robust speech recognition. In *ICASSP*, 2017.

[148] 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.

[149] Alexander Kolesnikov, Alexey Dosovitskiy, Dirk Weissenborn, Georg Heigold, Jakob Uszkoreit, Lucas Beyer, Matthias Minderer, Mostafa Dehghani, Neil Houlsby, Sylvain Gelly, Thomas Unterthiner, and Xiaohua Zhai. An image is worth 16x16 words: Transformers for image recognition at scale. In *ICLR*, 2021.

[150] Eric Kolve, Roozbeh Mottaghi, Winson Han, Eli VanderBilt, Luca Weihs, Alvaro Herrasti, Daniel Gordon, Yuke Zhu, Abhinav Gupta, and Ali Farhadi. AI2-THOR: An Interactive 3D Environment for Visual AI. *arXiv*, 2017.

[151] Jungil Kong, Jaehyeon Kim, and Jaekyoung Bae. Hifi-gan: Generative adversarial networks for efficient and high fidelity speech synthesis. *Advances in Neural Information Processing Systems*, 33:17022–17033, 2020.

[152] Junghyun Koo, Seungryeol Paik, and Kyogu Lee. Reverb conversion of mixed vocal tracks using an end-to-end convolutional deep neural network. In *ICASSP 2021-2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 81–85. IEEE, 2021.

[153] Bruno Korbar, Du Tran, and Lorenzo Torresani. Cooperative learning of audio and video models from self-supervised synchronization. In *NeurIPS*, 2018.

[154] Jacob Krantz, Erik Wijmans, Arjun Majumdar, Dhruv Batra, and Stefan Lee. Beyond the nav-graph: Vision-and-language navigation in continuous environments. In *ECCV*, 2020.

[155] Asbjørn Krokstad, S Strom, and Svein Sørsdal. Calculating the acoustical room response by the use of a ray tracing technique. *Journal of Sound and Vibration 8, 1 (1968), 118–125.*, 1968.

[156] Jonáš Kulhánek, Erik Derner, Torsten Sattler, and Robert Babuška. ViewFormer: NeRF-free neural rendering from few images using transformers. In *eccv*, 2022.

[157] Kundan Kumar, Rithesh Kumar, Thibault de Boissiere, Lucas Gestin, Wei Zhen Teoh, Jose Sotelo, Alexandre de Brebisson, Yoshua Bengio, and Aaron Courville. Melgan: Generative adversarial networks for conditional waveform synthesis. In *NeurIPS*, 2019.

[158] H. Kuttruff. *Room acoustics*. CRC Press, 2016.

[159] K Heinrich Kuttruff. Auralization of impulse responses modeled on the basis of ray-tracing results. *Journal of the Audio Engineering Society*, 1993.

[160] Sam Lapp, Tessa Rhinehart, Louis Freeland-Haynes, Jatin Khilnani, Alexandra Syunkova, and Justin Kitzes. Opensoundscape: An open-source bioacoustics analysis package for python. *Methods in Ecology and Evolution 2023*, 2023.

[161] A. Lerer, S. Gross, and R. Fergus. Learning physical intuition of block towers by example. In *ICML*, 2016.

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

[163] Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401, 2020.

[164] Zhengqi Li, Simon Niklaus, Noah Snavely, and Oliver Wang. Neural scene flow fields for space-time view synthesis of dynamic scenes. *corr*, abs/2011.13084, 2020.

[165] Kevin Qinghong Lin, Jinpeng Wang, Mattia Soldan, Michael Wray, Rui Yan, Zhongcong Xu, Difei Gao, Rong-Cheng Tu, Wenzhe Zhao, Weijie Kong, Chengfei Cai, Wang HongFa, Dima Damen, Bernard Ghanem, Wei Liu, and Mike Zheng Shou. Egocentric video-language pretraining. In *Advances in Neural Information Processing Systems*, 2022.

[166] Yan-Bo Lin and Yu-Chiang Frank Wang. Audiovisual transformer with instance attention for audio-visual event localization. In *ACCV*, 2020.

[167] Haohe Liu, Zehua Chen, Yiitan Yuan, Xinhao Mei, Xubo Liu, Danilo P. Mandic, Wenwu Wang, and MarkD . Plumbley. Audioldm: Text-to-audio generation with latent diffusion models. In *International Conference on Machine Learning*, 2023.

[168] Lingjie Liu, Marc Habermann, Viktor Rudnev, Kripasindhu Sarkar, Jiatao Gu, and Christian Theobalt. Neural humans: Pose-controlled free-view synthesis of human actors with template-guided neural radiance fields. In *arxiv*, 2021.

[169] Shiguang Liu and Dinesh Manocha. Sound synthesis, propagation, and rendering: A survey. *arxiv*, 2020.

[170] Chen-Chou Lo, Szu-Wei Fu, Wen-Chin Huang, Xin Wang, Junichi Yamagishi, Yu Tsao, and Hsin-Min Wang. Mosnet: Deep learning based objective assessment for voice conversion. In *Proc. Interspeech 2019*, 2019.

[171] Barbara Locher, André Piquerez, Manuel Habermacher, Martina Ragettli, Martin Röösli, Mark Brink, Christian Cajochen, Danielle Vienneau, Maria Foraster, Uwe Müller, et al. Differences between outdoor and indoor sound levels for open, tilted, and closed windows. *International journal of environmental research and public health*, 2018.

[172] Cheng Lu, Yuhao Zhou, Fan Bao, Jianfei Chen, Chongxuan Li, and Jun Zhu. Dpm-solver: A fast ode solver for diffusion probabilistic model sampling in around 10 steps. *arXiv preprint arXiv:2206.00927*, 2022.

[173] Andrew Luo, Yilun Du, Michael J Tarr, Joshua B Tenenbaum, Antonio Torralba, and Chuang Gan. Learning neural acoustic fields. In *NeurIPS 2022*, 2022.

[174] Simian Luo, Chuanhao Yan, Chenxu Hu, and Hang Zhao. Diff-foley: Synchronized video-to-audio synthesis with latent diffusion models. In *NeurIPS*, 2023.

[175] Shuang Ma, Zhaoyang Zeng, Daniel McDuff, and Yale Song. Active contrastive learning of audio-visual video representations. In *ICLR*, 2021.

[176] Wolfgang Mack, Shuwen Deng, and Emanuël AP Habets. Single-channel blind direct-to-reverberation ratio estimation using masking. In *INTERSPEECH*, pages 5066–5070, 2020.

[177] Sagnik Majumder, Ziad Al-Halah, and Kristen Grauman. Move2hear: Active audio-visual source separation. In *ICCV*, 2021.

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

[179] Mishaim Malik, Muhammad Kamran Malik, Khawar Mehmood, and Imran Makhdoom. Automatic speech recognition: a survey. *Multimedia Tools and Applications 80, 6 (2021), 9411–9457.*, 2021.

[180] Xudong Mao, Qing Li, Haoran Xie, Raymond Y.K. Lau, Zhen Wang, and Stephen Paul Smolley. Least squares generative adversarial networks. *arXiv preprint arXiv:1611.04076*, 2016.

[181] Peter Vary Marco Jeub, Magnus Schafer. A binaural room impulse response database for the evaluation of dereverberation algorithms. In *Proceedings of International Conference on Digital Signal Processing*, 2009.

[182] Ricardo Martin-Brualla, Noha Radwan, Mehdi S. M. Sajjadi, Jonathan T. Barron, Alexey Dosovitskiy, and Daniel Duckworth. Nerf in the wild: Neural radiance fields for unconstrained photo collections. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7210–7219, June 2021.

[183] Daniela Massiceti, Stephen Lloyd Hicks, and Joram Jacob van Rheede. Stereosonic vision: Exploring visual-to-auditory sensory substitution mappings in an immersive virtual reality navigation paradigm. *PloS one*, 2018.

[184] Lotfi Merabet and Jaime Sanchez. Audio-based navigation using virtual environments: combining technology and neuroscience. *AER Journal: Research and Practice in Visual Impairment and Blindness*, 2009.

[185] Lotfi B Merabet and Alvaro Pascual-Leone. Neural reorganization following sensory loss: the opportunity of change. *Nature Reviews Neuroscience*, 2010.

[186] 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. In *ECCV*, 2020.

[187] 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. In *ICLR*, 2017.

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

[189] Himangi Mittal, Pedro Morgado, Unnat Jain, and Abhinav Gupta. Learning state-aware visual representations from audible interactions. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho, editors, *Advances in Neural Information Processing Systems*, 2022.

[190] Masato Miyoshi and Y. Kaneda. Inverse filtering of room acoustics. *IEEE Transactions on Acoustics, Speech, and Signal Processing*, 1988.

[191] Pedro Morgado, Nono Vasconcelos, Timothy Langlois, and Oliver Wang. Self-supervised generation of spatial audio for 360 video. In *NeurIPS*, 2018.

[192] Pedro Morgado, Ishan Misra, and Nuno Vasconcelos. Robust audio-visual instance discrimination. In *CVPR*, 2021.

[193] Arsalan Mousavian, Alexander Toshev, Marek Fiser, Jana Kosecka, and James Davidson. Visual representations for semantic target driven navigation. *arXiv preprint arXiv:1805.06066*, 2018.

[194] Prateek Murgai, Mark Rau, and Jean-Marc Jot. Blind estimation of the reverberation fingerprint of unknown acoustic environments. In *Audio Engineering Society Convention 143*. Audio Engineering Society, 2017.

[195] Damian T Murphy and Simon Shelley. Openair: An interactive auralization web resource and database. In *Audio Engineering Society Convention 129*, 2010.

[196] D.T. Murphy, Antti Kelloniemi, Jack Mullen, and Simon Shelley. Acoustic modeling using the digital waveguide mesh. *Signal Processing Magazine, IEEE*, 24:55 – 66, 04 2007.

[197] Christian Müller-Tomfelde. Time-varying filter in non-uniform block convolution. In *Proceedings of the COST G-6 Conference on Digital Audio Effects (DAFX-01)*, 2001.

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

[199] Arsha Nagrani, Shan Yang, Anurag Arnab, Cordelia Schmid, and Chen Sun. Attention bottlenecks for multimodal fusion. In *NeurIPS*, 2021.

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

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

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

[203] Kazuhiro Nakadai, Hiroshi G Okuno, and Hiroaki Kitano. Epipolar geometry based sound localization and extraction for humanoid audition. In *IROS Workshops*. IEEE, 2001.

[204] Satoshi Nakamura, Kazuo Hiyane, Futoshi Asano, Takanobu Nishiura, and Takeshi Yamada. Acoustical sound database in real environments for sound scene understanding and hands-free speech recognition. In *LREC*, 2000.

[205] Tomohiro Nakatani, Takuya Yoshioka, Keisuke Kinoshita, Masato Miyoshi, and Biing-Hwang Juang. Speech dereverberation based on variance-normalized delayed linear prediction. In *IEEE Transactions on Audio, Speech, and Language Processing*, 2010.

[206] Patrick A. Naylor and Nikolay D. Gaubitch. *Speech Dereverberation*. Springer Publishing Company, Incorporated, 1st edition, 2010.

[207] Stephen T. Neely and Jont B. Allen. Invertibility of a room impulse response. In *Journal of the Acoustical Society of America*, 1979.

[208] Michael Niemeyer, Jonathan T. Barron, Ben Mildenhall, Mehdi S. M. Sajjadi, Andreas Geiger, and Noha Radwan. RegNeRF: Regularizing neural radiance fields for view synthesis from sparse inputs. In *cvpr*, 2022.

[209] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio, 2016.

[210] EAI Organizers. Embodied AI workshop challenges. `https://embodied-ai.org`, 2022.

[211] Andrew Owens and Alexei A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.

[212] Andrew Owens, Phillip Isola, Josh McDermott, Antonio Torralba, Edward H Adelson, and William T Freeman. Visually indicated sounds. In *CVPR*, 2016.

[213] Andrew Owens, Jiajun Wu, Josh H McDermott, William T Freeman, and Antonio Torralba. Ambient sound provides supervision for visual learning. In *ECCV*, 2016.

[214] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur. Librispeech: an asr corpus based on public domain audio books. In *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 5206–5210. IEEE, 2015.

[215] Keunhong Park, Utkarsh Sinha, Jonathan T. Barron, Sofien Bouaziz, Dan B. Goldman, Steven M. Seitz, and Ricardo Martin-Brualla. Deformable neural radiance fields. *CoRR*, abs/2011.12948, 2020.

[216] Mandela Patrick, Yuki M Asano, Bernie Huang, Ishan Misra, Florian Metze, Joao Henriques, and Andrea Vedaldi. Space-time crop & attend: Improving cross-modal video representation learning. *arXiv preprint arXiv:2103.10211*, 2021.

[217] Sudipta Paul, Amit K. Roy-Chowdhury, and Anoop Cherian. Avlen: Audio-visual-language embodied navigation in 3d environments. In *NeurIPS*, 2022.

[218] Xue Bin Peng, Marcin Andrychowicz, Wojciech Zaremba, and Pieter Abbeel. Sim-to-real transfer of robotic control with dynamics randomization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 3803–3810, 2018.

[219] L. Picinali, A. Afonso, M. Denis, and B. Katz. Exploration of architectural spaces by blind people using auditory virtual reality for the construction of spatial knowledge. *International Journal of Human-Computer Studies*, 72(4): 393–407, 2014.

[220] Karol J. Piczak. ESC: Dataset for Environmental Sound Classification. In *Proceedings of the 23rd Annual ACM Conference on Multimedia*, pages 1015–1018. ACM Press, 2015. ISBN 978-1-4503-3459-4.

[221] Ben Poole, Sherjil Ozair, Aaron van den Oord amd Alexander A. Alemi, and George Tucker. On variational bounds of mutual information. In *ICML*, 2019.

[222] Ryan Prenger, Rafael Valle, and Bryan Catanzaro. Waveglow: A flow-based generative network for speech synthesis. In *arXiv*, 2018.

[223] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-NeRF: Neural radiance fields for dynamic scenes. *corr*, abs/2011.13961, 2020.

[224] Albert Pumarola, Enric Corona, Gerard Pons-Moll, and Francesc Moreno-Noguer. D-nerf: Neural radiance fields for dynamic scenes. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 10318–10327, June 2021.

[225] Senthil Purushwalkam, Sebastian Vicenc Amengual Gari, Vamsi Krishna Ithapu, Carl Schissler, Philip Robinson, Abhinav Gupta, and Kristen Grauman. Audio-visual floorplan reconstruction. *arXiv:2012.15470v1*, 2020.

[226] Jianzhao Qin, Jun Cheng, Xinyu Wu, and Yangsheng Xu. A learning based approach to audio surveillance in household environment. *International Journal of Information Acquisition*, 2006.

[227] Santhosh K. Ramakrishnan, Ziad Al-Halah, and Kristen Grauman. Occupancy anticipation for efficient exploration and navigation. In *ECCV*, 2020.

[228] Santhosh Kumar Ramakrishnan, Aaron Gokaslan, Erik Wijmans, Oleksandr Maksymets, Alexander Clegg, John M Turner, Eric Undersander, Wojciech Galuba, Andrew Westbury, Angel X Chang, Manolis Savva, Yili Zhao, and Dhruv Batra. Habitat-matterport 3d dataset (HM3d): 1000 large-scale 3d environments for embodied AI. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*, 2021.

[229] Santhosh Kumar Ramakrishnan, Devendra Singh Chaplot, Ziad Al-Halah, Jitendra Malik, and Kristen Grauman. Poni: Potential functions for objectgoal navigation with interaction-free learning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 18868–18878, 2022.

[230] Merey Ramazanova, Victor Escorcia, Fabian Caba Heilbron, Chen Zhao, and Bernard Ghanem. Owl (observe, watch, listen): Localizing actions in egocentric video via audiovisual temporal context. *arXiv*, 2022.

[231] Ram Ramrakhya, Eric Undersander, Dhruv Batra, and Abhishek Das. Habitat-web: Learning embodied object-search strategies from human demonstrations at scale. In *CVPR*, 2022.

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

[233] Mirco Ravanelli, Titouan Parcollet, Peter Plantinga, Aku Rouhe, Samuele Cornell, Loren Lugosch, Cem Subakan, Nauman Dawalatabad, Abdelwahab Heba, Jianyuan Zhong, Ju-Chieh Chou, Sung-Lin Yeh, Szu-Wei Fu, Chien-Feng Liao, Elena Rastorgueva, François Grondin, William Aris, Hwidong Na, Yan Gao, Renato De Mori, and Yoshua Bengio. SpeechBrain: A general-purpose speech toolkit, 2021. arXiv:2106.04624.

[234] Jeremy Reizenstein, Roman Shapovalov, Philipp Henzler, Luca Sbordone, Patrick Labatut, and David Novotny. Common Objects in 3D: Large-scale learning and evaluation of real-life 3D category reconstruction. In *cvpr*, 2021.

[235] Alexander Richard, Dejan Markovic, Israel D Gebru, Steven Krenn, Gladstone Butler, Fernando de la Torre, and Yaser Sheikh. Neural synthesis of binaural speech from mono audio. In *International Conference on Learning Representations*, 2021.

[236] Colleen Richey, Maria A. Barrios, Zeb Armstrong, Chris Bartels, Horacio Franco, Martin Graciarena, Aaron Lawson, Mahesh Kumar Nandwana, Allen Stauffer, Julien van Hout, Paul Gamble, Jeff Hetherly, Cory Stephenson, and Karl Ni. Voices obscured in complex environmental settings (voices) corpus, 2018.

[237] Antony W. Rix, John G. Beerends, Michael P. Hollier, and Andries P. Hekstra. Perceptual evaluation of speech quality (PESQ)-a new method for speech quality assessment of telephone networks and codecs. In *2001 IEEE International Conference on Acoustics, Speech, and Signal Processing. Proceedings*, 2001.

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

[239] Joseph M Romano, Jordan P Brindza, and Katherine J Kuchenbecker. Ros open-source audio recognizer: Roar environmental sound detection tools for robot programming. *Autonomous robots*, 2013.

[240] Olaf Ronneberger, Philipp Fischer, and Thomas Brox. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, pages 234–241. Springer, 2015.

[241] Joseph Roth, Sourish Chaudhuri, Ondrej Klejch, Radhika Marvin, Andrew Gallagher, Liat Kaver, Sharadh Ramaswamy, Arkadiusz Stopczynski, Cordelia Schmid, Zhonghua Xi, and Caroline Pantofaru. Ava-activespeaker: An audio-visual dataset for active speaker detection. In *ICASSP*, 2020.

[242] Victor Sanh, Lysandre Debut, Julien Chaumond, and Thomas Wolf. Distilbert, a distilled version of bert: smaller, faster, cheaper and lighter. *ArXiv*, abs/1910.01108, 2019.

[243] Andy Sarroff and Roth Michaels. Blind arbitrary reverb matching. In *Proceedings of the 23rd International Conference on Digital Audio Effects (DAFx-2020)*, 2020.

[244] Nikolay Savinov, Alexey Dosovitskiy, and Vladlen Koltun. Semi-parametric topological memory for navigation. In *ICLR*, 2018.

[245] 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.

[246] Lauri Savioja and Ning Xiang. Simulation-based auralization of room acoustics. *Acoust. Today*, 16(4):48–55, 2020.

[247] 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.

[248] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A python package for audio room simulations and array processing algorithms. *arXiv*, 2017.

[249] Carl Schissler and Dinesh Manocha. Adaptive impulse response modeling for interactive sound propagation. In *Proceedings of the 20th ACM SIGGRAPH Symposium on Interactive 3D Graphics and Games*, pages 71–78, 2016.

[250] Carl Schissler, Ravish Mehra, and Dinesh Manocha. High-order diffraction and diffuse reflections for interactive sound propagation in large environments. *ACM Transactions on Graphics (TOG) 33, 4 (2014), 39.*, 2014.

[251] Carl Schissler, Christian Loftin, and Dinesh Manocha. Acoustic classification and optimization for multi-modal rendering of real-world scenes. *IEEE transactions on visualization and computer graphics*, 24(3):1246–1259, 2017.

[252] Carl Schissler, Peter Stirling, and Ravish Mehra. Efficient construction of the spatial room impulse response. In *2017 IEEE Virtual Reality (VR)*, pages 122–130. IEEE, 2017.

[253] Carl Schissler, Gregor Mückl, and Paul Calamia. Fast diffraction pathfinding for dynamic sound propagation. *ACM Transactions on Graphics (TOG)*, 40 (4):1–13, 2021.

[254] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms, 2017.

[255] 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.

[256] James A Sethian. Fast marching methods. *SIAM review*, 41(2):199–235, 1999.

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

[258] Vincent Sitzmann, Michael Zollhöfer, and Gordon Wetzstein. Scene representation networks: Continuous 3D-structure-aware neural scene representations. *nips*, 2019.

[259] David Snyder, Daniel Garcia-Romero, Gregory Sell, Daniel Povey, and Sanjeev Khudanpur. X-vectors: Robust DNN embeddings for speaker recognition. In *2018 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 5329–5333. IEEE, 2018.

[260] Arjun Somayazulu1, Changan Chen, and Kristen Grauman. Self-supervised visual acoustic matching. In *NeurIPS*, 2023.

[261] Khurram Soomro, Amir Roshan Zamir, and Mubarak Shah. Ucf101: A dataset of 101 human actions classes from videos in the wild. *CoRR*, 2012.

[262] 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.

[263] Christian Steinmetz, Vamsi Krishna Ithapu, and Paul Calamia. Filtered noise shaping for time domain room impulse response estimation from reverberant speech. In *2021 IEEE Workshop on Applications of Signal Processing to Audio and Acoustics (WASPAA)*, 2021.

[264] Daniel Stoller, Sebastian Ewert, and Simon Dixon. Adversarial semi-supervised audio source separation applied to singing voice extraction. In *ICASSP*, 2018.

[265] 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.

[266] Jiaqi Su, Zeyu Jin, and Adam Finkelstein. Acoustic matching by embedding impulse responses. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 426–430. IEEE, 2020.

[267] Jiaqi Su, Zeyu Jin, and Adam Finkelstein. HiFi-GAN: High-fidelity denoising and dereverberation based on speech deep features in adversarial networks. In *INTERSPEECH*, 2020.

[268] Kun Su, Kaizhi Qian, Eli Shlizerman, Antonio Torralba, and Chuang Gan. Physics-driven diffusion models for impact sound synthesis from videos. *2023*

IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR), pages 9749–9759, 2023.

[269] Shih-Yang Su, Frank Yu, Michael Zollhöfer, and Helge Rhodin. A-NeRF: Surface-free human 3d pose refinement via neural rendering. *corr*, abs/2102.06199, 2021.

[270] Sainbayar Sukhbaatar, Arthur Szlam, Gabriel Synnaeve, Soumith Chintala, and Rob Fergus. Mazebase: A sandbox for learning from games. *arXiv preprint arXiv:1511.07401*, 2015.

[271] Igor Szöke, Miroslav Skácel, Ladislav Mošner, Jakub Paliesek, and Jan Černocký. Building and evaluation of a real room impulse response dataset. *IEEE Journal of Selected Topics in Signal Processing*, 2019.

[272] MIgor Szoke, Miroslav Skacel, Ladislav Mosner, Jakub Paliesek, and Jan "Honza" Cernocky. Building and evaluation of a real room impulse response dataset. *arXiv*, 2018.

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

[274] Ke Tan, Yong Xu, Shi-Xiong Zhang, Meng Yu, and Dong Yu. Audio-visual speech separation and dereverberation with a two-stage multimodal network. In *IEEE Journal of Selected Topics in Signal Processing*, 2020.

[275] Zhenyu Tang, Rohith Aralikatti, Anton Ratnarajah, , and Dinesh Manocha. Gwa: A large geometric-wave acoustic dataset for audio processing. In *Special*

Interest Group on Computer Graphics and Interactive Techniques Conference Proceedings (SIGGRAPH '22 Conference Proceedings), 2022. URL `https://doi.org/10.1145/3528233.3530731`.

[276] Ruijie Tao, Zexu Pan, Rohan Kumar Das, Xinyuan Qian, Mike Zheng Shou, and Haizhou Li. Is someone speaking? exploring long-term temporal features for audio-visual active speaker detection. In *Proceedings of the 29th ACM International Conference on Multimedia*, page 3927–3935, 2021.

[277] A. Tharwat. Classification assessment methods. *Applied Computing and Informatics*, 17(1):168–192, 2021. doi: https://doi.org/10.1016/j.aci.2018.08.003.

[278] Catherine Thinus-Blanc and Florence Gaunet. Representation of space in blind persons: vision as a spatial sense? *Psychological bulletin*, 1997.

[279] Sebastian Thrun, Wolfram Burgard, and Dieter Fox. *Probabilistic robotics*. MIT Press, 2005.

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

[281] Yonglong Tian, Dilip Krishnan, and Phillip Isola. Contrastive multiview coding. In *ECCV*, 2020.

[282] Josh Tobin, Rachel Fong, Alex Ray, Jonas Schneider, Wojciech Zaremba, and Pieter Abbeel. Domain randomization for transferring deep neural networks from simulation to the real world. In *IROS*, 2017.

[283] Edgar Tretschk, Ayush Tewari, Vladislav Golyanik, Michael Zollhöfer, Christoph Lassner, and Christian Theobalt. Non-rigid neural radiance fields: Reconstruction and novel view synthesis of a deforming scene from monocular video. *corr*, abs/2012.12247, 2020.

[284] Thanh-Dat Truong, Chi Nhan Duong, The De Vu, Hoang Anh Pham, Bhiksha Raj, Ngan Le, and Khoa Luu. The right to talk: An audio-visual transformer approach. In *ICCV*, 2021.

[285] Yao-Hung Hubert Tsai, Shaojie Bai, Paul Pu Liang, J. Zico Kolter, Louis-Philippe Morency, and Ruslan Salakhutdinov. Multimodal transformer for unaligned multimodal language sequences. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, Florence, Italy, 7 2019. Association for Computational Linguistics.

[286] Juliano Vacaro, Guilherme Marques, Bruna Oliveira, Gabriel Paz, Thomas Paula, Wagston Staehler, and David Murphy. Sim-to-real in reinforcement learning for everyone. *LARS-SBR-WRE*, 2019.

[287] Aaron van den Oord, Sander Dieleman, Heiga Zen, Karen Simonyan, Oriol Vinyals, Alex Graves, Nal Kalchbrenner, Andrew Senior, and Koray Kavukcuoglu. Wavenet: A generative model for raw audio. In *arXiv*, 2016.

[288] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. Representation learning with contrastive predictive coding, 2018.

[289] Laurens van der Maaten and Geoffrey Hinton. Visualizing high-dimensional data using t-sne. In *Journal of Machine Learning Research*, volume 9, pages 2579–2605, 2008.

[290] Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. In *NeurIPS*, 2017.

[291] Eric Veach and Leonidas Guibas. Bidirectional estimators for light transport. *Photorealistic Rendering Techniques*, 1995.

[292] Raquel Viciana-Abad, Rebeca Marfil, Jose Perez-Lorenzo, Juan Bandera, Adrian Romero-Garces, and Pedro Reche-Lopez. Audio-visual perception system for a humanoid robotic head. *Sensors*, 2014.

[293] Michael Vorländer. *Auralization*. Germany: Springer International Publishing, 2020.

[294] Patrice Voss, Maryse Lassonde, Frederic Gougoux, Madeleine Fortin, Jean-Paul Guillemot, and Franco Lepore. Early-and late-onset blind individuals show supra-normal auditory abilities in far-space. *Current Biology*, 2004.

[295] Vesa Välimäki, Julian Parker, Lauri Savioja, Julius O. Smith, and Jonathan Abel. More than 50 years of artificial reverberation. In *60th International Conference: DREAMS*, 2016.

[296] Sanna Wager, Keunwoo Choi, and Simon Durand. Dereverberation using joint estimation of dry speech signal and acoustic system. *arXiv preprint arXiv:2007.12581*, 2020.

[297] DeLiang Wang and Jitong Chen. Supervised speech separation based on deep learning: An overview. *arxiv*, 201.

[298] Yu Wang, Mubbasir Kapadia, Pengfei Huang, Ladislav Kavan, and Norman Badler. Sound localization and multi-modal steering for autonomous virtual agents. In *Symposium on Interactive 3D Graphics and Games*, 2014.

[299] Tomi Westerlund Wenshuai Zhao, Jorge Peña Queralta. Sim-to-real transfer in deep reinforcement learning for robotics: a survey. *arxiv*, 2020.

[300] Stephan Werner, Florian Klein, Annika Neidhardt, Ulrike Sloma, Christian Schneiderwind, and Karlheinz Brandenburg. Creation of auditory augmented reality using a position-dynamic binaural synthesis system—technical components, psychoacoustic needs, and perceptual evaluation. *Applied Sciences*, 11 (3), 2021. ISSN 2076-3417. doi: 10.3390/app11031150.

[301] Norbert Wiener. Extrapolation, interpolation, and smoothing of stationary time series. *Report of the Services 19, Research Project DIC-6037 MIT*, 1942.

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

[303] John Wood, Mark Magennis, Elena Francisca Cano Arias, Teresa Gutierrez, Helen Graupp, and Massimo Bergamasco. The design and evaluation of a computer game for the blind in the grab haptic audio virtual environment. *Proceedings of Eurohpatics*, 2003.

[304] Mitchell Wortsman, Kiana Ehsani, Mohammad Rastegari, Ali Farhadi, and Roozbeh Mottaghi. Learning to learn how to learn: Self-adaptive visual navigation using meta-learning. In *CVPR*, 2019.

[305] Abraham Woubie, Anssi Kanervisto, Janne Karttunen, and Ville Hautamaki. Do autonomous agents benefit from hearing? *arXiv preprint arXiv:1905.04192*, 2019.

[306] Bo Wu, Kehuang Li, Minglei Yang, and Chin-Hui Lee. A reverberation-time-aware approach to speech dereverberation based on deep neural networks. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2016.

[307] Bo Wu, Kehuang Li, Fengpei Ge, Zhen Huang, Minglei Yang, Sabato Marco Siniscalchi, and Chin-Hui Lee. An end-to-end deep learning approach to simultaneous speech dereverberation and acoustic modeling for robust speech recognition. In *IEEE Journal of Selected Topics in Signal Processing*, 2017.

[308] 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.

[309] Xinyu Wu, Haitao Gong, Pei Chen, Zhi Zhong, and Yangsheng Xu. Surveillance robot utilizing video and audio information. *Journal of Intelligent and Robotic Systems*, 2009.

[310] Yusong Wu*, Ke Chen*, Tianyu Zhang*, Yuchen Hui*, Taylor Berg-Kirkpatrick, and Shlomo Dubnov. Large-scale contrastive language-audio pretraining with feature fusion and keyword-to-caption augmentation. In *IEEE International Conference on Acoustics, Speech and Signal Processing, ICASSP*, 2023.

[311] B. Wymann, E. Espié, C. Guionneau, C. Dimitrakakis, R. Coulom, and A. Sumner. Torcs, the open racing car simulator, 2013.

[312] Fei Xia, Amir R. Zamir, Zhiyang He, Alexander Sax, Jitendra Malik, and Silvio Savarese. Gibson env: Real-world perception for embodied agents. In *2018 IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9068–9079, 2018.

[313] 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.

[314] Feifei Xiong, Stefan Goetze, Birger Kollmeier, and Bernd T Meyer. Joint estimation of reverberation time and early-to-late reverberation ratio from single-channel speech signals. *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 27(2):255–267, 2018.

[315] Ruilin Xu, Rundi Wu, Yuko Ishiwaka, Carl Vondrick, and Changxi Zheng. Listening to sounds of silence for speech denoising. In *NeurIPS*, 2019.

[316] Ryuichi Yamamoto, Eunwoo Song, and Jae-Min Kim. Parallel wavegan: A fast waveform generation model based on generative adversarial networks with multi-resolution spectrogram. In *ICASSP*, 2020.

[317] Takami Yoshida, Kazuhiro Nakadai, and Hiroshi G Okuno. Automatic speech recognition improved by two-layered audio-visual integration for robot audition. In *2009 9th IEEE-RAS International Conference on Humanoid Robots*, pages 604–609. IEEE, 2009.

[318] Abdelrahman Younes, Daniel Honerkamp, Tim Welschehold, and Abhinav Valada. Catch me if you hear me: Audio-visual navigation in complex unmapped environments with moving sounds. *IEEE Robotics and Automation Letters*, 2023.

[319] Alex Yu, Vickie Ye, Matthew Tancik, and Angjoo Kanazawa. PixelNeRF: Neural radiance fields from one or few images. In *cvpr*, 2021.

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

[321] Markus Zaunschirm, Christian Schörkhuber, and Robert Höldrich. Binaural rendering of ambisonic signals by head-related impulse response time alignment and a diffuseness constraint. *The Journal of the Acoustical Society of America*, 143(6):3616–3627, 2018.

[322] Hang Zhao, Chuang Gan, Andrew Rouditchenko, Carl Vondrick, Josh McDermott, and Antonio Torralba. The sound of pixels. In *ECCV*, 2018.

[323] Hang Zhao, Chuang Gan, Wei-Chiu Ma, and Antonio Torralba. The sound of motions. In *ICCV*, 2019.

[324] Wenshuai Zhao, Jorge Pena Queralta, and Tomi Westerlund. Sim-to-real transfer in deep reinforcement learning for robotics: A survey. In *2020 IEEE Symposium Series on Computational Intelligence (SSCI)*, pages 737–744, 2020.

[325] Yan Zhao, Zhong-Qiu Wang, and DeLiang Wang. Two-stage deep learning for noisy-reverberant speech enhancement. *IEEE/ACM Trans. on Audio, Speech, and Language Processing*, 2019.

[326] Yan Zhao, DeLiang Wang, Buye Xu, and Tao Zhang. Monaural speech dereverberation using temporal convolutional networks with self attention. In *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2020.

[327] Yuke Zhu, Daniel Gordon, Eric Kolve, Dieter Fox, Li Fei-Fei, Abhinav Gupta, Roozbeh Mottaghi, and Ali Farhadi. Visual Semantic Planning using Deep Successor Representations. In *ICCV*, 2017.

Vita

Changan Chen was born on October 27th, 1995, in Wenling, Zhejiang, China. He graduated from Wenling High School in 2014. Between 2014 and 2019, he was in a dual degree program between Zhejiang University (ZJU) in China and Simon Fraser University (SFU) in Canada. During the first two years of the program, he studied at ZJU, and he spent the final three years at SFU. He graduated with distinction, earning a Bachelor of Engineering in Computer Science from ZJU and a Bachelor of Science in Computing Science from SFU. In the fall of 2019, he began his graduate studies in the Department of Computer Science at the University of Texas at Austin, where he is advised by Professor Kristen Grauman. During his PhD, he was awarded the Adobe Research Fellowship in 2022 and the Spring Dissertation Fellowship in 2024. He also served as a visiting researcher at Facebook AI Research (FAIR) from 2020 to 2022.

Address: changanvr@gmail.com

This dissertation was typeset with LaTeX[†] by the author.

[†]LaTeX is a document preparation system developed by Leslie Lamport as a special version of Donald Knuth's TeX Program.