

GENERATING HIGH-FIDELITY IMPULSE RESPONSES USING CONDITIONAL FLOWS

*Chaitanya Amballa¹, Ayush Barik¹, Haoxiang Lin¹, Zhen Sun², Mehmet Ergezer²,
Vamshi Krishna Chillara², Wontak Kim², Romit Roy Choudhury^{1,2}*

¹University of Illinois Urbana-Champaign, ²Amazon

ABSTRACT

Room impulse responses (RIRs) describe how sound propagates between a source and a listener, and underlie applications ranging from immersive audio to speech enhancement and recognition. RIRs are typically obtained from hybrid acoustic simulators, which combine wave solvers with geometric methods; this is accurate but slow and computationally demanding. We show that generative models can take the place of such a simulator: given a mesh of the room and the source and listener positions, our method CHOIR produces a high-fidelity RIR in a single network evaluation. To reduce ambiguity in this ill-posed problem, a few RIRs measured at different locations in the same room are provided as additional inputs. CHOIR is trained across many rooms via flow matching, and on AcousticRooms it improves reverberation time, clarity, and early-decay time over prior methods in both seen and unseen rooms. We also show results on sim2real transfer to real rooms, and on a downstream speech recognition task, where speech rendered with CHOIR’s RIRs gives error rates closest to those obtained with the ground-truth RIRs.

Project Page at: <https://choirflow.github.io/>

Index Terms— room impulse response, acoustic simulation, generative models, flow matching, speech recognition

1. INTRODUCTION

Room impulse responses fundamentally characterize how a space sounds [1]. It is the signal recorded at a listener when a source emits an impulse, and it carries the effects of reflection, diffraction, and absorption encountered along the propagation path. RIR is a complete representation: convolving it with a dry (anechoic) recording reproduces what would have been heard at the listener, without any further knowledge of the geometry or the materials.

RIRs underpin many applications, including immersive audio in virtual and augmented reality, where what a user hears must stay consistent with the space as they navigate. Furthermore, in speech processing, simulated RIRs are used to build reverberant datasets to train models on far-field recognition, dereverberation, and source separation. RIRs are also central to sound source localization and acoustic echo cancellation. Most of these applications need RIRs in large numbers to be able to train larger and better models.

To cater to this increasing demand for RIRs, high-fidelity hybrid simulators [2, 3, 4] are used. Given a mesh of the room, the material properties of the surfaces, and the source and receiver positions, they combine full-wave solvers at low frequencies and geometric methods (e.g., ray tracing) at higher frequencies to produce the RIR. Although accurate, these simulators are typically slow (on the order of minutes) and computationally expensive. *In this paper*, we seek an alternative and faster route to high-fidelity RIR generation. We ask *can generative models be used as surrogates for high-fidelity RIR simulators?* This would allow training pipelines to generate reverberant data on demand (instead of a fixed precomputed dataset), and AR/VR systems to render audio in real time (order of milliseconds) as the source and receiver move.

Towards this goal, we propose Conditional flows for **H**igh-fidelity **r**oom **I**mpulse **R**esponses (CHOIR). CHOIR takes the same inputs as a simulator, except for the material assignments. Surface materials are typically unknown in practice, making our problem ill-posed: multiple RIRs remain consistent with the same scene, source, and listener positions. Using a predictor model to learn such a multimodal distribution would average over these possibilities and produce an RIR that is inconsistent with any of the plausible RIRs. We therefore turn to generative models that are inherently capable of producing multiple and consistent outputs. We treat the task as conditional generation and train across many rooms with flow matching, which samples among these possibilities rather than averaging over them. Since RIRs are typically long irregular signals, learning a flow over them directly is difficult; we therefore propose building the flow in a compact space derived from a time-frequency view of the response, one that retains certain desirable properties: how the signal decays, when early reflections arrive. To help the model generalize in unknown test scenes, following works in [5, 6], a few reference RIRs (measured at other locations in the scene) are provided as additional evidence. These references implicitly carry material information (e.g., how absorptive the surfaces are), helping the model sharpen its predictions. Modeling this evidence as standalone information, however, is subpar; we show that with careful design, each reference can be paired coherently, letting the model weigh them by their relevance to the queried position.

We evaluate CHOIR on AcousticRooms, comparing against prior methods on reverberation time, clarity, and early-decay time, both in rooms seen during training and in rooms held out entirely. CHOIR improves on all three metrics in both settings. In addition, we also show results on a real-world dataset [7] where we adapt the model trained on AcousticRooms. Since acoustic metrics alone do not fully confirm whether such a model could substitute a simulator, we therefore evaluate on the downstream task of measuring the word error rates from an automatic speech recognition (ASR) system. By rendering speech from both generated and real responses, and transcribing them under various additive noises, we show CHOIR tracks and imitates the ground truth more closely than baselines.

2. BACKGROUND AND RELATED WORK

Modeling Acoustic Fields. Following the remarkable success of the Neural Radiance Fields in vision to synthesize novel views of a scene [8, 9, 10], a line of work has successfully extended them to study room acoustics. Given many measurements inside a space, a network is trained to map a source and listener position to the response between them, sometimes conditioned on visual or geometric observations of the same room [11, 12, 13, 14, 15]. NAF [11] maintains a grid of learnable local features that a small network queries to predict the RIR, and INRAS [14] encodes the source, listener, and surrounding surfaces through dedicated modules so the model explicitly captures spatial structure. Later works increase the context that is available at each query: NACF [12], on RGB and depth

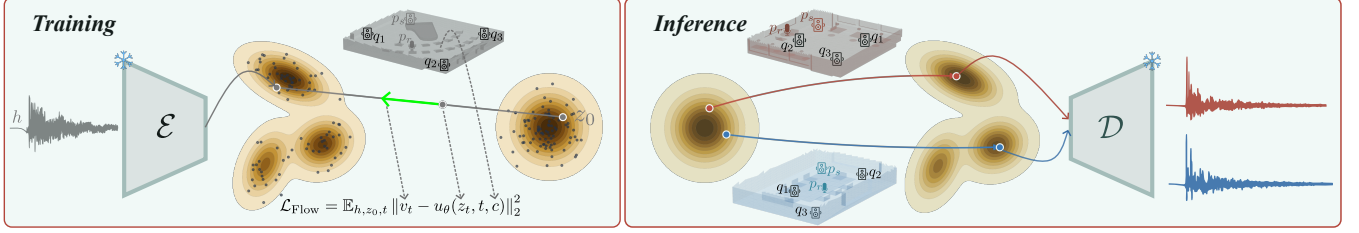


Fig. 1: Overview of CHOIR. On the left is training (Stage 2): an RIR is first encoded by the frozen $\mathcal{E}$ and interpolated with noise to give z_t . u_θ then regresses this interpolant to the velocity under the context c . At inference (right), two contexts (in red and blue) representing completely different test scenarios, carry noise along different trajectories and are decoded into their respective RIRs by $\mathcal{D}$.

images, NeRAF [15] on extracting visual features (colour and density) from a radiance field fitted to the same scene, and MiNAF [13] probes a rough mesh to obtain features (e.g., distances) that describe the local geometry around a queried point. Across these works, the acoustic field is represented *implicitly*: the acoustics of the space are absorbed into the weights of the network fitted to measurements. Consequently, the acoustic field learned from these measurements explains and, more importantly, is valid *only* for that room. We depart from this line of work in that we seek a generalized model that can few-shot adapt to any arbitrary scene, which we discuss next.

Generalizing across scenes. Early work in this direction used generative adversarial networks to synthesize RIRs [16, 17, 18]. Given a source and receiver pair together with the room dimensions and reverberation time in FastRIR [16], an image of the scene in Image2Reverb [18], or a triangle mesh in Mesh2IR [17], they produce the response in a single forward pass. The responses are plausible for the scene as a whole but remain inaccurate at arbitrary placements [5]. To address this, works in [19, 5, 6] rely on a few observations recorded inside the target room and can therefore better adapt to an unseen space with the additional acoustic information. FewShotRIR [19] takes twenty RGB, depth, and binaural observations of the environment, MAGIC [20] adds semantic features from a segmentation network alongside the acoustic and visual inputs, and xRIR [5] reduces the requirement to eight recordings and a single panoramic depth map. These predictions in [5, 19, 20] are deterministic; moreover, surface materials are rarely observed from a handful of measurements; many RIRs are hence consistent with the same inputs. Recently, diffusion and flow-based models have superseded adversarial training across multiple tasks [21, 22, 23], and can sample multiple plausible outputs under the same settings. FLAC [6] builds on this by sampling from a conditional distribution of a learned flow matching model, and is the closest work to ours. An autoencoder is first trained to compress the time-domain RIR, and a conditional flow is then trained in the latent space. CHOIR also adapts this two-stage strategy, and we defer the details to Sec. 3. Finally, we note that simpler non-parametric methods such as a nearest-neighbour estimate or distance-weighted interpolation are faster alternatives; we consider a few as our baselines.

Flow matching is at the core of CHOIR, and we describe the underlying mechanism briefly. The objective is to transport a distribution that is simple to sample from, typically a Gaussian, onto a target distribution of which we only draw samples. Flow matching [23] approaches this by learning a time dependent velocity field $u_\theta(\cdot, t)$ that carries a sample from the source at $t=0$ to the target at $t=1$. The marginal velocity that induces this transport is intractable, but it can be shown that regressing the conditional velocities, which are readily available, recovers it [23]. In the stochastic interpolant framework [24], a path is constructed between a noise sample ε and a data sample x_1 , and the conditional velocity along that path

is known in closed form. In the simplest case of a linear interpolant $x_t = (1-t)\varepsilon + tx_1$ the target velocity is simply $x_1 - \varepsilon$, and regressing u_θ against it over many such pairs recovers the field that transports the source distribution to the target distribution. Generation then amounts to drawing a sample from the source and integrating the learned field forward in small Euler steps until $t=1$.

3. METHOD

3.1. Problem formulation

Let $h \in \mathbb{R}^T$ denote a monaural RIR of T samples between a source at $p_s \in \mathbb{R}^3$ and a receiver at $p_r \in \mathbb{R}^3$ in a scene represented by its mesh $\mathcal{M}$. Given $(\mathcal{M}, p_s, p_r)$, our task is to synthesize h . We do not assume any material assignments on $\mathcal{M}$. Many RIRs can be simulated from the same $(\mathcal{M}, p_s, p_r)$ under different material settings, which makes the problem ill-posed. To reduce this ambiguity and narrow the set of consistent RIRs, we additionally use K reference RIRs $\{h_k\}_{k=1}^K$, measured at the same receiver from K other source positions $\{q_k\}_{k=1}^K$. Our goal is therefore to sample from the conditional distribution $p(h | c)$, where $c = \{\mathcal{M}, p_s, p_r, \{h_k\}, \{q_k\}\}$, at a source and receiver pair in a test scene. We achieve this by first learning a compact representation of h , and then training a conditional flow matching model in that compact space. Next, we describe both stages and the modeling of c .

3.2. Stage 1. Auto-encoding Impulse Responses

RIRs are high-dimensional signals (typically $T \approx 10000$) and non-smooth in time, making them expensive and tricky to model directly. To address these, we first train a variational autoencoder [25] ($\mathcal{E}, \mathcal{D}$) that compresses h into a latent $z = \mathcal{E}(h)$, and can be reconstructed back using $\hat{h} = \mathcal{D}(z)$. We represent h in the complex STFT domain with the real and imaginary parts as two channels, allowing compression along both time and frequency axes. The output channels are finally converted back into a waveform using the inverse STFT. While desirable, just using the reconstruction loss turned out to be not sufficient; matching the exact waveforms in this highly discontinuous space is challenging. Matching the statistics of h such as energy decay and its early peaks, however, is possible and useful for all the downstream tasks. We therefore train our VAE with a multi-resolution STFT loss, an energy decay loss matching the decay curves of h and $\hat{h}$, a weighted time-domain loss that emphasizes the early parts of the response, together with the prior KL term of VAE. Our final loss function becomes:

$$\mathcal{L}_{\text{VAE}} = \mathcal{L}_{\text{MRSTFT}} + \lambda_1 \mathcal{L}_{\text{EDC}} + \lambda_2 \mathcal{L}_{\text{w-time}} + \lambda_3 \mathcal{L}_{\text{KL}}. \quad (1)$$

The additional losses preserve the direct path and the early reflections, which in turn improve the acoustic metrics. Once trained, the VAE is kept frozen for the subsequent training.

Table 1: Reconstruction (blue) and generation (beige) results on AcousticRooms. **Bold** indicates best, and underline is second best.

Method	unseen			seen		
	T60↓	C50↓	EDT↓	T60↓	C50↓	EDT↓
VAE-FLAC	3.57	0.32	14.29	2.43	0.24	10.51
VAE-CHOIR	1.00	0.17	6.12	1.03	0.12	4.91
Nearest neighbor	10.91	2.79	90.08	8.31	1.80	64.77
Fast-RIR [16]	17.71	3.25	121.21	15.19	2.06	86.46
xRIR [5]	9.98	1.35	49.40	6.99	0.92	34.15
FLAC [6]	8.60	0.97	37.13	5.32	0.64	25.69
CHOIR	<u>7.78</u>	<u>0.88</u>	<u>33.97</u>	<u>5.26</u>	<u>0.60</u>	<u>24.73</u>
CHOIR_{ETC}	6.80	0.88	33.39	4.97	0.59	23.91

3.3. Stage 2. Training Conditional Flows with CHOIR

Given a RIR h and its context c , we encode $z_1 = \mathcal{E}(h)$, draw a noise sample $z_0 \sim \mathcal{N}(0, I)$, and interpolate between the two to obtain the stochastic interpolant: $z_t = (1 - t)z_0 + tz_1$, $t \in [0, 1]$. A network u_θ is then trained to regress the conditional velocity $v_t = z_1 - z_0$:

$$\mathcal{L}_{\text{Flow}} = \mathbb{E}_{h, z_0, t} \|u_\theta(z_t, t, c) - (z_1 - z_0)\|_2^2. \quad (2)$$

We sample t from a log-SNR schedule that concentrates on the moderately noisy latents that is found to be effective [6]. Following classifier-free guidance [26], we drop c with a small probability so that the model also learns the unconditional velocity. At inference, we draw z_0 and integrate the learnt velocity u_θ from $t=0$ to $t=1$; we found a single Euler step to be sufficient in our experiments.

$$\hat{z}_1 = z_0 + u_\theta(z_0, 0, c), \quad (3)$$

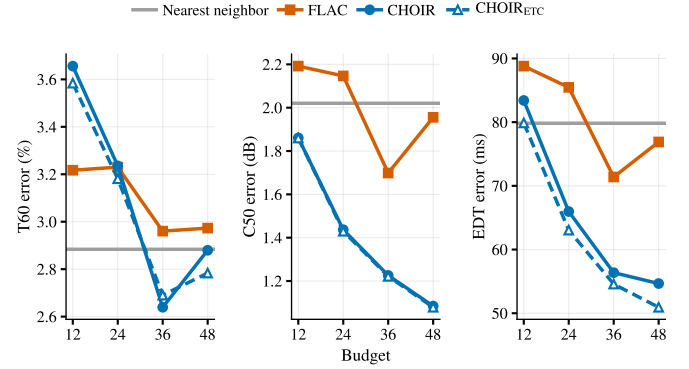
Finally the latent $\hat{z}_1$ decoded back into a RIR as $\hat{h} = \mathcal{D}(\hat{z}_1)$. Fig. 1 visualizes our training (Stage 2.) and inference. We parameterize u_θ with a diffusion transformer (DIT) [27], which operates on z_t as a sequence of tokens. Since z_t is in a time-frequency representation, flattening it into a sequence destroys its underlying grid structure. We therefore replace the (rotary) position embedding [28] of the transformer with its axial 2D variant [29], rotating one half of the head dimension by the frequency index and the other half by its time index, so that each token retains both of its coordinates.

3.4. Designing the Context c

From the scene mesh $\mathcal{M}$, we render a panoramic depth map captured at the receiver (shifted to the origin), following [5, 6]. This map is then oriented towards a specific source position and later encoded with a DINOv3 ViT-S/16 [30] to derive the scene features, D_s for the target source and D_k for the k -th reference. The target source position p_s and the reference source positions $\{q_k\}$ are expressed in the receiver’s frame and encoded with sinusoidal embeddings.

The K references enter the DIT through cross-attention, allowing the model to adaptively interpolate $\{h_k\}_{k=1}^K$ based on their relevance to the target source p_s . Each reference is described by three quantities: its RIR h_k , the position q_k it was measured from, and the geometry D_k . Different from FLAC, which tokenizes the context by modality, we tokenize it by the reference tuple $\langle h_k, q_k, D_k \rangle$ and group them into a single token. This is crucial, as tokenizing by modality leaves this correspondence implicit, forcing the model to infer whether, say, h_j came from q_j or q_k . Grouping by reference instead makes this explicit, helping the model encode the tuple as a whole. The features it learns from a reference then remain tied to the position from which they were measured.

Fig. 2: Performance on real-world rooms. CHOIR and CHOIR_{ETC} outperforms baselines and generalizes well with increasing budgets.



3.5. Performance Improvements from the Late Decay

In practice, we observe that the context c does not constrain the early and late parts of the RIR equally. This is partly expected because: i) the early arrivals are the most dominant and are governed by the geometry between p_s and p_r , ii) the late reverberations tend to be noisy and weaker. These reverberations, however, can be characterised mainly by how their energy decays. Since the context already contains the K references measured in the same room, it is equipped to describe the late decay well. We therefore construct a small network e_ω (consisting of MLPs) to estimate the decay statistics of the energy curve, specifically the energy time curve. Using the references $\{(h_k, q_k)\}_{k=1}^K$ and the target positions (p_s, p_r) , the network is trained separately to predict the energy decay of the target RIR. To arrive at the final prediction, we first obtain a rough estimate of the mixing time using the scene volume [31]. Finally, past the mixing time, we linearly blend (in dB scale) the model’s prediction e_ω into the decay of the decoded RIR $\hat{h}$ to create the final output. We denote this variant CHOIR_{ETC}.

4. EXPERIMENTS AND RESULTS

Datasets. We use AcousticRooms [5] which consists of monaural RIRs at 22.05 kHz along with panoramic depth maps, over 260 rooms. Following prior work, we report results on *seen* rooms (6,217 held-out source-receiver pairs in rooms present during training) and *unseen* rooms (6,337 pairs in rooms held out entirely). Additionally, to show real-world performance, we use Hearing Anything Anywhere [7], which provides measured RIRs from 4 real rooms.

Metrics. We report the standard acoustic metrics comparing the generated and the ground truth RIR: the relative error in reverberation time T60 (%), and the absolute errors in clarity C50 (dB) and early decay time EDT (ms). We compute the word error rate for ASR task.

Baselines and Implementation. We compare CHOIR against (i) a nearest neighbour, which returns the reference RIR measured closest to p_s , (ii) Fast-RIR [16] is adapted to the few-shot setting by conditioning on T60 averaged over the K reference RIRs, and the two few-shot models (iii) xRIR [5] and (iv) FLAC [6]. All methods observe $K=8$ reference RIRs randomly drawn from the scene. We train both stages of CHOIR on 2 NVIDIA RTX 3090 GPUs. Both CHOIR and FLAC use a single Euler step with a guidance scale of one for generation. (We have experimented with more inference steps and the CFG scale parameter, but they did not improve our results.)

4.1. Results

■ **Stage 1.** We set $\lambda_1 = 0.3$, $\lambda_2 = 10$, $\lambda_3 = 1e-4$. Table 1 (top) reports the reconstruction error across acoustic metrics for FLAC and

Table 2: Word error rate (%) under pink and babble noise at varying SNR levels under two ASR models. **Bold** indicates the closest to the ground truth.

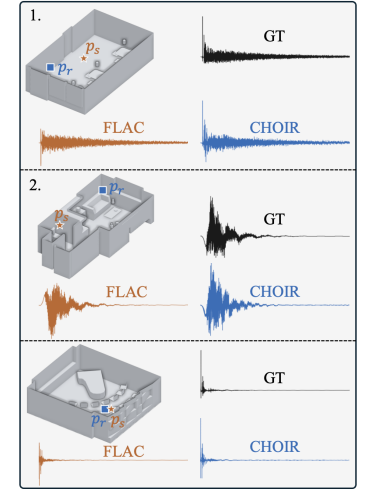
SNR	Pink noise						Babble noise					
	Parakeet			Qwen3			Parakeet			Qwen3		
	True	Ours	FLAC	True	Ours	FLAC	True	Ours	FLAC	True	Ours	FLAC
<i>Unseen rooms</i>												
+10	3.68	3.62	3.53	3.56	3.47	3.41	4.30	4.20	4.09	3.75	3.59	3.55
+5	6.46	6.25	6.04	6.44	6.19	6.02	9.46	8.92	8.59	7.62	7.23	6.98
0	15.18	14.35	13.72	15.46	14.79	14.16	26.17	24.52	23.83	21.02	19.70	19.14
-5	37.69	36.19	34.94	39.30	37.78	36.77	59.76	57.99	57.48	52.15	50.59	49.74
-10	72.23	71.01	70.21	73.99	72.81	72.08	88.50	87.79	87.55	84.07	83.31	83.19
<i>Seen rooms</i>												
+10	3.74	3.50	3.48	3.51	3.36	3.38	4.36	4.05	4.05	3.74	3.50	3.46
+5	6.40	5.90	5.83	6.36	5.91	5.80	9.51	8.61	8.49	7.63	6.92	6.89
0	14.94	13.69	13.38	15.31	14.19	13.90	26.75	24.77	24.32	21.71	19.86	19.64
-5	37.97	35.88	35.19	39.73	37.61	36.84	61.29	59.00	58.84	54.24	51.79	51.58
-10	72.64	71.14	70.60	74.36	72.91	72.91	89.41	88.51	88.36	85.66	84.58	84.64

CHOIR. Our method improves over FLAC’s autoencoder on both seen and unseen scenes, with about $2\times$ performance improvement on all 3 metrics.

■**Stage 2.** In Table 1 (bottom), we report our main results (averaged over 5 seeds) on both seen and unseen rooms of AcousticRooms. CHOIR attains the best performance on all 3 metrics and on both splits, outperforming all baselines. On unseen rooms, CHOIR achieves a 9.5% lower T60, a 9.3% lower C50, and an 8.5% lower EDT than FLAC, and a 22.0% lower T60, a 34.8% lower C50, and a 31.2% lower EDT than xRIR. Fast-RIR, conditioned only on the T60 and room size estimated from the references and the depth map, performs worse than the nearest neighbour, indicating the need for better representations of the scene $\mathcal{M}$. CHOIR_{ETC} further improves on CHOIR, especially on T60, by 12.6% and 5.5% on unseen and seen cases, respectively, attributed to e_ω ’s energy decay estimation. In Fig. 3, we show the performance on three unseen scenes; CHOIR estimates the overall waveform, including early reflections closer to GT than the baseline. More qualitative results, including Stage 1, are available on our project website <https://choirflow.github.io/>. Finally, on a single RTX 3090, CHOIR produces an RIR in 15.3 ms, timed over a batch of 16; the small MLP e_ω adds overhead of 0.5 ms.

■**Real-world rooms.** To show sim2real transfer, we fine-tune CHOIR on Hearing Anything Anywhere [7]. The setup is slightly mirrored relative to training: the reference RIRs share a common source rather than a receiver, so we capture the depth map at the source. For each of the 4 rooms, we randomly select a subset of the measured RIRs for fine-tuning, and adapt the model from its pre-trained checkpoint on AcousticRooms. We do not retrain the VAE in Stage 1 and evaluate on the remaining held-out test set. To show a fair comparison, we repeat the fine-tuning for the baseline [6]. At a 12 RIRs-per-room budget, CHOIR leads FLAC on C50 and EDT but not on T60, and interestingly, neither reaches the performance of nearest neighbour. We therefore vary the number of RIRs used for fine-tuning, giving all models the same data, and report this in Fig. 2. As the budget increases, CHOIR improves on all 3 metrics, and at 36 RIRs it outperforms both FLAC and the nearest neighbour on all metrics, with a 11.0% lower T60 and a 21.6% lower EDT than FLAC. Beyond this budget, the baseline saturates and degrades on C50 and EDT while CHOIR keeps improving, generalizing well with increasing data. We also fine-tune e_ω on the same responses used to adapt CHOIR to each room. CHOIR_{ETC} improves over

Fig. 3: Qualitative results on 3 unseen rooms on AcousticRooms.



CHOIR on the real rooms as well, most clearly on the early decay, where it gains 3% to 7% across various budgets.

■**Speech recognition.** The above results show that the CHOIR-generated RIRs are closer to the ground truth than the baselines. But since RIRs are generally used by downstream tasks possibly involving noisy environments, matching the ground truth may not by itself show that one can be used in place of the other. We therefore evaluate CHOIR on one such task, automatic speech recognition (ASR), and describe the setup briefly. We convolve each RIR in both test splits with 4 utterances from LibriSpeech [32] and transcribe them with 2 ASR models, Parakeet [33] and Qwen3-ASR [34]. To also test robustness, we add pink noise and multi-talker babble at 5 SNR levels, using the same noise excerpt for the ground truth and prediction; we repeat this process for FLAC as well. We report the word error rate (WER) computed with jiwer [35], once for each set of RIRs: ground truth, CHOIR, and FLAC. Table 2 reports the results with around 25,000 paired utterances per cell. First, we observe that both models have better WER than the ground truth, with the baseline having the lowest among the three. This is likely because learned models generate smoother responses than a simulator, and the ASR models are biased towards them since they are also learned. But since our goal is to imitate the simulator, we seek the model that is closer to the ground truth WER, not the lowest one; CHOIR is closer to the ground truth than FLAC in almost all of the cases. This is consistent across the SNR levels, both noise types and both ASR models, with CHOIR on average 24% closer to the ground truth WER than FLAC. Audio samples are available at <https://choirflow.github.io/>.

5. CONCLUSION

We propose CHOIR, a flow matching model that synthesizes RIRs at a source and receiver pair in a scene, given a few reference RIRs measured in it. CHOIR outperforms all baselines across metrics on AcousticRooms and on real rooms. In the ASR task, the word error rate from the RIRs generated using our method stays closest to the ground truth across noise types, SNR levels, and models. Future work includes extensions to multi-channel impulse responses (such as higher-order ambisonics), physics-consistent models, and eliminating the need for few-shot references. In sum, we believe generative models can serve as surrogates for acoustic simulators, producing faster and cheaper high-fidelity impulse responses; CHOIR is a step in that direction.

6. REFERENCES

- [1] H Kuttruff, *Room Acoustics*, Elsevier, Applied Science, 1973.
- [2] Treble Technologies, “Treble technologies,” <https://www.treble.tech/>, Accessed: 2026-07-04.
- [3] COMSOL, “Acoustics module,” <https://www.comsol.com/acoustics-module>, Accessed: 2026-07-27.
- [4] Zhenyu Tang, Rohith Aralikatti, Anton Jeran Ratnarajah, and Dinesh Manocha, “Gwa: A large high-quality acoustic dataset for audio processing,” in *SIGGRAPH*, 2022.
- [5] Xiulong Liu, Anurag Kumar, Paul Calamia, Sebastià V. Amengual Garí, Calvin Murdock, Ishwarya Ananthabhotla, Philip Robinson, Eli Shlizerman, Vamsi Krishna Ithapu, and Ruohan Gao, “Hearing anywhere in any environment,” in *CVPR*, 2025.
- [6] Amandine Brunetto, “Few-shot acoustic synthesis with multi-modal flow matching,” in *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2026.
- [7] Mason Long Wang, Ryosuke Sawata, Samuel Clarke, Ruohan Gao, Shangzhe Wu, and Jiajun Wu, “Hearing anything anywhere,” in *CVPR*, 2024.
- [8] 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,” *CACM*, 2021.
- [9] Kai Zhang, Gernot Riegler, Noah Snavely, and Vladlen Koltun, “Nerf++: Analyzing and improving neural radiance fields,” *arXiv preprint arXiv:2010.07492*, 2020.
- [10] Ricardo Martin-Brualla, Noha Radwan, Mehdi SM Sajjadi, Jonathan T Barron, Alexey Dosovitskiy, and Daniel Duckworth, “Nerf in the wild: Neural radiance fields for unconstrained photo collections,” in *CVPR*, 2021.
- [11] Andrew Luo, Yilun Du, Michael Tarr, Josh Tenenbaum, Antonio Torralba, and Chuhan Gan, “Learning neural acoustic fields,” in *NeurIPS*, 2022.
- [12] Susan Liang, Chao Huang, Yapeng Tian, Anurag Kumar, and Chenliang Xu, “Neural acoustic context field: Rendering realistic room impulse response with neural fields,” *ICCVW*, 2023.
- [13] Chen Si, Qianyi Wu, Chaitanya Amballa, and Romit Roy Choudhury, “Explicit context-driven neural acoustic modeling for high-fidelity rir generation,” *arXiv preprint arXiv:2509.15210*, 2025.
- [14] Kun Su, Mingfei Chen, and Eli Shlizerman, “Inras: Implicit neural representation for audio scenes,” in *NeurIPS*, 2022.
- [15] Amandine Brunetto, Sascha Hornauer, and Fabien Moutarde, “NeRAF: 3D scene infused neural radiance and acoustic fields,” in *ICLR*, 2025.
- [16] Anton Ratnarajah, Shi-Xiong Zhang, Meng Yu, Zhenyu Tang, Dinesh Manocha, and Dong Yu, “Fast-rir: Fast neural diffuse room impulse response generator,” in *ICASSP*, 2022.
- [17] Anton Ratnarajah, Zhenyu Tang, Rohith Aralikatti, and Dinesh Manocha, “Mesh2ir: Neural acoustic impulse response generator for complex 3d scenes,” in *ACM International Conference on Multimedia*, 2022, pp. 924–933.
- [18] Nikhil Singh, Jeff Mentch, Jerry Ng, Matthew Beveridge, and Iddo Drori, “Image2reverb: Cross-modal reverb impulse response synthesis,” in *ICCV*, 2021.
- [19] Sagnik Majumder, Changan Chen, Ziad Al-Halah, and Kristen Grauman, “Few-shot audio-visual learning of environment acoustics,” *NeurIPS*, 2022.
- [20] Diwei Huang, Kunyang Lin, Peihao Chen, and Qing Du, “Map-guided few-shot audio-visual acoustics modeling,” in *ICASSP*, 2025.
- [21] Prafulla Dhariwal and Alexander Nichol, “Diffusion models beat gans on image synthesis,” *Advances in neural information processing systems*, vol. 34, pp. 8780–8794, 2021.
- [22] François Mazé and Faez Ahmed, “Diffusion models beat gans on topology optimization,” in *Proceedings of the AAAI conference on artificial intelligence*, 2023, vol. 37, pp. 9108–9116.
- [23] Yaron Lipman, Ricky TQ Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le, “Flow matching for generative modeling,” *arXiv preprint arXiv:2210.02747*, 2022.
- [24] Michael S Albergo and Eric Vanden-Eijnden, “Building normalizing flows with stochastic interpolants,” *arXiv preprint arXiv:2209.15571*, 2022.
- [25] Diederik P Kingma and Max Welling, “Auto-encoding variational bayes,” *arXiv preprint arXiv:1312.6114*, 2013.
- [26] Jonathan Ho and Tim Salimans, “Classifier-free diffusion guidance,” *arXiv preprint arXiv:2207.12598*, 2022.
- [27] William Peebles and Saining Xie, “Scalable diffusion models with transformers,” in *2023 IEEE/CVF International Conference on Computer Vision (ICCV)*. IEEE, 2023, pp. 4172–4182.
- [28] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu, “Roformer: Enhanced transformer with rotary position embedding,” *Neurocomputing*, 2024.
- [29] Byeongho Heo, Song Park, Dongyoon Han, and Sangdoo Yun, “Rotary position embedding for vision transformer,” in *European Conference on Computer Vision*. Springer, 2024.
- [30] Oriane Siméoni, Huy V Vo, Maximilian Seitzer, Federico Baldassarre, Maxime Oquab, Cijo Jose, Vasil Khalidov, Marc Szafraniec, Seungeun Yi, Michaël Ramamonjisoa, et al., “Dinov3,” *arXiv preprint arXiv:2508.10104*, 2025.
- [31] Alexander Lindau, Linda Kossanke, and Stefan Weinzierl, “Perceptual evaluation of model-and signal-based predictors of the mixing time in binaural room impulse responses,” *Journal of the Audio Engineering Society*, vol. 60, no. 11, pp. 887–898, 2012.
- [32] Vassil Panayotov, Guoguo Chen, Daniel Povey, and Sanjeev Khudanpur, “Librispeech: an asr corpus based on public domain audio books,” in *2015 IEEE international conference on acoustics, speech and signal processing (ICASSP)*. IEEE, 2015, pp. 5206–5210.
- [33] Monica Sekoyan, Nithin Rao Koluguri, Nune Tadevosyan, Piotr Zelasko, Travis Bartley, Nikolay Karpov, Jagadeesh Balam, and Boris Ginsburg, “Canary-1b-v2 & parakeet-tdt-0.6 b-v3: Efficient and high-performance models for multilingual asr and ast,” *arXiv preprint arXiv:2509.14128*, 2025.
- [34] Xian Shi, Xiong Wang, Zhifang Guo, Yongqi Wang, Pei Zhang, Xinyu Zhang, Zishan Guo, Hongkun Hao, Yu Xi, Baosong Yang, et al., “Qwen3-asr technical report,” *arXiv preprint arXiv:2601.21337*, 2026.
- [35] Nik Vaessen, “Jiwer: Similarity measures for automatic speech recognition evaluation,” <https://github.com/jitsi/jiwer>, 2018.