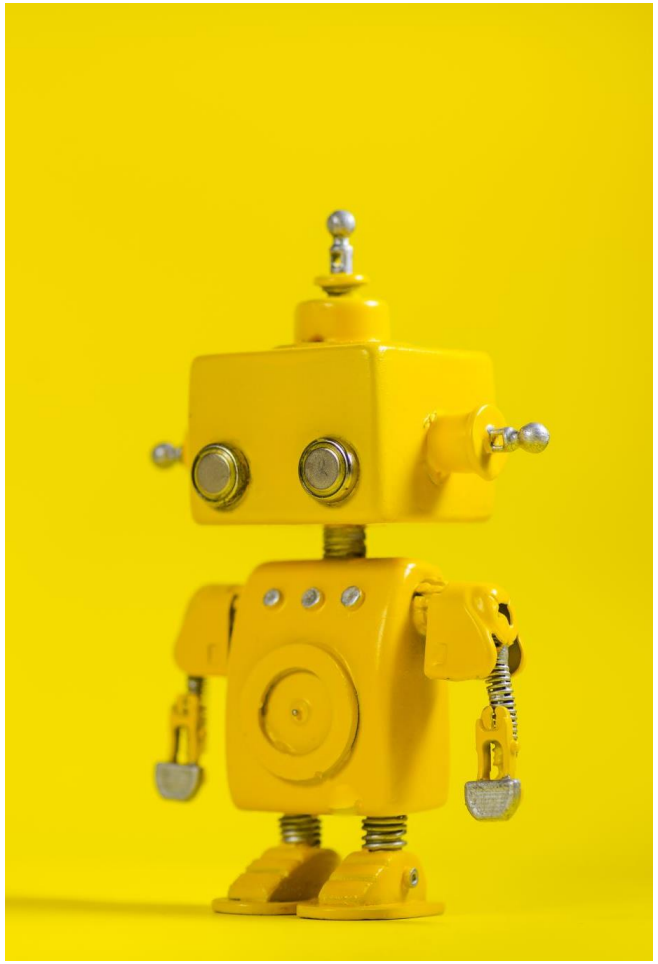
Audio-embodied Indoor navigation

- Yulang Wu, Yifan Lin, Xuanchen Zhou

Outline

- Introduction
- Method
- Synthetic examples
- Discussion and Conclusion

1. Introduction



Indoor/Household robots

- Applications:
  - Cleaning
  - Personal assistant
  - Entertainment
  - Security
  - Educational

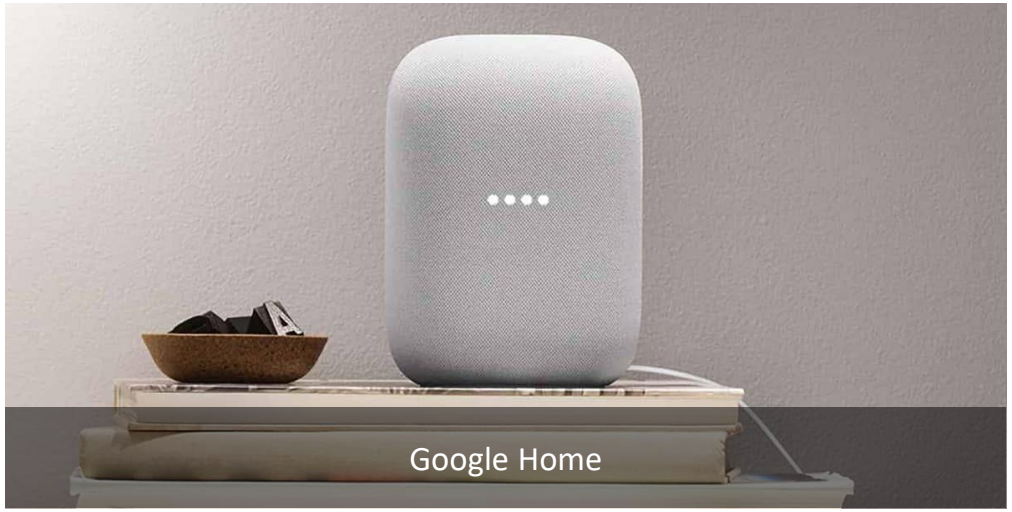
Cleaning Robots

Roomba vacuum cleaner, the LG Hom-Bot Turbo+

HOBOT-268 window cleaning robot

Assistant Robots

Amazon’s Echo


Google Home

Entertainment robots

- Robotic Pets: These robots are designed to mimic the appearance and behaviors of real animals, such as dogs, cats, or birds.

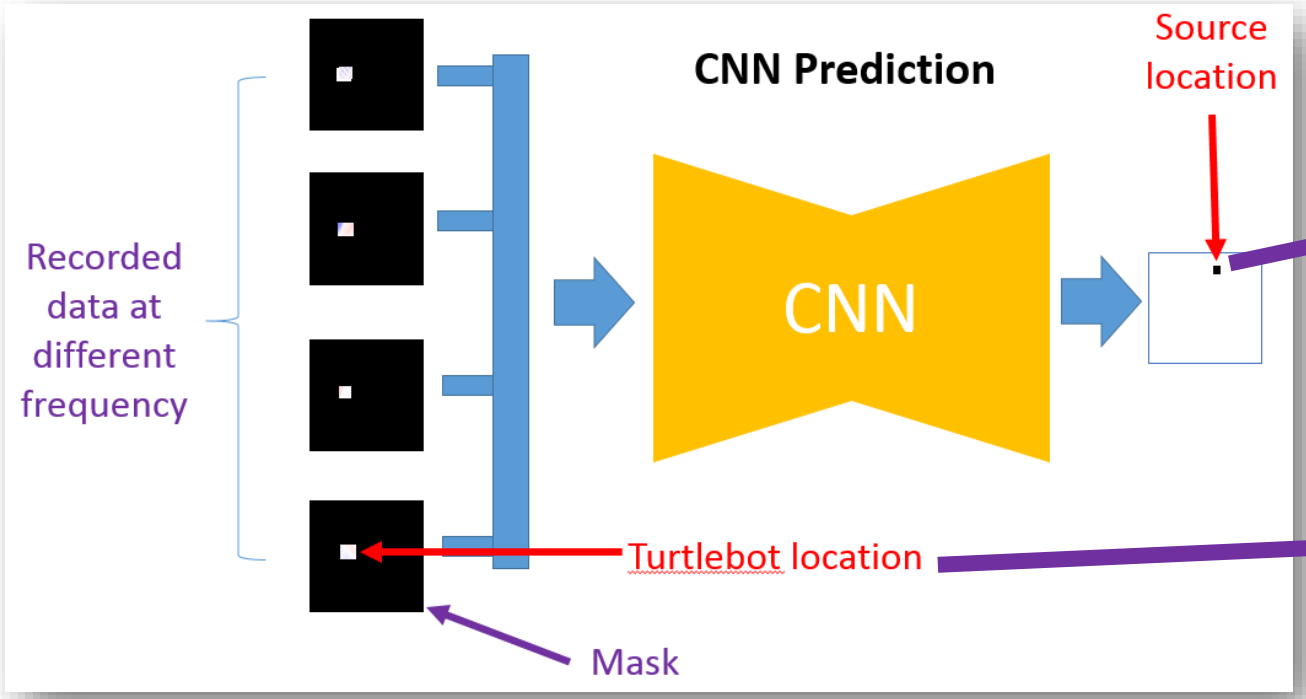
Our Purpose

- Given a known scene (map), a robot should efficiently and correctly detect and locate the sound source and then move to the source location.

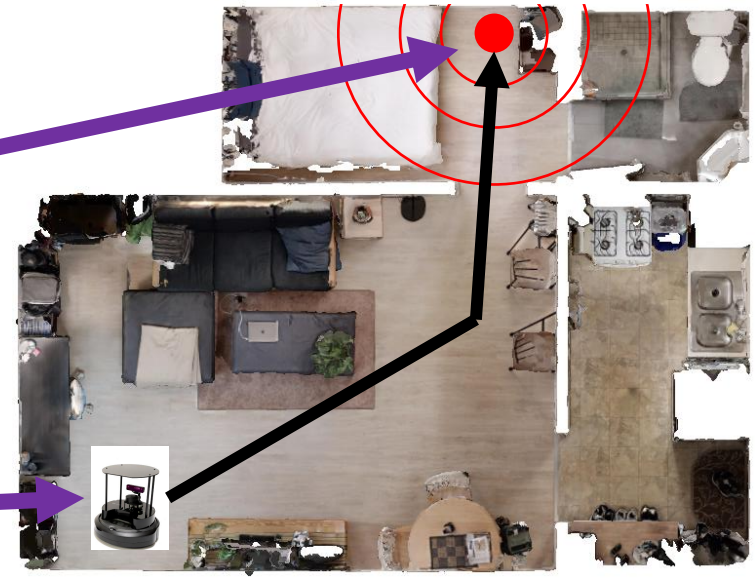
2. Method

Audio-embodied navigation

Step 1: Estimate source location from recorded data



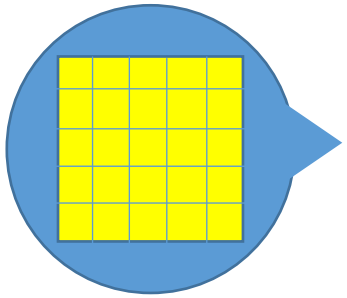
Step 2: Find shortest path



2.1 Source location estimation

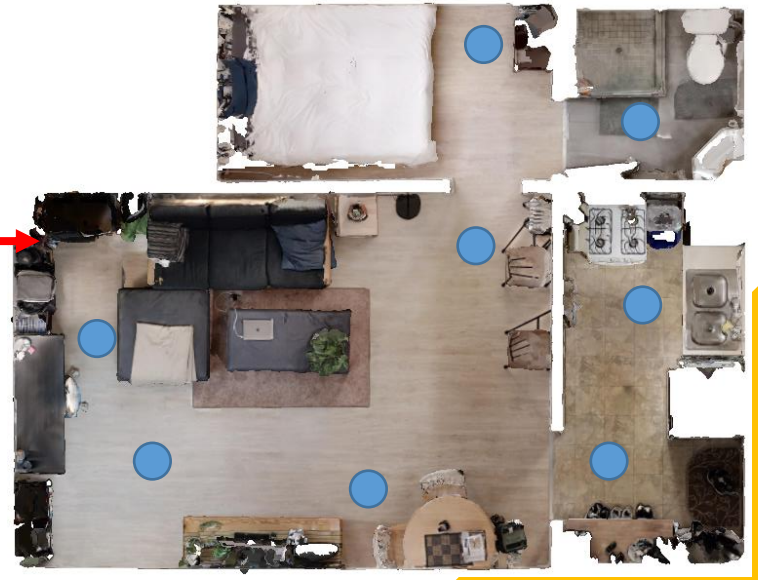
1) Robot and scene

Turtlebot



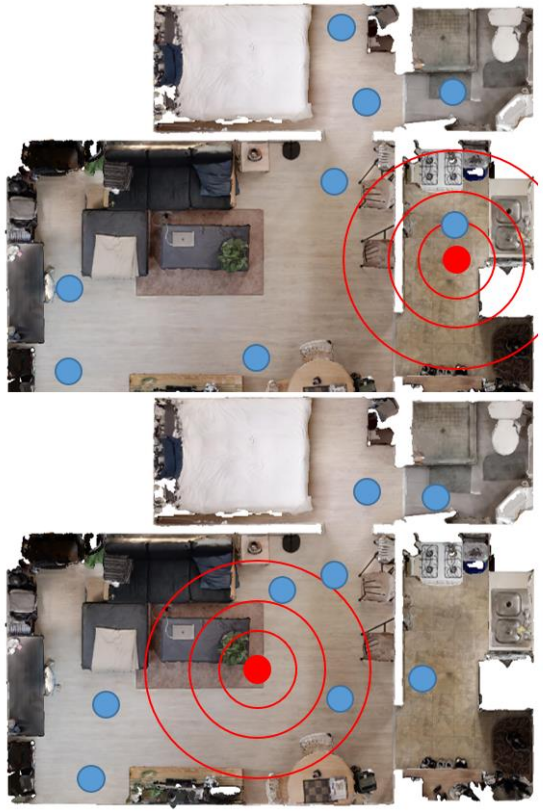
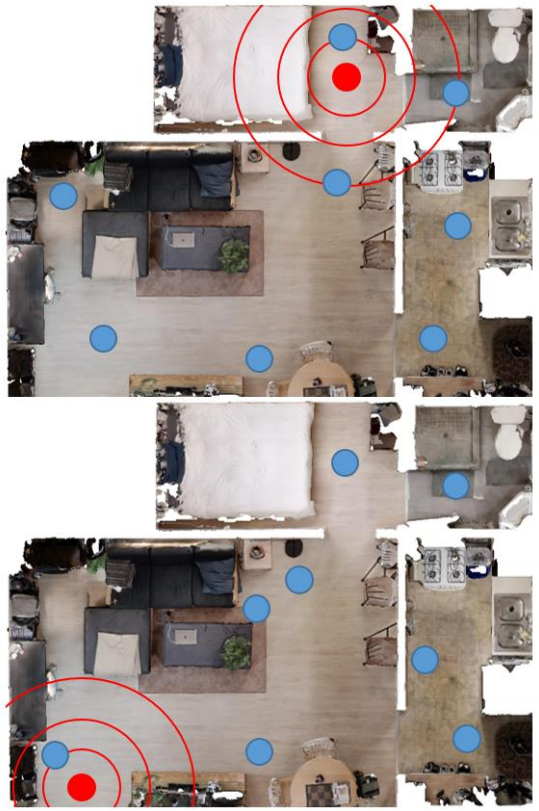
25 microphones
Assumed to be placed
on the top

Turtlebot randomly
walks on the scene



2.1 Source location estimation

2) Training data collection

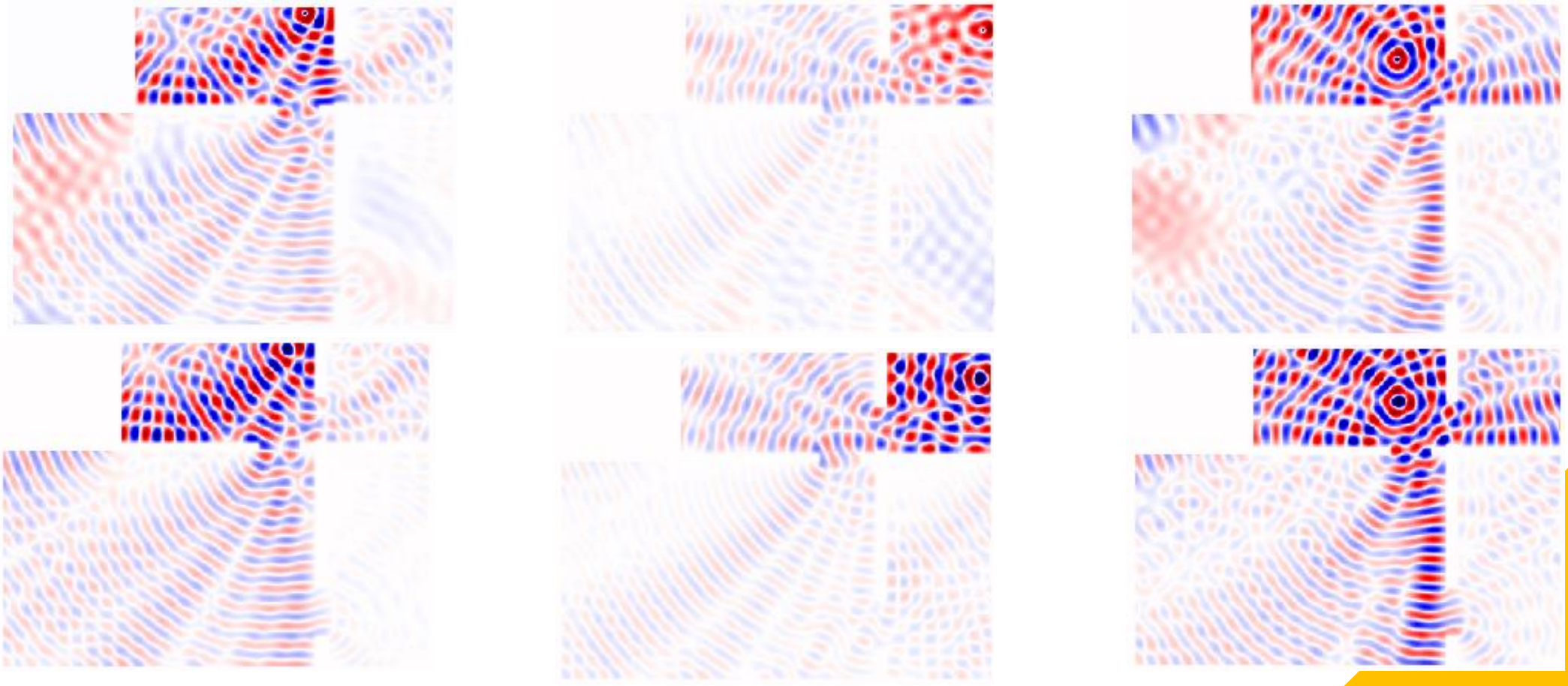




2.1 Source location estimation

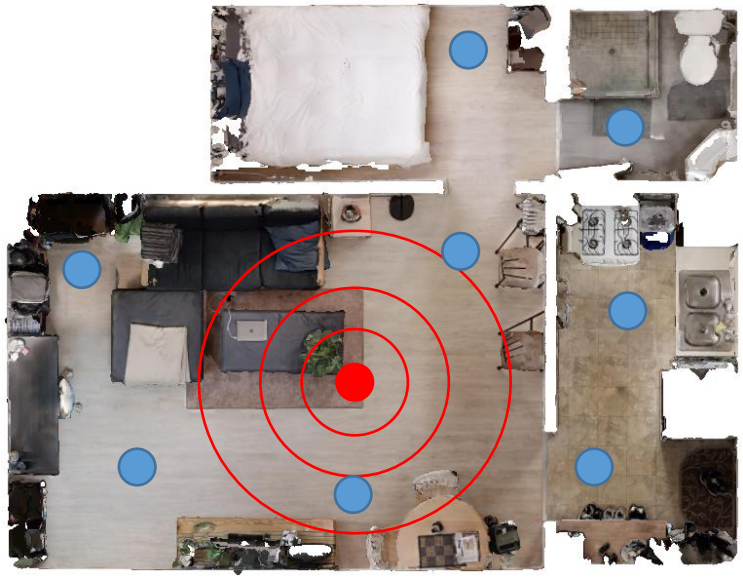
3) Features extracted from the recorded data

- Frequency-domain acoustic data
- Illumination map

Frequency-domain data $d(\vec{x}, \omega) = FFT(d(\vec{x}, t))$

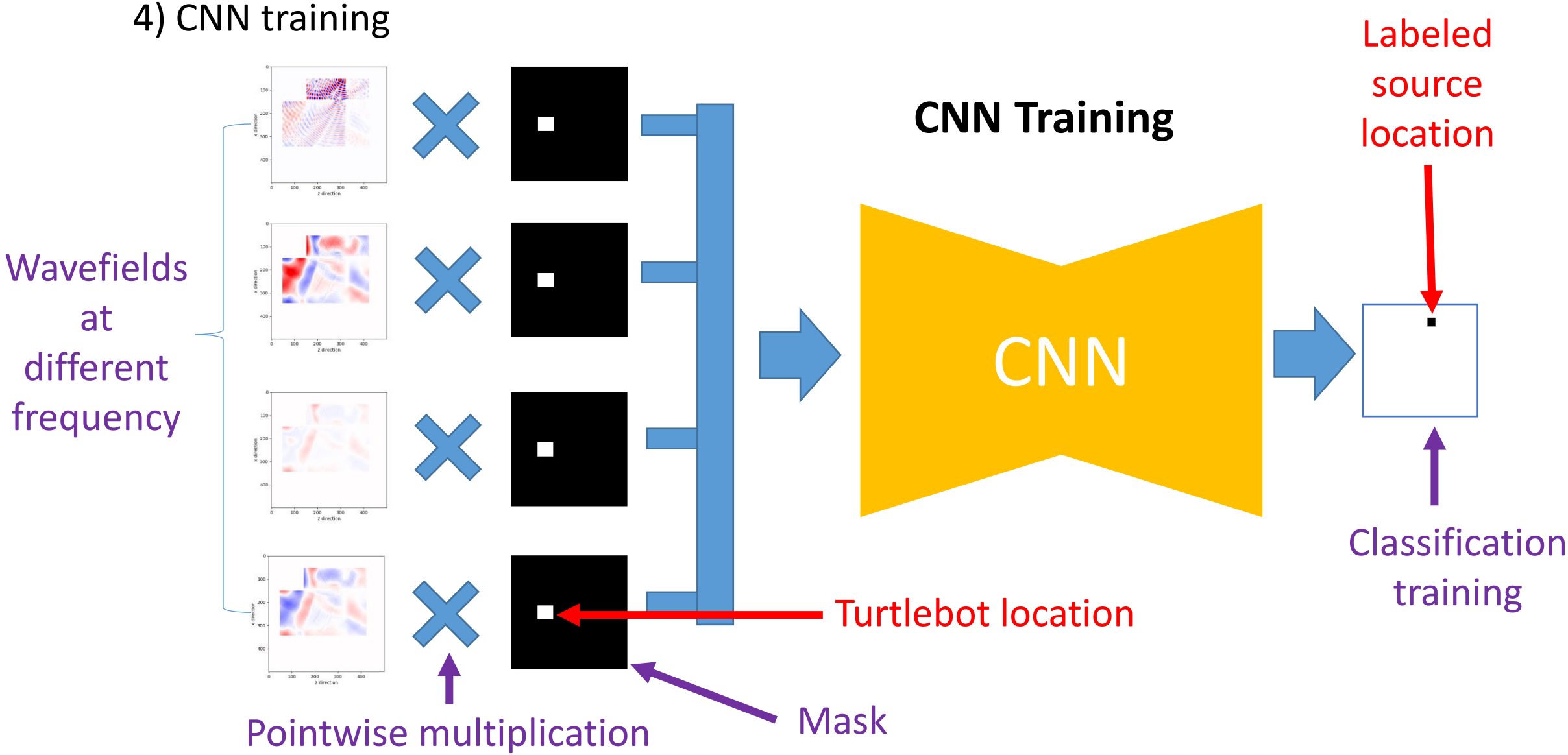


Illumination (energy) of the wavefield $\int_o^T [d(\vec{x}, t)]^2 dt$



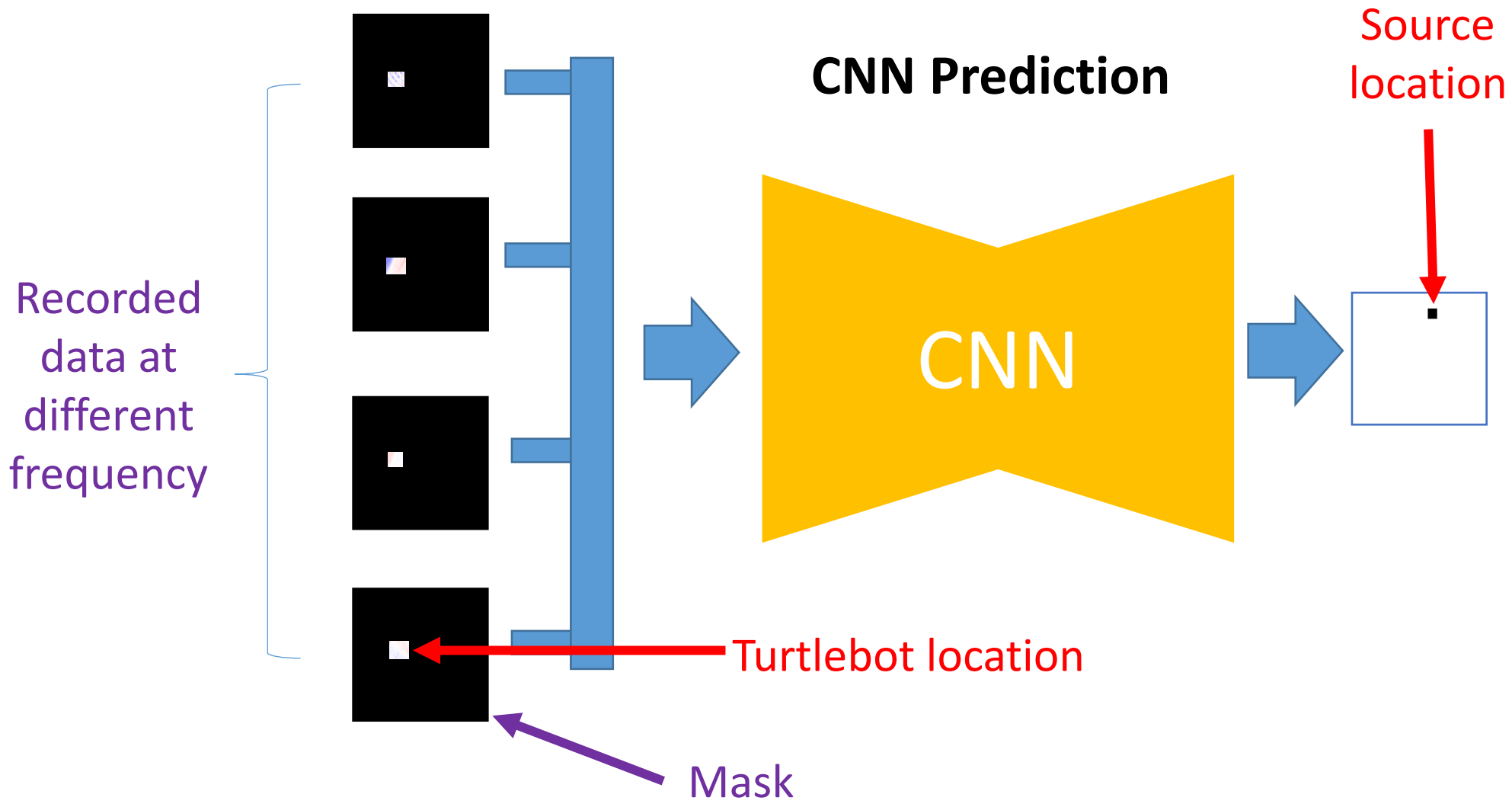
2.1 Source location estimation

4) CNN training

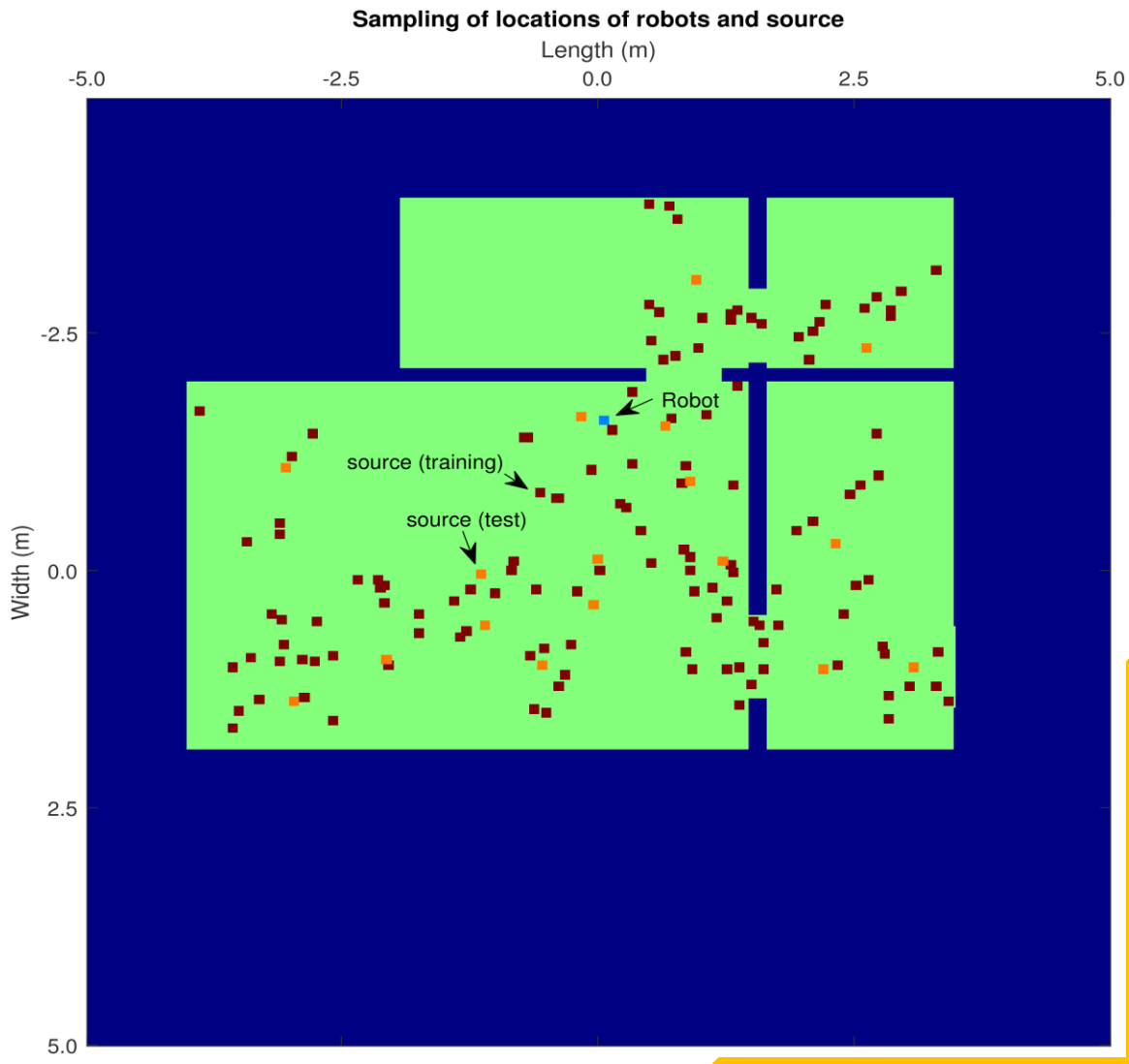


2.1 Source location estimation

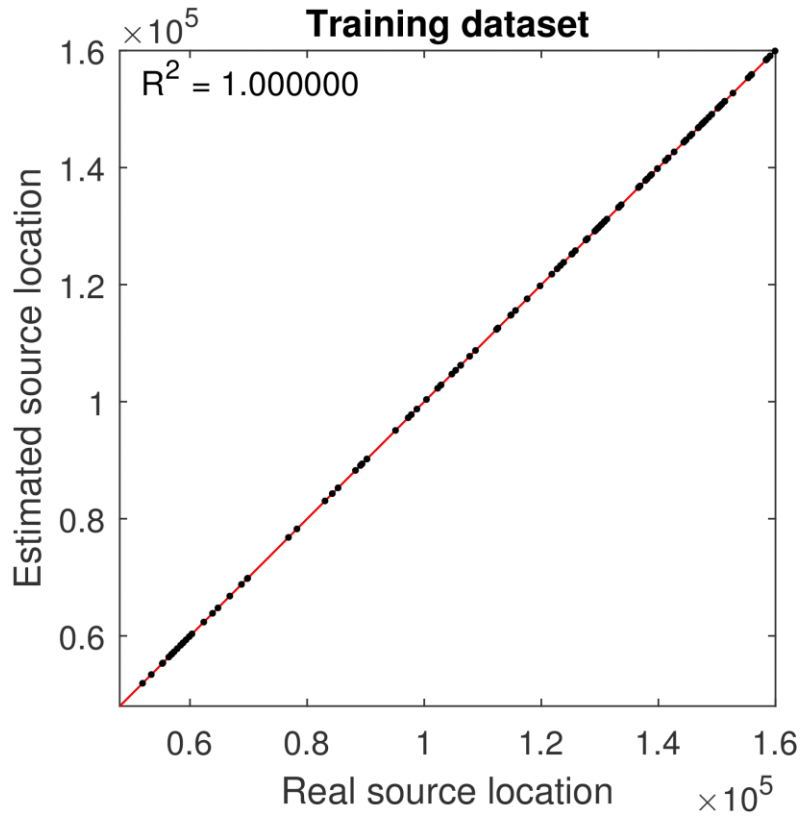
5) CNN prediction

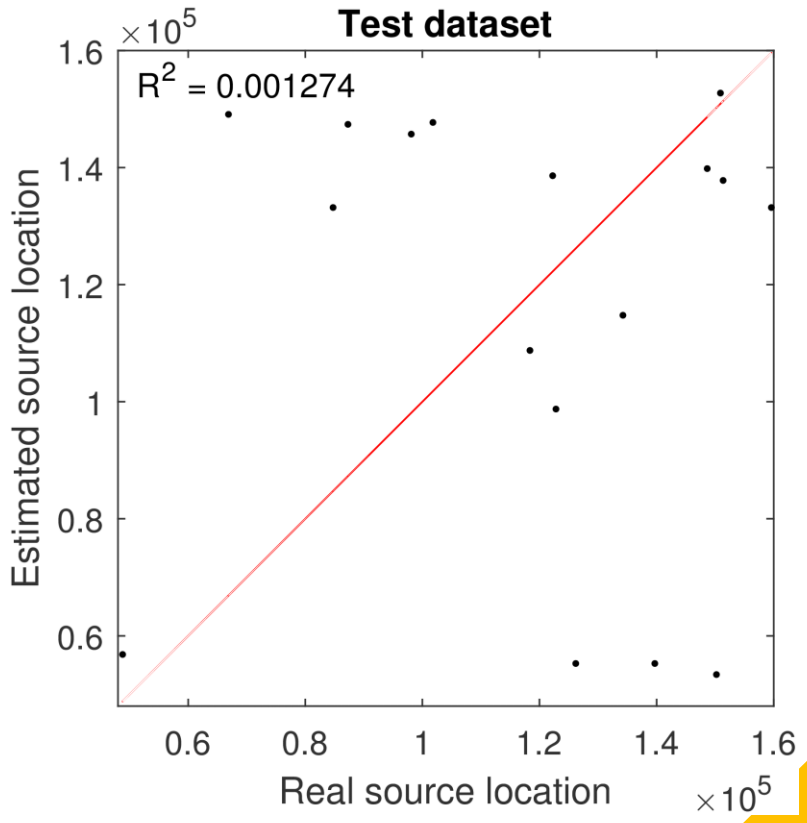


The source locations in both the training (the red squares) and test (the orange squares) dataset and the robot's location (the light blue square)



The training (the left panel) and test (the right panel) errors





2.2 Path planning

A* search algorithm

find the shortest path from robot position to target position

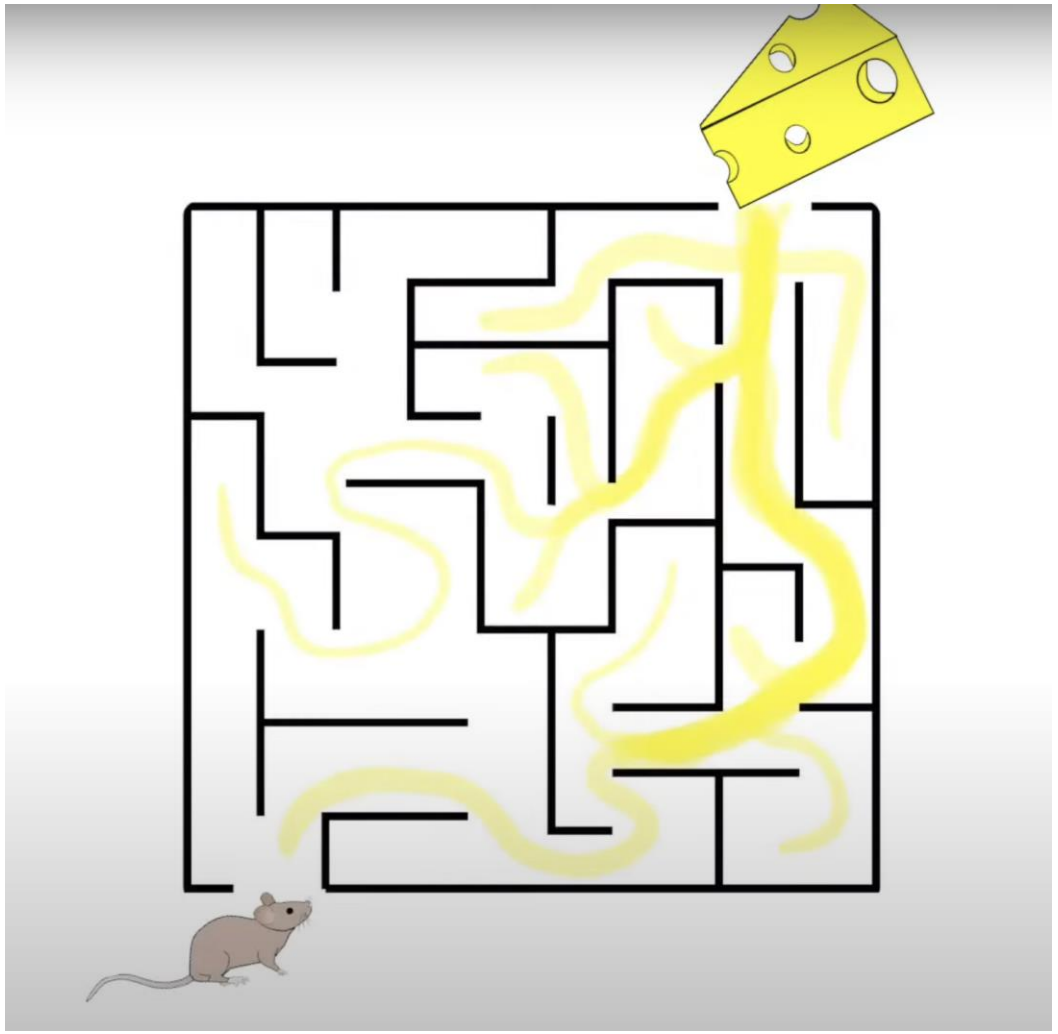
A* algorithm is a pathfinding algorithm that uses a heuristic function to find the optimal path from a start node to a goal node.

$F(x) = d(x) + h(x)$

d(x): current cost

h(x): estimated remaining cost

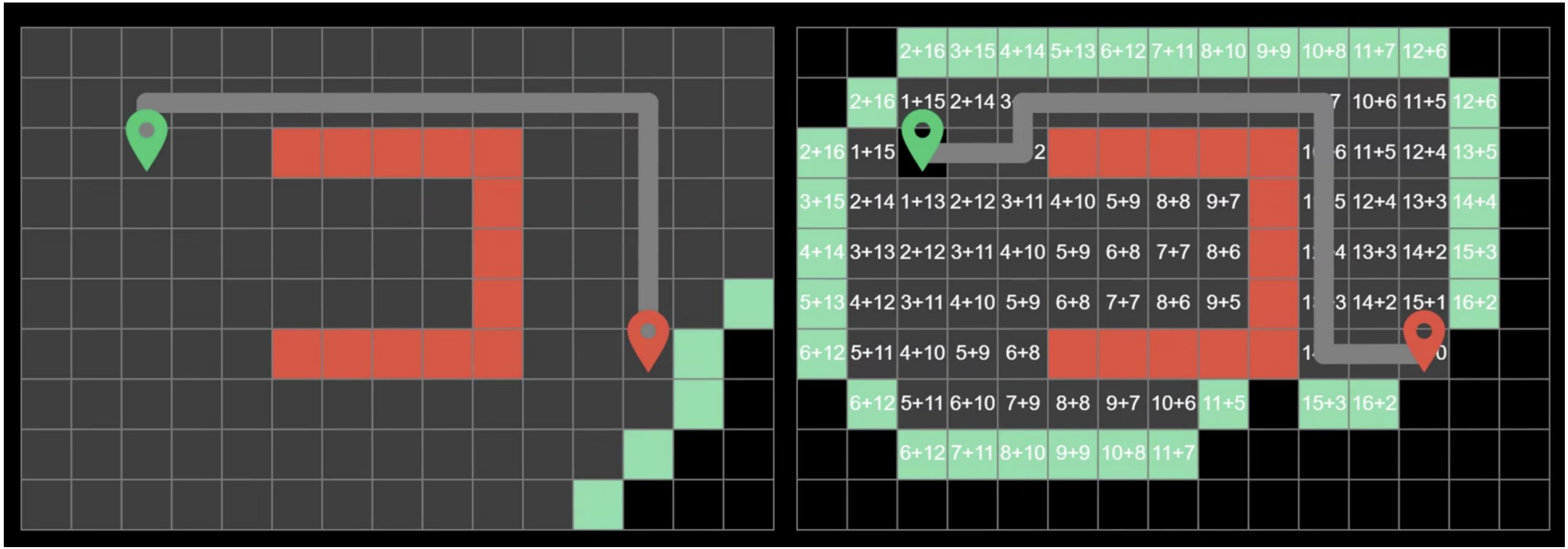
Euler/Manhattan distance



2.2 Path planning

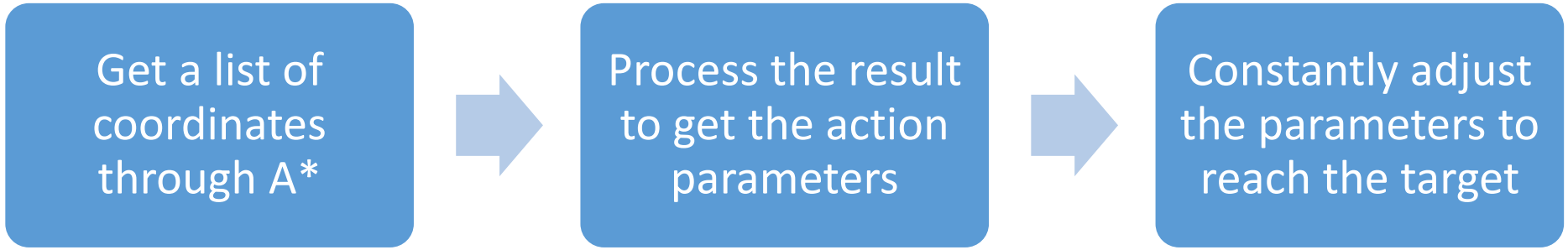
☐ A* search algorithm

Comparison of normal BFS and A*



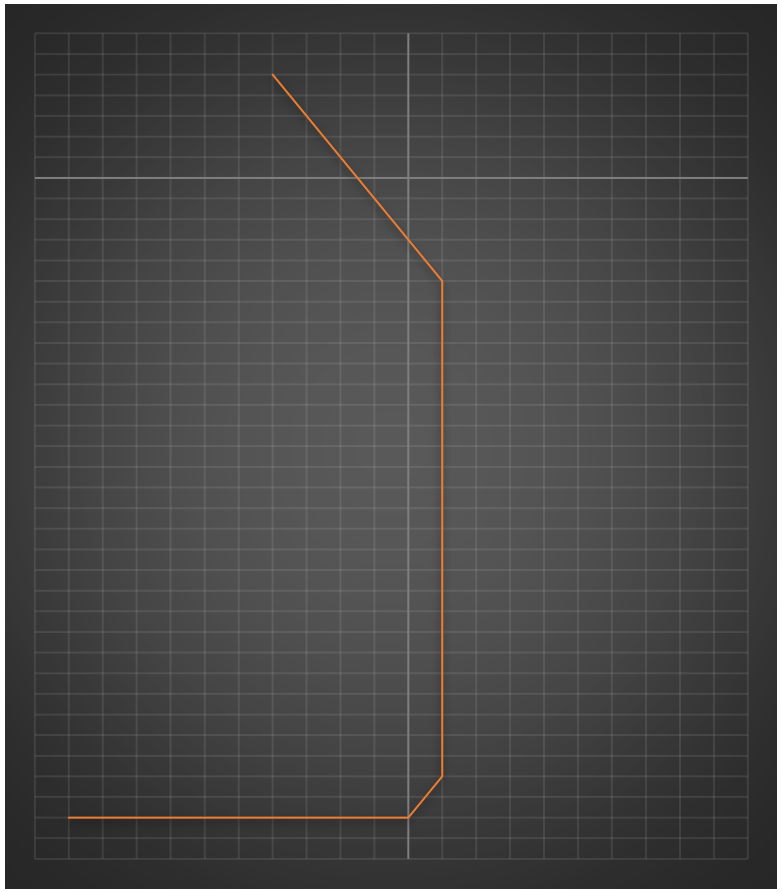
2.2 Path planning

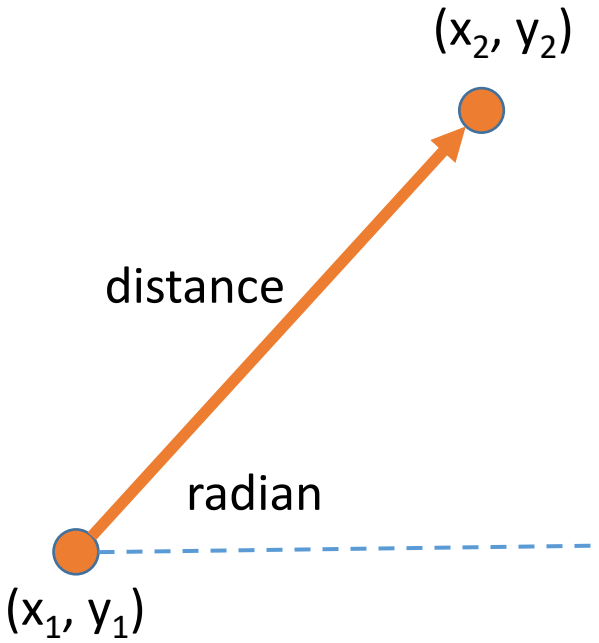
☐ Implement it in iGibson



2.2 Path planning

- **Implement it in iGibson**





action =
(distance, radian)

env.step()

4. Discussion and Conclusion

- **Limitations and requirements:**
  - Map is assumed to be known
  - Require random sampling for CNN training
  - Require two acoustic signals sent from the target at different time to get the time window
  - Require unchanged audio sequence

- **Potential applications:**
  - Hide-and-seek
  - Emergency call
  - Private assistant

Thank you!