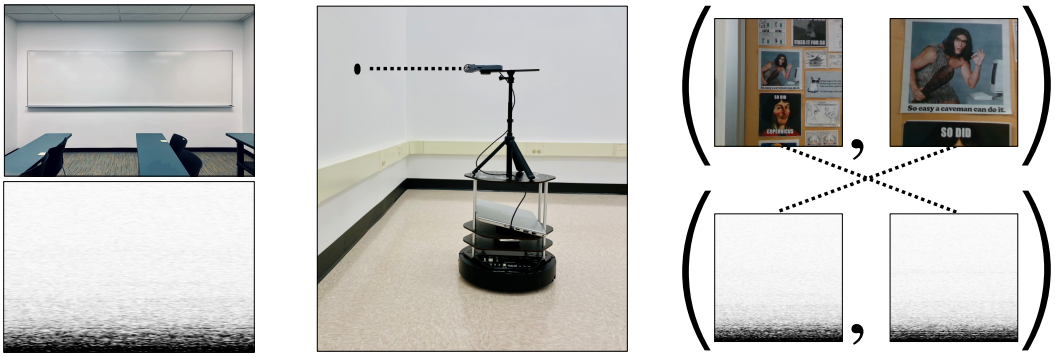
Structure from Silence: Learning Scene Structure from Ambient Sound

Anonymous Author(s)
Affiliation
Address
email



(a) *Quiet Campus* dataset (b) Depth estimation (c) Multimodal self-supervision

Figure 1: What does ambient sound tell us about scene structure? (a) We collect an "in-the-wild" dataset of paired audio and RGB-D recordings from quiet indoor scenes. (b) Given audio from a scene, we estimate distance to a wall. (c) We use this ambient audio to learn useful representations using multimodal self-supervision.

Abstract: The sounds that a robot hears within a scene subtly vary as it moves through it. In this paper, we ask whether these ambient sounds convey information about scene structure. To study this, we collect a dataset of "in-the-wild" paired audio and RGB-D recordings in a variety of quiet, indoor scenes, typical of what a robot would encounter when performing navigation tasks. Using this data, we train models that successfully estimate depth in a number of evaluation settings. Finally, we show that these sounds provide a useful learning signal, and that we can obtain a useful representation by associating a visual signal with sound. These results suggest that ambient sound conveys a surprising amount of information about scene structure, and that this information can be successfully exploited by learning systems.

Keywords: audio perception, multi-modal learning, self-supervision, navigation

1 Introduction

Humans make extensive use of acoustic signals to help understand the physical structure of a scene. Recent work has reproduced aspects of these capabilities in machines, typically by using distinctive sound landmarks, such as ringing phones [1]. However, since these methods have largely been studied in simulation, and it is unclear how often these types of sound are available to robots in practice. Real-world scenes are often very quiet, and the sounds that robots hear within them — e.g., ventilation noise, the hum of a machine — are ambiguous, and confounded with reverberation and other artifacts.

Human hearing, by contrast, is capable of estimating scene structure through a variety of subtle cues, often from only quiet ambient sound. For example, work in psychology has proposed that humans can—perhaps surprisingly—use spatial variations in the ambient sound spectrum to tell whether they are near a wall, and use "sound shadows" to detect obstructions [2]. Humans are also able to use reverberation to interpret scene geometry [3], and potentially exploit high-level regularities in scene composition, such as the typical locations of vents, doors, and windows.

Not only do these ambient sounds convey scene structure, but they can also provide a "free" learning signal for other modalities. Visual and audio signals often change in predictable ways, such as when

Submitted to the 5th Conference on Robot Learning (CoRL 2021). Do not distribute.

a buzzing refrigerator simultaneously gets louder and visually enlarges as one moves toward it. A model that can successfully associate vision and sound must understand the underlying 3D structure that explains them both.

In this paper, we ask whether ambient sounds that occur in *real-world* scenes convey 3D structure. To study this, we collect a large dataset of "in-the-wild" audio recordings from quiet, indoor scenes typical of what a robot would encounter when solving navigation tasks. Each sound in our dataset is paired with a corresponding recording from an RGB-D sensor, which provides a visual signal and pseudo ground-truth depth estimate. The resulting dataset, which we call the *Quiet Campus* dataset (Fig. 1a), covers a variety of room shapes, background sounds, and materials.

Using this dataset, we first conduct an experimental study for depth estimation from audio. We show that audio can be used to estimate distance to nearby walls in a variety of scenarios, including prediction of the relative depth between two recordings. We demonstrate that the model can be used as part of a very simple robotic navigation system, in which a wheeled robot moves along a wall using ambient audio cues (Fig. 1b).

Next, we ask whether this ambient audio can be used as a learning signal, without explicit use of depth. We study several multimodal self-supervised learning formulations, each of which requires a model to associate subtle changes in sound with a visual signal (Fig. 1c). We show that the resulting model learns a feature set that can be used to solve downstream distance-to-wall prediction tasks.

We make the following contributions: i) a dataset of paired audio and RGB-D data from real-world scenes, ii) an experimental study of in-the-wild audio-based depth estimation, iii) we show that ambient sound, when paired with image data, can provide useful self-supervision for depth estimation tasks. Our results suggest that ambient sound conveys a surprising amount of information about structure, and that it provides a useful training signal for multimodal learning.

2 Related Work

Human auditory perception. Psychoacoustics researchers have explored what humans can infer from sounds, such as size, material, shape, and depth. Perhaps the most familiar cue is *echolocation*, which is often used by people with visual impairments to estimate the distance with objects via self-produced sounds [4, 5]. However, humans also infer a great deal from *passive* observation of existing sounds. Our work is inspired by Ashmead *et al*.'s work in studying the navigation ability of blind children in ambient sound field [2, 6, 7]. They propose that blind people might use the accumulations of low-frequency ambient sounds to distinguish obstacles, and use physical modeling to study the difference between audio near and far from walls. Traer and McDermott [3] show that humans can use phase change of natural reverberation to distinguish different spaces. While these cues have been studied extensively in psychoacoustics, they are not widely used in robotic perception and self-supervised learning.

Audio-based depth estimation. Previous works have shown that audio contains rich spatial information. Gan *et al*. use stereo sound to perform cross-modal auditory localization for vehicles [8]. Yang *et al*. learn spatial correspondence of sight and binaural audio [9]. Purushwalkam *et al*. use an audio-visual signal to reconstruct floor plans. To do this, they use a simulated environment in which distinctive sound effects (e.g. toilet flush, alarm clock, dishwasher) from FreeSound [10] are synthetically inserted into the scenes via a physical simulation. In contrast, we experimentally study audio-based depth estimation in real-world scenes, using ambient sounds (e.g. ventilation noise). Other work measures the reflections of self-produced echolocation sounds to provide a consistent audio problem. BatVision [11] predicts depth map in real scenes from echo responses only while VisualEchoes [12] learns visual representations by echolocation in a simulation called SoundSpaces [13]. Instead of obtaining scene structure from active sensing, our method works entirely via passive observation of sounds, and does not require special hardware.

Audio-visual learning. Many recent works have proposed to use paired audio-visual data for representation learning. In seminal work, de Sa [14] proposed multimodal self-supervised learning as an alternative to single-modal models. Later, Ngiam *et al*. [15] learned a multimodal Boltzmann machine. Owens *et al*. [16] learned visual representations for materials from impact sounds, and used ambient sound to learn visual features [17]. In the latter, the ambient sounds are taken from internet video and thus encompass a much wider range of events, whereas we only consider very quiet indoor scenes. Later work learned multisensory representations [18, 19, 20, 21]. Other work has learned cross-modal distillation [22], sound localization [23, 24, 25, 19, 26, 27, 28, 29, 30], active speaker

Figure 2: **Quiet Campus Dataset**. Selected images from the *Static* and *Motion* subsets. Please check the supplementary to listen sounds.

detection [31], sound source separation [32, 33, 34, 35]. We take inspiration from this work and show that quiet, indoor sounds also provide a useful self-supervised learning signal. We also propose new audio-visual models that learn about relative depth in multiple modalities.

Visual depth prediction. Recent works have tackled recovering 3D structure from RGB images through learning-based approaches and researchers have developed increasingly detailed single image reconstructions [36, 37, 38, 39, 40, 41, 42], via voxel grids, point clouds, and meshes [43, 44, 45, 46]. At the same time, the significant progress has been made in predicting the relative camera pose [47, 48, 49, 50, 51]. We take inspiration from the techniques proposed in these models, though we predict depth from audio instead of vision.

Robot navigation. Robots often use visual streams to navigate to a destination in unseen environments [52, 53, 54, 55]. Recently, researchers leverage audio signals as useful cues for robot navigation tasks. Those robotic systems are designed to localize the traceable sound sources and navigate to audio goals in indoor environments [56, 57, 13, 58, 1, 59]. Unlike these methods, which use known and clear sound sources, we take advantage of ambient sounds collected in real-world scenes.

3 The *Quiet Campus* Dataset

To study the structure-from-audio problem, we collected a dataset containing a large number of "in-the-wild" indoor ambient audio recordings, paired with concurrently recorded RGB-D data. We collected our data from a large number of classrooms and hallways on a university campus. To ensure that the sounds within the recordings were subtle ambient sounds (e.g. ventilation noise), rather than distinctive audio events (e.g. speech), we collected the data at times when there were few people around. The recorded audio qualitatively sounds very quiet[1]. Please refer to our supplementary material for a random sample of sounds.

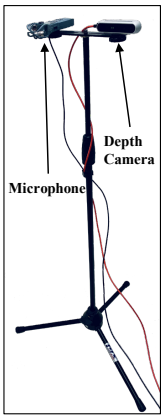


Figure 3: Capturing equipment.

We collected two types of data: *static* recordings in which the microphone and camera are stationary, and *motion* recordings where a human collector moves slowly toward or away from a wall. We show some visual examples from our collected dataset in Fig. 2. In our experiments, we split the data so that there is no physical overlap, such that the rooms in the training and tests sets are disjoint.

Static recordings. We collect data in 46 classrooms in 12 buildings, containing approximately 200 minutes audio. For each classroom, we select $16 - 30$ positions and record 10 secs. of audio at each one. Since we are interested in predicting the distance to the wall, we point the camera and microphone toward the nearest wall when recording, so that the distance to the wall is well-defined.

Static recordings with dense coverage. In order to evaluate generalization to different orientations, we use four classrooms from *Static* recordings whose size ranges from 20 to 35 cells total, and collect 10 hours of audio data densely with a wide range of angles (rather than always facing a wall). We call this these *Static-dense* recordings. These two datasets were recorded at different times of the year and with different usage patterns. The *Static* recordings were made in spring 2021, when buildings were at reduced occupancy due to the COVID-19 pandemic, and when air conditioners may have been in use, while *Static-dense* were recorded in the winter of 2020, when buildings were at normal occupancy and when heating may have been in use.

Motion recordings. To help understand whether small motion (and the subtle changes in audio over space) provides useful information about structure, we collect approximately 90 minutes of videos in motion (222 videos total). To ensure that there are no unintended obstructions nearby, we collect this data in hallways. During recording, the microphone and RGB-D camera move toward or away from a wall (Fig. 2). As in the static setting, we point the microphone toward the wall. To reduce possible sources of sound, the human operator moves slowly and does not take footsteps. We split

[1]We note that the audio is not *literally* silent: it might be described by listeners as being very quiet or background noise. Our title is a play on the "ambient" music *Structures from Silence* by Roach.

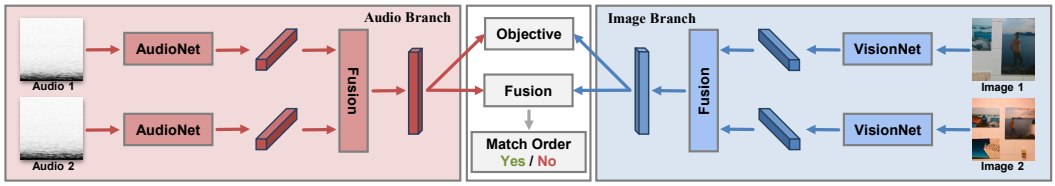


Figure 4: **Self-supervised audio-visual model**. We encode a pair of visual and pair of audio examples using CNNs, and fuse each pair independently using concatenation, resulting in 256-dimensional vectors. The two modalities are then combined and classified using a small multi-layer perceptron, which predicts whether the two pairs are in the same order. Please see the supplementary material for full details.

this dataset based on *building*, *i.e.*, the videos in the training and test set are hallways recorded in different buildings.

Estimating wall distance. For experiments that require distance to the wall, we obtain this by cropping the center 320×240 region from the depth image, and averaging the depth values. Since the camera is oriented toward the wall, this is generally a simple and accurate approximation.

Hardware. We use a ZOOM H1n Handy Recorder for collecting audio since it is a simple, inexpensive, and widely-used stereo audio recorder. For the RGB-D data, we use Intel Real-Sense Depth Camera D415 (Fig. 3).

4 Predicting Structure from Ambient Sound

We design models and experiments that will experimentally evaluate the ability of a model to learn to predict depth from audio in natural scenes.

4.1 Depth estimation tasks

Inspired by work in human obstacle detection with sound [5], we study the ability of a machine hearing system to predict the distance to a wall ahead of them. We pose this problem in multiple ways, which has applications in various robotic navigation scenarios.

Obstacle detection. First, we pose the problem of estimating whether the microphone is within a small distance of a wall (we use 0.5 meters).

Relative depth order. In many cases, a robotic system needs only to move toward or away from an obstacle, rather than to know its actual distance. We also hypothesize that relative depth estimation may be an easier task than absolute estimation, as it is in visual depth perception [60, 36]. Given two audio clips, we train a network to predict which one is closer to a wall.

Relative depth estimation. Given two audio clips, we train a network to predict the difference between their distance to the wall. We also consider predicting the log of their difference (*i.e.*, the ratio), following the work in visual depth estimation, which finds this easier to predict [38, 37], and as a multi-way classification task, after clustering depth differences in the training set with k-means.

Absolute depth estimation. In the end, we train a model to regress the distance to a wall, rather than a thresholded "near-or-far" prediction.

4.2 Self-supervised audio-visual learning

Having evaluated our model's ability to estimate depth directly from audio, we now ask whether we can learn about structure through audio-visual self-supervision. We hypothesize that, through the task of associating a raw (RGB) visual signal with audio, we will learn a representation with features that will be helpful for downstream depth estimation tasks.

One simple model, considered in prior work, is to solve a audio-visual synchronization task [31, 19, 27]: given the video, distinguish whether audio and visual stream are misaligned by randomly shifting audio. We call this model the **AV-Sync** model. Finally, we pose a new audio-visual learning problem that requires a model to learn *relative* depth between examples. We train a model to distinguish between *ordered* audio-visual pairs, which we call the **AV-Order** model. Given a pair of audio tracks and a pair of images (corresponding to the two audio tracks), we train a network to determine whether the two pairs are in the same (or different) order, which we frame as a classification problem.

4.3 Audio-based robotic navigation

To demonstrate the effectiveness of our depth estimation model, we use it to guide a wheeled robot (TurtleBot) to follow a wall. The robot uses the obstacle detection model to predict whether there is wall near the left/right of the robot, and moves accordingly. We use a simple policy, based on that of Gandhi *et al*. [61] (please see the supplement for full details). The robot moves forward in a sequence of steps. For each step, it rotates to an orientation that is derived from the obstacle detection model.

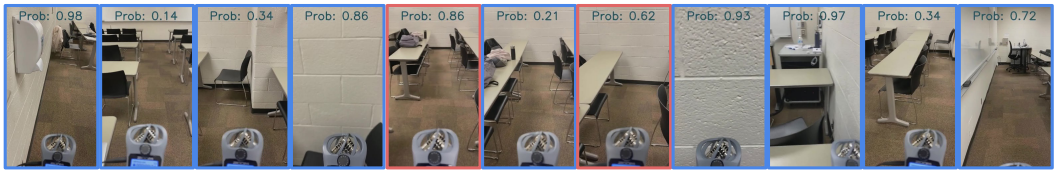


Figure 5: **Obstacle detection visualization**. After training our model, we walk through a room and show the probability of being near to a wall. The frames with blue border are correct predictions while frames with red border are failure cases. Please see the supplementary material for video results (with audio).

It senses the audio on its left side, then rotates $180°$ and senses audio to its right. We run the obstacle detection model on both audio recordings, and choose an orientation such that the robot moves away from the wall that has higher probability of being near a wall.

4.4 Models

Network backbone. We represent audio using the VGGish network [62]. For our experiments, we either replace the final layer with a new regression or classification head, or we use the (128-dimensional) penultimate layer. The network takes 0.96 sec of audio as input, in the form of a log-mel spectrogram. Due to the importance of low-frequency sound in physical models of wall-sound interaction [7], we do not remove the very lowest frequencies of the audio, in contrast to common practice in speech and audio event recognition [62]. We analyze the importance of this decision through experiments. We convert the stereo audio recordings to mono by averaging the two channels.

For models that use visual data, we use ResNet-18 [63], with 224×224 images, and modify the last linear layer to output the feature vectors with the same dimension as audio features. In addition, we also finetune the weights from ImageNet [64] to see the upper bound performances.

Learning details. For fully supervised tasks, we use SGD [65] optimizer with a learning rate 10^{-3}, momentum 0.9, and weight decay 5×10^{-4}. For the self-supervised learning experiments, we use Adam [66] with a learning rate of 10^{-4} for representation learning. For linear probing experiments, we use SGD with a learning rate of 10^{-2}. We schedule learning rates via cosine decay and pick the best model from validation for all the training jobs.

Audio processing. Our audio preprocessing resembles that of [62]. We resample the audio to 16 kHz and extract segments of 0.96 secs. We then convert them to spectrograms by applying a Short-time Fourier transform at a stride of 10 ms and 25 ms, and convert them into 96×64 log-mel spectrograms. For augmentation, we randomly crop the waveform in time and multiply the amplitudes by a random factor uniformly sampled from [0.5, 1.5].

Depth estimation networks. For the regression and binary classification tasks, we add a small multi-layer perceptron on top of the VGGish network. For the relative estimation tasks, our network aims to identify objectives with two audio inputs. We approach this by building a Siamese network that fuses the extracted feature vectors via concatenation prior to the multi-layer perceptron. For regression, we normalize depths to be in the range $[-0.5, 0.5]$ during training.

Network for self-supervision tasks. For AV-Sync task, we concatenate image and audio features before passing through the multi-layer perceptron. For the AV-Order model, we fuse Siamese networks from both modalities via concatenation and add a multi-layer perceptron (Fig. 4). In the downstream task, we take the output features from the last convolution layer after global average pooling and freeze them. We pass our 512-d learned features to a linear layer to get the objectives and only optimize it during training.

5 Experiments

Having collected a dataset of in-the-wild audio and models for inferring structure from it, we conduct an experimental study of what can be inferred from ambient sound in real-world scenes. We report our results along with 95% confidence intervals.

5.1 An experimental study of audio-based depth estimation

Obstacle detection. First, we evaluate our model's ability to solve the obstacle detection problem (Tab. 1). We split the data such that the test rooms were not heard during training. We report accuracy on this binary classification task, as well as average precision (to account for possible dataset shift from using held-out rooms). We evaluated our model in both the static and motion settings. To help interpret the effectiveness of audio in the context of other modalities, we also included simple visual models, with both random initialization and ImageNet pretraining [2]. Interestingly, we see that the

[2] We do not use a pretrained VGGish audio model since existing models remove low frequencies and are not compatible without remarkable changes.

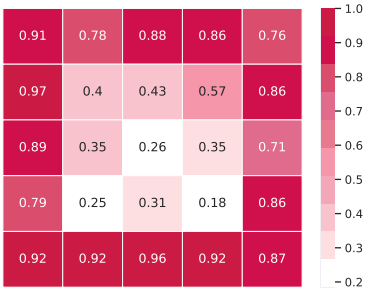


Figure 6: **Wall detection in a single room.** We divide a room into a grid, and record audio within each cell. We show the average obstacle detection probability in each one.

Table 1: **Obstacle detection and relative depth order.** We evaluate our model's ability to determine whether a microphone is within 0.5 meters of a wall and identify which sound has a smaller distance to the wall. *Pre* refers to pretraining.

			Obstacle detection		Relative order	
Model	Pre.	Task	AP(%)	Acc(%)	AP(%)	Acc(%)
Audio		Static	68.3 (±1.3)	60.0 (±0.9)	85.5 (±1.0)	77.2 (±0.8)
Image		Static	99.2 (±0.2)	95.5 (±0.5)	94.6 (±0.5)	86.4 (±0.7)
Image	✓	Static	99.5 (±0.1)	98.4 (±0.4)	97.7 (±0.2)	92.1 (±0.5)
Chance		Static	46.4 (±1.4)	50.0 (±1.0)	47.2 (±1.3)	50.0 (±1.0)
Audio		Motion	65.6 (±1.4)	64.5 (±0.9)	87.1 (±1.0)	81.3 (±0.8)
Image		Motion	73.4 (±1.2)	68.2 (±1.0)	87.9 (±0.9)	81.2 (±0.8)
Image	✓	Motion	88.6 (±0.7)	78.5 (±0.8)	97.1 (±0.3)	90.6 (±0.6)
Chance		Motion	50.4 (±1.4)	50.0 (±1.0)	50.5 (±1.4)	50.0 (±1.0)

audio-based model performs better than chance in both cases, suggesting that ambient sound, indeed, conveys information about scene structure.

We also evaluated our model's ability to generalize across orientations and recording times. We found that a model trained on *Static* recordings could achieve 86.1% AP and 72.9% Acc on *Static-dense* recordings, indicating strong generalization performance.

We qualitatively demonstrate the obstacle detection results by training a version of the model on the *Static-dense* data, since it contains a large number of viewpoint variations and angles (see supplement for quantitative results). We apply the learned obstacle detection model to every frame of a video in which a person walks through a room (Fig. 5). In Fig. 6, we also apply this model densely within a room divided according to its spatial grid. We averaged the probability of the model for each grid cell. We see that it successfully distinguishes between the boundaries of the room and the center.

Relative depth order. In Tab. 1, we evaluate our model's ability to predict which of two audio clips is closer to a wall. Once again, we find that the model obtains better results than chance (see supplementary for more results). We also see that, as is the case with visual prediction tasks, the relative prediction task is easier than predicting obstacles.

Table 2: **Relative depth ratio.** We evaluate our model's ability of predicting relative depth ratio from two ambient sounds, for the *Motion* recordings.

	Regression			Regression-by-Classification		
Model	MAE ↓	Med. ↓	R^2 ↑	Top-1 ↑	Top-5 ↑	Avg. Dist ↓
Audio	0.55 (±.01)	0.44 (±.01)	0.48 (±.02)	22.8 (±0.8)	80.7 (±0.7)	1.66 (±.03)
Image	0.54 (±.01)	0.42 (±.01)	0.49 (±.02)	26.6 (±0.8)	83.6 (±0.7)	1.47 (±.02)
Image (Pre.)	0.39 (±.01)	0.29 (±.01)	0.72 (±.01)	34.2 (±0.9)	90.5 (±0.6)	1.15 (±.02)
Chance	0.89 (±.01)	0.79 (±.01)	0.00 (±.00)	9.45 (±0.6)	52.6 (±1.0)	2.79 (±.04)
No input	0.82 (±.01)	0.75 (±.01)	0.00 (±.00)	10.7 (±0.6)	51.6 (±1.0)	4.50 (±.05)

Relative depth ratio. Since the network is able to tell relative depth order well, we explore relative depth setup further by predicting depth difference of audio pairs (log depth ratio). We use regression to predict the log-ratio from ambient sounds with L1 Loss function and via classification. We report the performance of mean absolute error, median error, and R^2. For classification, we report top-1 and top-5 accuracy, and report average distance from the ground truth bin using the distance between centers of paired bins. We report the performance of zeroing out two inputs as a baseline. As shown in Tab. 2, we can see that our method outperforms chance and, interestingly, has a relatively small gap between the vision-based approach.

Table 3: **Absolute depth estimation.** We evaluate our model's ability of predicting absolute distance to the wall for the *Motion* recordings.

		Regression			Regression-by-Classification		
	Model	MAE ↓	Med. ↓	R^2 ↑	Top-1 ↑	Top-5 ↑	Avg. Dist ↓
Single	Audio	0.28 (±.00)	0.25 (±.01)	-0.34 (±.03)	30.8 (±0.9)	88.3 (±0.6)	1.11 (±.02)
	Image	0.31 (±.00)	0.27 (±.01)	-0.67 (±.07)	35.6 (±0.9)	95.9 (±0.4)	1.05 (±.02)
	Image (Pre.)	0.26 (±.00)	0.21 (±.01)	-0.24 (±.04)	50.8 (±1.0)	99.2 (±0.2)	0.62 (±.01)
	No input	0.28 (±.00)	0.27 (±.01)	-0.19 (±.02)	24.3 (±0.8)	88.3 (±0.6)	1.07 (±.01)
Conditional	Audio	0.21 (±.00)	0.17 (±.00)	0.19 (±.02)	36.9 (±1.0)	90.0 (±.06)	1.17 (±.02)
	Image	0.22 (±.00)	0.18 (±.00)	0.12 (±.02)	38.2 (±0.9)	95.5 (±0.4)	0.93 (±.02)
	Image (Pre.)	0.18 (±.00)	0.14 (±.00)	0.39 (±.02)	51.7 (±1.0)	99.8 (±0.1)	0.59 (±.01)
	No input	0.25 (±.00)	0.23 (±.00)	0.01 (±.01)	26.4 (±0.8)	95.9 (±0.4)	1.43 (±.03)
	Chance	0.78 (±.01)	0.84 (±.01)	-3.38 (±0.23)	23.3 (±1.2)	56.9 (±0.9)	2.83 (±.02)

Absolute depth estimation. We evaluated our model's ability to predict depth from a **single** audio example, using similar experiments as the relative depth ratio (Tab. 3). Our model is more accurate than chance (which we obtain for all metrics by simply ablating the input and training). Likewise, the regression-by-classification model improves over chance performance.

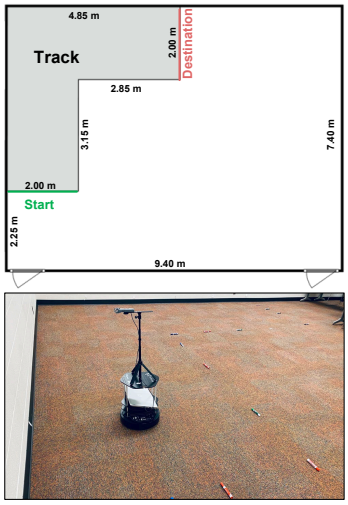


Figure 7: Classroom floor plan and track setting.

Table 4: **Linear probing experiments.** We evaluate our self-supervised feature set for **obstacle detection** and **relative depth order**, for the *Motion* recordings. Here, *Audio* means taking audio only as inputs. *Visual* means taking images only as inputs. *Both* means taking both audio and image as inputs.

			Obstacle detection		Relative order	
	Model	Pre.	AP(%)	Acc(%)	AP(%)	Acc(%)
Audio	Scratch		61.9 (±1.5)	60.3 (±0.9)	78.0 (±1.4)	73.1 (±0.9)
	VGGish [62]		58.2 (±1.3)	56.0 (±1.0)	61.1 (±1.4)	61.2 (±1.0)
	AV-Sync		**69.1** (±1.4)	**64.0** (±0.9)	80.2 (±1.3)	74.1 (±0.8)
	AV-Order		63.4 (±1.4)	61.5 (±0.9)	**84.2** (±1.2)	**79.4** (±0.7)
	VGGish [62]	✓	59.0 (±1.5)	56.7 (±1.0)	67.7 (±1.4)	64.5 (±0.9)
	AV-Sync	✓	**65.3** (±1.4)	**62.8** (±0.9)	82.1 (±1.2)	76.4 (±0.8)
	AV-Order	✓	62.8 (±1.5)	64.5 (±0.9)	**85.5** (±1.1)	**80.7** (±0.8)
Visual	Scratch		70.1 (±1.3)	64.0 (±0.9)	79.7 (±1.1)	71.5 (±0.9)
	AV-Sync		**77.1** (±1.1)	**69.2** (±0.9)	85.3 (±0.9)	76.1 (±0.8)
	AV-Order		76.8 (±1.1)	68.8 (±0.9)	**87.4** (±0.9)	**79.1** (±0.8)
	ImageNet [64, 63]	✓	80.4 (±1.2)	74.5 (±0.8)	94.0 (±0.5)	85.8 (±0.7)
	AV-Sync	✓	**89.0** (±0.8)	75.6 (±0.8)	92.8 (±0.6)	85.4 (±0.7)
	AV-Order	✓	86.5 (±1.1)	**76.3** (±0.8)	**95.8** (±0.4)	**88.9** (±0.6)
Both	AV-Order		77.1 (±1.1)	69.1 (±0.9)	89.0 (±0.8)	80.8 (±0.8)
	AV-Order	✓	88.1 (±0.9)	76.9 (±0.8)	95.8 (±0.4)	88.9 (±0.6)

We also consider another CNN model inspired by the physical model of Ashmead et al. [6] and our relative depth model. This new model estimates depth for a given audio input after conditioning on a reference sound recorded within the same scene and its depth (see the supplementary material for details). We call the **conditional** absolute depth model. This model is better than our model with only a single audio input, suggesting that the information from reference audio is a useful cue.

5.2 Audio-visual representation learning

Finally, we investigate whether we learn useful representation with **AV-Order** and **AV-Sync** models. In the AV-Order binary classification pretext task, our model obtains 85.8% AP and 75.8% accuracy (80.3% AP and 70.7% accuracy with random initialization) on the test set where the random chance is 50%. Given its performance on the pretext task, we test whether the learned audio and visual representations about depth are useful for the downstream tasks: (1) obstacle detection and (2) relative depth estimation. To evaluate the quality of our spatial representation, we freeze the learned features and train a linear classifier on the depth estimation tasks (Section 4.1). In the downstream tasks, we validate image and audio features individually and compare them with random features, pre-trained ImageNet features [64, 63], pre-trained AudioSet features [62] and **AV-Sync** features. Since AudioSet pretrained weights require a special audio pre-processing that differs from ours, we provide another random feature baseline based on the same processing as [62].

Table 5: **Linear probing experiments.** We evaluate our learned representation for **relative depth ratio** for the *Motion* recordings.

	Model	Pre.	Top-1 (%) ↑	Top-5 (%) ↑	Avg. Dist ↓
Audio	Scratch		19.2 (±0.8)	72.8 (±0.8)	2.33 (±0.04)
	VGGish [62]		14.4 (±0.7)	53.9 (±1.0)	3.78 (±0.06)
	AV-Sync.		19.2 (±0.7)	72.7 (±0.8)	2.07 (±0.03)
	AV-Order		**22.2** (±0.9)	**79.6** (±0.8)	**1.86** (±0.03)
	VGGish [62]	✓	15.6 (±0.7)	54.0 (±1.0)	3.59 (±0.05)
	AV-Sync.	✓	20.7 (±0.8)	75.1 (±0.9)	1.99 (±0.03)
	AV-Order	✓	**23.6** (±0.9)	**80.5** (±0.7)	**1.75** (±0.03)
Visual	Scratch		18.5 (±0.8)	70.8 (±0.8)	2.66 (±0.05)
	AV-Sync.		22.2 (±0.8)	76.8 (±0.8)	1.85 (±0.03)
	AV-Order		**24.7** (±0.8)	**80.2** (±0.8)	**1.71** (±0.03)
	ImageNet [64, 63]	✓	27.4 (±0.9)	87.1 (±0.7)	1.60 (±0.03)
	AV-Sync.	✓	27.5 (±0.8)	85.2 (±0.7)	1.53 (±0.03)
	AV-Order	✓	**28.9** (±0.9)	**88.6** (±0.6)	**1.40** (±0.03)
Both	AV-Order		23.8 (±0.8)	81.5 (±0.7)	1.59 (±0.03)
	AV-Order	✓	30.0 (±0.9)	89.3 (±0.6)	1.31 (±0.03)

Relative depth. We report results for relative depth order and relative depth ratio in Tab. 4 and Tab. 5 respectively. Our method outperforms baselines in both audio and visual modalities suggesting that we learn useful depth representations. Moreover, our results are often close to supervised features.

Obstacle detection. As the results shown in Tab. 4, AV-Order model outperforms Scratch and AudioSet pretraining, while it loses to AV-Sync model. It is reasonable that AV-Order model get worse performance since AV-Order model deals with two inputs during the proxy task. We provide an additional experiment by fine-tuning our learned AV-Order feature and we compare with the fully supervised method in Tab. 1. This obtains a 4% AP boost (70.0% v.s. 65.6%), suggesting that our learned representation is helpful.

Furthermore, the models with both audio and visual features as inputs perform consistently better than visual features solely, suggesting the audio is useful in addition to the visual signal.

5.3 Audio cues

We ask what audio information that the network uses to solve its task. We take inspiration from physical models of audio spectra near walls [6], which suggest that the distribution of low-frequency sounds (20-200Hz) are an important cue. We zero out different frequency ranges in the mel spectrogram, then re-train the relative depth order model. As can be seen in Fig. 8, the accuracy of the model with only very low frequencies (0-250Hz) is comparable to the model with the full frequency range, suggesting that these frequencies are indeed very important. For comparison, we conducted the same experiment with instrument categories from the VGGSound dataset [67] (see supp. for details). Interestingly, there these low frequencies are less important, suggesting that our setting relies on different cues.

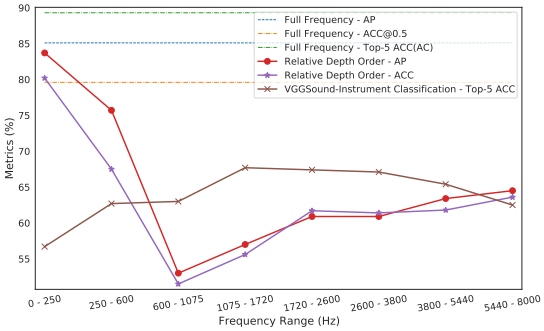


Figure 8: Performance vs. different frequency ranges.

5.4 Robotic navigation with ambient sound

We ask whether our depth estimation model can be used to guide a robot to follow a wall, solely using ambient sound. For this, we used a model trained on *Static* recordings, and evaluate on a classroom that was not included in the training set. The track is 8m long, and forces the robot to turn a corner (Fig. 7). The robot starts navigating from one of several unknown locations (40cm, 80cm, or 120cm from the wall) and orientations (30° left or right, or facing forward). We repeat the experiment 18 times and measure the average distance along the track it attained before colliding with a boundary. Quantitative results are shown in Fig. 9. We compare with a random policy, a "straight line" policy that always drives forward, and a policy based on the vision model. For the random policy, the setting is the same as our method, except we use a randomly initialized network for obstacle prediction. The straight line policy is effective when the robot starts with an orientation facing front, and it has the advantage that it does not rotate to sense audio, and hence avoids accumulating errors[3]. It's not surprising that the vision model outperforms our audio model, since we have known that vision performs well for those kind of tasks. We see that our method performs better than random and straight line policy, and successfully guides the robot pass the corner in nearly half of the trials.

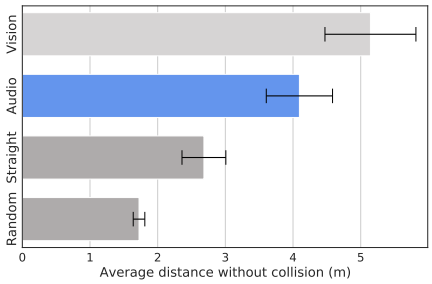


Figure 9: Robot navigation results.

6 Conclusion

In this paper, we have shown that real-world, indoor ambient sound provides a great deal of information about structure. We contributed the *Quiet Campus* dataset, a large dataset of in-the-wild paired audio and RGB-D recordings. We showed that distance to a wall can be predicted from ambient audio in a number of settings. We also showed that these ambient sounds can be used for multimodal self-supervised learning, including with a novel formulation that captures relative depth.

We see this work as opening two directions. The first is in creating robotic systems that exploit real-world, passively obtained audio signals. While related systems have previously been demonstrated in simulation with distinctive audio events (or with active echolocation), we take a step toward interpreting *in-the-wild* ambient sound. Second, we see our work as a step toward using the "free" audio signals to help train other modalities, such as vision, about scene structure.

Limitations. While ambient sound provides information about scene structure, it is a weaker signal when compared to vision and potentially self-produced sound, and we do not yet have a full understanding of what can (and cannot) be perceived from it, e.g. whether there are fundamental limitations on resolution, material, or angle. Additionally, while we will release our pretrained model for others to use, our experiments were conducted on a single college campus, and hence may not be representative of all indoor spaces.

[3]These errors could be reduced by using a multi-microphone array, though for simplicity we use a single microphone.

References

[1] C. Chen, S. Majumder, Z. Al-Halah, R. Gao, S. K. Ramakrishnan, and K. Grauman. Learning to set waypoints for audio-visual navigation. *arXiv preprint arXiv:2008.09622*, 1(2):6, 2020.

[2] D. H. Ashmead, R. S. Wall, S. B. Eaton, K. A. Ebinger, M.-M. Snook-Hill, D. A. Guth, and X. Yang. Echolocation reconsidered: Using spatial variations in the ambient sound field to guide locomotion. *Journal of Visual Impairment & Blindness*, 92(9):615–632, 1998.

[3] J. Traer and J. H. McDermott. Statistics of natural reverberation enable perceptual separation of sound and space. *Proceedings of the National Academy of Sciences*, 113(48):E7856–E7865, 2016.

[4] M. R. Rosenzweig, D. A. Riley, and D. Krech. Evidence for echolocation in the rat. *Science*, 1955.

[5] D. H. Ashmead, E. W. Hill, and C. R. Talor. Obstacle perception by ongenitally blind children. *Perception & psychophysics*, 46(5):425–433, 1989.

[6] D. H. Ashmead and R. S. Wall. Auditory perception of walls via spectral variations in the ambient sound field. *Journal of rehabilitation research and development*, 36(4):313—322, October 1999. ISSN 0748-7711.

[7] D. H. Ashmead and R. S. Wall. Low frequency sound as a navigational tool for people with visual impairments. *Journal of Low Frequency Noise, Vibration and Active Control*, 21(4): 199–205, 2002. doi:10.1260/026309202321834645.

[8] C. Gan, H. Zhao, P. Chen, D. Cox, and A. Torralba. Self-supervised moving vehicle tracking with stereo sound. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 7053–7062, 2019.

[9] K. Yang, B. Russell, and J. Salamon. Telling left from right: Learning spatial correspondence of sight and sound. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 9932–9941, 2020.

[10] F. Font, G. Roma, and X. Serra. Freesound technical demo. In *ACM International Conference on Multimedia (MM'13)*, pages 411–412, Barcelona, Spain, 21/10/2013 2013. ACM, ACM. ISBN 978-1-4503-2404-5. doi:10.1145/2502081.2502245.

[11] J. H. Christensen, S. Hornauer, and X. Y. Stella. Batvision: Learning to see 3d spatial layout with two ears. In *2020 IEEE International Conference on Robotics and Automation (ICRA)*, pages 1581–1587. IEEE, 2020.

[12] R. Gao, C. Chen, Z. Al-Halah, C. Schissler, and K. Grauman. Visualechoes: Spatial image representation learning through echolocation. In *European Conference on Computer Vision*, pages 658–676. Springer, 2020.

[13] C. Chen, U. Jain, C. Schissler, S. V. A. Gari, Z. Al-Halah, V. K. Ithapu, P. Robinson, and K. Grauman. Soundspaces: Audio-visual navigation in 3d environments. In *Proceedings of the European Conference on Computer Vision (ECCV)*. Springer, 2020.

[14] V. R. de Sa. Learning classification with unlabeled data. In *Advances in neural information processing systems*, pages 112–119. Citeseer, 1994.

[15] J. Ngiam, A. Khosla, M. Kim, J. Nam, H. Lee, and A. Y. Ng. Multimodal deep learning. In *ICML*, 2011.

[16] A. Owens, P. Isola, J. McDermott, A. Torralba, E. H. Adelson, and W. T. Freeman. Visually indicated sounds. In *CVPR*, 2016.

[17] A. Owens, J. Wu, J. H. McDermott, W. T. Freeman, and A. Torralba. Ambient sound provides supervision for visual learning. In *ECCV*. Springer, 2016.

[18] R. Arandjelovic and A. Zisserman. Look, listen and learn. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 609–617, 2017.

[19] A. Owens and A. A. Efros. Audio-visual scene analysis with self-supervised multisensory features. In *ECCV*, 2018.

[20] B. Korbar, D. Tran, and L. Torresani. Cooperative learning of audio and video models from self-supervised synchronization. *arXiv preprint arXiv:1807.00230*, 2018.

[21] F. Xiao, Y. J. Lee, K. Grauman, J. Malik, and C. Feichtenhofer. Audiovisual slowfast networks for video recognition. *arXiv preprint arXiv:2001.08740*, 2020.

[22] Y. Aytar, C. Vondrick, and A. Torralba. Soundnet: Learning sound representations from unlabeled video. In *Advances in neural information processing systems*, pages 892–900, 2016.

[23] A. Senocak, T.-H. Oh, J. Kim, M.-H. Yang, and I. So Kweon. Learning to localize sound source in visual scenes. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 4358–4366, 2018.

[24] R. Arandjelovic and A. Zisserman. Objects that sound. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 435–451, 2018.

[25] D. Harwath, A. Recasens, D. Surís, G. Chuang, A. Torralba, and J. Glass. Jointly discovering visual objects and spoken words from raw sensory input. In *Proceedings of the European conference on computer vision (ECCV)*, pages 649–665, 2018.

[26] Y. Tian, J. Shi, B. Li, Z. Duan, and C. Xu. Audio-visual event localization in unconstrained videos. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pages 247–263, 2018.

[27] T. Afouras, A. Owens, J. S. Chung, and A. Zisserman. Self-supervised learning of audio-visual objects from video. *arXiv preprint arXiv:2008.04237*, 2020.

[28] R. Qian, D. Hu, H. Dinkel, M. Wu, N. Xu, and W. Lin. Multiple sound sources localization from coarse to fine. *arXiv preprint arXiv:2007.06355*, 2020.

[29] D. Hu, R. Qian, M. Jiang, X. Tan, S. Wen, E. Ding, W. Lin, and D. Dou. Discriminative sounding objects localization via self-supervised audiovisual matching. *Advances in Neural Information Processing Systems*, 33, 2020.

[30] H. Chen, W. Xie, T. Afouras, A. Nagrani, A. Vedaldi, and A. Zisserman. Localizing visual sounds the hard way. *arXiv preprint arXiv:2104.02691*, 2021.

[31] J. S. Chung and A. Zisserman. Out of time: automated lip sync in the wild. In *Asian conference on computer vision*, pages 251–263. Springer, 2016.

[32] R. Gao, R. Feris, and K. Grauman. Learning to separate object sounds by watching unlabeled video. In *ECCV*, 2018.

[33] H. Zhao, C. Gan, A. Rouditchenko, C. Vondrick, J. McDermott, and A. Torralba. The sound of pixels. In *ECCV*, 2018.

[34] H. Zhao, C. Gan, W.-C. Ma, and A. Torralba. The sound of motions. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 1735–1744, 2019.

[35] R. Gao and K. Grauman. Visualvoice: Audio-visual speech separation with cross-modal consistency. *arXiv preprint arXiv:2101.03149*, 2021.

[36] D. Hoiem, A. A. Efros, and M. Hebert. Recovering surface layout from an image. *International Journal of Computer Vision*, 75(1):151–172, 2007.

[37] A. Saxena, M. Sun, and A. Y. Ng. Make3d: Learning 3d scene structure from a single still image. *IEEE transactions on pattern analysis and machine intelligence*, 31(5):824–840, 2008.

[38] D. Eigen and R. Fergus. Predicting depth, surface normals and semantic labels with a common multi-scale convolutional architecture. In *ICCV*, 2015.

[39] W. Chen, Z. Fu, D. Yang, and J. Deng. Single-image depth perception in the wild. In *Advances in neural information processing systems*, pages 730–738, 2016.

[40] W. Chen, S. Qian, and J. Deng. Learning single-image depth from videos using quality assessment networks. In *CVPR*, 2019.

[41] Z. Li and N. Snavely. Megadepth: Learning single-view depth prediction from internet photos. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, pages 2041–2050, 2018.

[42] C. Godard, O. Mac Aodha, M. Firman, and G. J. Brostow. Digging into self-supervised monocular depth estimation. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 3828–3838, 2019.

[43] S. Tulsiani, S. Gupta, D. F. Fouhey, A. A. Efros, and J. Malik. Factoring shape, pose, and layout from the 2d image of a 3d scene. In *CVPR*, 2018.

[44] G. Gkioxari, J. Malik, and J. Johnson. Mesh r-cnn. In *ICCV*, 2019.

[45] R. Girdhar, D. F. Fouhey, M. Rodriguez, and A. Gupta. Learning a predictable and generative vector representation for objects. In *ECCV*. Springer, 2016.

[46] W. Yin, J. Zhang, O. Wang, S. Niklaus, L. Mai, S. Chen, and C. Shen. Learning to recover 3d scene shape from a single image. *arXiv preprint arXiv:2012.09365*, 2020.

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

[48] R. I. Hartley. In defense of the eight-point algorithm. *IEEE Transactions on pattern analysis and machine intelligence*, 19(6):580–593, 1997.

[49] C. Wu, B. Clipp, X. Li, J.-M. Frahm, and M. Pollefeys. 3d model matching with viewpoint-invariant patches (vip). In *2008 IEEE Conference on Computer Vision and Pattern Recognition*, pages 1–8. IEEE, 2008.

[50] S. Qian, L. Jin, and D. F. Fouhey. Associative3d: Volumetric reconstruction from sparse views. In *European Conference on Computer Vision*, pages 140–157. Springer, 2020.

[51] R. Cai, B. Hariharan, N. Snavely, and H. Averbuch-Elor. Extreme rotation estimation using dense correlation volumes. *arXiv preprint arXiv:2104.13530*, 2021.

[52] Y. Zhu, R. Mottaghi, E. Kolve, J. J. Lim, A. Gupta, L. Fei-Fei, and A. Farhadi. Target-driven visual navigation in indoor scenes using deep reinforcement learning. In *2017 IEEE international conference on robotics and automation (ICRA)*, pages 3357–3364. IEEE, 2017.

[53] P. Mirowski, R. Pascanu, F. Viola, H. Soyer, A. J. Ballard, A. Banino, M. Denil, R. Goroshin, L. Sifre, K. Kavukcuoglu, et al. Learning to navigate in complex environments. *arXiv preprint arXiv:1611.03673*, 2016.

[54] M. Savva, A. Kadian, O. Maksymets, Y. Zhao, E. Wijmans, B. Jain, J. Straub, J. Liu, V. Koltun, J. Malik, et al. Habitat: A platform for embodied ai research. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 9339–9347, 2019.

[55] E. Wijmans, A. Kadian, A. Morcos, S. Lee, I. Essa, D. Parikh, M. Savva, and D. Batra. Dd-ppo: Learning near-perfect pointgoal navigators from 2.5 billion frames. *arXiv preprint arXiv:1911.00357*, 2019.

[56] C. Rascon and I. Meza. Localization of sound sources in robotics: A review. *Robotics and Autonomous Systems*, 96:184–210, 2017.

[57] I. An, M. Son, D. Manocha, and S.-E. Yoon. Reflection-aware sound source localization. In *2018 IEEE International Conference on Robotics and Automation (ICRA)*, pages 66–73. IEEE, 2018.

[58] C. Gan, Y. Zhang, J. Wu, B. Gong, and J. B. Tenenbaum. Look, listen, and act: Towards audio-visual embodied navigation. *arXiv preprint arXiv:1912.11684*, 2019.

[59] C. Chen, Z. Al-Halah, and K. Grauman. Semantic audio-visual navigation. *arXiv preprint arXiv:2012.11583*, 2020.

[60] S. E. Palmer. *Vision science: Photons to phenomenology*. MIT press, 1999.

[61] D. Gandhi, L. Pinto, and A. Gupta. Learning to fly by crashing. In *2017 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS)*, pages 3948–3955. IEEE, 2017.

[62] S. Hershey, S. Chaudhuri, D. P. Ellis, J. F. Gemmeke, A. Jansen, R. C. Moore, M. Plakal, D. Platt, R. A. Saurous, B. Seybold, et al. Cnn architectures for large-scale audio classification. In *2017 ieee international conference on acoustics, speech and signal processing (icassp)*, pages 131–135. IEEE, 2017.

[63] K. He, X. Zhang, S. Ren, and J. Sun. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, pages 770–778, 2016.

[64] J. Deng, W. Dong, R. Socher, L.-J. Li, K. Li, and L. 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] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. In *International conference on machine learning*, pages 1139–1147. PMLR, 2013.

[66] D. P. Kingma and J. Ba. Adam: A method for stochastic optimization. 2015.

[67] H. Chen, W. Xie, A. Vedaldi, and A. Zisserman. Vggsound: A large-scale audio-visual dataset. In *ICASSP 2020-2020 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 721–725. IEEE, 2020.

[68] E. Fonseca, J. Pons Puig, X. Favory, F. Font Corbera, D. Bogdanov, A. Ferraro, S. Oramas, A. Porter, and X. Serra. Freesound datasets: a platform for the creation of open audio datasets. In *Hu X, Cunningham SJ, Turnbull D, Duan Z, editors. Proceedings of the 18th ISMIR Conference; 2017 oct 23-27; Suzhou, China.[Canada]: International Society for Music Information Retrieval; 2017. p. 486-93.* International Society for Music Information Retrieval (ISMIR), 2017.

[69] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, Ł. Kaiser, and I. Polosukhin. Attention is all you need. In *Advances in neural information processing systems*, pages 5998–6008, 2017.

Supplementary Material for: Structure from Silence

A Dataset Examples

We show video samples of *Static* recordings (`static_near.mp4` and `static_far.mp4`) and video samples of *Motion* recordings (`motion.mp4`) along with audio.

B Additional robot navigation experiments

We run more robot navigation experiments to further verify the generalization ability to learn from ambient sound. For these, we use a robot equipped with two microphones, oriented in opposite directions, simply to avoid the need for the robot to rotate to sense both sides of the scene (the model is otherwise unchanged).

Experiment setting. We evaluate our model in 3 additional rooms, different from the one in the main paper. These 4 rooms are located in 4 individual buildings and are not included in the training set. Their floor plans and track designs are shown in Fig. 11. Similarly, the robot starts from one of several unknown locations (40cm, 80cm, or 120cm from the wall) and orientations (30° left or right, or facing forward). For each room, we repeat the experiment 9 times and measure the longest distance along the track it attained before colliding with a boundary. For the 'L'-shaped track in Room #2, we determine the distance by projecting the final position to the center line of the track.

Algorithm. The detailed algorithm to control the robot is shown in Algorithm 1. When computing the near-wall probability $P(L)$ and $P(R)$, we average the predictions of 20 clips of 3s audio to make the prediction more accurate and robust. When running the experiments in different rooms, we manually adjust the threshold p, which essentially controls the expected distance to the wall, to accommodate the different environments.

Experimental results. We compare with the same "straight line" policy as in the main paper. We measure the performance by computing the percentage of the completed track, and take the average of all trials, as well as the average of the trials starting with the same orientation, as shown in Fig. 10. It can be seen that the "straight line" policy performs well when reaching the goal does not require turning, as expected. However, it performs significantly worse than our model when the robot starts at an angle that does not follow the wall. In contrast, our model's performance is more consistent with all three starting orientations, indicating that the model successfully guides the robot to navigate along the wall. When looking at the average performance with different starting orientation on individual rooms, we also find that the "straight line" policy only outperforms our method when starting facing front on straight tracks, and performs worse than ours with any other starting orientation or on the 'L'-shaped track in Room #2.

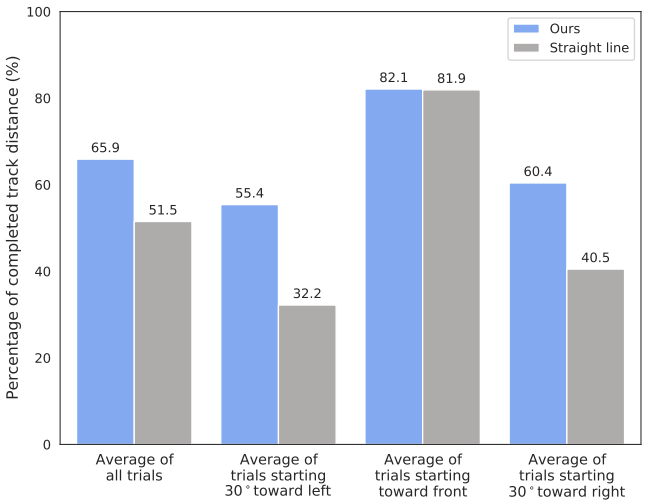
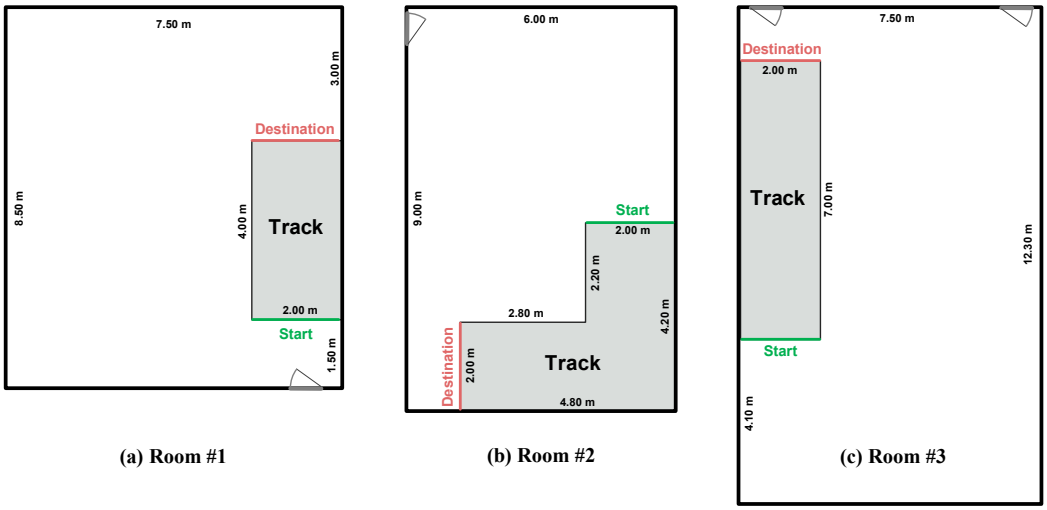


Figure 10: Additional Robot navigation results.

Vision model. While we suspect that vision-based navigation is very effective (and is capable of significantly outperforming audio-based navigation), we found that a vision-based model, when trained solely on our paired RGB-D data, did not generalize well to out-of-distribution viewpoints. Therefore, it performed very poorly (worse than the audio-based model) when deployed on a robot. To make the vision baseline work, we collect extra data in the similar scenes with more diverse viewpoints and finetune the model to apply to the robotic system.

Algorithm for vision model. Additional, we provided the detailed algorithm for the vision model in our paper, which is shown in Algorithm 2. To keep the control algorithm as close as possible to the controller for audio model, we made a small modification to Algorithm 1 that the turning angle is



(a) Room #1 (b) Room #2 (c) Room #3

Figure 11: Classroom floor plans and corresponding track settings.

based on the prediction of near-wall side. The reason for this is to avoid the noise from the prediction of far-wall side, considering the threshold p is usually very high (we set it as 0.97 in the experiment).

Video demo. We provided demo video (`robot_demo.mp4`) in the supplementary. Please check the video for qualitative results of robot navigation.

C Obstacle detection on Static-dense recordings

We evaluate the obstacle detection task for *Static-dense* recordings by training a model on 3 rooms and testing on an unseen room with the same method as the main paper. Furthermore, to evaluate whether the success rate for wall perception increases when participants have self-motion [6], we simulate the motion of the recorder by concatenating two audio with the same angle from adjacent grids and predict whether the agent is moving towards or far away from the obstacles. We report the performance of the model (**Ours-Static**) trained on *Static* recordings for comparison as well.

Table 6: **Obstacle detection** for *Static-dense* recordings. Here, **S** denotes *static* and **M** means *motion*.

Model	Task	AP(%)	Acc(%)
Chance	S	46.4	50.0
Chance	M	52.4	49.8
Ours	S	**96.4**	**83.9**
Ours	M	**98.3**	**94.2**
Ours-Static	S	93.9	84.6

As shown in Table 6, the average precision above 90% in *Static-dense* recordings with both settings can claim the performance is extremely good. Comparing experimental results of with or without motion, we can find the network predicts better with motion involved which denotes motion of the agent improves the performance in the obstacle detection task. We also evaluate our *Static-dense* model on *Static* recordings and it could achieve 62.8% AP and 60.9% Acc, close to the *Static* model. Obstacle detection video demo of Fig. [5] is available in the supplementary (`obstacle_detect_video_demo.mp4`).

D Indoor localization on Static-dense recordings

We are also interested in if ambient sounds can tell us the actual location in the rooms, therefore, we conduct indoor localization experiment for *Static-dense* recordings.

We use the same network as the obstacle detection task and replace the last linear layer with a N-class FC layer, where N is the number of grids in the room. We formulate it as a multi-way classification task and predict which grid the given audio comes from. We train the model with the cross entropy loss and evaluate it with top-1, top-5 and average distance (*i.e.*, the Euclidean distance distance between our predicted grid and the ground truth grid).

Grid classification. We ask to what extent ambient sound can allow a model to predict which grid cell within a room the audio was recorded in. For the train/test split, we randomly select audio samples with three angles in each grid cell as training, then test on sound with the remaining angle. The experimental results are shown in Table 7. It can be seen that the performance is significantly higher the random guessing baselines. Interestingly, we see that our model does not generalize well to other rooms. We believe that this is because there is a great deal of variation within a room. While certain aspects of the sound are predictable, such as distance to a wall, this is not true in general

Table 7: **Indoor Localization** for *Static-dense* recordings.

EXP.	Room 4			Room 5			Room 6		
	Random	Non time-shift	Time-shift	Random	Non time-shift	Time-shift	Random	Non time-shift	Time-shift
Top-1 (%)	4.00	32.3	18.3	2.86	29.5	18.8	4.00	34.3	22.8
Top-5 (%)	20.0	77.7	50.3	14.5	70.8	51.8	20.0	74.3	55.6
Avg. Distance	3.17	1.23	1.71	3.59	1.98	2.17	3.17	1.31	1.87

Generalization over time. Next, we evaluate our model's ability to predict the audio samples recorded from a different time to see how well the model could generalize over time. In this experiment, training and test sets are from the same room but different times. The time intervals for Room #4, #5 and #6 are 1 week, 1 week, and 1 hour, respectively.

We show the experimental results in Table 7. It can be seen that the performances drops compared to the non time-shifted experiments, confirming the assumption that the network does cheat using global time changing of audio. Yet the performances are still significantly better than random guessing, even recording the audio after one week later. Overall, this experiment shows that the ambient sound conveys information of location, but that the sounds that one hears in a room may change over time.

E Relative depth order on Motion recordings

We design experiments to investigate the accuracy of relative depth estimation as the magnitude of the depth difference changes. We paired audio in the test set using 5 different distance ranges. We test on those sampled data without re-training and compare audio-based and vision-based methods (with random initialization). The results are shown in Figure 12. We see that AP and accuracy increase as the relative depth increases, and the models are more confident when the distances are at least 0.2 meters. Along with distance increasing, the gap between vision-based and audio-based method are reduced.

F Generalization between *Motion* and *Static* recordings

We give the results of the model trained on Motion recordings (recorded in hallways) and tested on Static recording (recorded in the classrooms). The experiment results are shown in Table 8 . Our model can still perform well above random chance despite evaluating on a different domain.

Table 8: **Generalization between *motion* and *static* recordings** on obstacle detection and depth order task.

	AP(%)	Acc(%)
Obstacle detection	61.3	51.9
Depth order	78.8	70.2

G Effects of Non-ambient sounds

To investigate how non-ambient sounds from environment could affect our prediction, we augment Static recordings with various clear sounds from FreeSound Dataset [68] to simulate other sound source in the scenes and re-perform the obstacle detection experiment. We test our models without or with re-training on the augmented data and the results are shown in Fig. 13. We can see that the performance drops as the volume of distracting sound increases while it still outperforms chance and re-training the model with augmentation makes our model more robust.

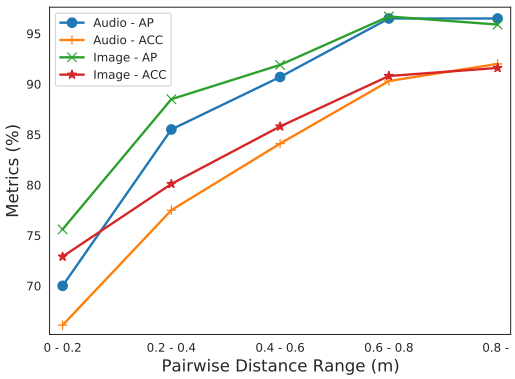


Figure 12: Relative depth order performance vs. distance.

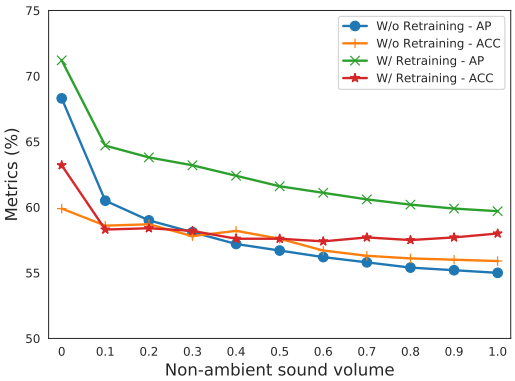


Figure 13: Obstacle detection performance vs.volume of distracting sound.

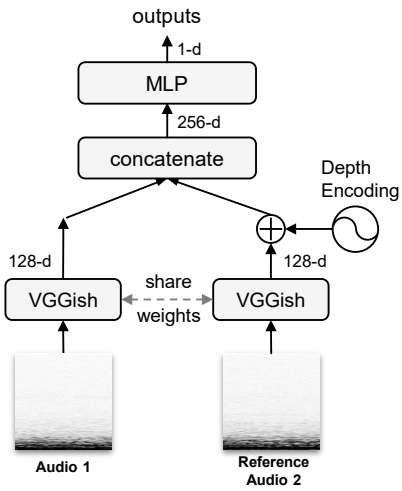


Figure 14: Conditional absolute depth model.

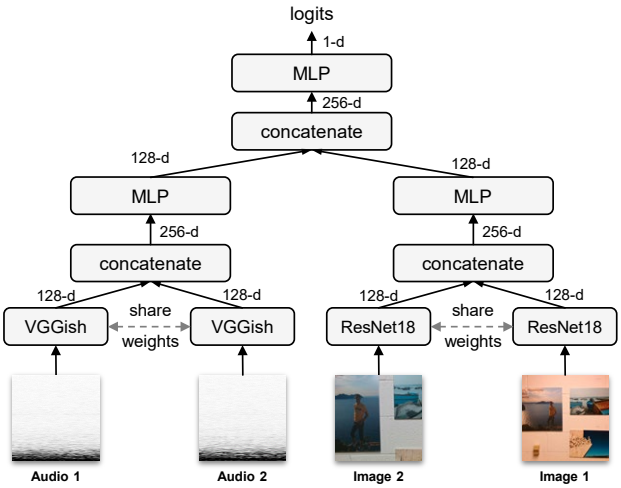


Figure 15: The detail of AV-Order model.

H Conditional absolute depth estimation

Implementation. We use a Siamese network from relative tasks to build conditional absolute depth (**CAD**) model. With three inputs: audio s_1, reference audio s_2 and ground-truth depth x_2, we first extract audio features with VGGish backbone and calculate the depth embedding following [69]. We add up the depth embedding and reference audio feature and fuse features via concatenation prior to the multi-layer perceptron (*i.e.*, FC-ReLU-FC-ReLU-FC layers). During the training, the reference sounds are randomly selected from the same scene while those are fixed for each sample in the test set (Fig. 14).

I Self-supervised audio-visual learning

Implementation. We approach AV-Order model by building two Siamese networks for both audio and visual modalities. As shown in Fig. 15, for each modality, we first extract features from inputs and concatenate them prior to the multi-layer perceptron. Then we concatenate two 128-d feature vectors from image and audio branches, and forward it to MLP (*i.e.*, FC-ReLU-FC layers, all the MLP blocks are the same) for the final logits prediction.

AudioSet feature. The low performance of the AudioSet features in Table 4, 5 could be due to differences in the types of sounds considered, as well as the preprocessing in each method. The preprocessing procedure used in the AudioSet model (which is standard on the dataset) throws away all but the 125Hz-7500Hz frequency range, which our analysis (Section 5.3) suggests is important. To test this, we trained a random feature baseline with (and without) this preprocessing, and see a large difference in their performances (Table 4, 5).

Few-shot learning. To help understand how audio-visual self-supervision improves fine-tuned models, we measure the performance using various numbers of labeled examples as training set (5%, 10%, 20%, 40%, 100%). The experiment results are shown in 16. We can see a large improvement from self-supervised initialization when there is more unlabeled data than labeled data, especially in few-shot training regimes. As expected, when the number of labeled examples equals the number of unlabeled examples in our experiment, the trained-from-scratch model catches up since the two datasets are exactly the same and the labels provide strictly more information than the audio.

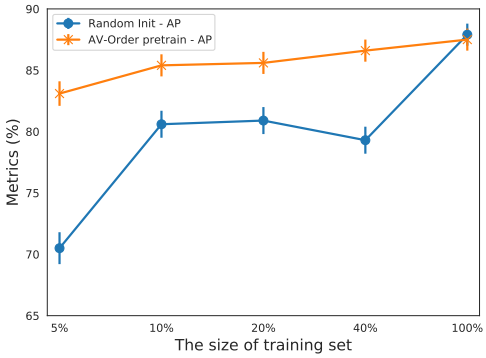


Figure 16: Relative depth order performance vs. training set size.

J VGGSound-Instrument dataset

To perform the frequency analysis experiment, we sampled 37 video classes of musical instruments with 32k video clips of 10s length from VGGSound dataset, and we call this subset VGGSound-Instruments. Video classes are listed as below:

playing accordion	playing acoustic guitar	playing banjo	playing bass drum	playing bass guitar
playing bongo	playing cello	playing clarinet	playing congas	playing cornet
playing cymbal	playing djembe	playing double bass	playing drum kit	playing electric guitar
playing electronic organ	playing erhu	playing flute	playing glockenspiel	playing guiro
playing hammond organ	playing harp	playing harpsichord	playing mandolin	playing marimba, xylophone
playing piano	playing saxophone	playing sitar	playing snare drum	playing steel guitar, slide guitar
playing tabla	playing timbales	playing trumpet	playing ukulele	playing vibraphone
playing violin, fiddle	playing zither			

Algorithm 1 Policy for navigating along the wall using ambient sound

Require: threshold p, α; step length v
while No crash **do**
  Input: left and right side sound
  $P(L)$ = Network(left sound)
  $P(R)$ = Network(right sound)
  Angle $\propto |0.5(P(L) + P(R)) - p|$
  if $|P(L) - P(R)| < \alpha$ **then**
    Go forward
  else
    if $0.5(P(L) + P(R)) > p$ **then**
      # Robot might be near to the wall
      if P(L) > P(R) **then**
        Turn right and go forward
      else
        Turn left and go forward
      end if
    else
      # Robot might be far from the wall
      if P(L) > P(R) **then**
        Turn left and go forward
      else
        Turn right and go forward
      end if
    end if
  end if
end while

Algorithm 2 Policy for navigating along the wall using RGB images

Require: threshold p; step length v
Determine which side is near-wall side
Input: left and right RGB image
$P(L)$ = Network(left RGB image)
$P(R)$ = Network(right RGB image)
if $P(L) < P(R)$ **then**
  Near-wall side is on the robot's right; Far-wall side is on the robot's left
else
  Near-wall side is on the robot's left; Far-wall side is is on the robot's right
end if
while No crash **do**
  Input: left and right RGB image
  $P(N)$ = Network(near-wall side RGB image)
  Angle $\propto |P(n) - p|$
  if $P(N) > p$ **then**
    # Robot might be near to the wall
    Turn to the far-wall side and go forward
  else
    # Robot might be far from the wall
    Turn to the near-wall side and go forward
  end if
end while