WavNAF: Learning Wave Propagation Priors for Neural Acoustic Fields

Taeho Kim, Hyunjun Kim, MinKyu Lee, SuBeen Lee, Inkyu An*, Jae-Pil Heo*

NeurIPS 2026

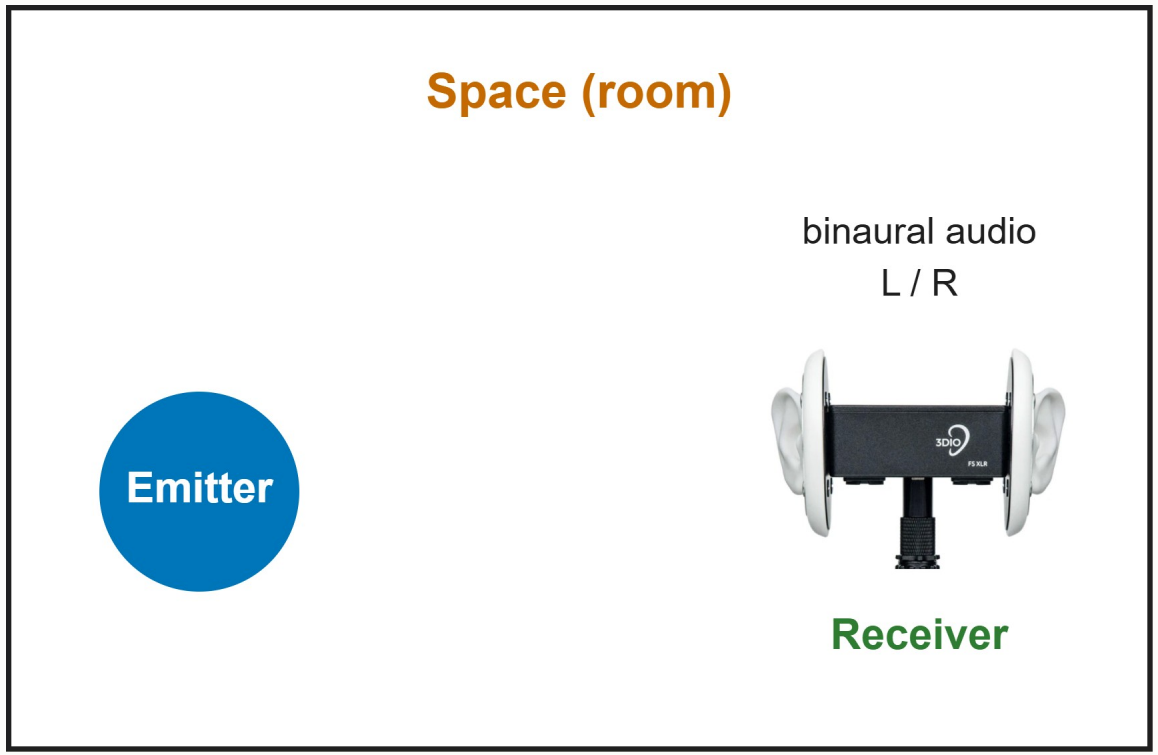
* Co-corresponding authors

Contents

1. Task Definition
2. Related Works
3. Motivation
4. Method
5. Experiments

Task Definition

What is spatial audio?



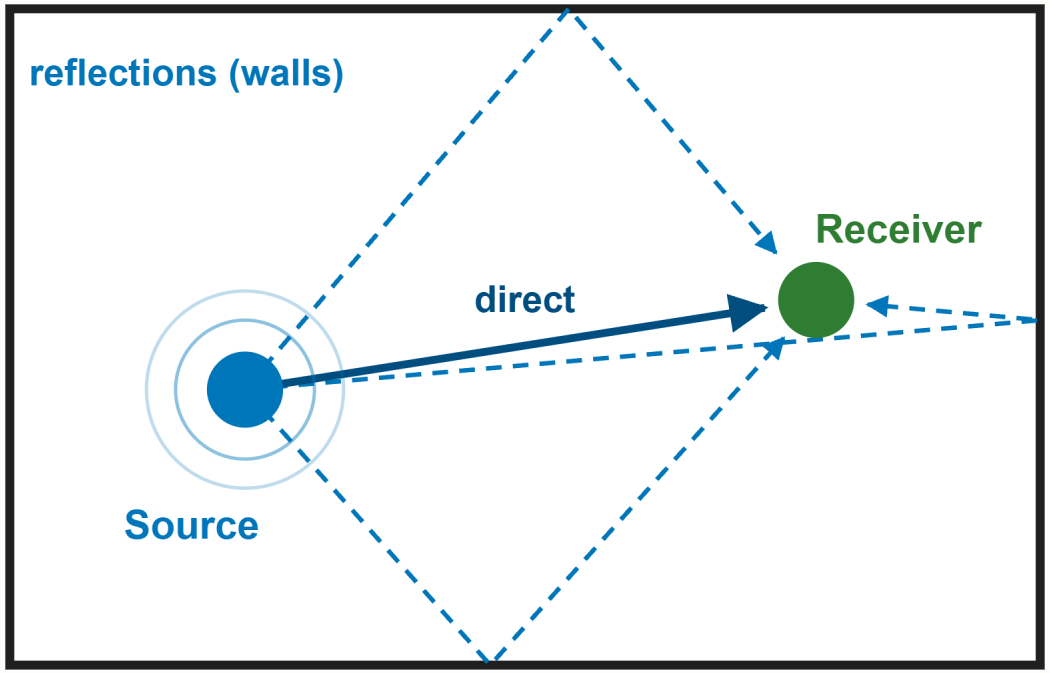
Given an emitter, a receiver and a space,

synthesize the sound heard at the receiver, wherever it is placed in the room.

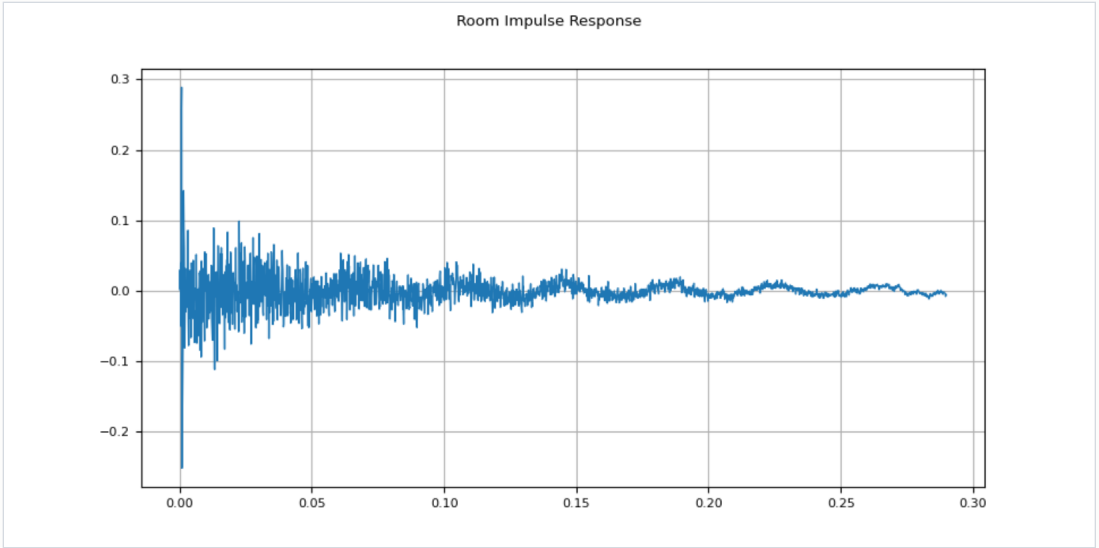
Binaural audio (left / right ear) carries direction, distance and the “feel” of the room. Used in VR/AR, games and embodied agents.

Task Definition

Room Impulse Response (RIR)



Measured RIR

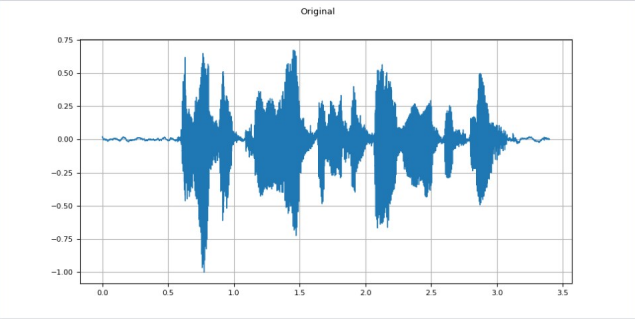


Fire a short impulse at the source, record at the receiver → **Room Impulse Response (RIR)**

One RIR per (source, receiver) pair.

Task Definition

From RIR to spatial audio

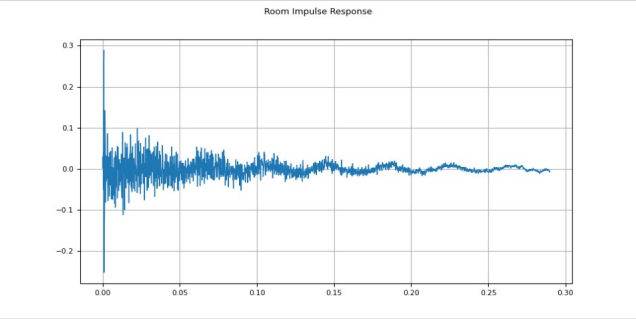


Anechoic sound

recorded once in a dry studio

*

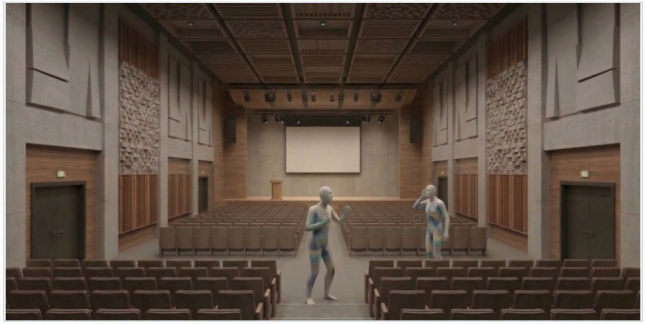
convolution



RIR

of the target location

=



Spatial audio

as heard at that location

y(t) = (x ∗ h)(t) (∗ = convolution) —any dry sound x becomes spatial audio once we know the RIR h at that location.

So instead of predicting audio directly, we predict the RIR for any source / receiver position.

Task Definition

Neural RIR estimation: same implicit network as NeRF, different output

NeRF (vision)

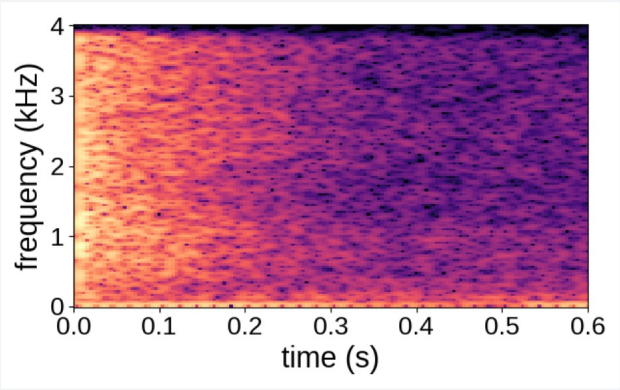
position (x, y, z)
view dir. (θ, φ)
→ **MLP** →
color c
density σ



volume rendering → **image** from any viewpoint

Neural acoustic field (audio)

source pose
receiver pose
time t
→ **MLP** →
RIR spectrum at time t

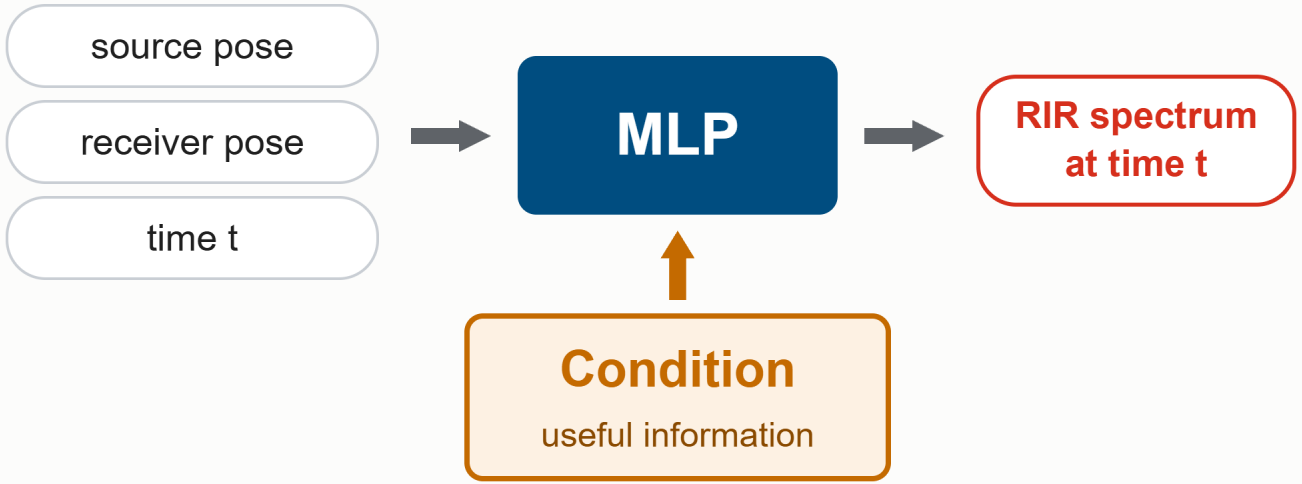


all t → **RIR spectrogram** for any source / receiver

Related Works

Typical framework of neural RIR estimation

Typical neural RIR estimation framework: an implicit neural network, plus an additional **feature condition** that carries useful information.



Neural RIR estimation models have evolved by **enriching this condition**.

Related Works

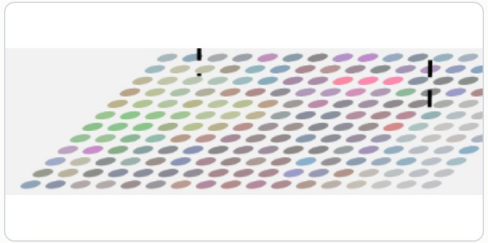
Development of neural RIR estimation

What has been used as the **condition**?

NAF
Luo et al., NeurIPS 2022

Learned feature grid

local features learned jointly with the network



AV-NeRF
Liang et al., NeurIPS 2023

Rendered image

RGB-D rendered by NeRF at the receiver view



NeRAF
Brunetto et al., ICLR 2025

Room voxel grid

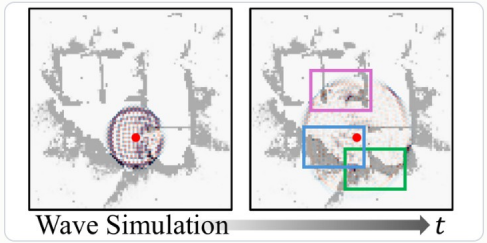
NeRF density & color of the whole room



WavNAF (ours)
NeurIPS 2026

Wave simulation

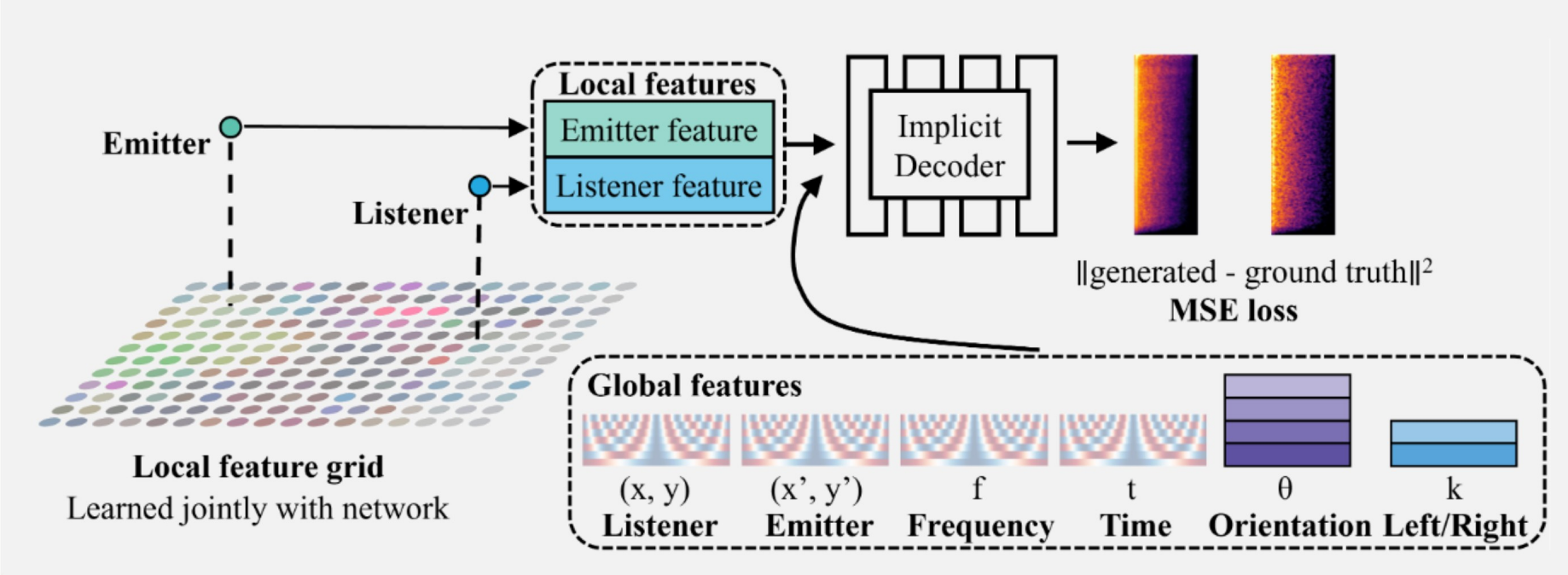
how a wave from this source propagates in the room



richer, more useful condition

Related Works

Implicit neural representation: Neural Acoustic Fields (NAF)



Condition = learned feature grid

Emitter / listener positions index a local feature grid learned jointly with the network.

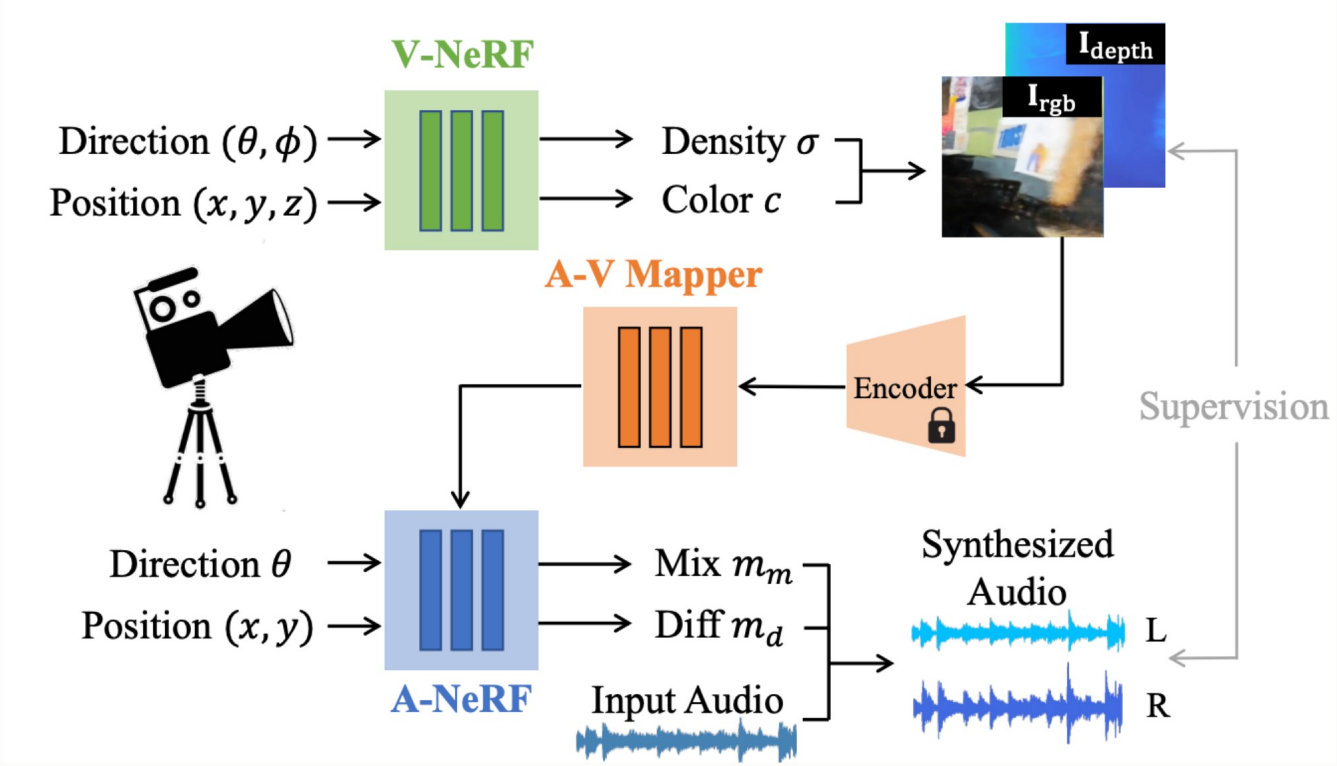
Limitation

No explicit scene geometry: walls and obstacles are only learned implicitly from data.

[1] A. Luo et al., "Learning neural acoustic fields," NeurIPS 2022.

Related Works

Image-guided acoustic modeling: AV-NeRF



Condition = rendered image

A NeRF renders RGB-D at the receiver pose; an encoder turns it into the audio condition.

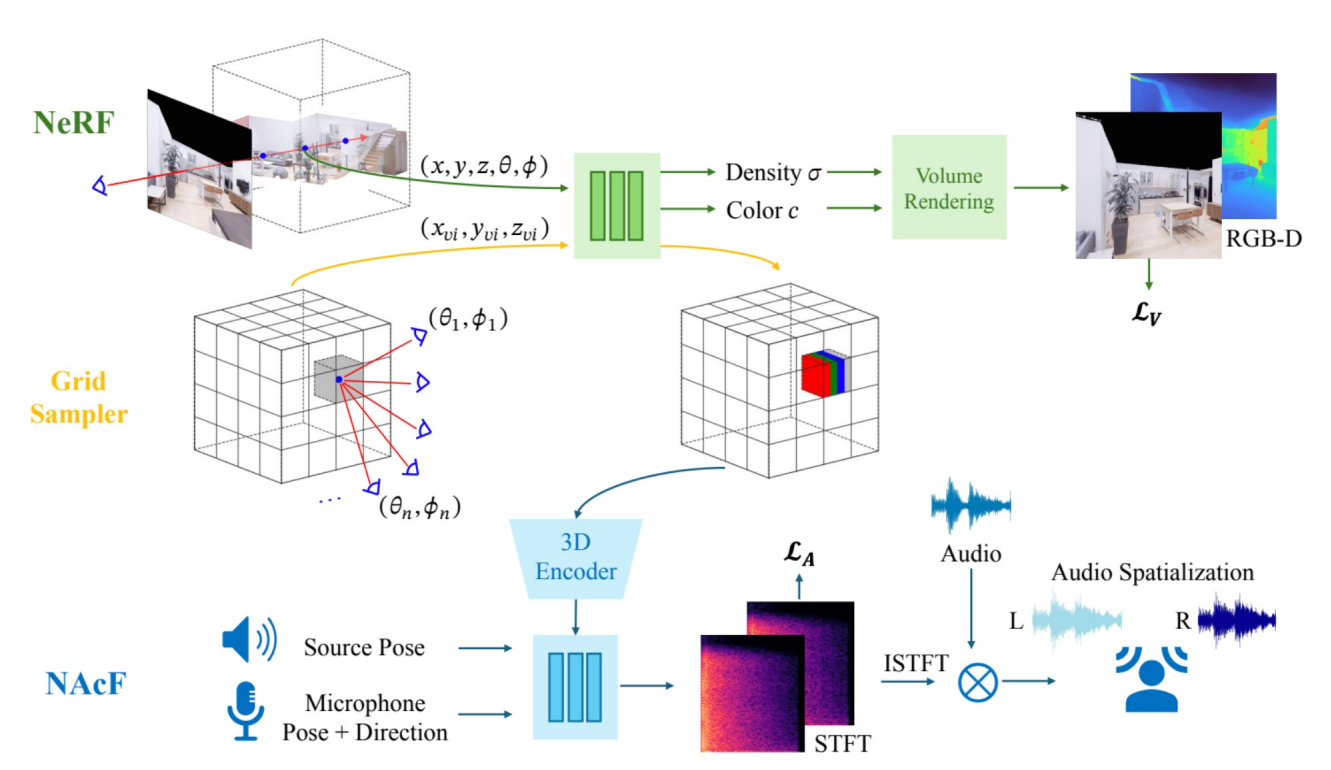
Limitation

Only what is visible from that viewpoint: acoustic interactions outside the rendered view are missed.

[2] S. Liang et al., “AV-NeRF: Learning neural fields for real-world audio-visual scene synthesis,” NeurIPS 2023.

Related Works

Explicit spatial acoustic representation: NeRAF



Condition = room voxel grid

NeRF density & color are sampled into a 3D grid and a 3D ResNet encodes the whole room into one feature.

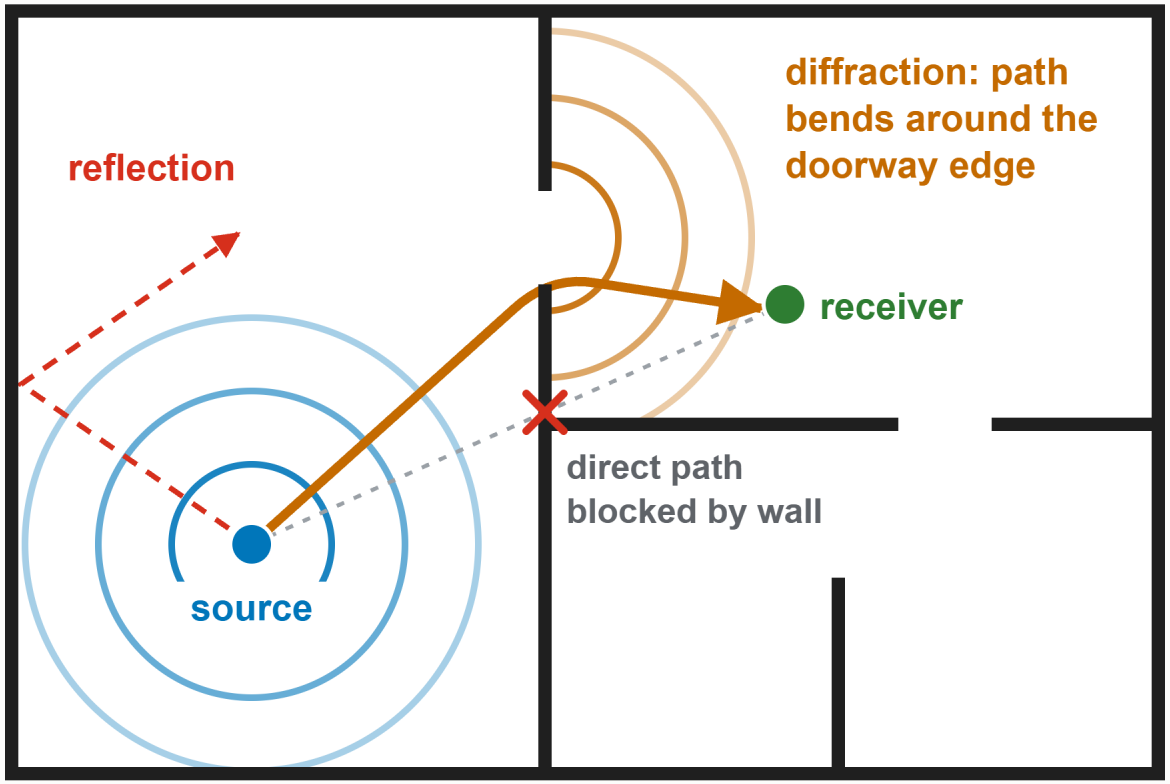
Limitation

Static scene geometry: no dynamic sound–geometry interaction.

[3] A. Brunetto, S. Hornauer, F. Moutarde, “NeRAF: 3D scene infused neural radiance and acoustic fields,” ICLR 2025.

Motivation

What static geometry misses



Sound is a **wave**: it reflects, diffuses, refracts, **diffracts around doorways** and interferes.

A voxel grid tells *where* the walls are — not *how* a wave from this source travels around them.

Static geometry alone cannot reflect how the wave interacts with the geometry.

Method

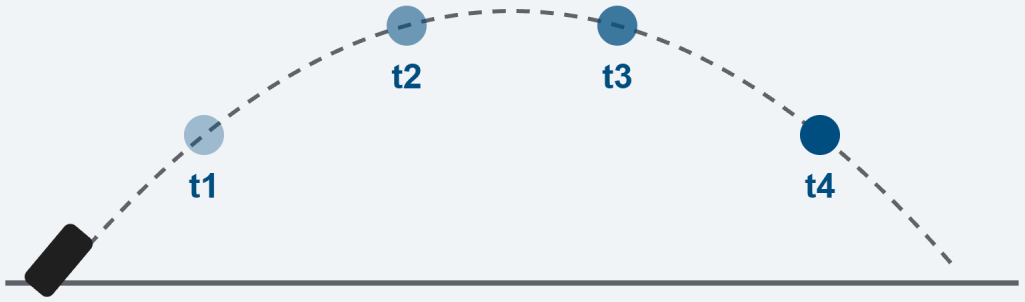
Modeling wave–geometry interaction: solve the wave equation

Then how can we model the interaction between the wave and the geometry?

→ **Solve the wave equation**, the **equation of motion** for sound.

Cannonball (projectile motion)

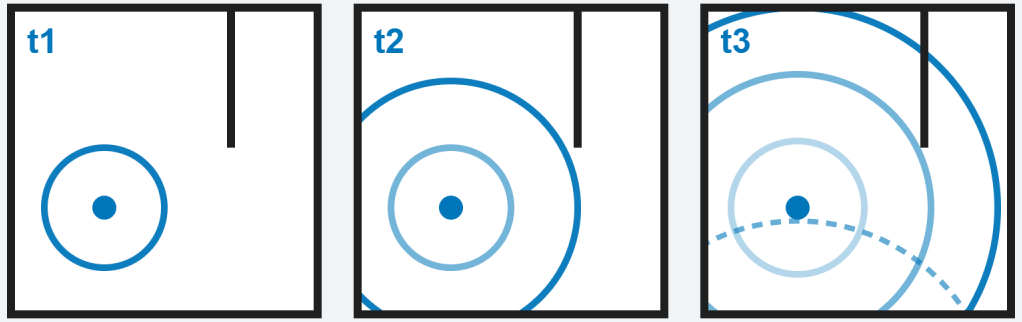
$$d^2y/dt^2 = -g,\ d^2x/dt^2 = 0$$



solve → **position** at any time t

Sound (acoustic wave equation)

$$\partial^2 p/\partial t^2 = c^2\ \nabla^2 p + f(X,\ t)$$



solve → **pressure field p(X, t)** at any time t

Solving the equation of motion tells us the state of the system at any time.

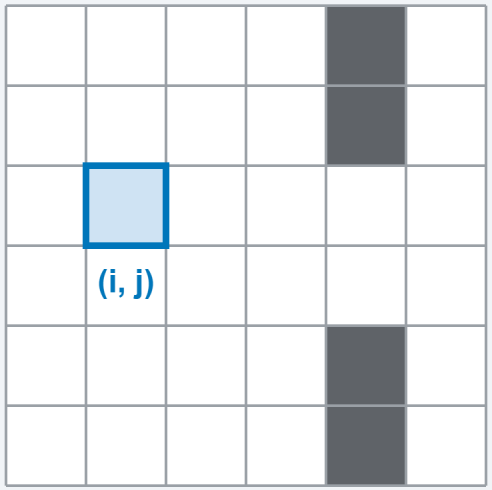
[4] D. Botteldooren, “Finite-difference time-domain simulation of low-frequency room acoustic problems,” JASA 1995.

Method

How FDTD solves the wave equation

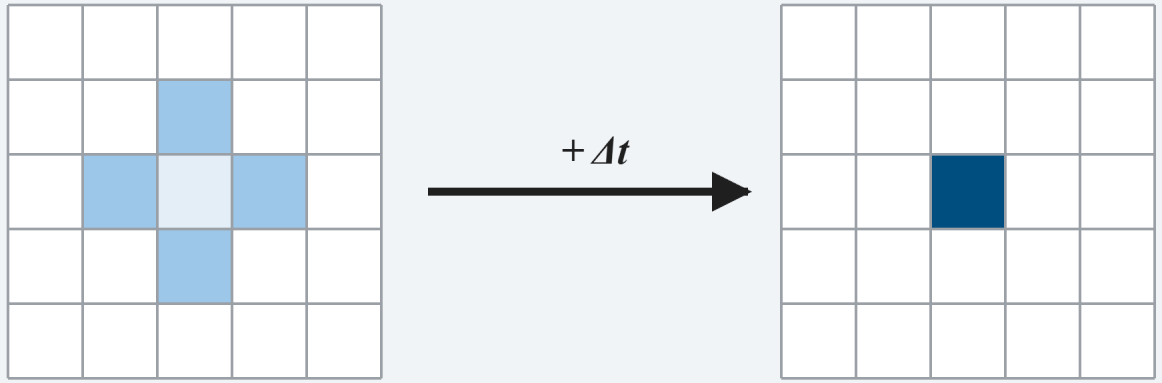
FDTD (Finite-Difference Time-Domain): discretize space and time, then update **step by step**.

Space: a grid of cells



cells of Δx × Δy, each with its own ρ, c (gray = wall)

Time: update each cell from its neighbors



step n **step n + 1**

new value at (i, j) ← current values of (i, j) and its 4 neighbors

Repeat for n = 1, 2, 3, … → the pressure field at every time step.

[4] D. Botteldooren, “Finite-difference time-domain simulation of low-frequency room acoustic problems,” JASA 1995.

Method

The update rule is the wave equation in discrete form

Each FDTD update is a **discretized wave equation** — so **the wave physics is inside every step**.

Wave equation

$$\partial^2 p/\partial t^2 = c^2 \ \nabla^2 p$$

continuous in space and time

finite differences →

Discrete update rule (per cell, per step)

$$\nabla p \approx (p[i+1, j] - p[i, j]) / \Delta x$$

$$v \leftarrow v - (\Delta t / \rho) \ \nabla p$$

$$p \leftarrow p - \rho \, c^2 \, \Delta t \, (\nabla \cdot v)$$

local: each cell reads only its neighbors

repeat every Δt →

Wave behavior emerges

- propagation
- reflection
- diffraction
- interference

No hand-crafted model of each effect: the update process itself carries the wave characteristics.

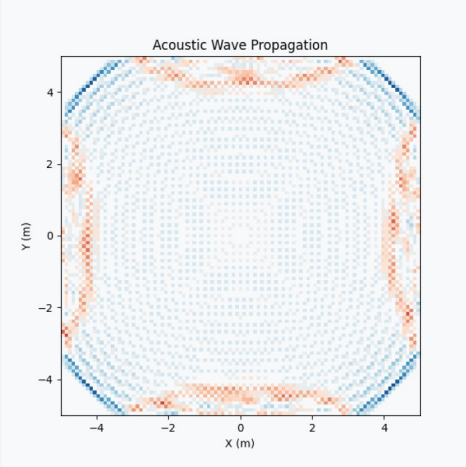
[4] D. Botteldooren, “Finite-difference time-domain simulation of low-frequency room acoustic problems,” JASA 1995.

Method

Wave behavior emerges from the update rule: examples

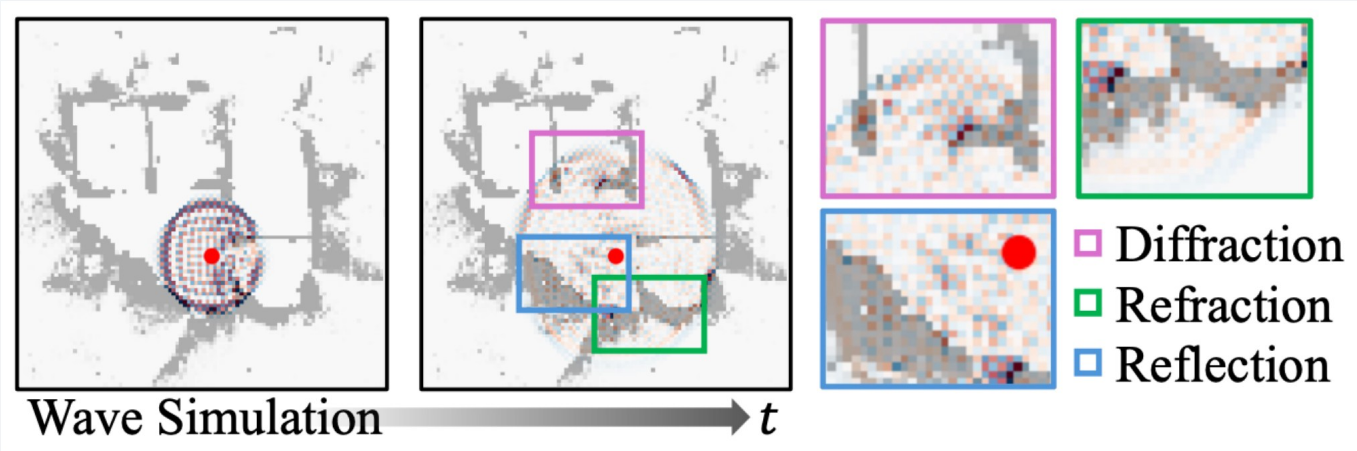
Run the same local update step by step → **wave behavior emerges** on its own.

simple room



propagation | reflection

real floor plan



diffraction | refraction | reflection

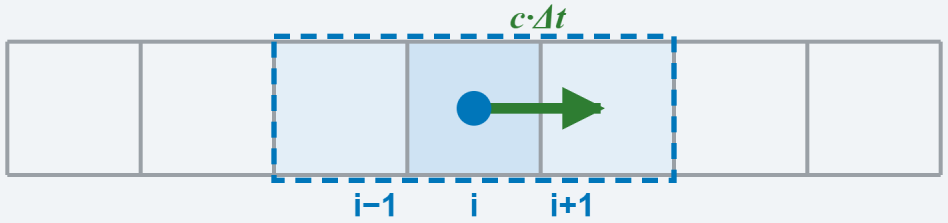
No separate model for each effect — they all come from the same wave-equation update.

Method

Why not just run FDTD?

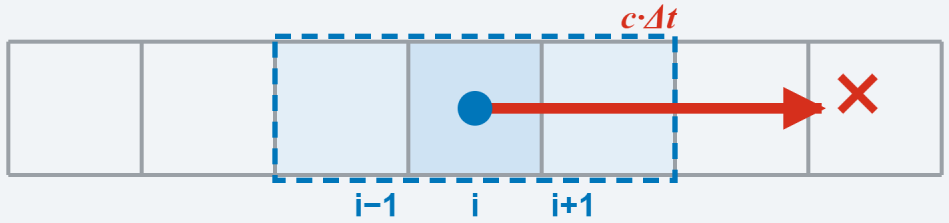
Each update only reads its **neighbors** — so in one step Δt, the wave may travel **at most one cell**.

OK: c·Δt ≤ Δx → stable



the wave stays within the neighbors the update reads

Violated: c·Δt > Δx → unstable



the wave skips past the neighbors → the update misses it, errors blow up

$$\text{2D CFL condition: } \Delta t \leq \Delta x\, \Delta y \,/\, (c \sqrt{(\Delta x^2 + \Delta y^2)})$$

Fine grid ⇒ tiny Δt ⇒ a full RIR needs **~25× more steps** than our probe (Apartment 1) — and a new simulation **for every source**.

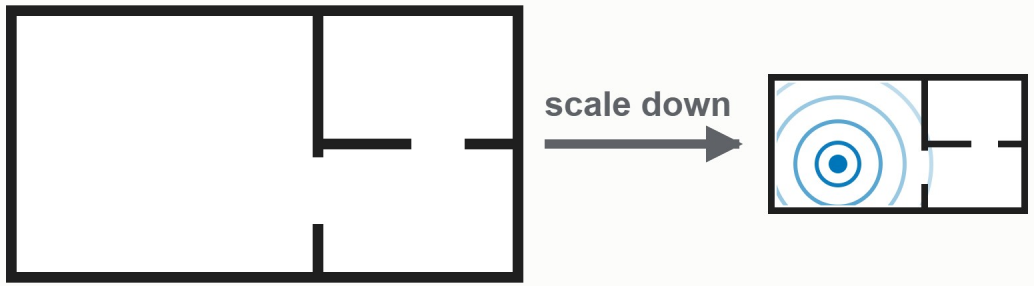
Running full-scale FDTD for every source is too expensive.

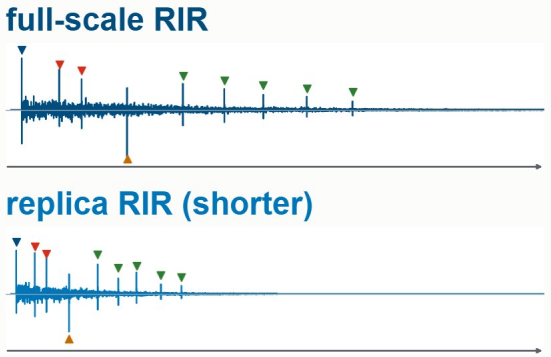
[5] R. Courant, K. Friedrichs, H. Lewy, Math. Annalen 1928.

Method

Inspiration: acoustic scale models

An **acoustic scale model**: a 1:10–1:50 miniature of a space, used to study its acoustics.
→ We borrow the idea: probe a small **replica** instead of the full-scale room.







How a 1:n model works

$$p\lambda(X, t) = p(X/\lambda, t/\lambda)$$

sizes ÷ n → test frequencies × n;
measured times × n give the full-scale RIR

Not a perfect copy

Geometry is preserved, but air absorption (too high at scaled frequencies) and materials are only approximated → **correction needed**, mostly for the late tail.

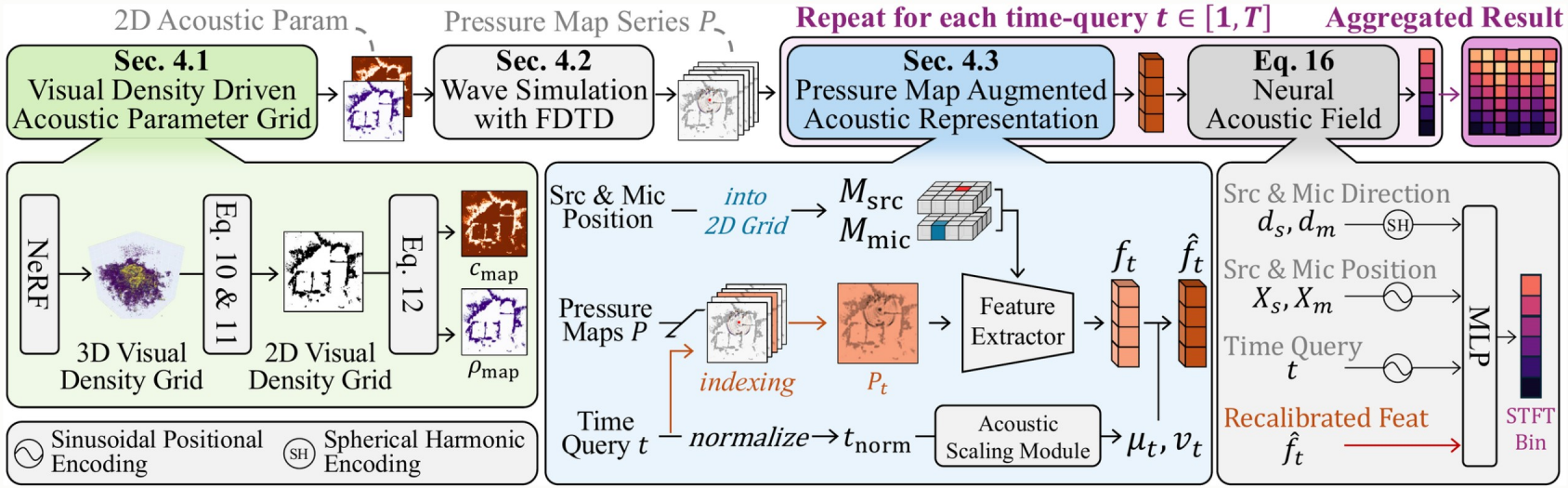
Our idea: a digital scale model — a short, scaled FDTD probe + a learned correction.

[6] M. Barron, IoA 2002. [7] J. H. Rindel, Building Acoustics 2011. [8] K. Suzuki & T. Hidaka, ISRA 2019.

Method

WavNAF overview: a digital acoustic scale model

WavNAF = the **digital scale model** idea, built into a neural acoustic field.



① Digital replica

NeRF → 2D acoustic parameter grid of the room

② Short FDTD probe

pressure maps → feature encoder → wave feature

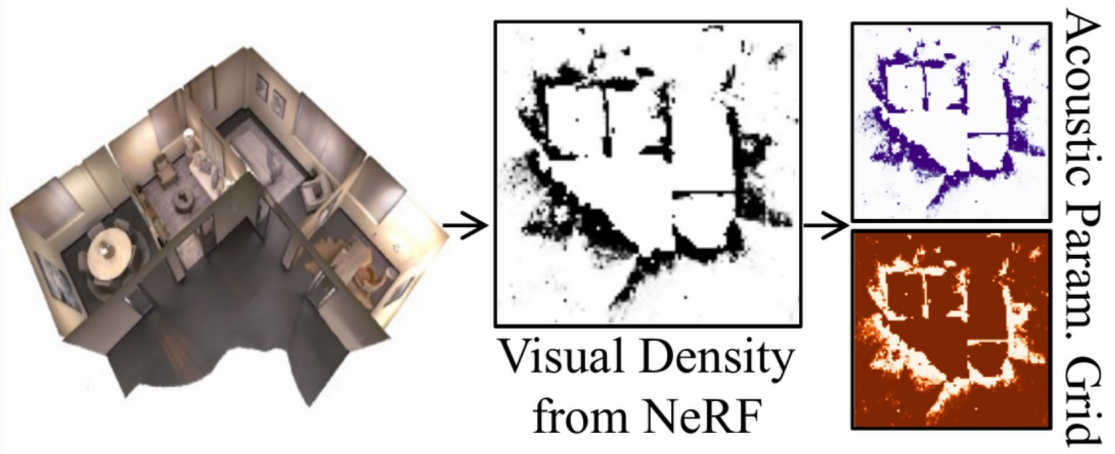
③ Learned correction

Neural Acoustic Scaling Module → condition → MLP → RIR

Method

① Digital replica from NeRF

FDTD needs a sound speed *c* and density *ρ* in every cell — never measured, so we **derive them from the NeRF**.



(a) NeRF density → α

Query NeRF density σ on an N^3 grid; $\alpha = 1 - \exp(-\sigma\delta)$.

(b) 2D slice at mic height

Max-pool α over a vertical window ($\Delta k_z = 2$) around the receiver height.

(c) α → acoustic parameters

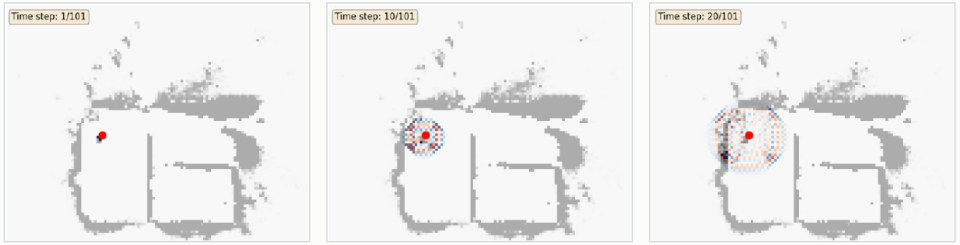
$c = c_{min} + (c_{max} - c_{min})(1 - \alpha)$, $\rho = \rho_{air}(1 + \alpha)$

Approximate by design (like a scale model): all c_{min}/c_{max} ratios 0.6–0.9 beat the no-wave baseline.

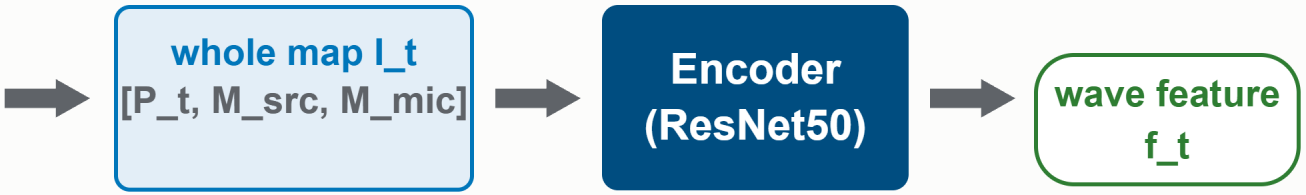
Method

② Short FDTD probe: encode the whole pressure map

We do **not** read pressure at individual points — the **whole pressure map** goes into the encoder, so it learns the **propagation pattern** itself.



pressure maps P over time (first steps shown)*



Not: a point readout

p at one location (e.g., the receiver) — misses how the wave moves through the room

Ours: the whole map → encoder

learns the propagation pattern: wavefront shape, occlusion, diffraction-like spreading

* Gray walls are drawn for visualization only; the encoder input contains no geometry.

Method

③ Learned correction: Neural Acoustic Scaling Module

Like a physical scale model, the short probe on the replica is **shorter and slightly different** from full scale
→ **learn the correction**, then use the corrected wave feature as the **condition**.

wave feature f_t → **Neural Acoustic Scaling Module**

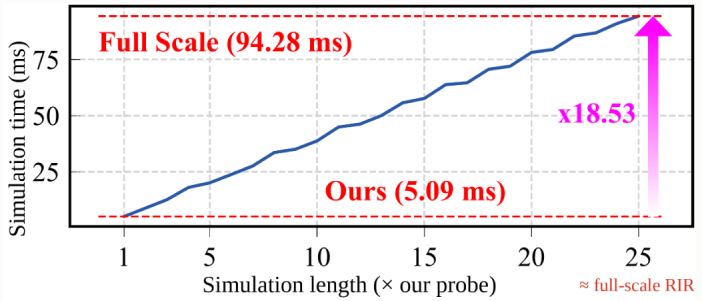
$$\hat{f}_t = \mu_t \odot f_t + v_t$$

μ_t, v_t = MLP(t_norm): time-conditioned (FiLM)

→ **Condition** $\hat{f}_t$ → **MLP** (+ positions, directions, t) → **RIR spectrum at time t**

Why time-conditioned?

Early reflections depend mainly on room geometry, which the replica preserves. The late tail accumulates absorption and air-loss mismatch, so **each time bin needs its own correction**.



simulation cost: ours 5.09 ms vs. full scale 94.28 ms (×18.53)

Experiments

Main results: does the wave condition help?

Same neural acoustic field, different **condition** — tested on synthetic and real scenes.

SoundSpaces (synthetic) — gray rows: GT mesh

Method · condition	GT	T60 ↓	C50 ↓	EDT ↓
Opus-nearest		10.10	3.58	0.115
Opus-linear		8.64	3.13	0.097
AAC-nearest		9.35	1.67	0.059
AAC-linear		7.88	1.68	0.057
INRAS	✓	3.14	0.60	0.019
NACF	✓	2.36	0.50	0.014
NACF w/ T	✓	2.17	0.49	0.014
NAF · learned feature grid		3.18	1.06	0.031
AV-NeRF · image		2.47	0.57	0.016
NeRAF · voxel grid		2.14	0.38	0.010
WavNAF · wave		1.95	0.33	0.009

Best on all metrics , even vs. GT-mesh methods

RAF (real)

T60 ↓	C50 ↓	EDT ↓	STFT ↓
10.03	0.76	0.021	0.49
10.19	0.86	0.029	0.92
22.83	1.97	0.064	1.04
25.64	2.49	0.085	1.26
8.01	0.79	0.025	0.36
6.62	0.59	0.017	0.39
7.31	0.59	0.018	0.39
10.08	0.71	0.021	0.64
8.11	0.73	0.021	0.39
7.47	0.61	0.020	0.17
7.22	0.59	0.018	0.16

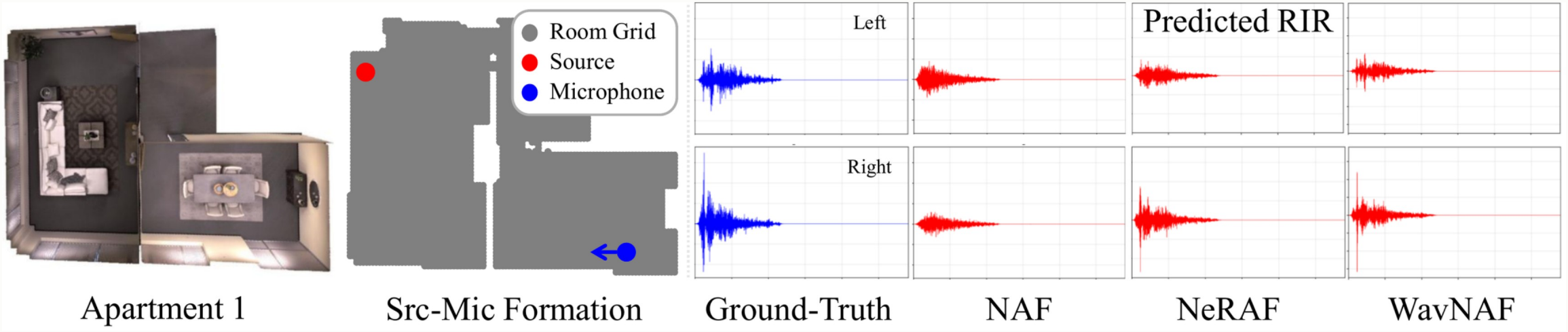
Best without a GT mesh on all metrics

T60: late reverberation · **C50**: clarity · **EDT**: early decay (early reflections) · **STFT**: spectrogram — errors, lower is better

Experiments

Qualitative results: the doorway case (SoundSpaces, Apartment 1)

The case static geometry struggles with: sound reaching **another room through a doorway**.



Source and receiver in different rooms

The sound must **diffract through doorways** and reflect several times before it arrives.

Wave condition keeps the early spikes

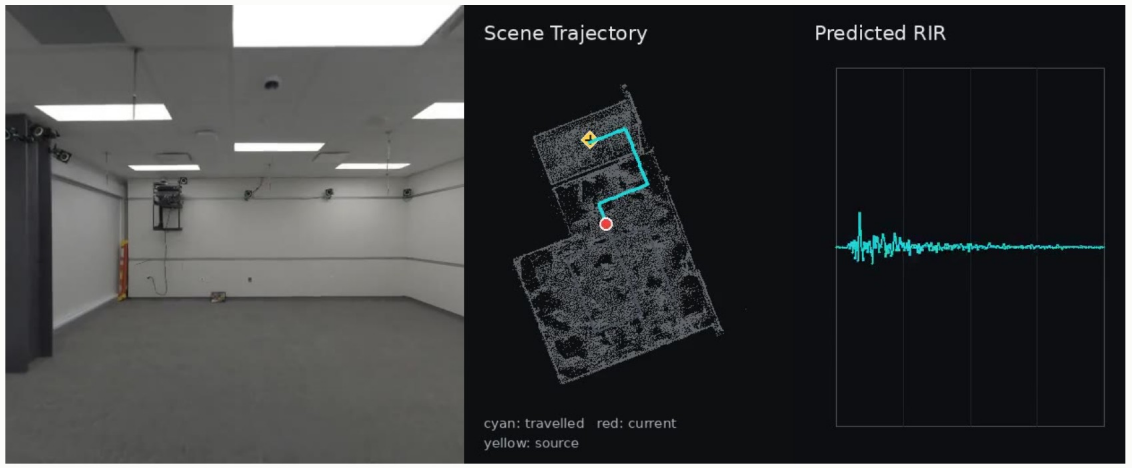
NAF (learned grid) and NeRAF (voxel grid) blur them; WavNAF follows the GT spike pattern.

Experiments

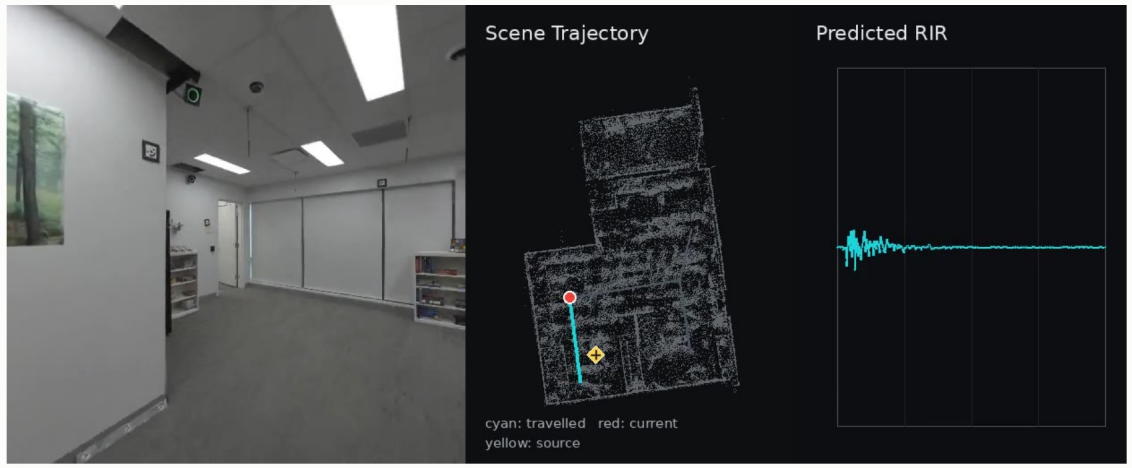
Demo: RAF Empty Room vs. Furnished Room (music)

Real scenes (RAF), music played through the RIR predicted by WavNAF along a walkthrough.

Empty Room



Furnished Room



Watch the demo videos with sound

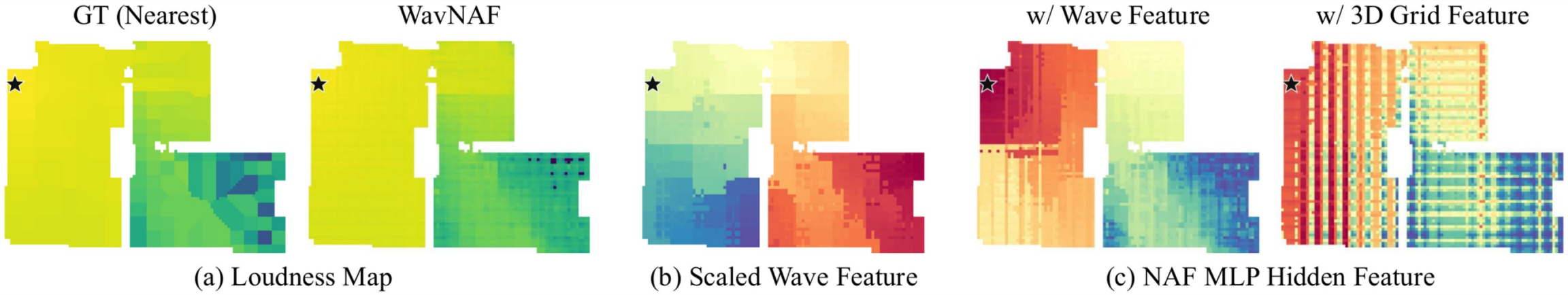
kth0522.github.io/WavNAF_project_page

Left: camera view · middle: trajectory (yellow = source) · right: RIR predicted by WavNAF.

Experiments

What does the network learn? Inside the neural acoustic field

Why does it work? Look inside: the **condition** shapes what the neural acoustic field learns.



(a) Loudness map

WavNAF follows the spatial loudness pattern of the GT (★ = source).

(b) Condition $\hat{f}_t$

The corrected wave feature varies smoothly across the room.

(c) Hidden features of the field

Wave condition → **diffraction-like** patterns like (a); voxel-grid condition → grid stripes.

Wave in → wave-like representation inside the network.

Colors in (b, c) from independent 1D t-SNE projections; not comparable across panels.

Experiments

Ablation: is it the wave, or just geometry?

Is it the wave, or just geometry? Same neural acoustic field — **only the condition changes**.

SoundSpaces, Apartment 1

Condition	T60 ↓	C50 ↓	EDT ↓
no condition	2.776	0.590	0.0151
voxel grid (NeRAF-style)	2.725	0.574	0.0149
wave feature	2.406	0.467	0.0122
wave + Scaling Module (ours)	2.306	0.446	0.0119

Static voxel grid barely helps → the **wave condition** gives the big jump → the learned correction adds more.

Data-efficient (RAF few-shot)

With 50% of the data, WavNAF's EDT (0.0206) beats NeRAF trained on 75% (0.0210).

Short probe is enough

Simulating 2–3× longer per frame gives no further gain: the short probe already carries the useful wave cues.

Conclusion

Summary

Neural RIR estimation advances by enriching the **condition** — WavNAF conditions it on **wave propagation**.

Summary

Problem	Static geometry cannot capture how the wave interacts with the room.
Idea	Solve the wave equation on a **digital scale model**: NeRF replica + short FDTD probe.
Method	Encode the whole pressure map; the **Neural Acoustic Scaling Module** corrects it into the condition.
Result	Best on SoundSpaces and RAF (w/o GT mesh); learned features become **wave-like**.

Limitations & future work

2D probe

No vertical propagation or floor / ceiling reflections → efficient 3D or multi-plane probing.

Approximate materials

Frequency-independent parameters from NeRF density → material-aware, frequency-dependent parameters.