Geometry-Aware Multi-Task Learning for Binaural Audio Generation from Video

Rishabh Garg
rishabh@cs.utexas.edu
The University of Texas at Austin

Ruohan Gao
rhgao@cs.stanford.edu
Stanford University

Kristen Grauman
grauman@cs.utexas.edu
Facebook AI Research

The Supplementary material consists of

A. Supplementary Video

B. RIR Prediction Case Study

C. SimBinaural Dataset Details

D. Implementation Details

E. Additional Ablations

A Supplementary Video

In our supplementary video, we show (a) examples of our SimBinaural dataset; (b) example results of the binaural audio prediction task on both SimBinaural and FAIR-Play datasets; and (c) examples of the interface for the user studies.

B RIR Prediction Case Study

We perform a case study on the task of predicting the binaural IR directly from a single visual frame. This helps us evaluate if it is feasible to learn this information just from a visual frame, so that it can be then used for our task as in Sec. 3.2 of the main paper. We predict the acoustic properties of the room by looking at one snapshot of the scene. We predict the magnitude spectrogram of the IR for the two channels. We also obtain the predicted waveform of the IR using the Griffin-Lim algorithm [5]. Figure 1 shows qualitative examples of predictions from the test set. It can be seen that we can get a fairly accurate general idea of the IR, and the difference between the response in each channel is also captured.

To evaluate if we capture the materials and geometry effectively, we train another task to predict the reverberation time RT_{60} of the IR from the visual frame. A more accurate prediction of RT_{60} means that our network understands how the wave will interact with the room and materials and whether it takes more or less time to decay. We formulate this as a classification task and discretize the range of the RT_{60} into 10 classes, each with roughly equal number of samples. We then use a classifier to predict this range class of RT_{60} using only the visual frame as input. The classifier, consisting of a ResNet-18, has a test accuracy of 61.5% which demonstrates the networks' ability to estimate the RT_{60} range quite well.

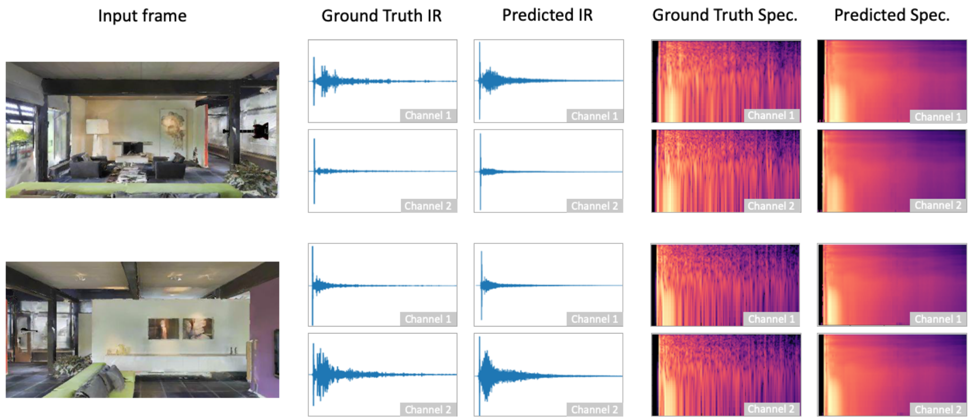

Figure 1: IR Prediction: The first column is the input frame to the encoder. The second column depicts the ground truth IR for the frame and the fourth column is the corresponding spectrogram of this IR. The third and fifth columns show the predicted IR waveform and spectrogram, respectively. This predicted IR waveform is estimated from the spectrogram generated by our network.

C SimBinaural dataset details

To construct the dataset, we insert diverse 3D models of various instruments like guitar, violin, flute etc. and other sound-making objects like phones and clocks into the scene. Each object has multiple models of that class for diversity, so we do not associate a sound with a particular 3D model. We have a total of 35 objects from 11 classes.

To generate realistic binaural sound in the environment as if it is coming from the source location and heard at the camera position, we convolve the appropriate SoundSpaces [1] room impulse response with an anechoic audio waveform (e.g., a guitar playing for an inserted guitar 3D object). We use sounds recorded in anechoic environments, so there is no existing reverberations to affect the data. The sounds are obtained from Freesound [3] and OpenAIR data [8] to form a set of 127 different sound clips spanning the 11 distinct object categories. Using this setup, we capture videos with simulated binaural sound.

The virtual camera and attached microphones are moved along trajectories such that the object remains in view, leading to diversity in views of the object and locations within each video clip. Using ray tracing, we ensure that the object is in view of the camera, and the source positions are densely sampled from the 3D environments. For a particular video, we use a fixed source position and the agent traverses a random path. The view of the object changes throughout the video as the camera moves and rotates, so we get diverse orientations of the object and positions within a video frame, for each video. The camera moves to a new position every 5 seconds and has a small translational motion during the five-second interval. The videos are generated at 5 frames per second, the average length of the videos in the dataset is 30.3s and the median length is 20s.

D Implementation Details

All networks are written in PyTorch [9]. The backbone network is based upon the networks used for 2.5D visual sound [4] and APNet [10]. The visual network is a ResNet-18 [6] with the pooling and fully connected layers removed. The U-Net consists of 5 convolution

Method	STFT	ENV
Spatial+Geometric	0.724	0.118
IR Pred+Geometric	0.707	0.117
IR Pred+Spatial	0.702	0.117

Table 1: Results on SimBinaural Position-Split with different combinations of constraints.

layers for downsampling and 5 upconvolution layers in the upsampling network and include skip connections. The encoder for spatial coherence follows the same architecture as the U-Net encoder for the audio feature extraction. The classifier combines the audio and visual features and uses a fully connected layer for prediction. The generator network is adapted from GANSynth [2], modified to fit the required dimensions of the audio spectrogram.

To preprocess both datasets, we follow the standard steps from [4]. We resampled all the audio to 16kHz and computed the STFT using a FFT size of 512, window size of 400, and hop length of 160. For training the backbone, we use 0.63s clips of the 10s audio and the corresponding frame. Frames are extracted at 10fps. The visual frames are randomly cropped to 448 × 224. For testing, we use a sliding window of 0.1s to compute the binaural audio for all methods.

We use the Adam optimizer [7] and a batch size of 64. The initial learning rates are 0.001 for the audio and fusion networks, and 0.0001 for all other networks. We trained the FAIR-Play dataset for 1000 epochs and SimBinaural for 100 epochs. We train the RIR prediction separately and use the weights for initialization while training jointly. The δ for choice of frame is set to 1s and the λ's used are set based on validation set performance to $\lambda_B = 10, \lambda_S = 1, \lambda_G = 0.01, \lambda_P = 1$.

E Additional Ablations

Table 2 in the main paper illustrates that adding each component of our method individually to the visual features helps improve the binaural audio quality performance. Table 1 provides additional analysis to evaluate the combination of different constraints with the backbone for the SimBinaural Position-Split. The constraints complement each other to learn better visual features, leading to better audio performance.

References

[1] 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.

[2] Jesse Engel, Kumar Krishna Agrawal, Shuo Chen, Ishaan Gulrajani, Chris Donahue, and Adam Roberts. Gansynth: Adversarial neural audio synthesis. In *ICLR*, 2019.

[3] Frederic Font, Gerard Roma, and Xavier Serra. Freesound technical demo. In *Proceedings of the 21st ACM International Conference on Multimedia*, 2013.

[4] Ruohan Gao and Kristen Grauman. 2.5d visual sound. In *CVPR*, 2019.

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

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, 2016.

[7] Diederik Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015.

[8] Damian T Murphy and Simon Shelley. Openair: An interactive auralization web resource and database. In *Audio Engineering Society Convention 129*. Audio Engineering Society, 2010.

[9] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, Alban Desmaison, Andreas Kopf, Edward Yang, Zachary DeVito, Martin Raison, Alykhan Tejani, Sasank Chilamkurthy, Benoit Steiner, Lu Fang, Junjie Bai, and Soumith Chintala. Pytorch: An imperative style, high-performance deep learning library. In *NeurIPS*. 2019.

[10] Hang Zhou, Xudong Xu, Dahua Lin, Xiaogang Wang, and Ziwei Liu. Sep-stereo: Visually guided stereophonic audio generation by associating source separation. In *ECCV*, 2020.