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

Gaétan Lepage

July 15, 2025

Supervised by:

- Xavier ALAMEDA-PINEDA (Research Director, Inria Grenoble)
- Laurent GIRIN (Professor, Grenoble INP - UGA)
- Chris REINKE (Doctor, CEA Grenoble)

Jury members:

- *Reviewer:* Patrick DANÈS (Professor, Université Paul Sabatier)
- *Reviewer:* Romain SERIZEL (Associate Professor, Université de Lorraine)
- *Examiner:* Elisa RICCI (Professor, University of Trento)
- *President:* Didier SCHWAB (Professor, UGA)

Social Robotics

The ARI robot, PAL 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

The ARI robot, PAL 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

TODO: Add image(s)

Challenges of Auditory Perception in Robotics

TODO: Add image(s)

- Humans mainly communicate through speech
- Robots must properly understand humans to have relevant interactions

Challenges of Auditory Perception in Robotics

TODO: Add image(s)

- Humans mainly communicate through speech
- Robots must properly understand humans to have relevant interactions
- Sound can also be used to localize speakers

Learning Robot Behaviors

TODO: Add image(s)

- Robots need to react to their environment and take actions
- Several objectives and constraints can be described

Plan

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

1

Acoustic Robot Simulator

Simulate dynamic acoustic environments

2

(Active) Sound Source Localization

Accurately localize speaker(s) in a reverberant room

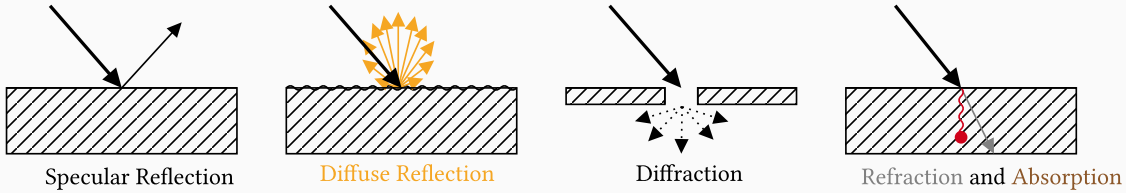
3

Deep RL for Sound-Based Navigation

Learn to navigate to hear humans better

Motivation

TODO: Probably missing another figure here

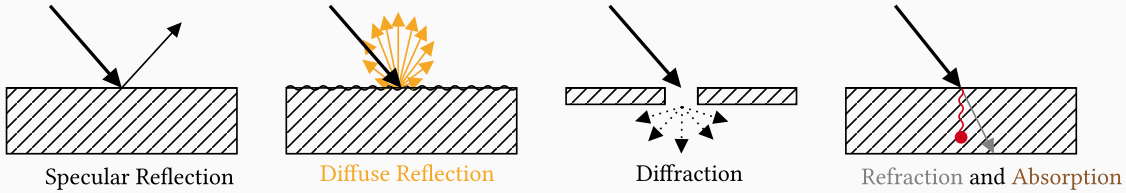


Objectives:

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

Motivation

TODO: Probably missing another figure here



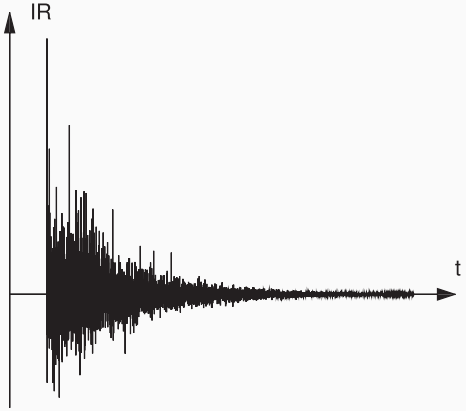
Objectives:

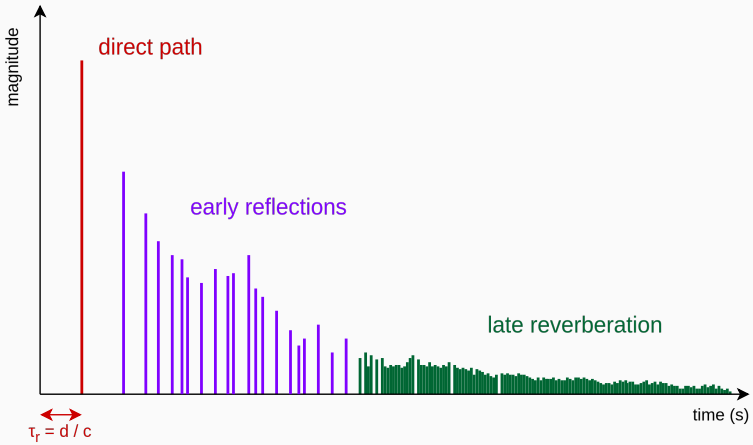
- Modelling realistic acoustic environments
- Simulating sound propagation in reverberant rooms
- Provide high-level primitives for experimenting with robotic auditory perception

Motivations:

- Collecting significant amounts of data
- Lack of holistic approaches to interactive acoustic simulation

Room Impulse Response

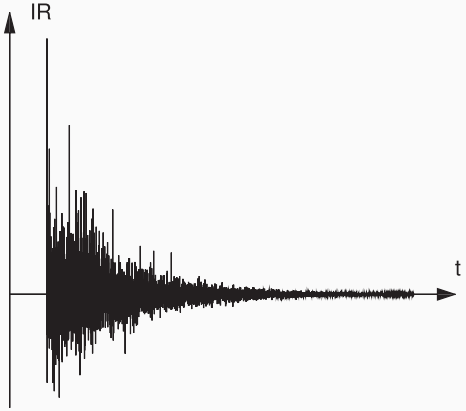


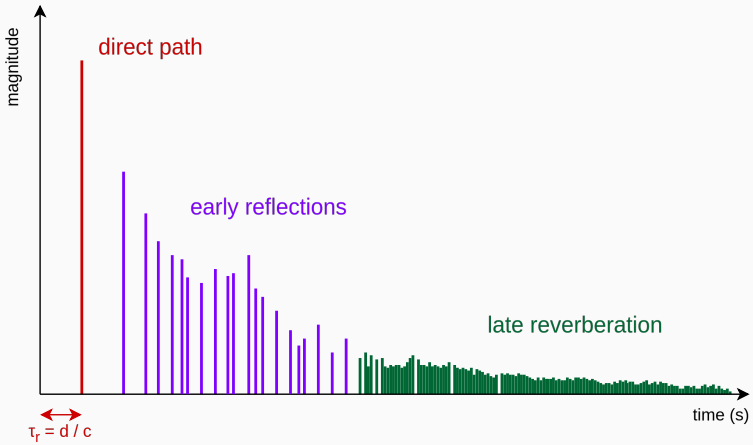


Room Impulse Response:

- 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





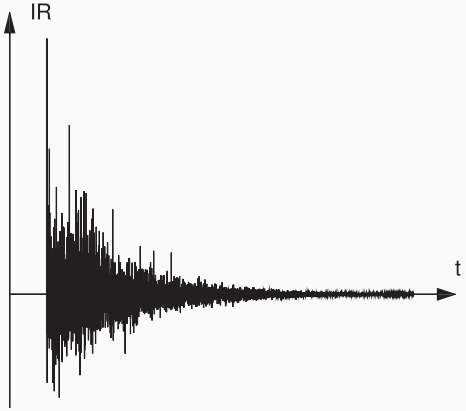
Room Impulse Response:

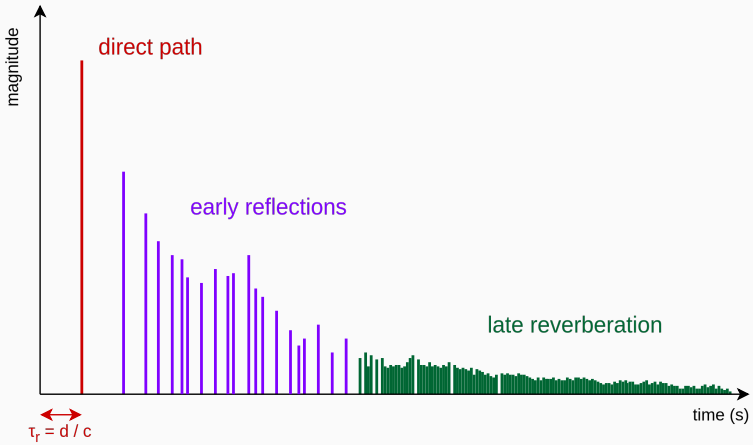
- Characterizes the reverberation properties of the room
- Computed for each source-microphone pair
- T_{60} measures the reverberation level
- The resulting image/microphone signal is obtained by convolving each source signal with the corresponding RIR, and summing over the sources

Single source:

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

Room Impulse Response





Room Impulse Response:

- 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:

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

Multi source:

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

Existing Simulation Methods

- Numerical simulation [1] [2]:

$$\nabla^2 \boldsymbol{p} = \frac{1}{c^2} \frac{\partial^2 \boldsymbol{p}}{\partial t^2} \quad \text{(Helmotz equation)}$$

[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 Transactions on Visualization and Computer Graphics*, 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," 1979.

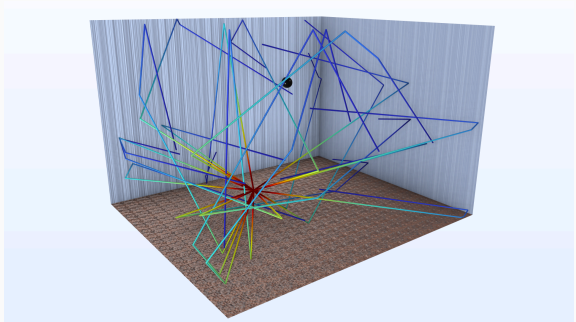
Existing Simulation Methods

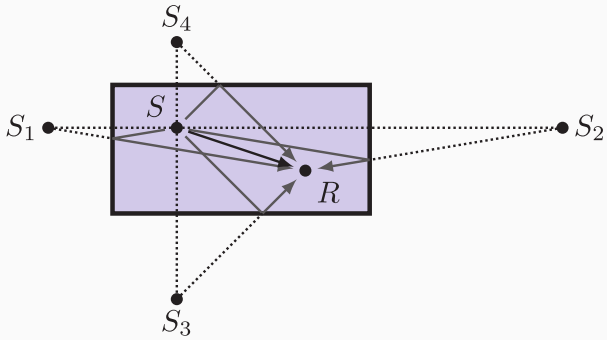
- Numerical simulation [1] [2]:

$$\nabla^2 \boldsymbol{p} = \frac{1}{c^2}\frac{\partial^2 \boldsymbol{p}}{\partial t^2} \quad \text{(Helmotz equation)}$$

- Geometrical Acoustics
  - Ray-tracing [3],
  - Image Source Model [4]

-> Generate RIR from a 3D room specification





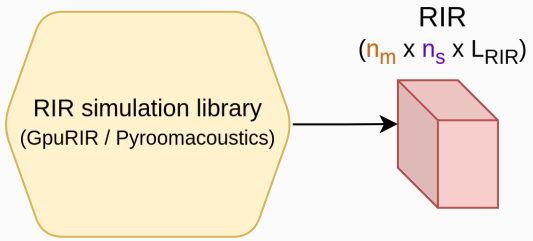
[1] D. Botteldooren, "Acoustical finite-difference time-domain simulation in a quasi-Cartesian grid," *JASA*, 1994.

[2] Raghuvanshi et al., "Efficient and accurate sound propagation using adaptive rectangular decomposition," *IEEE Transactions on Visualization and Computer Graphics*, 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," 1979.

Audio-Processing Pipeline



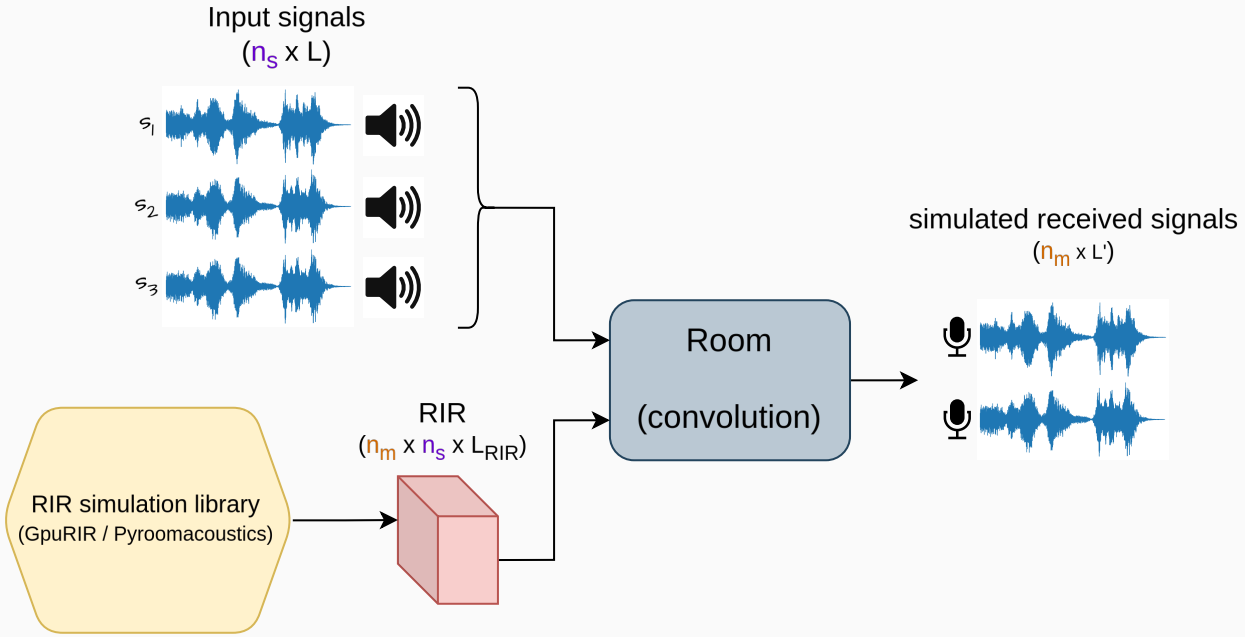
Support for two backend libraries: *PyroomAcoustics* [1] and *gpuRIR* [2].

---

[1] Scheibler et al., "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018.

[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



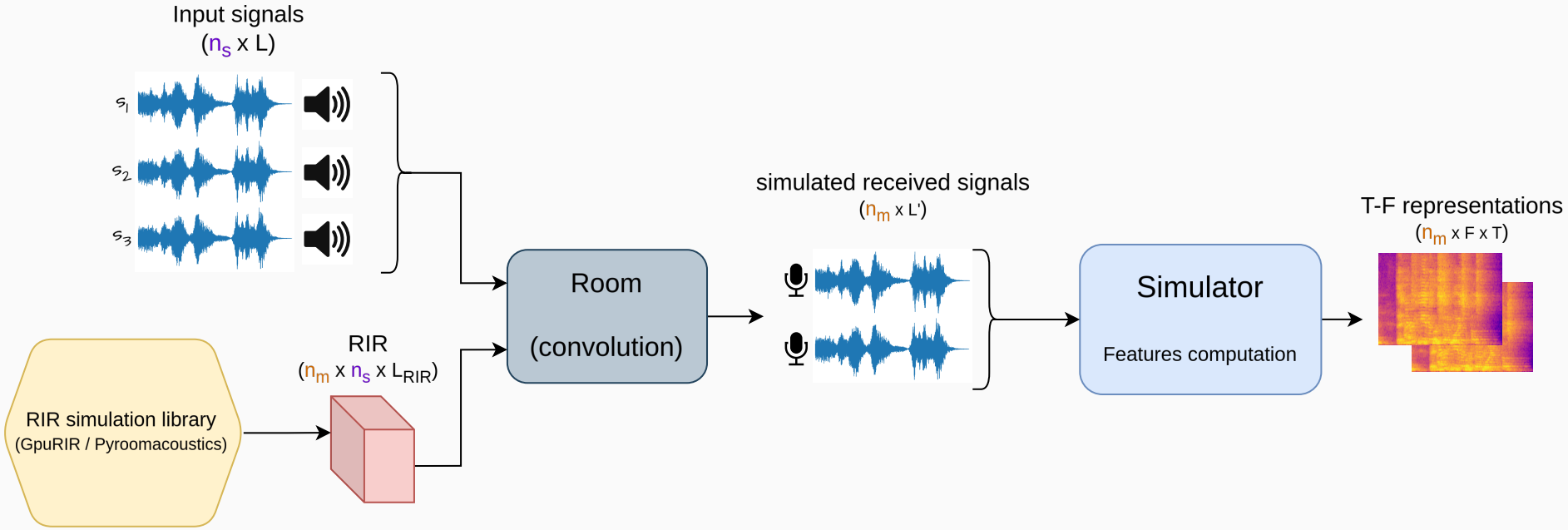
Support for two backend libraries: *PyroomAcoustics* [1] and *gpuRIR* [2].

---

[1] Scheibler et al., "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018.

[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



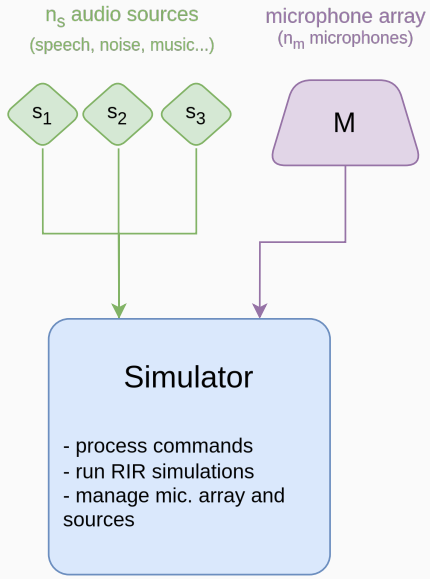
Support for two backend libraries: *PyroomAcoustics* [1] and *gpuRIR* [2].

---

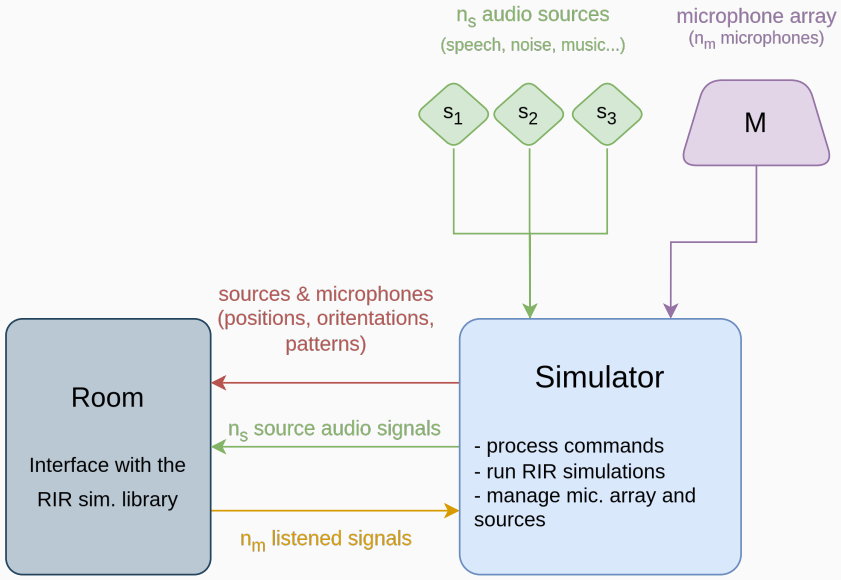
[1] Scheibler et al., "Pyroomacoustics: A python package for audio room simulation and array processing algorithms," in *2018 IEEE international conference on acoustics, speech and signal processing (ICASSP)*, 2018.

[2] Diaz-Guerra et al., "gpuRIR: A python library for room impulse response simulation with GPU acceleration," *Multimedia Tools and Applications*, 2021.

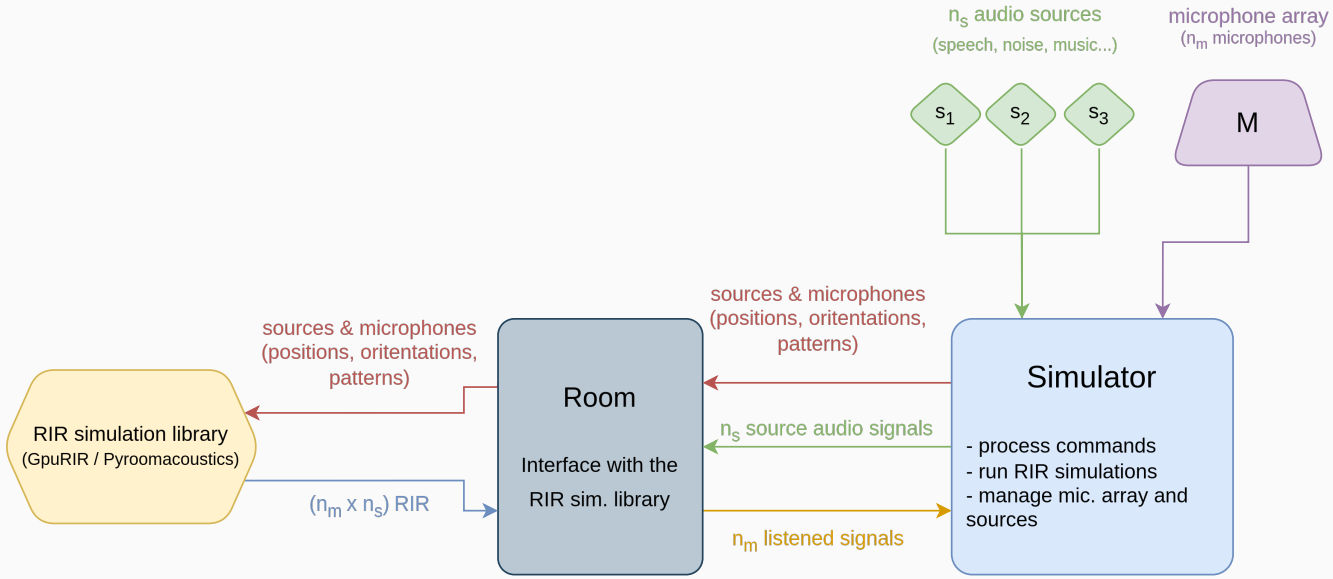
Simulator Architecture



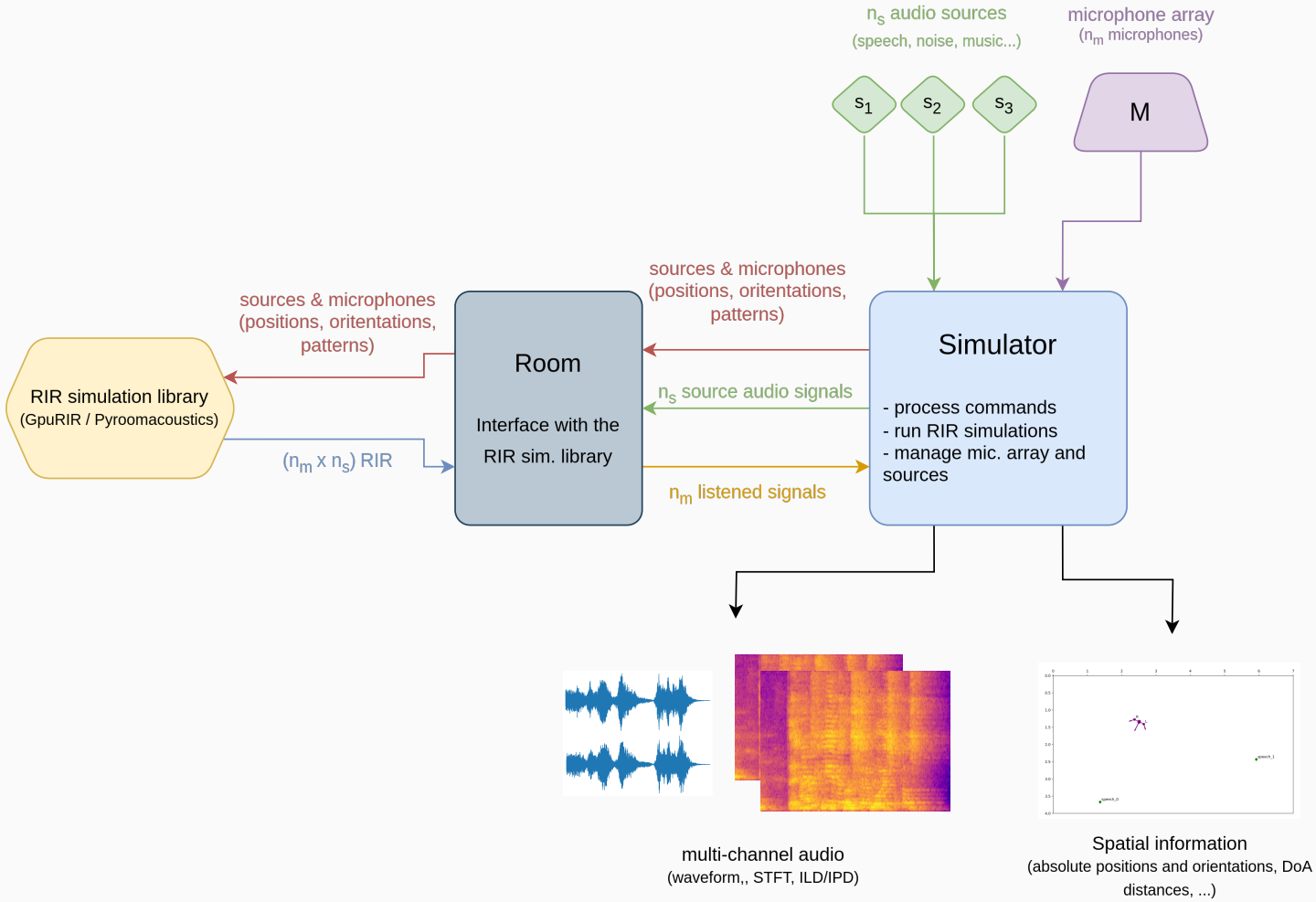
Simulator Architecture



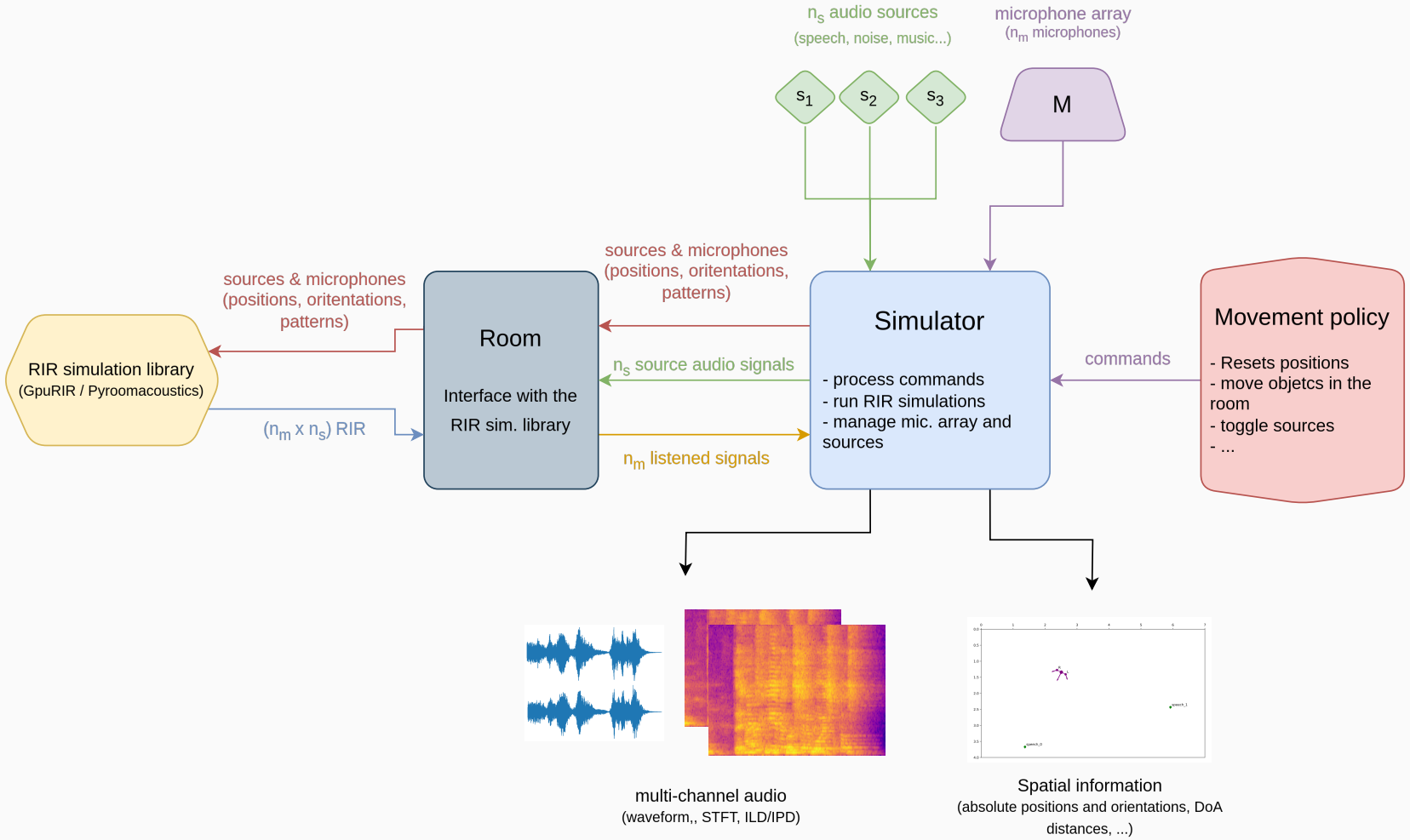
Simulator Architecture



Simulator Architecture



Simulator Architecture



Code Example

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

Code Example

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

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

Code Example

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


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

mic_array = SquareArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
)
```

Code Example

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


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

mic_array = SquareArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
)

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

Code Example

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


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

mic_array = SquareArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
)

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

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

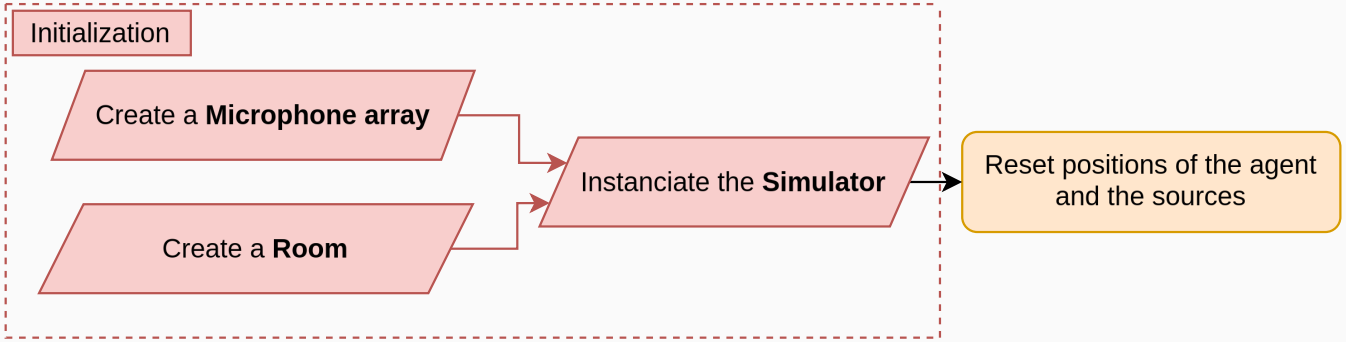
Code Example

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


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

mic_array = SquareArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
)

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

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

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

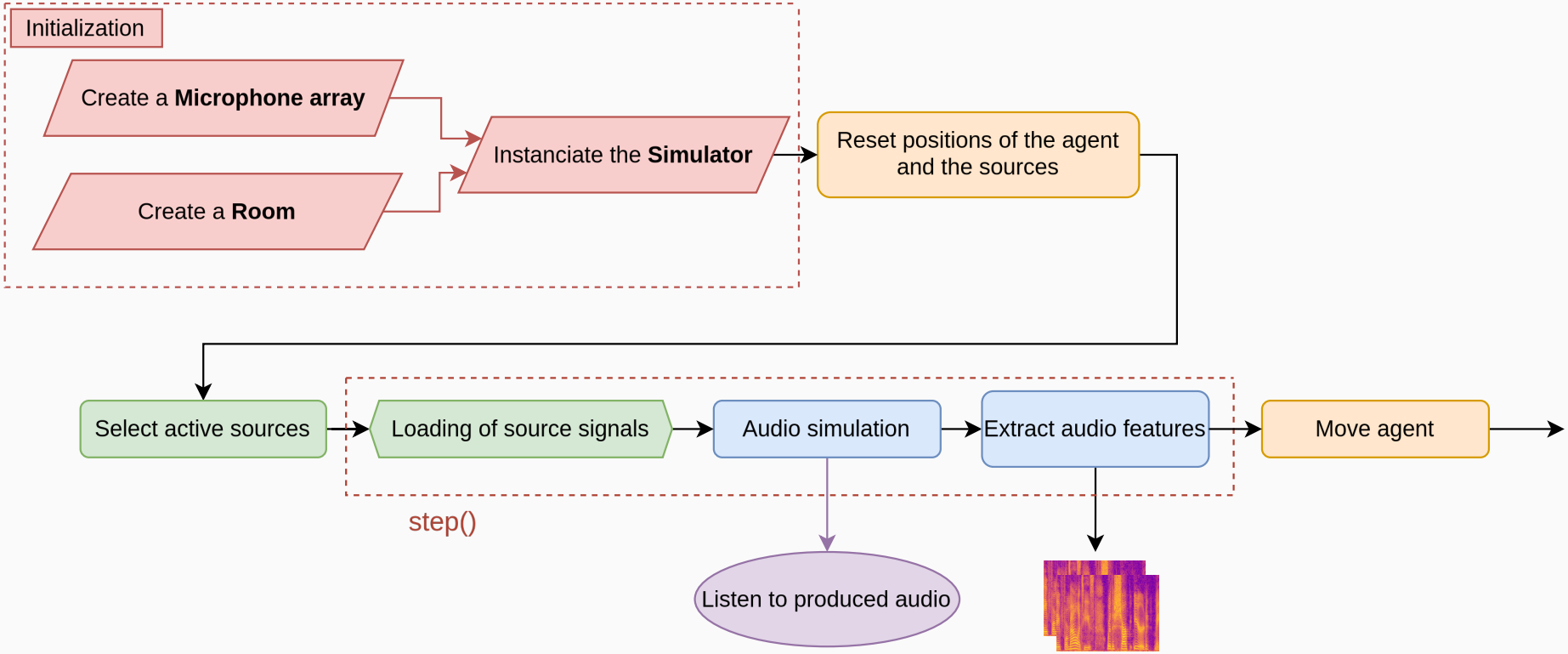
Code Example

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


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

mic_array = SquareArray(
  position=np.array([3.0, 3.0, 1.0]),
  orientation=np.array([-1.0, 1.0, 0.0]),
)

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

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

# (4, 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")
```

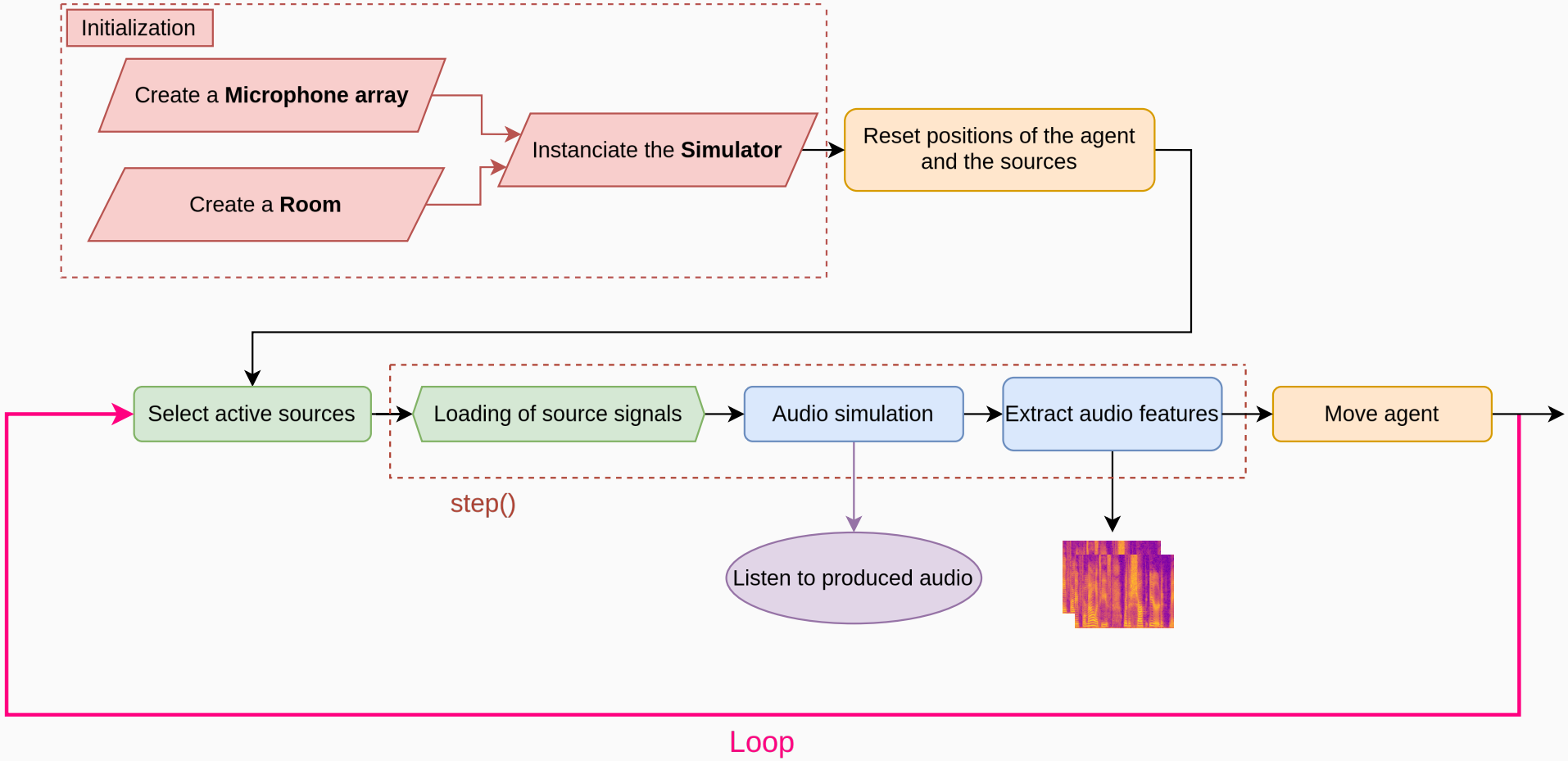
Modelling Active Scenarios



Modelling Active Scenarios



Modelling Active Scenarios



Summary

- Complete solution for modelling 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
  - Modelling interactive scenarios where both microphones and sources can move
  - Use as an environment to train Deep RL agents

1

Acoustic Robot Simulator

Simulate dynamic acoustic environments

2

(Active) Sound Source Localization

Accurately localize speaker(s) in a reverberant room

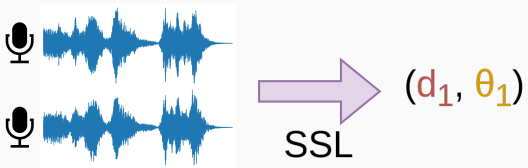
3

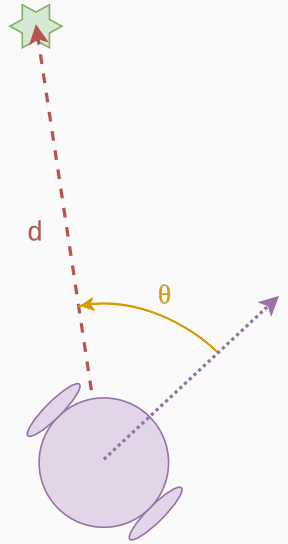
Deep RL for Sound-Based Navigation

Learn to navigate to hear humans better

From Static to Active SSL

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





---

[1] Gustafsson et al., "Source localization in reverberant environments: Modeling and statistical analysis," *IEEE Trans. Speech Audio Process.*, 2004.

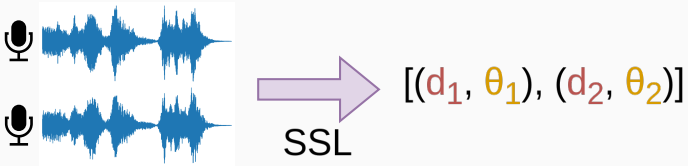
[2] Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE Trans. Antennas Propag.*, 1986.

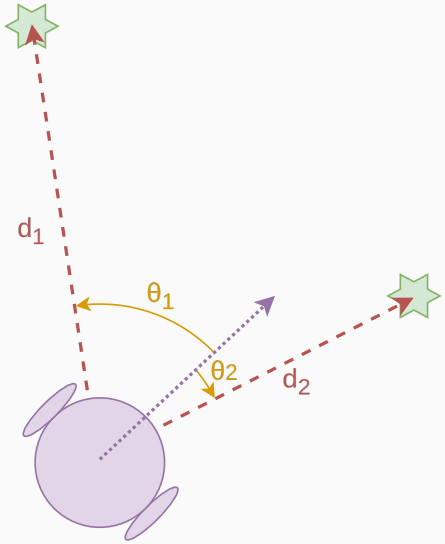
[3] Grumiaux et al., "A survey of sound source localization with deep learning methods," *JASA*, 2022.

[4] Argentieri et al., "A survey on sound source localization in robotics: From binaural to array processing methods," *Comput. Speech Lang.*, 2015.

From Static to Active SSL

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





---

[1] Gustafsson et al., "Source localization in reverberant environments: Modeling and statistical analysis," *IEEE Trans. Speech Audio Process.*, 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.

From Static to Active SSL

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., "Autonomous sensorimotor learning for sound source localization by a humanoid robot," in *IROS*, 2018.

[3] Bustamante et al., "Towards information-based feedback control for binaural active localization," in *ICASSP*, 2016.

[4] Evers et al., "The LOCATA challenge: Acoustic source localization and tracking," *IEEE/TASLP*, 2020.

From Static to 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., "Autonomous sensorimotor learning for sound source localization by a humanoid robot," in *IROS*, 2018.

[3] Bustamante et al., "Towards information-based feedback control for binaural active localization," in *ICASSP*, 2016.

[4] Evers et al., "The LOCATA challenge: Acoustic source localization and tracking," *IEEE/TASLP*, 2020.

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

---

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

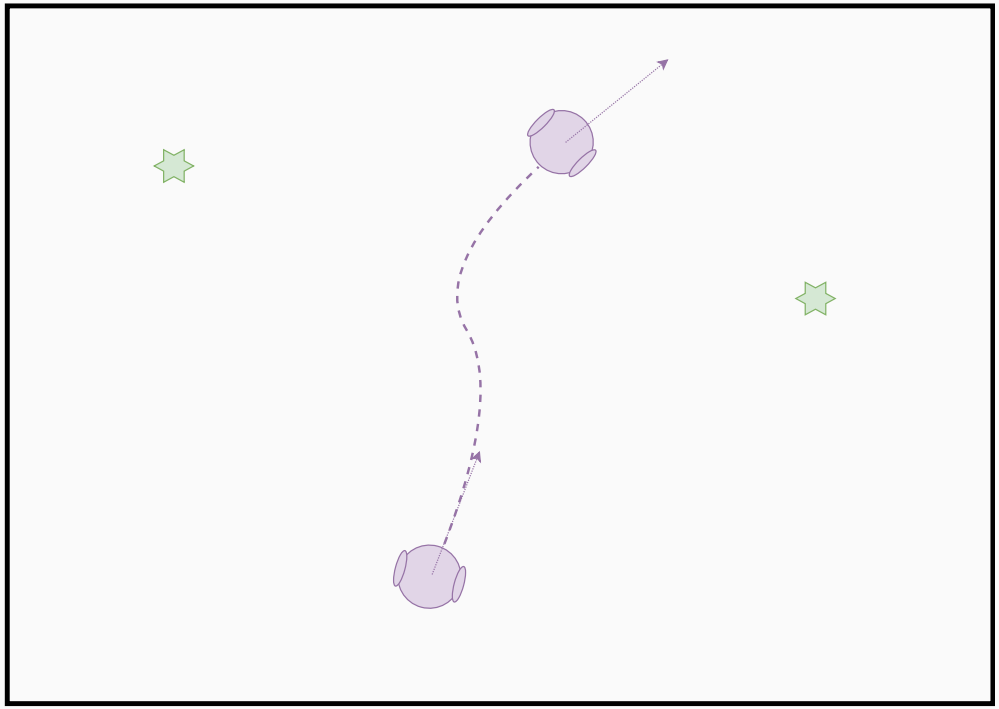
[2] Nguyen et al., "Autonomous sensorimotor learning for sound source localization by a humanoid robot," in *IROS*, 2018.

[3] Bustamante et al., "Towards information-based feedback control for binaural active localization," in *ICASSP*, 2016.

[4] Evers et al., "The LOCATA challenge: Acoustic source localization and tracking," *IEEE/TASLP*, 2020.

Approach

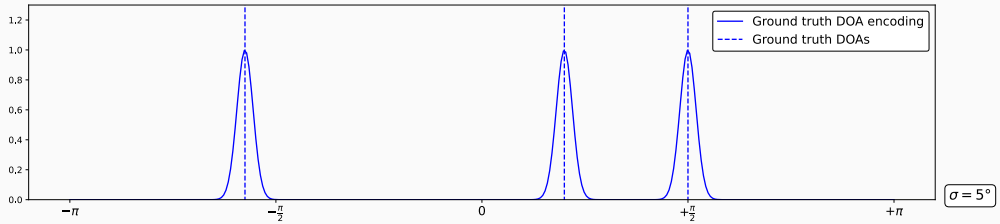
- Project static SSL DoA spectrum to egocentric maps
- Aggregate these maps into a single final heatmap



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

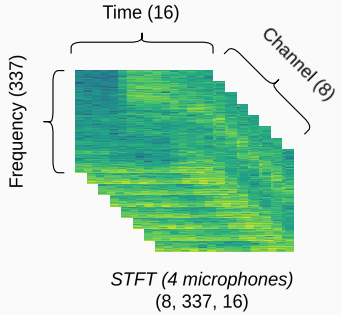
- Encode DoA as a continuous function over $[-\pi, \pi]$ [1]
- Discretized over 360 values
- Can represent an arbitrary number of sources
- Ground-truth DoA of each source is represented with a Gaussian window centered on it
- The SSL task becomes a DoA spectrum regression (with a DNN for e.g.):

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

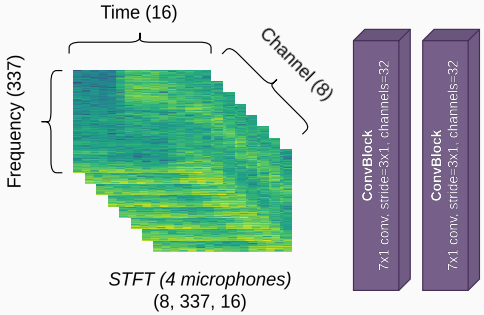


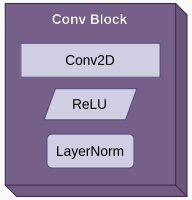
[1] He et al., "Neural Network Adaptation and Data Augmentation for Multi-Speaker Direction-of-Arrival Estimation," *IEEE/ACM Transactions on Audio, Speech, and Language Processing*, 2021.

Static SSL Model (2/2): Network Architecture

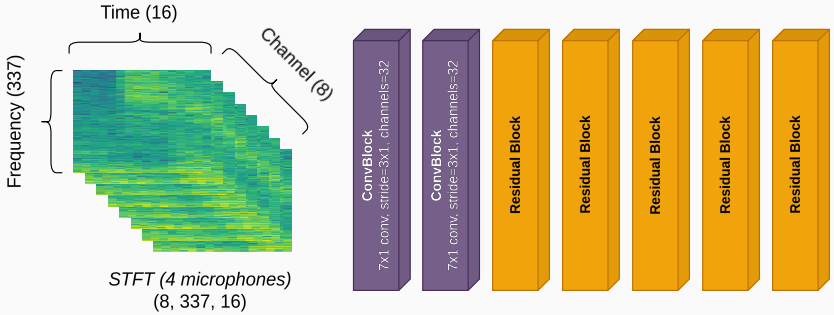


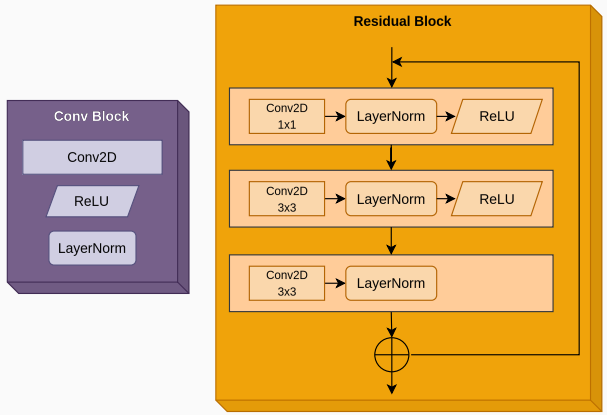
Static SSL Model (2/2): Network Architecture



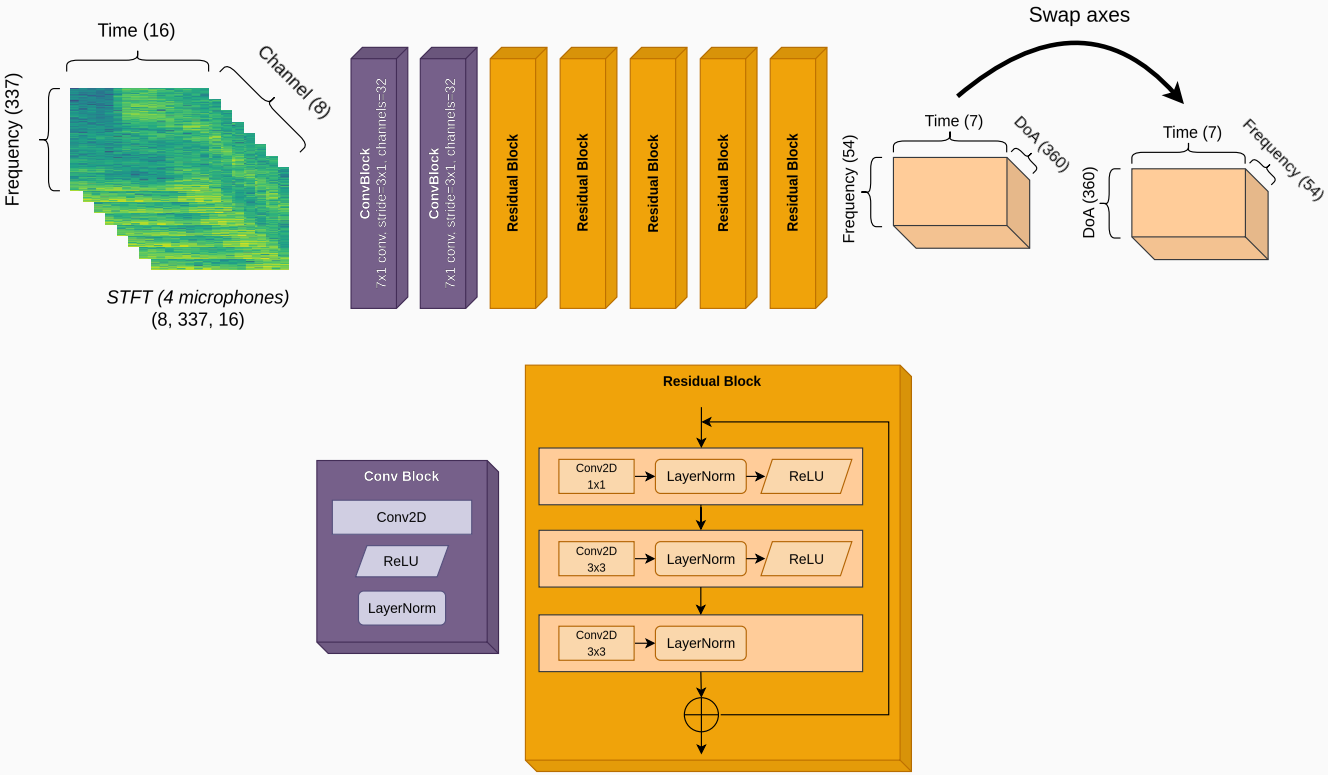


Static SSL Model (2/2): Network Architecture

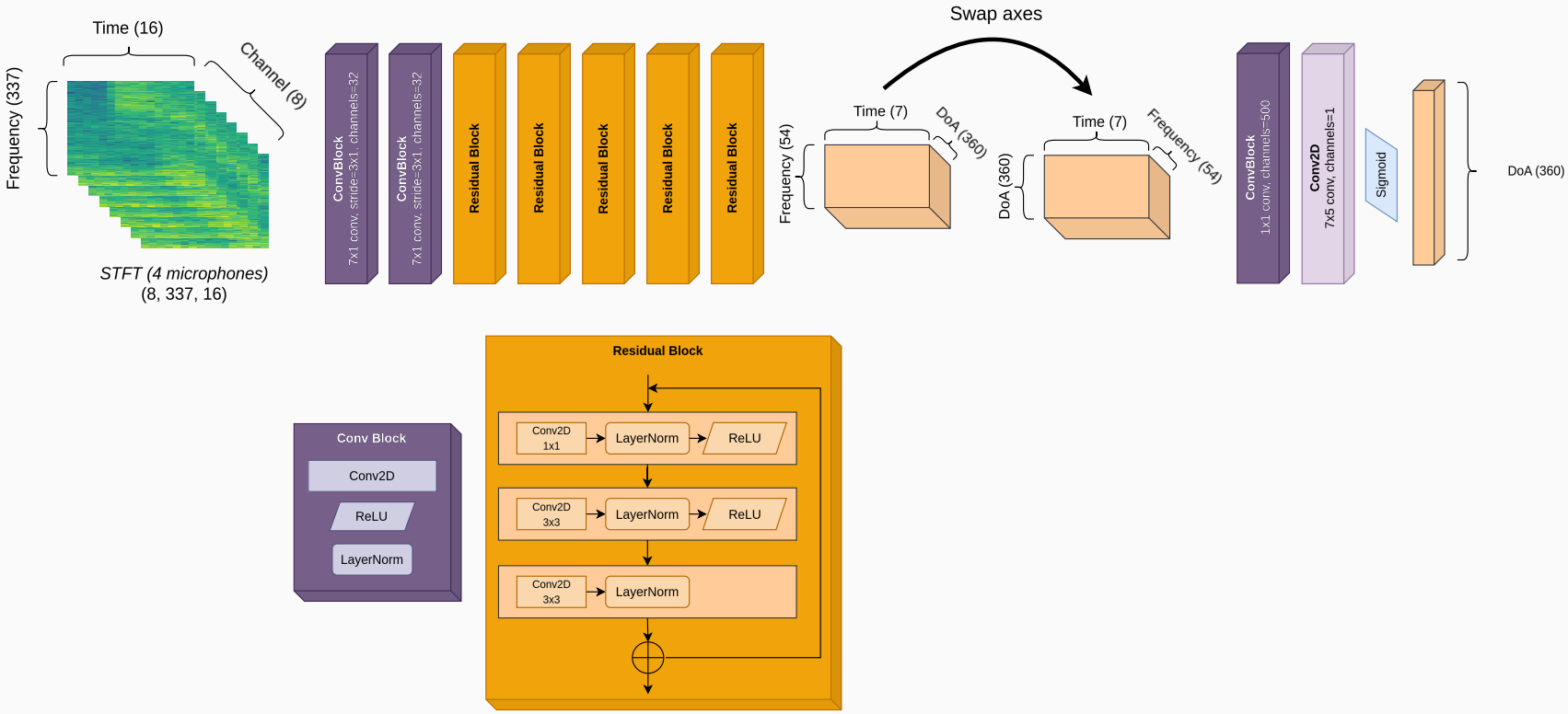




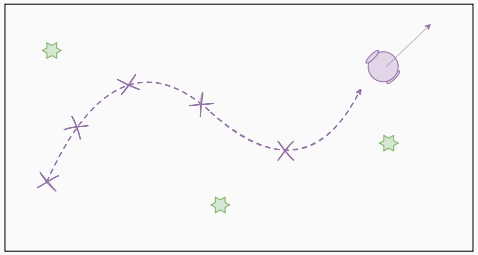
Static SSL Model (2/2): Network Architecture



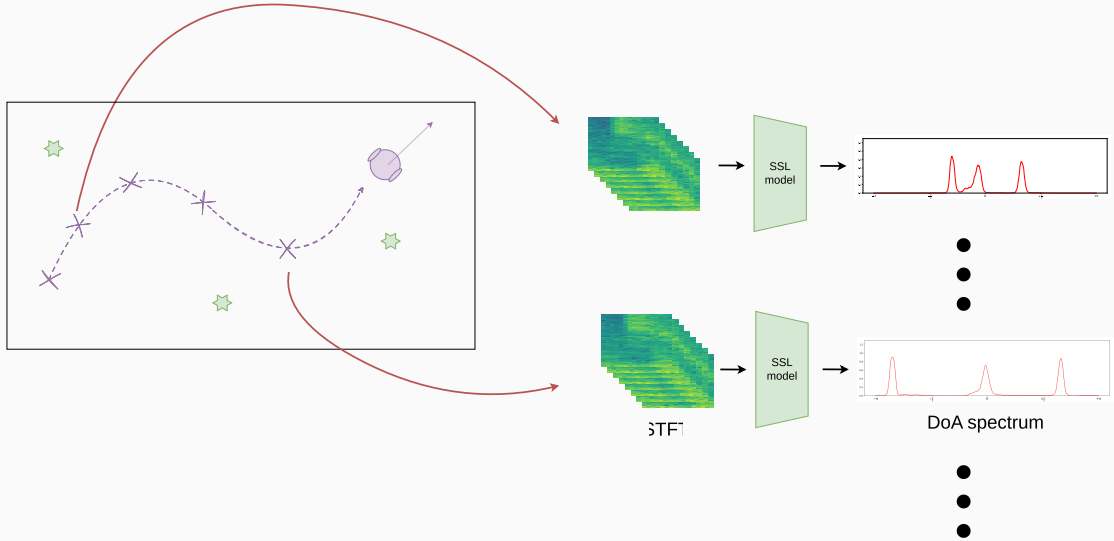
Static SSL Model (2/2): Network Architecture



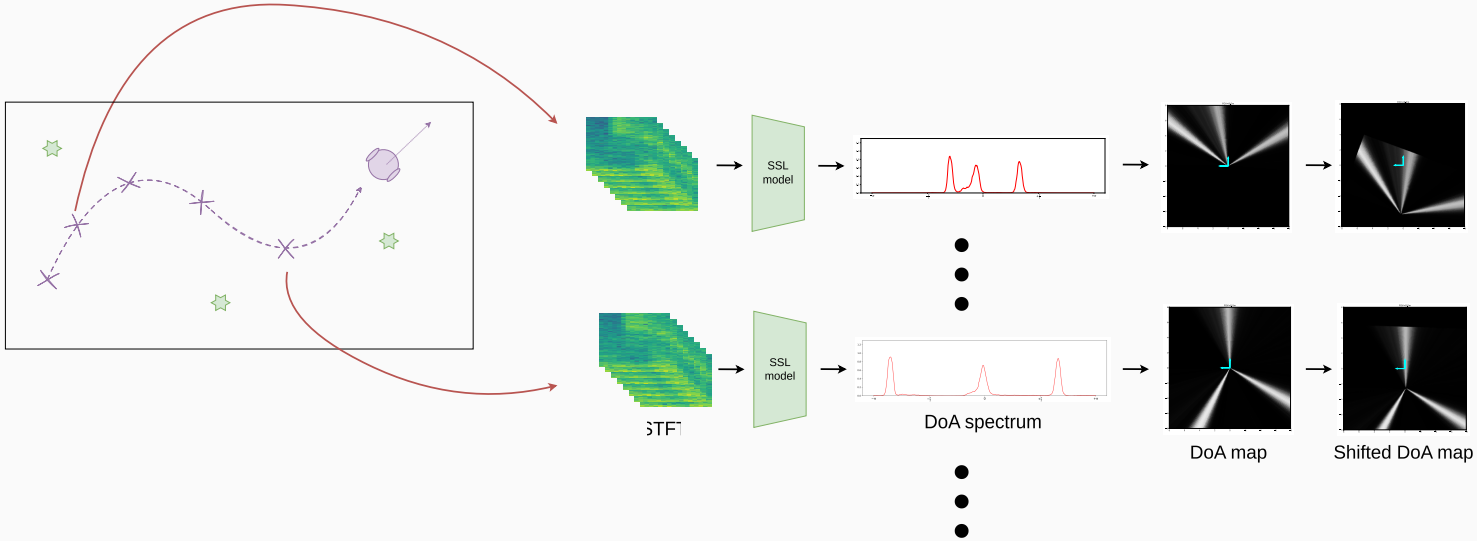
Active sound source localization pipeline



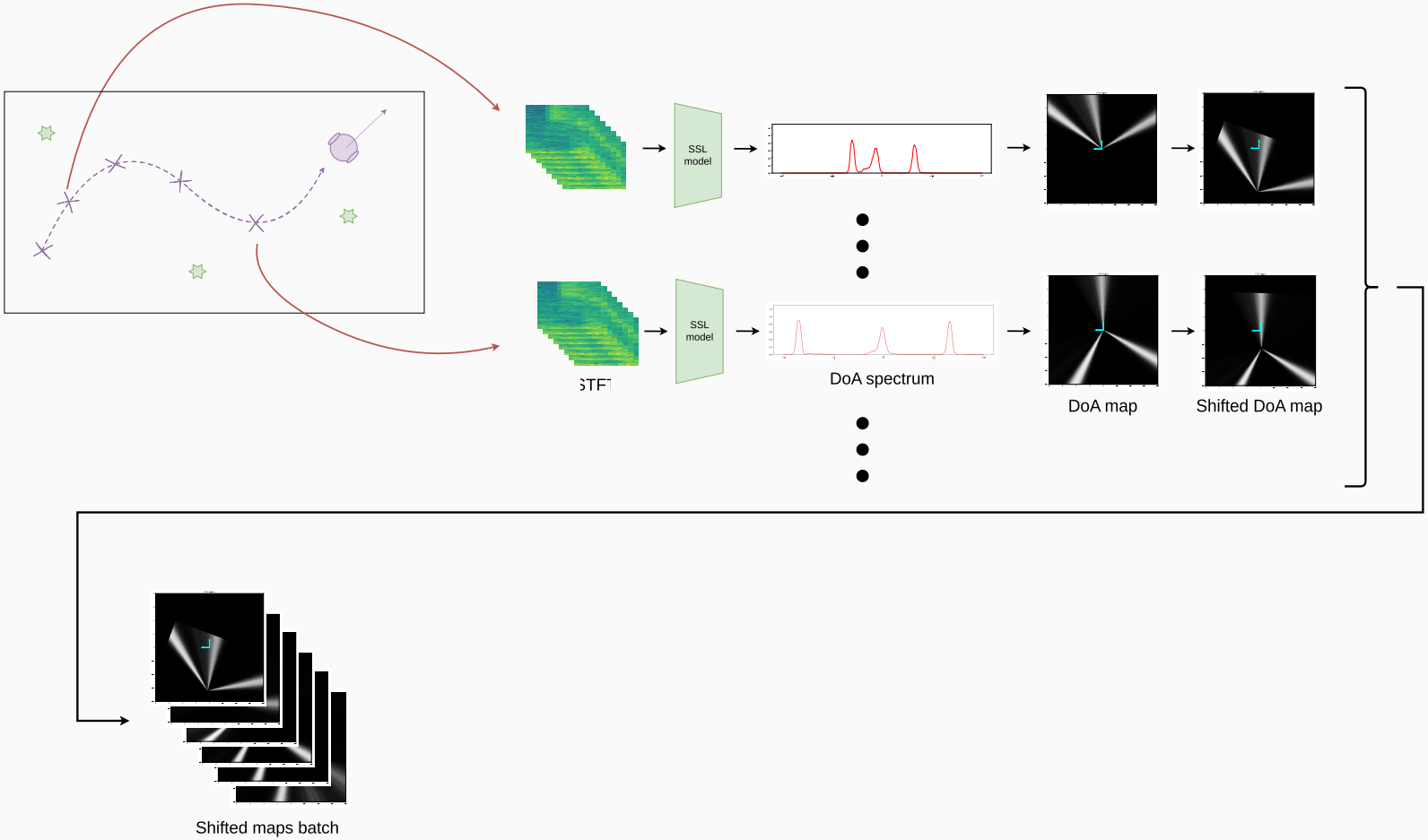
Active sound source localization pipeline



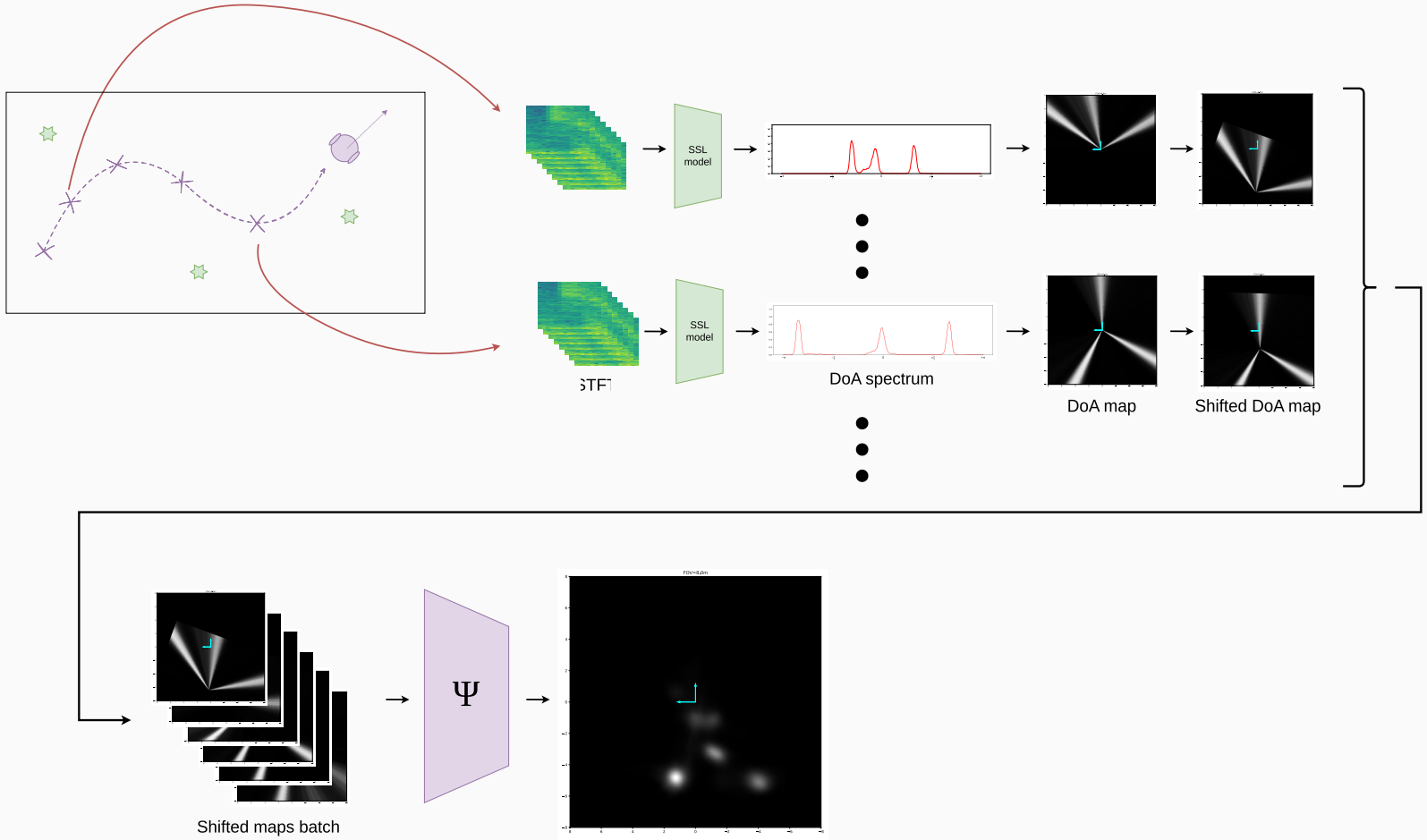
Active sound source localization pipeline



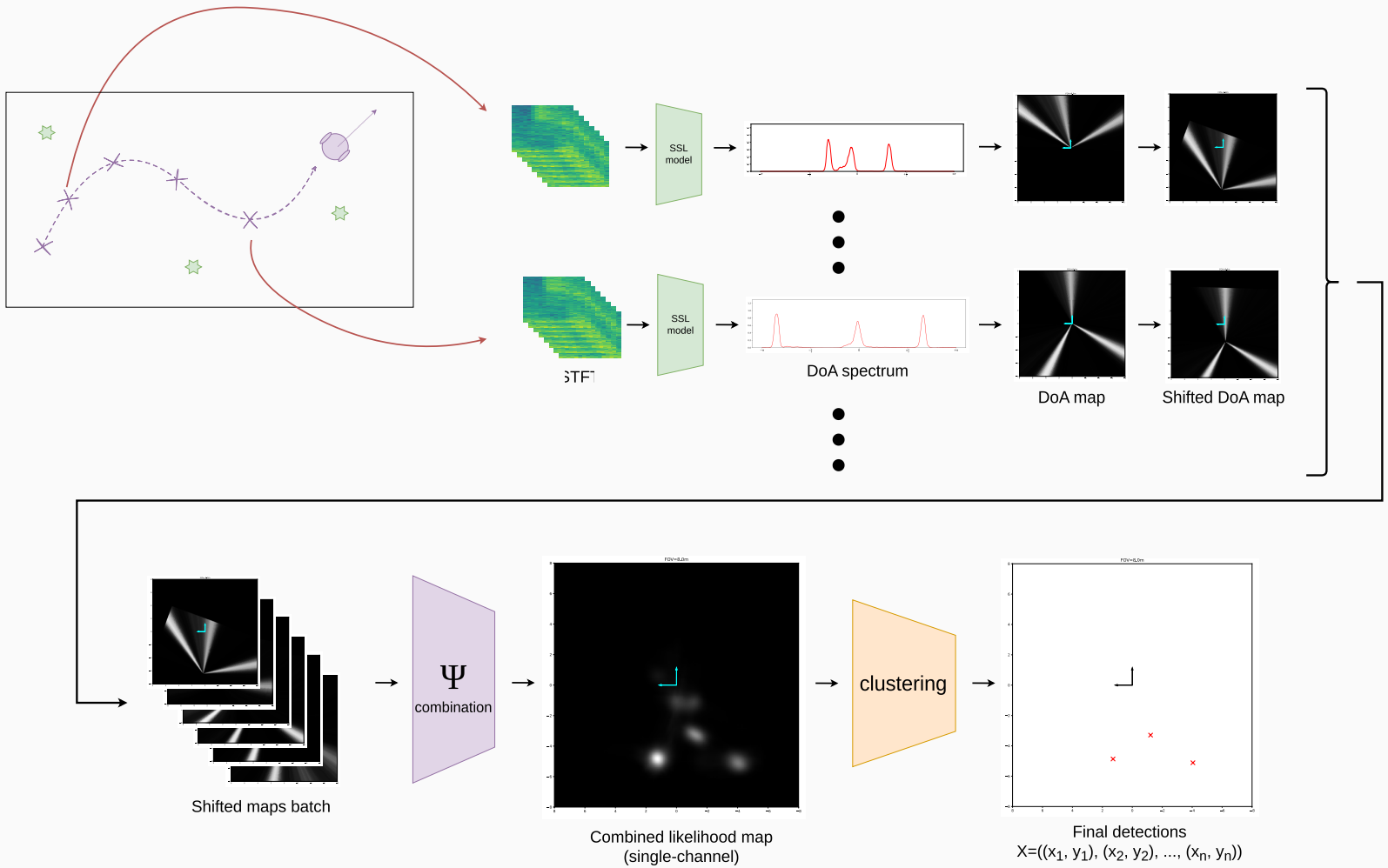
Active sound source localization pipeline



Active sound source localization pipeline



Active sound source localization pipeline



Aggregation strategy

→ **Aggregate shifted maps into a single heatmap**

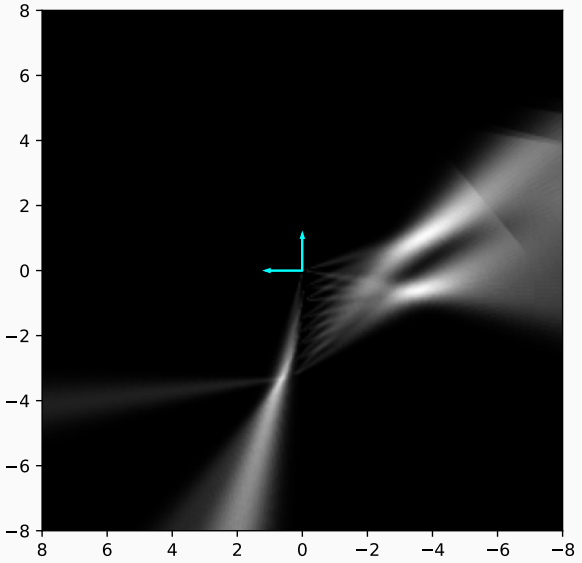
Aggregation strategy

→ Aggregate shifted maps into a single heatmap

Two methods were explored:

- Averaging:

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



Averaging

Aggregation strategy

→ Aggregate shifted maps into a single heatmap

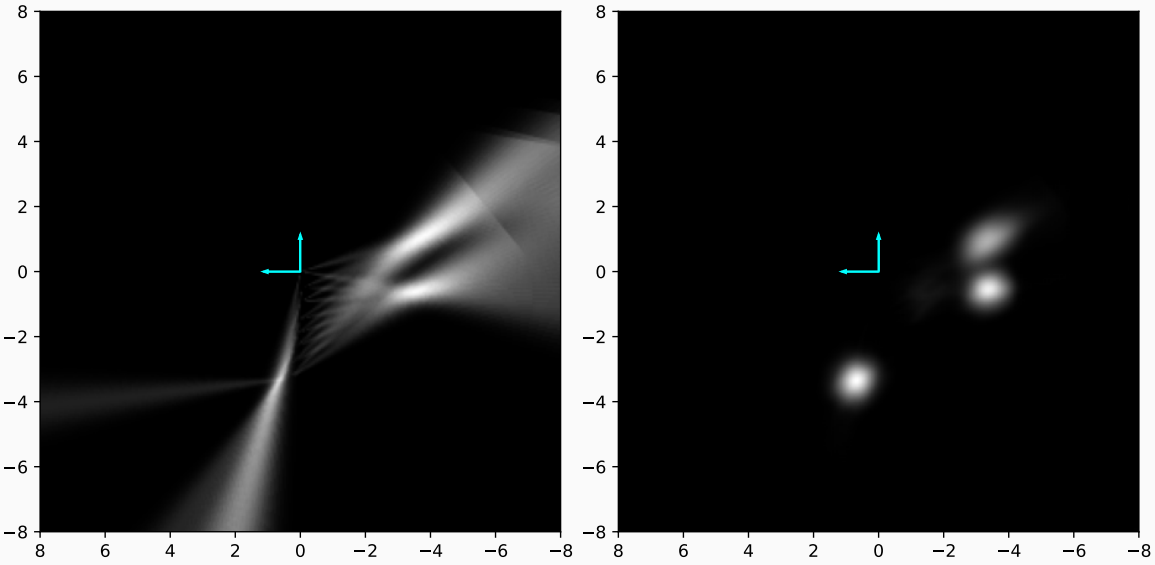
Two methods were explored:

- Averaging:

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

- U-Net [1]:

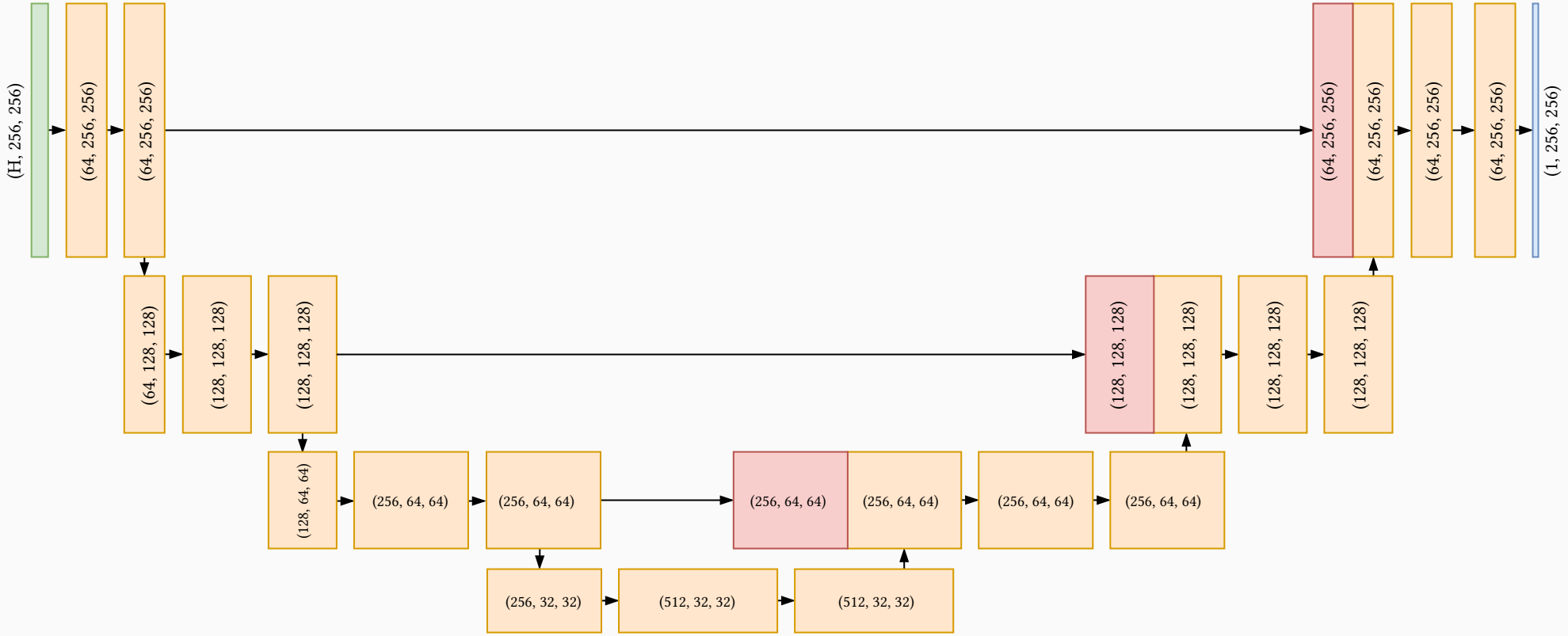
$$\widehat{M}_t = \Psi_{\text{DNN}}(M_{t-H+1}, \ldots, M_t)$$



Averaging DNN

[1] Ronneberger et al., "U-net: Convolutional networks for biomedical image segmentation," in *MICCAI*, 2015.

Neural network-based aggregation



→ **Extract discrete 2D position predictions from the heatmap**

1. Low values are filtered out from the egocentric heatmap (threshold τ)

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

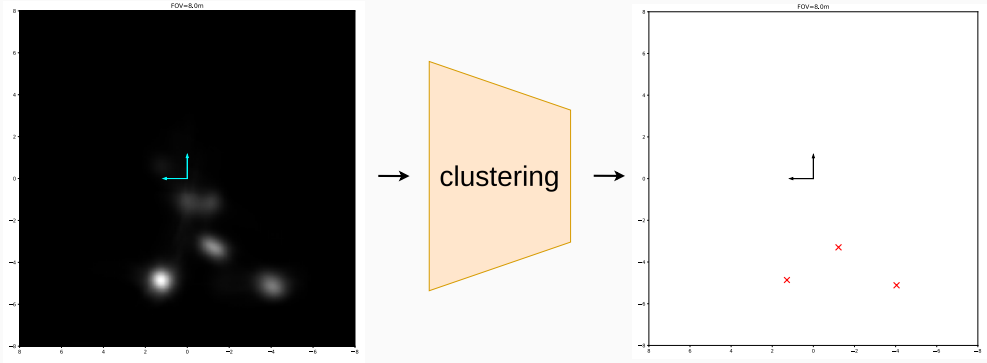
→ **Extract discrete 2D position predictions from the heatmap**

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

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

→ **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

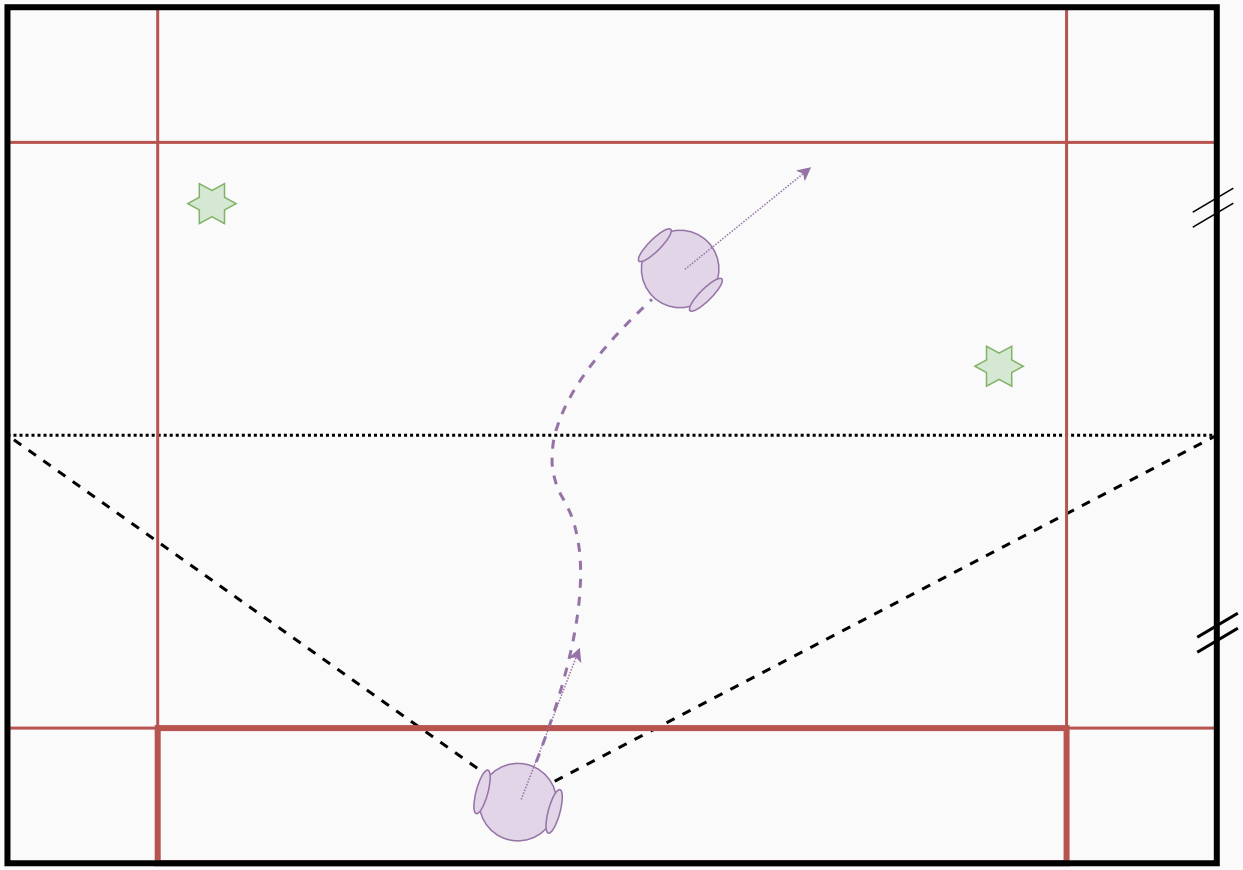


Combined likelihood map (single-channel)

Final detections
$X=((x_1, y_1), (x_2, y_2), ..., (x_n, y_n))$

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

Experimental Setup



Starting zone

Other starting/turn-around zones

Speech sources

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

Evaluation Metrics

We define a threshold δ for defining correct detections

Evaluation Metrics

We define a threshold δ for defining correct detections

1. m counts the number of positive prediction-GT matches

$$m\left(\hat{X}_k^i, X_j^i\right) = \begin{cases} 1 & \text{if } \left\|\hat{X}_{i,k} - X_{i,k}\right\|_2 < \delta \\ & \quad \text{and } k = \underset{k' \in \{1,\ldots,\hat{z}_i\}}{\operatorname{argmin}} \left\|\hat{X}_{i,k'} - X_{i,k'}\right\|_2 \\ 0 & \text{otherwise,} \end{cases}$$

Evaluation Metrics

We define a threshold δ for defining correct detections

1. m counts the number of positive prediction-GT matches
2. The Precision measures the proportion of correct matches among the predictions

$$m\left(\hat{X}_k^i, X_j^i\right) = \begin{cases} 1 & \text{if } \left\|\hat{X}_{i,k} - X_{i,k}\right\|_2 < \delta \\ & \quad \text{and } k = \underset{k' \in \{1,\dots,\hat{z}_i\}}{\operatorname{argmin}} \left\|\hat{X}_{i,k'} - X_{i,k'}\right\|_2 \\ 0 & \text{otherwise,} \end{cases}$$

$$\text{Precision} = \frac{\sum_i \sum_{j=1}^{z_i} \sum_{k=1}^{\hat{z}_i} m\left(\hat{X}_{i,k}, X_{i,k}\right)}{\sum_i \hat{z}_i},$$

Evaluation Metrics

We define a threshold δ for defining correct detections

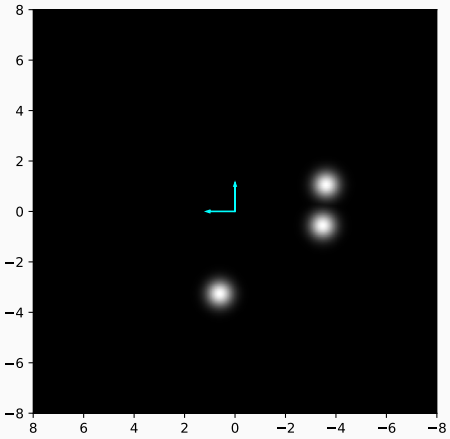
1. m counts the number of positive prediction-GT matches
2. The Precision measures the proportion of correct matches among the predictions
3. The Recall counts the proportion of GT positions that have correcty been identified

$$m\left(\hat{X}_k^i, X_j^i\right) = \begin{cases} 1 & \text{if } \left\|\hat{X}_{i,k} - X_{i,k}\right\|_2 < \delta \\ & \quad \text{and } k = \underset{k' \in \{1,\dots,\hat{z}_i\}}{\operatorname{argmin}} \left\|\hat{X}_{i,k'} - X_{i,k'}\right\|_2 \\ 0 & \text{otherwise,} \end{cases}$$

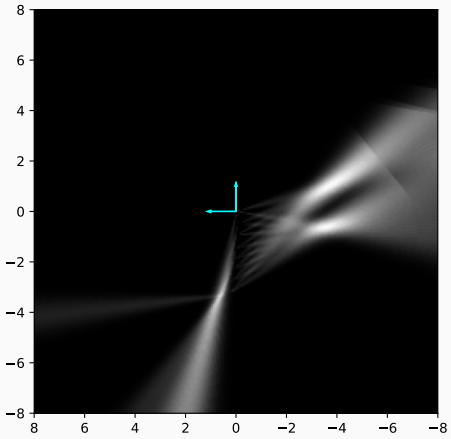
$$\text{Precision} = \frac{\sum_i \sum_{j=1}^{z_i} \sum_{k=1}^{\hat{z}_i} m\left(\hat{X}_{i,k}, X_{i,k}\right)}{\sum_i \hat{z}_i},$$

$$\text{Recall} = \frac{\sum_i \sum_{j=1}^{z_i} \sum_{k=1}^{\hat{z}_i} m\left(\hat{X}_{i,k}, X_{i,k}\right)}{\sum_i z_i}.$$

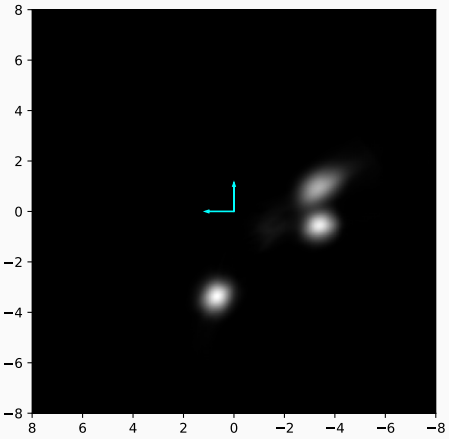
Comparison of Aggregation Methods



Ground truth

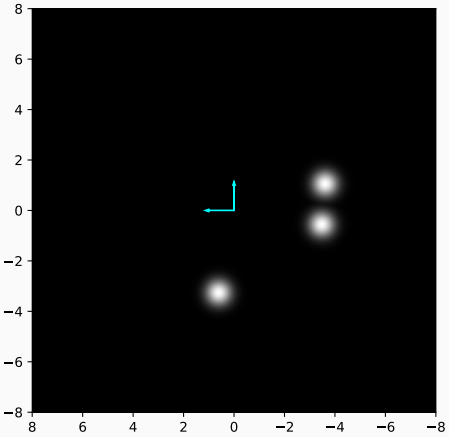


Average

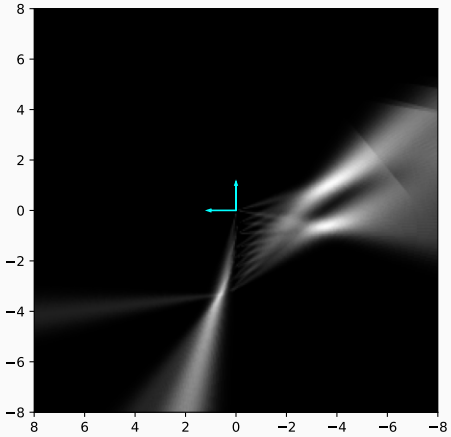


DNN

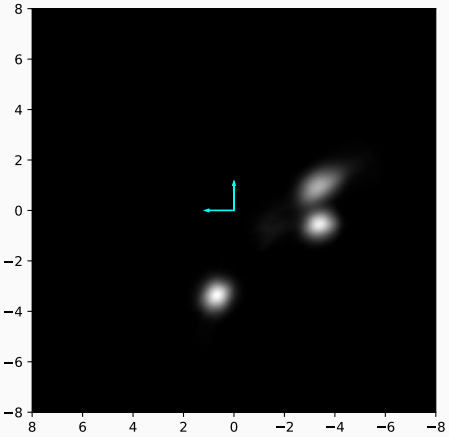
Comparison of Aggregation Methods



Ground truth



Average



DNN

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

Summary

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

Plan

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

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.

→ *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:

$$\text{WER} = \frac{s + d + i}{n}$$

- s: number of substitutions
- d: number of deletions
- i: number of insertions
- n: number of words in the reference

Measuring ASR performance

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

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

$$\text{WER} = \frac{s + d + i}{n}$$

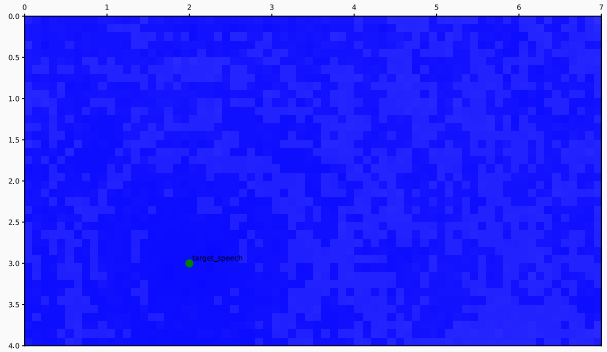
- s: number of substitutions
- d: number of deletions
- i: number of insertions
- n: number of words in the reference

Example:

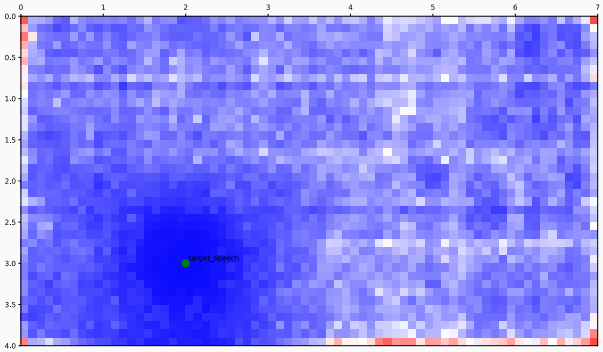
- Reference: *Obviously, he was __ able to catch the last bus on time today.*
- Prediction: *Obviously, he was not able to catch the past bus on time ____ .*

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

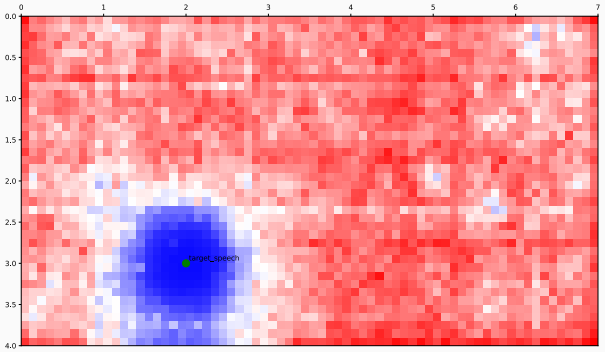
Reverberation impact on WER



(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.

Reinforcement Learning

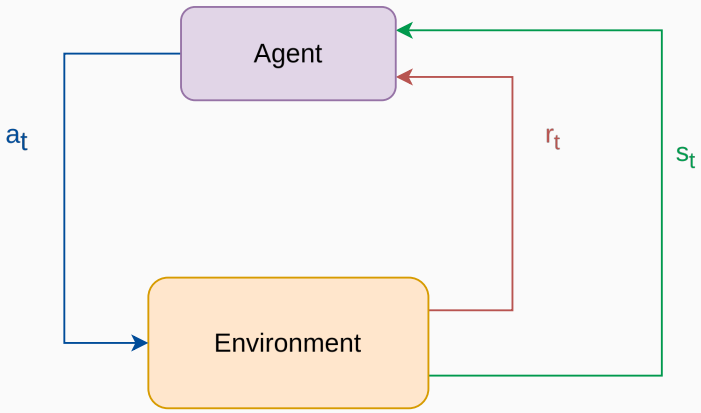
RL solves sequential decision problems, formalized as **Markov Decision Processes (MDPs)**.

Reinforcement Learning

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

At each step:

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

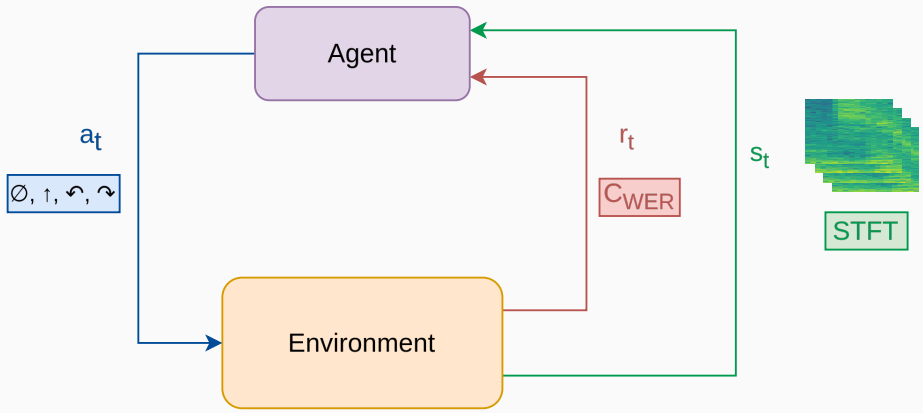


Reinforcement Learning

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

At each step:

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



Our environment:

- the reward is a decreasing function of the WER:

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

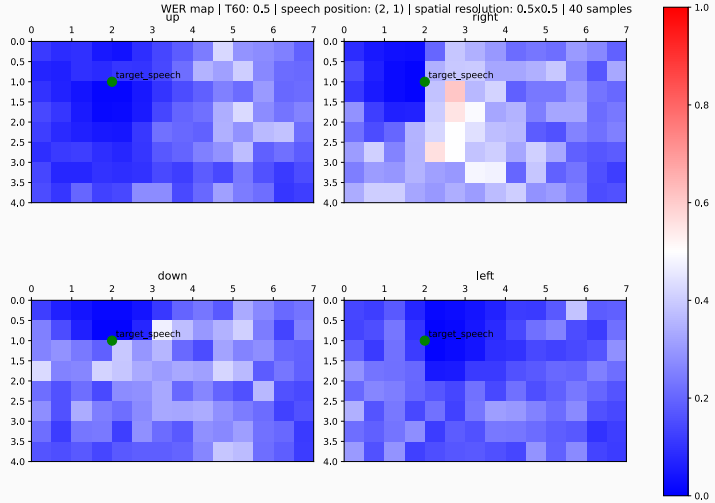
WER Cost Maps

- 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**;

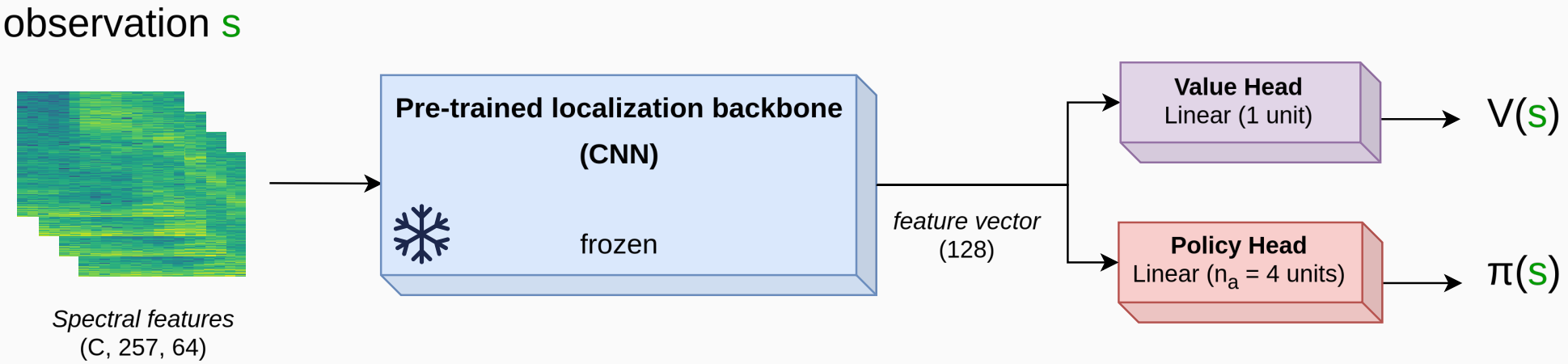
Problem: WER can't be computed at the environment run-time.

$\rightarrow$ 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 1rchitecture

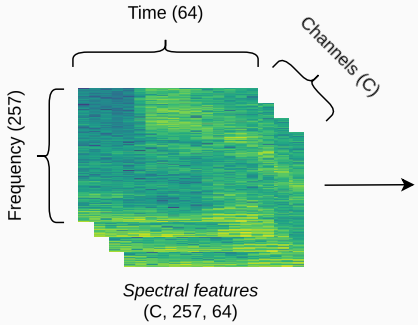


Two-stage training:

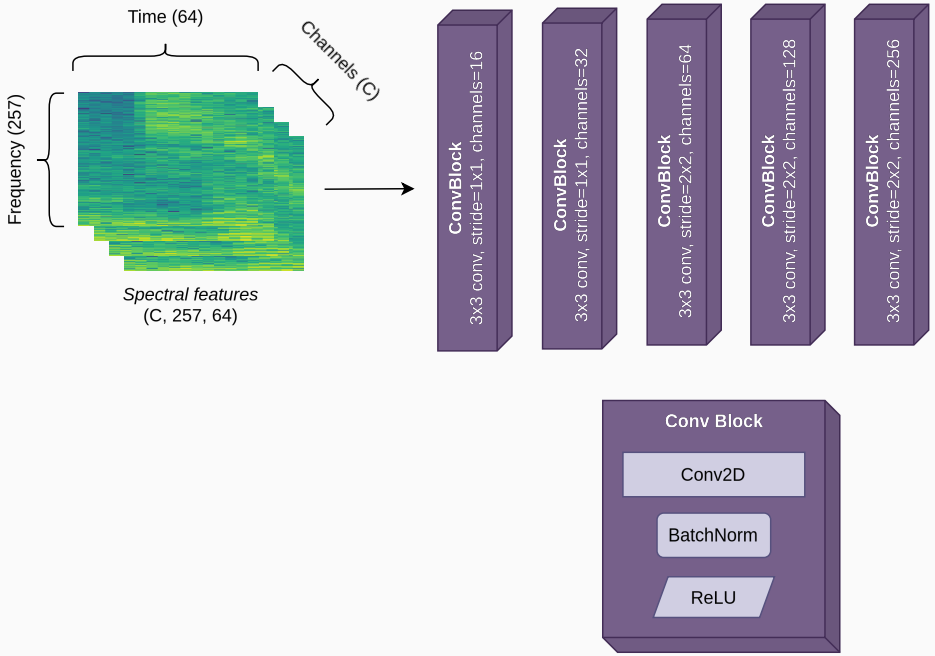
1. Train the backbone on a supervised single-source localization task
2. Train the value and policy heads with PPO [1]

[1] Schulman et al., "Proximal policy optimization algorithms," *arXiv preprint*, 2017.

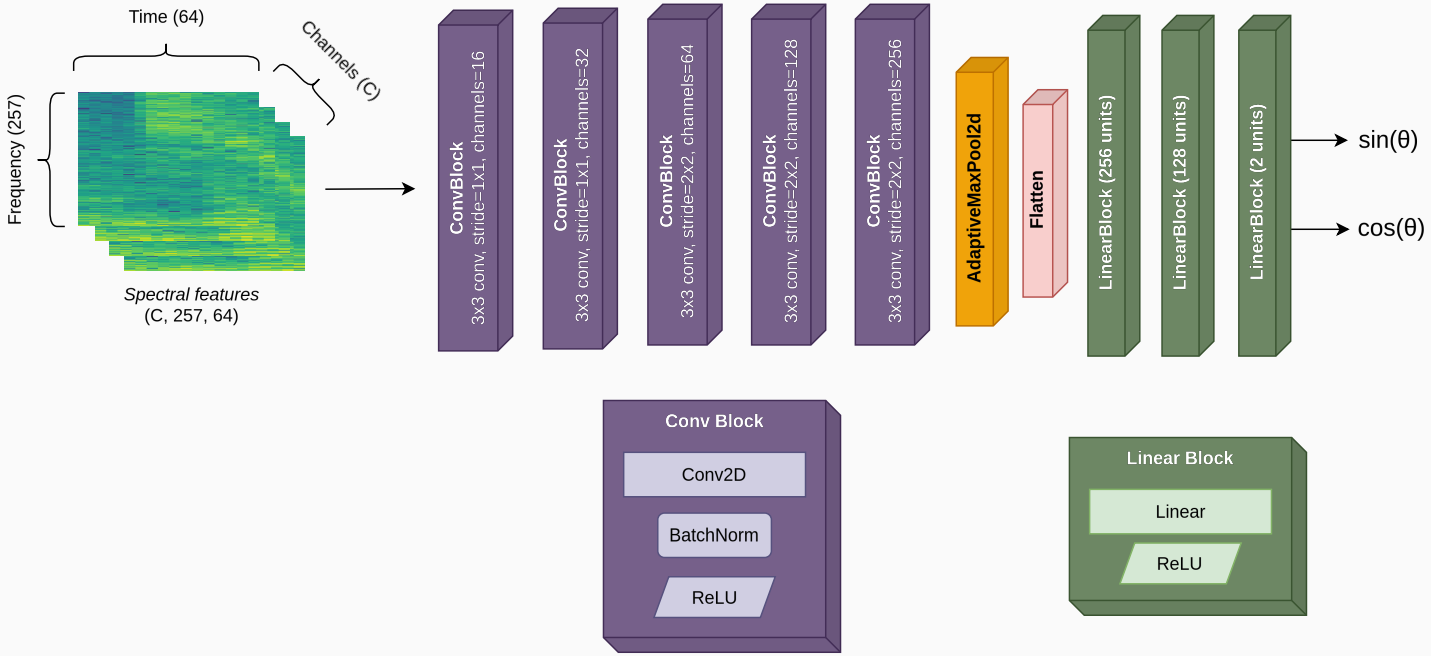
SSL Backbone Architecture



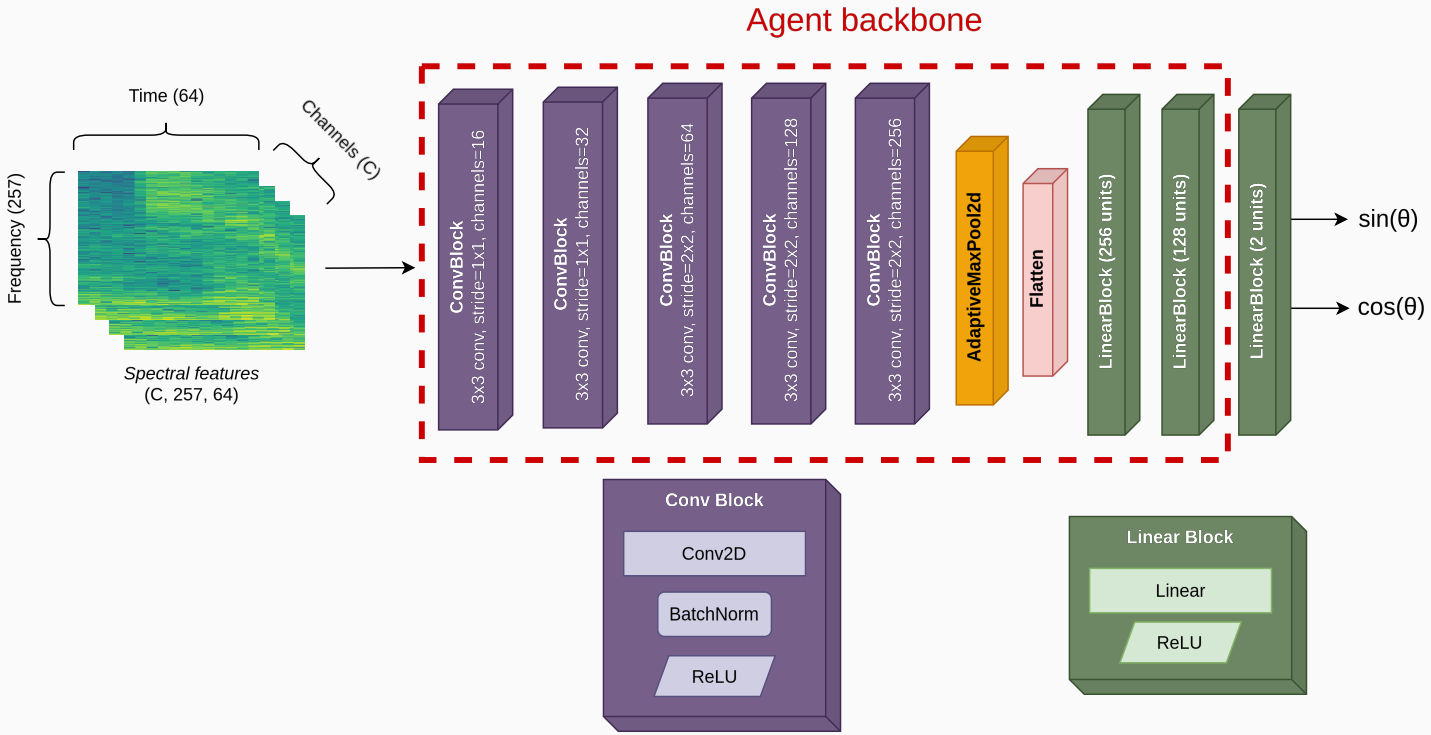
SSL Backbone Architecture



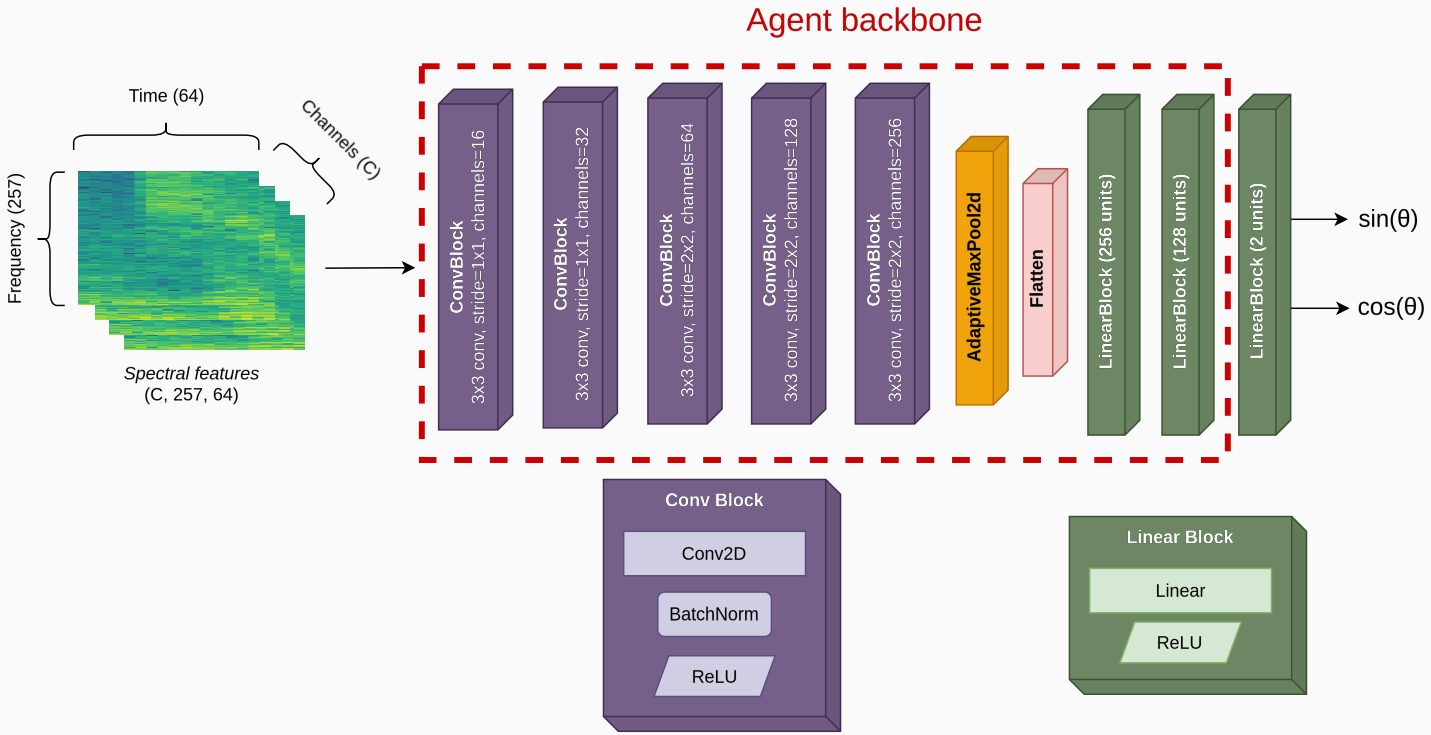
SSL Backbone Architecture



SSL Backbone Architecture



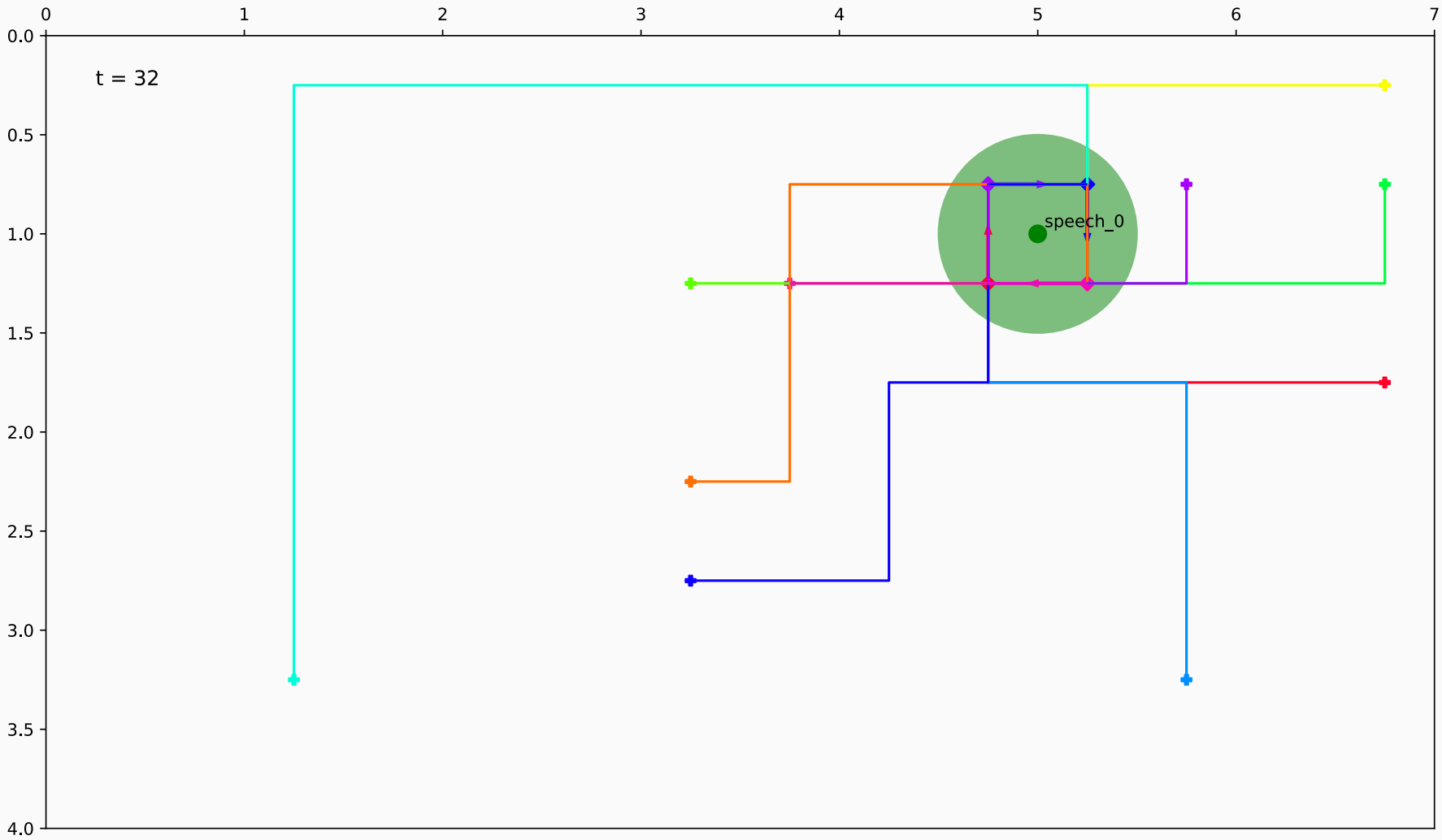
SSL Backbone Architecture



Training loss:

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

Agent Trajectories



Metrics

Undiscounted cumulated reward:

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

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:

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

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:

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

Results

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

Summary

- 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.

Summary of Contributions

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

Summary of Contributions

1. Design and implementation of an holistic **simulation library** for modelling 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 modelling 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 modelling 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.

Main Limitations

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

Main Limitations

- 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 **simplifying assumptions** were made in the different tasks.
  - Static sources,
  - Simplistic robot acoustic modelling: free-field microphone array (no HRTF)
  - Limitation to 2D geometric settings: no consideration for the elevation component

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

Main Limitations

- 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 **simplifying assumptions** were made in the different tasks.
  - Static sources,
  - Simplistic robot acoustic modelling: free-field microphone array (no HRTF)
  - Limitation to 2D geometric settings: no consideration for the 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.

Perspectives

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

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.

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.
- **Model efficiency and generalization:**
  - Investigate RL agents' lack of generalization.
  - Solve more diverse and challenging MDPs (changing room geometries, moving sources, noisy conditions, etc.)

Thank you!