Mean Absorption Coefficient Estimation from Impulse Responses

Deep Learning vs. Sabine

Corto Bastien\textsuperscript{a} \hspace{1cm} Antoine Deleforge\textsuperscript{a} \hspace{1cm} Cédric Foy\textsuperscript{b}

\textsuperscript{a}Inria Nancy – Grand Est
\textsuperscript{b}Cerema, Ifsttar, UMRAE, Strasbourg
OUTLINE

1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
1. Introduction

Noise disturbance

• Public health concern
• Affects speech intelligibility
• Reduces concentration
Noise disturbance

- Public health concern
- Affects speech intelligibility
- Reduces concentration

Renovation of existing buildings
1. Introduction

Noise disturbance

- Public health concern
- Affects speech intelligibility
- Reduces concentration

Renovation of existing buildings

Wall transmission
1. Introduction

Noise disturbance

- Public health concern
- Affects speech intelligibility
- Reduces concentration

Renovation of existing buildings

{Wall transmission
  Wall absorption}
1. Introduction

Noise disturbance

- Public health concern
- Affects speech intelligibility
- Reduces concentration

Renovation of existing buildings

- First, a **diagnosis** is required:

Absorption coefficients estimation
1. Introduction

Conventional approach

- Estimating the mean absorption coefficients in octave bands:

\[
\bar{\alpha}(b) \overset{\text{def}}{=} \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b) S_i \quad (b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz})
\]
1. Introduction

Conventional approach

• Estimating the mean absorption coefficients in octave bands:

\[ \bar{\alpha}(b) \overset{\text{def}}{=} \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b)S_i \quad (b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz}) \]

Absorption coefficient in [0,1]
Conventional approach

- Estimating the mean absorption coefficients in octave bands:

\[
\bar{\alpha}(b) \stackrel{\text{def}}{=} \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b) S_i \quad (b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz})
\]
1. Introduction

Conventional approach

• Estimating the mean absorption coefficients in octave bands:

\[ \bar{\alpha}(b) \overset{\text{def}}{=} \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b) S_i \quad (b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz}) \]

• Equations from reverberation theory:

\[ \bar{\alpha}_{\text{Sabine}}(b) = 0.163 \frac{V}{S_{\text{tot}} RT_{60}(b)}, \quad \bar{\alpha}_{\text{Eyring}}(b) = -\ln(1 - \bar{\alpha}_{\text{Sabine}}(b)) \]
1. Introduction

Conventional approach

• Estimating the mean absorption coefficients in octave bands:

\[ \bar{\alpha}(b) \overset{\text{def}}{=} \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b) S_i \quad (b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz}) \]

• Equations from reverberation theory:

\[ \bar{\alpha}_{\text{Sabine}}(b) = 0.163 \frac{V}{S_{\text{tot}} RT_{60}(b)}, \quad \bar{\alpha}_{\text{Eyring}}(b) = -\ln(1 - \bar{\alpha}_{\text{Sabine}}(b)) \]
1. Introduction

Conventional approach

• Estimating the mean absorption coefficients in octave bands:

\[ \bar{\alpha}(b) \overset{\text{def}}{=} \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b) S_i \quad \left( b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz} \right) \]

• Equations from reverberation theory:

\[ \bar{\alpha}_{\text{Sabine}}(b) = 0.163 \frac{V}{S_{\text{tot}} RT_{60}(b)} \]

\[ \bar{\alpha}_{\text{Eyring}}(b) = -\ln(1 - \bar{\alpha}_{\text{Sabine}}(b)) \]

Reverberation time in sec
1. Introduction

Conventional approach

- Estimating the mean absorption coefficients in octave bands:
  \[
  \bar{\alpha}(b) \overset{\text{def}}{=} \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b) S_i \quad (b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz})
  \]

- Equations from reverberation theory:
  \[
  \bar{\alpha}_{\text{Sabine}}(b) = 0.163 \frac{V}{S_{\text{tot}} RT_{60}(b)}, \quad \bar{\alpha}_{\text{Eyring}}(b) = -\ln(1 - \bar{\alpha}_{\text{Sabine}}(b))
  \]

Obtained by Schroeder integration on a Room Impulse Response (RIR)
1. Introduction

Conventional approach

• Estimating the mean absorption coefficients in octave bands:

\[
\bar{\alpha}(b) \overset{\text{def}}{=} \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b) S_i \quad (b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz})
\]

• Equations from reverberation theory:

\[
\bar{\alpha}_{\text{Sabine}}(b) = 0.163 \frac{V}{S_{\text{tot}} R T_{60}(b)}, \quad \bar{\alpha}_{\text{Eyring}}(b) = -\ln(1 - \bar{\alpha}_{\text{Sabine}}(b))
\]

Obtained by Schroeder integration on a Room Impulse Response (RIR)

\[ -60 \text{dB} \]

\[ R T_{60} \]
1. Introduction

Conventional approach

- Estimating the mean absorption coefficients in octave bands:

\[
\bar{\alpha}(b) = \frac{1}{S_{\text{tot}}} \sum_{\text{surface } i} \alpha_i(b) S_i \quad (b \in \{125, 250, 500, \ldots, 4000\} \text{ Hz})
\]

- Equations from reverberation theory:

\[
\bar{\alpha}_{\text{Sabine}}(b) = 0.163 \frac{V}{S_{\text{tot}} RT_{60}(b)}, \quad \bar{\alpha}_{\text{Eyring}}(b) = -\ln(1 - \bar{\alpha}_{\text{Sabine}}(b))
\]

Obtained by Schroeder integration on a Room Impulse Response (RIR)

Limits:
- Assumes a diffuse sound field
- Only valid in *homogeneous conditions*
- Requires \( V \) and \( S_{\text{tot}} \)
1. Introduction

Proposed approach

Mean absorption coefficients

\[ \bar{\alpha}_{NN}(125\text{Hz}) \]
\[ \bar{\alpha}_{NN}(250\text{Hz}) \]
\[ \vdots \]
\[ \bar{\alpha}_{NN}(4000\text{Hz}) \]
1. Introduction

Proposed approach

Training Dataset

Neural Network

Mean absorption coefficients

$\bar{\alpha}_{NN}(125\text{Hz})$

$\bar{\alpha}_{NN}(250\text{Hz})$

$\vdots$

$\bar{\alpha}_{NN}(4000\text{Hz})$

Input RIRs

Output $\bar{\alpha}(b)$
1. Introduction

Proposed approach

- **Neural Network**
- **Mean absorption coefficients**
  - $\bar{\alpha}_{NN}(125\text{Hz})$
  - $\bar{\alpha}_{NN}(250\text{Hz})$
  - $\vdots$
  - $\bar{\alpha}_{NN}(4000\text{Hz})$

**Training Dataset**
- Input RIRs
- Output $\bar{\alpha}(\theta)$

**Acoustic simulator**
1. Introduction

Proposed approach

![Diagram showing the proposed approach involving a neural network for mean absorption coefficients.]
1. Introduction

Proposed approach

**Training Dataset**

- **RIRs**
- **Input RIRs**
- **Output $\bar{\alpha}(b)$**

**Acoustic simulator**

**Mean absorption coefficients**

- $\bar{\alpha}_{NN}(125\text{Hz})$
- $\bar{\alpha}_{NN}(250\text{Hz})$
- $\vdots$
- $\bar{\alpha}_{NN}(4000\text{Hz})$

**Neural Network**

- **Input**
- **Output** $\bar{\alpha}(b)$
OUTLINE

1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
OUTLINE

1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
2. Simulated Datasets

RIR training set simulation

Tradeoffs needed to be considered
RIR training set simulation

Tradeoffs needed to be considered

<table>
<thead>
<tr>
<th>Realism vs. Computational complexity</th>
<th></th>
</tr>
</thead>
</table>
# RIR training set simulation

Tradeoffs needed to be considered

<table>
<thead>
<tr>
<th>Realism vs. Computational complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roomsim [Schimmel et al. 2007] (shoebox)</td>
</tr>
</tbody>
</table>

Image source method + Diffuse rain algorithm
2. Simulated Datasets

RIR training set simulation

Tradeoffs needed to be considered

<table>
<thead>
<tr>
<th>Realism vs. Computational complexity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roomsim [Schimmel et al. 2007] (shoebox)</td>
</tr>
</tbody>
</table>

Image source method + Diffuse rain algorithm

Echo order: 50  Unique, random diffusion coefficient
2. Simulated Datasets

RIR training set simulation

Tradeoffs needed to be considered

<table>
<thead>
<tr>
<th>Realism vs. Computational complexity</th>
<th>Diversity vs. Model capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roomsim [Schimmel et al. 2007] (shoebox)</td>
<td></td>
</tr>
<tr>
<td>Image source method + Diffuse rain algorithm</td>
<td>Unique, random diffusion coefficient</td>
</tr>
</tbody>
</table>

Echo order: 50
2. Simulated Datasets

RIR training set simulation

Tradeoffs needed to be considered

<table>
<thead>
<tr>
<th>Realism vs. Computational complexity</th>
<th>Diversity vs. Model capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roomsim [Schimmel et al. 2007] (shoebox)</td>
<td>Focus on housing or office-type buildings:</td>
</tr>
<tr>
<td>Image source method</td>
<td>• Heights in [2.5, 4] meters</td>
</tr>
<tr>
<td>+ Diffuse rain algorithm</td>
<td>• Length/Width in [1.5, 10] meters</td>
</tr>
<tr>
<td>Echo order: 50</td>
<td>Unique, random diffusion coefficient</td>
</tr>
</tbody>
</table>

Image source method

Diffuse rain algorithm

Unique, random diffusion coefficient
2. Simulated Datasets

**RIR training set simulation**

Tradeoffs needed to be considered

<table>
<thead>
<tr>
<th>Realism vs. Computational complexity</th>
<th>Diversity vs. Model capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roomsim [Schimmel et al. 2007] (shoebox)</td>
<td></td>
</tr>
<tr>
<td>Image source method + Diffuse rain algorithm</td>
<td></td>
</tr>
<tr>
<td>Echo order: 50 Unique, random diffusion coefficient</td>
<td></td>
</tr>
</tbody>
</table>

- Focus on housing or office-type buildings:
  - Heights in [2.5, 4] meters
  - Length/Width in [1.5, 10] meters

- Omnidirectional source and receiver placed uniformly at random in the room
## 2. Simulated Datasets

### RIR training set simulation

Tradeoffs needed to be considered

<table>
<thead>
<tr>
<th>Realism vs. Computational complexity</th>
<th>Diversity vs. Model capacity</th>
</tr>
</thead>
<tbody>
<tr>
<td>Roomsim [Schimmel et al. 2007] (shoebox)</td>
<td></td>
</tr>
<tr>
<td>Image source method</td>
<td></td>
</tr>
<tr>
<td>+ Diffuse rain algorithm</td>
<td></td>
</tr>
<tr>
<td>Echo order: 50</td>
<td></td>
</tr>
<tr>
<td>Unique, random diffusion coefficient</td>
<td></td>
</tr>
<tr>
<td>• Focus on housing or office-type buildings:</td>
<td></td>
</tr>
<tr>
<td>• Heights in [2.5, 4] meters</td>
<td></td>
</tr>
<tr>
<td>• Length/Width in [1.5, 10] meters</td>
<td></td>
</tr>
<tr>
<td>• Omnidirectional source and receiver placed uniformly at random in the room</td>
<td></td>
</tr>
<tr>
<td>• Training set: 15,000 RIRs</td>
<td></td>
</tr>
<tr>
<td>• Development set: 5,000 RIRs</td>
<td></td>
</tr>
</tbody>
</table>
2. Simulated Datasets

Room acoustics sampling

1) Uniformly sampled (Unif.) dataset
Absorption coefficients of 6 surfaces in 6 octave bands drawn uniformly at random in [0, 1]
2. Simulated Datasets

**Room acoustics sampling**

1) Uniformly sampled \((\text{Unif.})\) dataset

Absorption coefficients of 6 surfaces in 6 octave bands drawn uniformly at random in \([0, 1]\)
2. Simulated Datasets

Room acoustics sampling

2) Reflectivity-Biased (RB) dataset
For each surface type (wall, ceiling, floor), toss a coin:
2. Simulated Datasets

Room acoustics sampling

2) Reflectivity-Biased (RB) dataset

For each surface type (wall, ceiling, floor), toss a coin:

- On heads: frequency-independent absorption coefficient in [0,0.12] (reflective)
2. Simulated Datasets

Room acoustics sampling

2) Reflectivity-Biased (RB) dataset

For each surface type (wall, ceiling, floor), toss a coin:

- On heads: frequency-independent absorption coefficient in [0,0.12] (reflective)
- On tails: random absorption profile inside a realistic range (treated)
2. Simulated Datasets

Room acoustics sampling

2) Reflectivity-Biased (RB) dataset

For each surface type (wall, ceiling, floor), toss a coin:
- On heads: frequency-independent absorption coefficient in $[0,0.12]$ (reflective)
- On tails: random absorption profile inside a realistic range (treated)
2. Simulated Datasets

Room acoustics sampling

2) Reflectivity-Biased (RB) dataset

For each surface type (wall, ceiling, floor), toss a coin:

- On heads: frequency-independent absorption coefficient in [0,0.12] (reflective)
- On tails: random absorption profile inside a realistic range (treated)
OUTLINE

1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
OUTLINE

1. Introduction
2. Simulated Datasets
3. **Neural Networks**
4. Experiments & Results
5. Conclusion & Future Work
3. Neural Networks

1) Multilayer Perceptron (MLP)

| Pre-processing | FC (8000 × 128) | ELU | FC (128 × 64) | ELU | FC (64 × 32) | ELU | FC (32 × 16) | ELU | FC (16 × 6) | Sigmoid |

\[
\begin{align*}
\bar{\alpha}_{NN}(125Hz) \\
\bar{\alpha}_{NN}(250Hz) \\
\vdots \\
\bar{\alpha}_{NN}(4000Hz)
\end{align*}
\]
3. Neural Networks

1) Multilayer Perceptron (MLP)

- Resample to 16 kHz
- Crop to 0.5 sec
- Normalize to max = 1
- Additive white Gaussian noise (SNR= 30 dB)

→ Input vector in $\mathbb{R}^{8000}$

Pre-processing

<table>
<thead>
<tr>
<th>FC (8000 x 128)</th>
<th>ELU</th>
</tr>
</thead>
<tbody>
<tr>
<td>FC (128 x 64)</td>
<td>ELU</td>
</tr>
<tr>
<td>FC (64 x 32)</td>
<td>ELU</td>
</tr>
<tr>
<td>FC (32 x 16)</td>
<td>ELU</td>
</tr>
<tr>
<td>FC (16 x 6)</td>
<td>Sigmoid</td>
</tr>
</tbody>
</table>

$\alpha_{NN}(125\text{Hz})$
$\alpha_{NN}(250\text{Hz})$
$\vdots$
$\alpha_{NN}(4000\text{Hz})$
3. Neural Networks

1) Multilayer Perceptron (MLP)

Input RIR

Pre-processing → FC (8000 x 128) → ELU → FC (128 x 64) → ELU → FC (64 x 32) → ELU → FC (32 x 16) → ELU → FC (16 x 6) → Sigmoid

\begin{align*}
\bar{\alpha}_{NN}(125\text{Hz}) \\
\bar{\alpha}_{NN}(250\text{Hz}) \\
\vdots \\
\bar{\alpha}_{NN}(4000\text{Hz})
\end{align*}

Fully Connected layer:

\[ h_{i+1} = Wh_i + b \]
3. Neural Networks

1) Multilayer Perceptron (MLP)

Input RIR

Pre-processing

FC (8000 x 128)

ELU

FC (128 x 64)

ELU

FC (64 x 32)

ELU

FC (32 x 16)

ELU

FC (16 x 6)

Sigmoid

Exponential Linear Unit:

\[ y = \begin{cases} 
  x, & x \geq 0 \\
  \alpha(e^x - 1), & x < 0 
\end{cases} \]

\( \bar{\alpha}_\text{NN}(125\text{Hz}) \)

\( \bar{\alpha}_\text{NN}(250\text{Hz}) \)

\( \vdots \)

\( \bar{\alpha}_\text{NN}(4000\text{Hz}) \)
3. Neural Networks

1) Multilayer Perceptron (MLP)

- Pre-processing
- FC (8000 x 128) \rightarrow ELU
- FC (128 x 64) \rightarrow ELU
- FC (64 x 32) \rightarrow ELU
- FC (32 x 16) \rightarrow ELU
- FC (16 x 6) \rightarrow Sigmoid

### Sigmoid:

\[
y = \frac{1}{1 + e^{-x}}
\]

\[\bar{\alpha}_{NN}(125\text{Hz})
\bar{\alpha}_{NN}(250\text{Hz})
\vdots
\bar{\alpha}_{NN}(4000\text{Hz})\]
3. Neural Networks

1) Multilayer Perceptron (MLP)

- Pre-processing
- FC (8000 x 128) → ELU
- FC (128 x 64) → ELU
- FC (64 x 32) → ELU
- FC (32 x 16) → ELU
- FC (16 x 6) → Sigmoid

Output vector in $[0, 1]^6$
Loss Function = Mean Squared Error
Optimal parameters on dev. set over 200 epochs
3. Neural Networks

1) Multilayer Perceptron (MLP)

Input RIR

Pre-processing → FC (8000 x 128) → ELU → FC (128 x 64) → ELU → FC (64 x 32) → ELU → FC (32 x 16) → ELU → FC (16 x 6) → Sigmoid → \[\tilde{\alpha}_{NN}(125\text{Hz})\] → \[\tilde{\alpha}_{NN}(250\text{Hz})\] → \[\tilde{\alpha}_{NN}(4000\text{Hz})\]

2) Convolutional Neural Network (CNN)

Input RIR

Pre-processing → Conv1D (1x64)x33 → ReLU + MaxPool(4) → Conv1D (64x32)x17 → ReLU + MaxPool(4) → Conv1D (32x16)x9 → ReLU + MaxPool(4) → Conv1D (2000 x 32) → ReLU → FC (32 x 6) → Sigmoid → \[\tilde{\alpha}_{NN}(125\text{Hz})\] → \[\tilde{\alpha}_{NN}(250\text{Hz})\] → \[\tilde{\alpha}_{NN}(4000\text{Hz})\]
3. Neural Networks

1) Multilayer Perceptron (MLP)

Input RIR

- Pre-processing
- FC (8000 x 128) ELU
- FC (128 x 64) ELU
- FC (64 x 32) ELU
- FC (32 x 16) ELU
- FC (16 x 6) Sigmoid

\[ \bar{\alpha}_{NN}(125\text{Hz}), \bar{\alpha}_{NN}(250\text{Hz}), \cdots, \bar{\alpha}_{NN}(4000\text{Hz}) \]

1D convolutional layer:
- 64 input channels
- 32 output channels
- Kernel size: 17

2) Convolutional Neural Network (CNN)

Input RIR

- Pre-processing
- Conv1D (1x64)x33 ReLU
- MaxPool
- Conv1D (64x32)x17 ReLU + MaxPool
- Conv1D (32x16)x9 ReLU + MaxPool
- FC (2000 x 32) ReLU
- FC (32 x 6) Sigmoid

\[ \bar{\alpha}_{NN}(125\text{Hz}), \bar{\alpha}_{NN}(250\text{Hz}), \cdots, \bar{\alpha}_{NN}(4000\text{Hz}) \]
### 1) Multilayer Perceptron (MLP)

Input RIR → Pre-processing → FC (8000 x 128) → ELU → ... → FC (16 x 6) → Sigmoid → \( \bar{\alpha}_{NN}(125Hz) \)
\( \bar{\alpha}_{NN}(250Hz) \)
\( \vdots \)
\( \bar{\alpha}_{NN}(4000Hz) \)

**Rectified Linear Unit:**
\[ y = \max(0, x) \]

**MaxPool(4):**
\[ \max \]

### 2) Convolutional Neural Network (CNN)

Input RIR → Pre-processing → Conv1D (1x64)x33 → ReLU + MaxPool(4) → ... → Conv1D (2000 x 32) → ReLU → FC (32 x 6) → Sigmoid → \( \bar{\alpha}_{NN}(125Hz) \)
\( \bar{\alpha}_{NN}(250Hz) \)
\( \vdots \)
\( \bar{\alpha}_{NN}(4000Hz) \)
3. Neural Networks

1) Multilayer Perceptron (MLP)

Input RIR → Pre-processing → FC (8000 x 128) → ELU → FC (128 x 64) → ELU → FC (64 x 32) → ELU → FC (32 x 16) → Sigmoid → $\tilde{\alpha}_{NN}(125\text{Hz})$
$\tilde{\alpha}_{NN}(250\text{Hz})$
$\vdots$
$\tilde{\alpha}_{NN}(4000\text{Hz})$

2) Convolutional Neural Network (CNN)

Input RIR → Pre-processing → Conv1D (1x64)x33 → ReLU + MaxPool(4) → Conv1D (64x32)x17 → ReLU + MaxPool(4) → Conv1D (32x16)x9 → ReLU + MaxPool(4) → FC (2000 x 32) → ReLU → FC (32 x 6) → Sigmoid → $\tilde{\alpha}_{NN}(125\text{Hz})$
$\tilde{\alpha}_{NN}(250\text{Hz})$
$\vdots$
$\tilde{\alpha}_{NN}(4000\text{Hz})$
OUTLINE

1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
OUTLINE

1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
4. Experiments & Results

**Test set:** Same simulator, real material profiles, 5 room geometries, 500 RIRs
4. Experiments & Results

**Test set:** Same simulator, real material profiles, 5 room geometries, 500 RIRs
- Comparing the two training sets (Unif., RB) and the two neural networks (MLP, CNN) against Sabine and Eyring’s laws (given true $S_{tot}$ and $V$)

4. Experiments & Results

**Test set:** Same simulator, real material profiles, 5 room geometries, 500 RIRs

- Comparing the two training sets (Unif., RB) and the two neural networks (MLP, CNN) against Sabine and Eyring’s laws (given true $S_{tot}$ and $V$)
4. Experiments & Results

**Test set:** Same simulator, real material profiles, 5 room geometries, 500 RIRs
- Comparing the two training sets (Unif., RB) and the two neural networks (MLP, CNN) against Sabine and Eyring’s laws (given true $S_{\text{tot}}$ and $V$)

Training on uniformly sampled acoustics fails to outperform reverberation theory
4. Experiments & Results

**Test set:** Same simulator, real material profiles, 5 room geometries, 500 RIRs
- Comparing the two training sets (Unif., RB) and the two neural networks (MLP, CNN) against Sabine and Eyring’s laws (given true $S_{tot}$ and $V$)

- Training on uniformly sampled acoustics fails to outperform reverberation theory
- Training on the reflectivity-biased set significantly outperforms both baselines
4. Experiments & Results

- Influence of room geometry:

![Diagram showing the influence of room geometry on the median, mean, and absolute error on $\hat{a}(b)$ for different geometries: Homogenous, Flat, Elongated. The diagram indicates that the advantage of the learning-based approach is clearer under non-homogenous geometries.]

Advantage of the learning-based approach clearer under non-homogenous geometries
4. Experiments & Results

- Influence of reverberation time:

\[ RT_{60}(b) < 0.5 \text{s} \]
\[ 0.5 < RT_{60}(b) < 1.5 \text{s} \]
\[ RT_{60}(b) > 1.5 \text{s} \]

A higher reverberation time helps both approaches
4. Experiments & Results

- Influence of reverberation time:
  - $RT_{60}(b) < 0.5s$
  - $0.5s < RT_{60}(b) < 1.5s$
  - $RT_{60}(b) > 1.5s$

- Influence of noise:

![Graph showing the absolute error on $\tilde{\alpha}(b)$ vs SNR for CNN-RB and Eyring methods.](image)
4. Experiments & Results

- Tests on another simulator: **pyroomacoustics** [Scheibler et al. 2018]

![Roomsim RIR](image1)

- No diffusion
- Frequency-independent $\bar{\alpha}(b)$
- Sinc kernel
4. Experiments & Results

- Tests on another simulator: **pyroomacoustics** [Scheibler et al. 2018]

    |                  | Roomsim RIR | Pyroom. RIR |
    |------------------|-------------|-------------|
    | CNN-RB           | 0.026       | 0.085       |
    | MLP-RB           | 0.024       | 0.207       |
    | Eyring           | 0.043       | 0.093       |

Mean errors on $\bar{\alpha}(b)$:

- No diffusion
- Frequency-independent $\bar{\alpha}(b)$
- Sinc kernel

Encouraging robustness to the specific training simulator
OUTLINE

1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
OUTLINE

1. Introduction
2. Simulated Datasets
3. Neural Networks
4. Experiments & Results
5. Conclusion & Future Work
Conclusion

- Simulation-based learning is a promising approach for the acoustic diagnosis of rooms from RIRs
- By exploiting the fine temporal structure of RIRs, the networks outperform reverberation theory in non-homogenous conditions
- Careful acoustic sampling in the training set is crucial
4. Conclusions and Future Work

**Conclusion**

- Simulation-based learning is a promising approach for the acoustic diagnosis of rooms from RIRs
- By exploiting the fine temporal structure of RIRs, the networks outperform reverberation theory in non-homogenous conditions
- Careful acoustic sampling in the training set is crucial

**Future work**

- Generalization to real measured RIRs
  - Domain adaptation techniques
- Estimating the absorption profiles of individual surfaces
  - Geometry-informed models
THANK YOU