CS 4803 / 7643: Deep Learning

Topics: RL and Robotics
– Embodied AI
– Proximal Policy Optimization (PPO)
– Application: Robotics
  – PointGoal Navigation

Joanne Truong
Georgia Tech

Lecture Plan

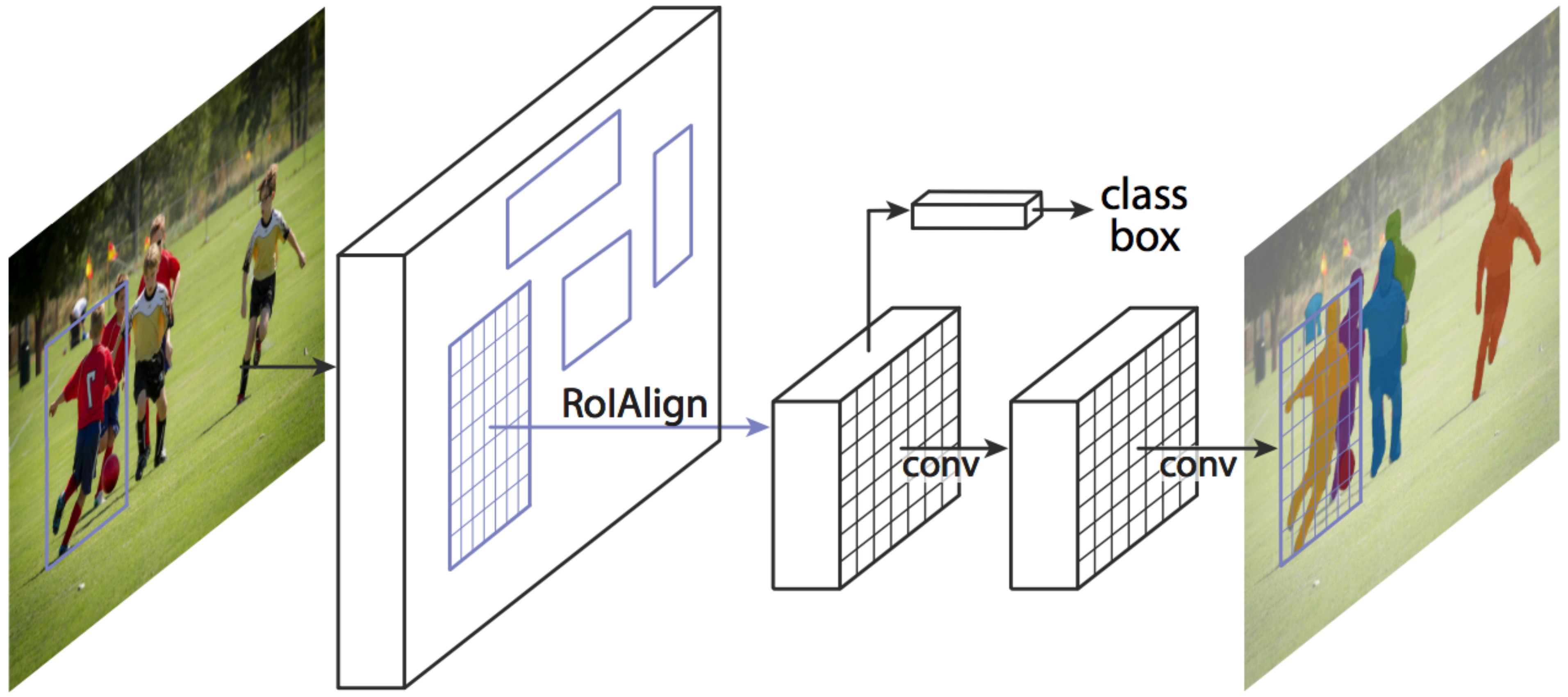
- Embodied AI
- Introduce more advanced RL — Proximal Policy Optimization (PPO)
- Application: Robotics
  - PointGoal Navigation: Combine CNNs, RNNs (LSTMs), and RL together

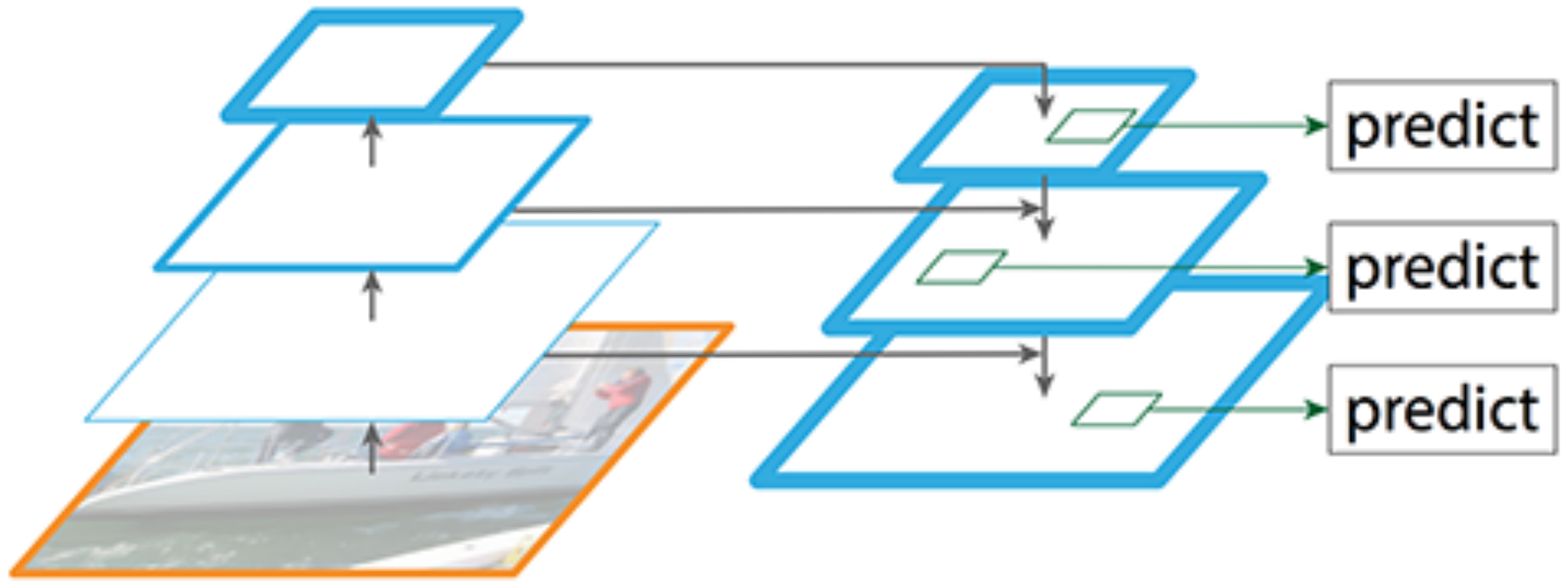
Slide credit: Erik Wijmans

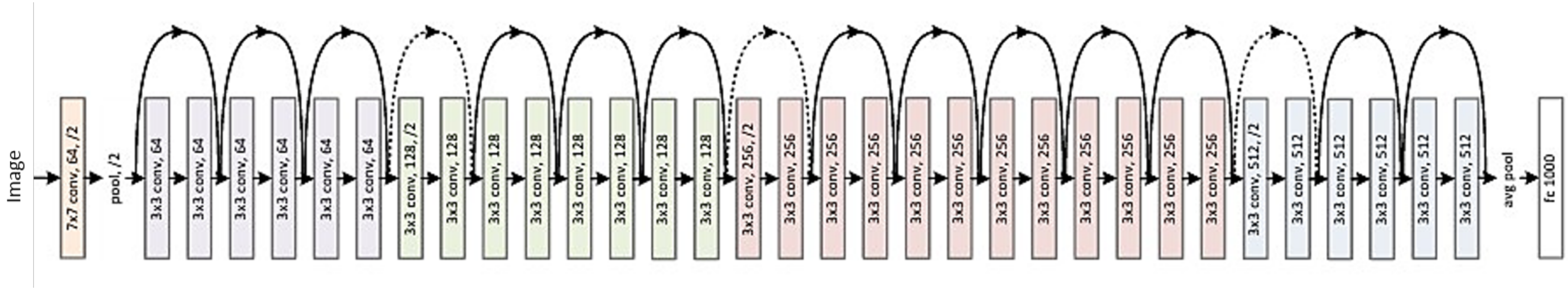
State-of-the-Art Visual Recognition

Slide credit: Abhishek Das

State-of-the-Art Visual Recognition



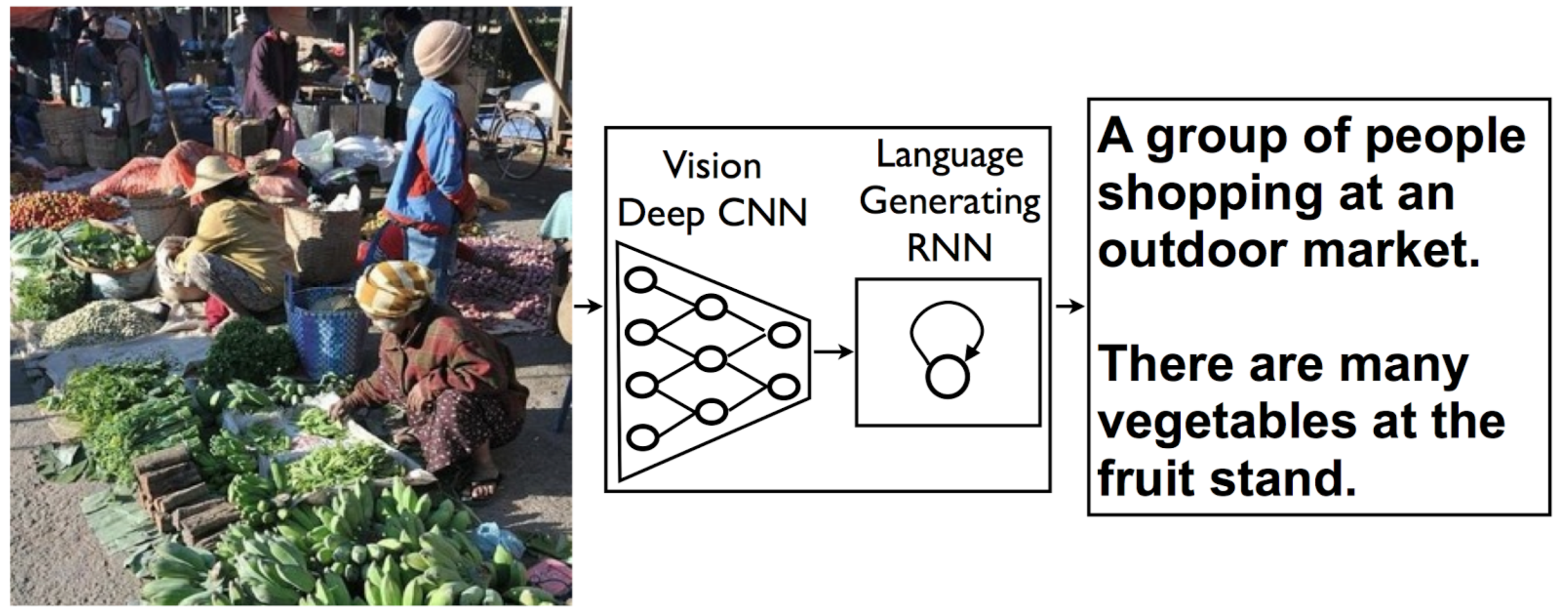


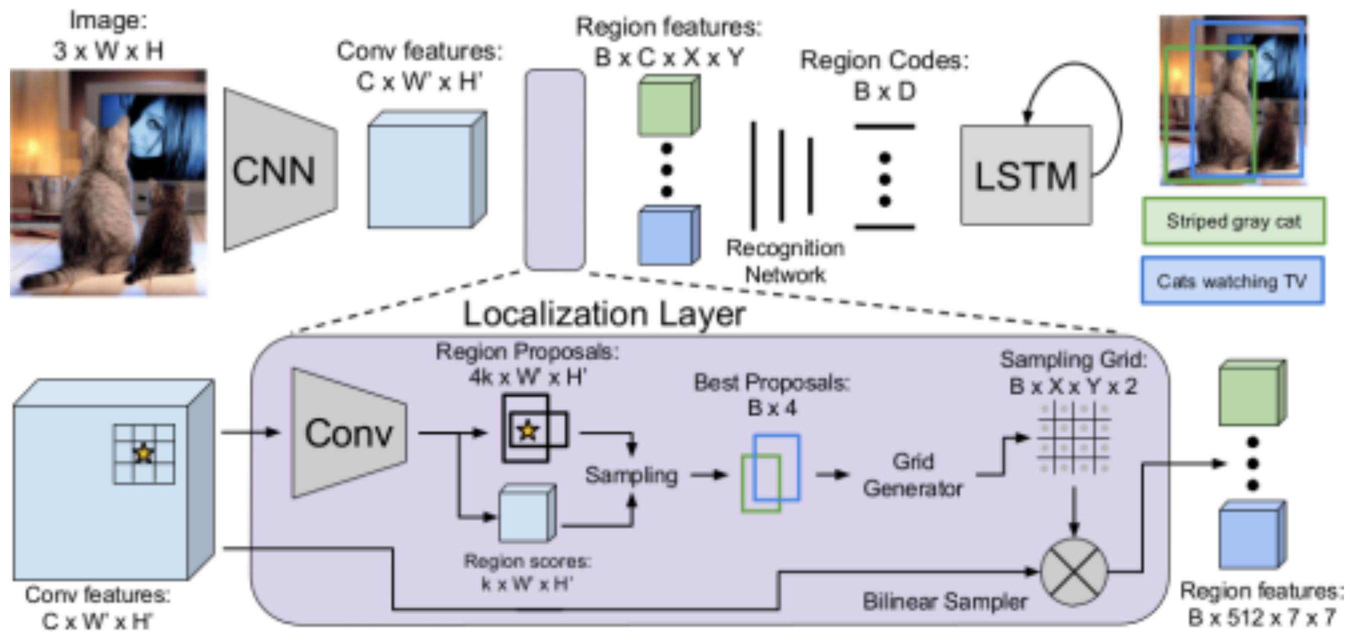


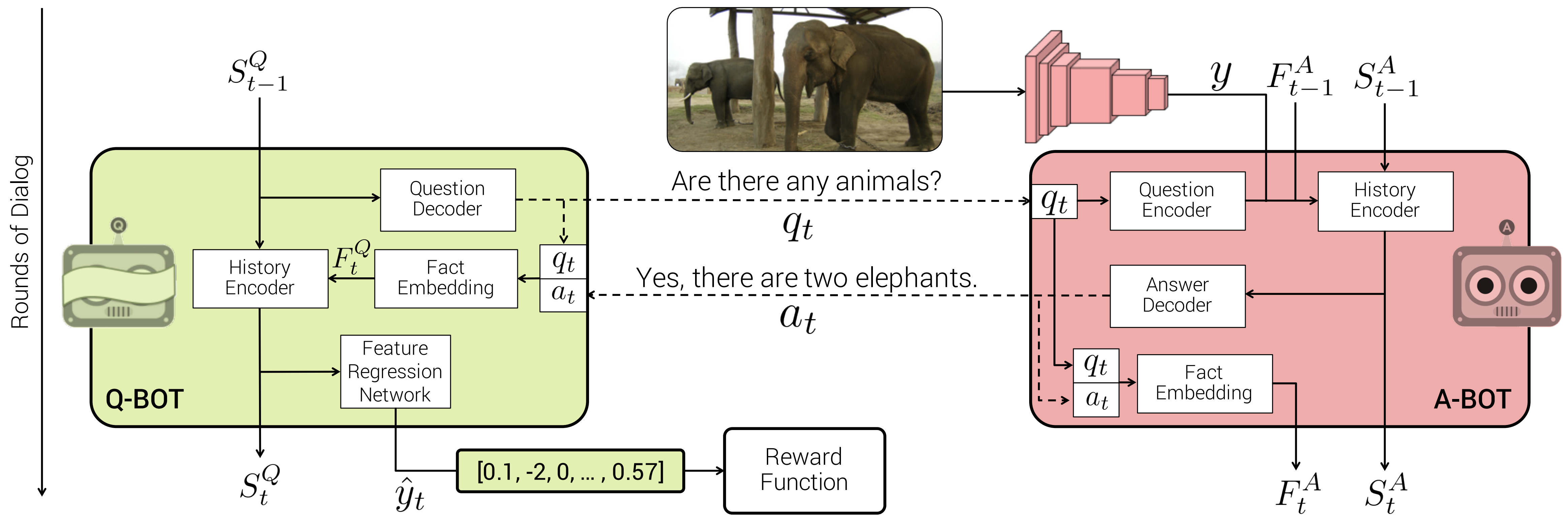
He et al., 2016a, b
He et al., 2017
Lin et al., 2017

Slide credit: Abhishek Das

State-of-the-Art Visual Recognition







Vinyals et al., 2015
Karpathy and Johnson., 2016
Das et al., 2017a, b

Slide credit: Abhishek Das

State-of-the-Art Visual Recognition

Slide credit: Abhishek Das

Image Credit: You et al., 2016

Applications

Slide credit: Abhishek Das

Applications

Applications

Physical agent

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Applications

Physical agent
capable of taking
actions in the world

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Applications

Physical agent
capable of taking
actions in the world

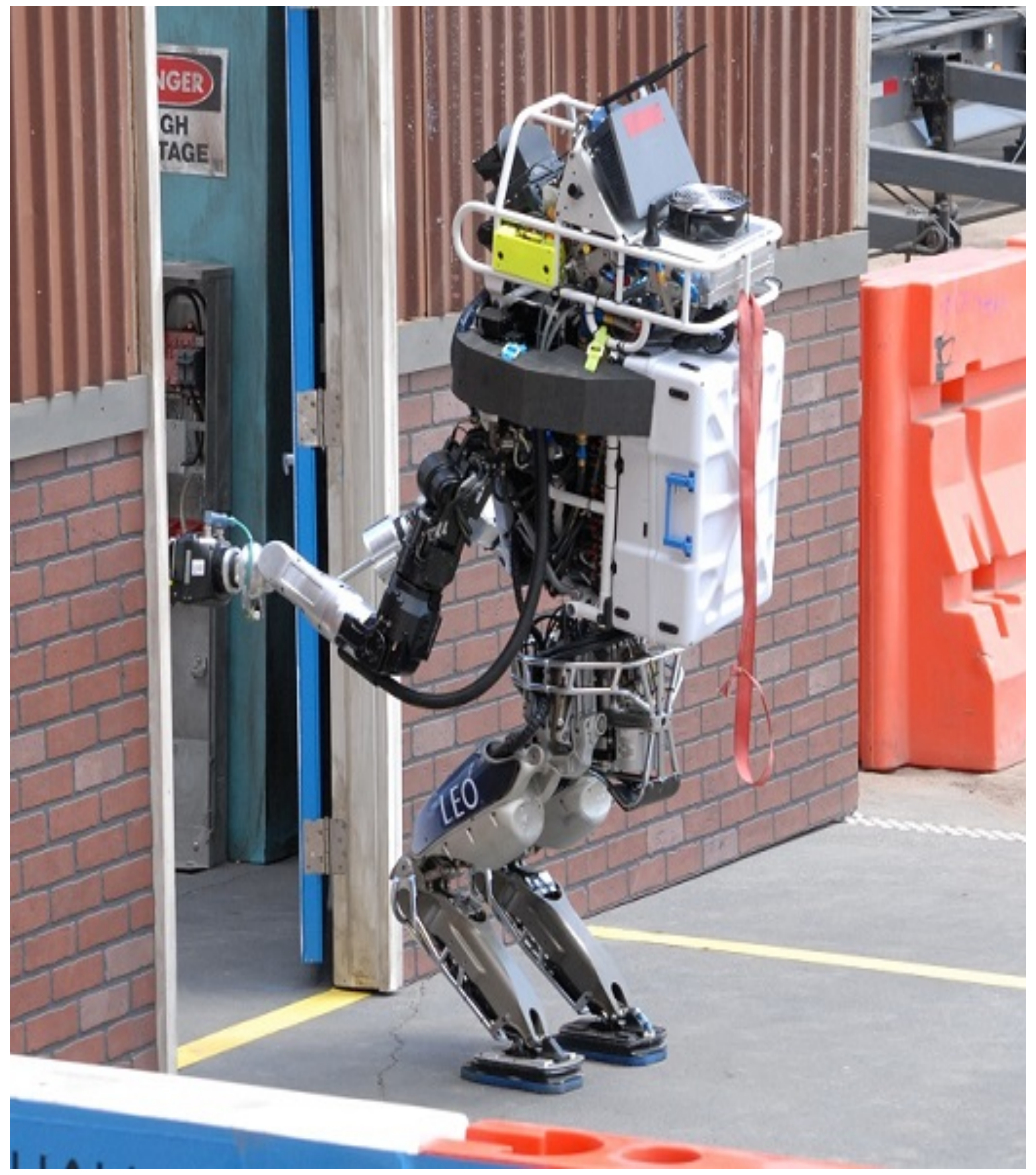
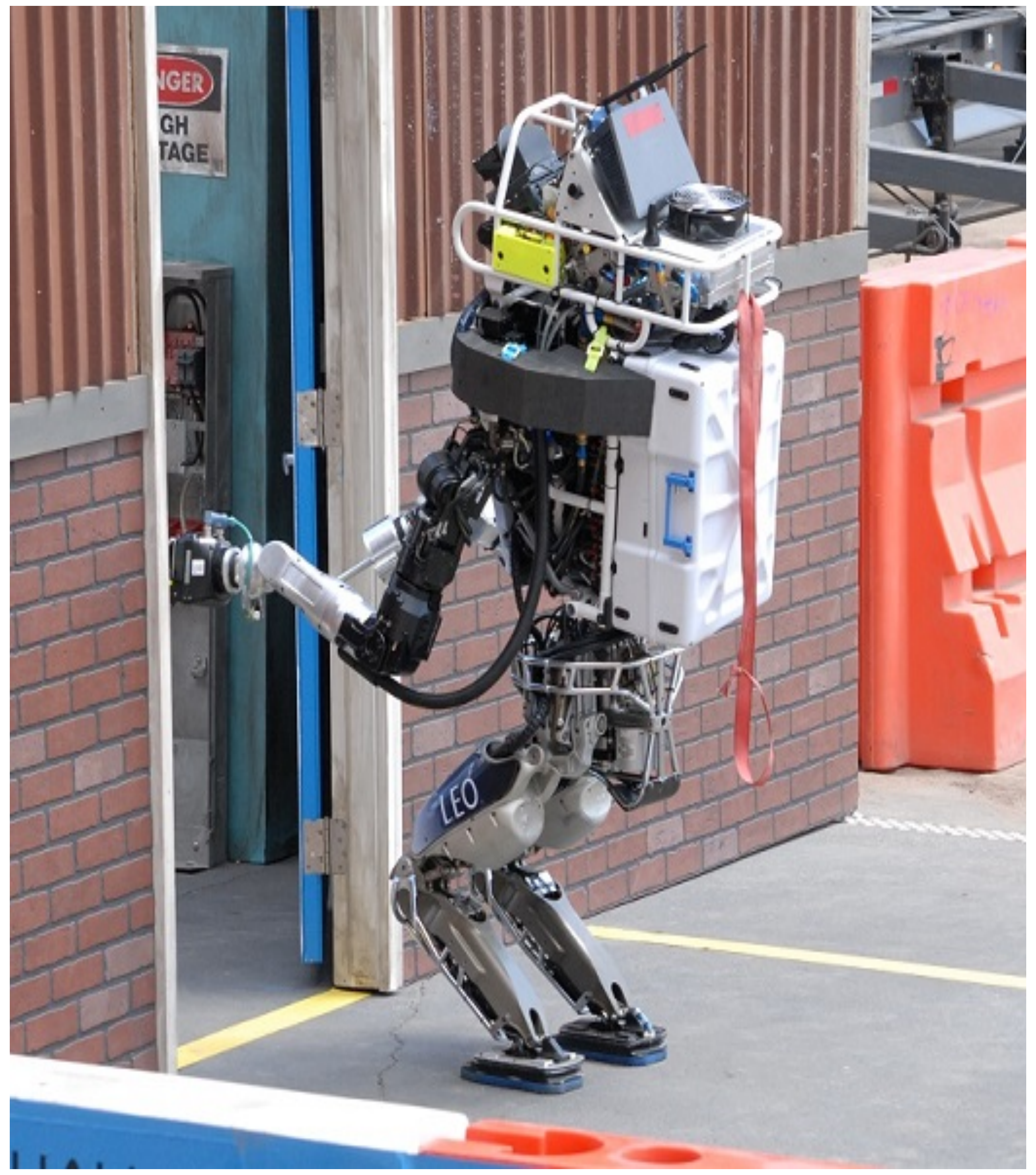


Image Credit: Lockheed Martin; DARPA Robotics Challenge

Applications

Physical agent capable of taking actions in the world and talking to humans in natural language



Is there smoke in any room around you?

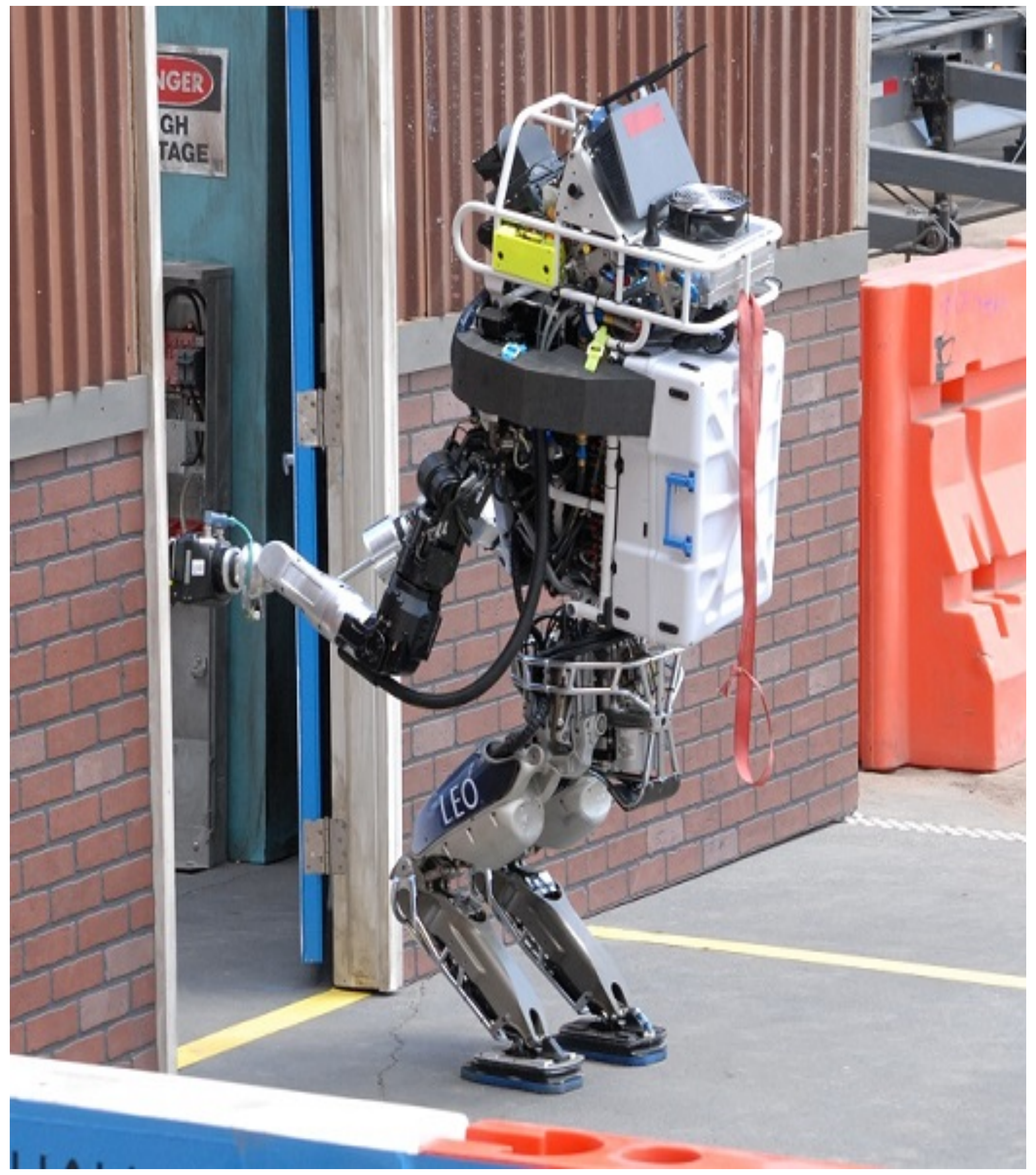
Yes, in one room

Go there and look for people

...

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Applications



Is there smoke in any room around you?

Yes, in one room

Go there and look for people

...

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Applications

Is there smoke in any room around you?

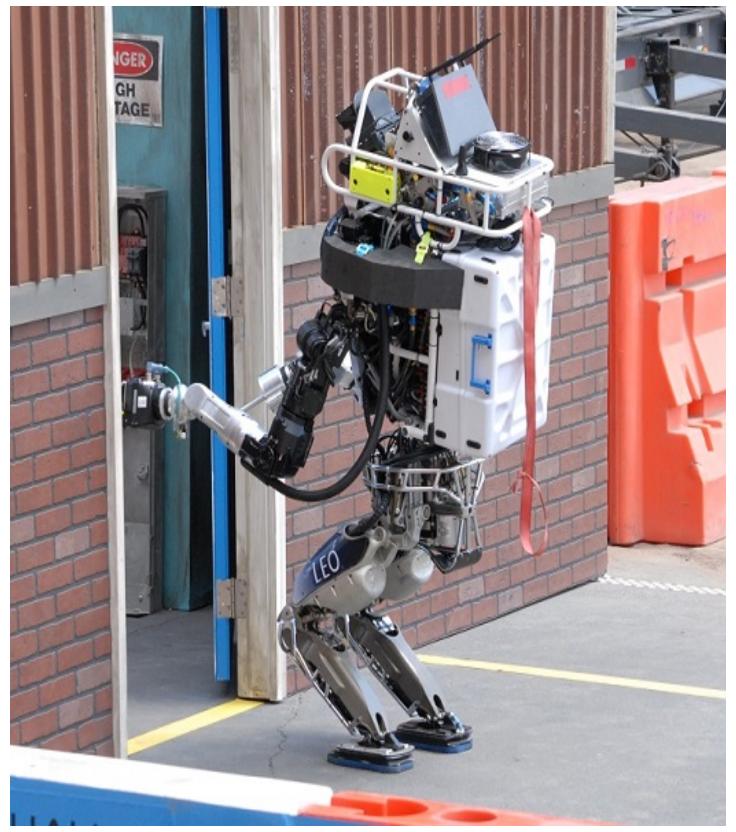
Yes, in one room

Go there and look for people

...

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Challenges



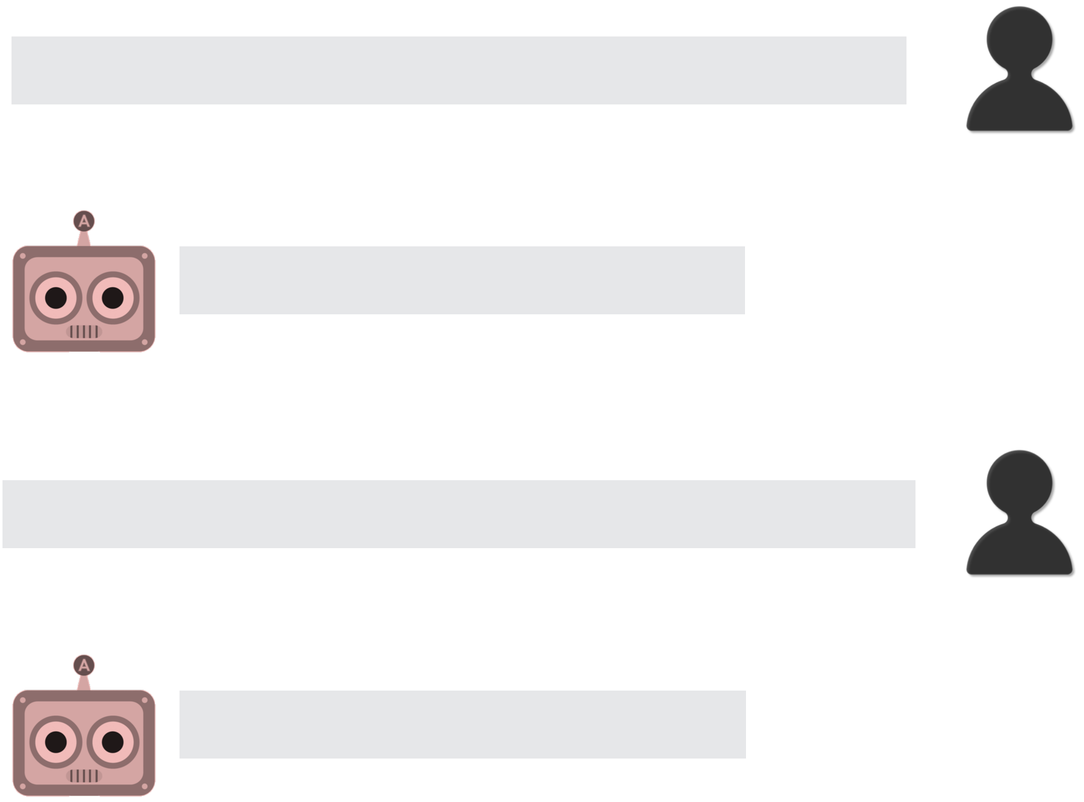
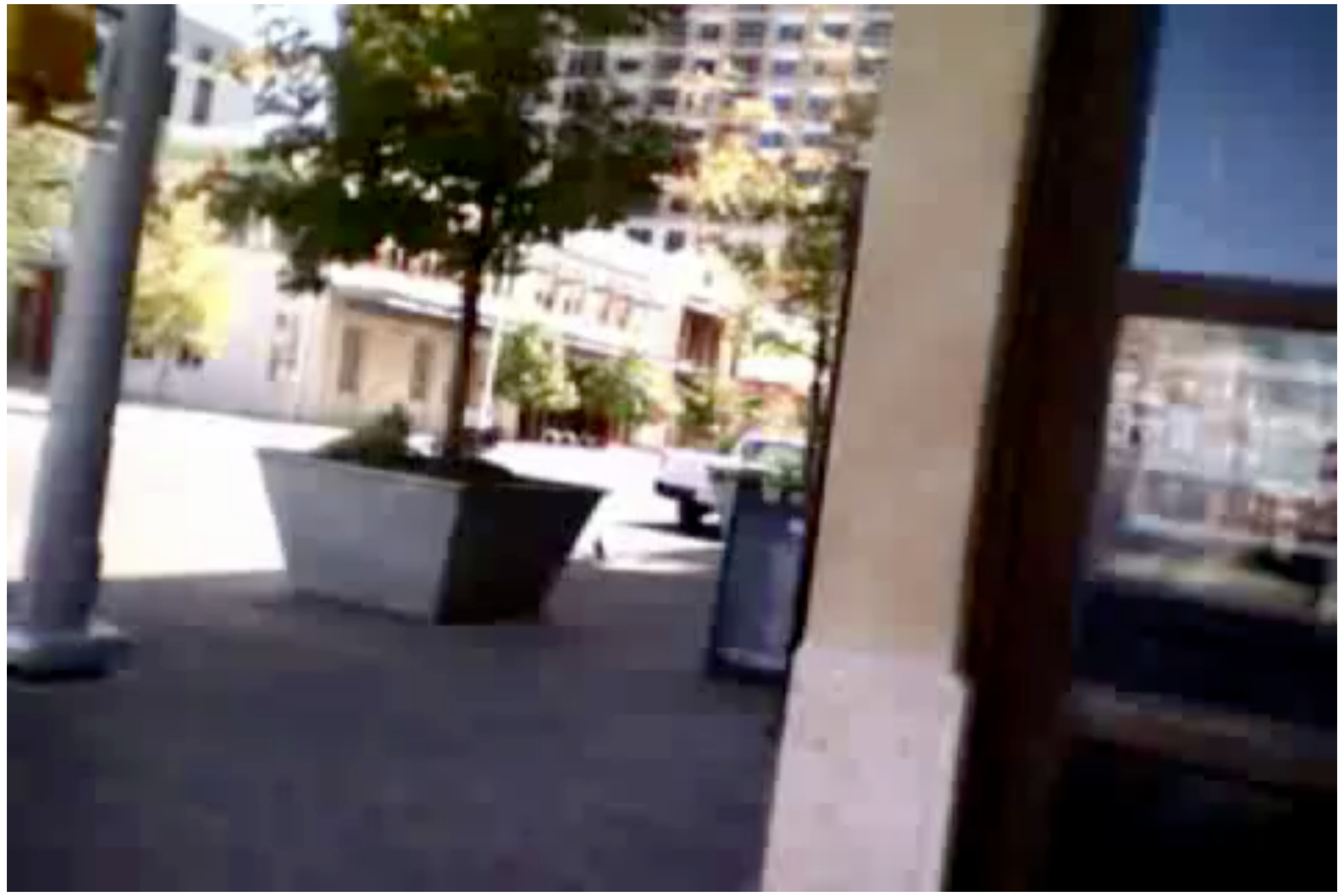


Image Credit: Image-Net
Image Credit: Lockheed Martin; DARPA Robotics Challenge

Slide credit: Abhishek Das

Challenges

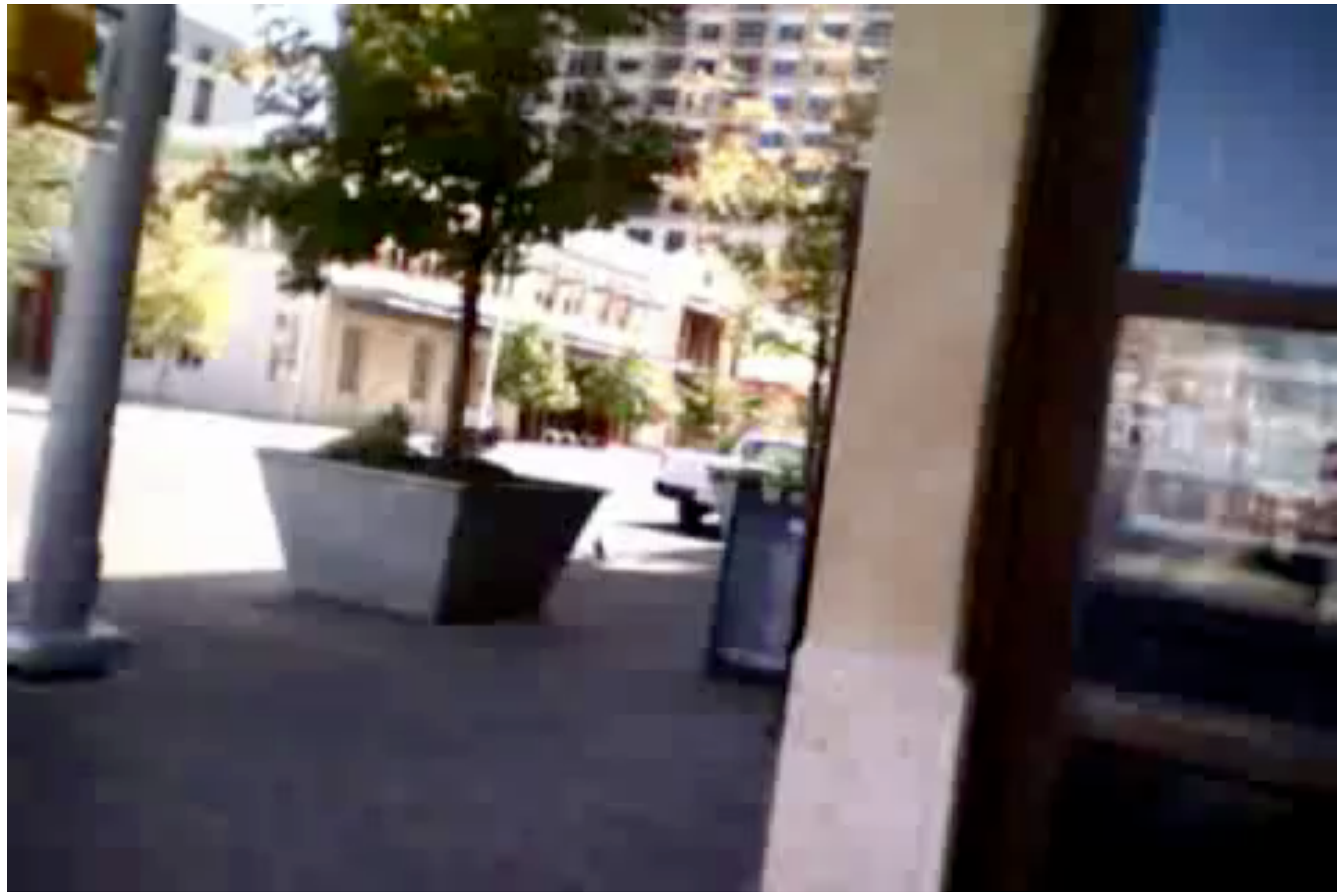
Egocentric vision



No access to well-composed, curated images

Challenges

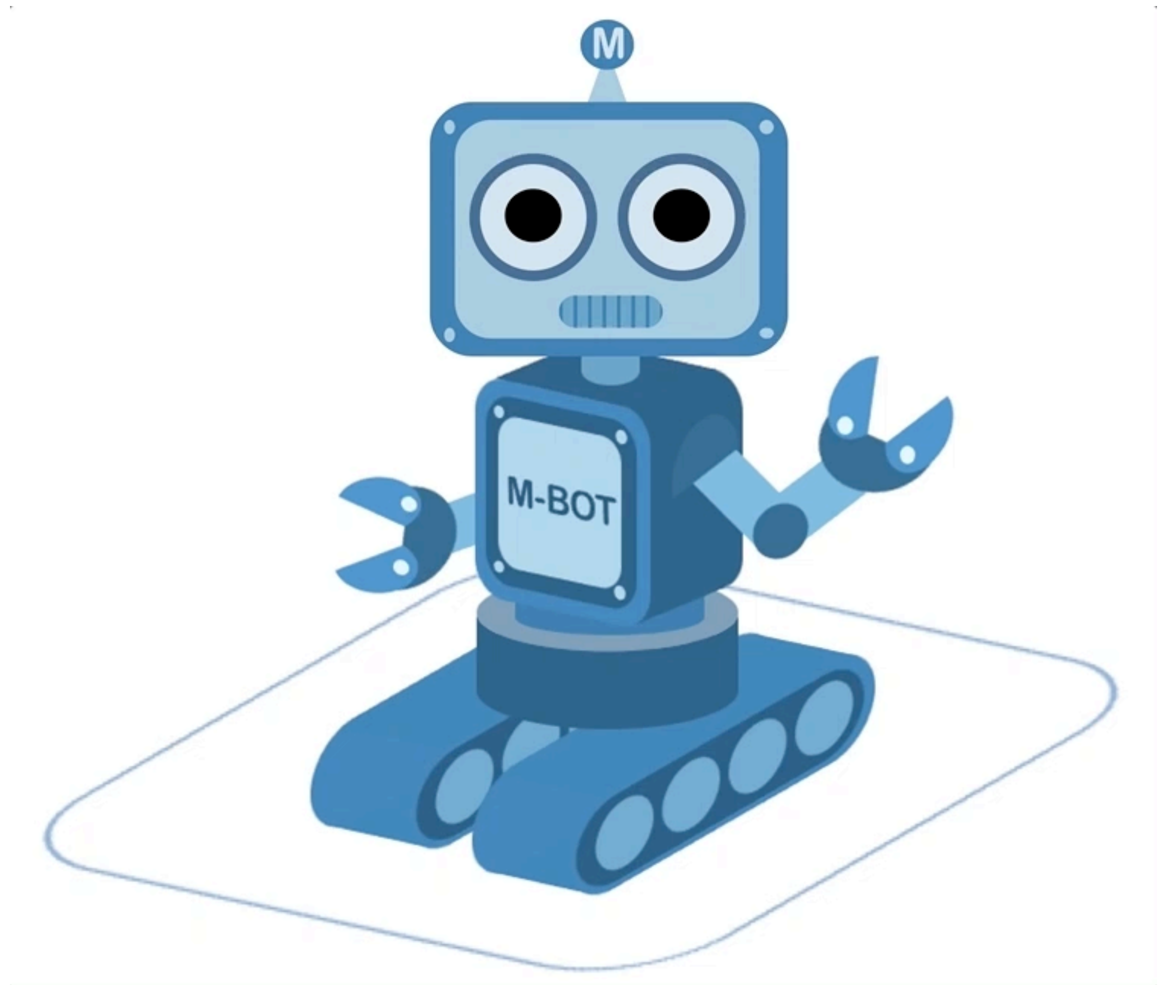
Egocentric vision



Challenges

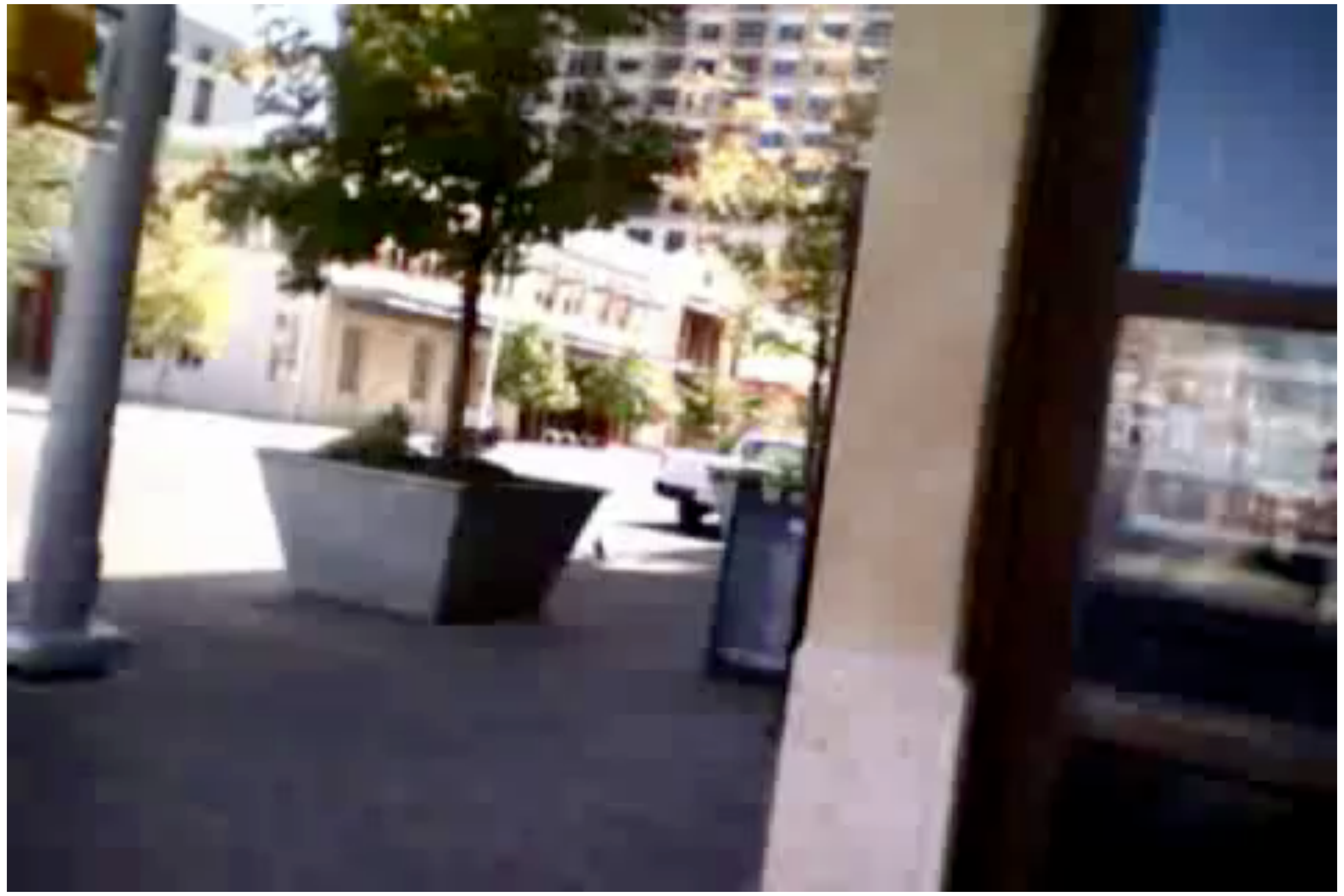
Egocentric vision

Active perception



Action

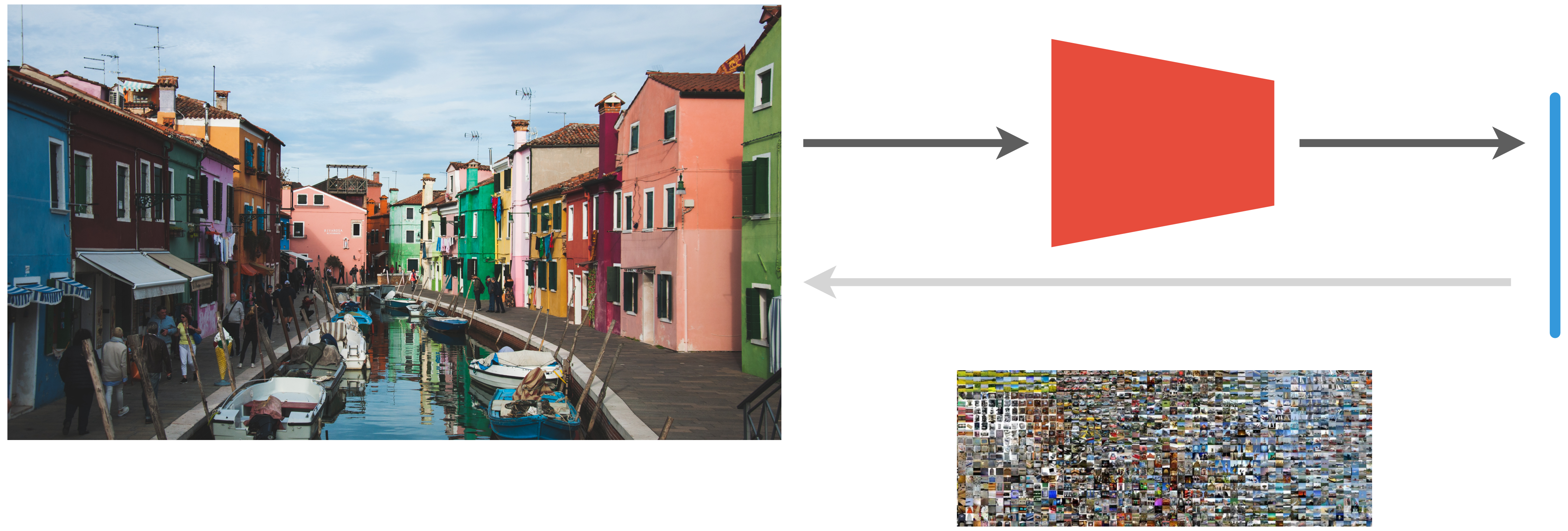
Observation

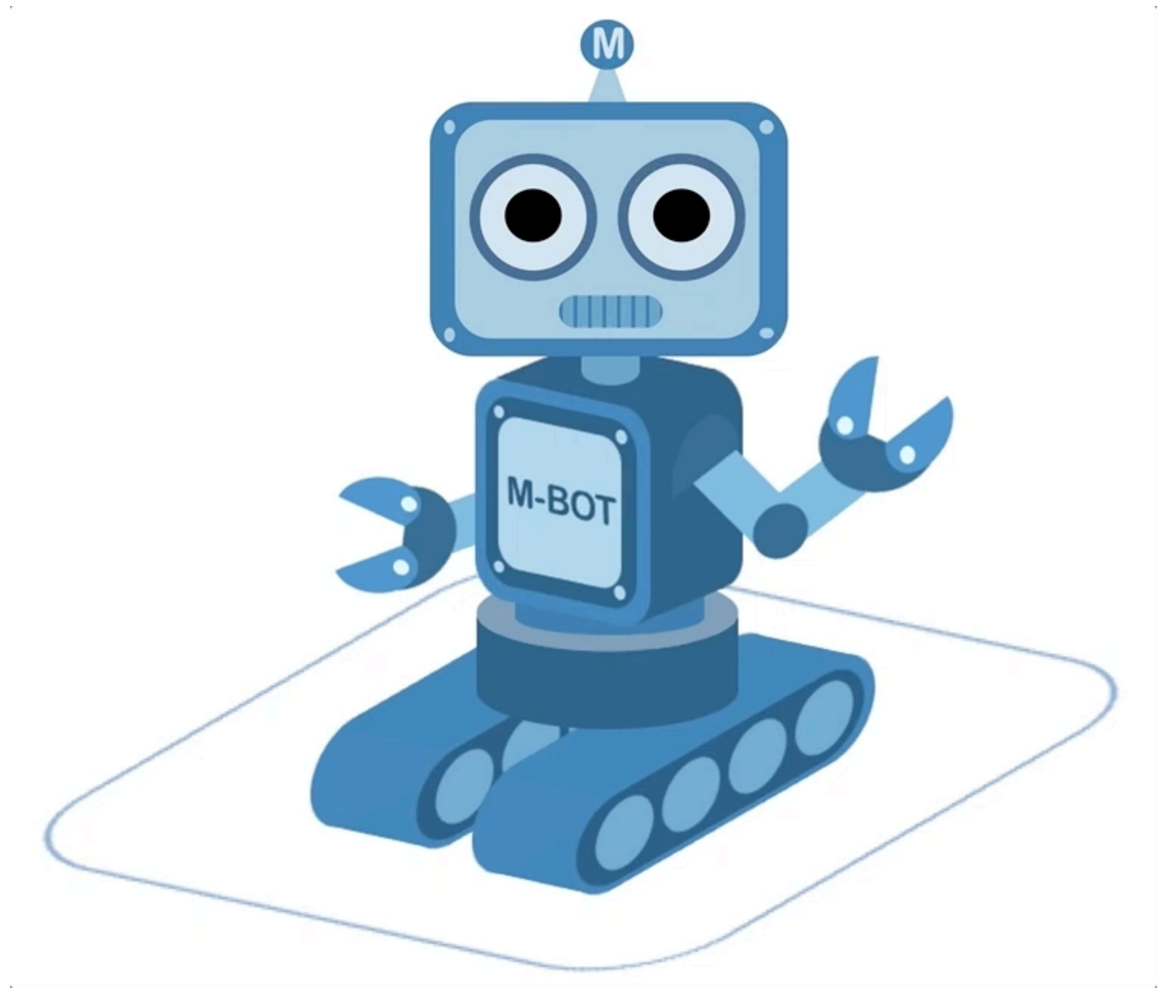


Agent controls incoming data distribution

Challenges

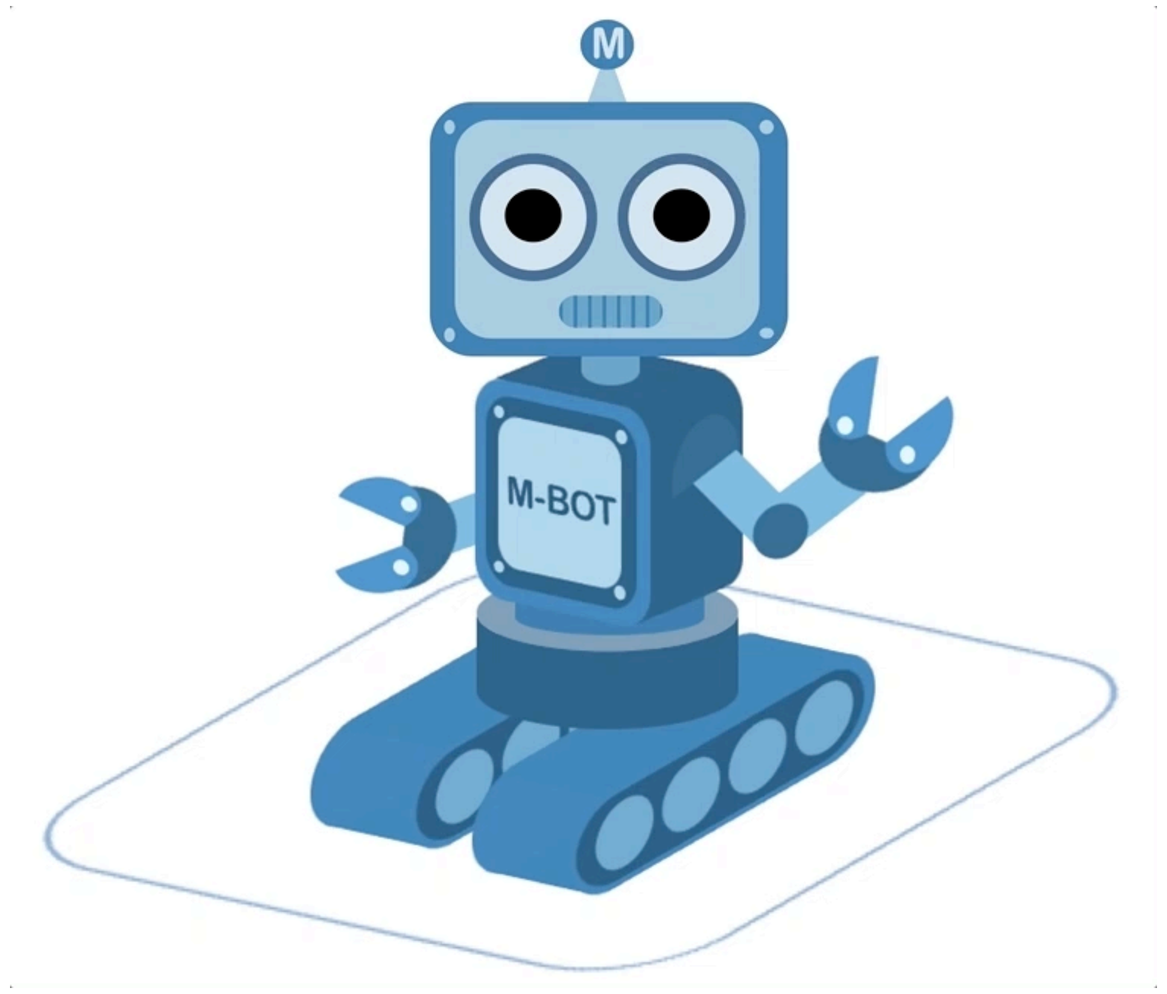
Egocentric vision
Active perception
Sparse rewards

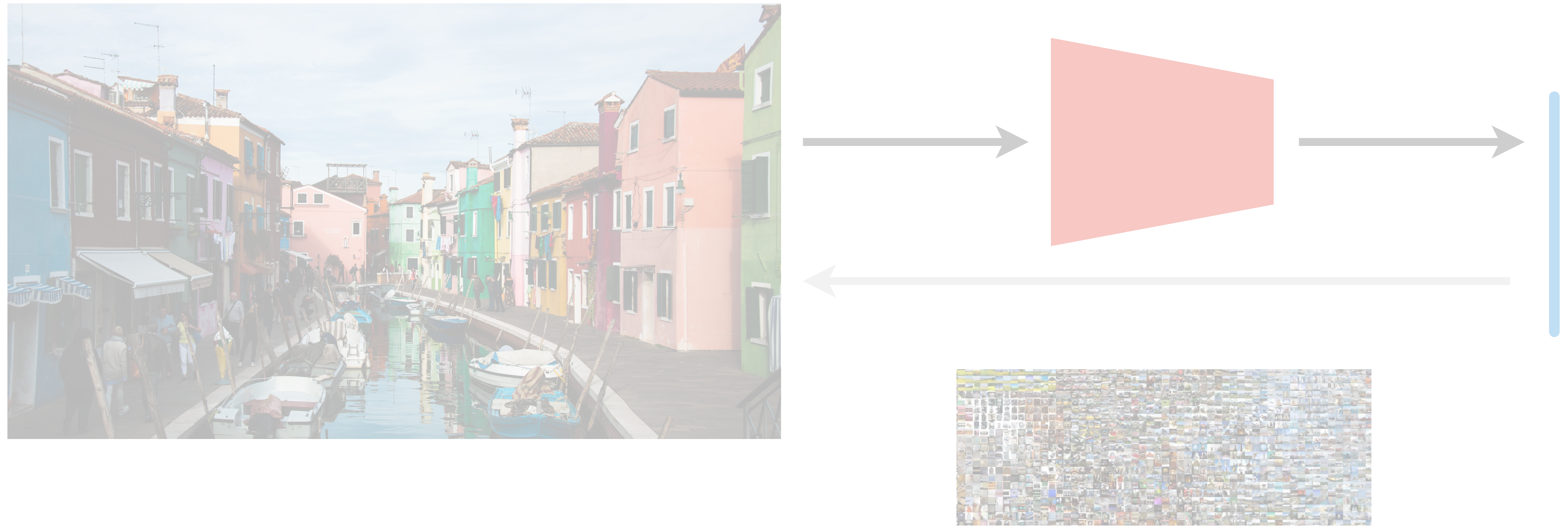


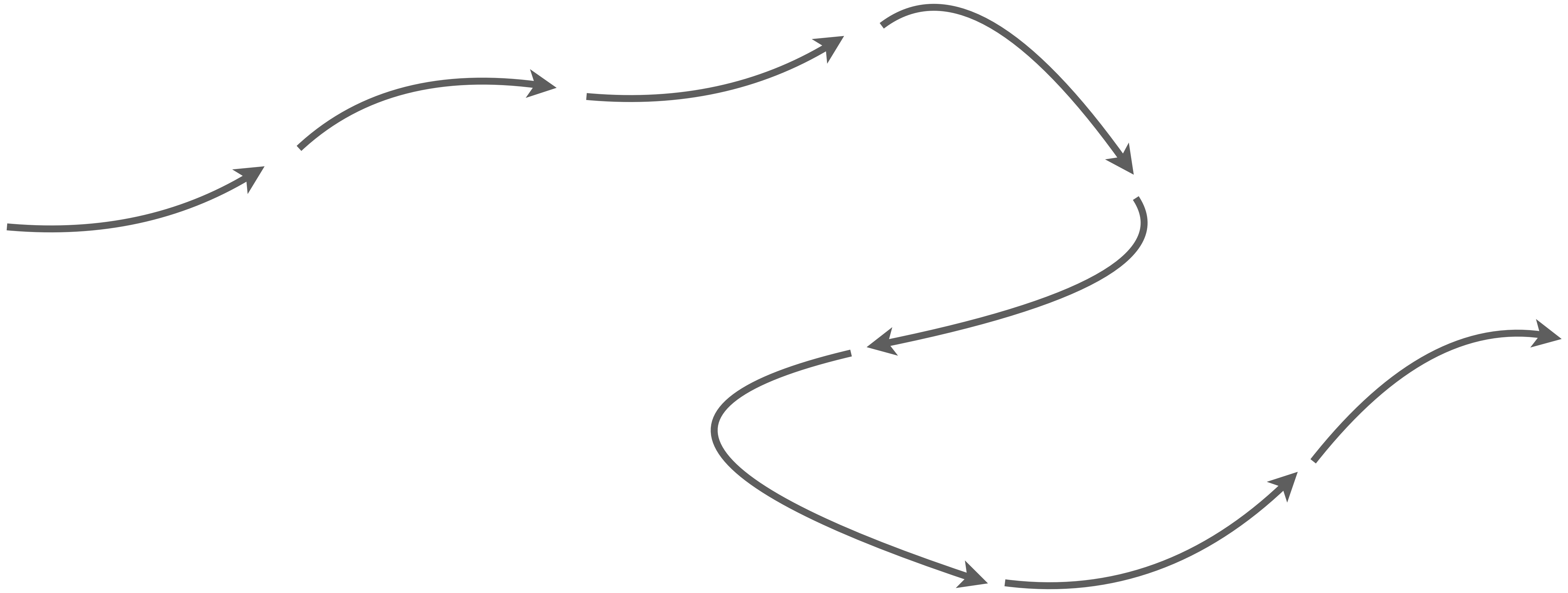


Challenges

Egocentric vision
Active perception
Sparse rewards





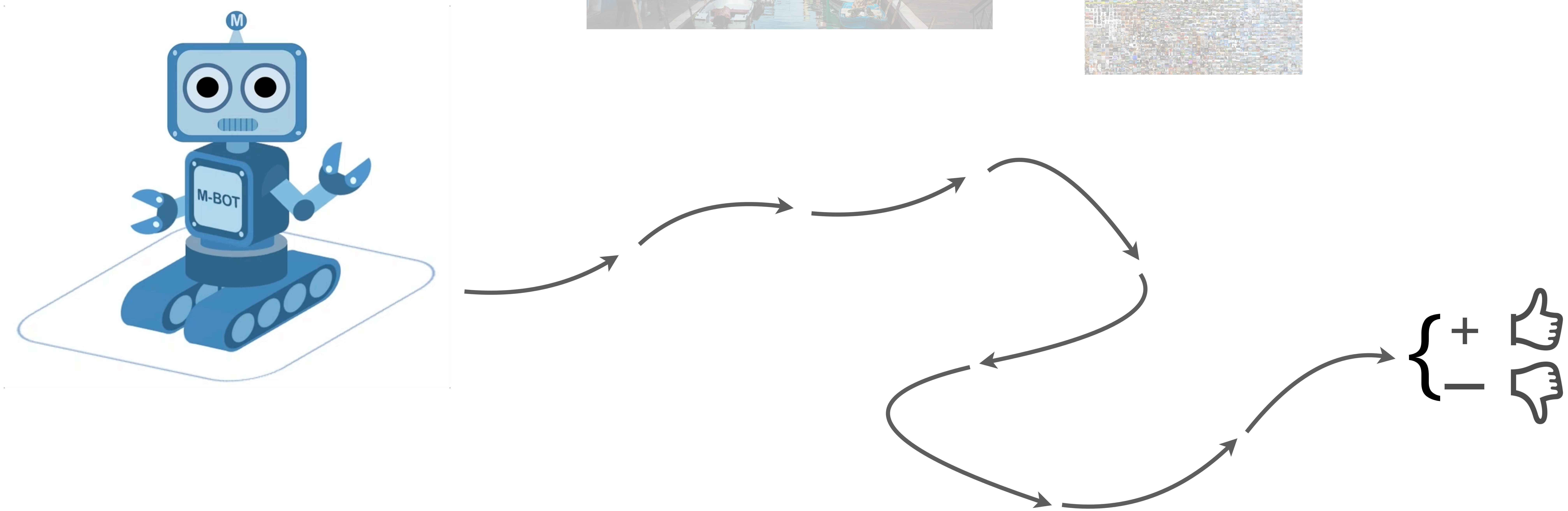


Slide credit: Abhishek Das

Image Credit: Image-Net

Challenges

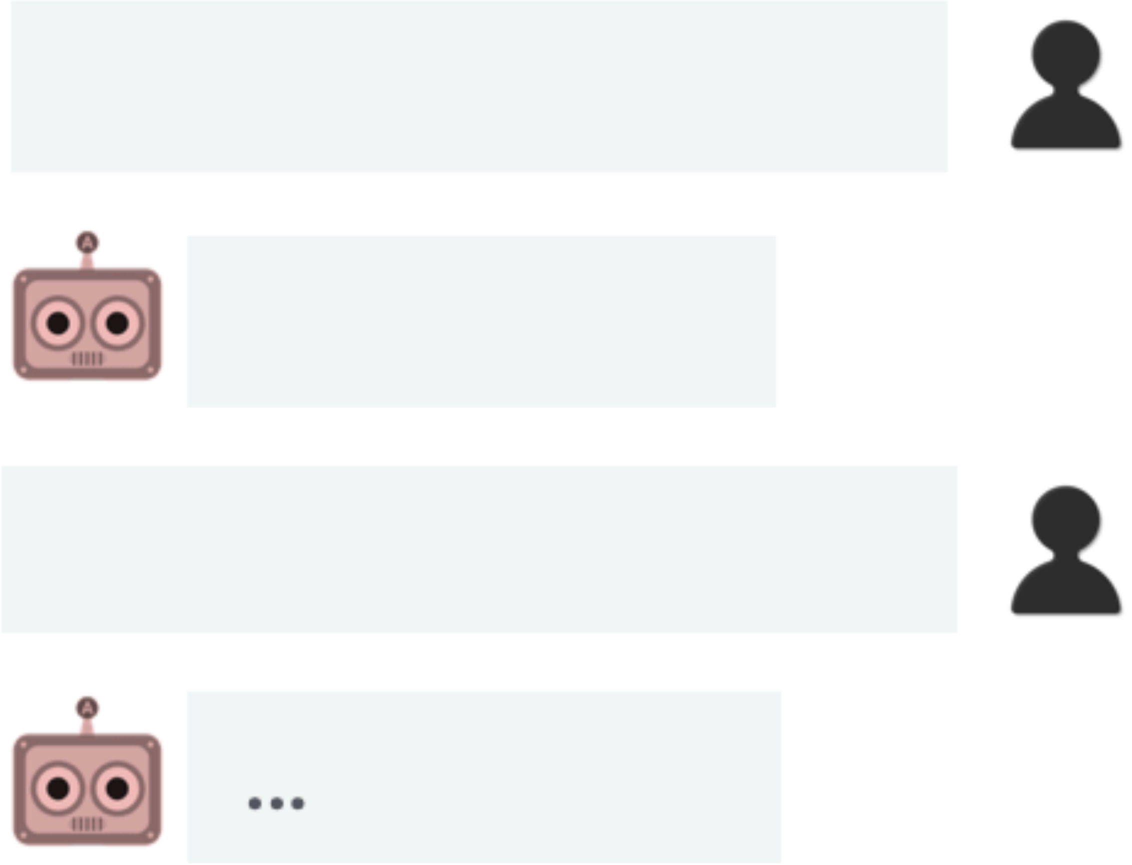
Egocentric vision
Active perception
Sparse rewards



Challenges

Egocentric vision
Active perception
Sparse rewards
Language understanding

Internet AI → Embodied AI



Slide credit: Abhishek Das

Image Credit: Image-Net

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Habitat


Manolis Savva[1,4*]


Abhishek Kadian[1*]


Oleksandr Maksymets[1*]


Yili Zhao[1]


Erik Wijmans[1,2,3]


Bhavana Jain[1]


Julian Straub[2]


Jia Liu[1]


Vladlen Koltun[5]


Jitendra Malik[1,6]


Devi Parikh[1,3]


Dhruv Batra [1,3]

* denotes equal contribution

facebook Artificial Intelligence Research 1

facebook Reality Labs 2

Georgia Tech 3

SFU 4

intel 5

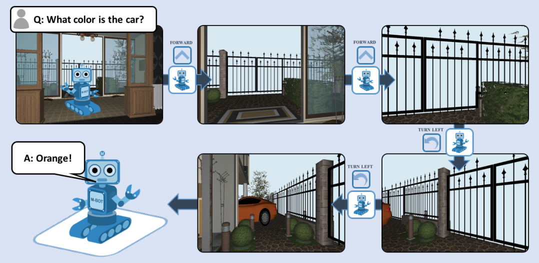
Berkeley UNIVERSITY OF CALIFORNIA 6

Standardizing the Embodied AI "software stack"

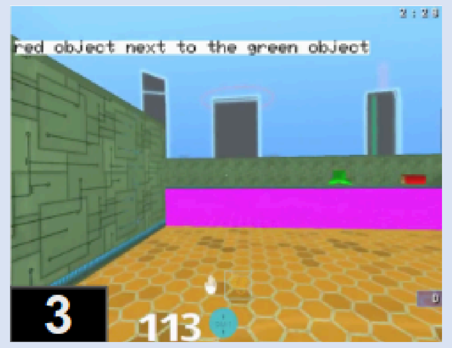
Slide credit: Erik Wijmans

Standardizing the Embodied AI "software stack"

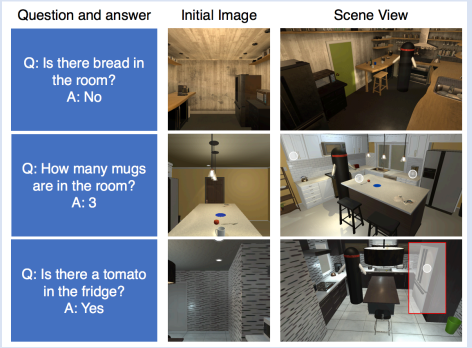
Tasks



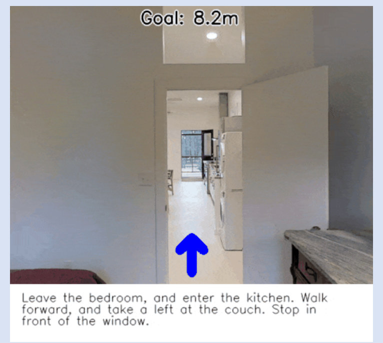
EmbodiedQA
(Das et al., 2018)



Language grounding
(Hill et al., 2017)



Interactive QA
(Gordon et al., 2018)

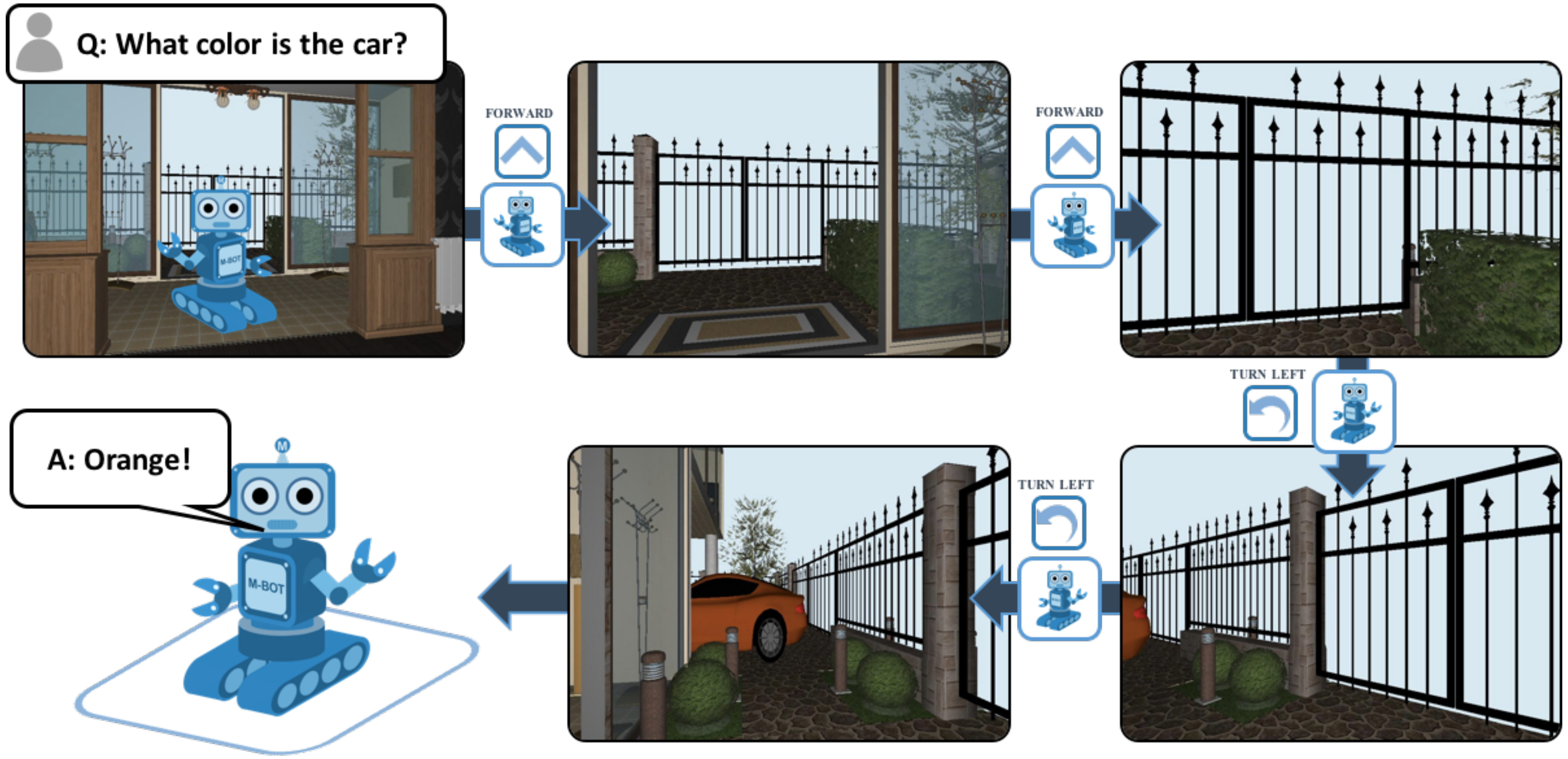


Vision-Language Navigation
(Anderson et al., 2018)

Visual Navigation
(Zhu et al., 2017, Gupta et al., 2017)

Slide credit: Erik Wijmans

Standardizing the Embodied AI "software stack"



EmbodiedQA
(Das et al., 2018)

Slide credit: Erik Wijmans

Standardizing the Embodied AI "software stack"

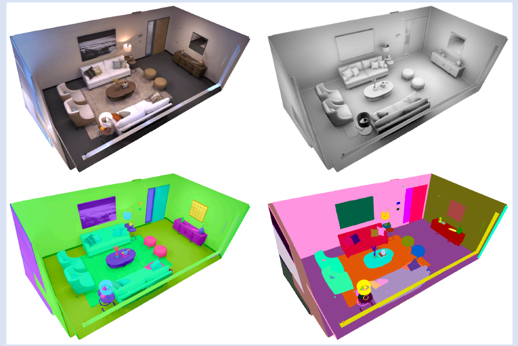
Leave the bedroom, and enter the kitchen. Walk forward, and take a left at the couch. Stop in front of the window.

Vision-Language Navigation
(Anderson et al., 2018)

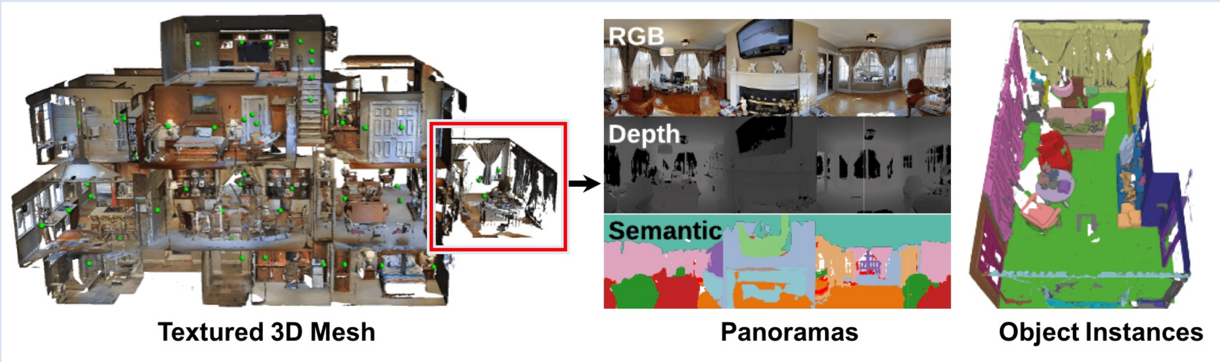
Slide credit: Erik Wijmans

Standardizing the Embodied AI “software stack”

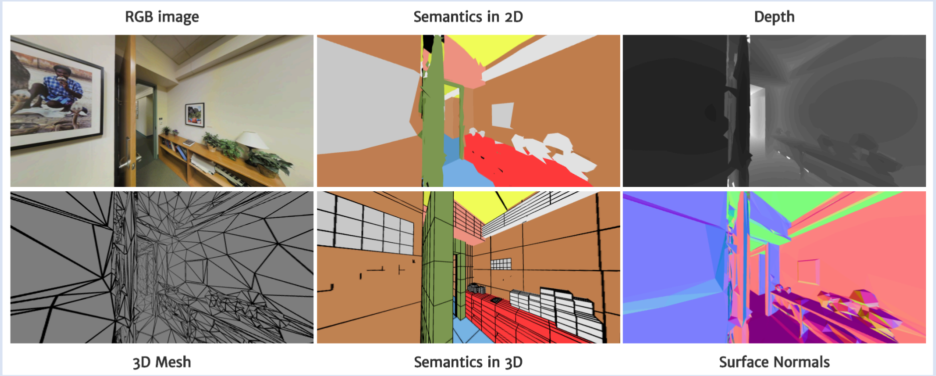
Datasets



Replica (Straub et al., 2019)



Matterport3D (Chang et al., 2017)

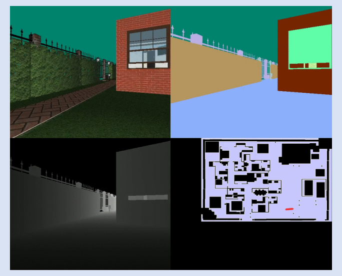


2D-3D-S (Armeni et al., 2017)

Slide credit: Erik Wijmans

Standardizing the Embodied AI "software stack"

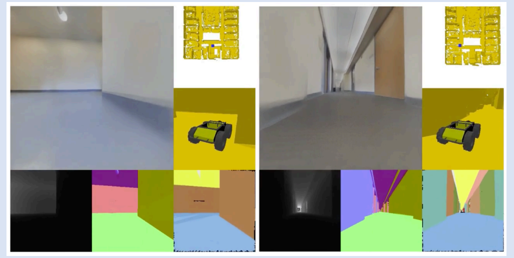
Simulators



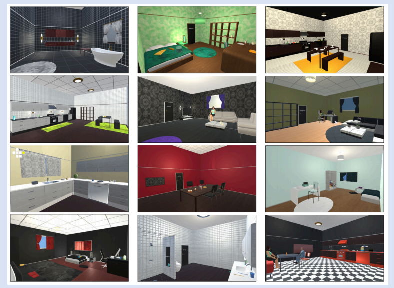
House3D
(Wu et al., 2017)

AI2-THOR
(Kolve et al., 2017)

MINOS
(Savva et al., 2017)

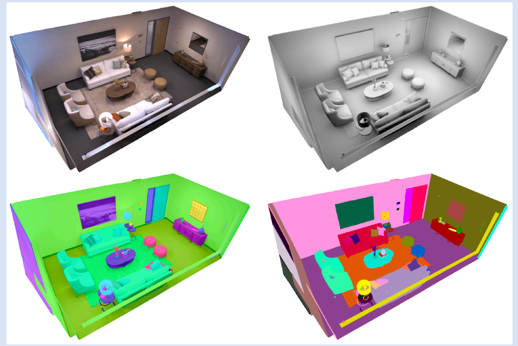


Gibson
(Zamir et al., 2018)

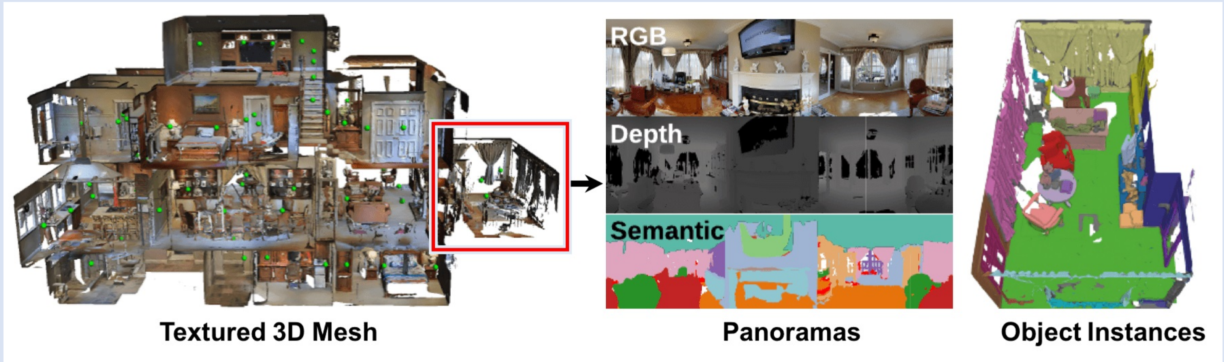


CHALET
(Yan et al., 2018)

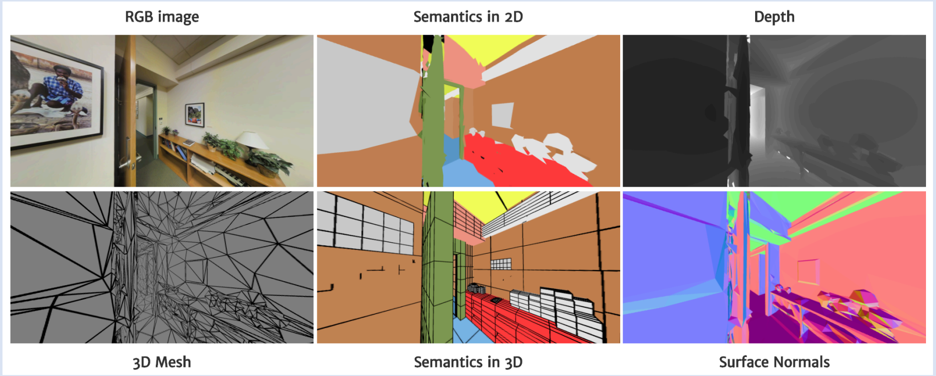
Datasets



Replica (Straub et al., 2019)



Matterport3D (Chang et al., 2017)

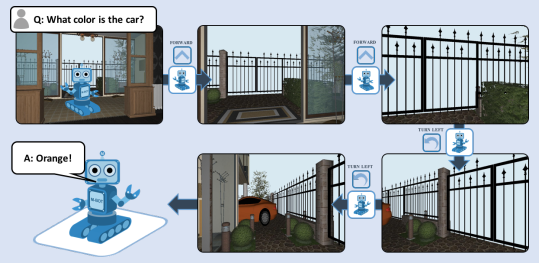


2D-3D-S (Armeni et al., 2017)

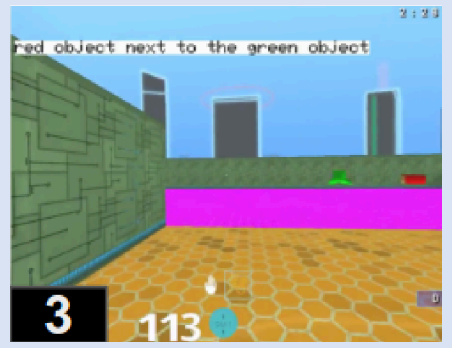
Slide credit: Erik Wijmans

Standardizing the Embodied AI “software stack”

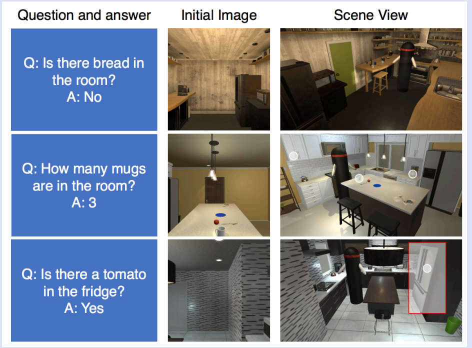
Tasks



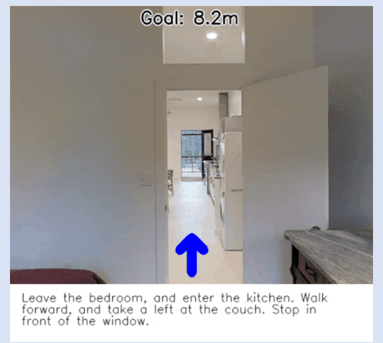
EmbodiedQA
(Das et al., 2018)



Language grounding
(Hill et al., 2017)



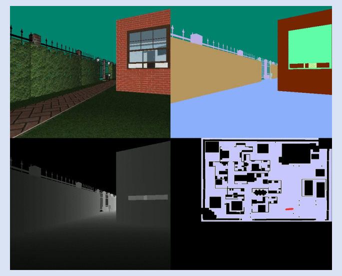
Interactive QA
(Gordon et al., 2018)



Vision-Language Navigation
(Anderson et al., 2018)

Visual Navigation
(Zhu et al., 2017, Gupta et al., 2017)

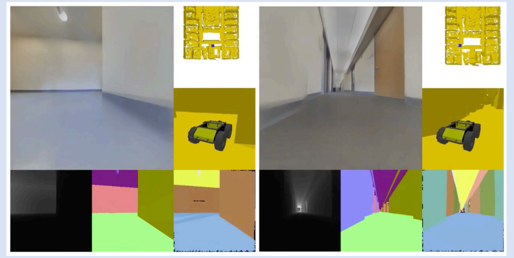
Simulators



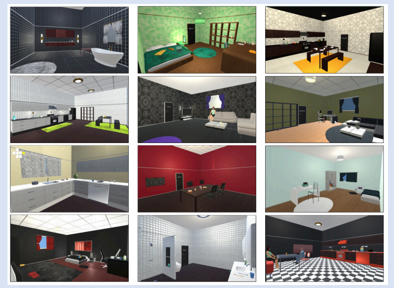
House3D
(Wu et al., 2017)

AI2-THOR
(Kolve et al., 2017)

MINOS
(Savva et al., 2017)

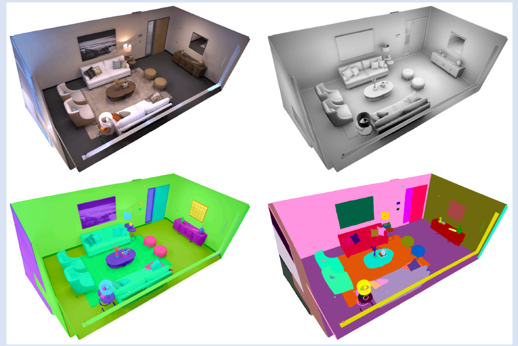


Gibson
(Zamir et al., 2018)

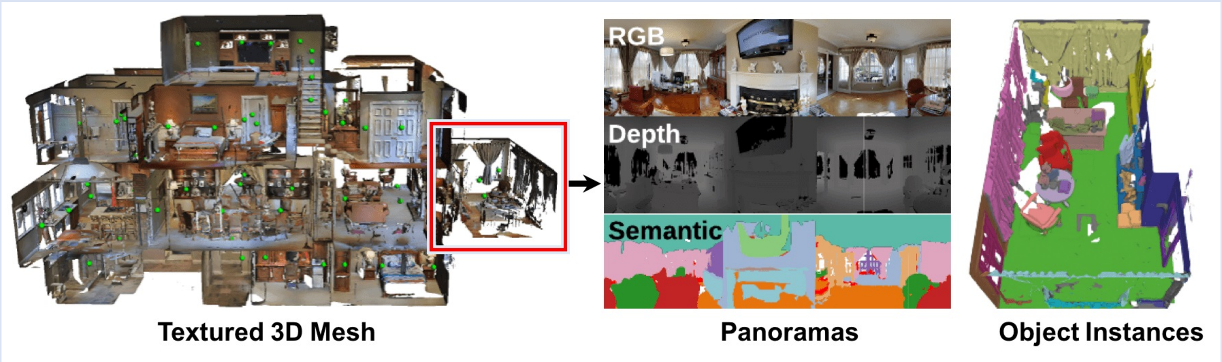


CHALET
(Yan et al., 2018)

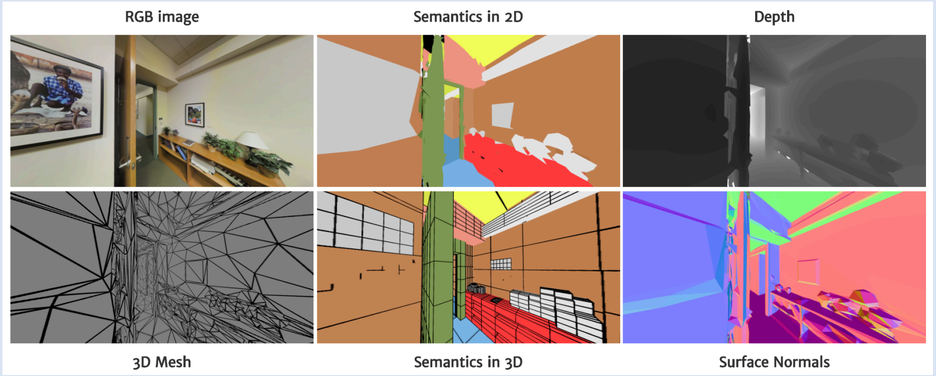
Datasets



Replica (Straub et al., 2019)



Matterport3D (Chang et al., 2017)

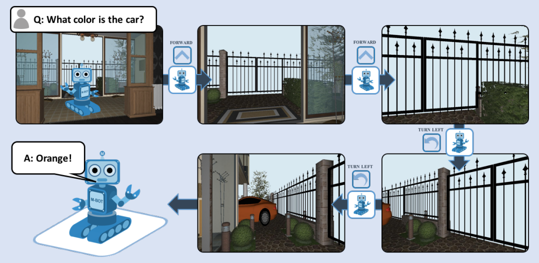


2D-3D-S (Armeni et al., 2017)

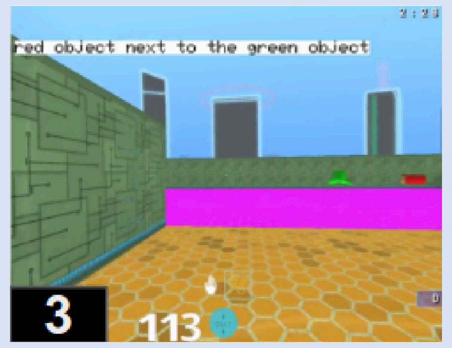
Slide credit: Erik Wijmans

Standardizing the Embodied AI "software stack"

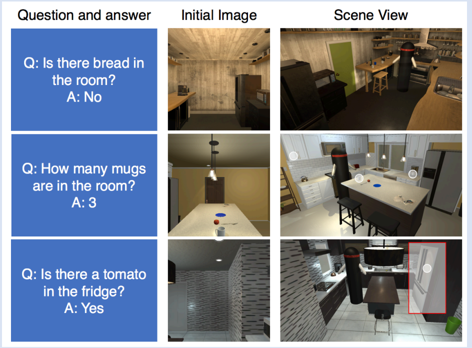
Tasks



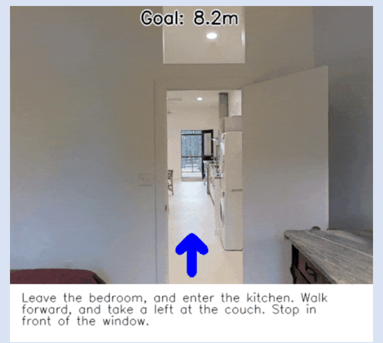
EmbodiedQA
(Das et al., 2018)



Language grounding
(Hill et al., 2017)



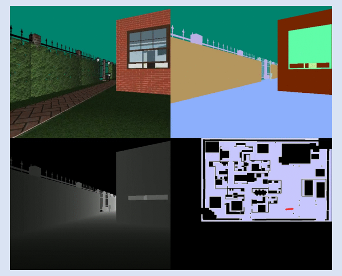
Interactive QA
(Gordon et al., 2018)



Vision-Language Navigation
(Anderson et al., 2018)

Visual Navigation
(Zhu et al., 2017, Gupta et al., 2017)

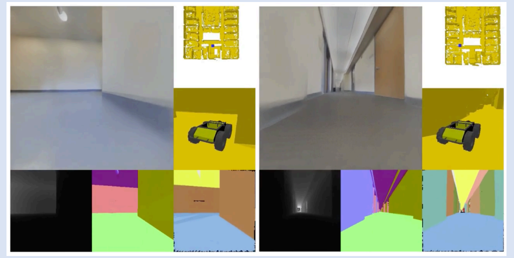
Simulators



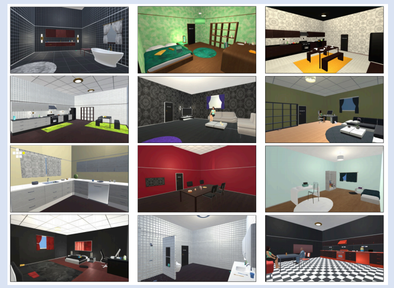
House3D
(Wu et al., 2017)

AI2-THOR
(Kolve et al., 2017)

MINOS
(Savva et al., 2017)

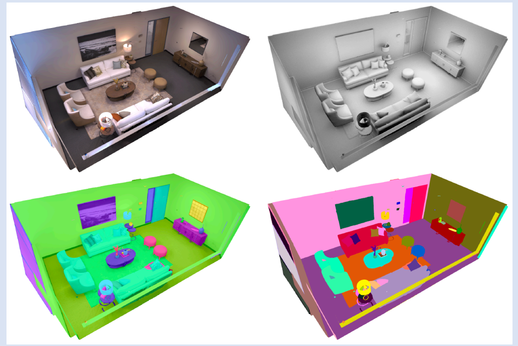


Gibson
(Zamir et al., 2018)

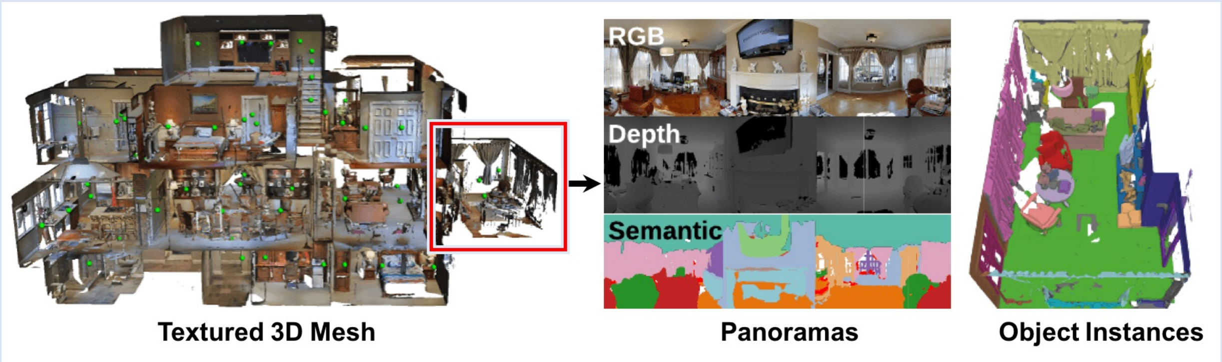


CHALET
(Yan et al., 2018)

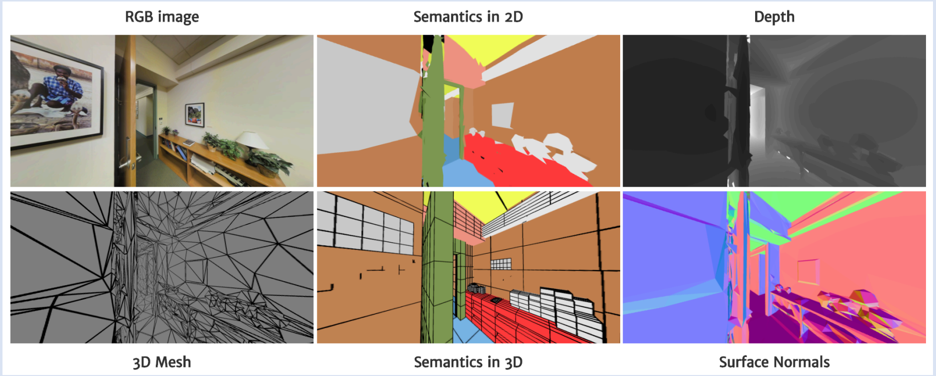
Datasets



Replica (Straub et al., 2019)



Matterport3D (Chang et al., 2017)



2D-3D-S (Armeni et al., 2017)

Habitat Platform

Habitat-API

↑

Habitat-Sim

↑

Generic Dataset Support

Slide credit: Erik Wijmans

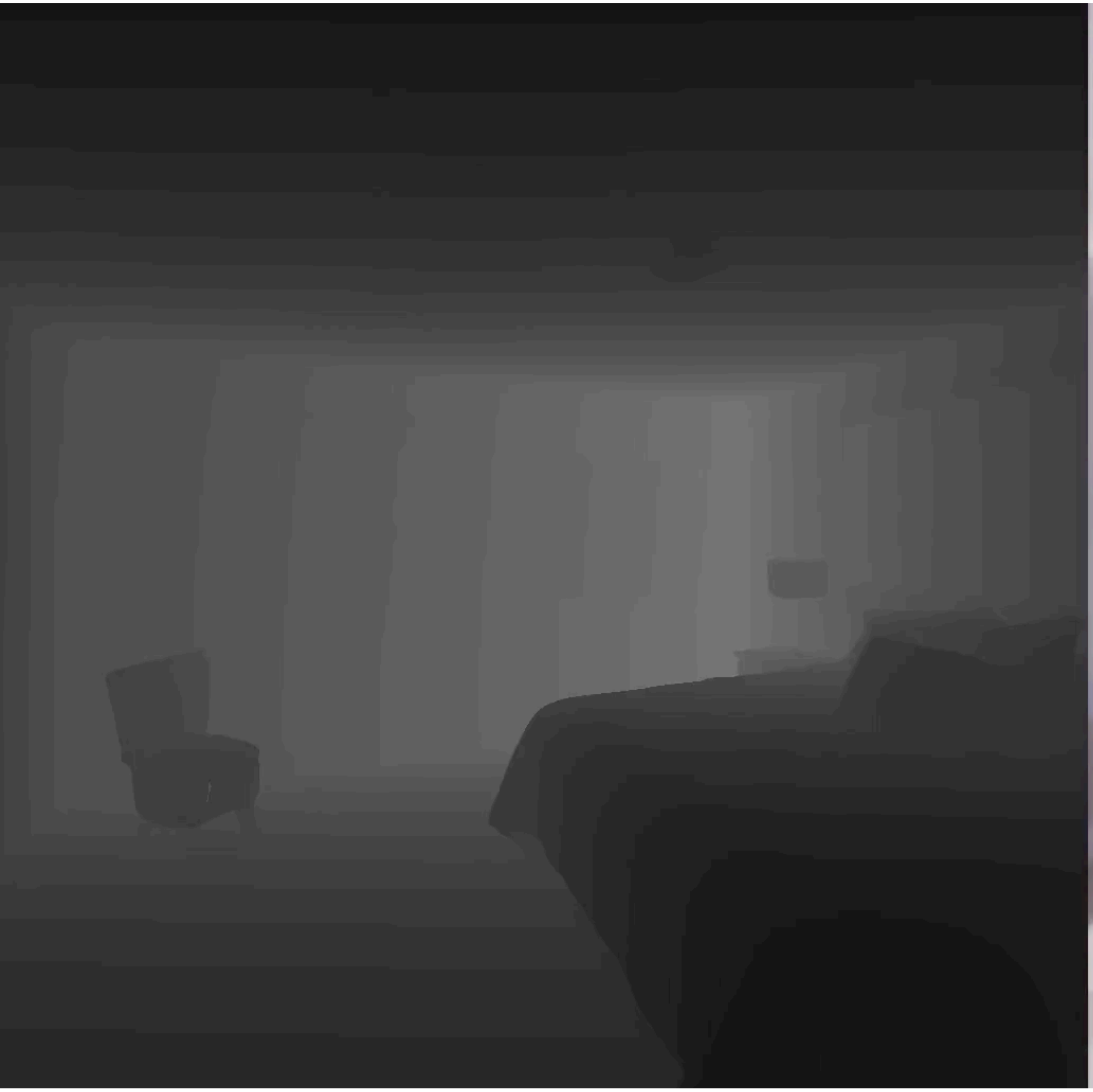
Habitat-Sim

Habitat-Sim Demo

Bring Your Own Scan: Virtualizing Reality

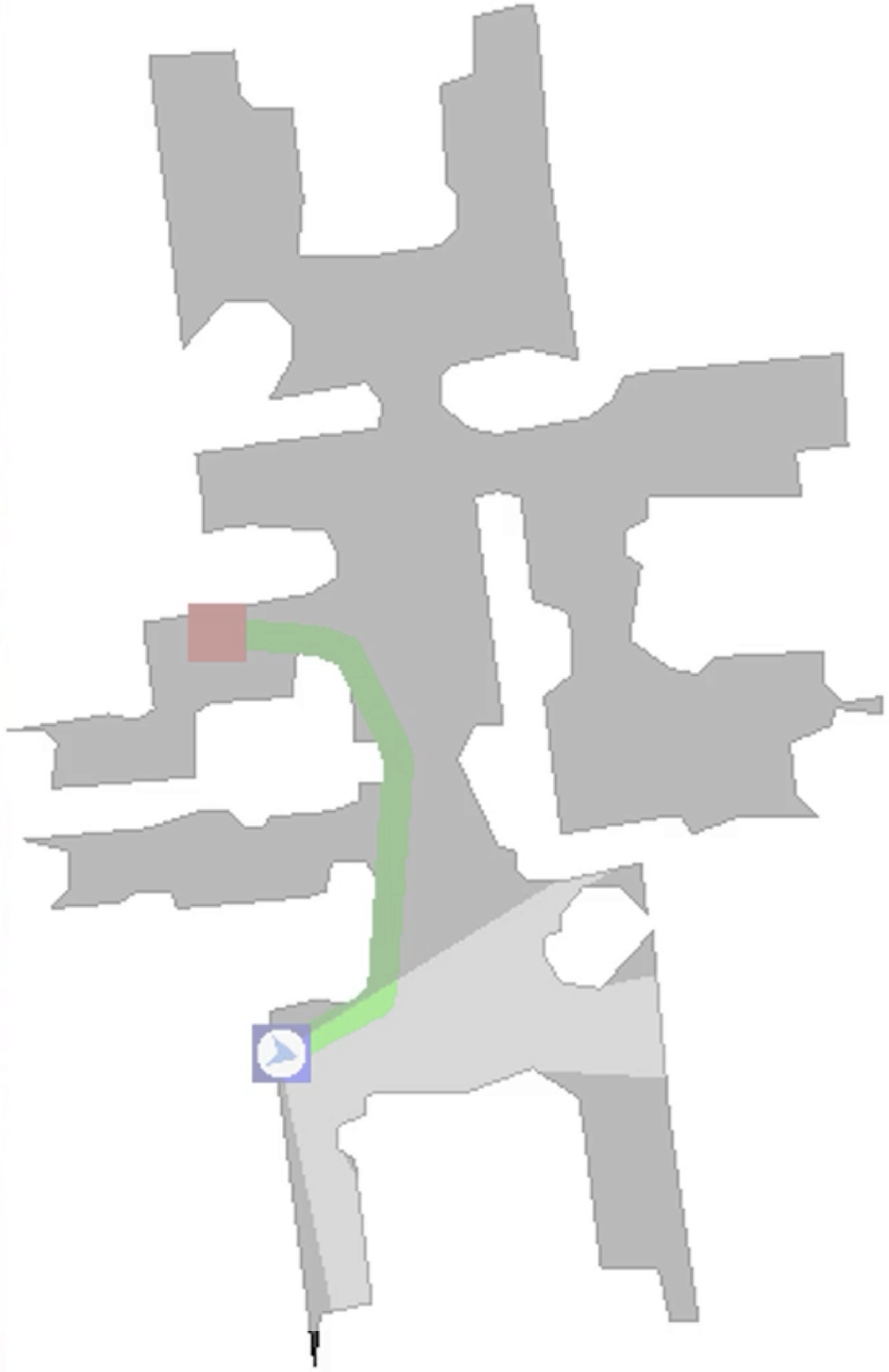
PointGoal Navigation

PointGoal Navigation



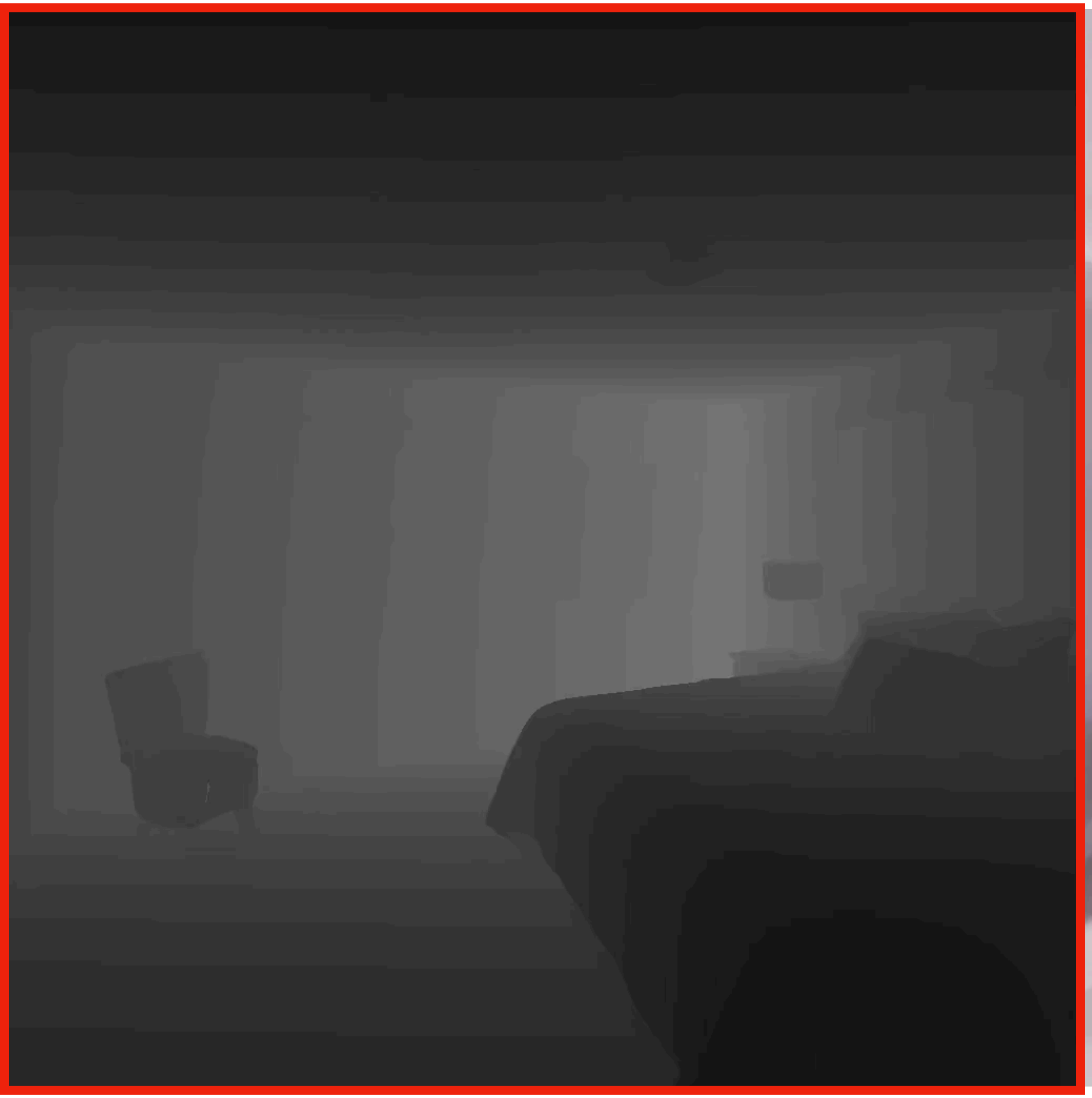
Depth

RGB and GPS+Compass

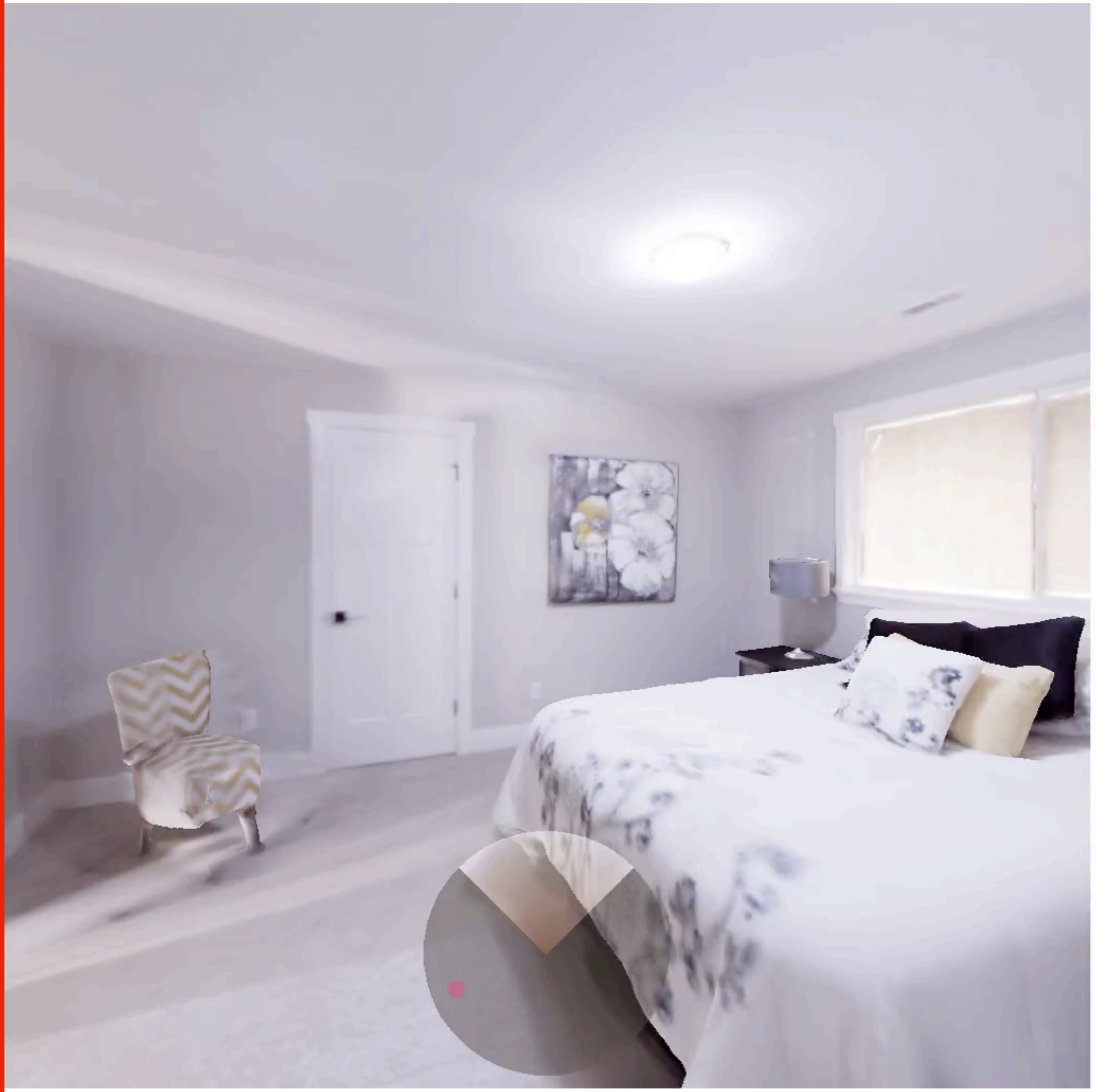


Top Down Map

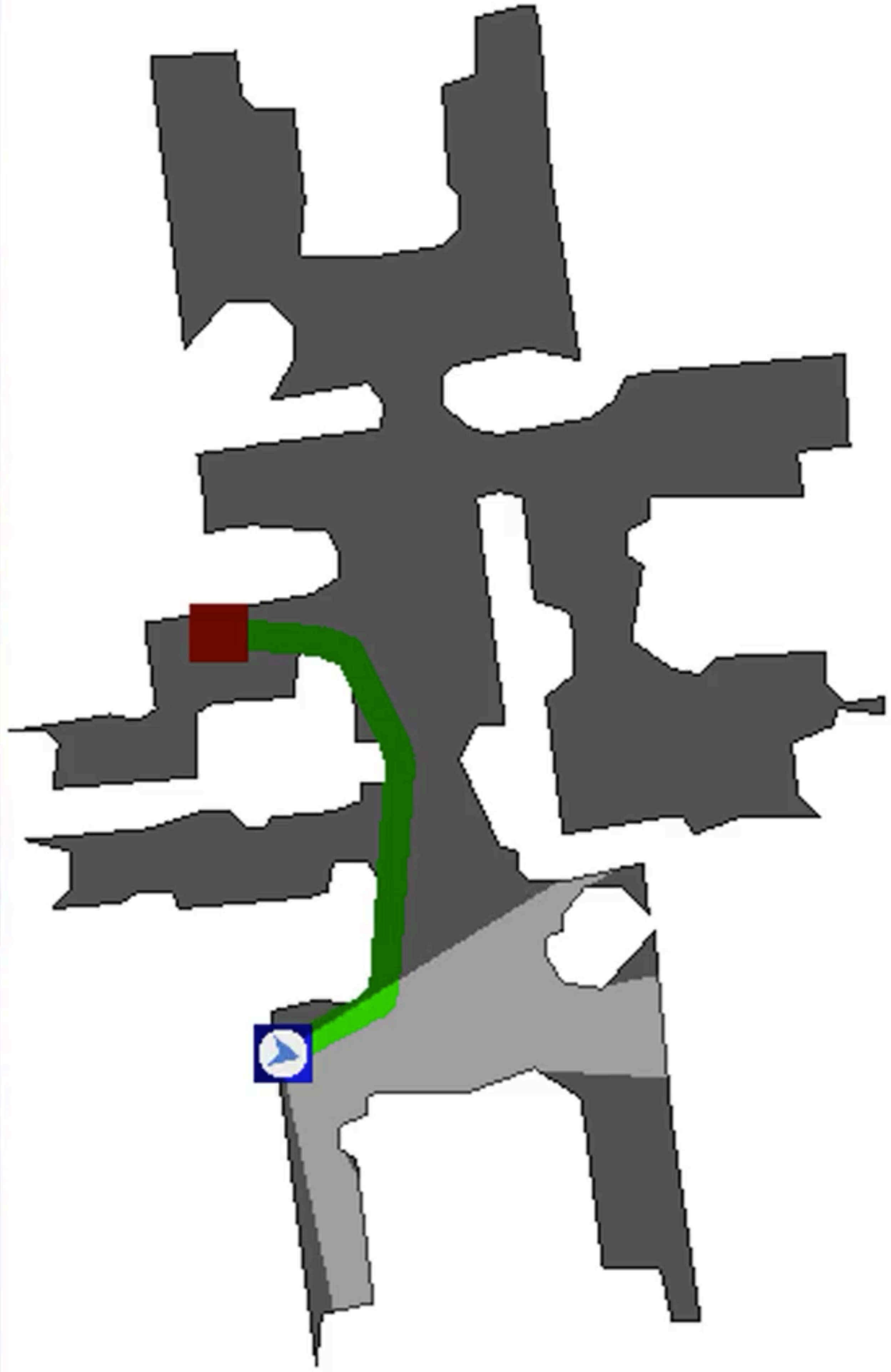
PointGoal Navigation



Depth



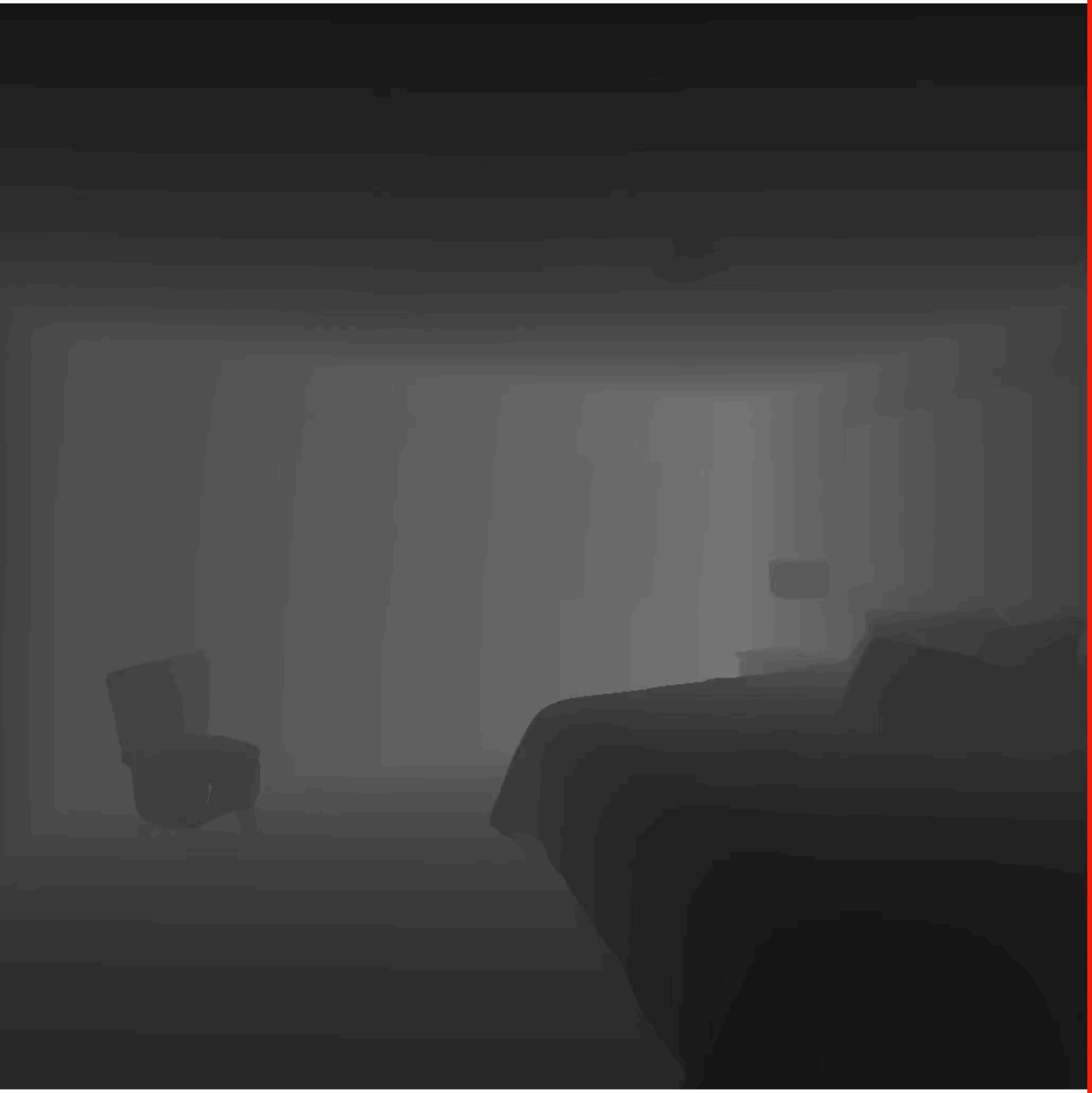
RGB and GPS+Compass



Top Down Map

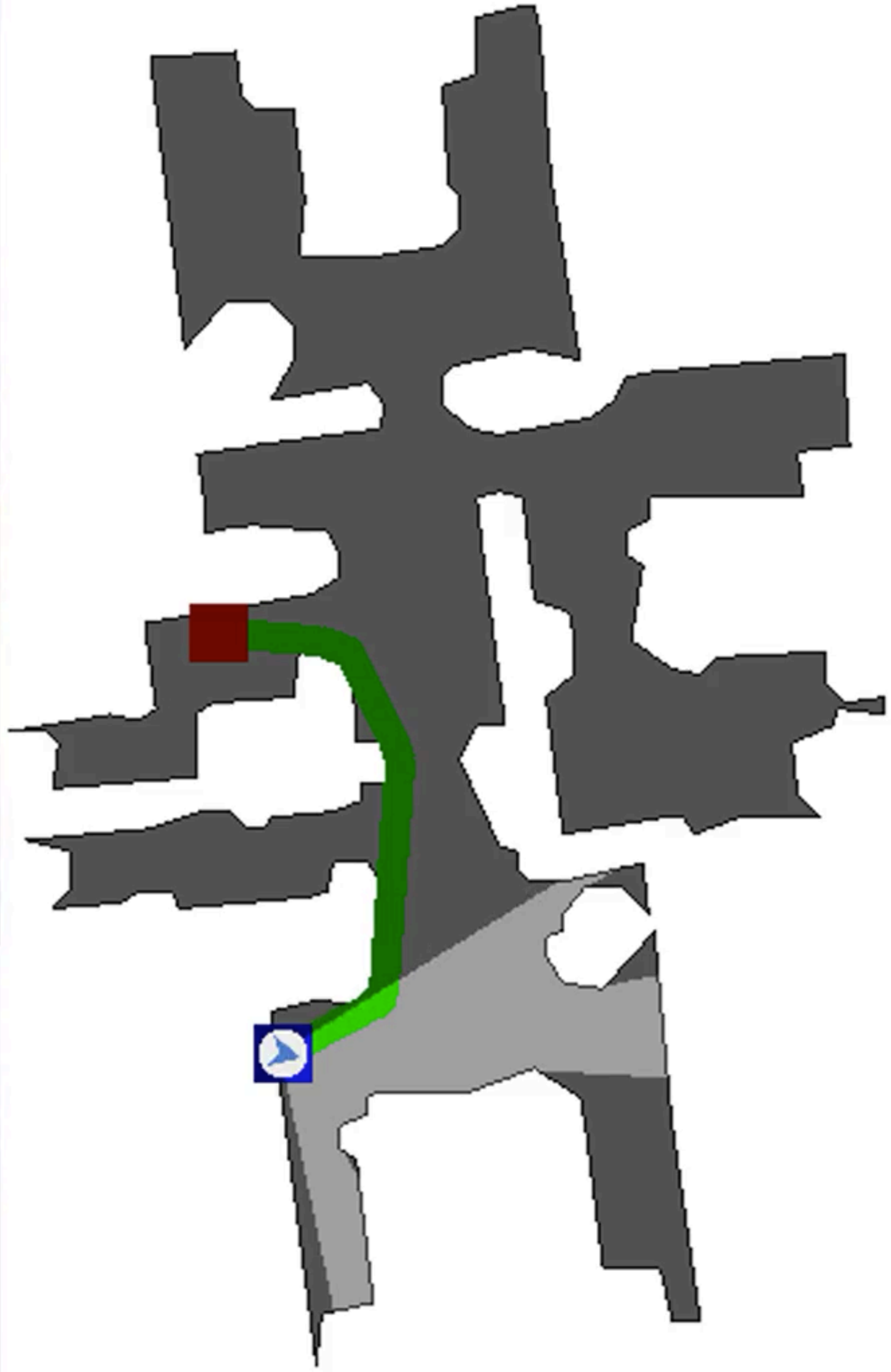
Slide credit: Erik Wijmans

PointGoal Navigation



Depth

RGB and GPS+Compass



Top Down Map

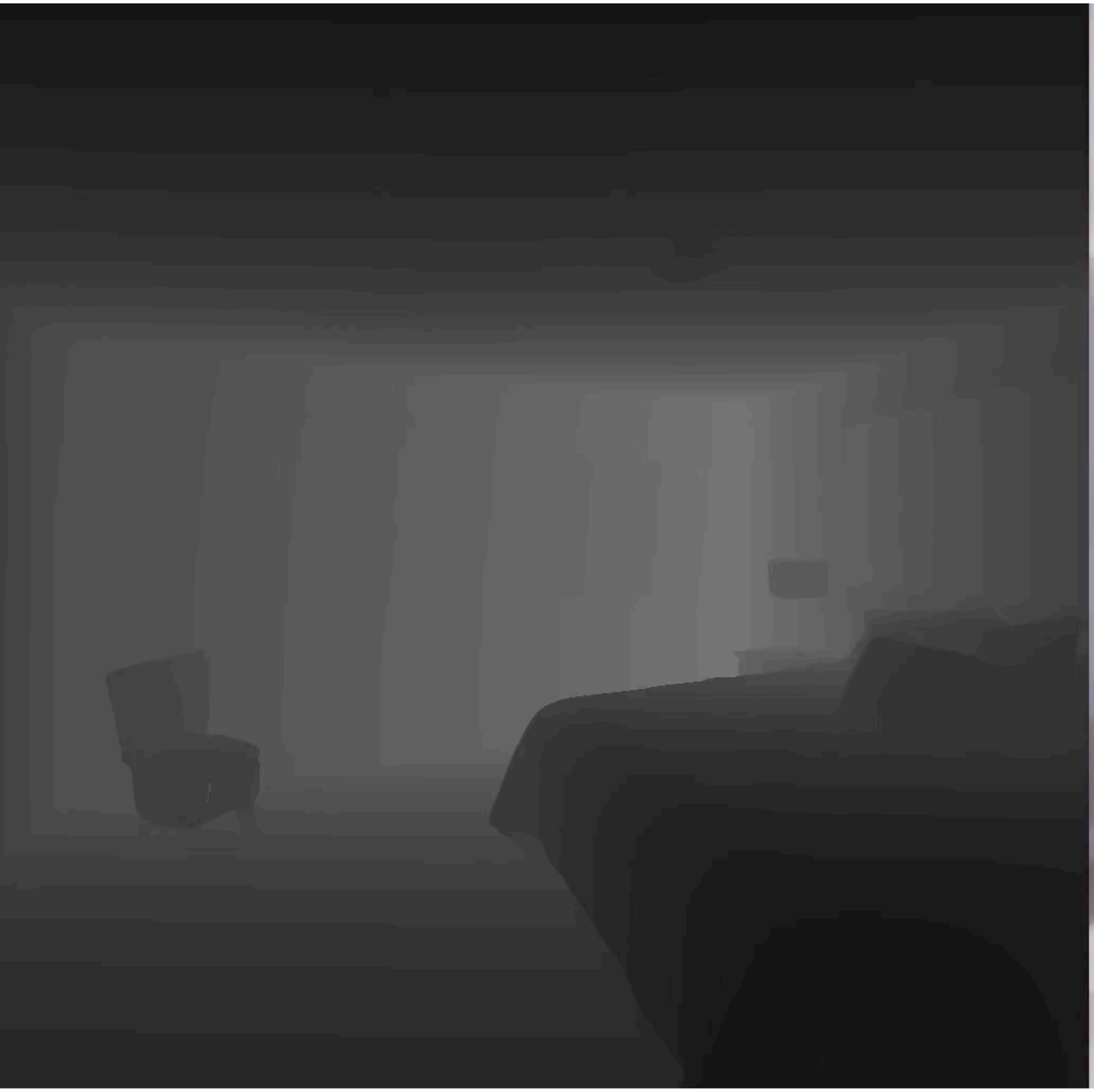
Slide credit: Erik Wijmans

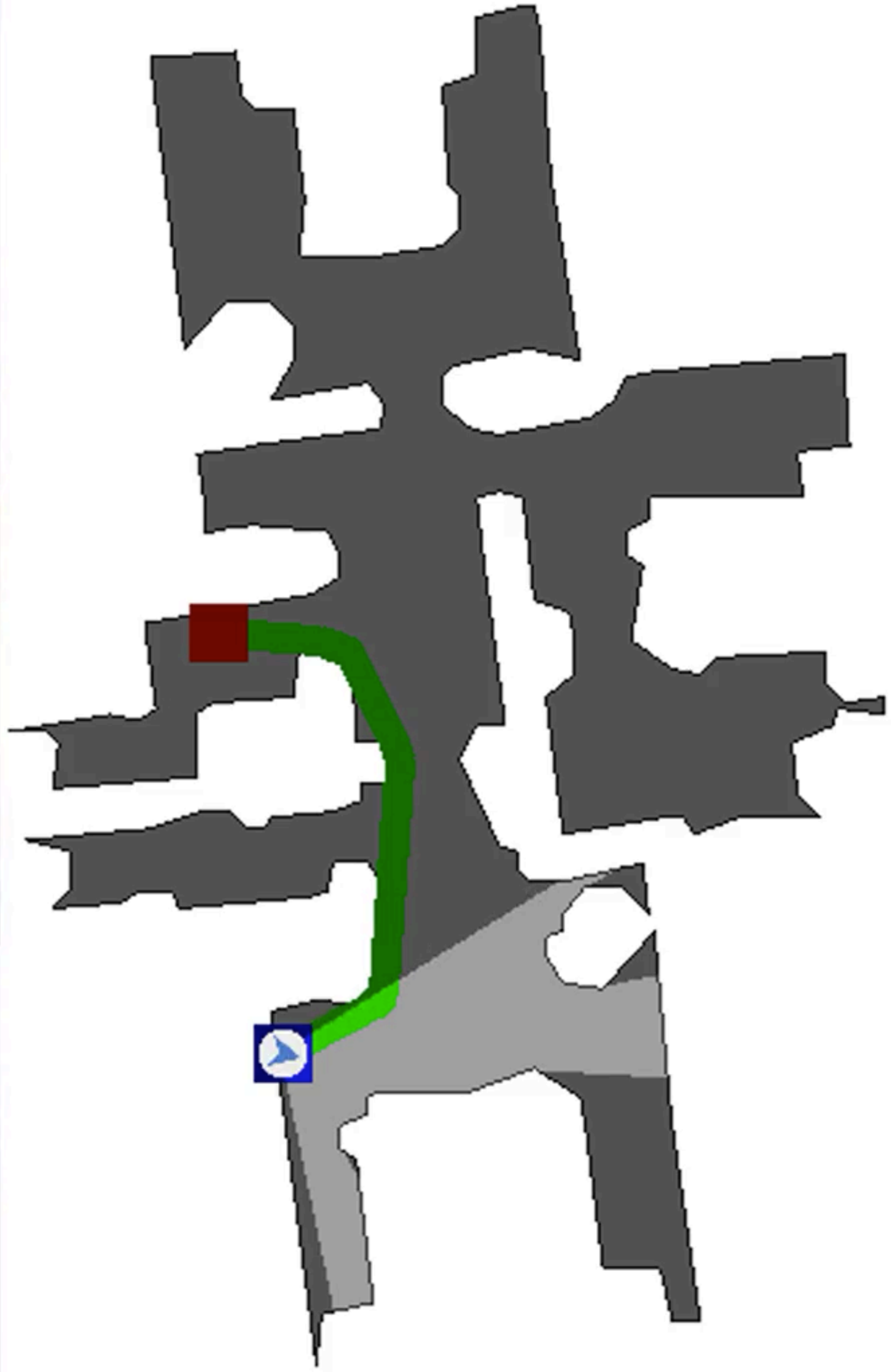
PointGoal Navigation

Depth RGB and GPS+Compass Top Down Map

Slide credit: Erik Wijmans

PointGoal Navigation





Depth

RGB and GPS+Compass

Top Down Map

Slide credit: Erik Wijmans

Agent and Model Design

Agent and Model Design

Agent and Model Design

Slide credit: Erik Wijmans

Agent and Model Design

- 0.6m tall cylinder with 0.17m radius

Agent and Model Design

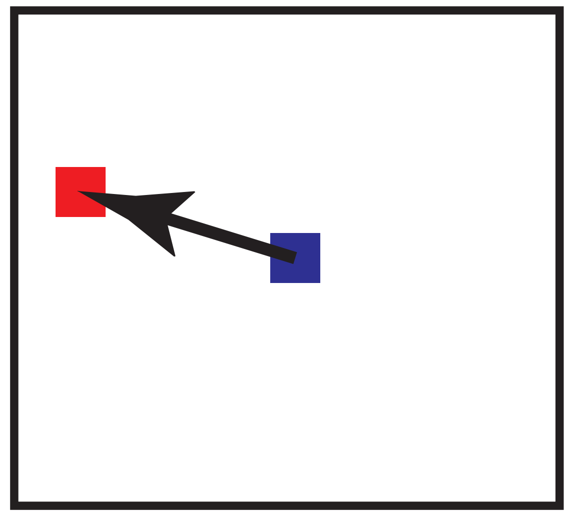
- 0.6m tall cylinder with 0.2m radius
- Actions:
  - <stop>: Indicates the agent believes it has completed the task
  - <forward>: Moves 0.25m forward
  - <left>, <right>: Turn 30 degrees

Agent and Model Design

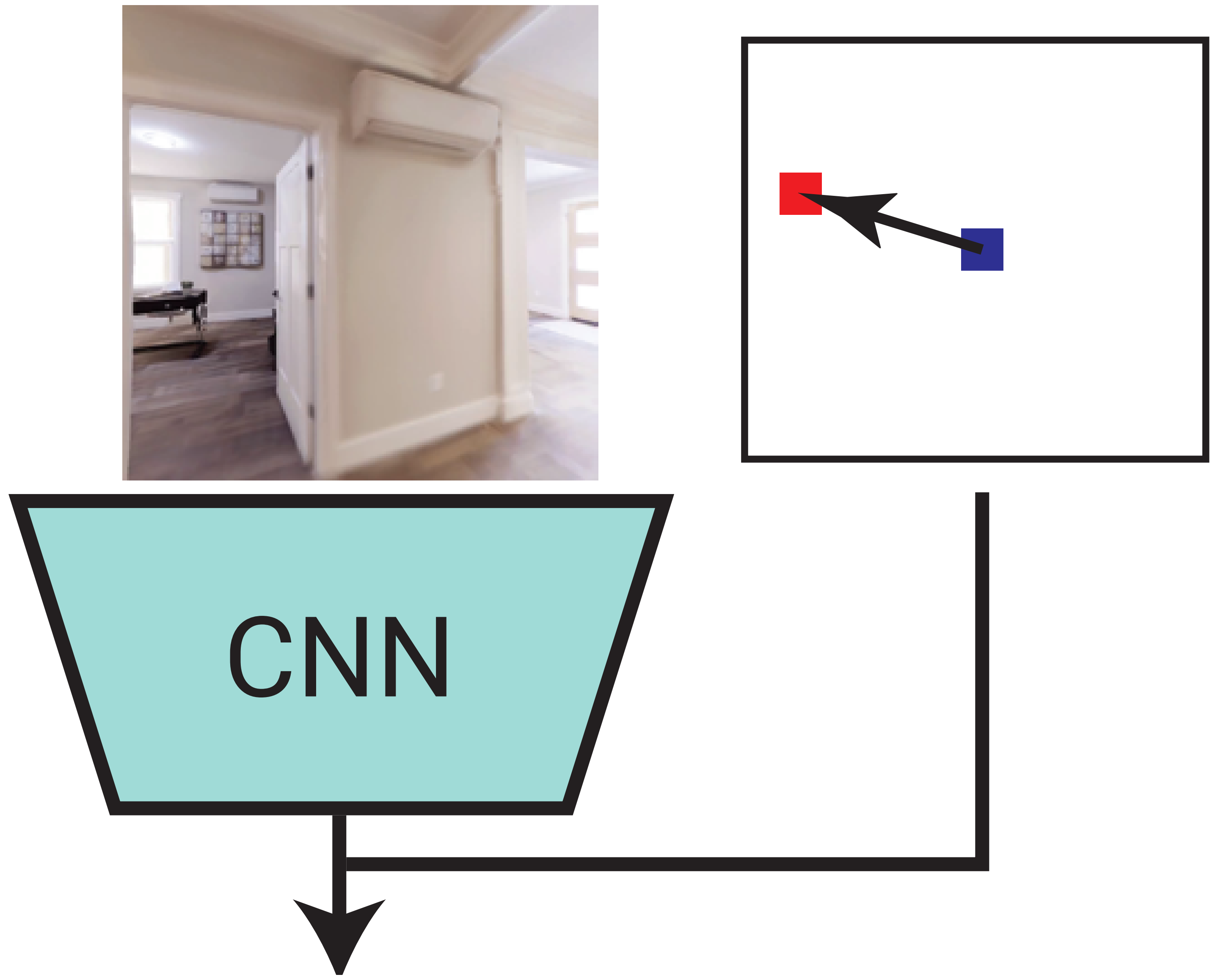
Agent and Model Design

Slide credit: Erik Wijmans

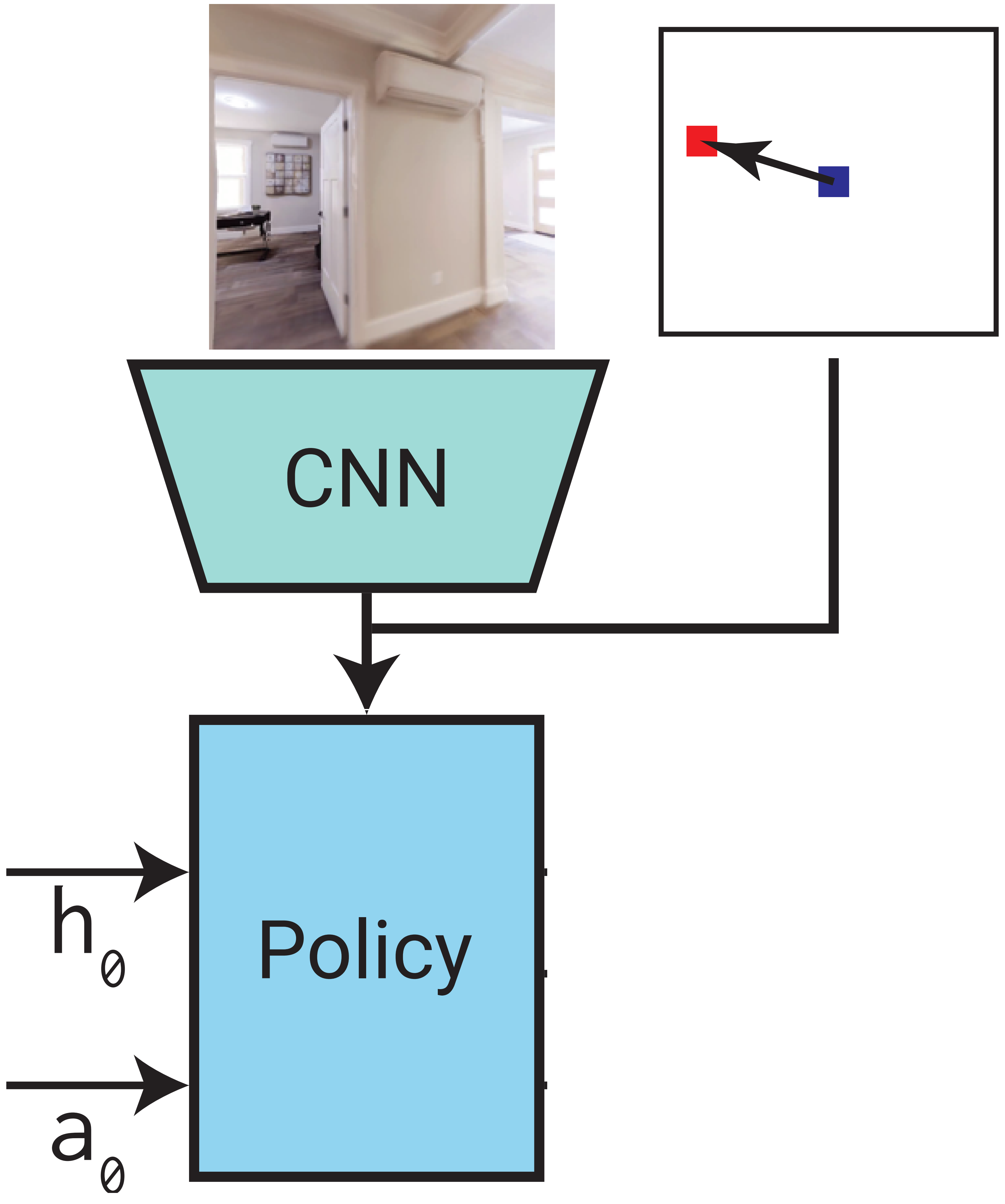
Agent and Model Design



Agent and Model Design

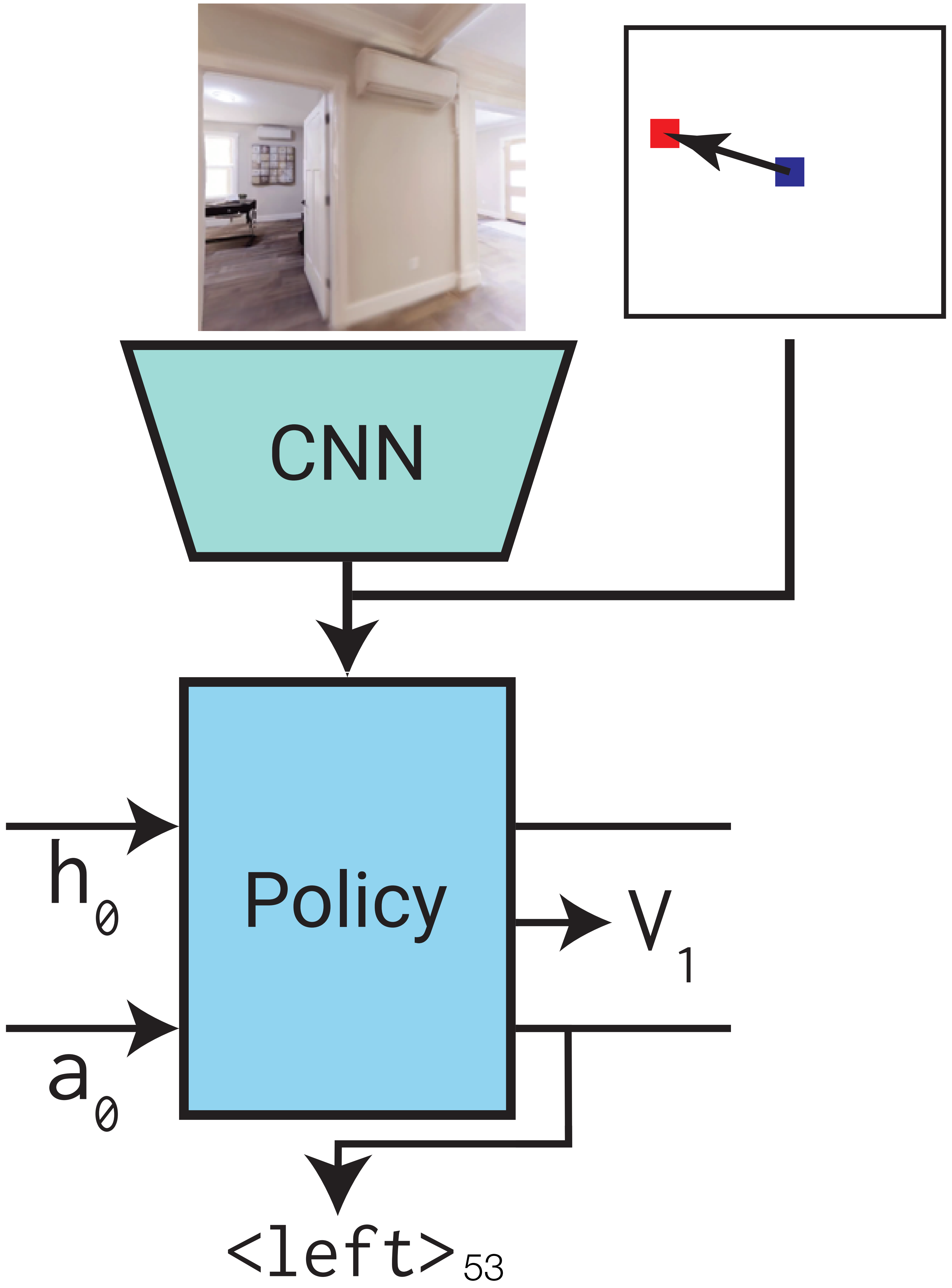


Agent and Model Design



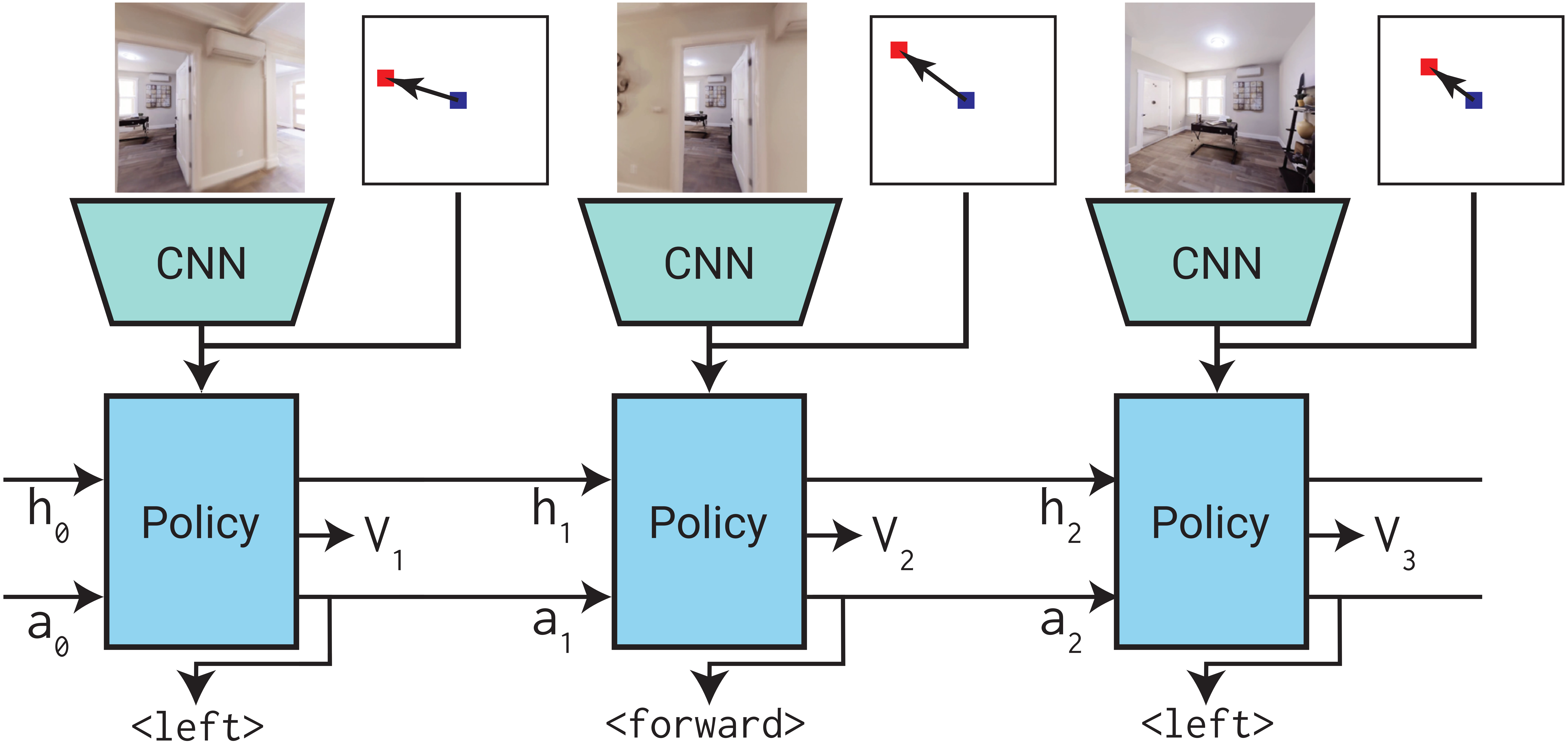
Slide credit: Erik Wijmans

Agent and Model Design



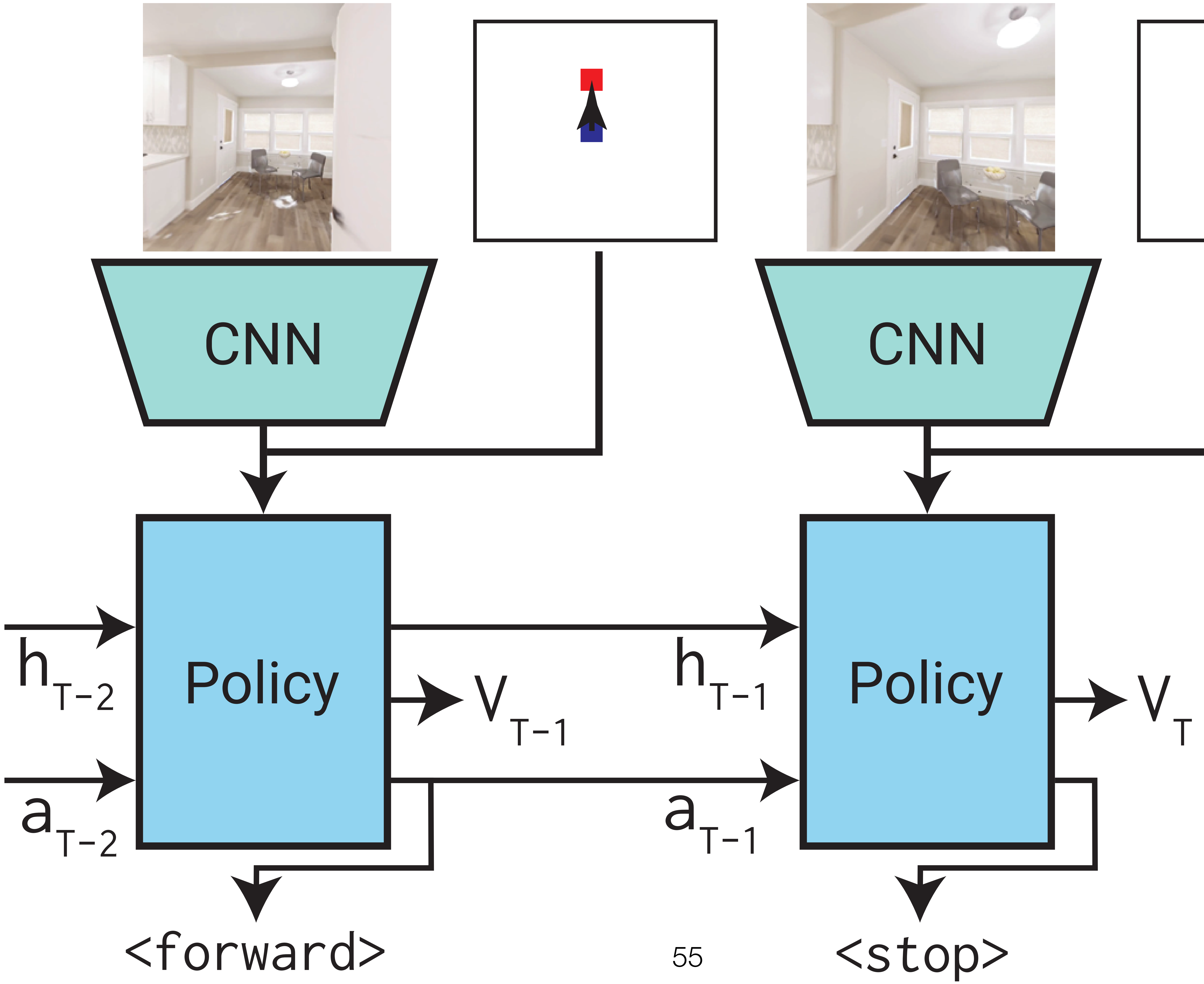
Slide credit: Erik Wijmans

Agent and Model Design



Slide credit: Erik Wijmans

Agent and Model Design



Slide credit: Erik Wijmans

Agent and Model Design

- How do we train this agent?

Agent and Model Design

- How do we train this agent?
- Both actions (they are discrete) and the simulation are non-differential-able

Agent and Model Design

- How do we train this agent?
- Both actions (they are discrete) and the simulation are non-differential-able
- Use reinforcement learning!

Slide credit: Erik Wijmans

Outline

- Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation
    - Sim2Real Transfer
    - Robot2Robot Transfer

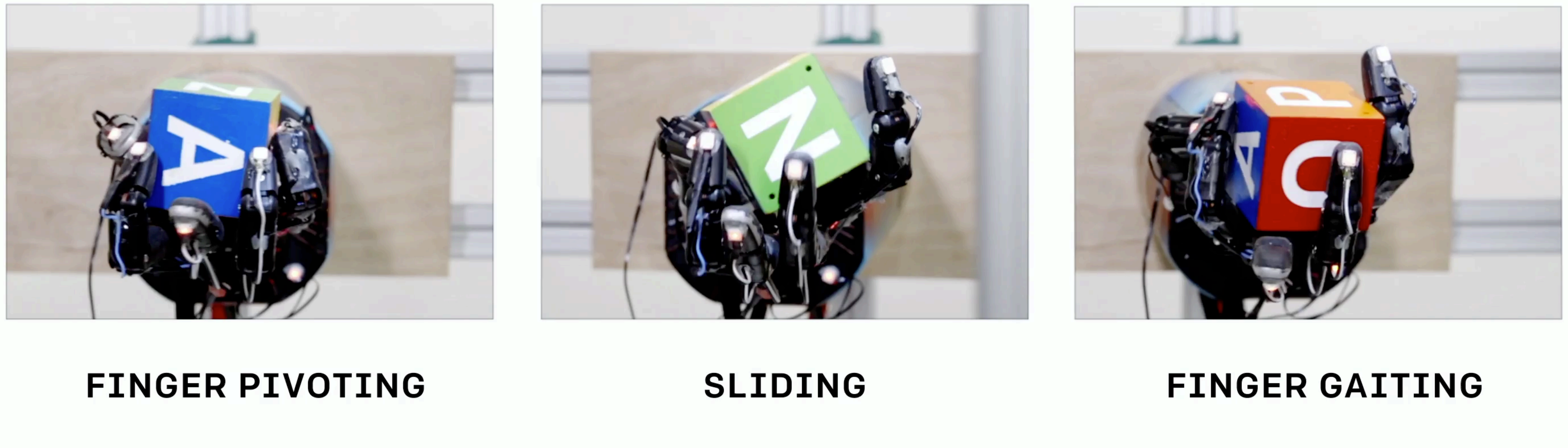
Outline

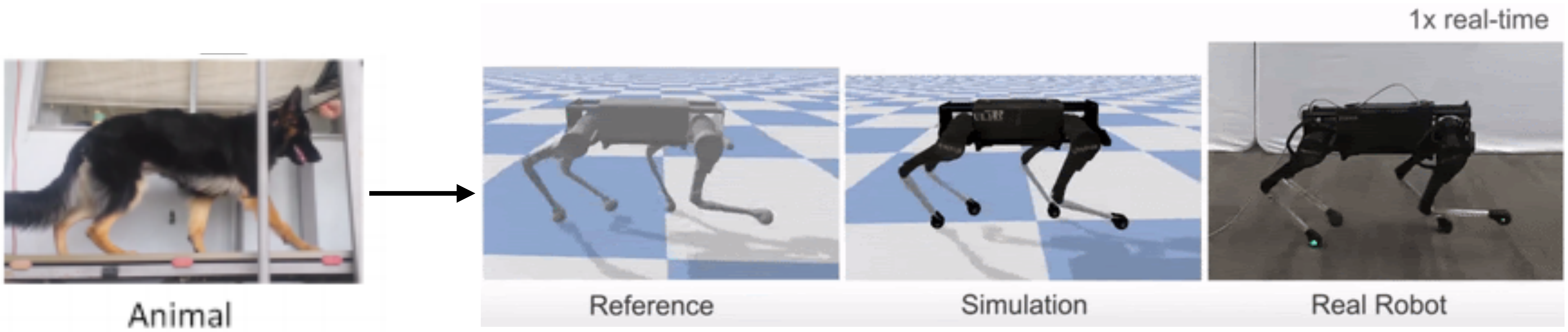
- Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation
    - Sim2Real Transfer
    - Robot2Robot Transfer

Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO)

AlphaStar: Mastering the Real-Time Strategy Game StarCraft II





Slide credit: Erik Wijmans

Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO)

Given a policy: $\pi_{\theta_{\text{old}}}(a_t \mid s_t)$

Proximal Policy Optimization (PPO)

Given a policy: $\pi_{\theta_{\text{old}}}(a_t \mid s_t)$

Objective— maximize probability ratio: $ratio_t(\theta) = \dfrac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$

Slide credit: Erik Wijmans

Proximal Policy Optimization (PPO)

Given a policy: $$\pi_{\theta_{\text{old}}}(a_t \mid s_t)$$

Objective— maximize probability ratio: $$ratio_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\text{old}}}(a_t \mid s_t)}$$

Advantage: $$A_{\pi_{\theta_{old}}}(s_t, a_t) = Q_{\pi_{\theta_{old}}}(s, a) - V_{\pi_{\theta_{old}}}(s)$$

Slide credit: Erik Wijmans

Proximal Policy Optimization (PPO)

Given a policy: $$\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)$$

Objective— maximize probability ratio: $$ratio_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{\mathrm{old}}}(a_t \mid s_t)}$$

Advantage: $$A_{\pi_{\theta_{old}}}(s_t, a_t) = Q_{\pi_{\theta_{old}}}(s, a) - V_{\pi_{\theta_{old}}}(s)$$

$$\mathcal{J}^{\mathrm{PPO}}(\theta) = A_{\pi_{\theta_{old}}}(s_t, a_t) \cdot \begin{cases} \min(r_t(\theta), 1 + \epsilon) & \text{if } A_{\pi_{\theta_{old}}}(s_t, a_t) > 0 \\ \max(r_t(\theta), 1 - \epsilon) & \text{if } A_{\pi_{\theta_{old}}}(s_t, a_t) < 0 \end{cases}$$

Slide credit: Erik Wijmans

Proximal Policy Optimization (PPO)

$$\mathcal{J}^{\mathrm{PPO}}(\theta) = A_{\pi_{\theta_{old}}}(s_t, a_t) \cdot \begin{cases} \min(r_t(\theta), 1+\epsilon) & \text{if } A_{\pi_{\theta_{old}}}(s_t, a_t) > 0 \\ \max(r_t(\theta), 1-\epsilon) & \text{if } A_{\pi_{\theta_{old}}}(s_t, a_t) < 0 \end{cases}$$

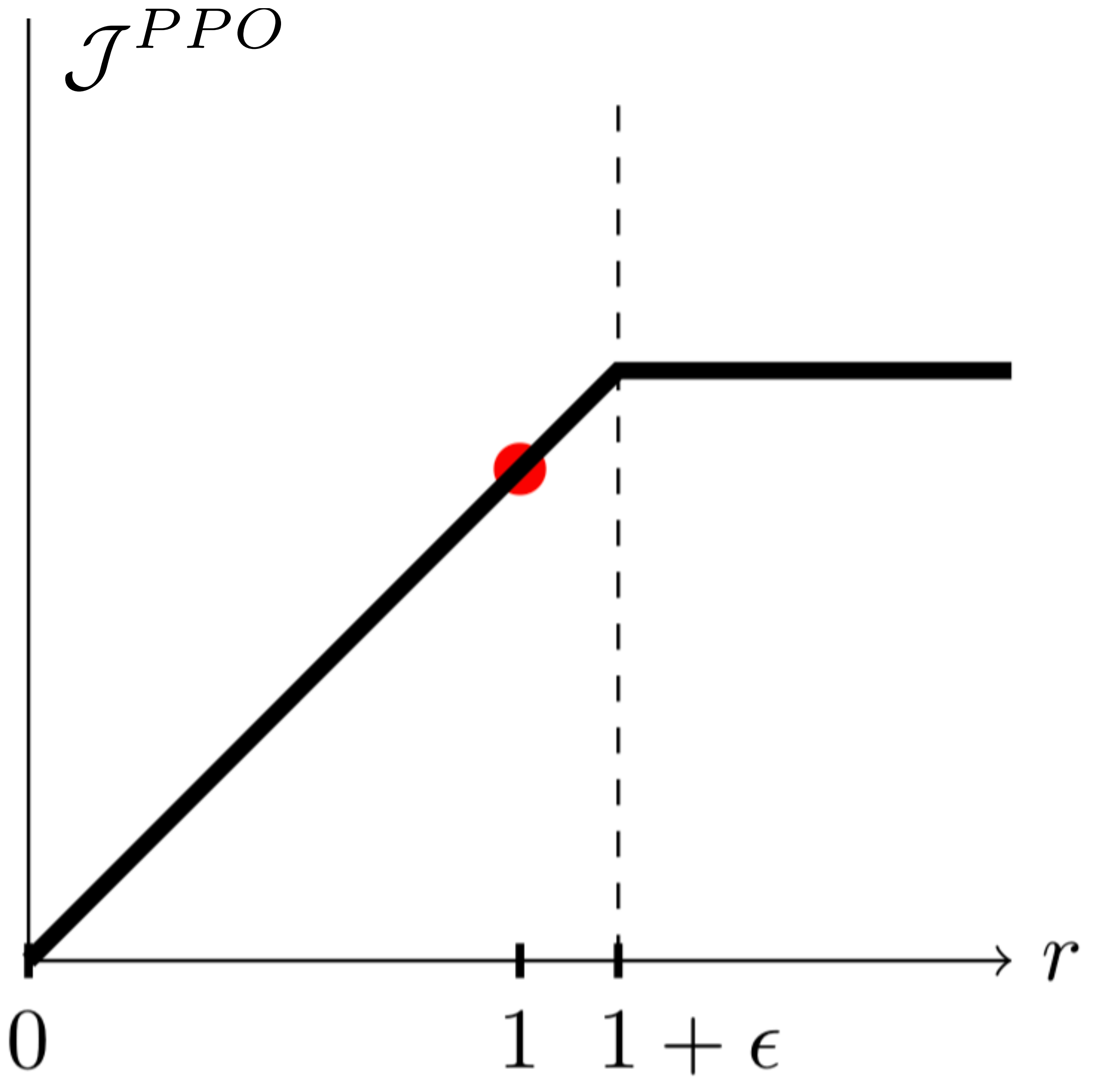
Slide credit: Erik Wijmans

Proximal Policy Optimization (PPO)

$$\mathcal{J}^{\text{PPO}}(\theta) = A_{\pi_{\theta_{old}}}(s_t, a_t) \cdot \begin{cases} \min(r_t(\theta), 1+\epsilon) & \text{if } A_{\pi_{\theta_{old}}}(s_t, a_t) > 0 \\ \max(r_t(\theta), 1-\epsilon) & \text{if } A_{\pi_{\theta_{old}}}(s_t, a_t) < 0 \end{cases}$$

$A_{\pi_{\theta_{old}}}(s_t, a_t) > 0$

Actions led to better than expected return

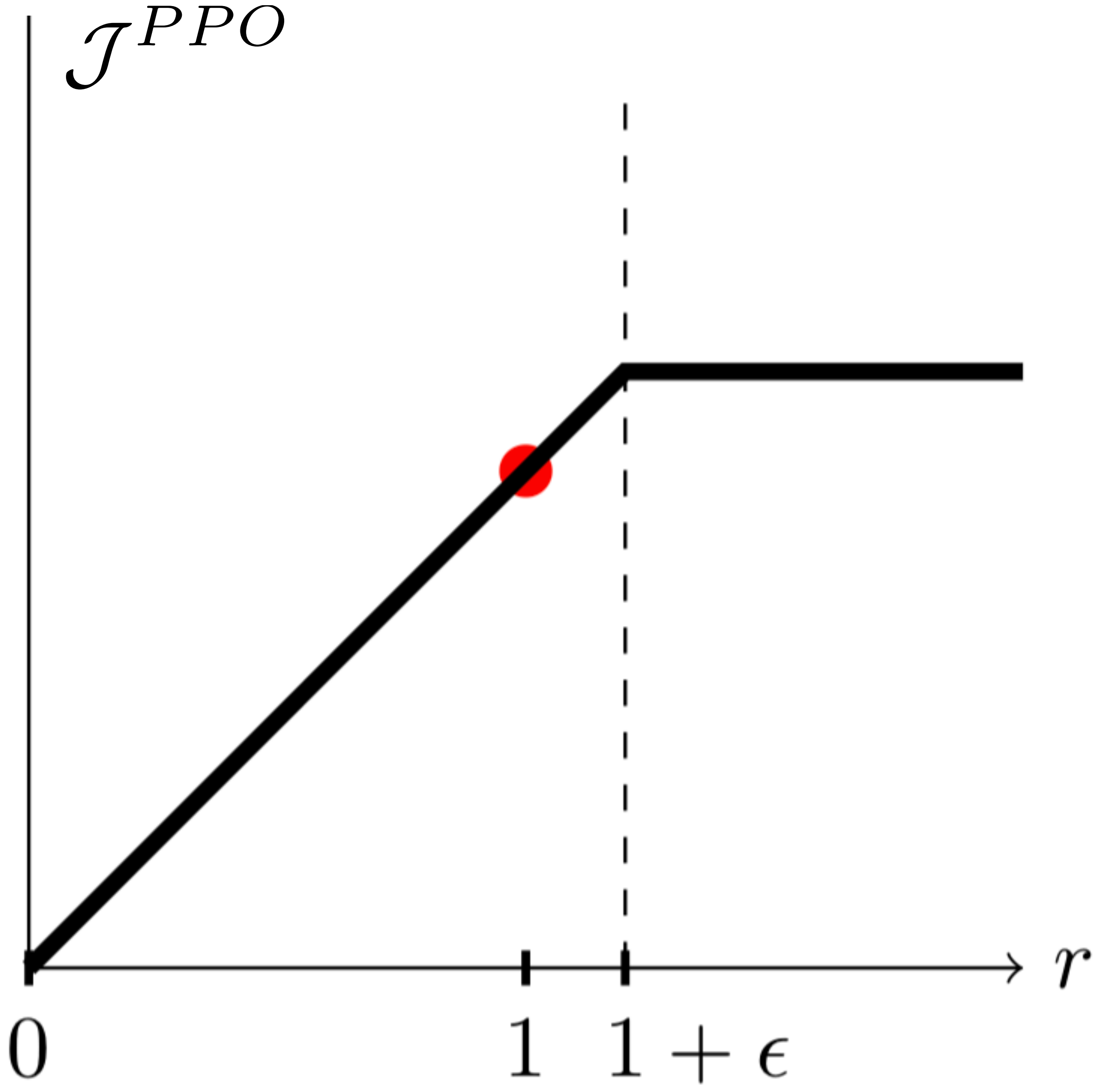


Proximal Policy Optimization (PPO)

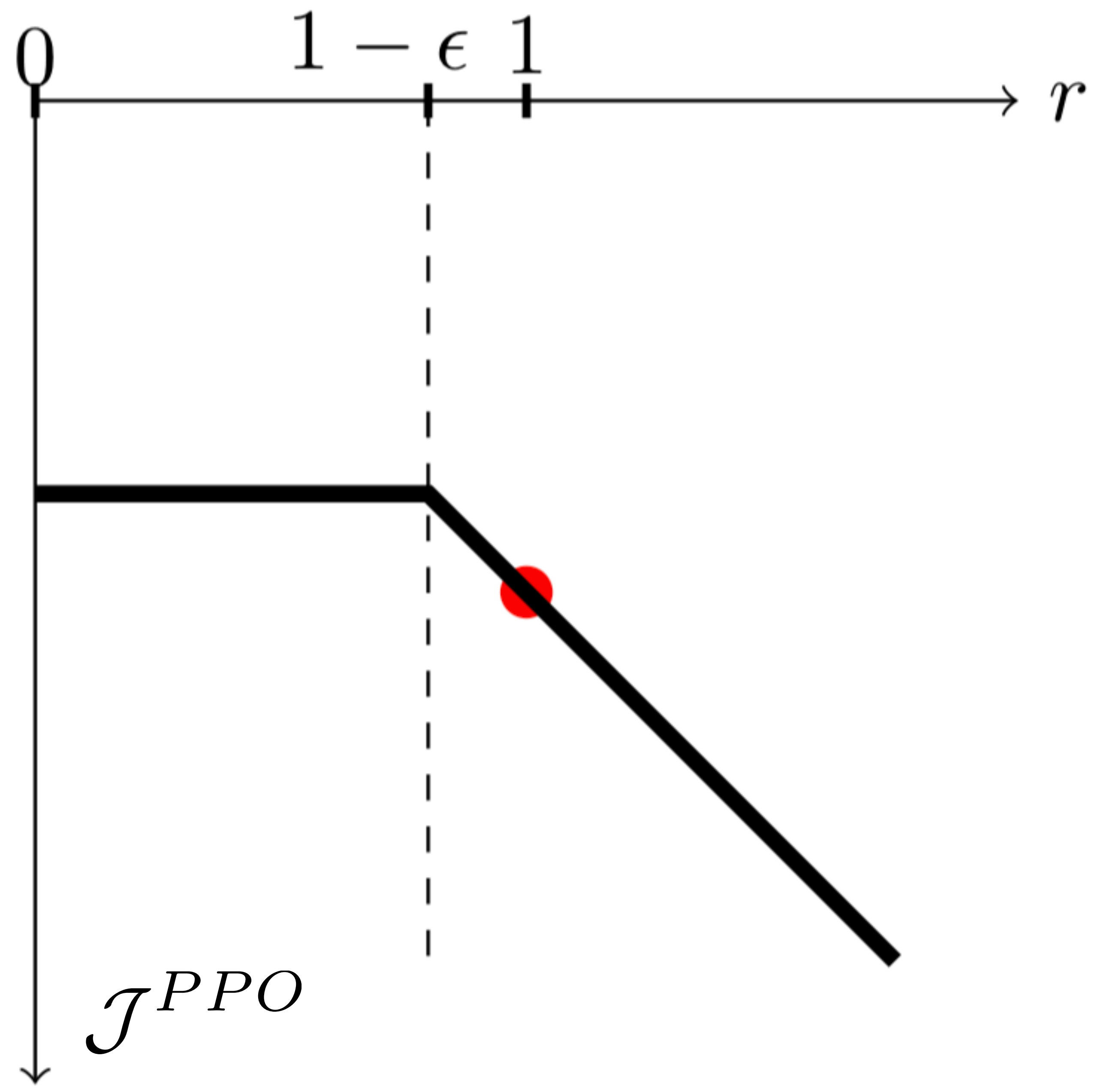
$$\mathcal{J}^{\mathrm{PPO}}(\theta) = A_{\pi_{\theta_{old}}}(s_t, a_t) \cdot \begin{cases} \min(r_t(\theta), 1+\epsilon) & \text{if } A_{\pi_{\theta_{old}}}(s_t, a_t) > 0 \\ \max(r_t(\theta), 1-\epsilon) & \text{if } A_{\pi_{\theta_{old}}}(s_t, a_t) < 0 \end{cases}$$

$A_{\pi_{\theta_{old}}}(s_t, a_t) > 0$

Actions led to better than expected return



$A_{\pi_{\theta_{old}}}(s_t, a_t) < 0$



Actions led to worse than expected return

Proximal Policy Optimization (PPO)

- Advantage
  - Able to perform multiple optimization steps per rollout
  - epsilon=0.2 “just works” in a lot of cases
  - Easily handles networks with hundreds of millions of parameters

Slide credit: Erik Wijmans

Proximal Policy Optimization (PPO)

- Advantage
  - Able to perform multiple optimization steps per rollout
  - epsilon=0.2 “just works” in a lot of cases
  - Easily handles networks with hundreds of millions of parameters
- Disadvantage
  - Other methods are more sample efficient

Slide credit: Erik Wijmans

PPO Implementation

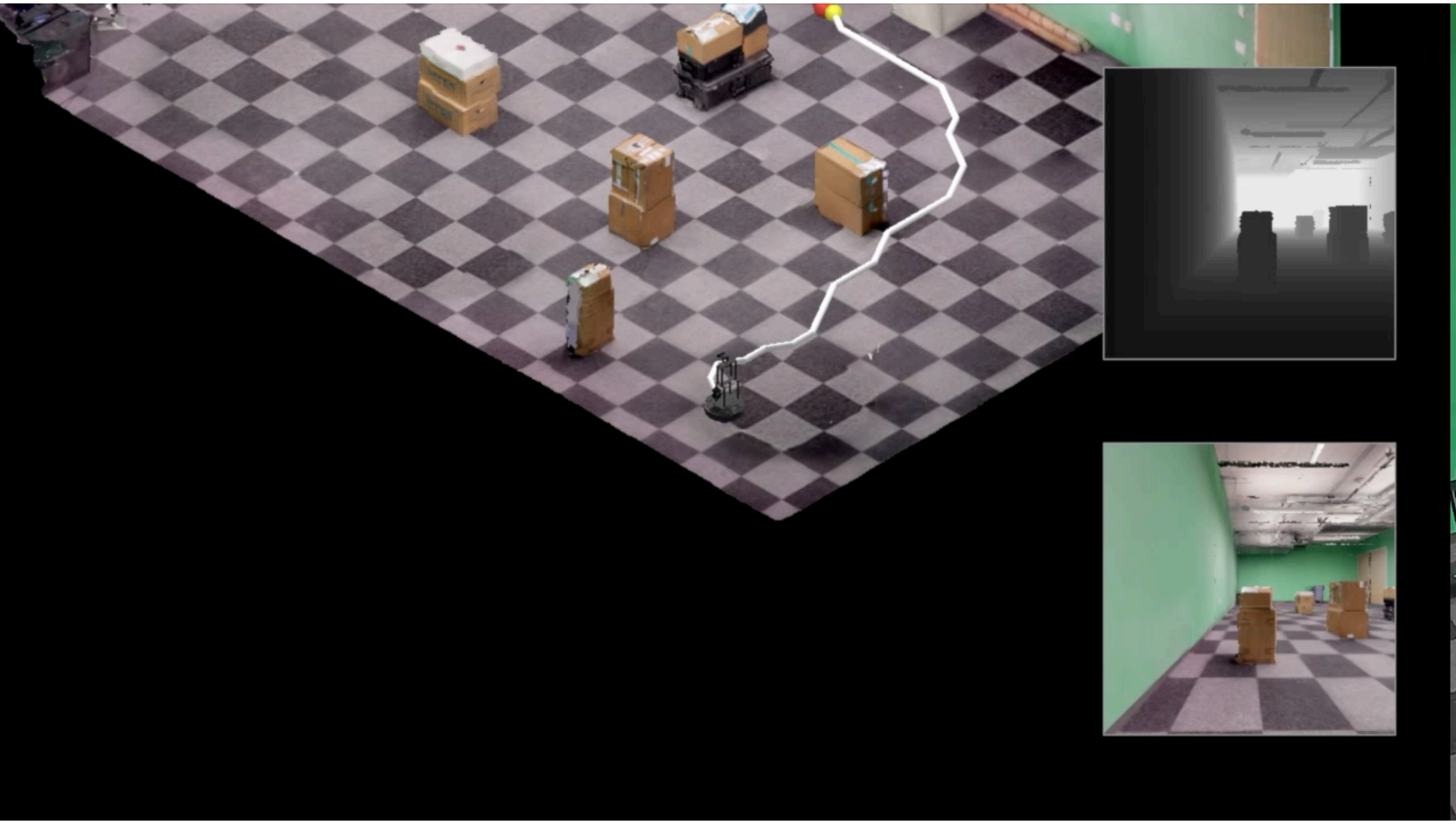
1. Collect a set of trajectories using current policy
2. For a few epochs (typically 2 or 4)
    1. Sample mini batches from rollout (typically 2 or 4)
        1. Update the policy via step of PPO/TRPO objective
3. Repeat

Slide credit: Erik Wijmans

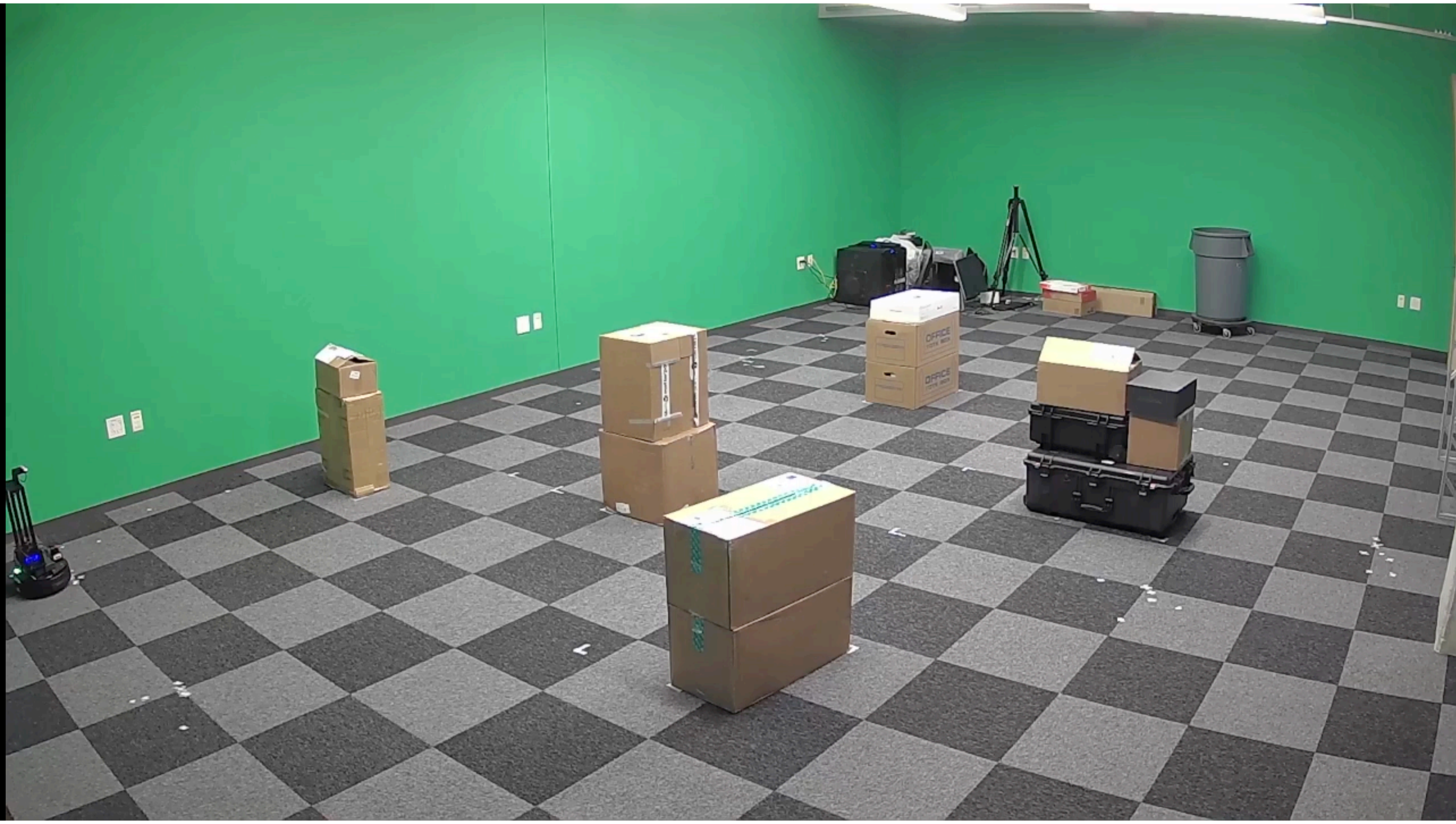
Outline

- Proximal Policy Optimization (PPO)
- Soft Actor Critic (SAC)
- Application: PointGoal Navigation
    - Sim2Real Transfer
    - Robot2Robot Transfer

Sim2Real Transfer



Simulation



Reality


Goal
20x

Outline

- Proximal Policy Optimization (PPO)
- Soft Actor Critic (SAC)
- Application: PointGoal Navigation
  - Sim2Real Transfer
  - Robot2Robot Transfer

Dynamics Aware Navigation

Dynamics Aware Navigation


A1


AlienGo


Daisy

How to generalize to new robots?

Laikago

Daisy

Sphere baseline (no dynamics)

Sphere baseline (no dynamics)

- Idealized agent: sphere
- Given direct access to points along the shortest path to the goal

Sphere baseline (no dynamics)

Zero shot transfer: Daisy —> AlienGo

Zero shot transfer: Daisy —> AlienGo

- Trained policy on Daisy robot, deploy on AlienGo

Zero shot transfer: Daisy —> AlienGo

Dynamics aware navigation

Dynamics aware navigation

Hierarchical Reinforcement Learning

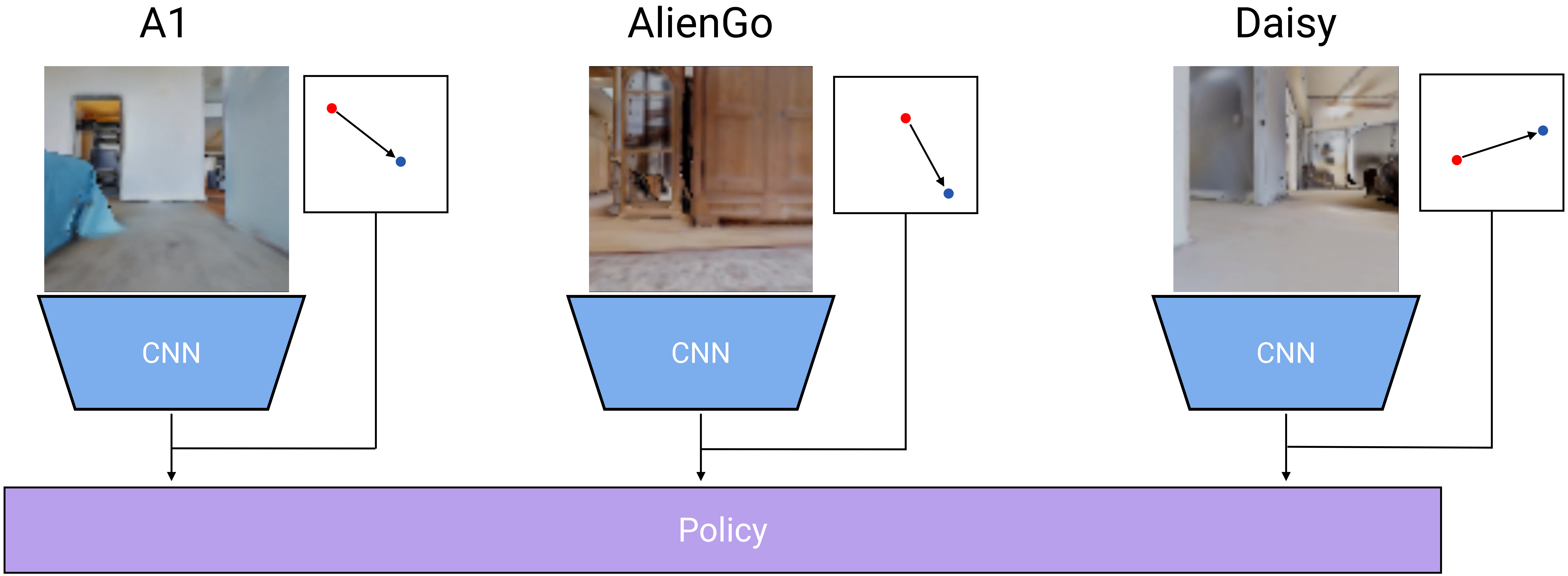
High level policy:
linear, angular velocity

Low-level controller

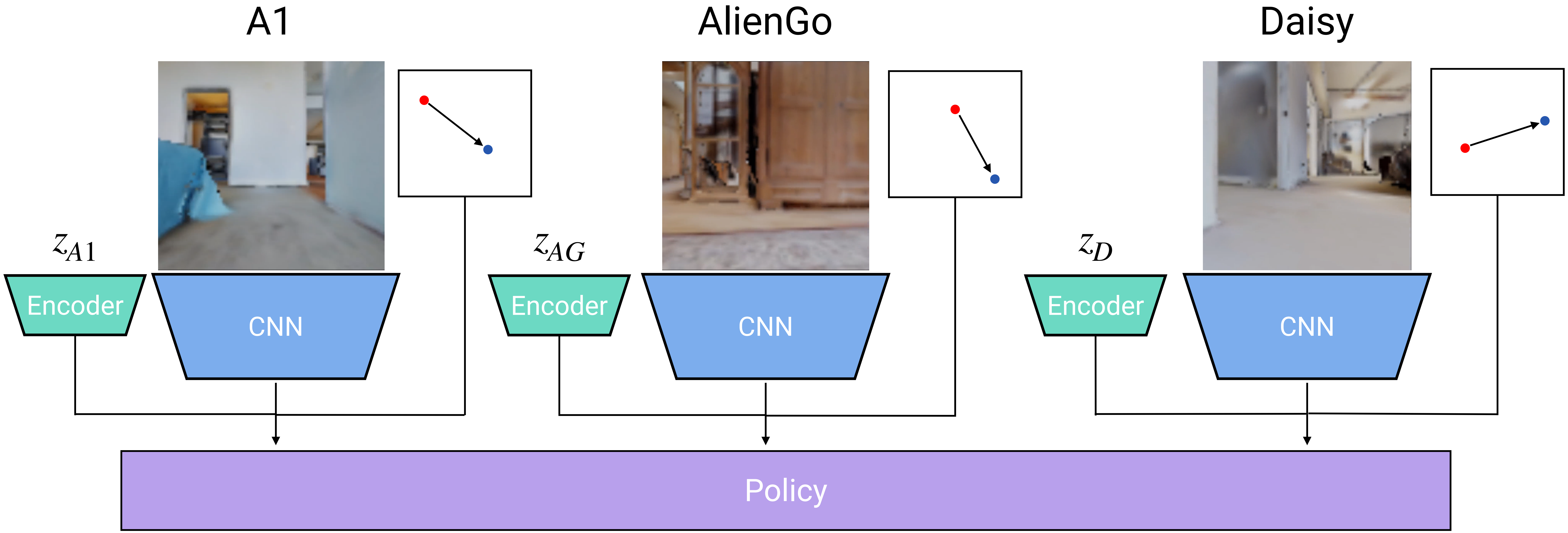
Low-level controller

Low-level controller

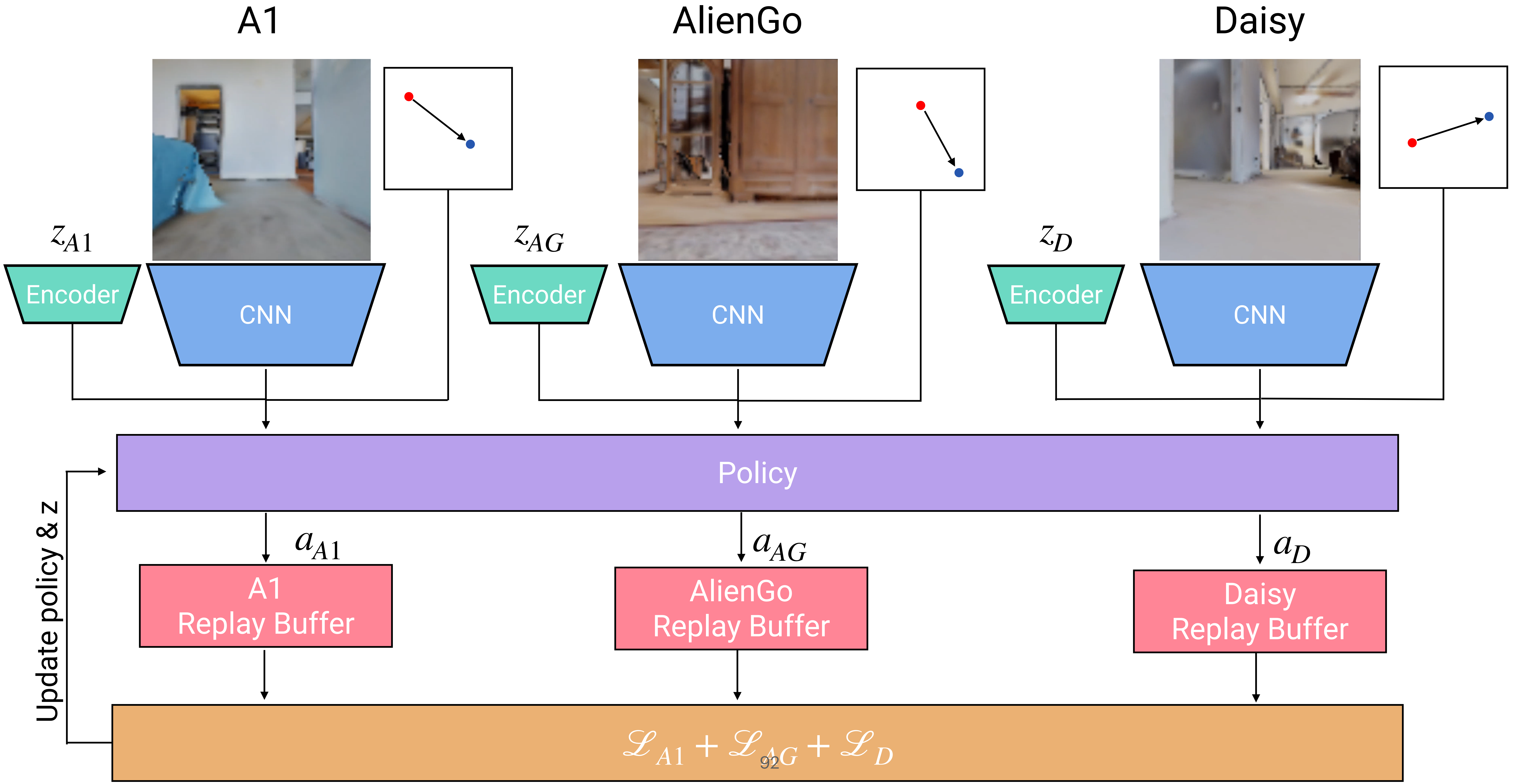
Our approach



Our approach



Our approach



A1 (Train robot) in novel environment

A1 (Train robot) in novel environment

AlienGo (Train robot) in novel environment

AlienGo (Train robot) in novel environment

Daisy (Train robot) in novel environment

Daisy (Train robot) in novel environment

Laikago (new robot) in novel environment

Laikago (new robot) in novel environment

Daisy-4Legged (new robot) in novel environment

Daisy-4Legged (new robot) in novel environment

Thank you!

Questions?