# Robust Robot Navigation Against Imperfect Sensor Data

Final Project Presentation

Team 1

Hyeongyeol Ryu, Minsung Yoon, Sebin Lee

20203397 20203408 20203474

2020.12.08

# Table of contents

# Table of contents

# Introduction

- These days, many mobile robots are around us


**Mobile service robots**

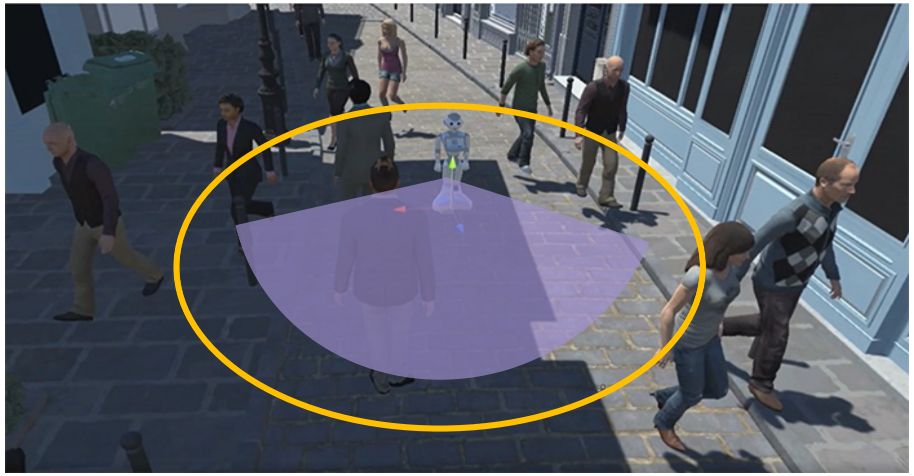
# Introduction

- These days, many mobile robots are around us
  - Autonomous navigation is one of the essential abilities for them


**Autonomous navigation using LiDAR**

# Introduction

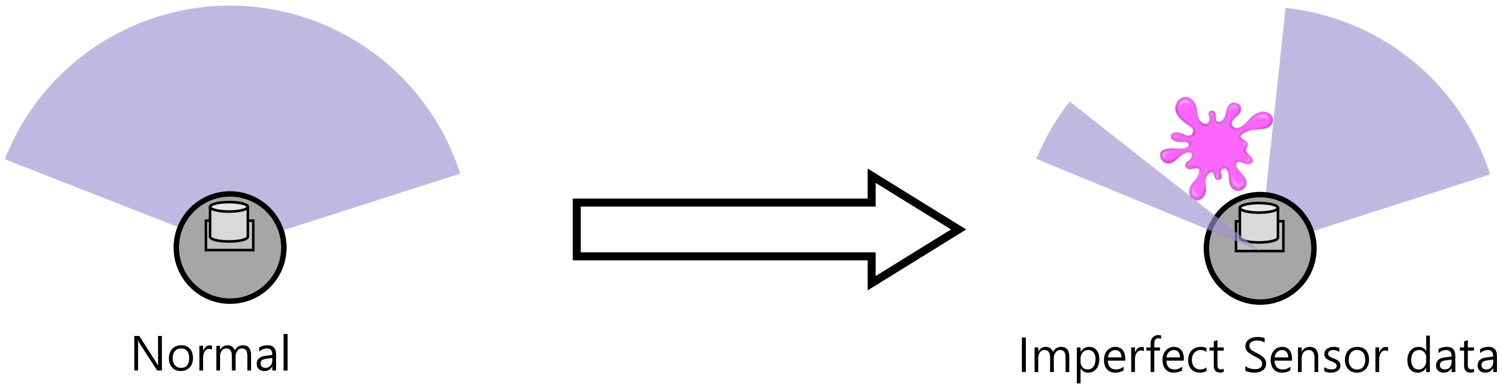
- What if the robot cannot get perfect sensor data because of
  - Dynamic external disturbance



**This might cause catastrophic actions in safety-critical tasks such as robot navigation in real-world**

# Research goal

- Goal: Propose robust navigation method dealing with imperfect sensor data



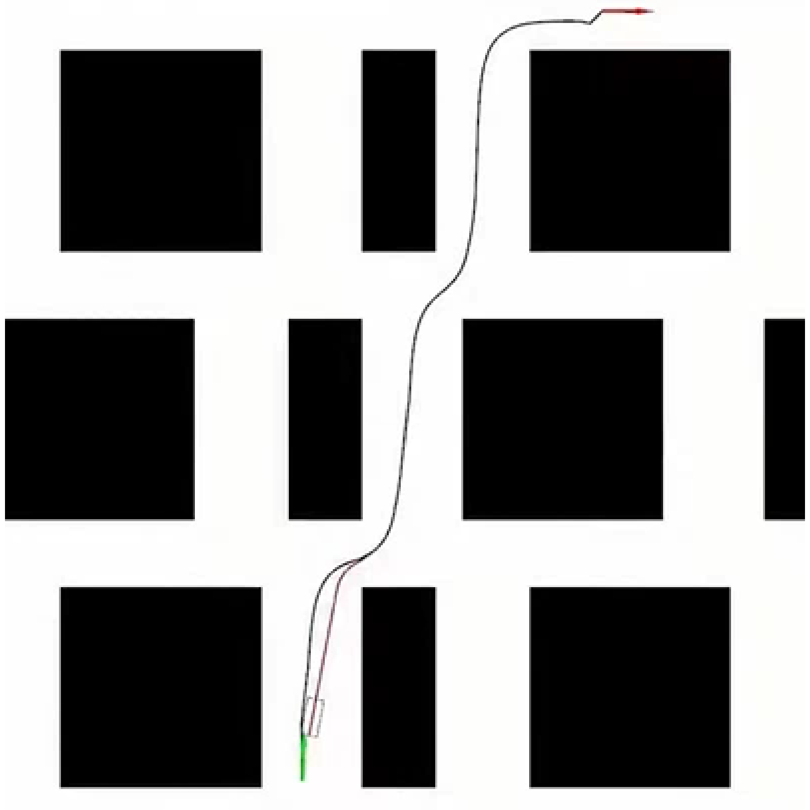

**Modeling of imperfect sensor data**
**&**
**Joint learning with RL**

# Table of contents

- Introduction & Research goal
- Background
- Our approach
- Experimental Result
- Conclusion
- Role assignment

# Background: Navigation

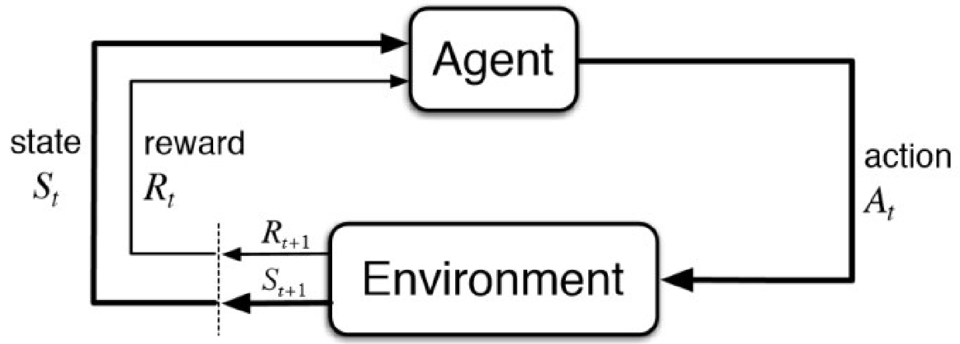
- **Navigation** (Global planning + **Local planning**)



Black Line: Global path
Red Line: Navigation (Local planning)

# Background: Actor-Critic

- Basic Concept of Reinforcement Learning

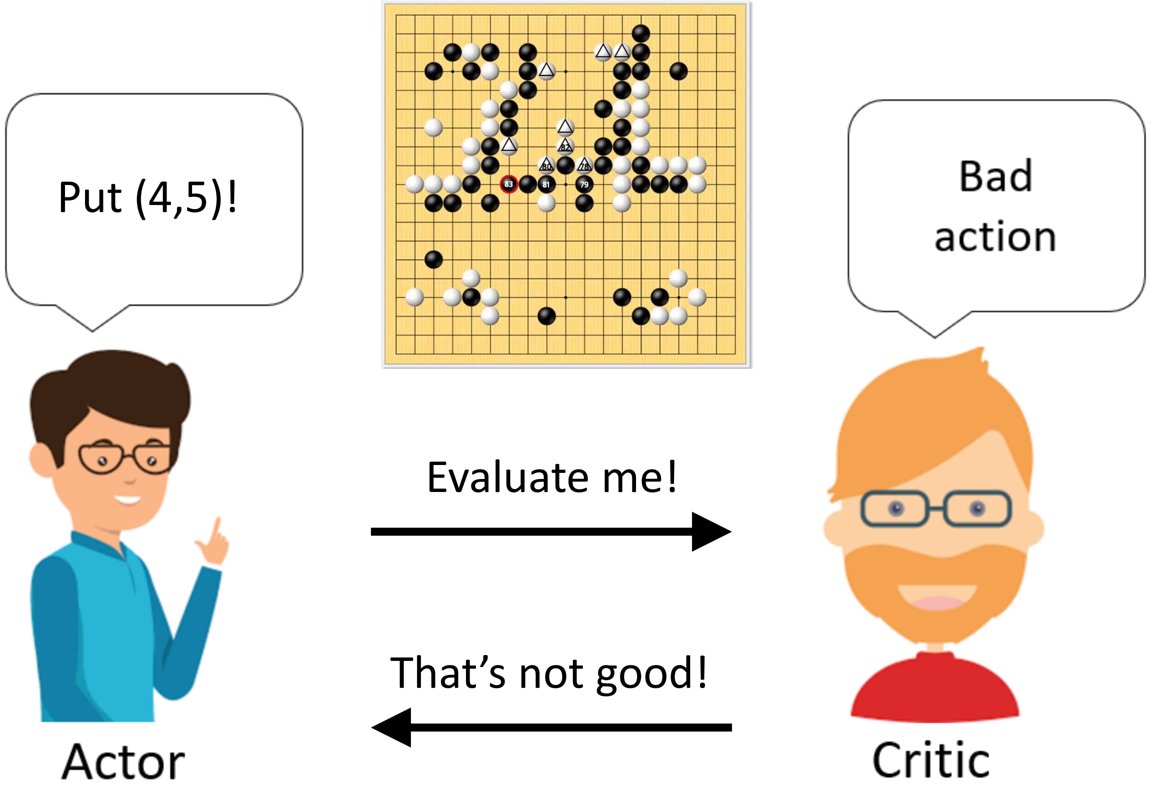



- Our goal is to find state/action maximizing Q value

$$\underbrace{Q^\pi(s_t, a_t)}_{\text{Q value for that state given that action}} = \underline{E}[\underbrace{R_{t+1} + \gamma R_{t+2} + \gamma^2 R_{t+3} + \ldots}_{\text{Expected discounted cumulative reward ...}} | \underbrace{s_t, a_t}_{\text{given that state and that action}}]$$

# Background: Actor-Critic

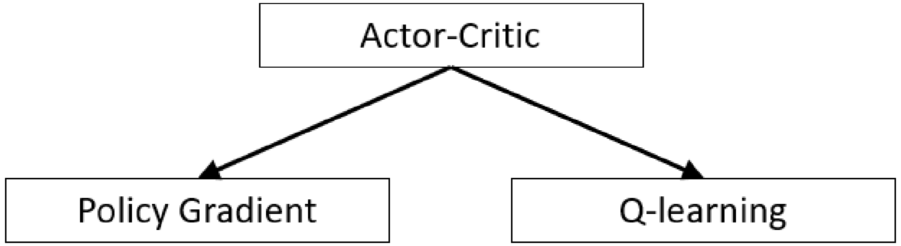
- Actor-Critic in RL

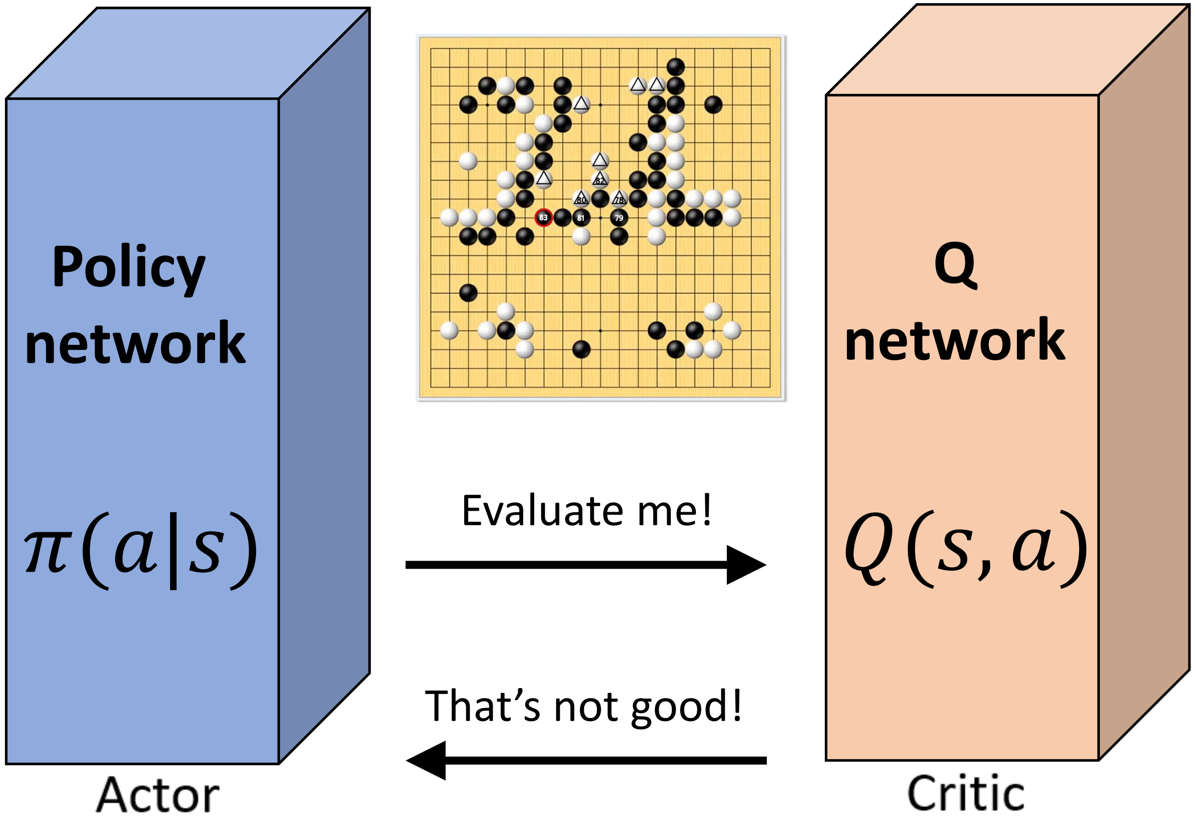



- **Actor**: A player that decides on an action to take
- **Critic**: A coach that criticizes the action that the actor selected, providing feedback on how to adjust

# Background: Actor-Critic

- Actor-Critic in RL




- **Actor**: A player that decides on an action to take (Policy network)
- **Critic**: A coach that criticizes the action that the actor selected, providing feedback on how to adjust (Q-network)

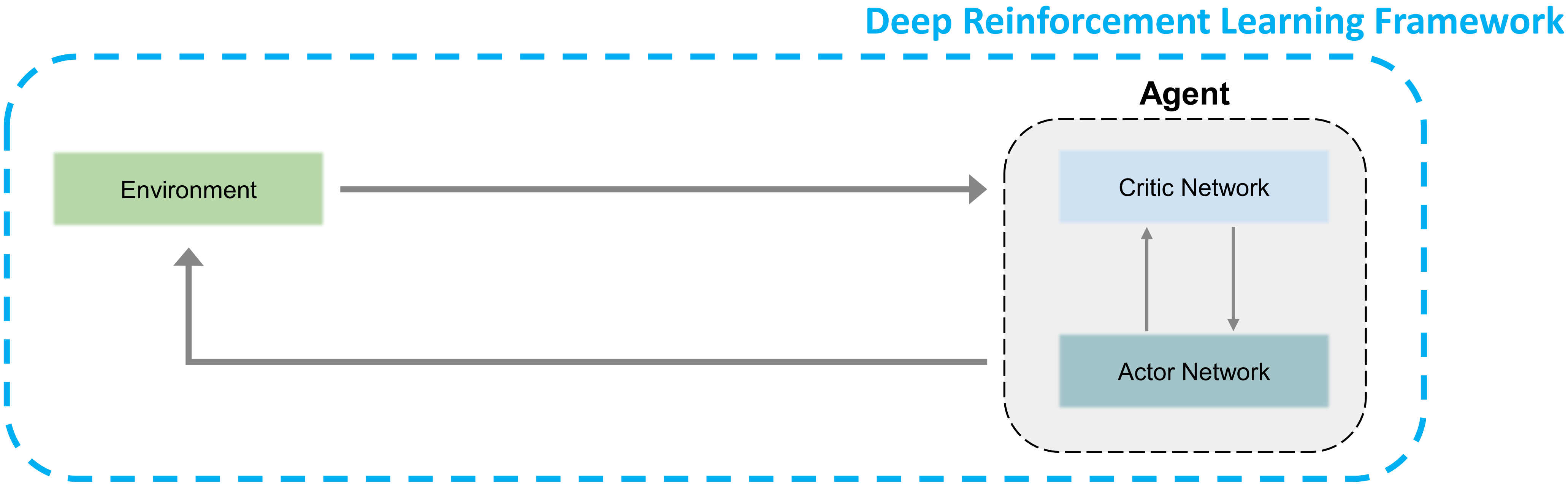
# Table of contents

# Our approach: Overview

- **Key points:**
  - Deep Reinforcement Learning
  - Various network architectures
  - Imperfect sensor data modeling

# Our approach: Overview

- **Key points:**
  - **Deep Reinforcement Learning**
  - Various network architectures
  - Imperfect sensor data modeling



# Our approach: Deep Reinforcement Learning

- **Problem Setting**
  - Environment: Static/Dynamic objects & Imperfect sensor data
  - Robot: Differential drive robot
  - Action: Linear & Angular velocity
  - Reward:

$$R = R_{Arrival} + R_{Collision} + R_{Distance} + R_{Heading}$$

# Our approach: Deep Reinforcement Learning

- **Reward shaping:** $R = R_{Arrival} + R_{Collision} + R_{Distance} + R_{Heading}$

  - $R_{Arrival} = \begin{cases} +5 & \textit{if the robot arrives at the goal} \\ 0 & \textit{else} \end{cases}$

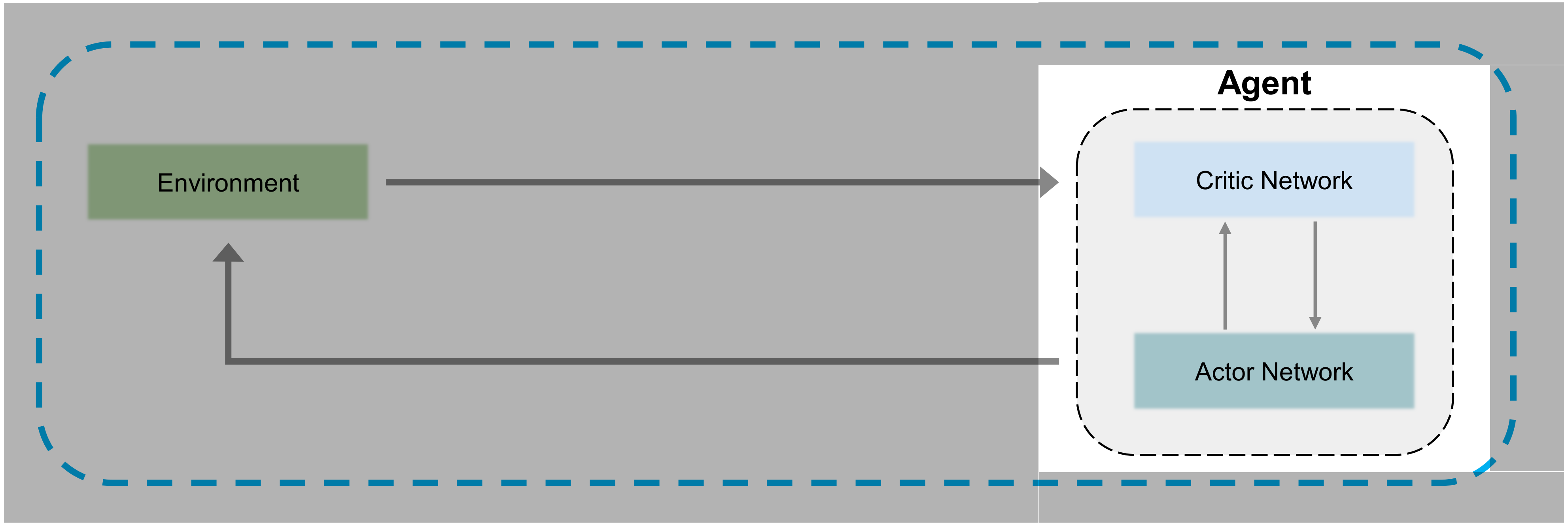
  - $R_{Collision} = \begin{cases} -5 & \textit{if the collision happens} \\ 0 & \textit{else} \end{cases}$

  - $R_{Distance} = 1.2 * (\|p^{t-1} - p_{goal}\| - \|p^{t} - p_{goal}\|)$ *Distance to the goal how closer than previous step*

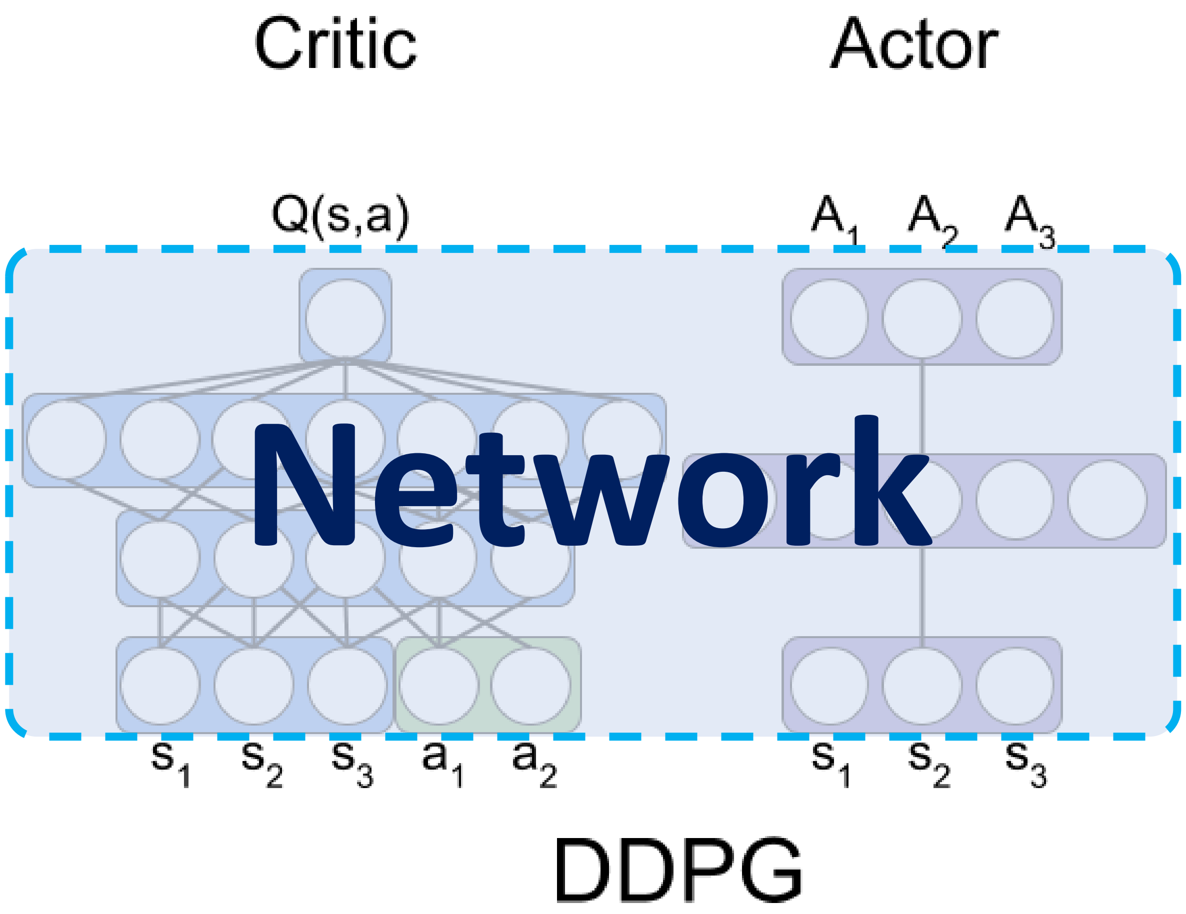
  - $R_{Heading} = -0.04 * \|Yaw - Relative\ angle\|$ *Degrees to face the goal*

# Our approach: Overview

- **Key points:**
  - Deep Reinforcement Learning
  - **Various network architectures**
  - Imperfect sensor data modeling



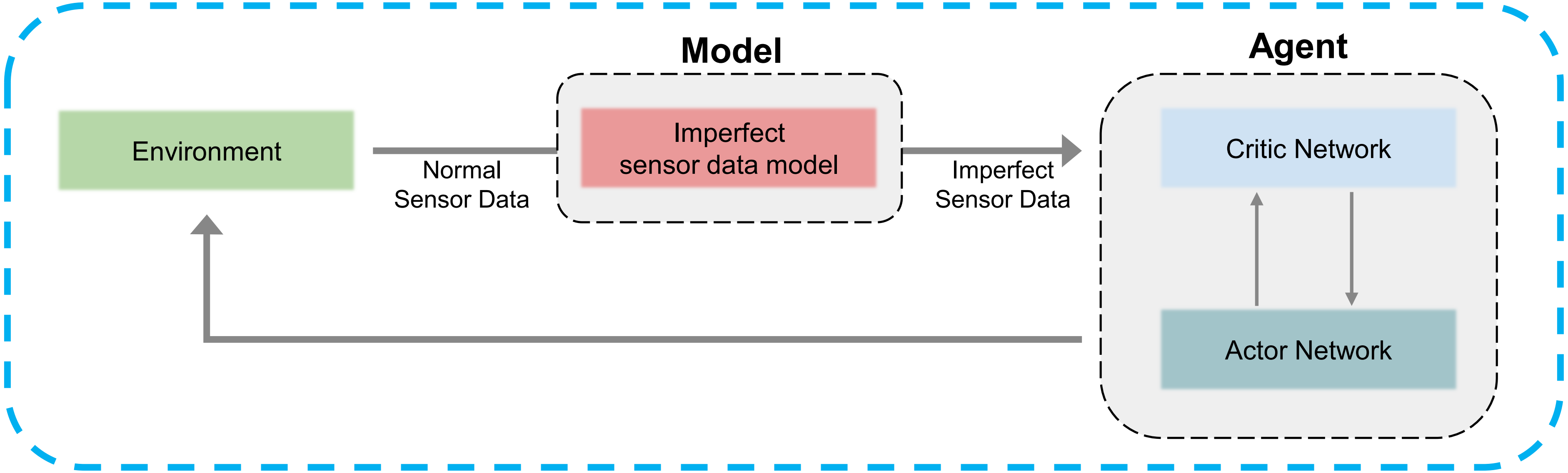
# Our approach: Various network architectures




| *Input* | *Network* |
|---|---|
| **Single frame sensor data** | MLP |
| | 1D CONV |
| | MLP + LSTM |
| **Stacked frame sensor data** | 2D CONV |
| | 2D Conv + LSTM |

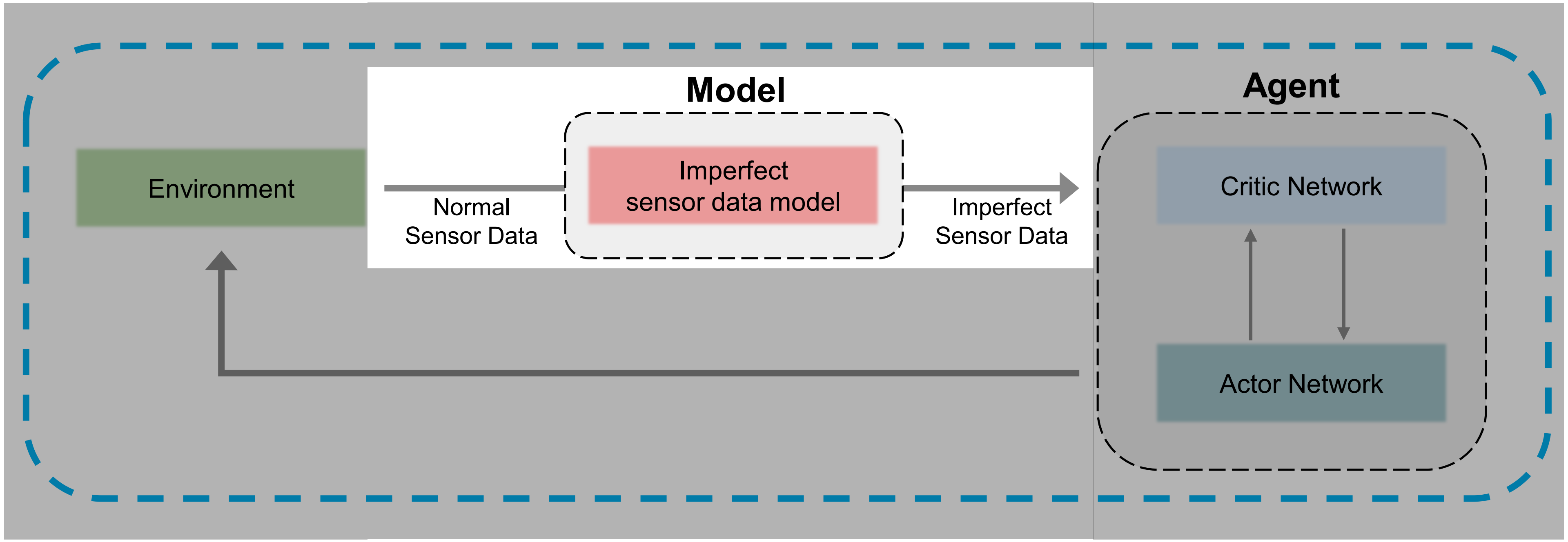
# Our approach: Overview

- **Key points:**
  - Deep Reinforcement Learning
  - Various network architectures
  - **Imperfect sensor data modeling**



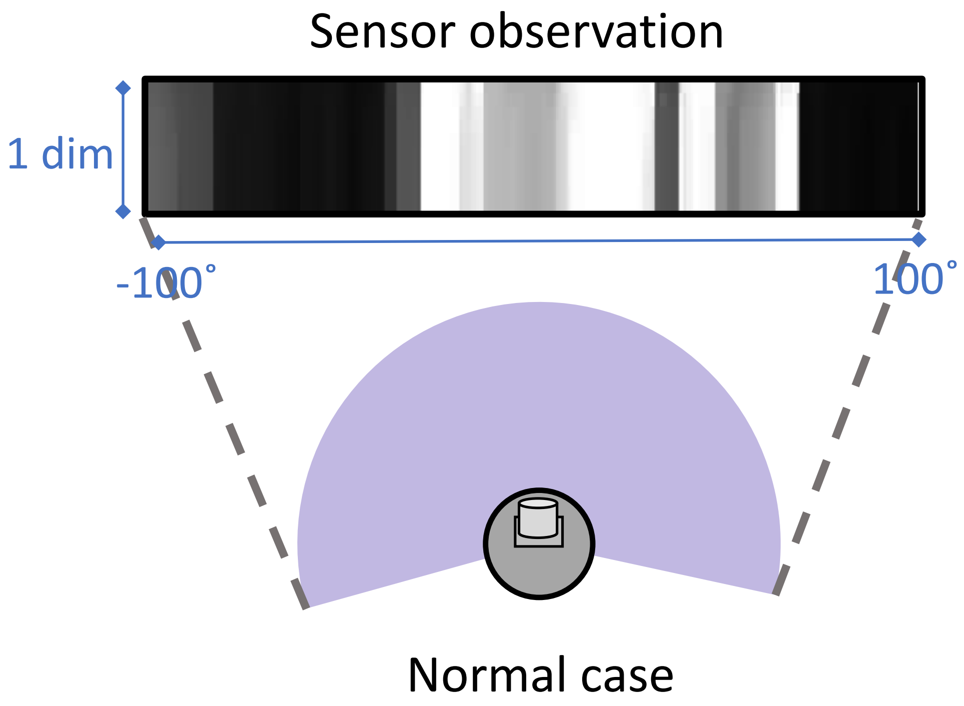
# Our approach: Overview

- **Key points:**
  - Deep Reinforcement Learning
  - Various network architectures
  - **Imperfect sensor data modeling**



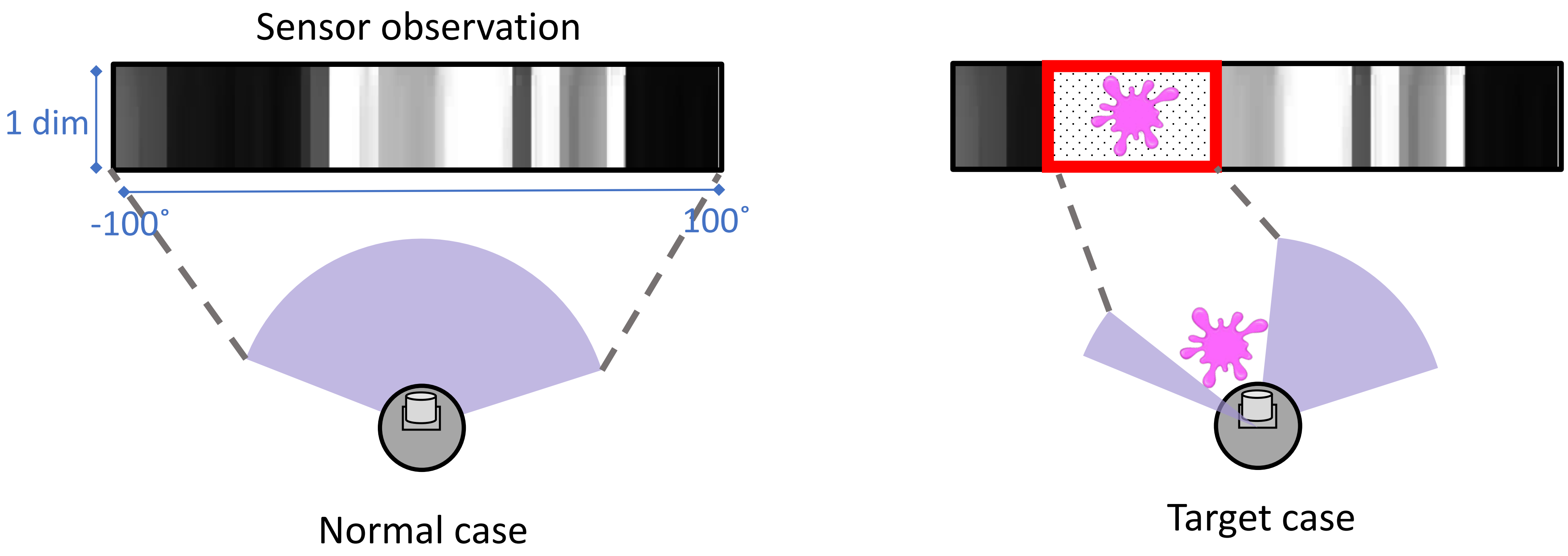
# Our approach: Imperfect sensor data modeling

- Sensor data visualization



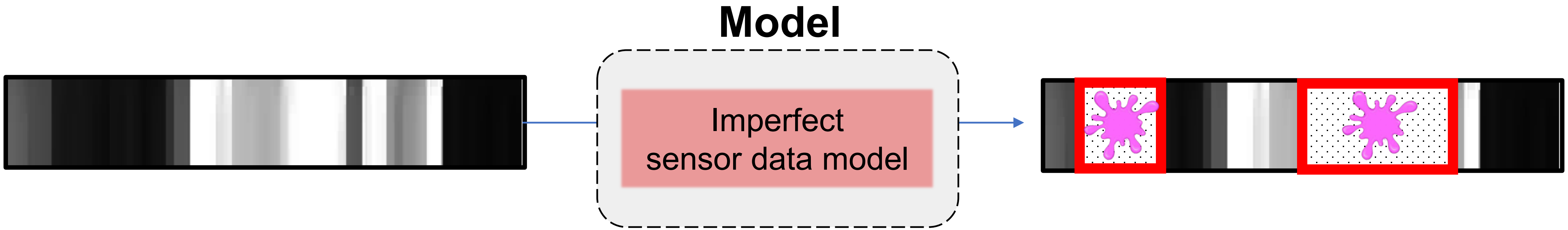
# Our approach: Imperfect sensor data modeling

- Modeling imperfect sensor data



# Our approach: Imperfect sensor data modeling

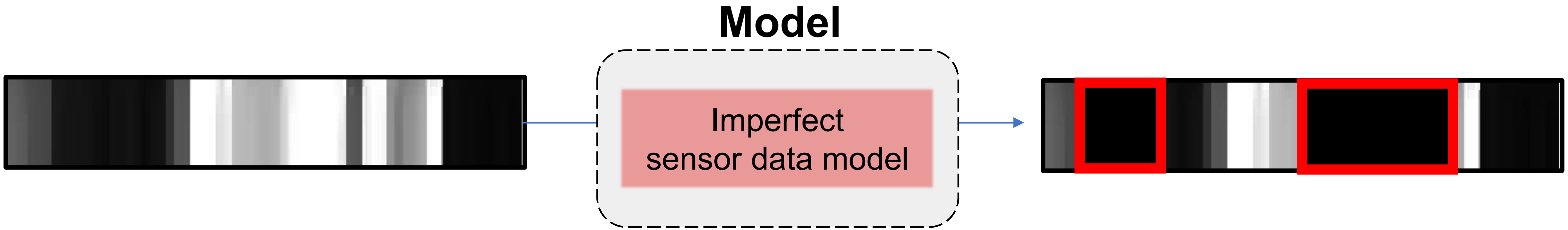
- Modeling imperfect sensor data




Randomly select some portions then turn them to zero

# Our approach: Imperfect sensor data modeling

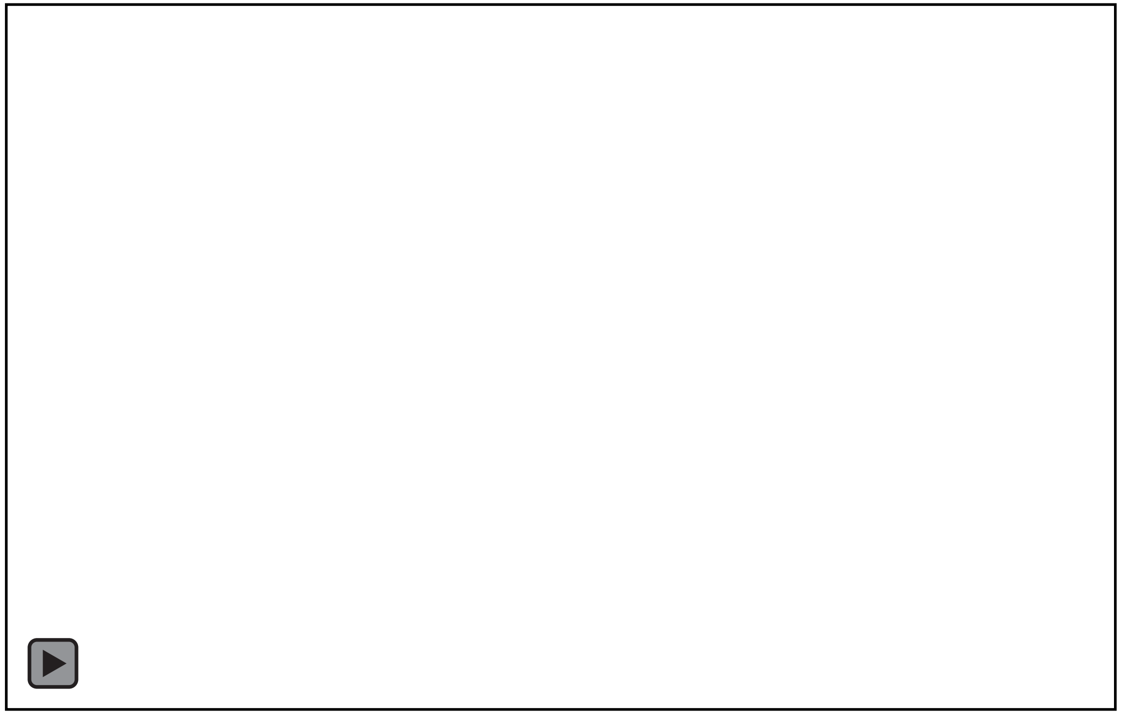
- Modeling imperfect sensor data




Randomly select some portions then turn them to zero

# Our approach: Imperfect sensor data modeling

- Modeling imperfect sensor data



Laser scan data

Normal

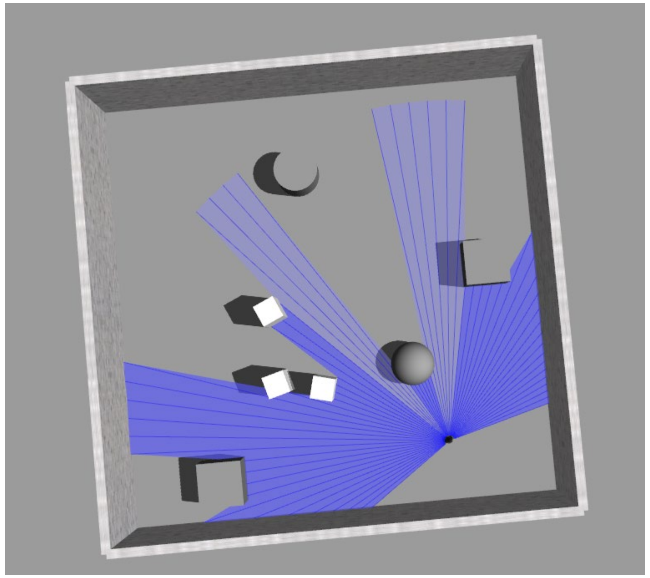
Damaged

# Table of contents

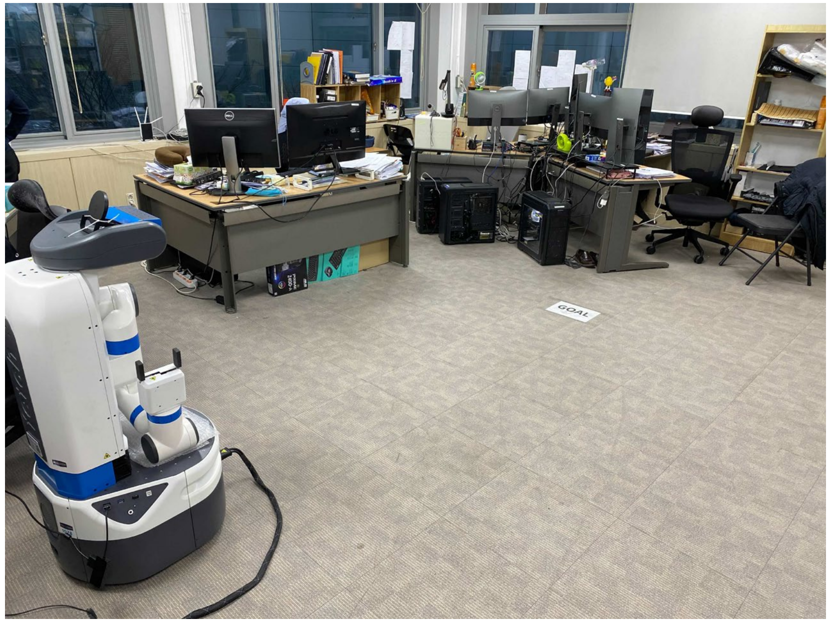
- Introduction & Research goal
- Background
- Our approach
- Experimental Result
- Conclusion
- Role assignment

# Experimental Result

- Simulation experiment

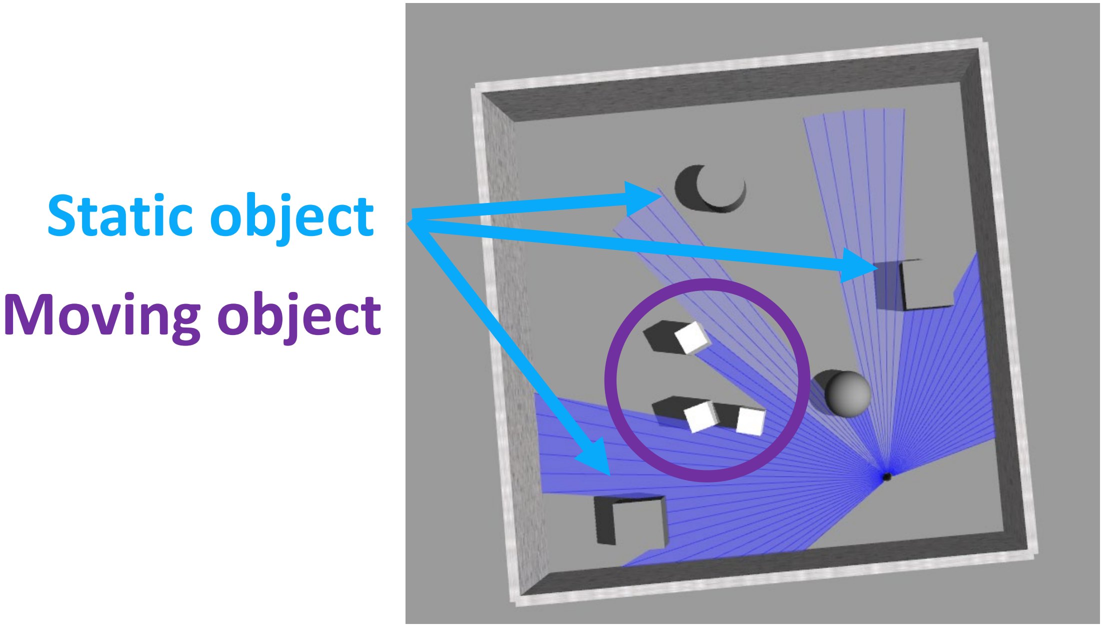


- Real-world experiment

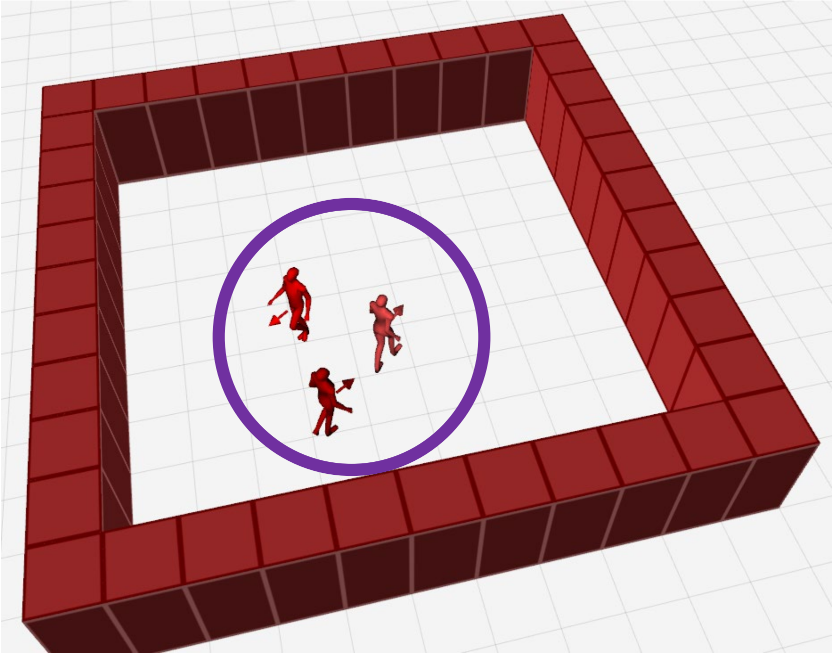


# Experimental Result: Simulation environment setting

- Gazebo: Physics simulator
- Pedestrian Simulation




**Gazebo**



**Pedsim**

# Experimental Result: Learning process

: **Goal position**

Rviz visualization

Gazebo visualization

# Experimental Result: evaluation results

- ***Evaluation** in the case that **some parts of the sensor are damaged***
- *Networks are **trained** with only **clean** simulation sensor data*

| ***Model*** | | ***Success rate*** | |
|---|---|---|---|
| ***Input*** | ***Network*** | ***Scene 1**** | ***Scene 2**** |
| **Single frame sensor data** | MLP | 3/10 | 2/10 |
| | 1D CONV | 3/10 | 2/10 |
| | FC + LSTM | 2/10 | 2/10 |
| **Stacked frame sensor data** | 2D CONV | **4/10** | **3/10** |
| | 2D Conv + LSTM | 2/10 | 1/10 |

- ***Scene 1****: only ***static*** objects
- ***Scene 2****: ***static*** and ***dynamic*** objects

# Experimental Result: evaluation results

- ***Evaluation** in the case that **some parts of the sensor are damaged***
- *Networks are trained with the **imperfect** sensor data **by our modeling***

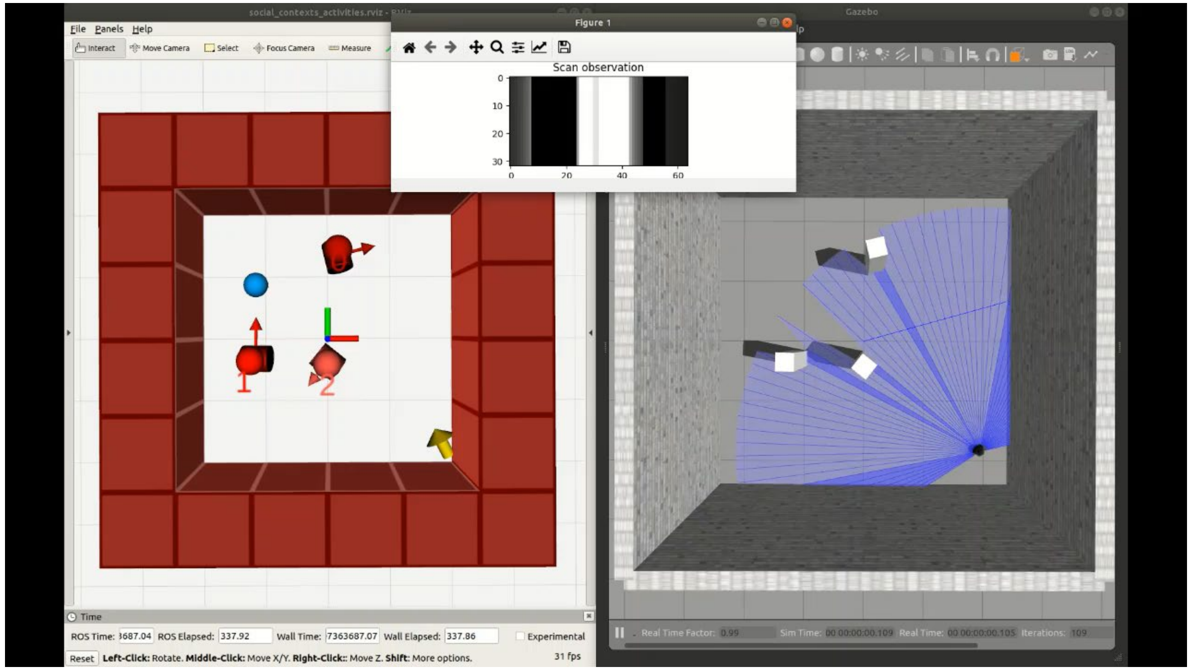
| ***Model*** | | ***Success rate*** | |
|---|---|---|---|
| ***Input*** | ***Network*** | ***Scene 1**** | ***Scene 2**** |
| **Single frame sensor data** | MLP | 5/10 | 4/10 |
| | 1D CONV | 5/10 | 4/10 |
| | FC + LSTM | 6/10 | 5/10 |
| **Stacked frame sensor data** | 2D CONV | **7/10** | **7/10** |
| | 2D Conv + LSTM | 4/10 | 4/10 |

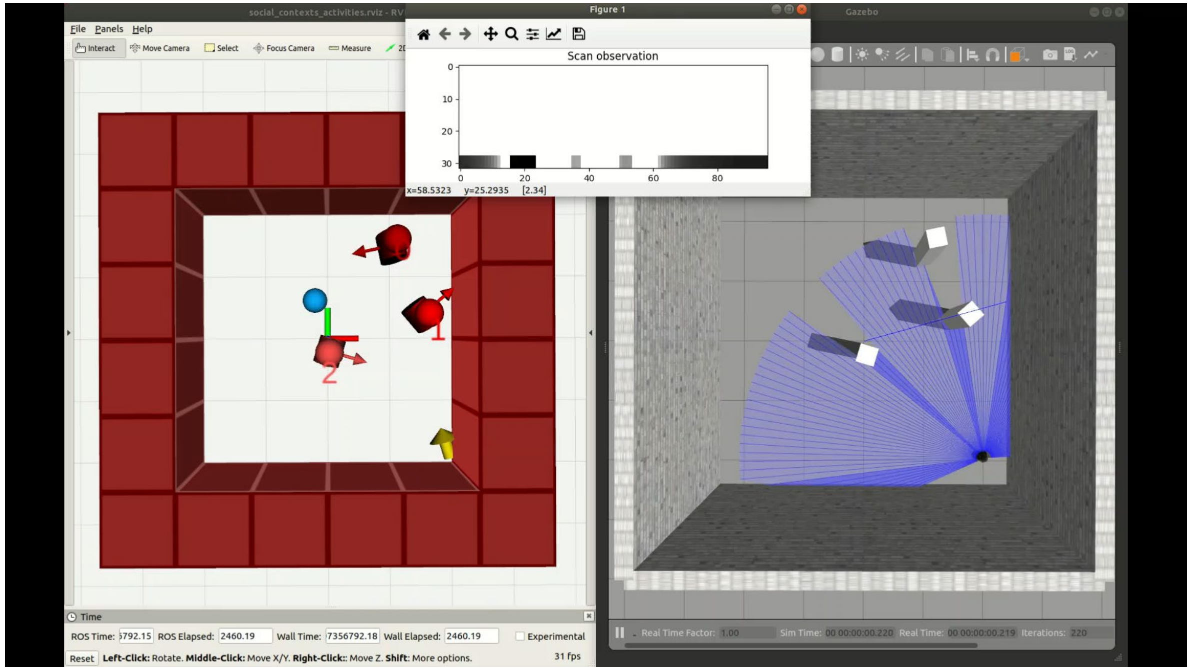
- ***Scene 1****: only ***static*** objects
- ***Scene 2****: ***static*** and ***dynamic*** objects

# Experimental Result

- Model using single frame vs Model using Stacked frames

: Goal position



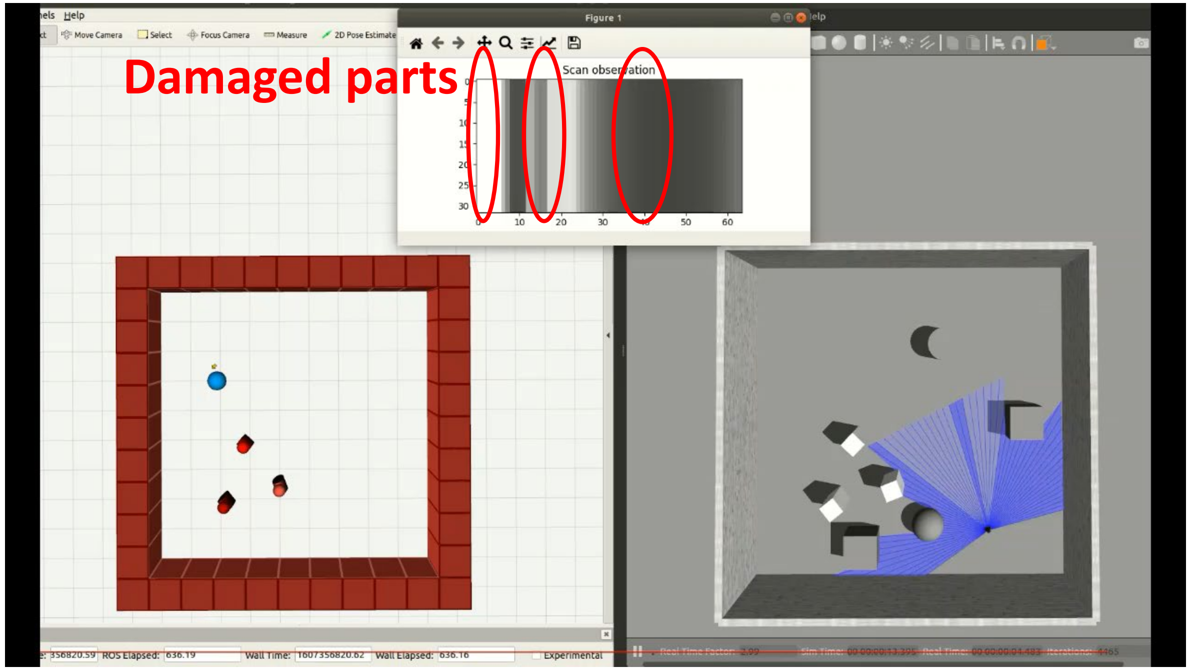

Model using **single** frame

Model using **stacked** frames

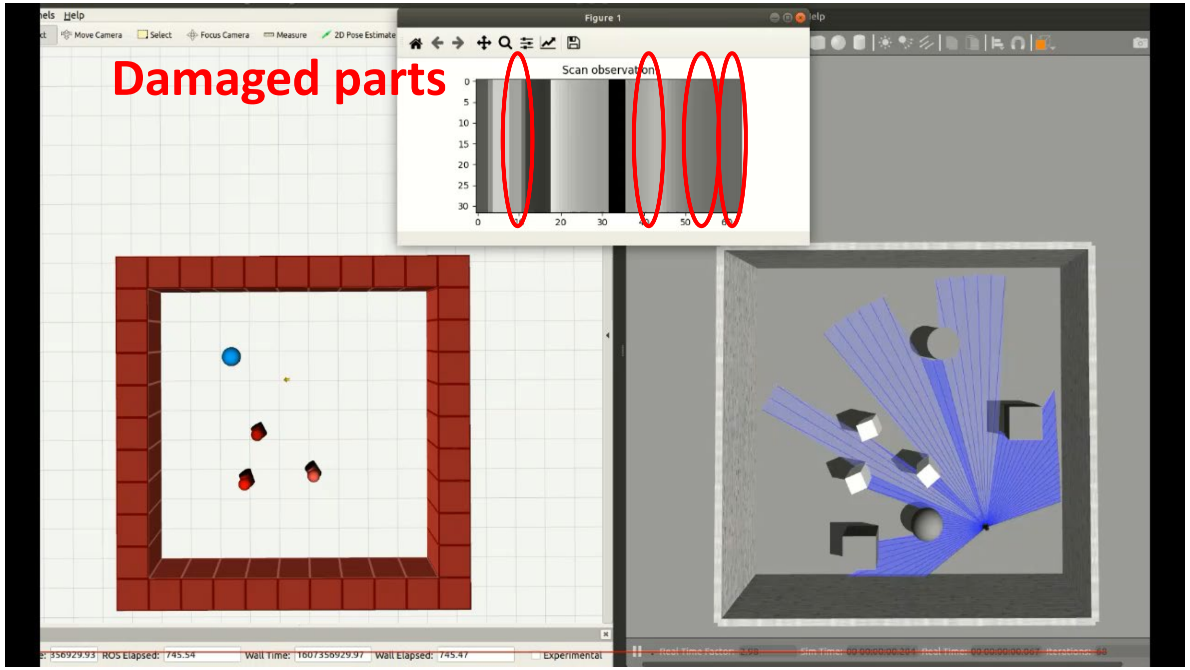
# Ablation Study: Imperfect sensor data modeling

- Navigation under the imperfect sensor data

: Goal position




Trained FC model
**without** our method




Trained FC model
**with** our method

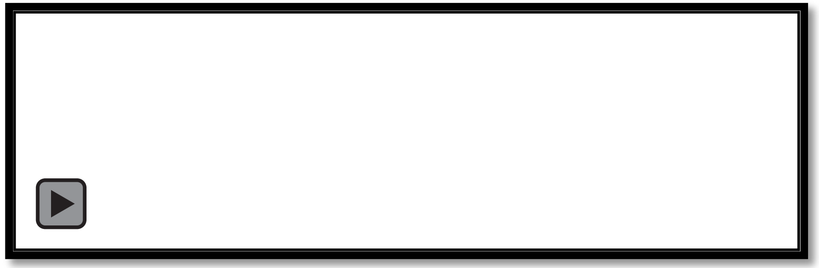
# Experimental Result: Real-world experiment

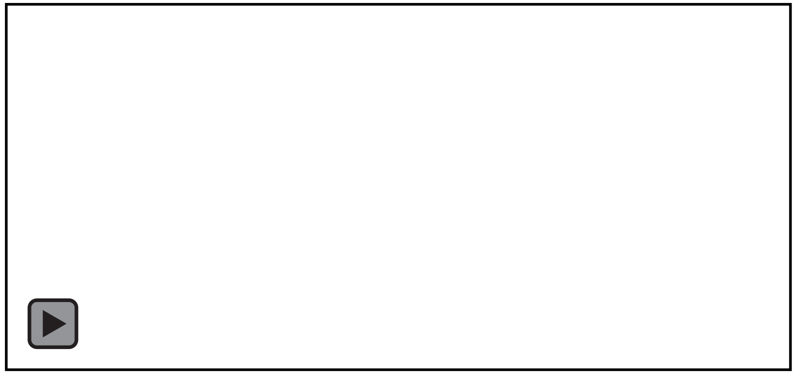
- Robot: Fetch (Differential drive)
- Sensor: 25m-range, 220 degree

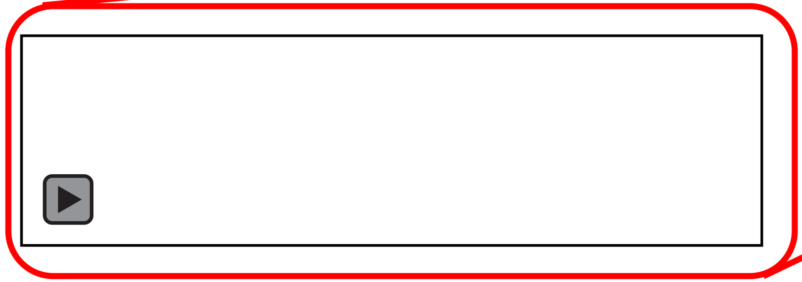
# Experimental Result: Real-world experiment

| Simulation | Real-world |
|---|---|
| Noise-less | Very noisy |

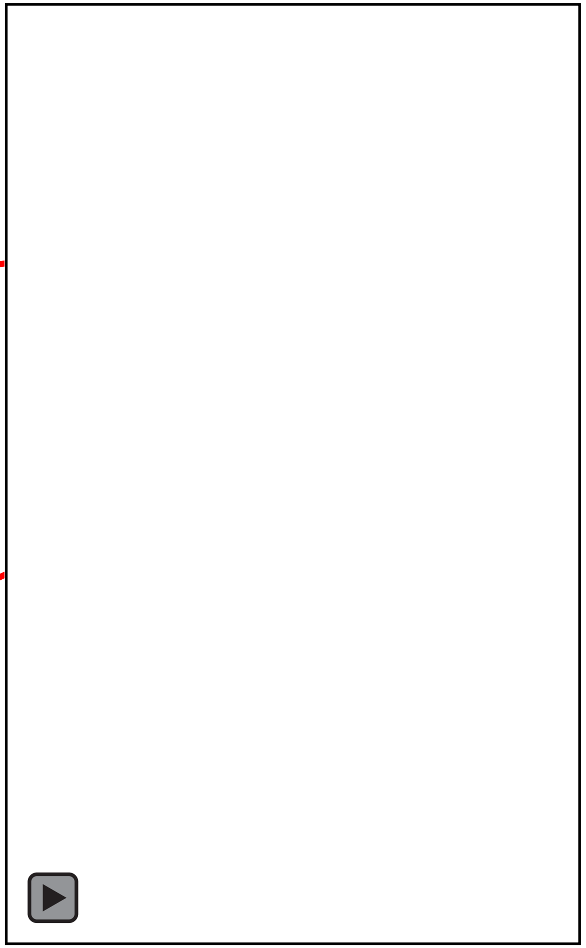
Simulation sensor data





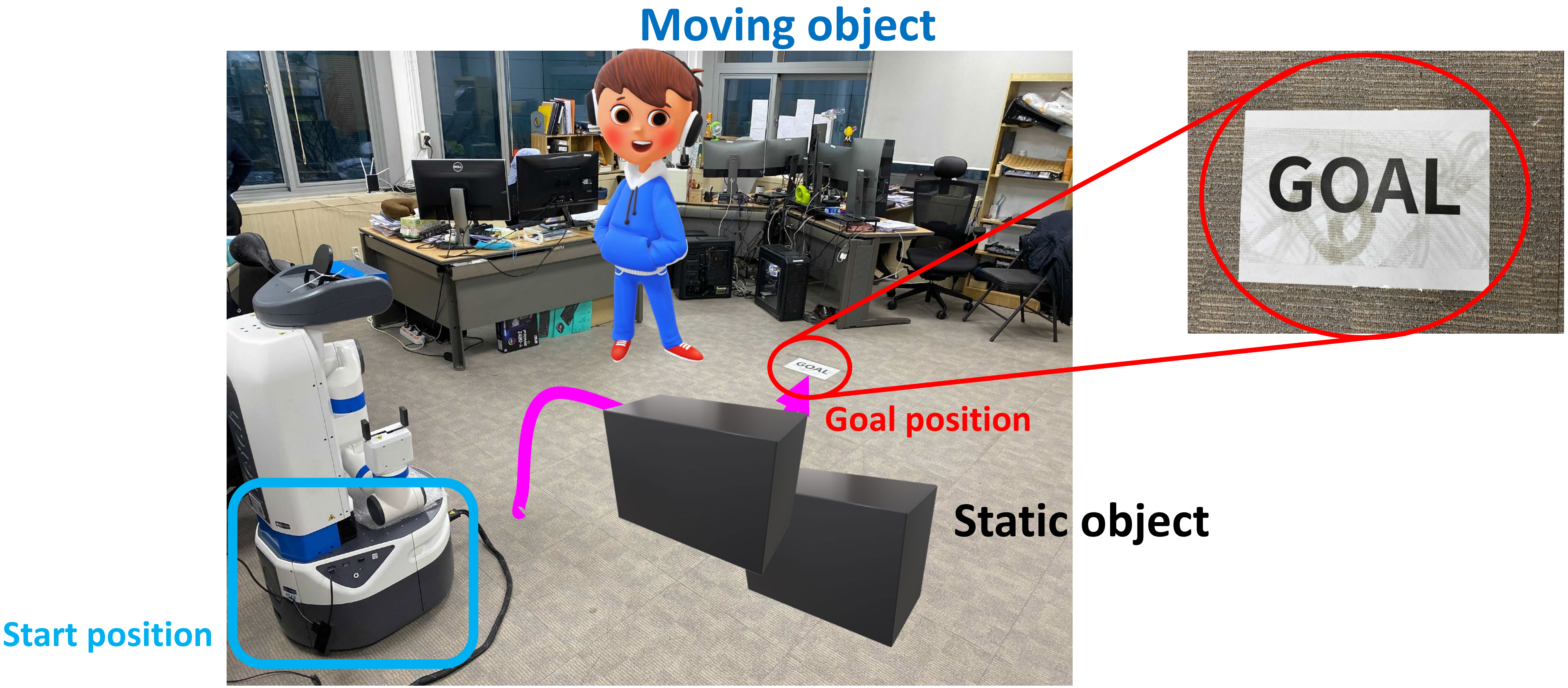


Real world sensor data



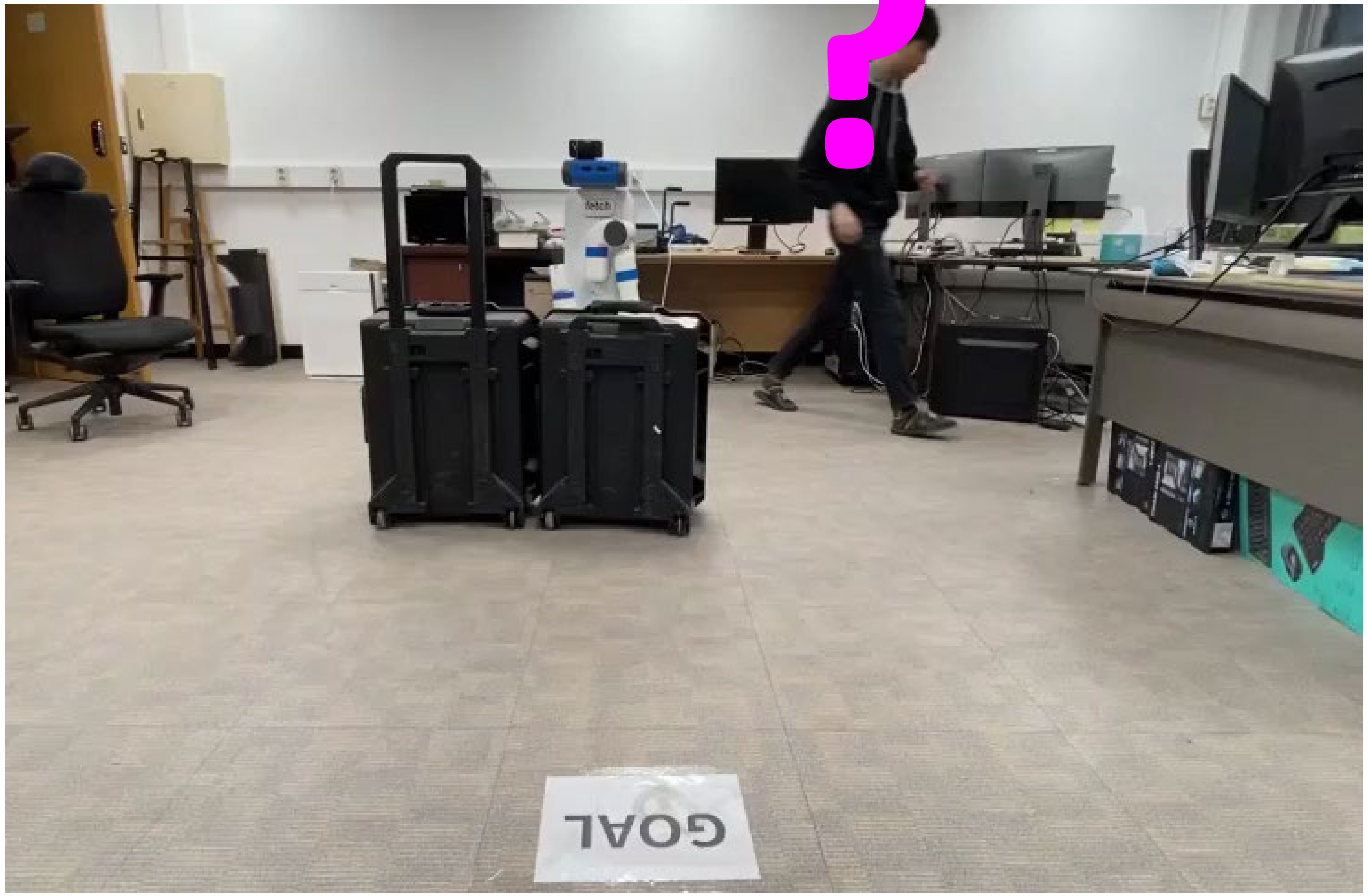
# Experimental Result: Navigation on Real robot

- Configuration of experiment environment



# Experimental Result: Navigation on Real robot

- Model trained **without** our method



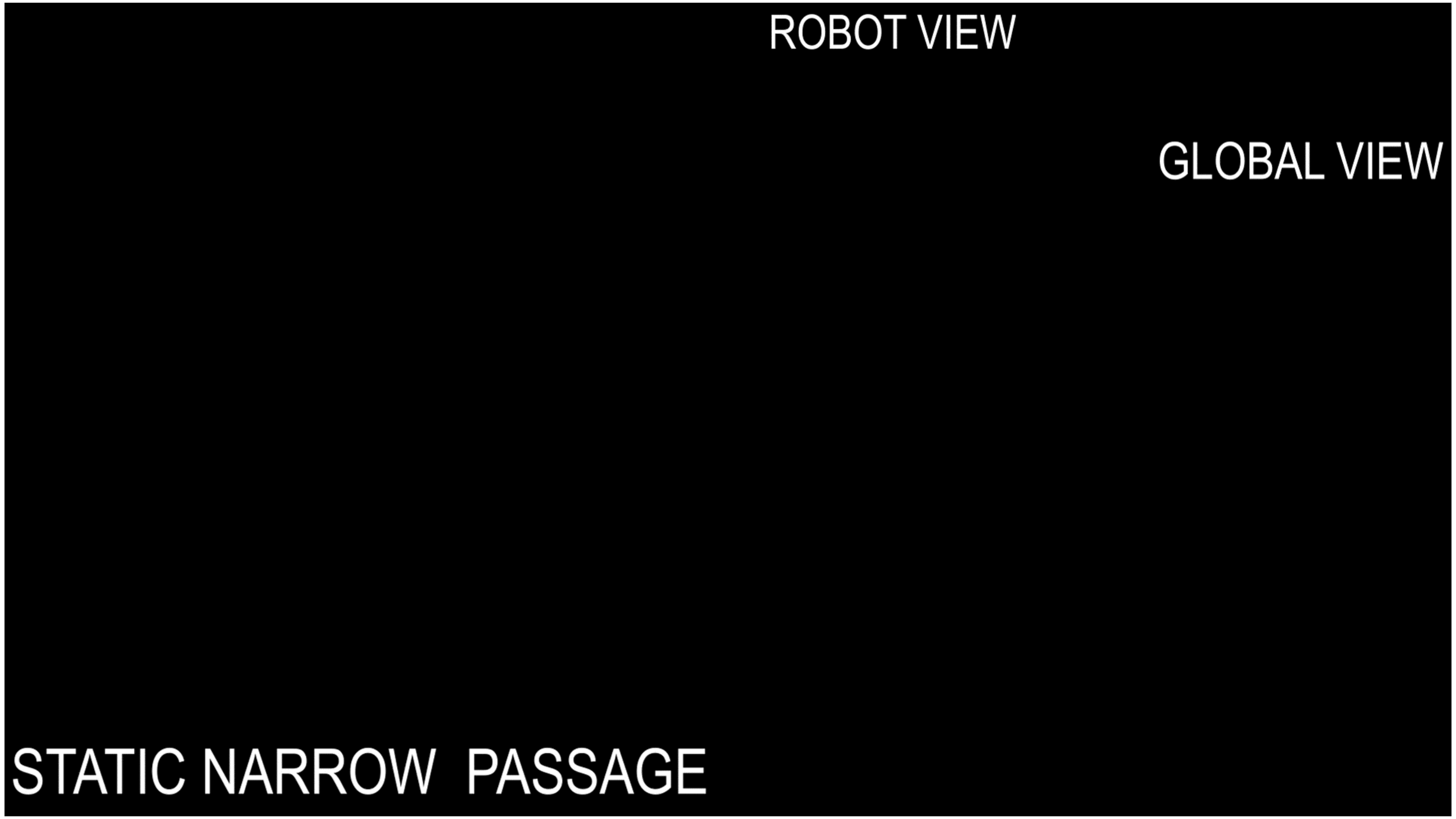
# Experimental Result: Navigation on Real robot

- Model trained **with** our method

**Static object**

# Experimental Result: Navigation on Real robot

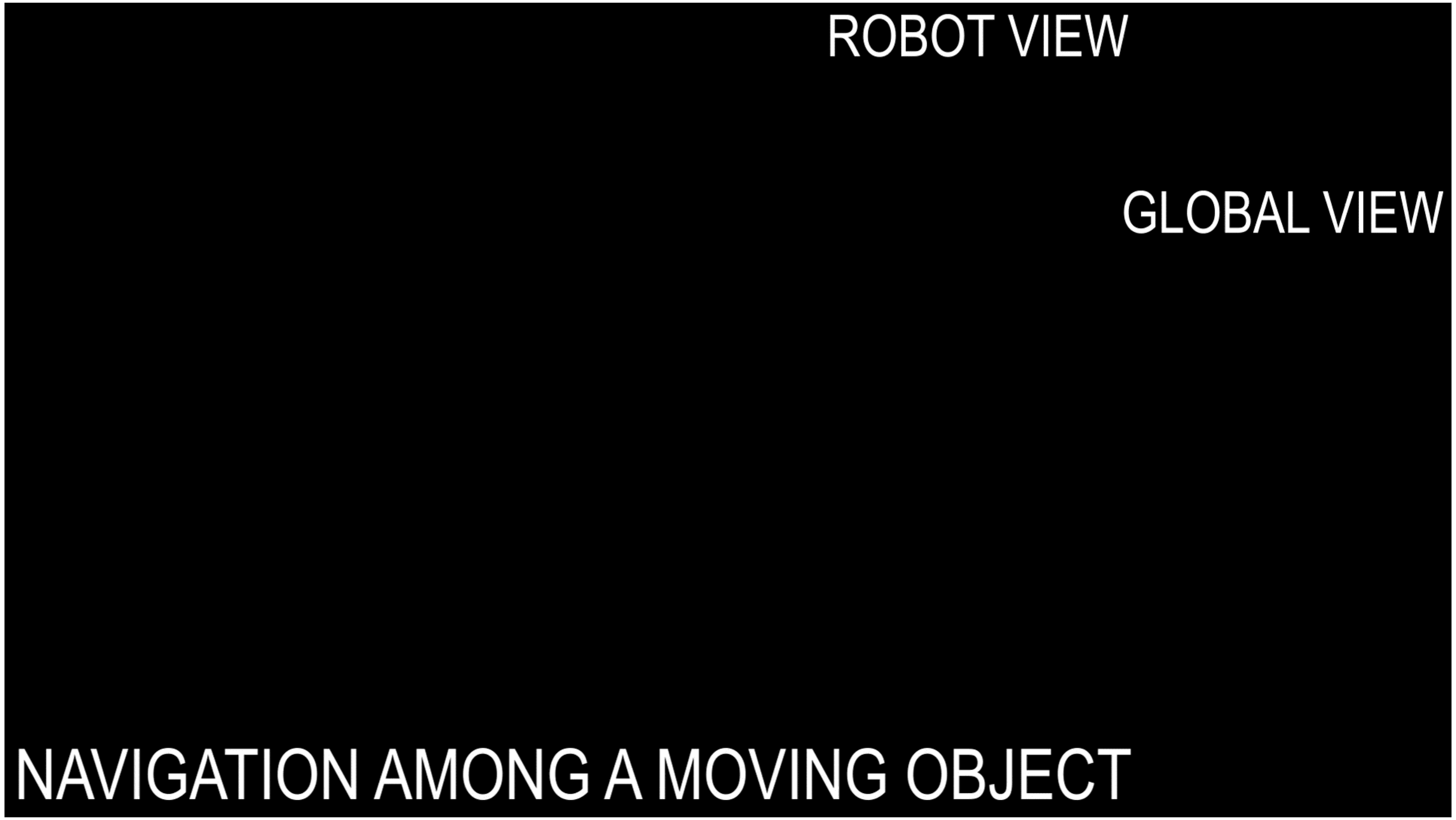
- Model trained **with** our method




**Static object**

# Experimental Result: Navigation on Real robot

- Model trained **with** our method




**Moving object**

# Table of contents

# Conclusion

## Contribution

• A more **robust** navigation was possible by learning RL agent with **the modeling of imperfect sensor data**.

• To verify the effect, not only simulation, but also experiments in **real robots** were performed.

## Limitation

• In modeling various imperfect situations, the value of **zero** is filled in some portions.

• For better performance, the model requires a lot of hyper-parameter **tuning**.

# Conclusion

**Future work**

• Implementation issues in LSTM brought some lower performance. However, it will show more performance if this resolves.

• More elaborated techniques such as predicting the part of damaged sensor will give more chance to avoid collisions.

# Table of contents

# Role assignment

| | Hyeongyeol | Sebin | Minsung |
|---|---|---|---|
| Build training environment | O | v | v |
| Build network architecture | v | O | v |
| Build structure of RL training | v | v | O |
| Training RL agent | O | O | O |
| Real-robot experiment | O | O | O |
| Testing & Collecting results | O | O | O |
| Preparing Final Presentation | O | O | O |

**O: lead**
**V: support**

# Reference

- Papers

[1] J. Choi, K. Park, M. Kim and S. Seok, "Deep Reinforcement Learning of Navigation in a Complex and Crowded Environment with a Limited Field of View," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 5993-6000, doi: 10.1109/ICRA.2019.8793979.
[2] T. Fan, P. Long, W. Liu, J. Pan, R. Yang and D. Manocha, "Learning Resilient Behaviors for Navigation Under Uncertainty," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 5299-5305, doi: 10.1109/ICRA40945.2020.9196785.
[3] A. Faust et al., "PRM-RL: Long-range Robotic Navigation Tasks by Combining Reinforcement Learning and Sampling-Based Planning," 2018 IEEE International Conference on Robotics and Automation (ICRA), Brisbane, QLD, 2018, pp. 5113-5120, doi: 10.1109/ICRA.2018.8461096.
[4] J. Jin, N. M. Nguyen, N. Sakib, D. Graves, H. Yao and M. Jagersand, "Mapless Navigation among Dynamics with Social-safety-awareness: a reinforcement learning approach from 2D laser scans," 2020 IEEE International Conference on Robotics and Automation (ICRA), Paris, France, 2020, pp. 6979-6985, doi: 10.1109/ICRA40945.2020.9197148.
[5] F. Leiva and J. Ruiz-del-Solar, "Robust RL-Based Map-Less Local Planning: Using 2D Point Clouds as Observations," in IEEE Robotics and Automation Letters, vol. 5, no. 4, pp. 5787-5794, Oct. 2020, doi: 10.1109/LRA.2020.3010732.
[6] C. Chen, Y. Liu, S. Kreiss and A. Alahi, "Crowd-Robot Interaction: Crowd-Aware Robot Navigation With Attention-Based Deep Reinforcement Learning," 2019 International Conference on Robotics and Automation (ICRA), Montreal, QC, Canada, 2019, pp. 6015-6022, doi: 10.1109/ICRA.2019.8794134.
[7] Jin, Jun, et al. "Mapless Navigation among Dynamics with Social-safety-awareness: a reinforcement learning approach from 2D laser scans." 2020 IEEE International Conference on Robotics and Automation (ICRA). IEEE, 2020.
[8] Gao, Wei, et al. "Intention-net: Integrating planning and deep learning for goal-directed autonomous navigation." arXiv preprint arXiv:1710.05627 (2017).
[9] Everett, Michael, Yu Fan Chen, and Jonathan P. How. "Motion planning among dynamic, decision-making agents with deep reinforcement learning." 2018 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS). IEEE, 2018.

- Media Contents

[1] Professionelle mobile Service-Roboter - MetraLabs
[2] HoLLiE Mobile Service Robot Can Bend to Reach the Floor – roboticgizmos
[3] https://www.artificialinventive.com/blog/
[4] https://developer.softbankrobotics.com/blog/crowdbot-safe-navigation-robots-dense-crowds

# Thank you for listening!

Feel free to ask any questions