π-AVAS: Can Physics-Integrated Audio-Visual Modeling Boost Neural Acoustic Synthesis?

Supplementary Material

1. Implementation Details

We use the open-sourced Steam Audio library [9] to effectively model the sound occlusion and reflection caused by scene geometry. Compared with the commonly used Pyroomacoustics library [1, 7], Steam Audio can model more complex 3D room geometry, such as a house with multiple rooms, allowing for the simulation of more realistic spatial audio for our problem.

We implement our audio refinement model with the PyTorch framework [6]. We use a total of 30 multi-scale gated convolution blocks and gradually increase the dilation size from 2^0 to 2^9. We use the Adam optimizer [3] with $\beta_1 = 0.9$, $\beta_2 = 0.999$, and a weight decay of $1e{-}4$. We set the learning rate to $5e{-}4$ and train each model for 200 epochs with a batch size of 4. We set the noise scale σ to $1e{-}4$. By default, we use the Euler solver with 4 iteration steps during inference.

2. Generalization Evaluation Setup

To examine the generalization limitation of existing approaches, we adapt the real-world audio-visual RWAVS dataset [4] to satisfy our requirement. The RWAVS dataset contains multiple videos within the same environment while varying the sound source locations. A common practice is to train and evaluate models on individual videos. Here, we design a new evaluation setup: for each environment, we select one video as training data, with the remaining videos used for evaluation. This setup allows us to effectively assess how well existing methods generalize to new sound source locations within the same environment. We show one example environment (apartment) in Fig. 1. In this environment, we display the location of sound sources in each video, as well as the training and evaluation poses. Since the sound source locations vary across videos, this experimental setup effectively measures the generalization ability of different approaches.

3. Audio Refinement With Flow Matching

In our second stage, we design a flow matching model [5, 8] for audio refinement. We present the pseudo-code for model training in Algorithm 1. Given a source audio x_{tx}, simulated audio x_{sim}, and recorded audio (ground-truth audio) x_{rx}, we randomly sample an intermediate audio $\psi_t(x)$ and train the neural network with the flow matching loss.

Here, we also provide the pseudo-code for model inference. We include both the Euler solver (see Algorithm 2)

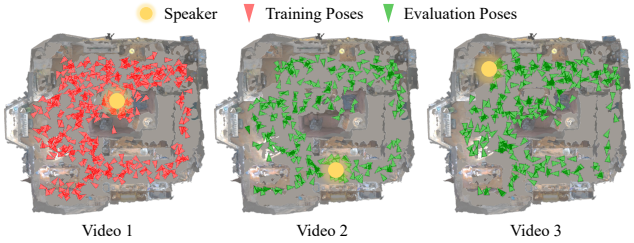

Figure 1. Generalization evaluation setup. In each environment, we select one video for training (red poses) and use the remaining videos for evaluation (green poses). We also show the location of sound sources (speakers).

Algorithm 1 Flow Matching Model Training

Require: Source audio x_{tx}, simulated audio x_{sim}, recorded audio x_{rx}, pose p, noise scale σ, initial network $u_t(\theta)$
while not converge **do**
  $\epsilon \sim \mathcal{N}(0, I)$
  $t \sim \mathcal{U}(0, 1)$
  $\psi_t(x) \leftarrow t x_{\text{rx}} + (1-t)x_{\text{sim}} + \sigma\epsilon$
  $\mathcal{L}_{\text{FM}}(\theta) \leftarrow \|u_t(\psi_t(x), x_{\text{tx}}, p; \theta) - (x_{\text{rx}} - x_{\text{sim}})\|^2$
  $\theta \leftarrow \text{Update}(\theta, \nabla_\theta \mathcal{L}_{\text{FM}}(\theta))$
end while
return θ

and the Midpoint solver (see Algorithm 3). Given a trained network $u_t(\theta)$, a simulated sound x_{sim}, and the number of inference steps n, these solvers iteratively estimate the solution of the Ordinary Differential Equation (ODE):

$$\begin{aligned} \frac{d}{dt}\psi_t(x) &= u_t(\psi_t(x), x_{\text{tx}}, p; \theta), \\ \psi_0(x) &= x_{\text{sim}}. \end{aligned} \tag{1}$$

The solution $\psi_1(x)$ is the estimated binaural audio. Readers may refer to the [2] for the definition of Heun solver used in our experiment.

4. Ablation Studies On Numeric Solver and Inference Step

As discussed earlier, the audio refinement process requires setting the number of iterations and selecting a numerical solver. Here, we evaluate how different flow steps and numerical solvers influence audio quality in a complex apartment environment with challenging acoustics. We compare three solvers—Euler (first-order), Heun [2], and Mid-

Algorithm 2 Euler Solver for Audio Refinement

Require: Trained network $u_t(\theta)$, simulated sound x_{sim}, inference steps n
$t \leftarrow 0$
$\psi_t(x) \leftarrow x_{\text{sim}}$
$\delta \leftarrow 1/n$
while $t < 1$ **do**
  $v \leftarrow u_t(\psi_t(x), x_{\text{tx}}, p; \theta)$
  $\psi_t(x) \leftarrow \psi_t(x) + v\delta$
  $t \leftarrow t + \delta$
end while
$\psi_1(x) \leftarrow \psi_t(x)$
return $\psi_1(x)$

Algorithm 3 Midpoint Solver for Audio Refinement

Require: Trained network $u_t(\theta)$, simulated sound x_{sim}, inference steps n
$t \leftarrow 0$
$\psi_t(x) \leftarrow x_{\text{sim}}$
$\delta \leftarrow 2/n$
while $t < 1$ **do**
  $v' \leftarrow u_t(\psi_t(x), x_{\text{tx}}, p; \theta)$
  $\psi_t(x)' \leftarrow \psi_t(x) + v'\delta$
  $v'' \leftarrow u_t(\psi_t(x)', x_{\text{tx}}, p; \theta)$
  $v = (v' + v'')/2$
  $\psi_t(x) \leftarrow \psi_t(x) + v\delta$
  $t \leftarrow t + \delta$
end while
$\psi_1(x) \leftarrow \psi_t(x)$
return $\psi_1(x)$

point (both second-order)—for different function evaluations (NFE) ranging from 1 to 10. Results are presented in Fig. 2. Although the Heun solver at NFE=4 yields the lowest error, it is highly sensitive to variations in NFE. In contrast, the Euler solver demonstrates strong robustness, with steady performance improvements as NFE increases from 1 to 6, after which its performance stabilizes. The Midpoint solver achieves its best performance at NFE=2, but its performance declines noticeably as NFE increases. Based on these results, we select the Euler solver with NFE=4 as our default inference strategy to strike a balance between efficiency, robustness, and quality.

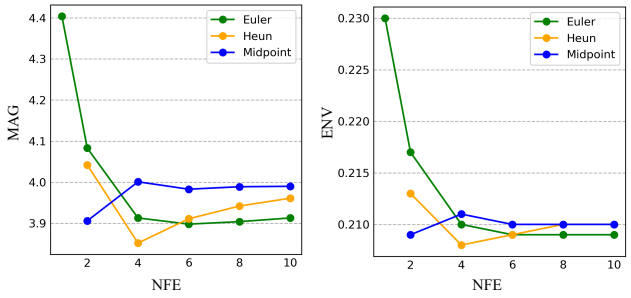

Figure 2. Visualization of the influence of flow steps and numerical solvers on audio generation quality.

References

[1] Ziyang Chen, Israel D Gebru, Christian Richardt, Anurag Kumar, William Laney, Andrew Owens, and Alexander Richard. Real acoustic fields: An audio-visual room acoustics dataset and benchmark. In *CVPR*, pages 21886–21896, 2024. 1

[2] Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35: 26565–26577, 2022. 1

[3] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *ICLR*, 2015. 1

[4] Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu. Av-nerf: Learning neural fields for real-world audio-visual scene synthesis. In *Thirty-seventh Conference on Neural Information Processing Systems*. 1

[5] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matthew Le. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*. 1

[6] 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. 1

[7] Robin Scheibler, Eric Bezzam, and Ivan Dokmanić. Pyroomacoustics: A python package for audio room simulation and array processing algorithms. In *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, pages 351–355. IEEE, 2018. 1

[8] Alexander Tong, Kilian FATRAS, Nikolay Malkin, Guillaume Huguet, Yanlei Zhang, Jarrid Rector-Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative models with minibatch optimal transport. *Transactions on Machine Learning Research*. 1

[9] Valve Corporation. STEAM Audio. https://steamcommunity.com/games/596420/announcements/detail/7745698166044243233, 2024. 1