Supplementary Material of AV-NeRF

Anonymous Author(s)
Affiliation
Address
email

A Demo Videos

For convincing results, we have created a webpage that contains the rendering videos of multiple scenes. **We highly recommend readers visit this webpage** (the "index.html" file in the folder). Please note that the webpage may not be fully supported on Safari browser; therefore, we recommend using Google Chrome for optimal viewing experience.

B Architectures

A-NeRF. A-NeRF consists of two Multilayer Perceptrons (MLPs), each comprising four linear layers with an additional residual connection. The width of each linear layer, denoted as c, is set to 128 for the RWAVS dataset and 256 for the SoundSpaces dataset. In A-NeRF, all linear layers are followed by ReLU activation layers, except for the last layer, where the ReLU activation is replaced with the Sigmoid function. The first MLP takes the listener's position (x, y) and the frequency $f \in [0, F]$ as input, where F represents the number of frequency bins. It predicts a mixture mask $m_m \in \mathcal{R}$ for the given frequency f and generates a feature vector with c channels. Prior to feeding them into the MLP, we apply positional encoding to the listener's position (x, y) and the frequency f. We set the maximum frequency used for positional encoding as 10.

Then, we adopt relative transformation (Sec. 4.4) to project the listener's direction θ into a high-frequency space. We concatenate the transformed listener's direction and the feature vector, and feed it into the second MLP. The second MLP is appended with a Sigmoid layer and a scaling layer, ensuring that the difference mask m_d estimated by the second MLP falls within the range of $[-1, 1]$. For each frequency query f, A-NeRF estimates two masks: m_m and m_d, both of which are scalars. We iterate over all frequencies $f \in [0, F]$ to obtain the complete masks $\mathbf{m}_m$ and $\mathbf{m}_d$. After computing the masks, we synthesize the target audio a_t according to the procedure discussed in Sec. 4.2.

V-NeRF. We utilize the nerfacto model provided by nerf-studio [1] as the V-NeRF. This model combines several well-established and successful methods, including camera pose refinement [2, 3], image appearance conditioning [4], hash encoding, and proposal sampling [5]. Due to its robust and effective performance on real-world data, we utilize the default settings of the nerfacto model without making any architectural modifications. For more detailed information regarding the architecture of V-NeRF, please refer to the documentation provided by nerf-studio.

AV-Mapper. For each camera pose, we render both RGB and depth images using V-NeRF. We resize images to 256×256 and center-crop to 224×224 prior to feeding them into a frozen ResNet-18 [6] image encoder pre-trained on ImageNet-1K dataset [7]. ResNet-18 embeds the input image as a 512-dimension feature vector. We concatenate the RGB and the depth feature vectors, and input them into the AV-Mapper to learn environmental knowledge of the sound acoustics. AV-Mapper projects the input feature vectors to a latent embedding of c channels. We implement the AV-Mapper as a 3-layer MLP, with each intermediate linear layer followed by a ReLU activation function.

Submitted to 37th Conference on Neural Information Processing Systems (NeurIPS 2023). Do not distribute.

C Implementation Details

RWAVS Dataset. We implement our method using the PyTorch framework [8]. We employ Adam optimizer [9] with $\beta_1 = 0.9$ and $\beta_2 = 0.999$ for model optimization. The initial learning rate is set to $5e{-}4$ and exponentially decreased to $5e{-}6$. We train the model for 100 epochs with a batch size of 32.

Before feeding the camera position (x, y) to A-NeRF, we normalize it within the range $[-1, 1] \times [-1, 1]$ and apply positional encoding [10]. Additionally, we resample all audios to a frequency of 22050 Hz and utilize the Short-Time Fourier Transform (STFT) to convert waveform audios into the time-frequency domain. For this transformation, we set the number of ffts as 512, the window length as 512, and the hop length as 128. A Hanning window is applied during the process. Finally, we compute both the magnitude and the phase from the spectrogram.

We use magnitude distance (MAG) [11] and envelope distance (ENV) [12] as evaluation metrics for audio quality. The MAG metric quantifies audio quality in the time-frequency domain and is defined as follows:

$$\mathrm{MAG}(\mathbf{m}_{\mathrm{prd}}, \mathbf{m}_{\mathrm{gt}}) = ||\mathbf{m}_{\mathrm{prd}} - \mathbf{m}_{\mathrm{gt}}||^2 \ , \tag{1}$$

where $\mathbf{m}_{\mathrm{prd}}$ is the predicted magnitude, and $\mathbf{m}_{\mathrm{gt}}$ is the ground-truth magnitude. ENV metric that measures the audio quality in the time domain is formatted as:

$$\mathrm{ENV}(a_{\mathrm{prd}}, a_{\mathrm{gt}}) = ||\mathrm{hilbert}(a_{\mathrm{prd}}) - \mathrm{hilbert}(a_{\mathrm{gt}})||^2 \ , \tag{2}$$

where a_{prd} is the predicted audio, a_{gt} is the ground-truth audio, and $\mathrm{hilbert}$ is the Hilbert transformation function [13].

SoundSpaces Dataset. Our model is trained on the SoundSpaces dataset using the same training settings as RWAVS dataset. We resample the impulse responses to 22050 Hz following INRAS [14]. The 2D position is normalized to $[-1, 1] \times [-1, 1]$ prior to positional encoding.

We tailor A-NeRF for impulse response prediction with some minor modifications: (1) The input frequency query f is replaced by a time query $t \in [0, T]$, where T represents the length of an impulse response signal; (2) the first MLP only generates a feature vector while discarding the mixture mask $\mathbf{m}_m$; (3) the second MLP predicts impulse response signals instead of difference mask $\mathbf{m}_d$.

Since the generated impulse responses are in the time domain, we employ STFT to convert them into the time-frequency domain and calculate their magnitudes. We utilize an STFT configuration with 512 FFTs, a sliding window width of 512, a hop stride of 128, and a Hanning window. We supervise the model training using the L2 distance between the ground-truth magnitudes and the predicted magnitudes.

For performance evaluation, we choose three metrics: T60, C50, and EDT [14]. T60 characterizes the reverberation effects in an audio signal by measuring the time it takes for the audio's energy to attenuate by 60 dB. The T60 distance is calculated as follows:

$$\mathrm{T60}(a_{\mathrm{prd}}, a_{\mathrm{gt}}) = \frac{|\mathrm{T60}(a_{\mathrm{prd}}) - \mathrm{T60}(a_{\mathrm{gt}})|}{\mathrm{T60}(a_{\mathrm{gt}})} \ , \tag{3}$$

where a_{prd} and a_{gt} are the predicted and ground-truth impulse responses, respectively. C50 quantifies the energy ratio between early reflections and late reverberation, allowing it to represent the clarity and loudness of the audio. We format the C50 distance as:

$$\mathrm{C50}(a_{\mathrm{prd}}, a_{\mathrm{gt}}) = |\mathrm{C50}(a_{\mathrm{prd}}) - \mathrm{C50}(a_{\mathrm{gt}})| \ . \tag{4}$$

The EDT metric shares similarities with T60 but places greater emphasis on capturing the early reflections of impulse responses. The EDT distance is defined as follows:

$$\mathrm{EDT}(a_{\mathrm{prd}}, a_{\mathrm{gt}}) = |\mathrm{EDT}(a_{\mathrm{prd}}) - \mathrm{EDT}(a_{\mathrm{gt}})| \ . \tag{5}$$

With these three metrics, we can evaluate the generation quality of impulse responses from different aspects, including clarity, energy, and reverberation.

D Setup of RWAVS Dataset

Recording Devices. We have assembled a recording system, as depicted in Fig. 7, to capture high-quality audio-visual scenes in real-world environments. Our system comprises a 3Dio Free Space XLR binaural microphone for capturing stereo audio, a TASCAM DR-60DMKII for recording and storing audio, and a GoPro Max for capturing accompanying videos.

This system is portable, allowing us to position it flexibly and capture scenes from different camera poses. In addition, we utilized an LG XBOOM 360 omnidirectional speaker to serve as a sound source, which plays music repeatedly. Figure 5 in the main paper illustrates the setup used to record data in four distinct environments: office, house, apartment, and outdoors. Within each environment, we positioned the speaker at multiple locations to capture diverse acoustic effects. Each combination of environment and sound source represents an audio-visual scene. We collected data ranging from 10 to 25 minutes for each scene, resulting in a total collection of 232 minutes (3.8 hours) of diverse data, encompassing various environments and source positions.

Figure 7: Recording system. It comprises a professional binaural microphone, a sports camera, and a recorder.

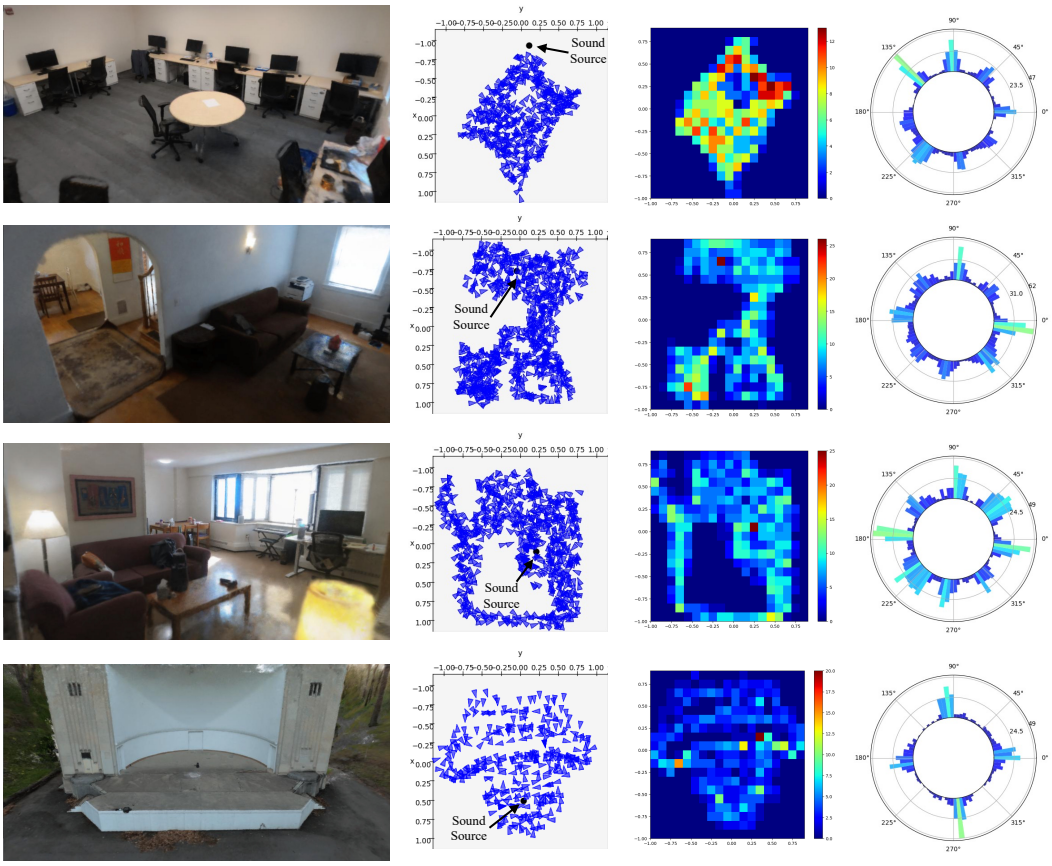

Figure 8: Example scenes. We present several example scenes along with their corresponding camera pose distributions. We display the position density heatmap and the direction distribution map. RWAVS dataset is composed of diverse environments with various camera poses.

Example Scenes. In Fig.8, we present four example scenes from RWAVS dataset along with the corresponding camera pose distributions. The first column showcases images of the example scenes. The second column displays the camera poses used for video recording: the black dot represents the sound source, and each blue triangle represents a camera pose. We normalize all camera poses to the range of $[-1, 1]$ and visualize them in an x-y plane from a top-down view. The third column contains

2D density heatmaps, which illustrate the distribution of camera poses in each unit area: each pixel represents a unit area and its color shows the number of camera poses in this area. As shown in the figure, RWAVS dataset encompasses densely covered camera poses for each environment. We also analyze the distribution of camera directions (shown in the last column of Fig.8). We present the direction distribution in a polar coordinate system with the angle representing the viewing direction and the radius meaning the number of camera poses in this region. RWAVS dataset consists of various viewpoints that approximately cover a 360°range of viewing directions. In summary, the RWAVS dataset comprises diverse environments with a wide range of camera poses.

References

[1] Matthew Tancik, Ethan Weber, Evonne Ng, Ruilong Li, Brent Yi, Justin Kerr, Terrance Wang, Alexander Kristoffersen, Jake Austin, Kamyar Salahi, Abhik Ahuja, David McAllister, and Angjoo Kanazawa. Nerfstudio: A modular framework for neural radiance field development. *arXiv preprint arXiv:2302.04264*, 2023.

[2] Zirui Wang, Shangzhe Wu, Weidi Xie, Min Chen, and Victor Adrian Prisacariu. NeRF−−: Neural radiance fields without known camera parameters. *arXiv preprint arXiv:2102.07064*, 2021.

[3] Chen-Hsuan Lin, Wei-Chiu Ma, Antonio Torralba, and Simon Lucey. Barf: Bundle-adjusting neural radiance fields. In *ICCV*, 2021.

[4] 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 *CVPR*, 2021.

[5] Thomas Müller, Alex Evans, Christoph Schied, and Alexander Keller. Instant neural graphics primitives with a multiresolution hash encoding. *ACM TOG*, 41(4):1–15, 2022.

[6] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Deep residual learning for image recognition. In *CVPR*, pages 770–778, 2016.

[7] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *CVPR*, pages 248–255, 2009.

[8] Adam Paszke, Sam Gross, Francisco Massa, Adam Lerer, James Bradbury, Gregory Chanan, Trevor Killeen, Zeming Lin, Natalia Gimelshein, Luca Antiga, et al. Pytorch: An imperative style, high-performance deep learning library. *NeurIPS*, 32, 2019.

[9] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In Yoshua Bengio and Yann LeCun, editors, *ICLR*, 2015.

[10] 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.

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

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

[13] Julius Orion Smith. *Mathematics of the discrete Fourier transform (DFT): with audio applications*. 2008.

[14] Kun Su, Mingfei Chen, and Eli Shlizerman. Inras: Implicit neural representation for audio scenes. In *NeurIPS*, 2022.