From Sound to Action: Deep Learning for Audio-Based Localization and Navigation in Robotics

Gaétan Lepage

July 15, 2025

Social Robotics

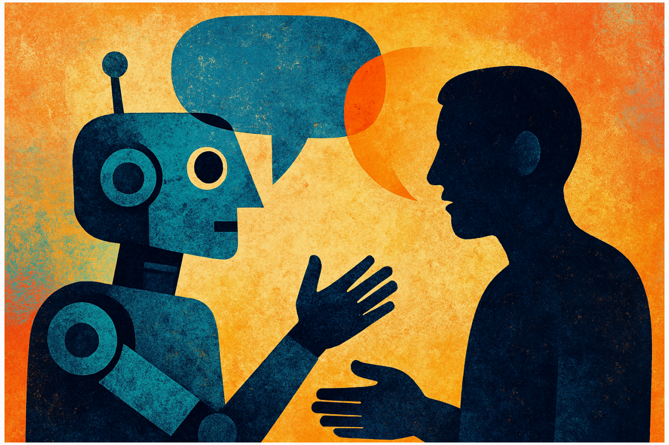
- Social robotics aims to build capable robotic agents.
- They must **collaborate with humans** (social acceptance, etc.)
- **Human Robot Interactions** entail a wide range of challenges

Social Robotics

- Social robotics aims to build capable robotic agents.
- They must **collaborate with humans** (social acceptance, etc.)
- **Human Robot Interactions** entail a wide range of challenges
- Key challenges:
  - **Perception:** Extract relevant information from *multi-modal data* captured by diverse sensors
  - **Action:** Learn relevant policies to achieve desirable behaviors (navigation, grasping, conversation, etc.)

Challenges of Auditory Perception in Robotics

- Humans mainly communicate through **speech**
- Robots must properly **understand humans** to have relevant interactions
- Sound can also be used to **localize speakers**
- Core **acoustic tasks** in robotics:
  - Automatic Speech Recognition (ASR) [1]
  - Sound Source Localization (SSL) [2][3]
  - Conversational Speech Generation [4]



[1] Yu et al., *Automatic Speech Recognition*. Springer, 2016.

[2] Argentieri et al., "A Survey on Sound Source Localization in Robotics: From Binaural to Array Processing Methods," *Comput. Speech Lang.*, 2015.

[3] Grumiaux et al., "A Survey of Sound Source Localization with Deep Learning Methods," *JASA*, 2022.

[4] Defossez et al., "Moshi: a Speech-Text Foundation Model for Real-Time Dialogue," *arXiv preprint*, 2024.

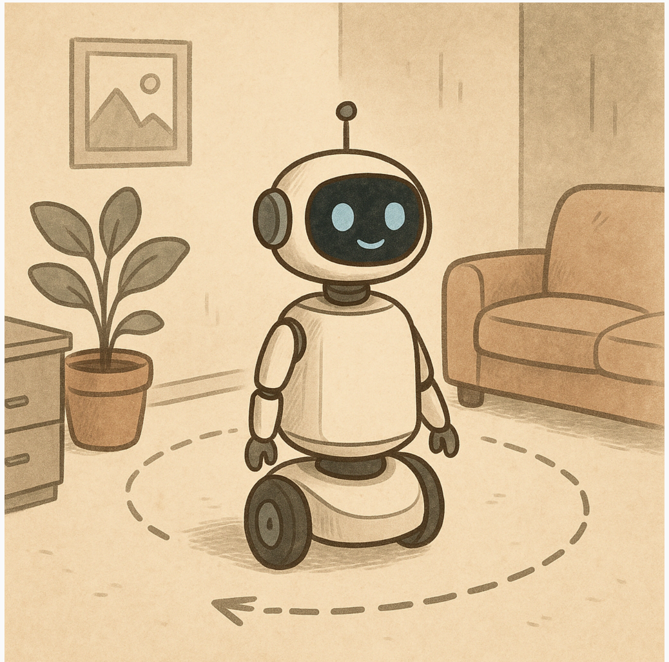
[1] Majumder et al., “Move2hear: Active Audio-Visual Source Separation,” in *ICCV*, 2021.

[2] Lathuilière et al., “Neural Network Based Reinforcement Learning for Audio-Visual Gaze Control in Human-Robot Interaction,” *Pattern Recognition Letters*, 2019.

[3] Zacharaki et al., “Safety Bounds in Human Robot Interaction: A Survey,” *Safety science*, 2020.

[4] Ottoni et al., “A Systematic Review of Human-Robot Interaction: the Use of Emotions and the Evaluation of Their Performance,” *International Journal of Social Robotics*, 2024.

Learning Robot Behaviors



Robots need to react to their environment and take actions

- React dynamically to the environment
- Accomplish interactive or collaborative tasks [1][2]
- Several objectives and constraints can be described

Challenges:

- Designing tractable objectives for robots behavior
- Ensuring humans safety [3]
- Making robots *socially accepted* by humans? [4]
- Detecting and reacting to external events?
- Learning flexible policies

Plan

Introduction

1 **Acoustic Robot Simulator**

Simulate dynamic acoustic environments

2 **(Active) Sound Source Localization**

Accurately localize speaker(s) in a reverberant room

3 **Deep RL for Sound-Based Navigation**

Learn to navigate to hear humans better

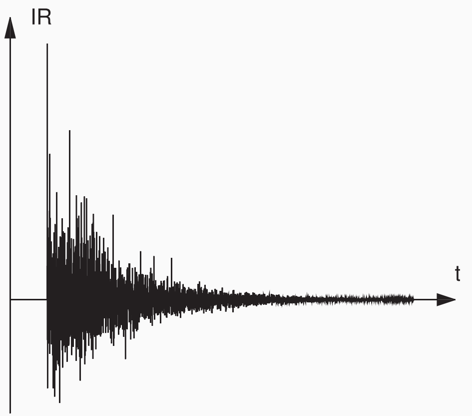
Conclusion

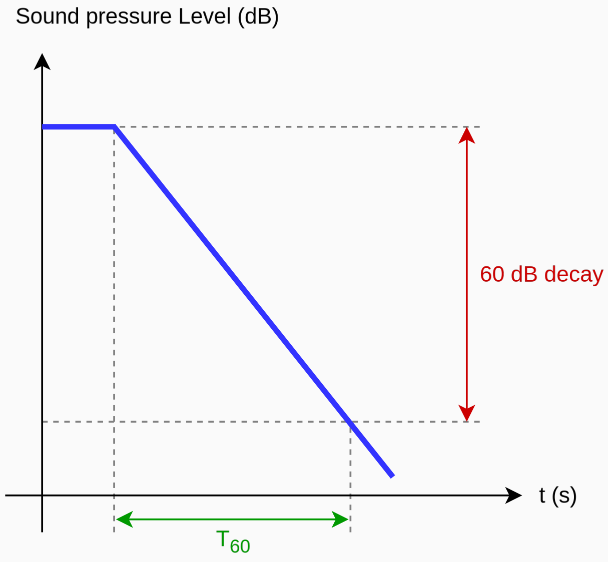
Motivations:

- Experimenting on real robotic platforms is limiting
- Collecting significant amounts of data
- Lack of holistic approaches to interactive acoustic simulation

Objectives:

- Modeling realistic acoustic environments
- Simulating sound propagation in reverberant rooms
- Provide high-level primitives for experimenting with robotic auditory perception

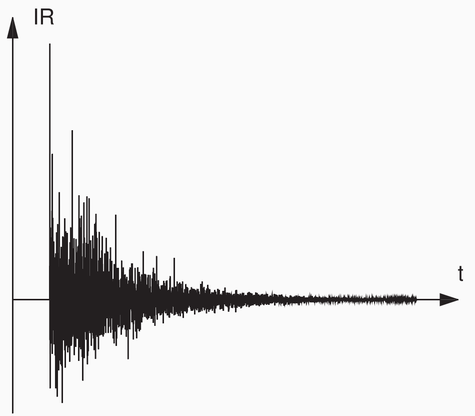


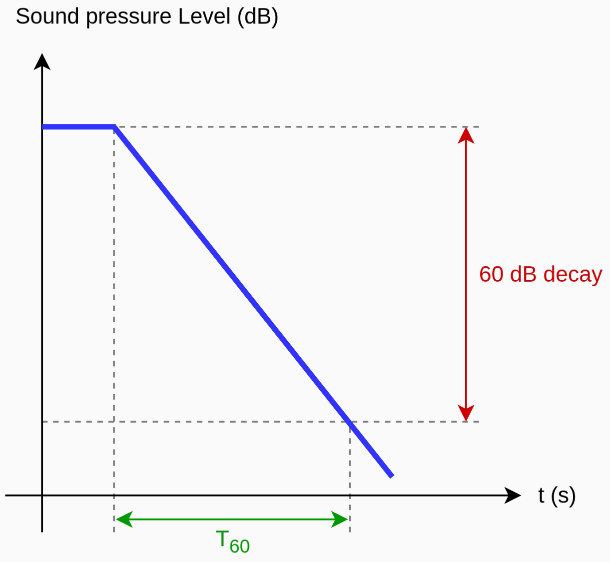


RIR properties:

- Characterizes the reverberation properties of the room
- Computed for each source-microphone pair
- T_{60} measures the reverberation level
- The resulting image/microphone signal is obtained by convolving each source signal with the corresponding RIR, and summing over the sources

Room Impulse Response





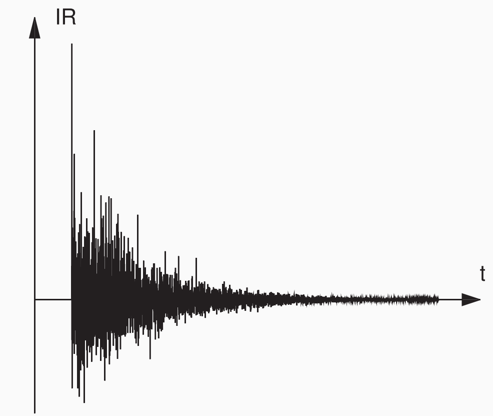
RIR properties:

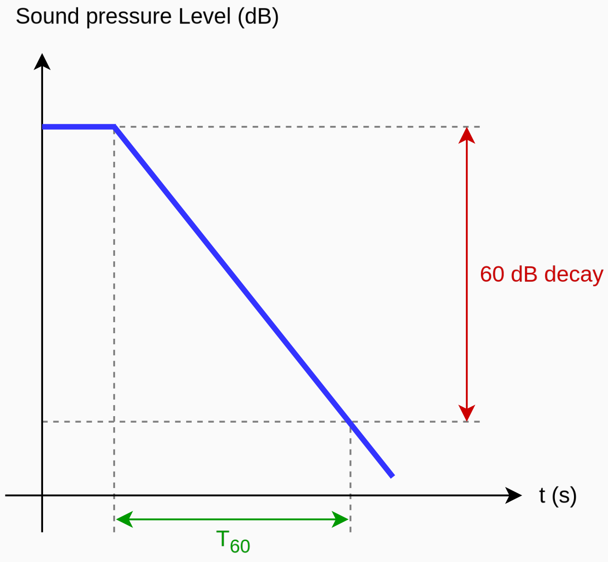
- Characterizes the reverberation properties of the room
- Computed for each source-microphone pair
- T_{60} measures the reverberation level
- The resulting image/microphone signal is obtained by convolving each source signal with the corresponding RIR, and summing over the sources

Single-source microphone signal:

$$y[n] = (h * x)[n]$$

Room Impulse Response





RIR properties:

- Characterizes the reverberation properties of the room
- Computed for each source-microphone pair
- T_{60} measures the reverberation level
- The resulting image/microphone signal is obtained by convolving each source signal with the corresponding RIR, and summing over the sources

Single-source microphone signal:

$$y[n] = (h * x)[n]$$

Multi-source microphone signal:

$$y[n] = \sum_{i=1}^{n_s} (h_i * x_i)[n]$$

Existing Simulation Methods

- **Numerical simulation** [1][2]:
  - Approximation of the solution of a physical equation (e.g. Helmholtz)
  - Numerical solver (FDTD, BEM, etc.)
  - Accurate, but computationally expensive

[1] D. Botteldooren, "Acoustical Finite-Difference Time-Domain Simulation in a Quasi-Cartesian Grid," *JASA*, 1994.

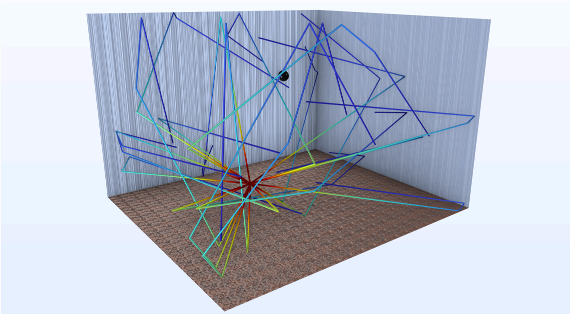
[2] Raghuvanshi et al., "Efficient and Accurate Sound Propagation Using Adaptive Rectangular Decomposition," *IEEE TVCG*, 2009.

[3] Cao et al., "Interactive Sound Propagation with Bidirectional Path Tracing," *ACM TOG*, 2016.

[4] Allen et al., "Image method for efficiently simulating small-room acoustics," *JASA*, 1979.

Existing Simulation Methods

- **Numerical simulation** [1][2]:
  - Approximation of the solution of a physical equation (e.g. Helmholtz)
  - Numerical solver (FDTD, BEM, etc.)
  - Accurate, but computationally expensive
- **Geometrical Acoustics**
  - Ray-tracing [3]
  - Image Source Model [4]



[1] D. Botteldooren, "Acoustical Finite-Difference Time-Domain Simulation in a Quasi-Cartesian Grid," *JASA*, 1994.

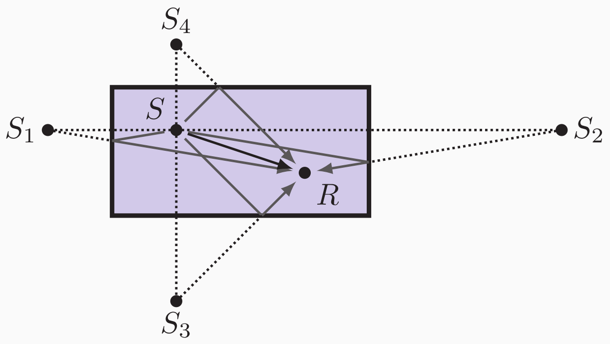
[2] Raghuvanshi et al., "Efficient and Accurate Sound Propagation Using Adaptive Rectangular Decomposition," *IEEE TVCG*, 2009.

[3] Cao et al., "Interactive Sound Propagation with Bidirectional Path Tracing," *ACM TOG*, 2016.

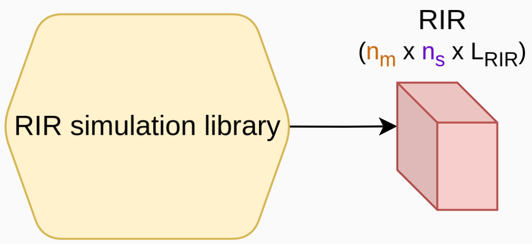
[4] Allen et al., "Image method for efficiently simulating small-room acoustics," *JASA*, 1979.

Existing Simulation Methods

RIR simulation: Generate RIR from a 3D room specification



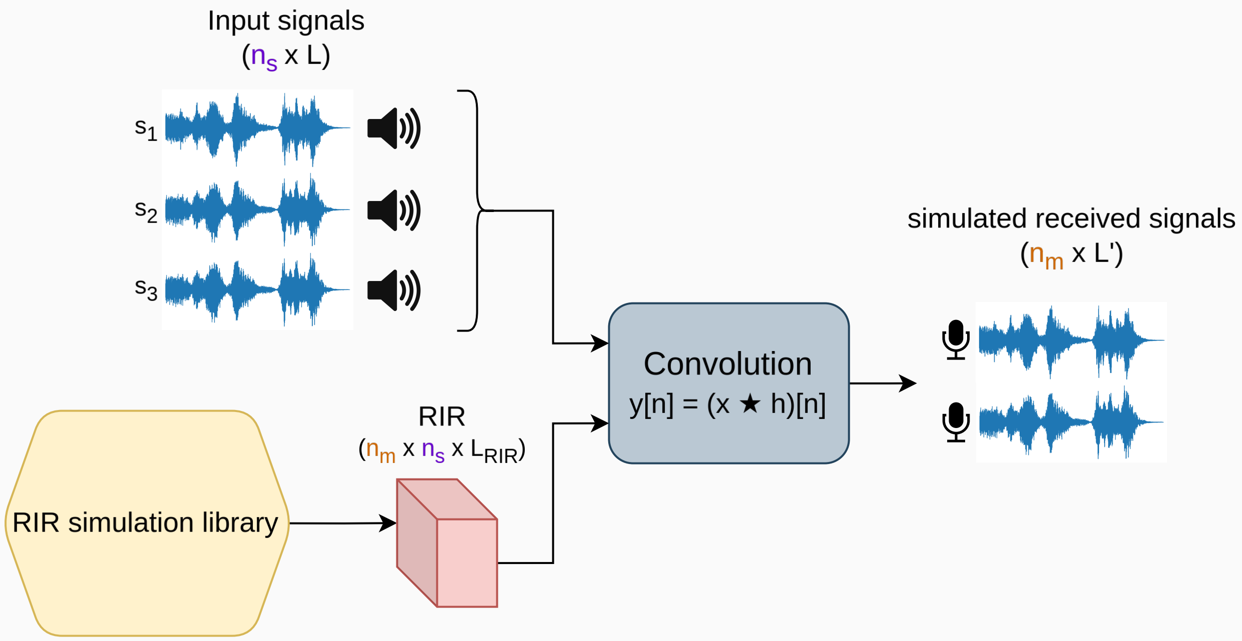
Audio-Processing Pipeline



Support for two backend libraries: *Pyroomacoustics* [1] and *gpuRIR* [2].

[1] Scheibler et al., “Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms,” in *ICASSP*, 2018.

[2] Diaz-Guerra et al., “gpuRIR: A Python Library for Room Impulse Response Simulation with GPU Acceleration,” *Multimedia Tools and Applications*, 2021.



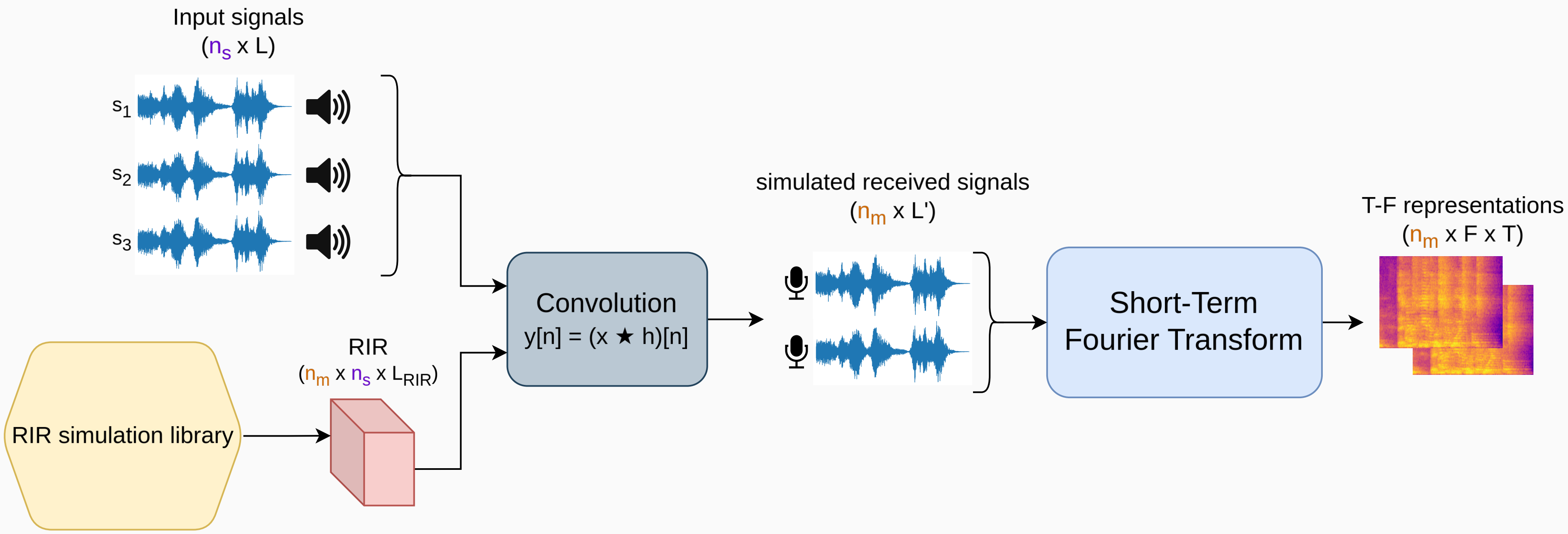
Support for two backend libraries: *Pyroomacoustics* [1] and *gpuRIR* [2].

[1] Scheibler et al., "Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms," in *ICASSP*, 2018.

Audio-Processing Pipeline

[2] Diaz-Guerra et al., "gpuRIR: A Python Library for Room Impulse Response Simulation with GPU Acceleration," *Multimedia Tools and Applications*, 2021.

Audio-Processing Pipeline



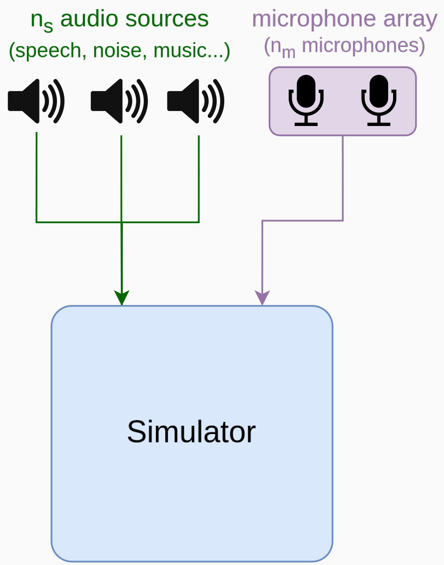
Support for two backend libraries: *Pyroomacoustics* [1] and *gpuRIR* [2].

[1] Scheibler et al., "Pyroomacoustics: A Python Package for Audio Room Simulation and Array Processing Algorithms," in *ICASSP*, 2018.

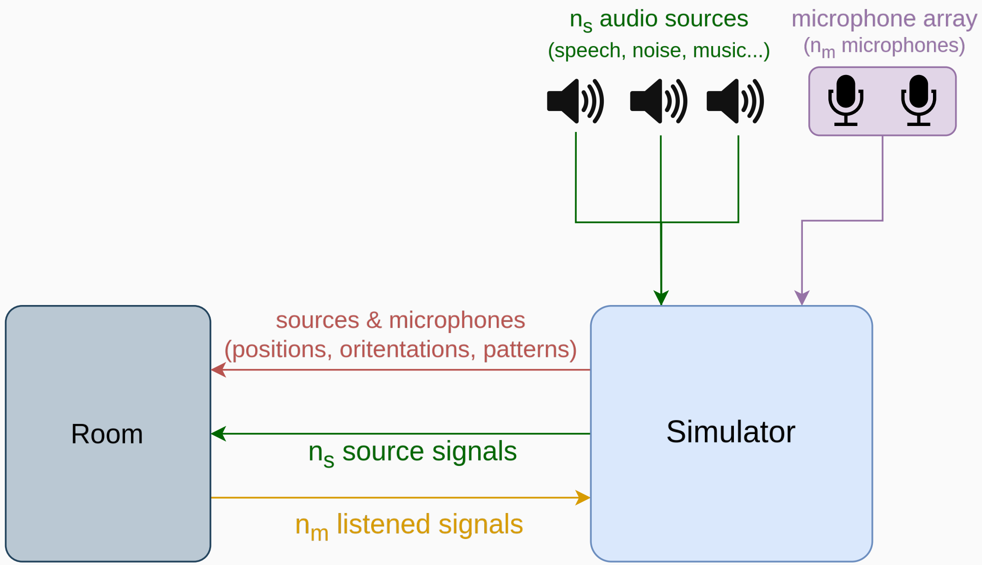
Audio-Processing Pipeline

[2] Diaz-Guerra et al., “gpuRIR: A Python Library for Room Impulse Response Simulation with GPU Acceleration,” *Multimedia Tools and Applications*, 2021.

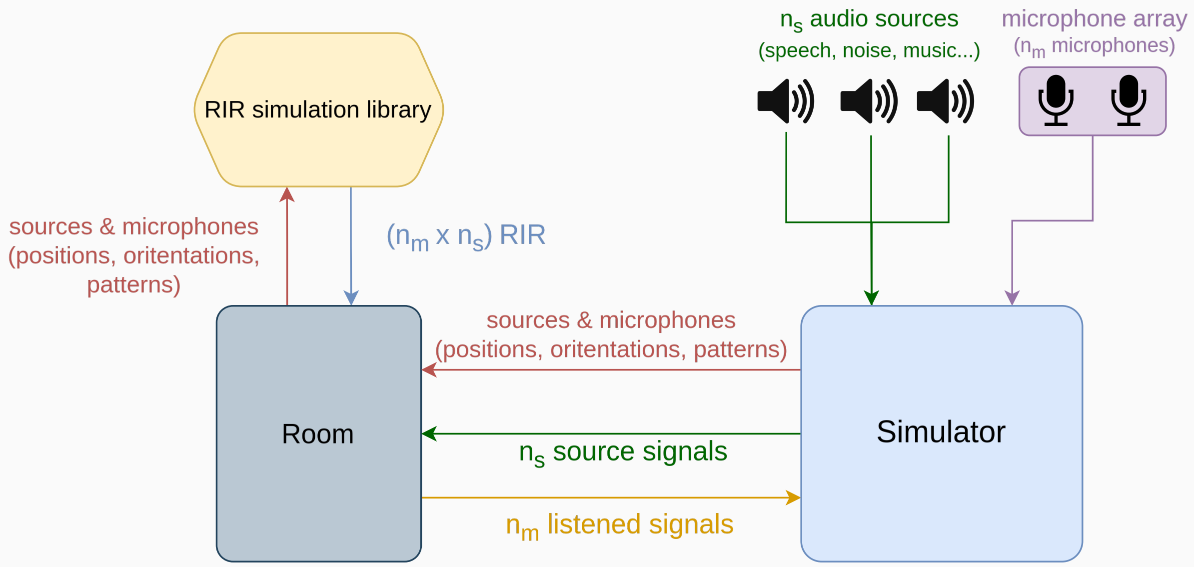
Simulator Architecture



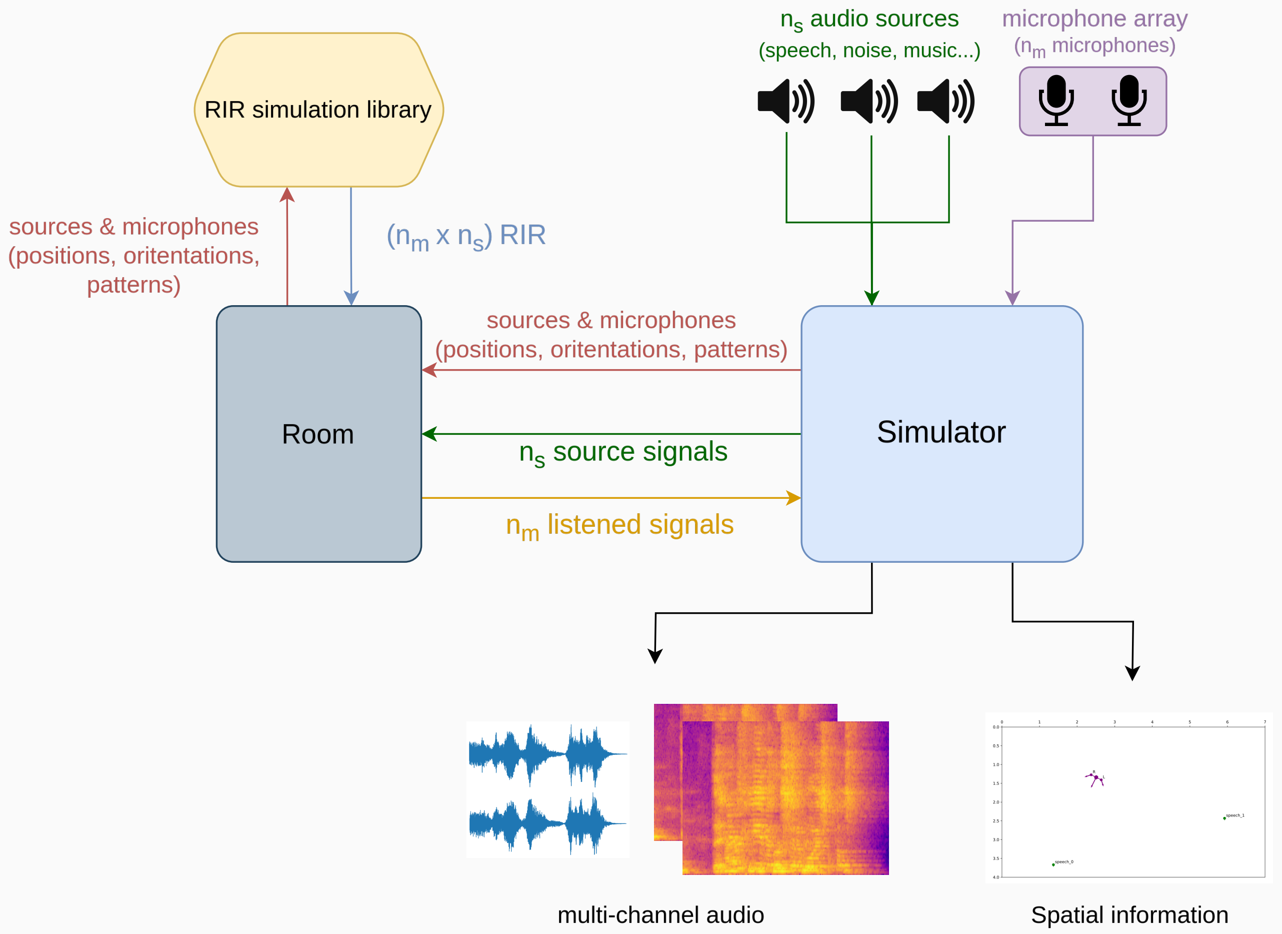
Simulator Architecture

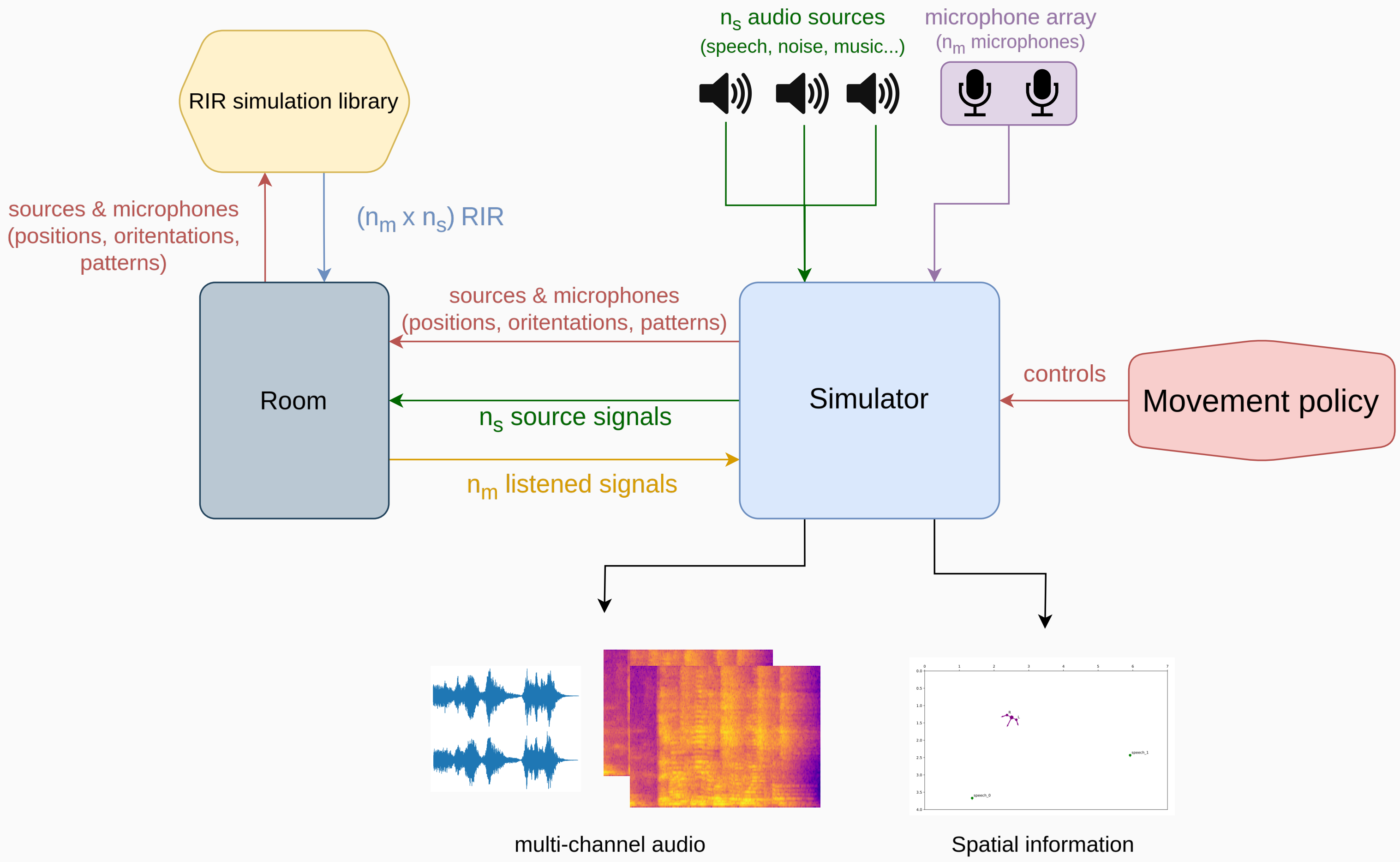


Simulator Architecture



Simulator Architecture





RIR simulation library
sources & microphones (positions, oritentations, patterns)
(nm x ns) RIR
ns audio sources (speech, noise, music...)
microphone array (nm microphones)
Room
sources & microphones (positions, oritentations, patterns)
ns source signals
nm listened signals
Simulator
controls
Movement policy
multi-channel audio
Spatial information

Code Example

```
from rl_audio_nav.audio_simulator import GpuRirRoom, BinauralArray, AudioSimilator
```

Code Example

```python
from rl_audio_nav.audio_simulator import GpuRirRoom, BinauralArray, AudioSimilator

# Initialization
room = GpuRirRoom(size_x=4, size_y=7, rt_60=0.3)
```

Code Example

```python
from rl_audio_nav.audio_simulator import GpuRirRoom, BinauralArray, AudioSimilator


# Initialization
room = GpuRirRoom(size_x=4, size_y=7, rt_60=0.3)

mic_array = BinauralArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
  mic_dist=2, # cm
)
```

Code Example

```python
from rl_audio_nav.audio_simulator import GpuRirRoom, BinauralArray, AudioSimilator

# Initialization
room = GpuRirRoom(size_x=4, size_y=7, rt_60=0.3)
mic_array = BinauralArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
  mic_dist=2, # cm
)
audio_simulator = AudioSimulator(room, mic_array, n_speech_sources=3)
```

Code Example

```python
from rl_audio_nav.audio_simulator import GpuRirRoom, BinauralArray, AudioSimilator


# Initialization
room = GpuRirRoom(size_x=4, size_y=7, rt_60=0.3)

mic_array = BinauralArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
  mic_dist=2, # cm
)

audio_simulator = AudioSimulator(room, mic_array, n_speech_sources=3)

# Load speech signals and perform simulation
audio_simulator.step()
```

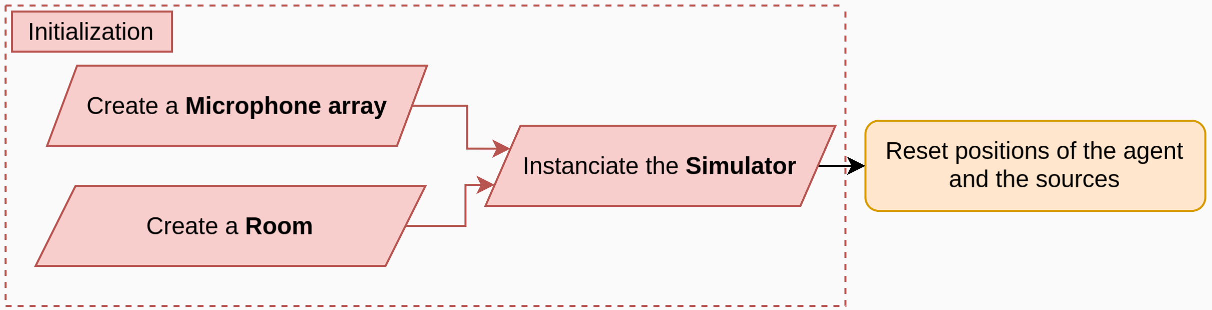
Code Example

```
from rl_audio_nav.audio_simulator import GpuRirRoom, BinauralArray, AudioSimilator

# Initialization
room = GpuRirRoom(size_x=4, size_y=7, rt_60=0.3)
mic_array = BinauralArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
  mic_dist=2, # cm
)
audio_simulator = AudioSimulator(room, mic_array, n_speech_sources=3)
# Load speech signals and perform simulation
audio_simulator.step()
# (2, F, T) complex tensor
stft = audio_simulator.get_agent_stft()
```

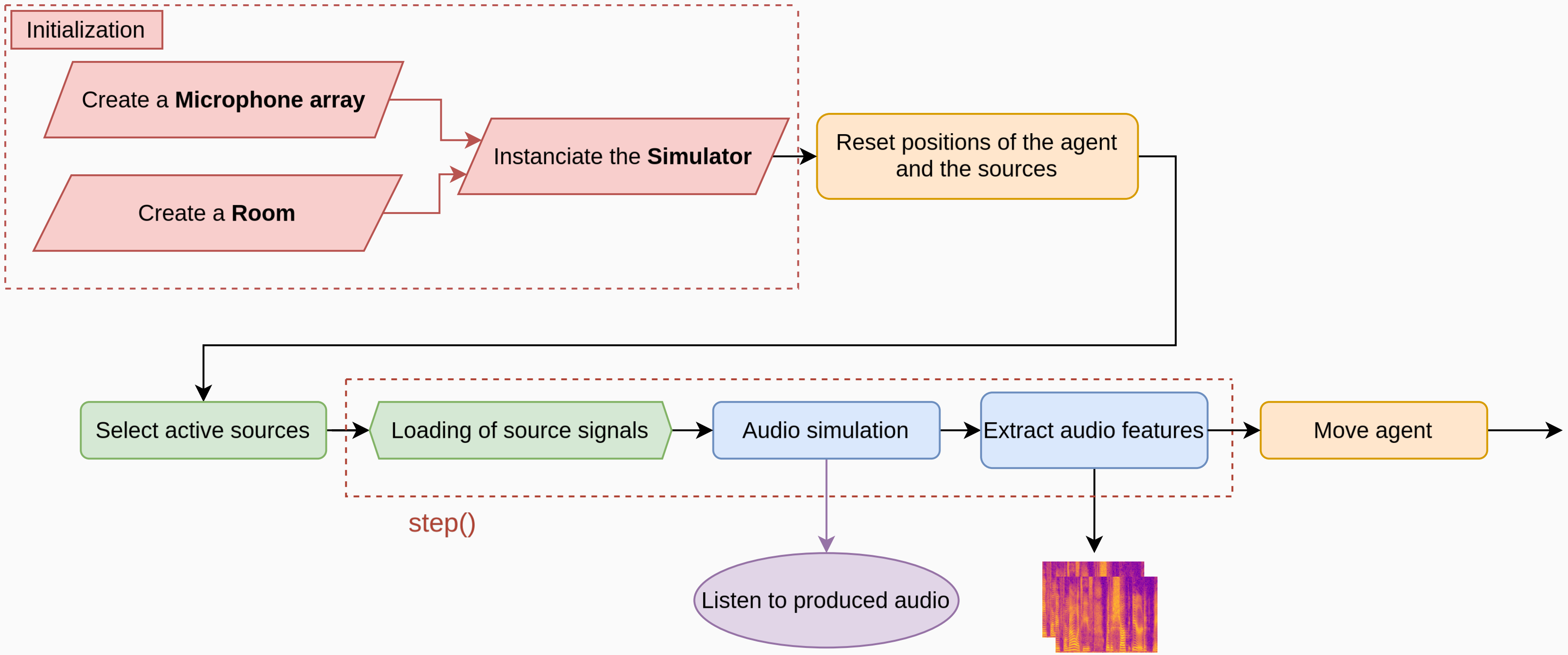
Code Example

```python
from rl_audio_nav.audio_simulator import GpuRirRoom, BinauralArray, AudioSimilator


# Initialization
room = GpuRirRoom(size_x=4, size_y=7, rt_60=0.3)

mic_array = BinauralArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
  mic_dist=2, # cm
)

audio_simulator = AudioSimulator(room, mic_array, n_speech_sources=3)

# Load speech signals and perform simulation
audio_simulator.step()

# (2, F, T) complex tensor
stft = audio_simulator.get_agent_stft()

# Compute the DoA with respect to the "speech_1" source
doa_source_1 = audio_simulator.get_doa("speech_1")
```

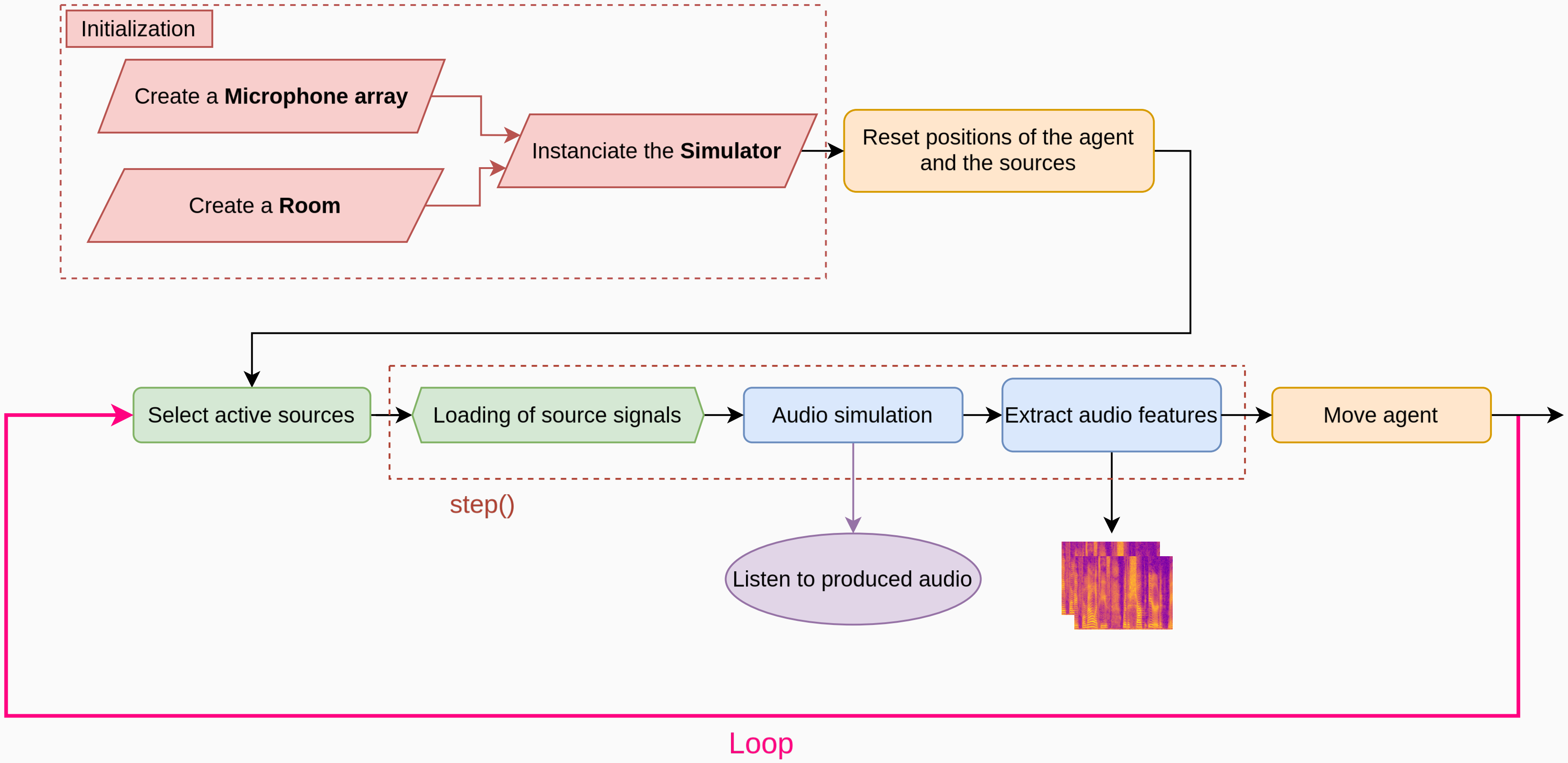
Modeling Active Scenarios

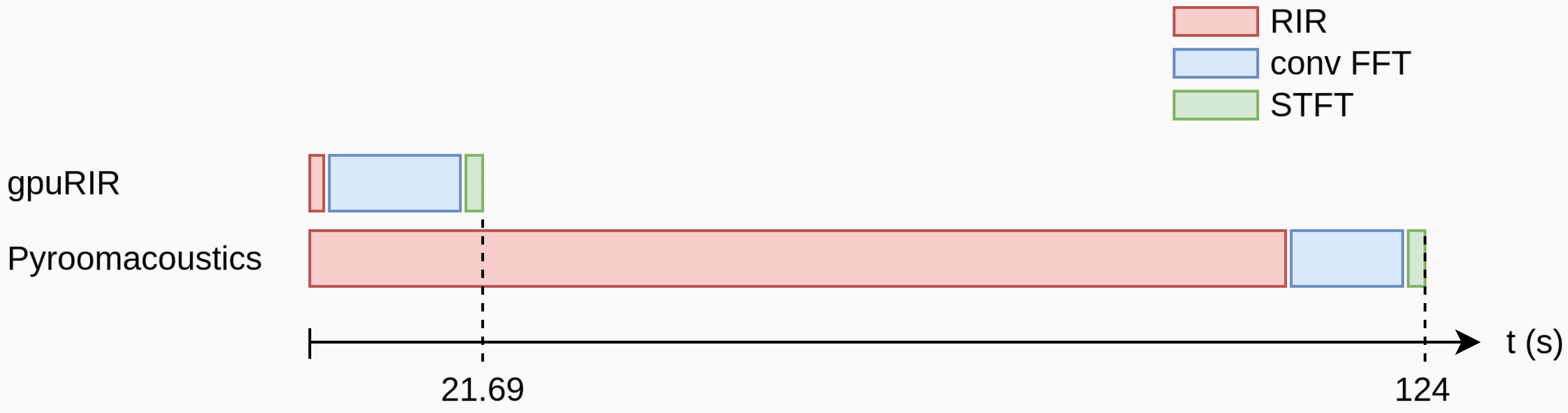


Modeling Active Scenarios



Modeling Active Scenarios





	T_{sim} (s, (%))	t_{RIR} (s, (%))	t_{conv} (s, (%))	t_{STFT} (s, (%))
gpuRIR	**21.7** (100%)	**3.69** (17%)	14.6 (67%)	2.4 (11%)
Pyroomacoustics	124 (100%)	109 (88%)	11.6 (9.4%)	2.3 (1.8%)

- Complete solution for modeling **various acoustic robotics scenarios**
- **High-level, intuitive API** to easily and quickly build on top of
- Extraction of **various spectral representations** of simulated signals
- Great **flexibility** allowing for various use-cases:
  - Dataset generation
  - Modeling interactive scenarios where both microphones and sources can move
  - Use as an environment to train Deep RL agents

[1] Gustafsson et al., “Source Localization in Reverberant Environments: Modeling and Statistical Analysis,” *IEEE TSAP*, 2004.

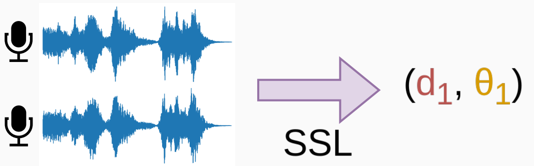
[2] Schmidt, “Multiple Emitter Location and Signal Parameter Estimation,” *IEEE Trans. Antennas Propag.*, 1986.

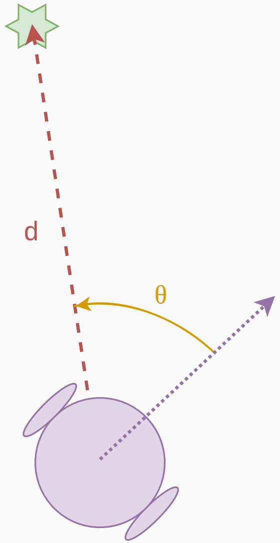
[3] Grumiaux et al., “A Survey of Sound Source Localization with Deep Learning Methods,” *JASA*, 2022.

[4] Argentieri et al., “A Survey on Sound Source Localization in Robotics: From Binaural to Array Processing Methods,” *Comput. Speech Lang.*, 2015.

The Static SSL Problem

- SSL (Sound Source Localization): estimate the position of one or multiple sound sources
  - Dense scientific literature: from classical sound processing methods [1][2] to deep learning techniques [3]
  - Often applied to robotics [4]
  - Multiple variations of the task





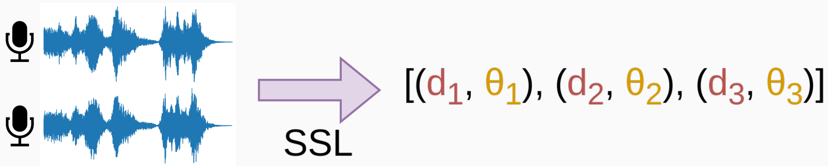
[1] Gustafsson et al., “Source Localization in Reverberant Environments: Modeling and Statistical Analysis,” *IEEE TSAP*, 2004.

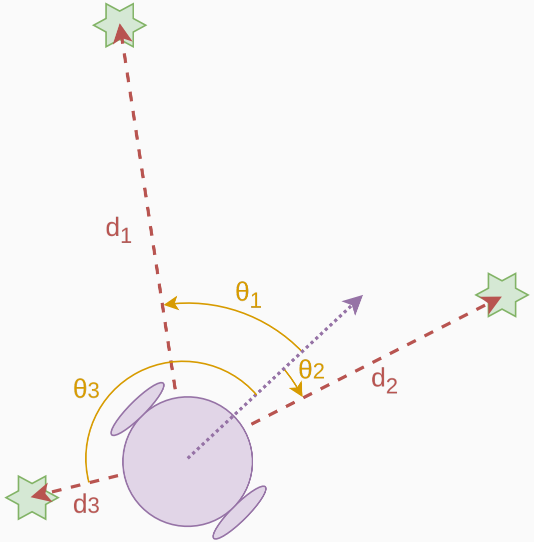
[2] Schmidt, “Multiple Emitter Location and Signal Parameter Estimation,” *IEEE Trans. Antennas Propag.*, 1986.

[3] Grumiaux et al., “A Survey of Sound Source Localization with Deep Learning Methods,” *JASA*, 2022.

[4] Argentieri et al., “A Survey on Sound Source Localization in Robotics: From Binaural to Array Processing Methods,” *Comput. Speech Lang.*, 2015.

- SSL (Sound Source Localization): estimate the position of one or multiple sound sources
  - Dense scientific literature: from classical sound processing methods [1][2] to deep learning techniques [3]
  - Often applied to robotics [4]
  - Multiple variations of the task





Motivation:

- Real-world robotics scenarios are often dynamic
- Static SSL frameworks struggle predicting the source-array distance

[1]Nakadai et al., "Active Audition for Humanoid," in *AAAI/IAAI*, 2000.

[2]Nguyen et al., "Long-Term Robot Motion Planning for Active Sound Source Localization with Monte Carlo Tree Search," in *IEEE/HSCMA*, 2017.

[3]Bustamante et al., "Towards Information-Based Feedback Control for Binaural Active Localization," in *ICASSP*, 2016.

[1]Evers et al., "The LOCATA Challenge: Acoustic Source Localization and Tracking," *IEEE/TASLP*, 2020.

Active SSL

Active SSL

Motivation:

- Real-world robotics scenarios are often dynamic
- Static SSL frameworks struggle predicting the source-array distance

Intuition:

- Aggregate instantaneous angular estimates over time
- Leverage the robot movement to refine the predictions of the sources' 2D position

[1] Nakadai et al., "Active Audition for Humanoid," in *AAAI/IAAI*, 2000.

[2] Nguyen et al., "Long-Term Robot Motion Planning for Active Sound Source Localization with Monte Carlo Tree Search," in *IEEE/HSCMA*, 2017.

[3] Bustamante et al., "Towards Information-Based Feedback Control for Binaural Active Localization," in *ICASSP*, 2016.

Active SSL

[4]Evers et al., “The LOCATA Challenge: Acoustic Source Localization and Tracking,” *IEEE/TASLP*, 2020.

Active SSL

Motivation:

- Real-world robotics scenarios are often dynamic
- Static SSL frameworks struggle predicting the source-array distance

Intuition:

- Aggregate instantaneous angular estimates over time
- Leverage the robot movement to refine the predictions of the sources' 2D position

Literature:

- Several works in the Robotics literature [1][2][3]

---

[1] Nakadai et al., "Active Audition for Humanoid," in *AAAI/IAAI*, 2000.

[2] Nguyen et al., "Long-Term Robot Motion Planning for Active Sound Source Localization with Monte Carlo Tree Search," in *IEEE/HSCMA*, 2017.

[3] Bustamante et al., "Towards Information-Based Feedback Control for Binaural Active Localization," in *ICASSP*, 2016.

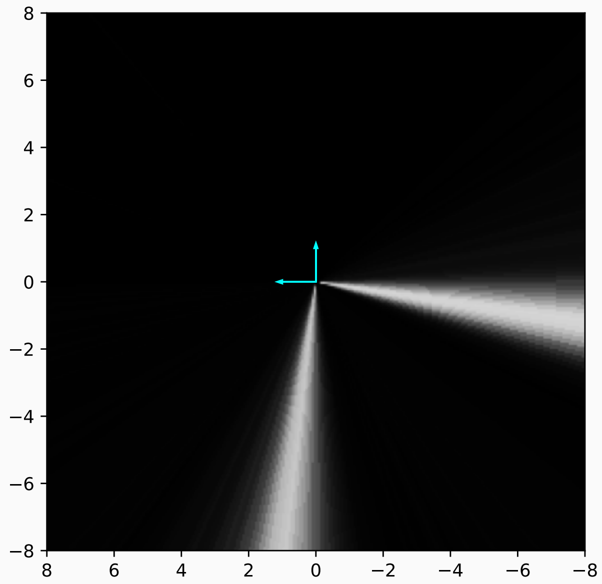
Active SSL

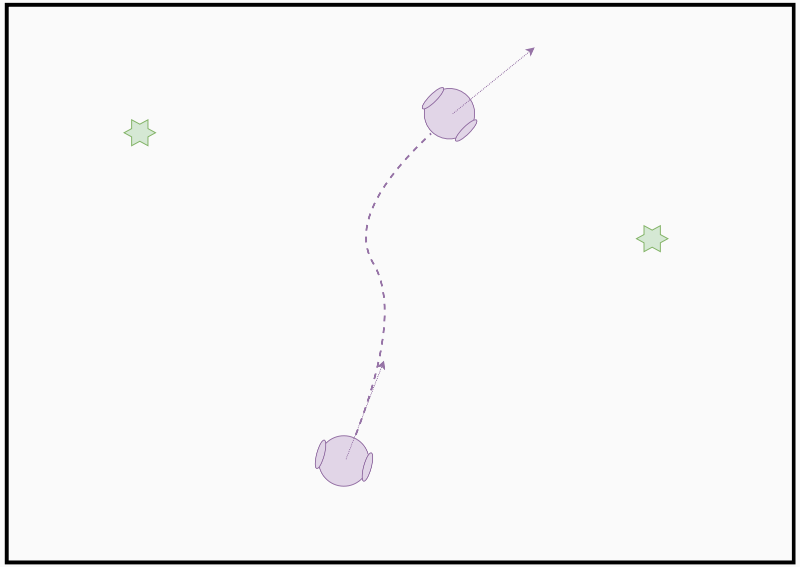
- Lack of deep-learning-based methods
  Multiple works involving moving sources (e.g. LOCATA challenge [4]), but only few considering mobile microphones

[4]Evers et al., “The LOCATA Challenge: Acoustic Source Localization and Tracking,” *IEEE/TASLP*, 2020.

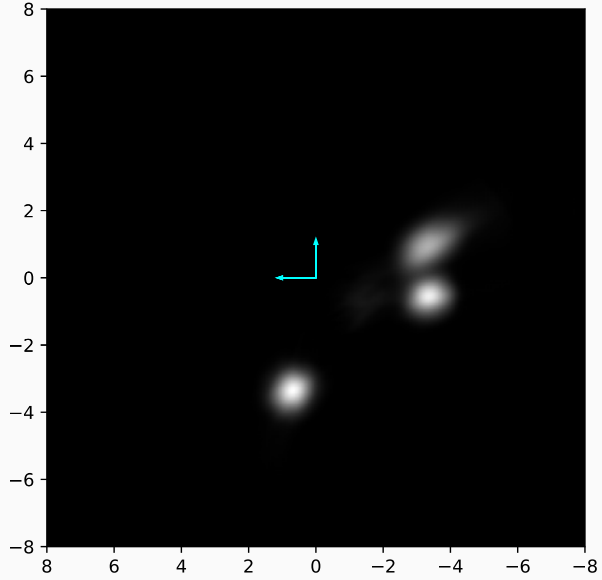
Proposed Approach

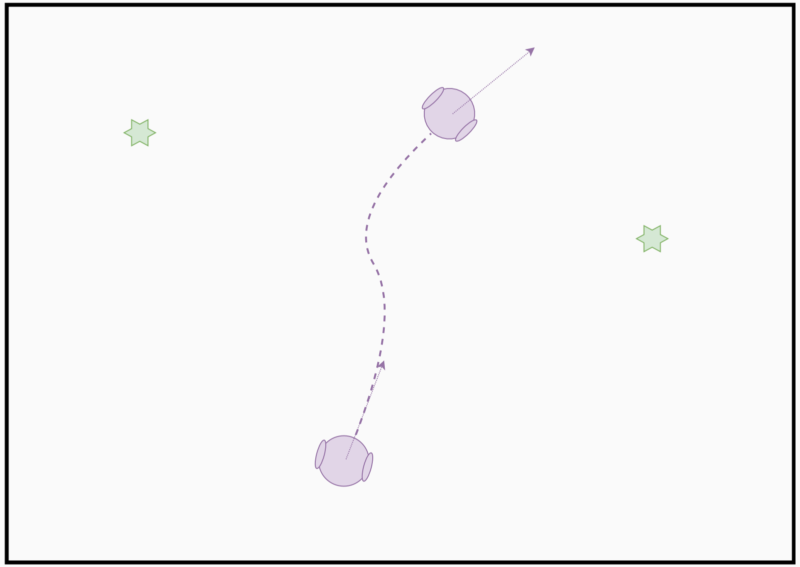
- Discrete step process
- Project static SSL predictions to a 2D egocentric view
- Aggregate these maps into a single final heatmap





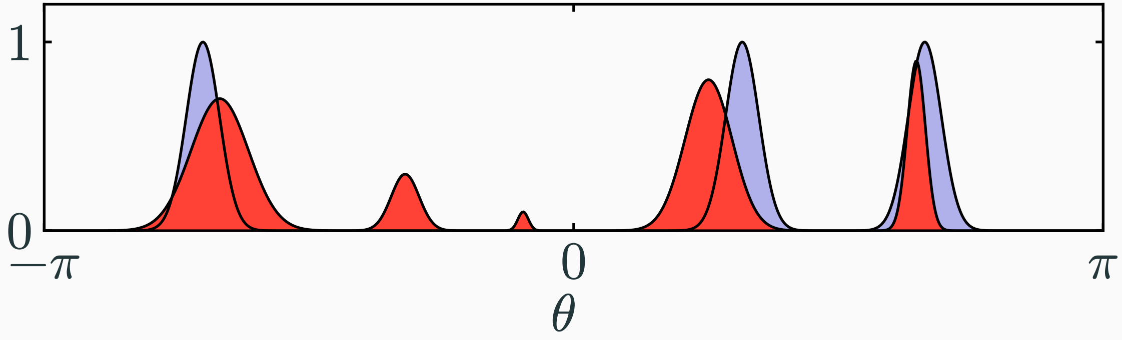
- Discrete step process
- Project static SSL predictions to a 2D egocentric view
- Aggregate these maps into a single final heatmap





Static SSL Model (1/2): DoA Spectrum Regression

- Encode DoA values over $[-\pi, \pi]$ (discretized) [1]
- Can represent an arbitrary number of sources
- Ground-truth DoA values are represented with Gaussians

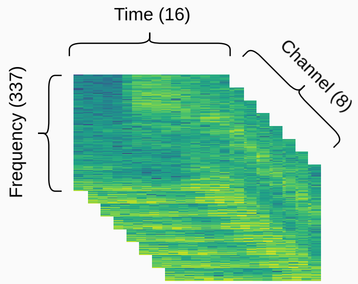


- Thanks to this representation, the SSL task becomes a DoA spectrum regression:

$$\mathcal{L} = \|\hat{o} - o\|_2^2$$

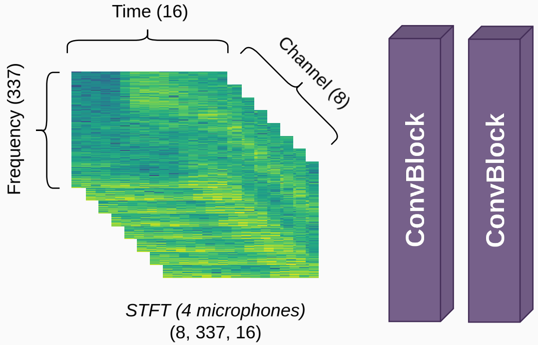
[1] He et al., "Neural Network Adaptation and Data Augmentation for Multi-Speaker Direction-of-Arrival Estimation," *IEEE/TASLP*, 2021.

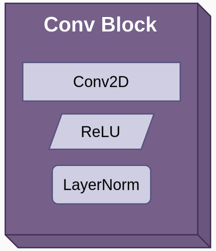
Static SSL Model (2/2): Network Architecture



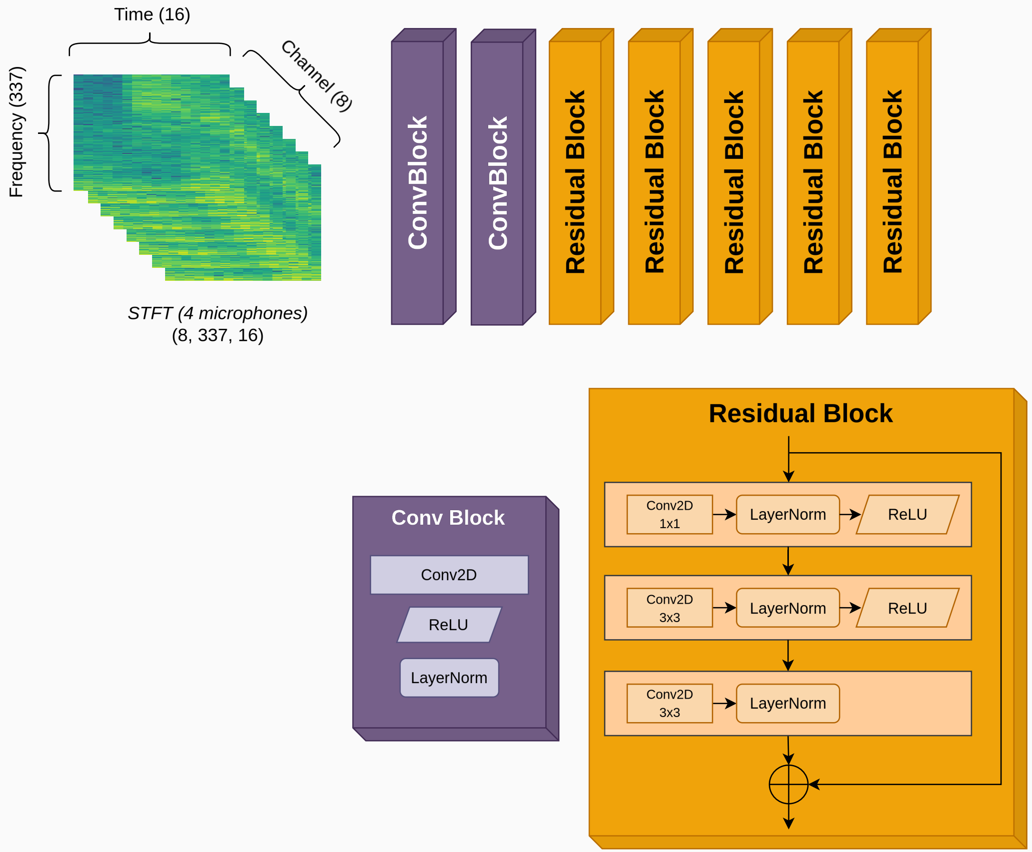
STFT (4 microphones)
(8, 337, 16)

Static SSL Model (2/2): Network Architecture

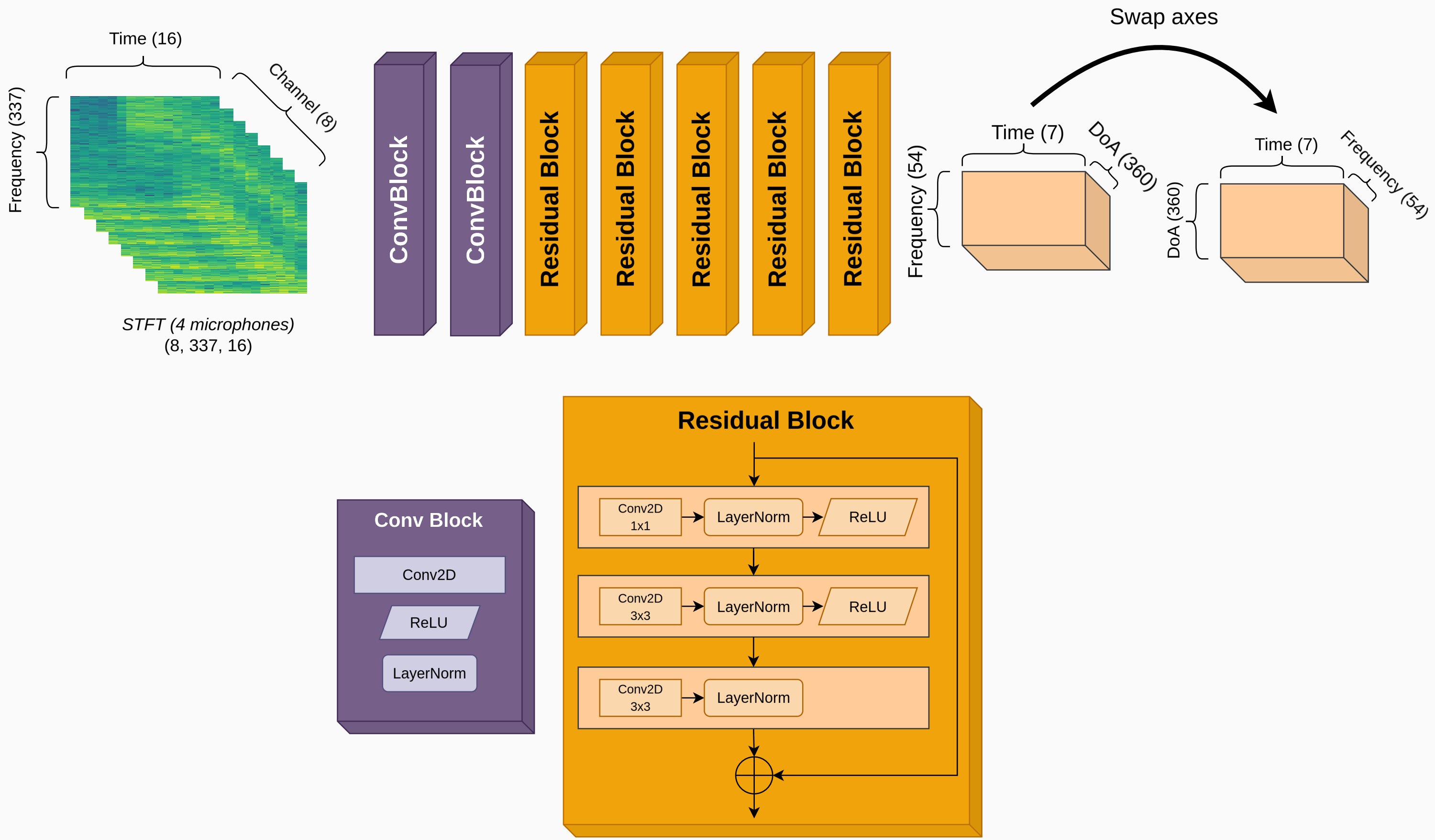


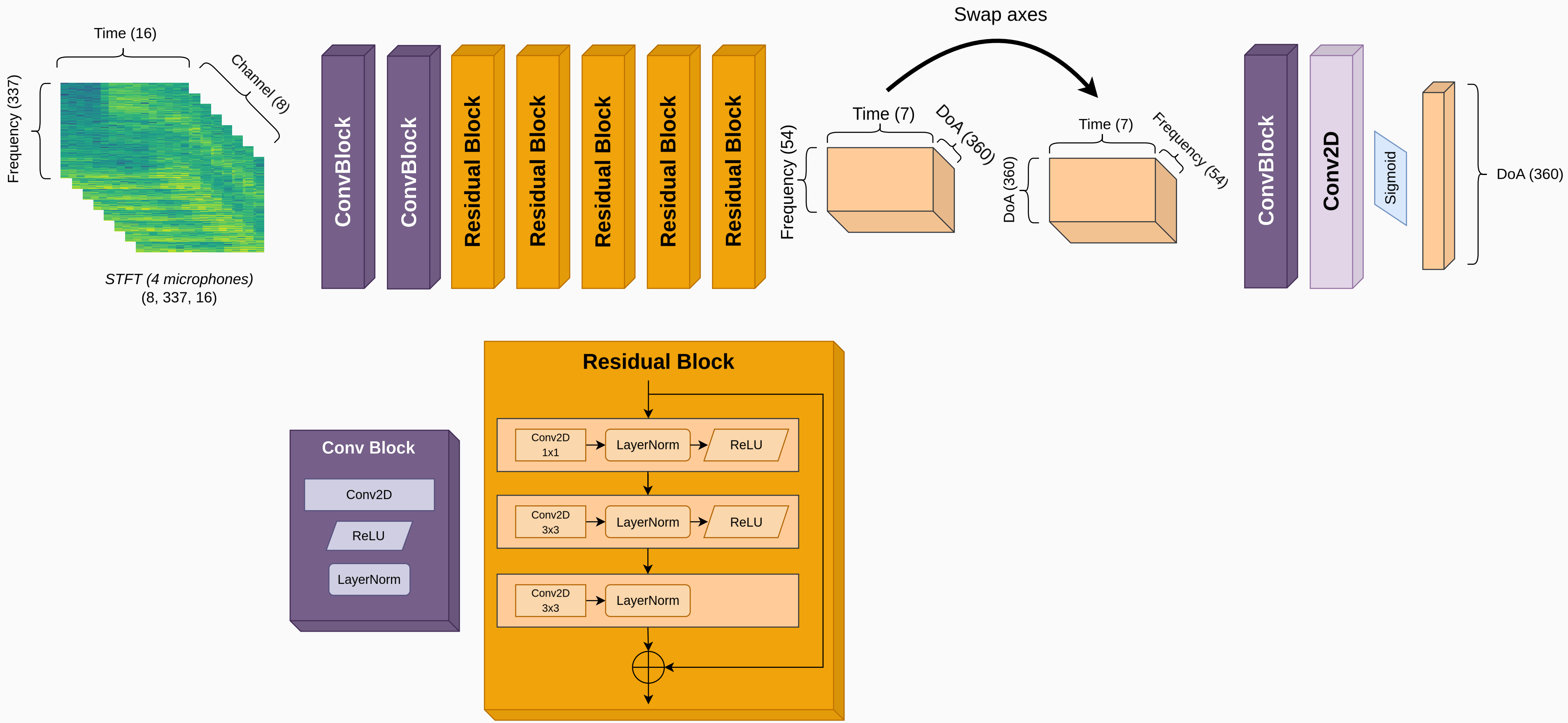


Static SSL Model (2/2): Network Architecture

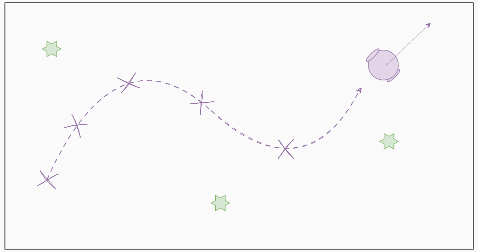


Static SSL Model (2/2): Network Architecture

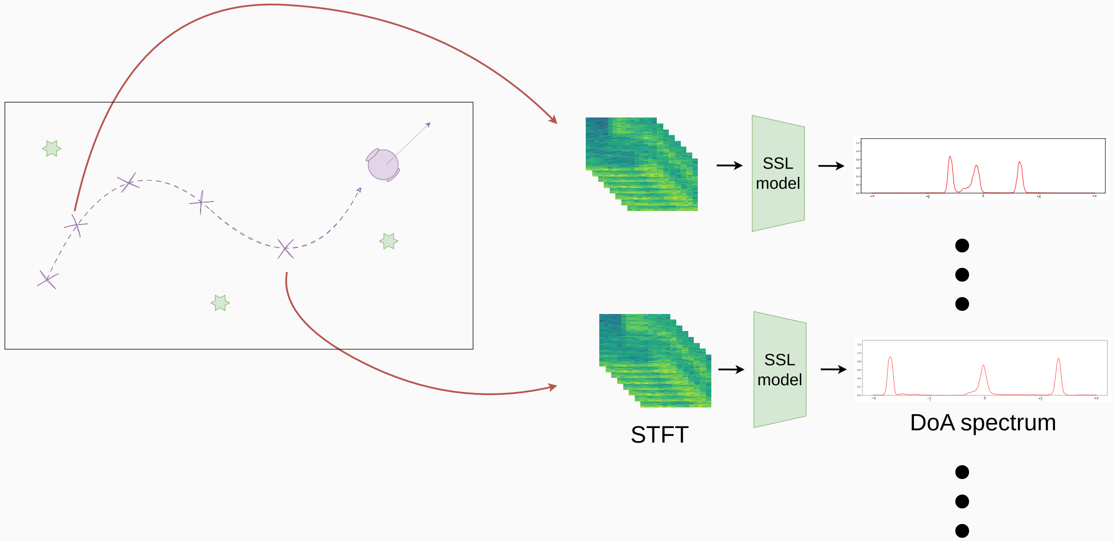



Time (16)
Channel (8)
Frequency (337)
STFT (4 microphones)
(8, 337, 16)
ConvBlock
ConvBlock
Residual Block
Residual Block
Residual Block
Residual Block
Residual Block
Swap axes
Time (7)
DoA (360)
Frequency (54)
Time (7)
Frequency (54)
DoA (360)
ConvBlock
Conv2D
Sigmoid
DoA (360)
Conv Block
Conv2D
ReLU
LayerNorm
Residual Block
Conv2D
1x1
LayerNorm
ReLU
Conv2D
3x3
LayerNorm
ReLU
Conv2D
3x3
LayerNorm

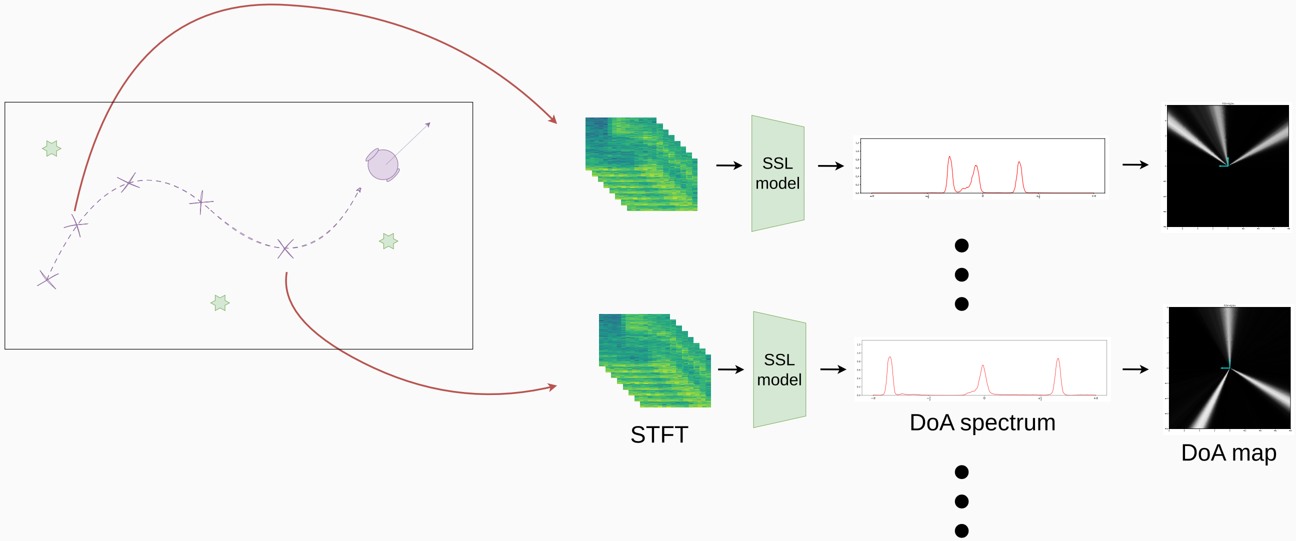
Active Sound Source Localization Pipeline



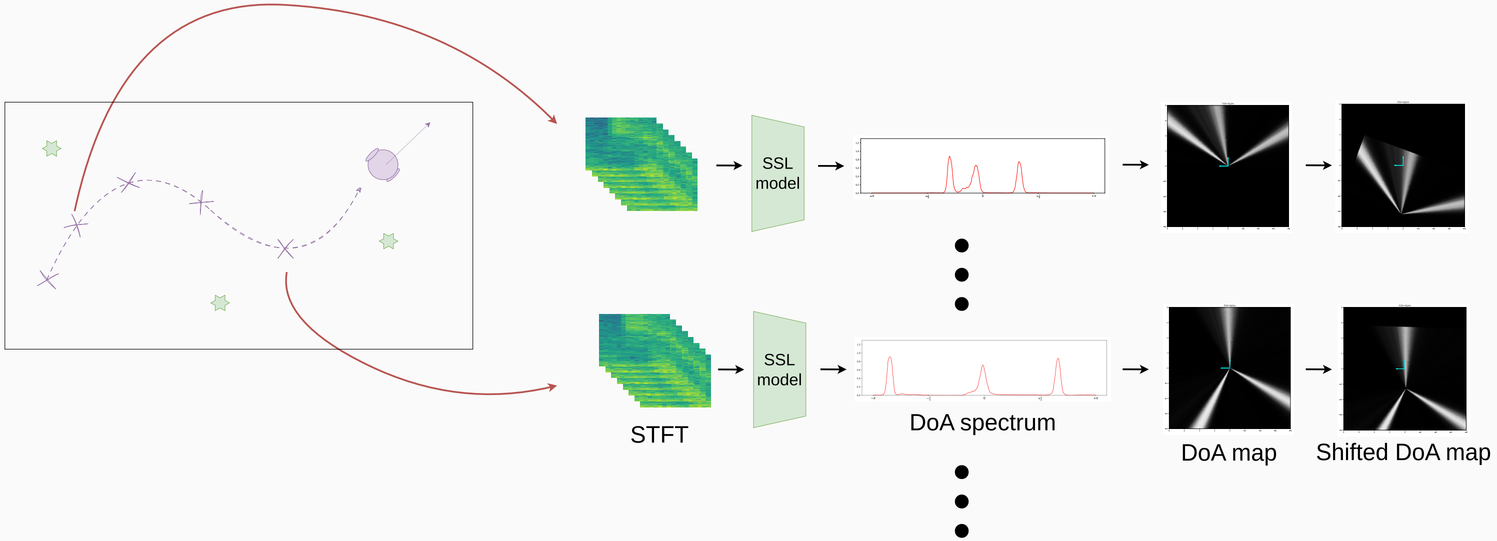
Active Sound Source Localization Pipeline



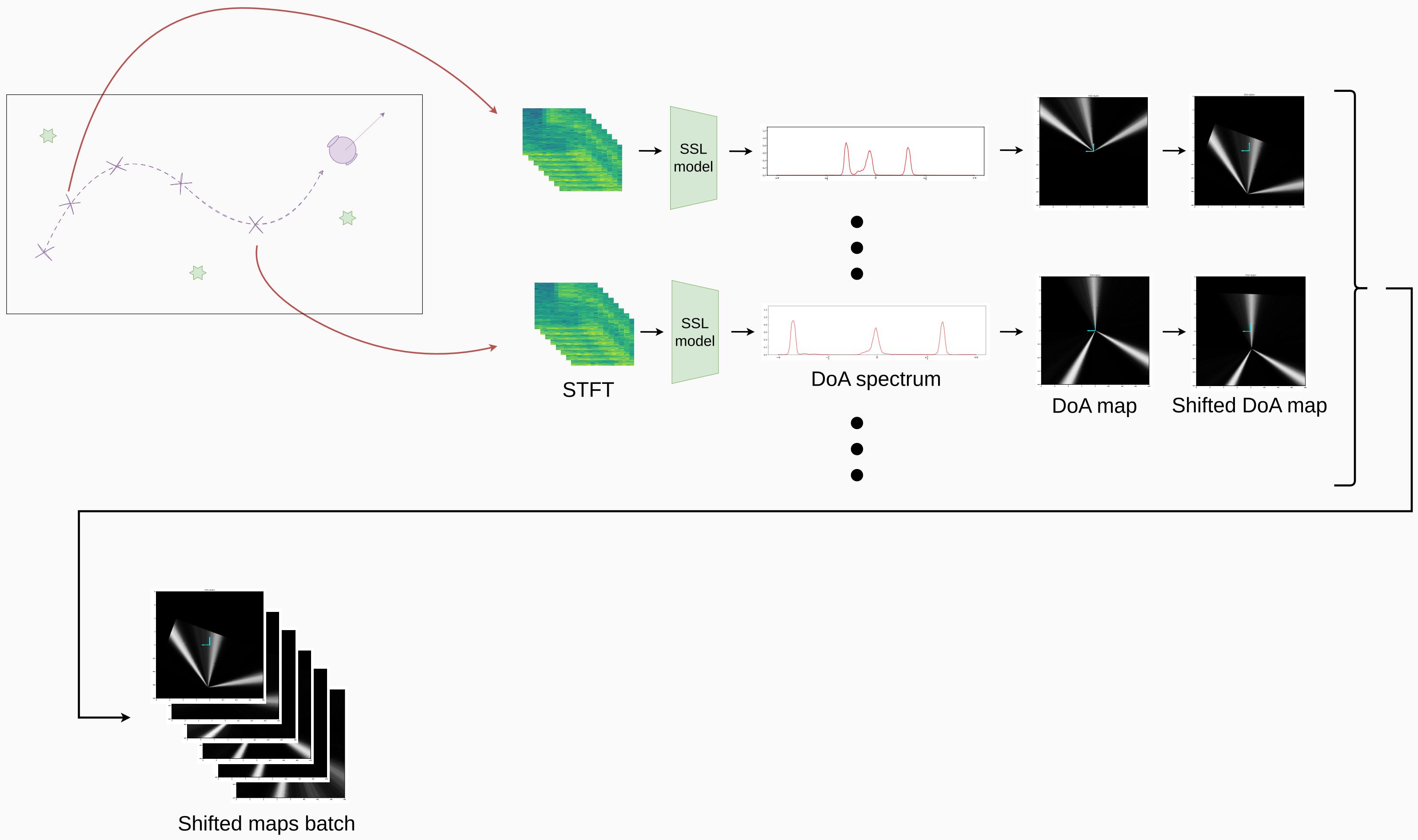
Active Sound Source Localization Pipeline



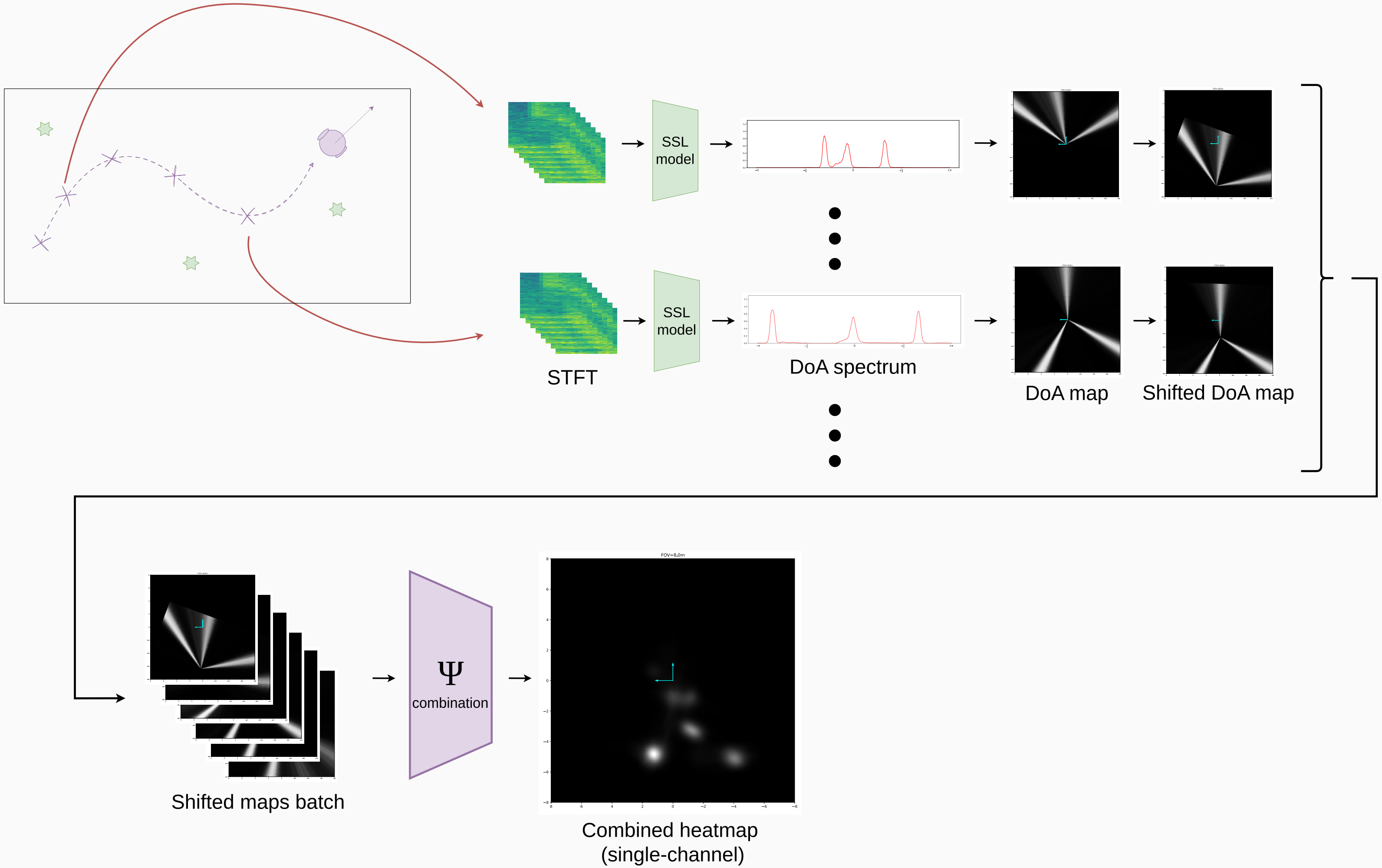
Active Sound Source Localization Pipeline



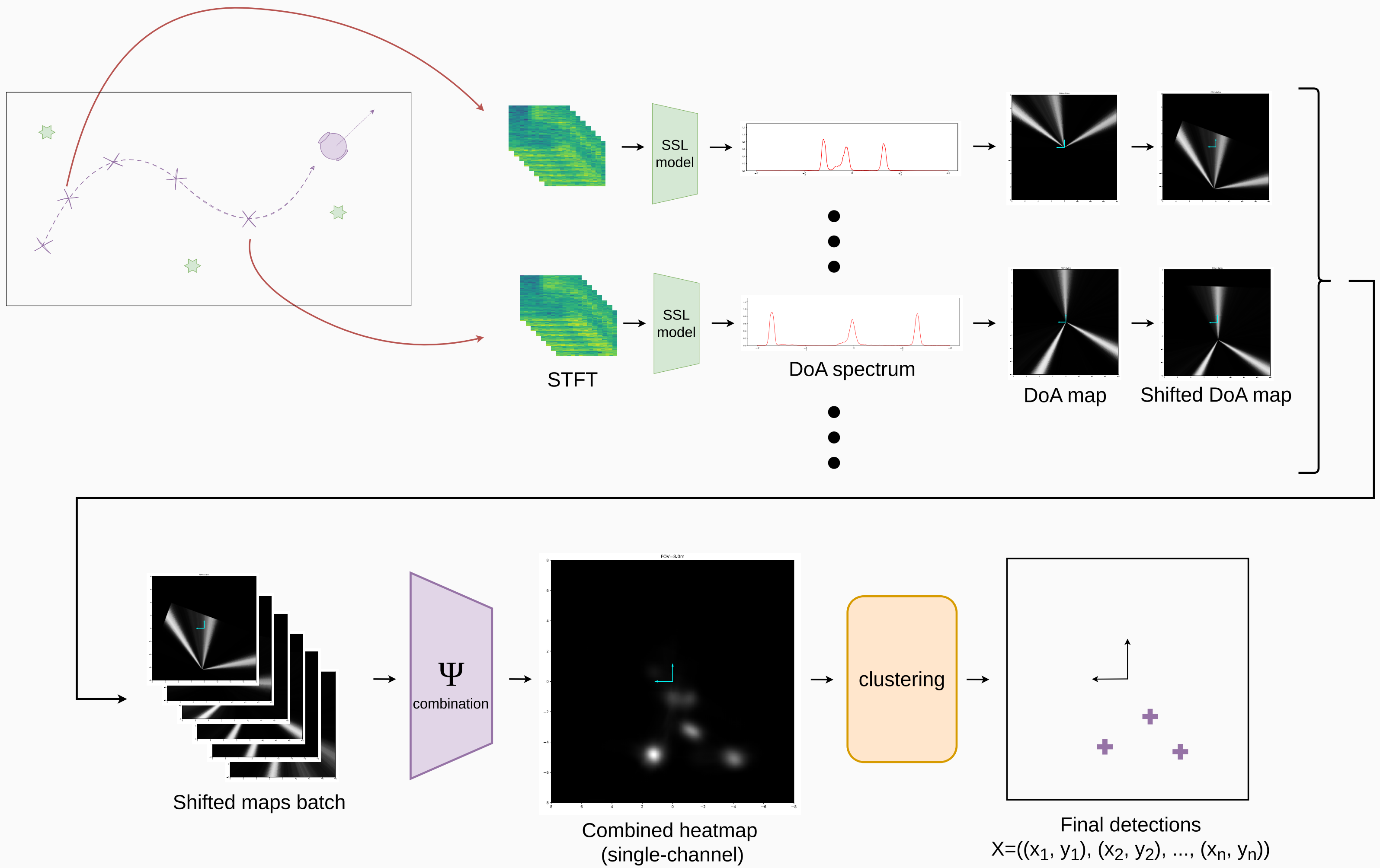
Active Sound Source Localization Pipeline



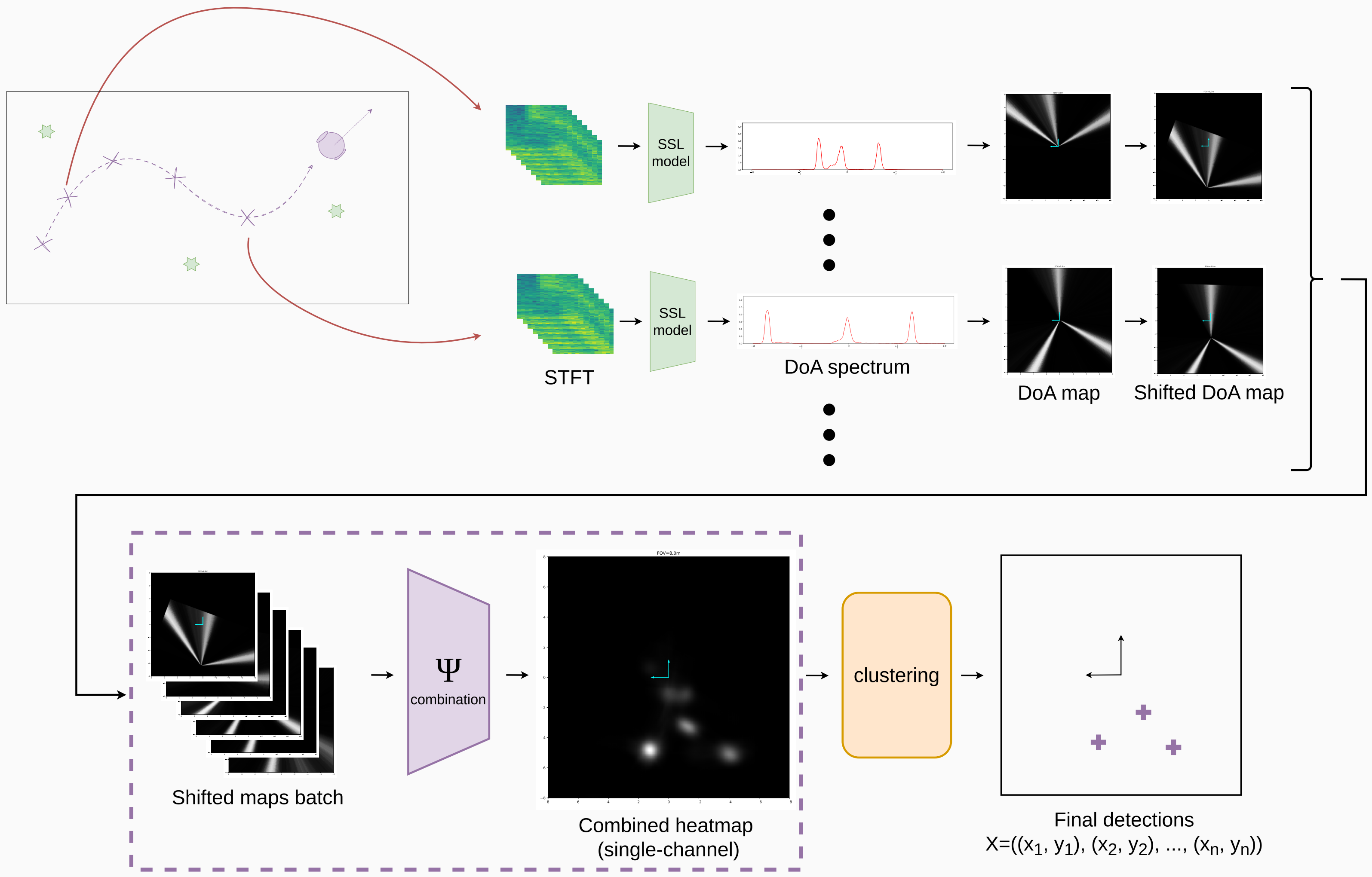
Active Sound Source Localization Pipeline



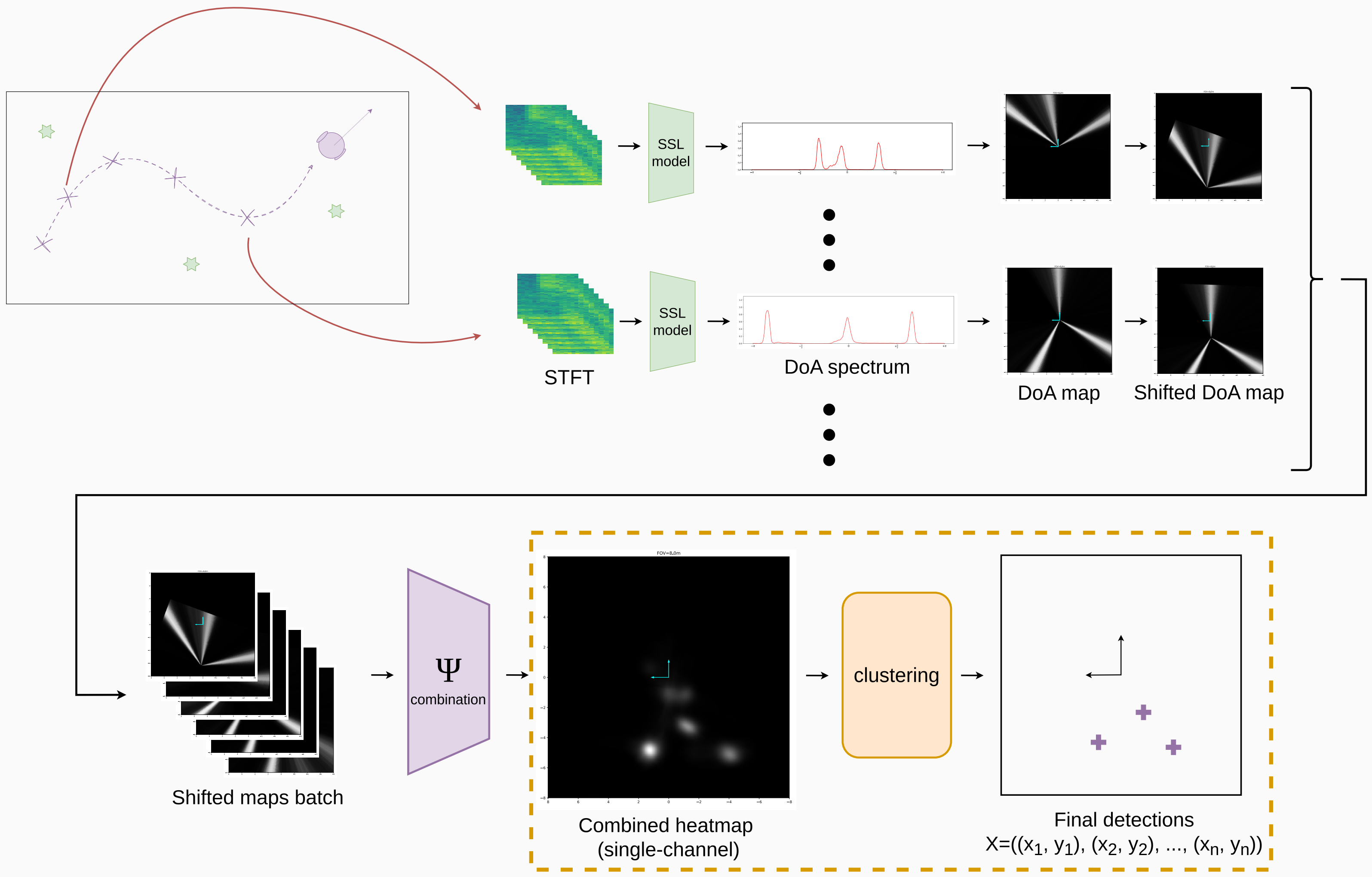
Active Sound Source Localization Pipeline



Active Sound Source Localization Pipeline



Active Sound Source Localization Pipeline



Aggregation strategy

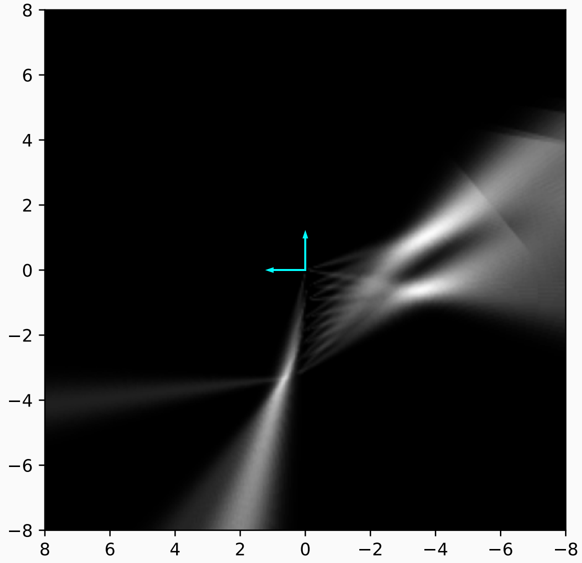
Aggregate shifted maps into a single heatmap

Aggregate shifted maps into a single heatmap

Two methods were explored:

- Naive averaging:

$$\widehat{M}_t = \frac{1}{H} \sum_{t'=0}^{H-1} M_{t-t'}$$



Averaging

Aggregate shifted maps into a single heatmap

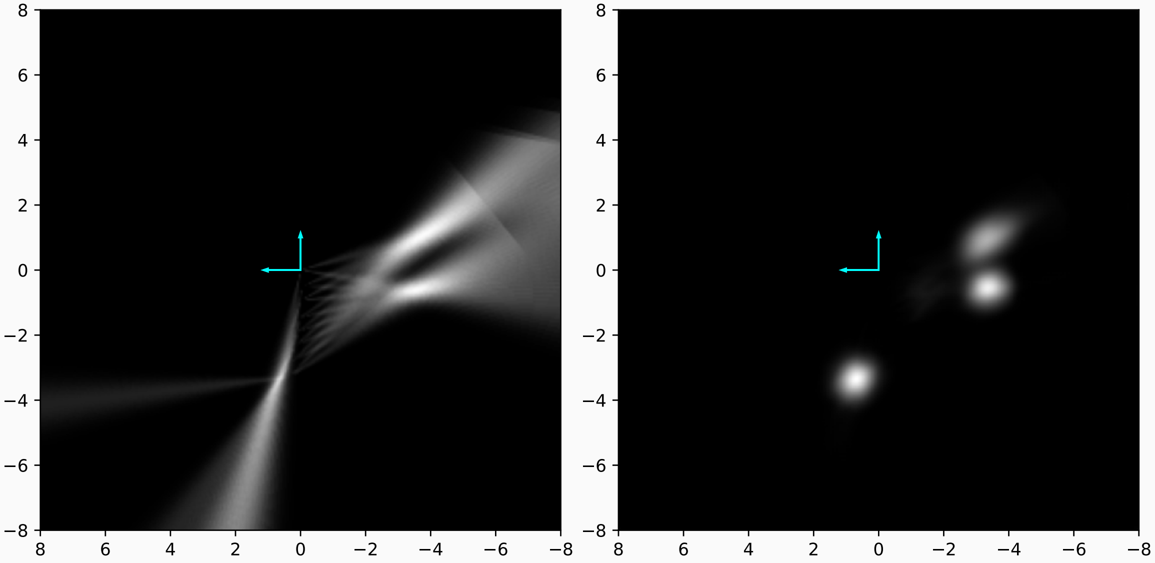
Two methods were explored:

- Naive averaging:

$$\widehat{M}_t = \frac{1}{H} \sum_{t'=0}^{H-1} M_{t-t'}$$

- U-Net model [1]:

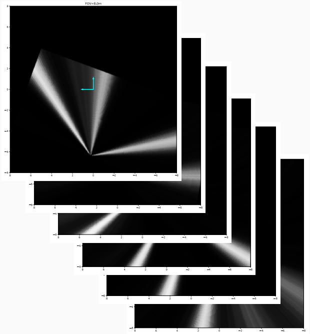
$$\widehat{M}_t = \Psi_{\mathrm{DNN}}(M_{t-H+1}, \ldots, M_t)$$



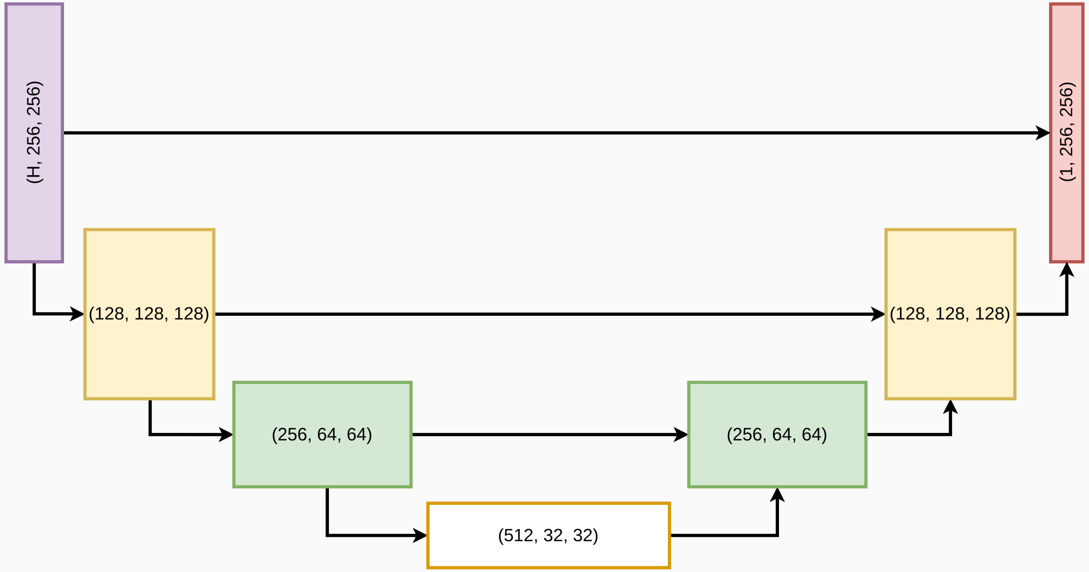
Averaging DNN

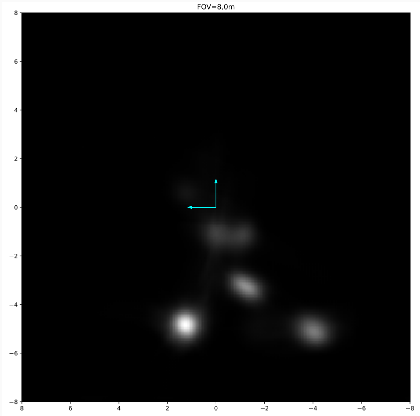
[1] Ronneberger et al., "U-net: Convolutional Networks for Biomedical Image Segmentation," in *MICCAI*, 2015.

Neural Network-Based Aggregation



Shifted maps batch



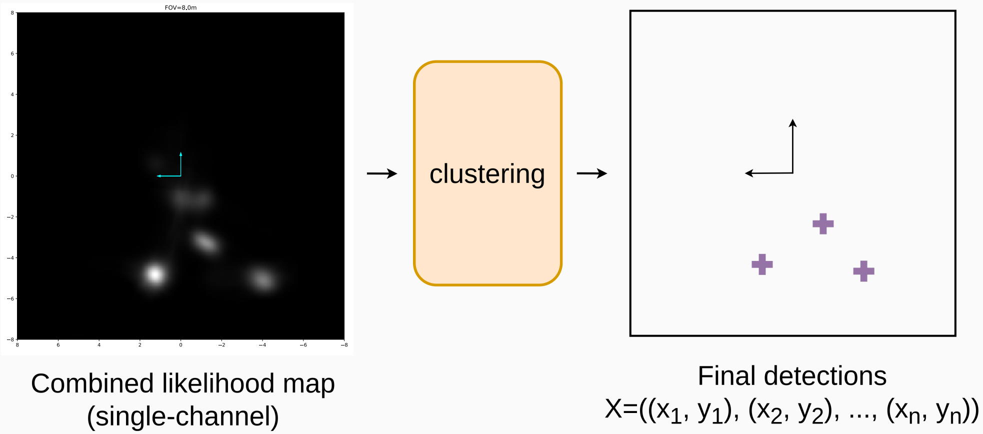


Combined likelihood map
(single-channel)

$$\mathcal{L} = \frac{1}{p^2} \| \mathcal{M}_t - \mathcal{M}_t^* \|_F^2$$

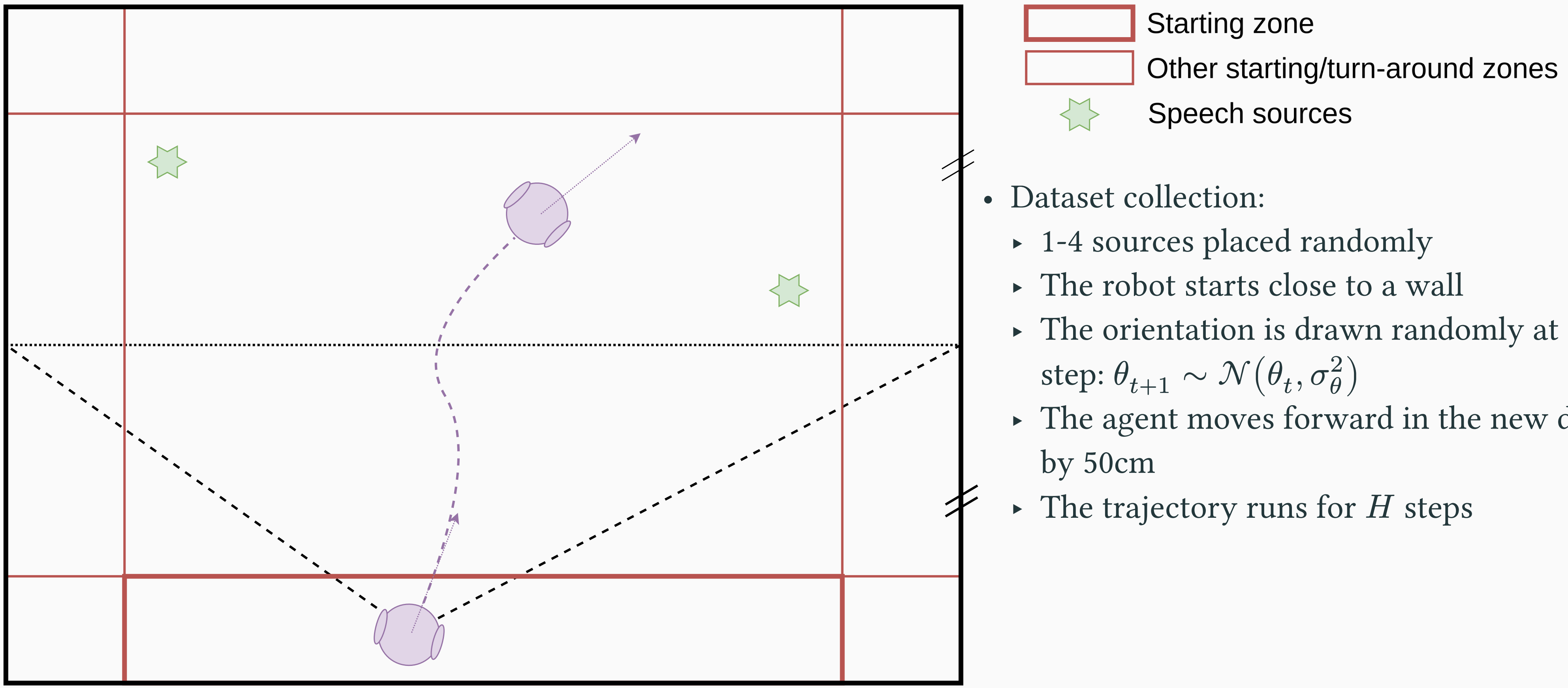
Extract discrete 2D position predictions from the heatmap

1. Low values are filtered out from the egocentric heatmap (threshold τ)
2. The DBSCAN algorithm [1] is used to cluster pixels into several groups
3. The position of the highest-value pixel of each cluster is used as the final detection



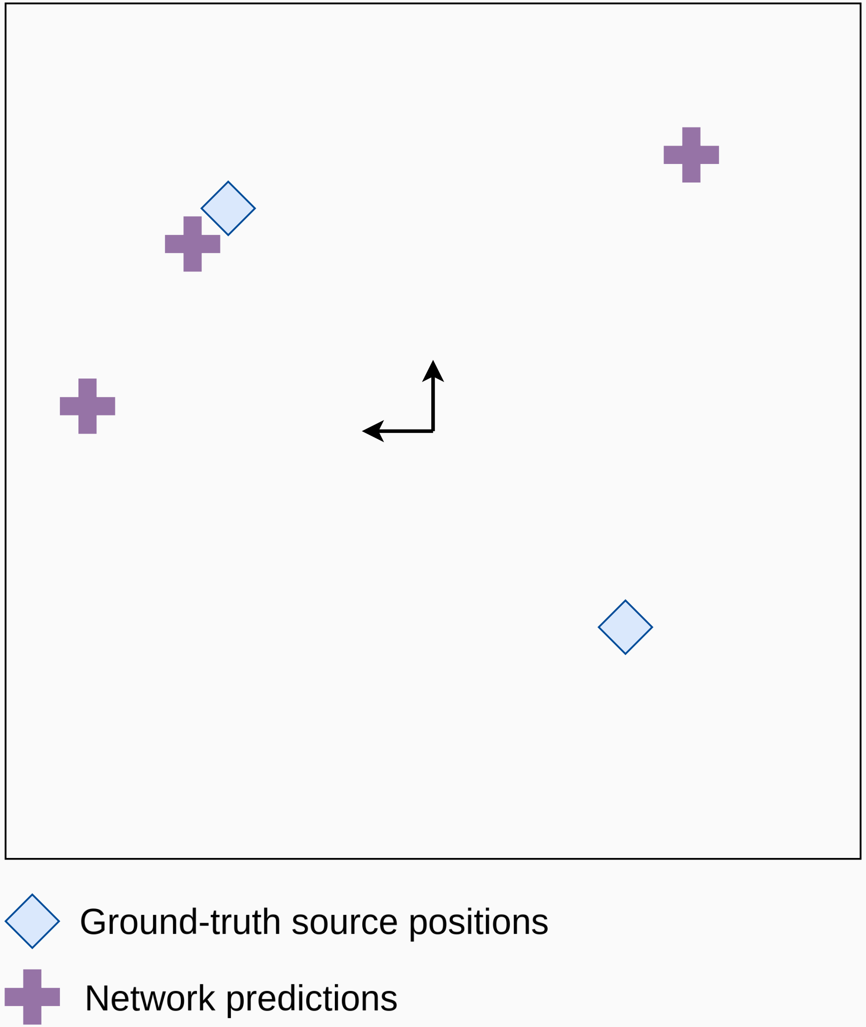
[1] Schubert et al., "DBSCAN Revisited, Revisited: Why and How You Should (Still) Use DBSCAN," *ACM TODS*, 2017.

Experimental Setup



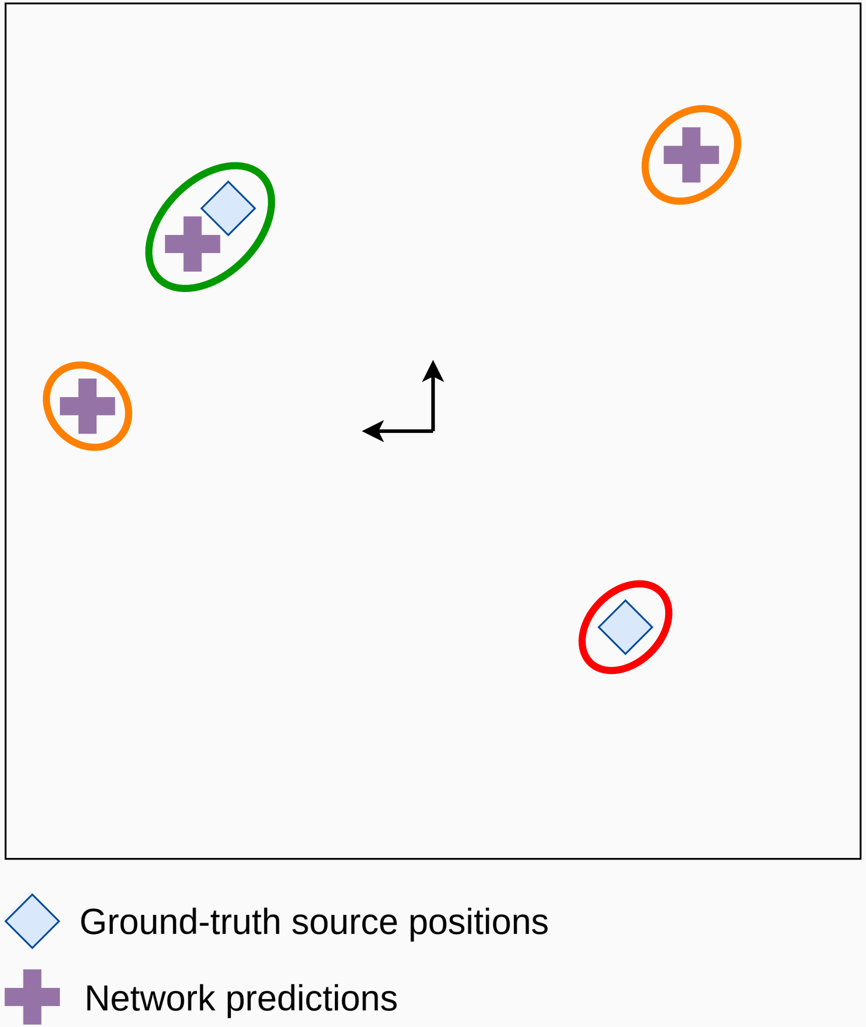
- Dataset collection:
  - 1-4 sources placed randomly
  - The robot starts close to a wall
  - The orientation is drawn randomly at each step: $\theta_{t+1} \sim \mathcal{N}\left(\theta_t, \sigma_\theta^2\right)$
  - The agent moves forward in the new direction by 50cm
  - The trajectory runs for H steps

Standard Evaluation Metrics



- Define a threshold δ for defining correct detections
- Match predictions and ground truths

Standard Evaluation Metrics

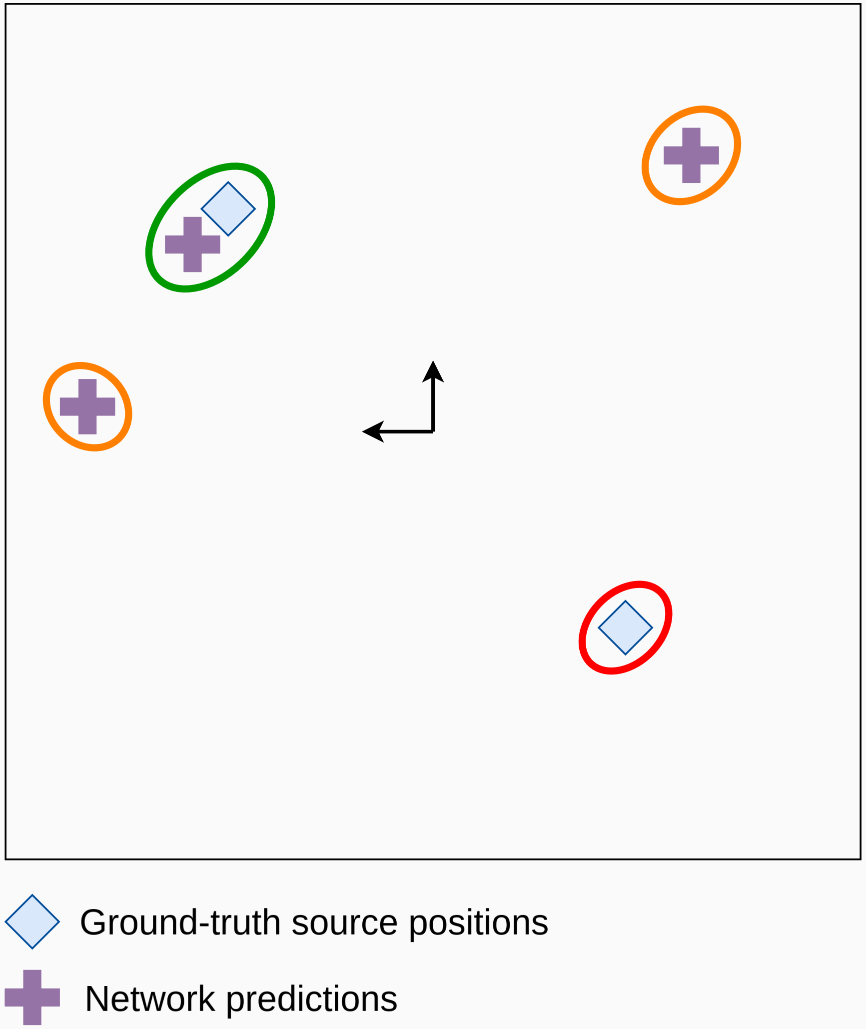


- Define a threshold δ for defining correct detections
- Match predictions and ground truths

$$\text{Precision} = \frac{\#\text{correct}}{\#\text{predictions}}$$

$$\text{Recall} = \frac{\#\text{correct}}{\#\text{sources}}$$

Standard Evaluation Metrics



- Define a threshold δ for defining correct detections
- Match predictions and ground truths

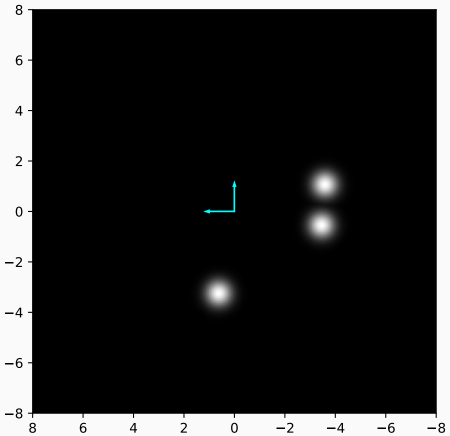
$$\text{Precision} = \frac{\#\text{correct}}{\#\text{predictions}}$$

$$\text{Recall} = \frac{\#\text{correct}}{\#\text{sources}}$$

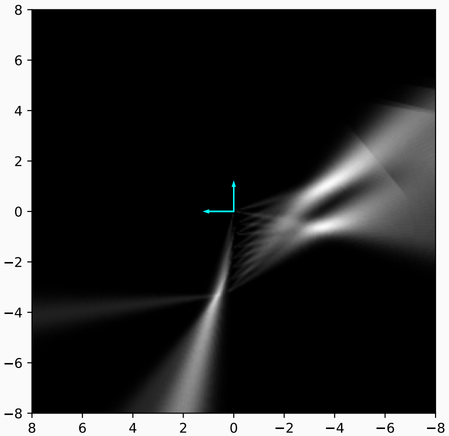
In this example:

- $\text{Precision} = \frac{1}{3} \approx 33\%$
- $\text{Recall} = \frac{1}{2} = 50\%$

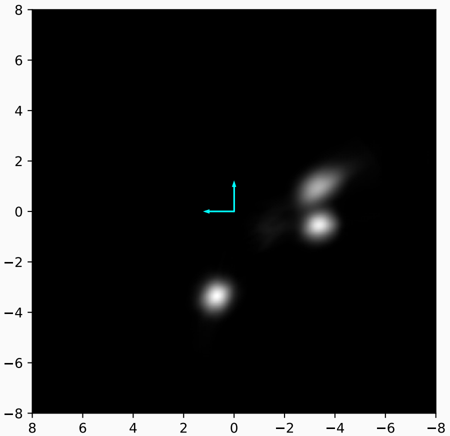
Comparison of Aggregation Methods



Ground truth

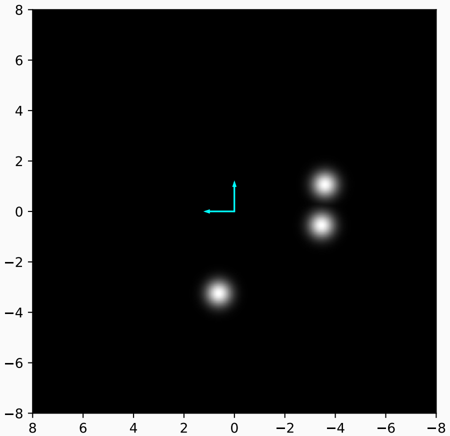


Average

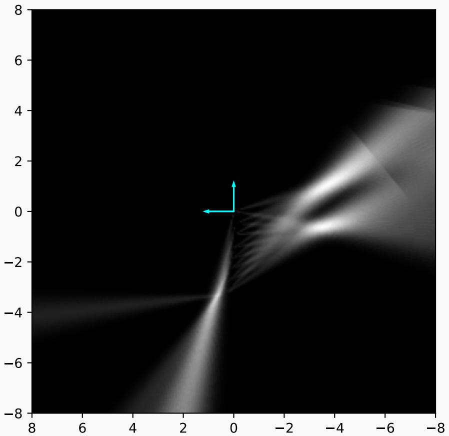


DNN

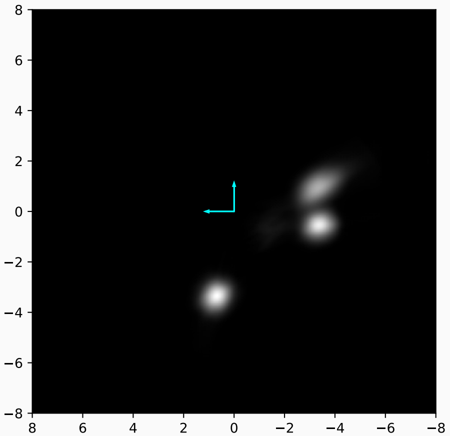
Comparison of Aggregation Methods



Ground truth



Average



DNN

Aggregation method	Estimated DoA spectrum $\hat{o}_t$		Ground truth DoA spectrum o_t	
	Precision (%) $\uparrow$	Recall (%) $\uparrow$	Precision (%) $\uparrow$	Recall (%) $\uparrow$
Average	72.33	46.60	96.02	77.70
$\Psi_{\mathrm{DNN}(\theta)}$	**86.05**	**53.28**	**99.74**	**90.54**

- **Complete pipeline** for active multi-source localization
- **Aggregation of information accross time** to build fine 2D position estimates
- Leveraging of a **static SSL deep-learning model**
- **Deep U-Net style architecture** for combining heatmaps
- Training of the **static SSL model** and the **U-Net blender** using synthetic datasets generated from our simulator

Plan

Motivation & problem statement

Goal: Perceptually motivated navigation [1]

- Robots are expected to *understand* human speech
- *Automatic Speech Recognition (ASR)* is the first step of the speech understanding pipeline
- How can navigation help with improving the robot's ASR performance?

[1] Majumder et al., "Move2hear: Active Audio-Visual Source Separation," in *ICCV*, 2021.

Measuring ASR performance

The *Word Error Rate (WER)* measures the ASR performance.

$\rightarrow$ *WER:* Minimum edit distance between two sentences:

Measuring ASR performance

The *Word Error Rate (WER)* measures the ASR performance.

→ *WER:* Minimum edit distance between two sentences:

$$\text{WER} = \frac{\#\text{substitutions} + \#\text{deletions} + \#\text{insertions}}{\#\text{words}}$$

The *Word Error Rate (WER)* measures the ASR performance.

$\rightarrow$ *WER:* Minimum edit distance between two sentences:

$$\mathrm{WER} = \frac{\#\text{substitutions} + \#\text{deletions} + \#\text{insertions}}{\#\text{words}}$$

Example:

- Reference: *Obviously, he was __ able to catch the last bus on time today.*
- Prediction: *Obviously, he was not able to catch the past bus on time ____ .*

The *Word Error Rate (WER)* measures the ASR performance.

→ *WER:* Minimum edit distance between two sentences:

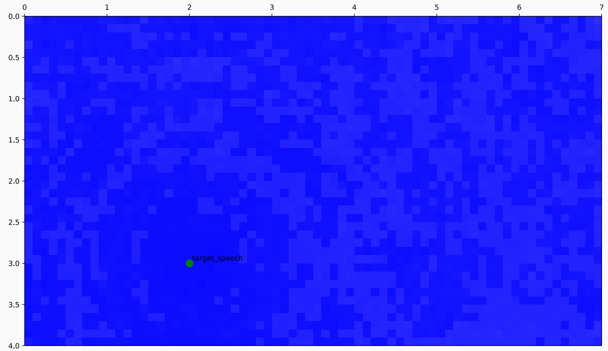
$$\text{WER} = \frac{\#\text{substitutions} + \#\text{deletions} + \#\text{insertions}}{\#\text{words}}$$

Example:

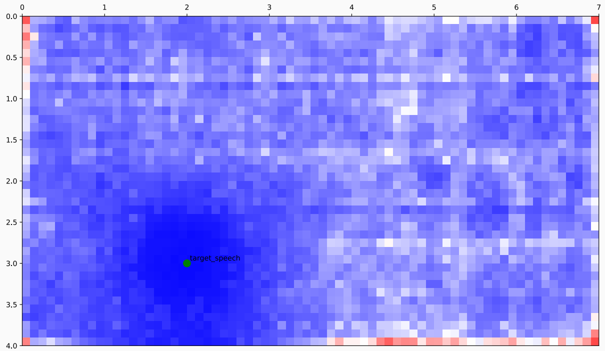
- Reference: *Obviously, he was __ able to catch the last bus on time today.*
- Prediction: *Obviously, he was not able to catch the past bus on time ____ .*

$$\text{WER} = \frac{1+1+1}{12} = 0.25$$

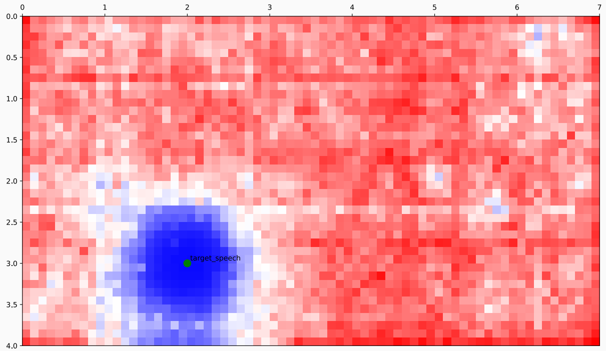
Reverberation impact on WER



(a) $T_{60} = 200\text{ms}$



(b) $T_{60} = 500\text{ms}$



(c) $T_{60} = 800\text{ms}$

- WER increases as reverberation grows
- Robot positioning impacts ASR performance
- Correct positioning matters more as T_{60} increases

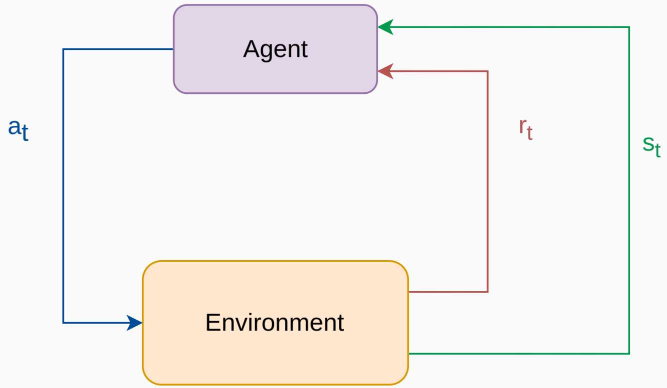
Problem Statement

Idea: Frame the navigation problem as a sequential decision problem

- At each step, the robot records a short audio snippet;
- Based on this observation, it decides what its next move should be;
- The environment rewards the robot based on a WER estimate for its current position;

$\rightarrow$ Reinforcement learning is very well suited to this problem.

RL [1] solves sequential decision problems, formalized as **Markov Decision Processes (MDPs)** [2].



At each step:

- The agent senses the environment by observing the state s_t in the state space $\mathcal{S}$
- It chooses an action a_t in the action set $\mathcal{A}$
- It receives a reward r_t

The goal is to maximize the cumulated discounted reward:

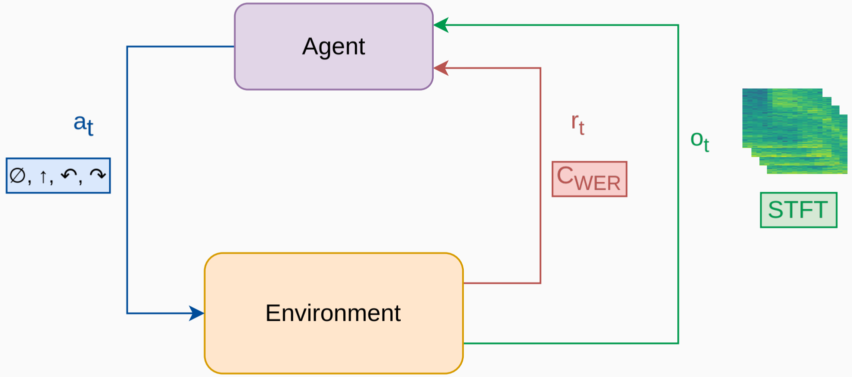
$$\max_{\pi} \mathbb{E}_{\pi} \left[\sum_{t=0}^{\infty} \gamma^t r_{t+1} \right]$$

[1] Sutton et al., *Reinforcement Learning: An Introduction.* MIT press Cambridge, 1998.
[2] R. Bellman, "A Markovian Decision Process," *Journal of mathematics and mechanics*, 1957.

Proposed Environment Formulation

Our environment is only **partially observable**.

- **State space:** possible agent positions in the room: $\mathcal{S} \subset \mathbb{R}^2 \times [0, 2\pi]$
- **Observation space:** Spectral representation of recorded audio: $\Omega \subset \mathbb{C}^{C \times F \times T}$
- **Action space:** $\mathcal{A} = \{\texttt{STAY}, \texttt{FORWARD}, \texttt{TURN\_LEFT}, \texttt{TURN\_RIGHT}\}$
- **Reward:** decreasing function of the WER:



$$r_t = \begin{cases} -\mu_W & \text{if the agent tries to hit a wall} \\ \mu_C \exp(-\xi_C C_t) - \mu_m \mathbb{1}(a_t = \texttt{FORWARD}) & \text{otherwise} \end{cases}$$

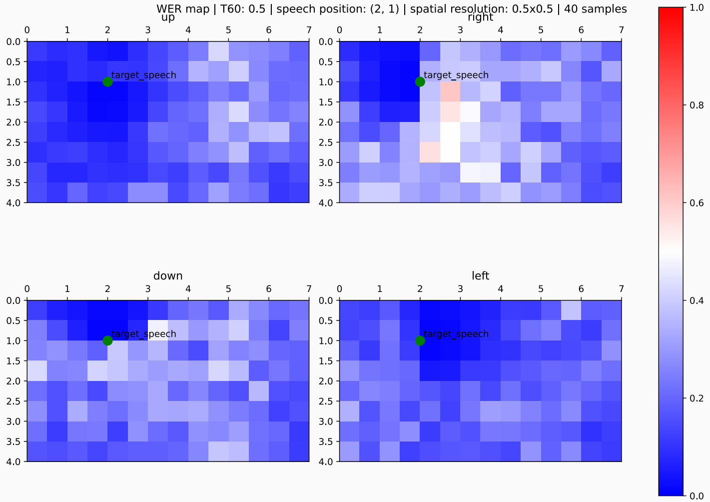
WER Cost Maps

- The cost of a state requires an estimate of the average WER for this position;
- The WER cost maps can be either **directional** or **omnidirectional**;

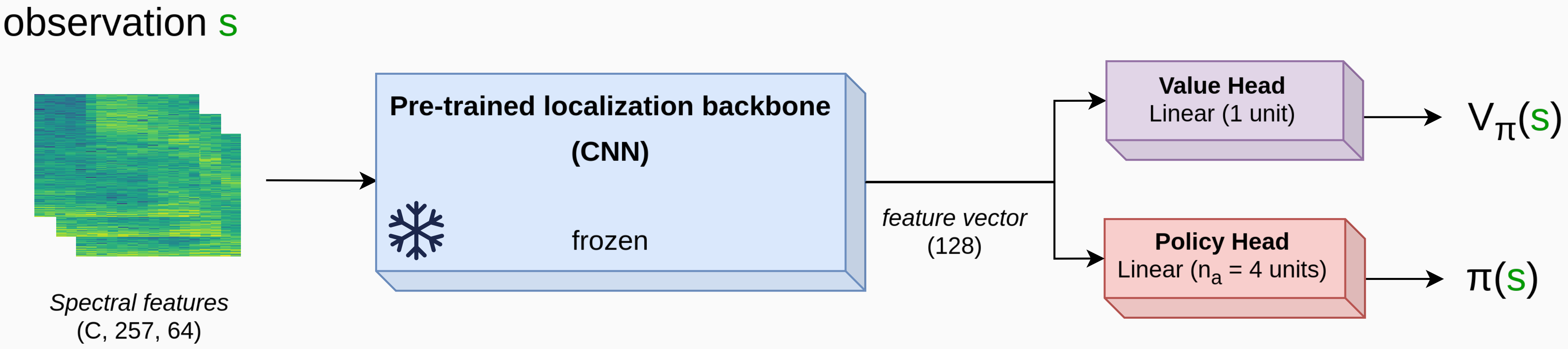
Problem: WER can't be computed at the environment run-time.

→ Pre-compute statistical estimates of the theoretical WER cost of a state.

$$C_{\mathrm{WER}}(\boldsymbol{x}_a, \alpha_a) = \mathbb{E}_{(v,t)\in\mathcal{D}}\left[\frac{1}{100}\,\mathrm{WER}\left(\underbrace{\mathrm{ASR}_\psi\left[\mathrm{listened}(v, \boldsymbol{x}_a, \alpha_a, \boldsymbol{x}_s)\right]}_{\text{predicted transcript } \hat{t}}, t\right)\right]$$



Agent Architecture



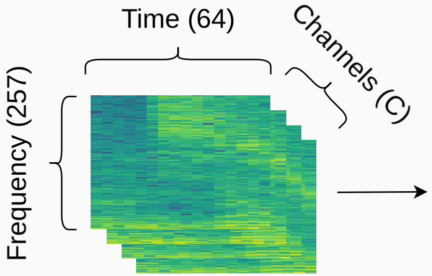
Two-stage training:

1. Train the backbone on a supervised single-source localization task
2. Train the value and policy heads with PPO [1]

---

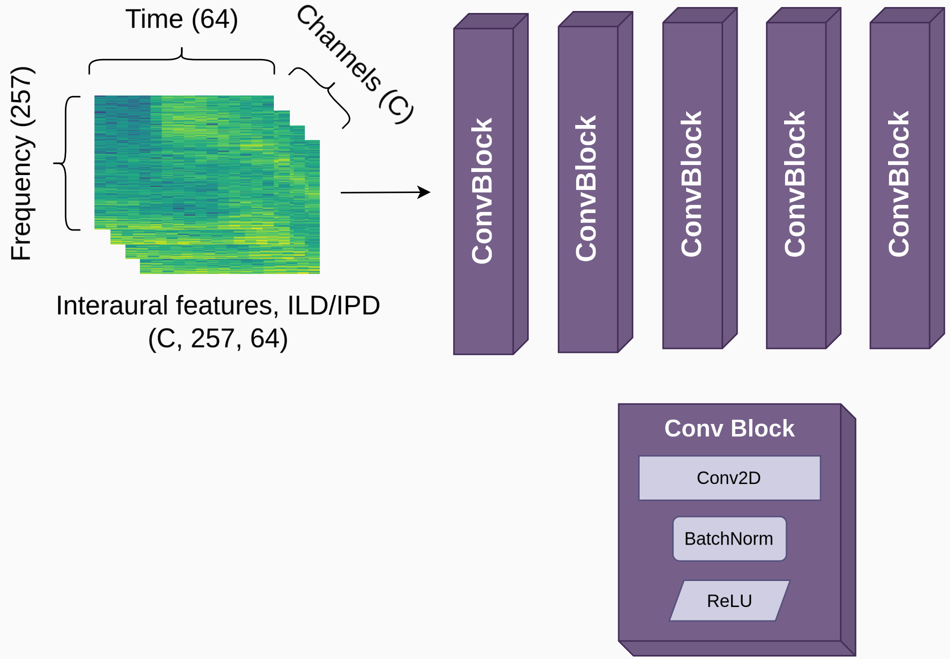
[1] Schulman et al., "Proximal Policy Optimization Algorithms," *arXiv preprint*, 2017.

SSL Backbone Architecture

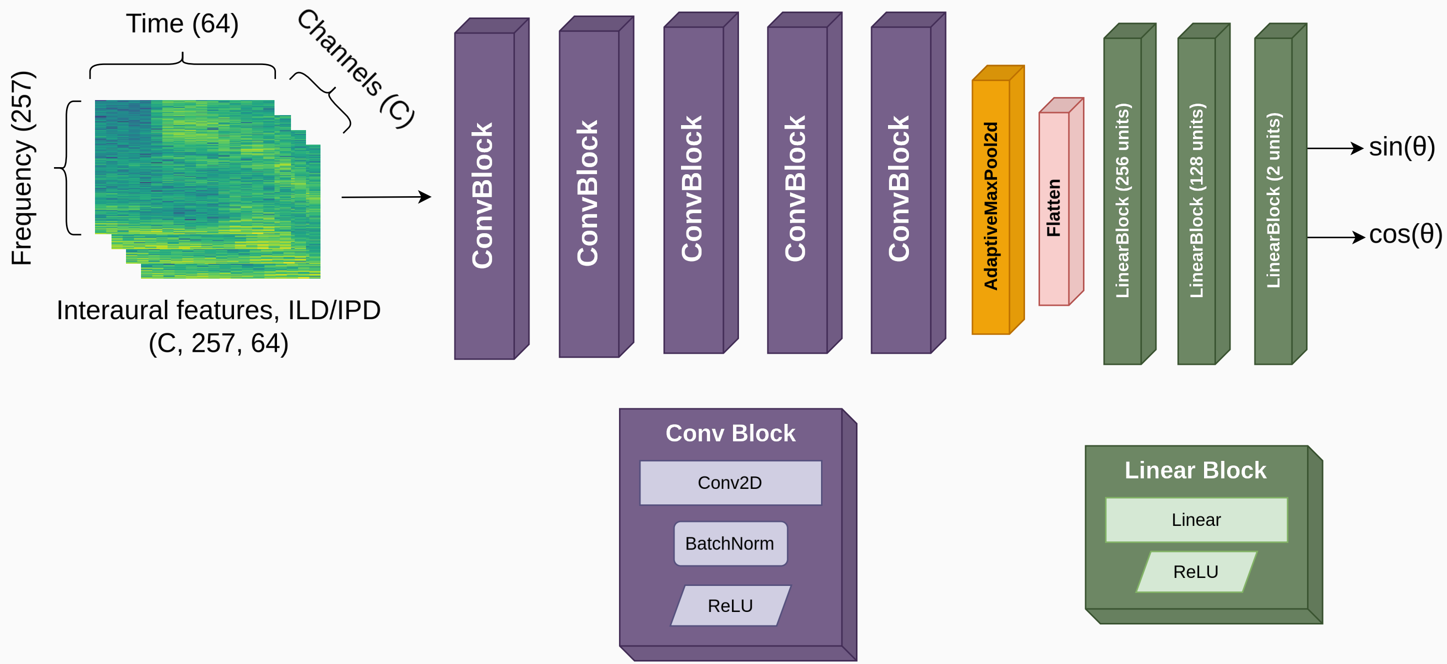


Interaural features, ILD/IPD
(C, 257, 64)

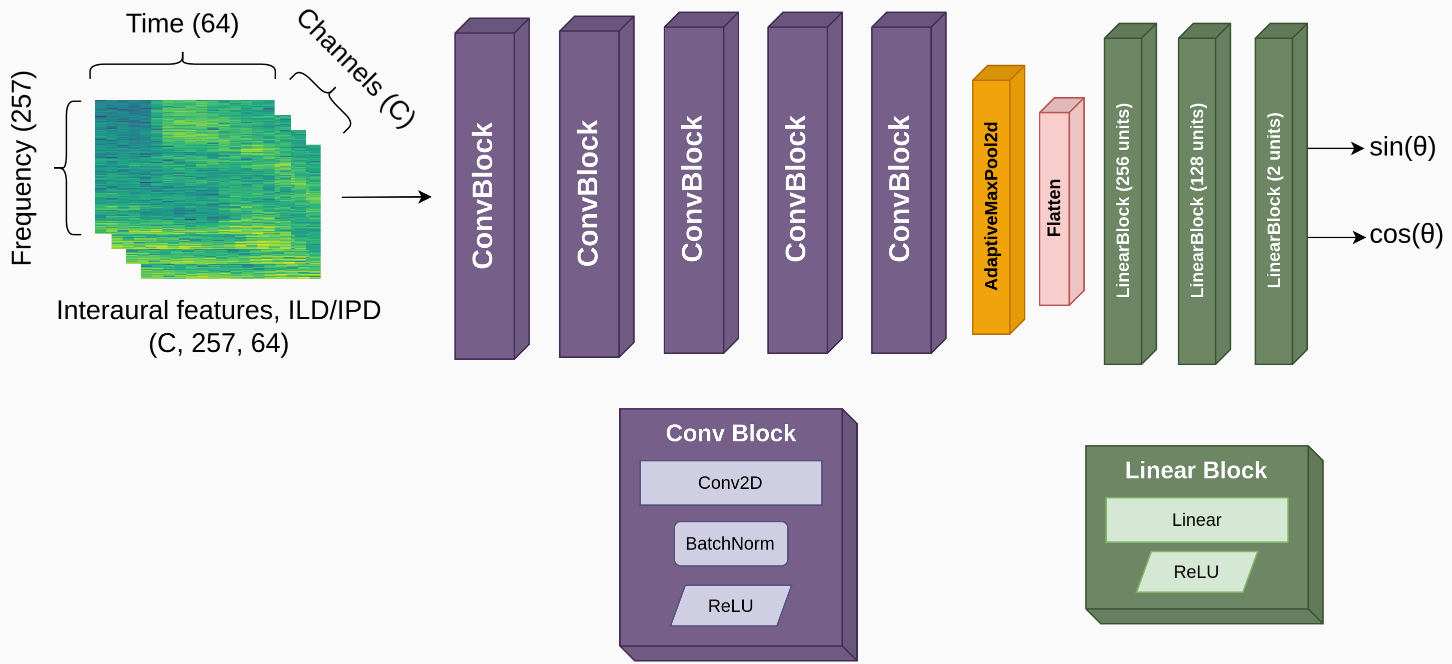
SSL Backbone Architecture



SSL Backbone Architecture



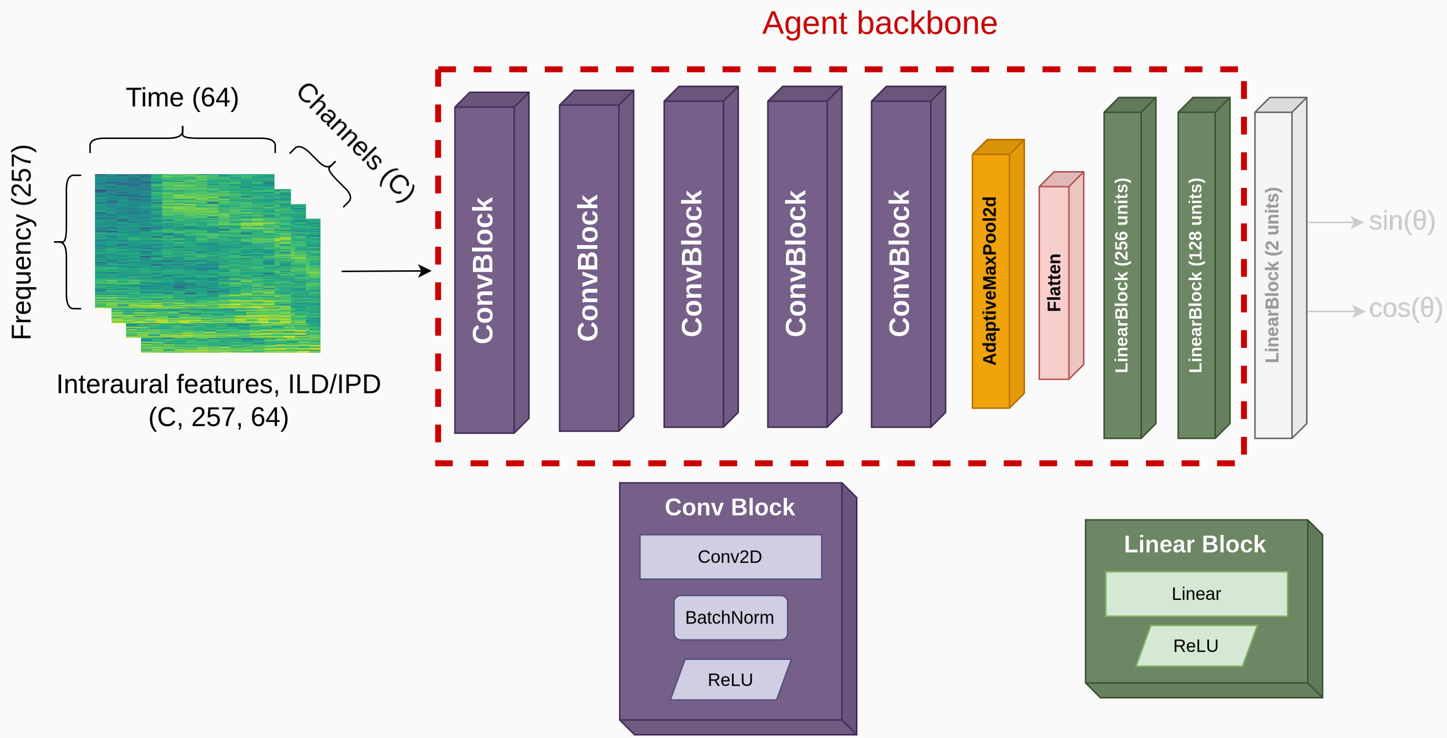
SSL Backbone Architecture



Training loss:

$$\mathcal{L}_{\mathrm{DoA}\left(\hat{\theta},\theta\right)} = 1 - \left(\sin(\theta)\sin\left(\hat{\theta}\right) + \cos(\theta)\cos\left(\hat{\theta}\right)\right)$$

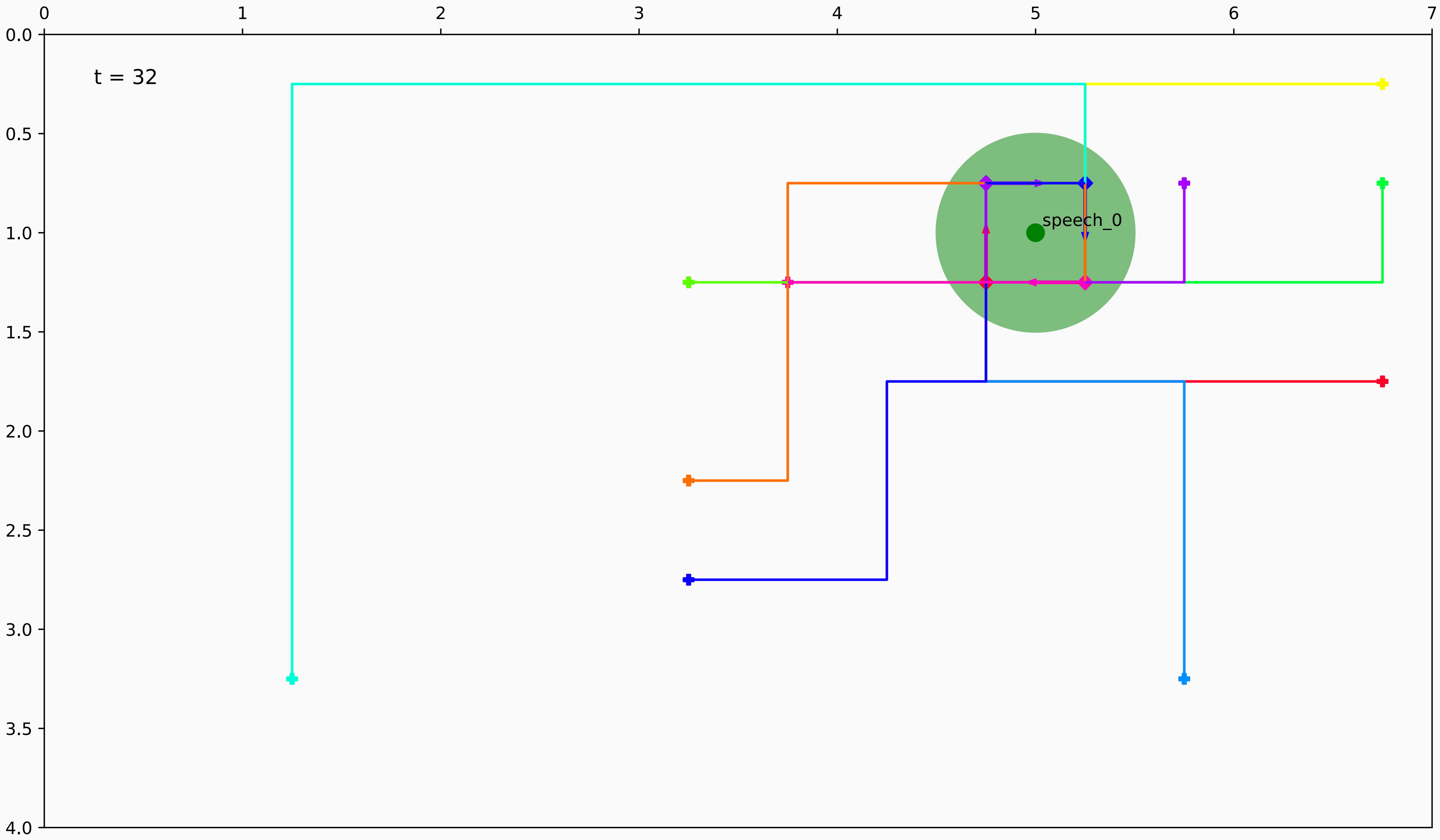
SSL Backbone Architecture



Training loss:

$$\mathcal{L}_{\text{DoA}\left(\hat{\theta},\theta\right)} = 1 - \left(\sin(\theta)\sin\left(\hat{\theta}\right) + \cos(\theta)\cos\left(\hat{\theta}\right)\right)$$

Agent Trajectories



Metrics

Undiscounted cumulated reward:

$$\bar{R} = \frac{1}{n_{\mathrm{ep}}} \sum_{i=1}^{n_{\mathrm{ep}}} \sum_{t=1}^{T} r_{i,t}$$

Metrics

Undiscounted cumulated reward:

$$\bar{R} = \frac{1}{n_{\text{ep}}} \sum_{i=1}^{n_{\text{ep}}} \sum_{t=1}^{T} r_{i,t}$$

Mean final cost:

$$\hat{C}_F = \frac{100}{n_{\text{ep}}} \sum_{i=1}^{n_{\text{ep}}} C\big(s_{i,T}\big)$$

Metrics

Undiscounted cumulated reward:

$$\bar{R} = \frac{1}{n_{\text{ep}}} \sum_{i=1}^{n_{\text{ep}}} \sum_{t=1}^{T} r_{i,t}$$

Mean final cost:

$$\hat{C}_F = \frac{100}{n_{\text{ep}}} \sum_{i=1}^{n_{\text{ep}}} C(s_{i,T})$$

Results

Policy	Omnidirectional cost		Directional cost	
	$\bar{R}\uparrow$	$\hat{C}_F$ (%) $\downarrow$	$\bar{R}\uparrow$	$\hat{C}_F$ (%) $\downarrow$
π_{random}	−25	21.16	−22	22.2
$\pi_{\text{safe random}}$	1420	20.99	1408	22.38
π_{still}	1481	21.13	1512	21.37
$\pi_{\text{still orient}}$	1495	20.87	1789	16.56
π_θ	**2432**	**4.18**	**2302**	**8.01**

Summary - Deep RL for Navigation

- Definition of a novel **perceptually-motivated navigation task**
- Improving the **ASR performance** by position optimization
- Implementation of a complete Gym-compatible [1] environment from our simulator
- Training of a **Deep RL agent** that successfully solves the task

[1] Brockmana et al., "Openai Gym," *arXiv preprint*, 2016.

Introduction

1 **Acoustic Robot Simulator**

Simulate dynamic acoustic environments

2 **(Active) Sound Source Localization**

Accurately localize speaker(s) in a reverberant room

3 **Deep RL for Sound-Based Navigation**

Learn to navigate to hear humans better

Conclusion

Summary of Contributions

1. Design and implementation of an holistic **simulation library** for modeling audio-based interactions.

Summary of Contributions

1. Design and implementation of an holistic **simulation library** for modeling audio-based interactions.
2. Extensive experimental studies of deep-learning-based methods solving two variations of the **static SSL** problem.

Summary of Contributions

1. Design and implementation of an holistic **simulation library** for modeling audio-based interactions.
2. Extensive experimental studies of deep-learning-based methods solving two variations of the **static SSL** problem.
3. Design and experimental evaluation of a novel deep-learning-based solution to an **active multi-source localization** problem.

Summary of Contributions

1. Design and implementation of an holistic **simulation library** for modeling audio-based interactions.
2. Extensive experimental studies of deep-learning-based methods solving two variations of the **static SSL** problem.
3. Design and experimental evaluation of a novel deep-learning-based solution to an **active multi-source localization** problem.
4. Introduction of a perceptually-motivated robotic navigation task.
   Training and evaluation of Deep-RL agent solving this task.

Limitations & Short-Term Perspectives

- Study **limited to simulated environments**. Transferring algorithms trained in virtual environments to real robots is a challenging, yet necessary endeavour.

Limitations & Short-Term Perspectives

- Study **limited to simulated environments**. Transferring algorithms trained in virtual environments to real robots is a challenging, yet necessary endeavour.
- Task and agent constraints. Several assumptions could be relaxed.
  - Static sources $\rightarrow$ **consider moving sources**
  - free-field microphone array $\rightarrow$ **consider HRTF**
  - Limitation to 2D geometric settings: $\rightarrow$ **extension to 3D (elevation component)**

Targetting **more challenging and realistic problem formulations** would improve the overall relevance of the proposed methods.

Limitations & Short-Term Perspectives

- Study **limited to simulated environments**. Transferring algorithms trained in virtual environments to real robots is a challenging, yet necessary endeavour.
- Task and agent constraints. Several assumptions could be relaxed.
  - Static sources → **consider moving sources**
  - free-field microphone array → **consider HRTF**
  - Limitation to 2D geometric settings: → **extension to 3D (elevation component)**

  Targetting **more challenging and realistic problem formulations** would improve the overall relevance of the proposed methods.
- **Engineering and algorithmic challenges:**
  - The RL agent's training is expensive, and tedious. Numerous engineering considerations are required to ensure a successful policy learning.
  - Relying on pre-computed WER cost maps allows the RL environment to run at a high refresh rate, but doesn't easily scale to multiple moving sources.

- **Embodied and multimodal audio perception:**
  - Combine auditory signals with visual cues to leverage social robots' sensors diversity.

[1] Majumder et al., "Move2hear: Active Audio-Visual Source Separation," in *ICCV*, 2021.

[2] Huang et al., "Audio Visual Language Maps for Robot Navigation," in *International Symposium on Experimental Robotics*, 2023.

[3] Huang et al., "Multimodal Spatial Language Maps for Robot Navigation and Manipulation," *arXiv preprint arXiv:2506.06862*, 2025.

Perspectives

- **Embodied and multimodal audio perception:**
  - Combine auditory signals with visual cues to leverage social robots' sensors diversity.
- **Active perception beyond localization:**
  - Explore other navigation objectives: speaker-following, audio-based exploration, information-seeking policy, etc.[1][2][3]

---

[1]Majumder et al., "Move2hear: Active Audio-Visual Source Separation," in *ICCV*, 2021.

[2]Huang et al., "Audio Visual Language Maps for Robot Navigation," in *International Symposium on Experimental Robotics*, 2023.

[3]Huang et al., "Multimodal Spatial Language Maps for Robot Navigation and Manipulation," *arXiv preprint arXiv:2506.06862*, 2025.

Perspectives

- **Embodied and multimodal audio perception:**
  - Combine auditory signals with visual cues to leverage social robots' sensors diversity.
- **Active perception beyond localization:**
  - Explore other navigation objectives: speaker-following, audio-based exploration, information-seeking policy, etc.[1][2][3]
- **Model efficiency and generalization:**
  - Improve RL agents training efficiency and generalization capabilities.
  - Solve more diverse and challenging MDPs (changing room geometries, moving sources, noisy conditions, etc.)

---

[1] Majumder et al., "Move2hear: Active Audio-Visual Source Separation," in *ICCV*, 2021.

[2] Huang et al., "Audio Visual Language Maps for Robot Navigation," in *International Symposium on Experimental Robotics*, 2023.

[3] Huang et al., "Multimodal Spatial Language Maps for Robot Navigation and Manipulation," *arXiv preprint arXiv:2506.06862*, 2025.

Thank you!