CS 4803 / 7643: Deep Learning

Topics: RL and Robotics

– Advantage Actor Critic (A2C)

– Trust Region Policy Optimization (TRPO)

– Proximal Policy Optimization (PPO)

– Application: Robotics + PointGoal Navigation

Joanne Truong
Georgia Tech

Who Am I?

Joanne Truong

3rd year Robotics PhD student
Advisors: Dhruv Batra and Sonia Chernova

Research Interests

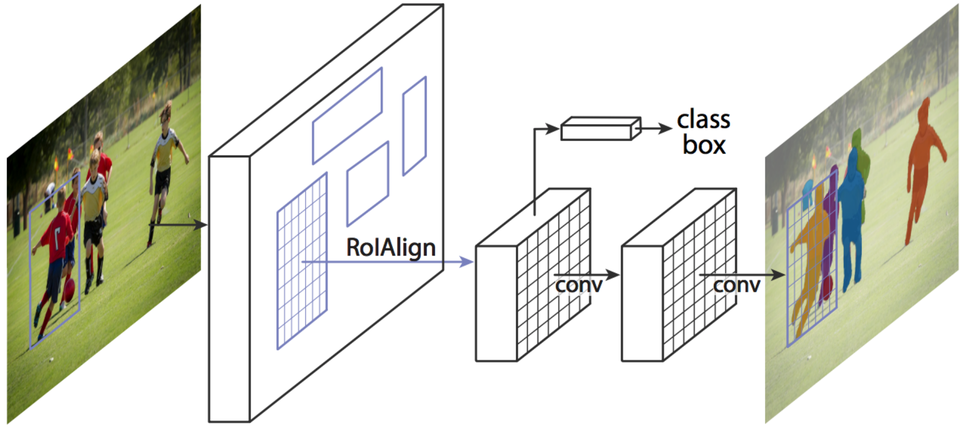
- Robotics
- Simulation to reality transfer

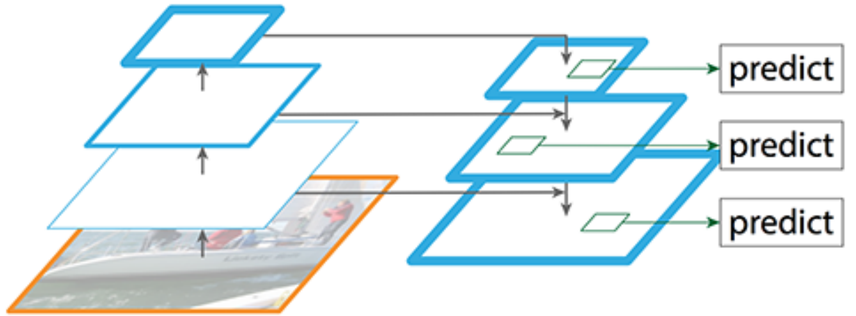
Lecture Plan

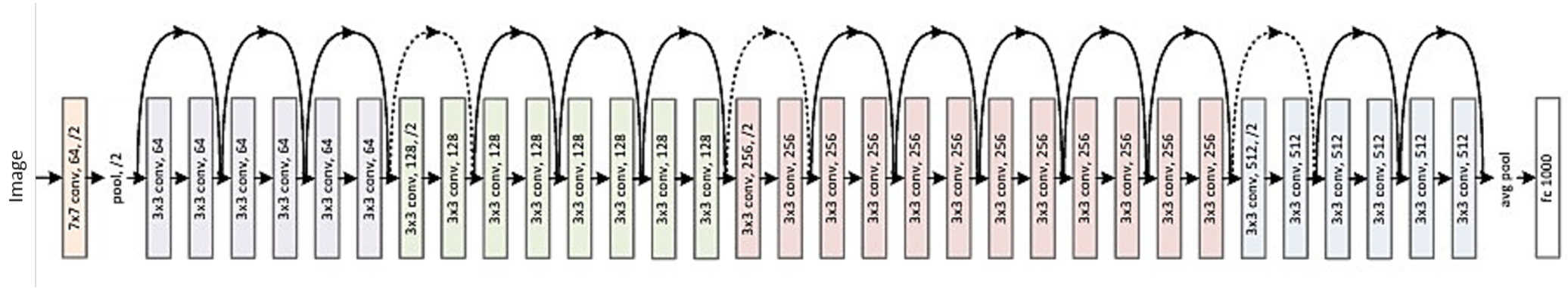
- Combine CNNs, RNNs (LSTMs), and RL together — all things you have learned about in this course — through a task called PointGoal Navigation
- Introduce more advanced RL — A2C, TRPO, PPO
- Show results using PPO on PointGoal Navigation on real robots!

Slide credit: Erik Wijmans

State-of-the-Art Visual Recognition



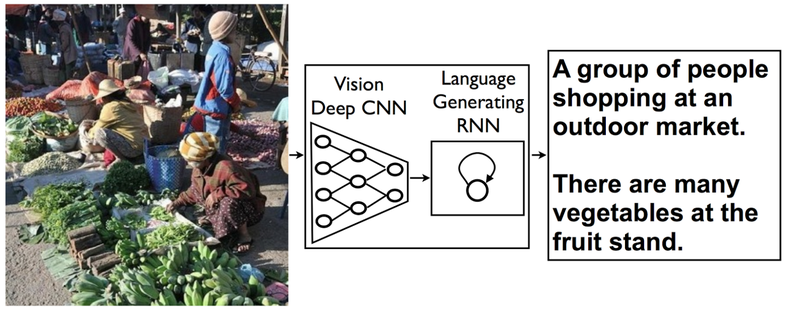


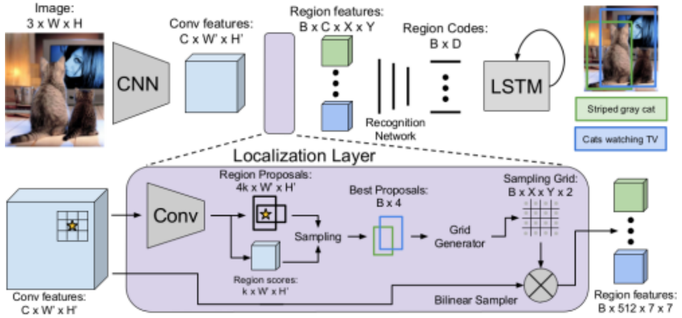


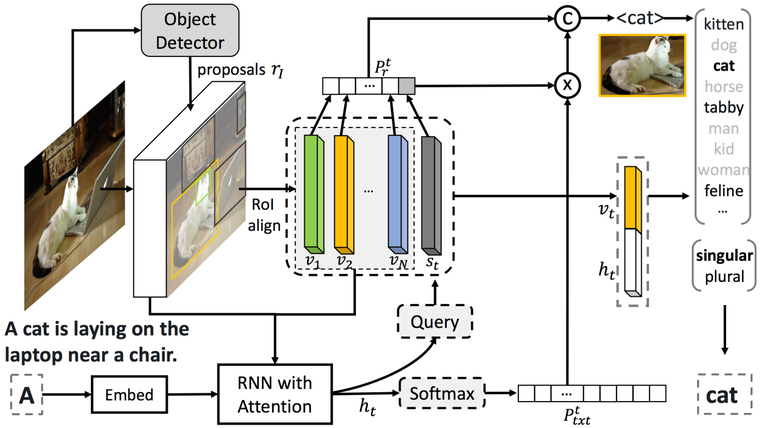
He et al., 2016a, b
He et al., 2017
Lin et al., 2017

Slide credit: Abhishek Das

State-of-the-Art Visual Recognition



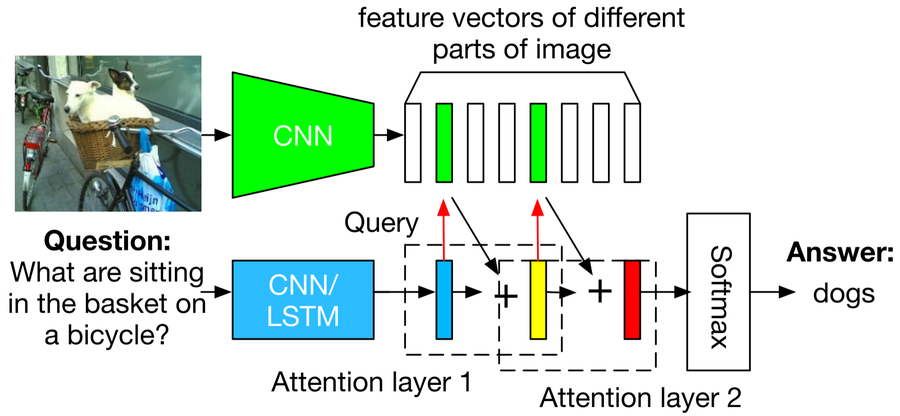


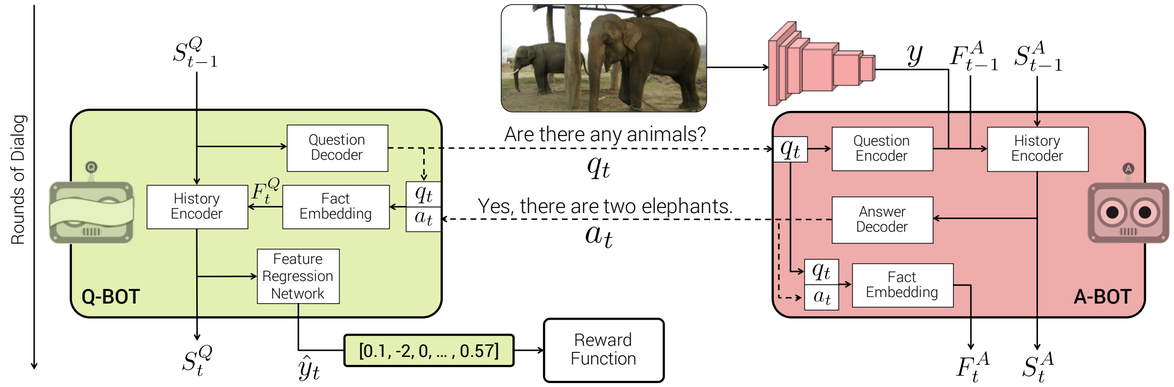


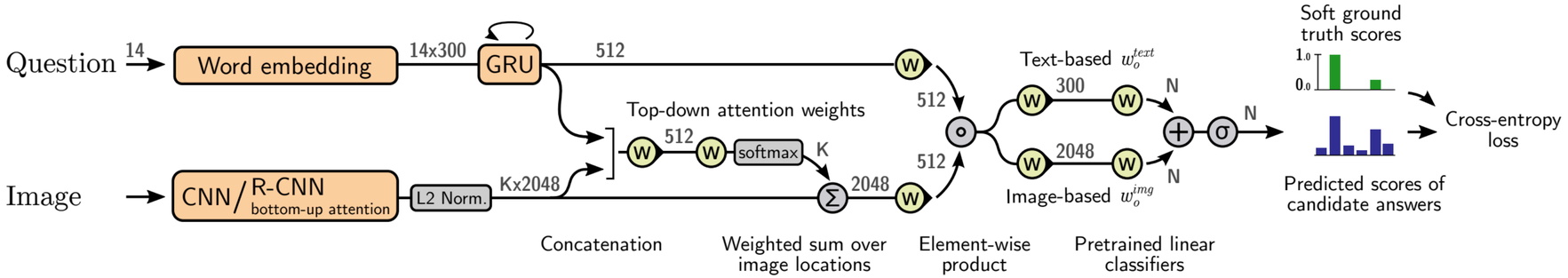
Vinyals et al., 2015
Karpathy and Johnson., 2016
Lu et al., 2018

Slide credit: Abhishek Das

State-of-the-Art Visual Recognition

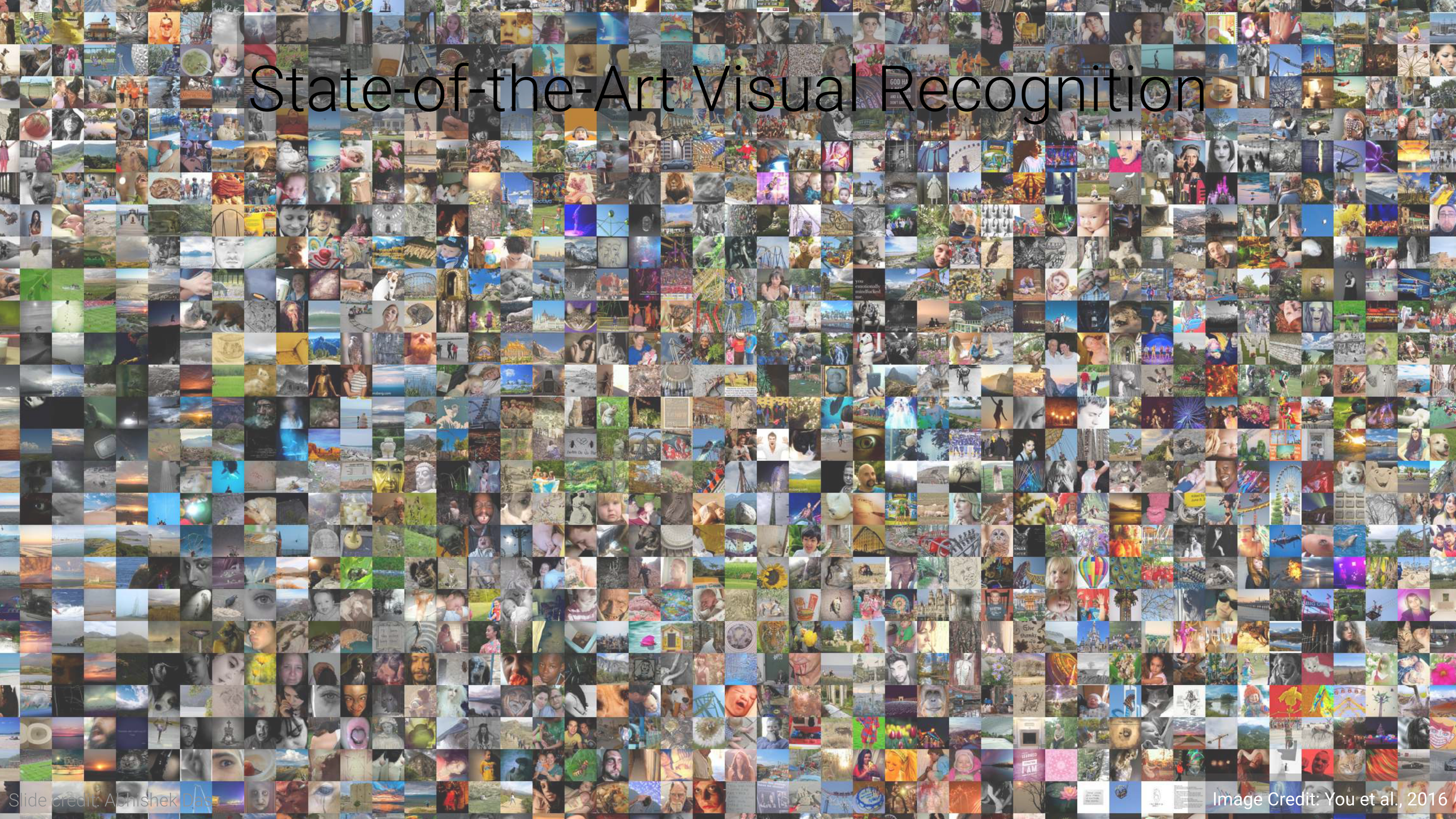






Yang et al., 2016
Das et al., 2017a, b
Anderson et al., 2016

State-of-the-Art Visual Recognition



Slide credit: Abhishek Das

Image Credit: You et al., 2016

Applications

Slide credit: Abhishek Das

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Applications

Physical agent

Image Credit: Lockheed Martin; DARPA Robotics Challenge

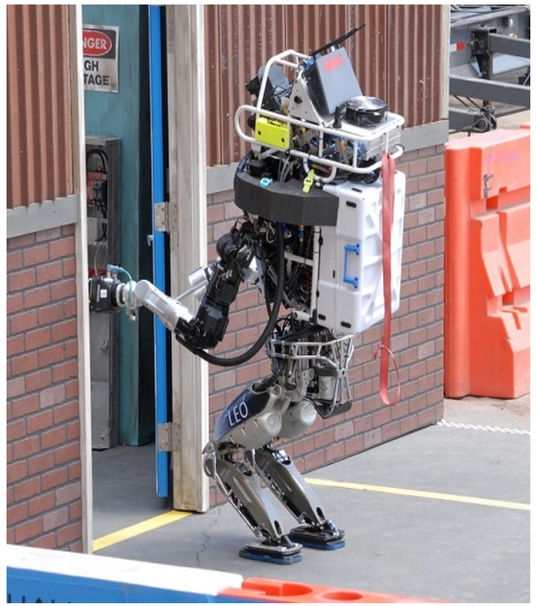
Applications

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 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

...

Slide credit: Abhishek Das

Image Credit: Lockheed Martin; DARPA Robotics Challenge

Challenges

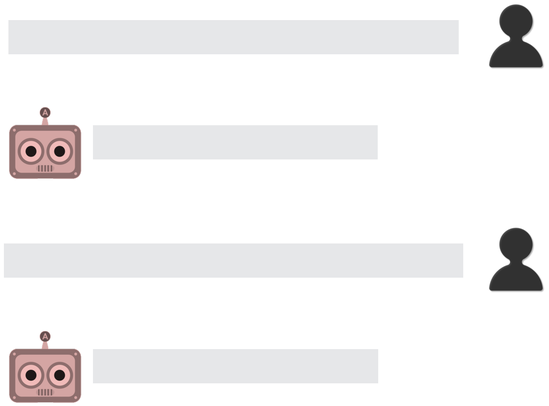
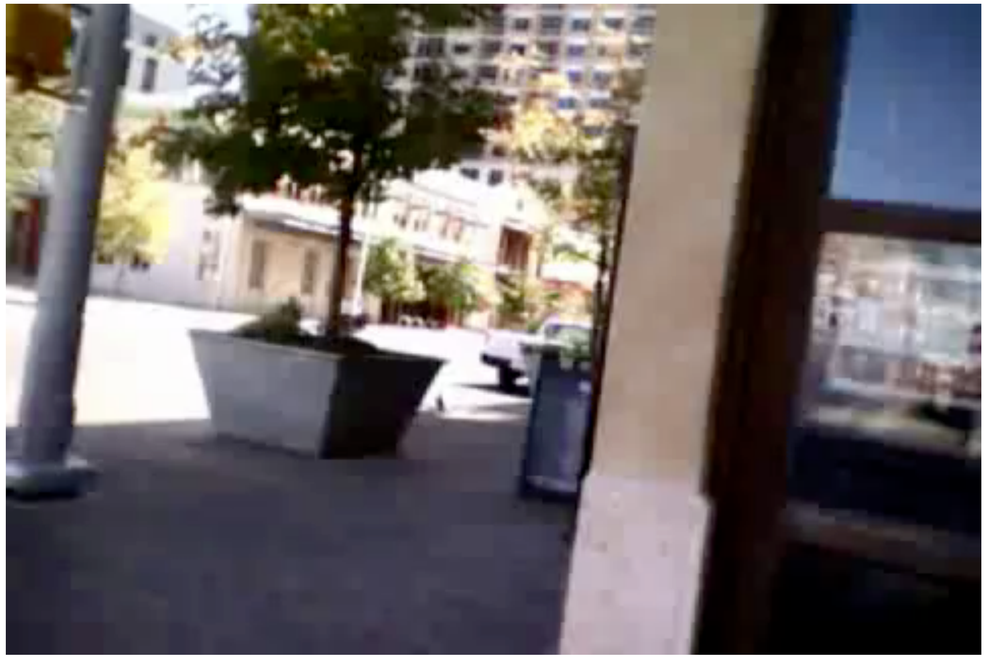


Image Credit: Image-Net
Image Credit: Lockheed Martin; DARPA Robotics Challenge

Slide credit: Abhishek Das

Challenges

Egocentric vision



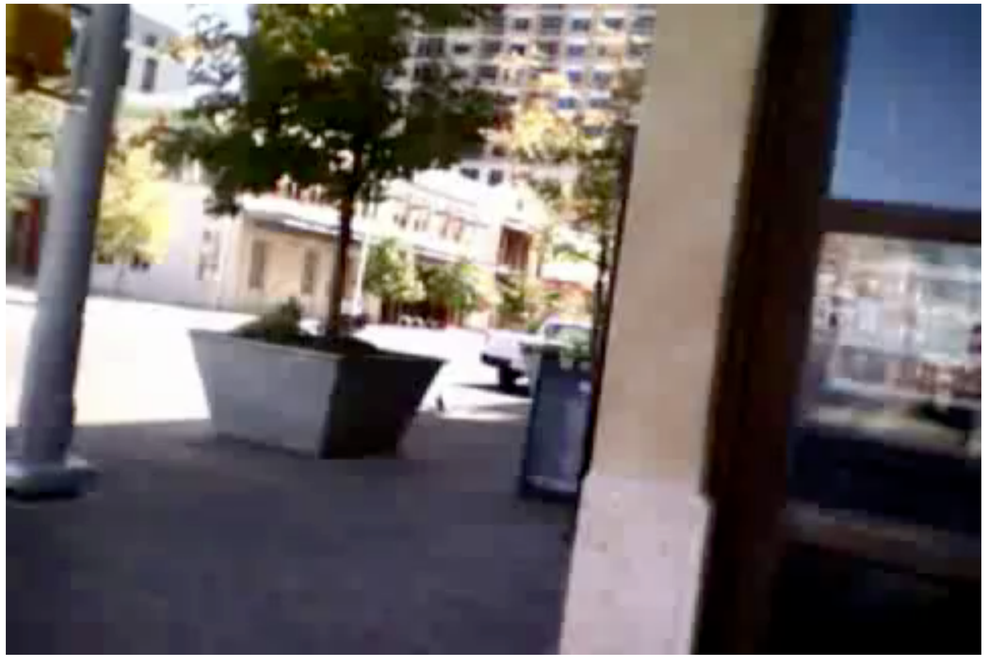
No access to well-composed, curated images

Slide credit: Abhishek Das

Image Credit: Image-Net, Video Credit: Lee et al., 2012

Challenges

Egocentric vision

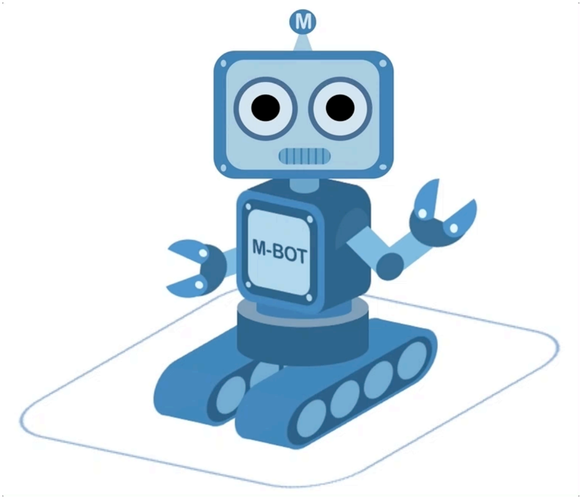


Video Credit: Lee et al., 2012

Challenges

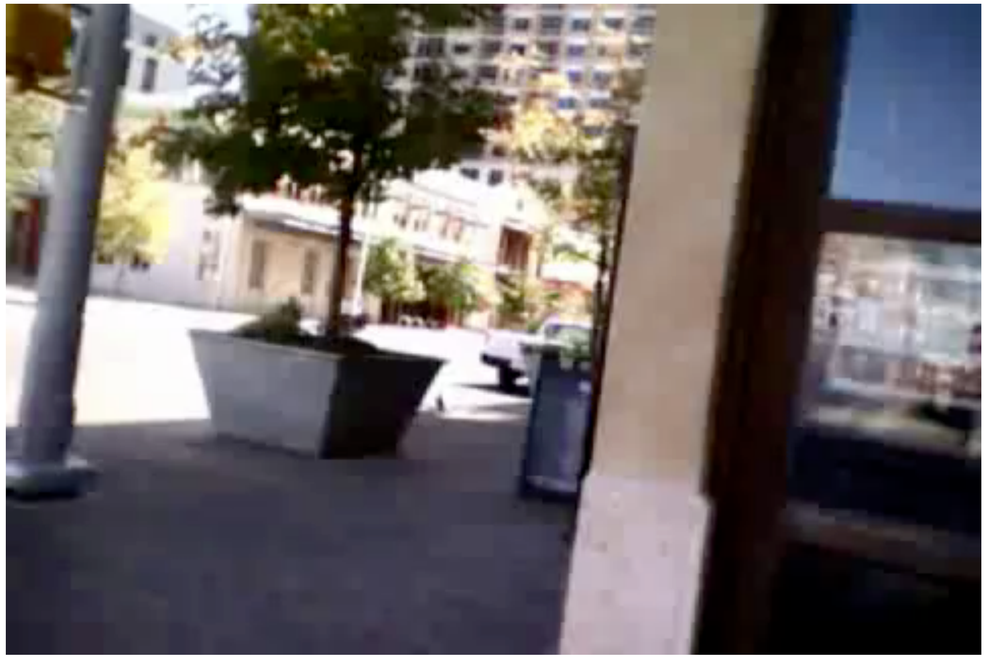
Egocentric vision

Active perception



Action

Observation



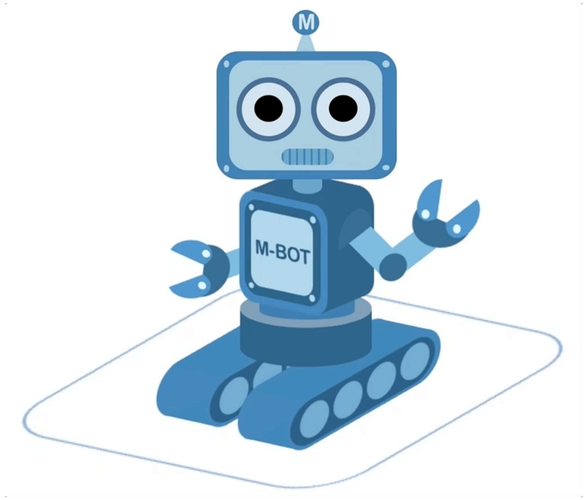
Agent controls incoming data distribution

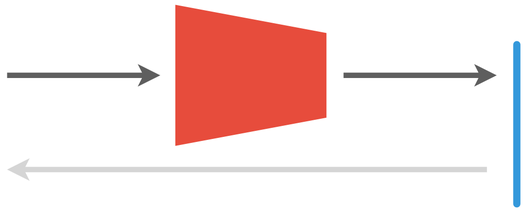
Slide credit: Abhishek Das

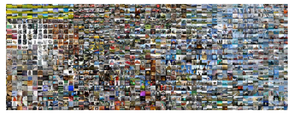
Video Credit: Lee et al., 2012

Challenges

Egocentric vision
Active perception
Sparse rewards





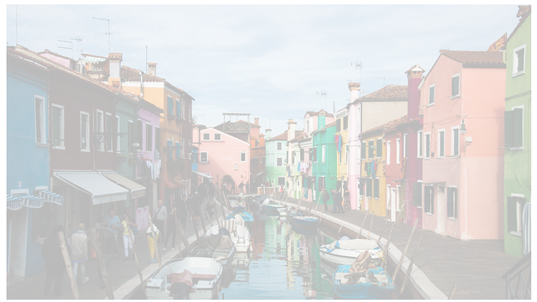


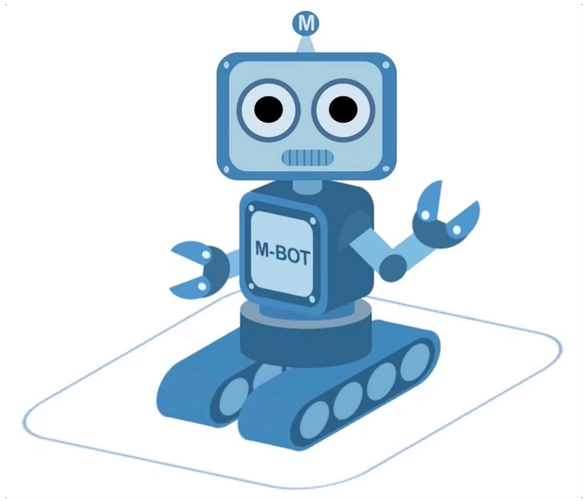
Slide credit: Abhishek Das

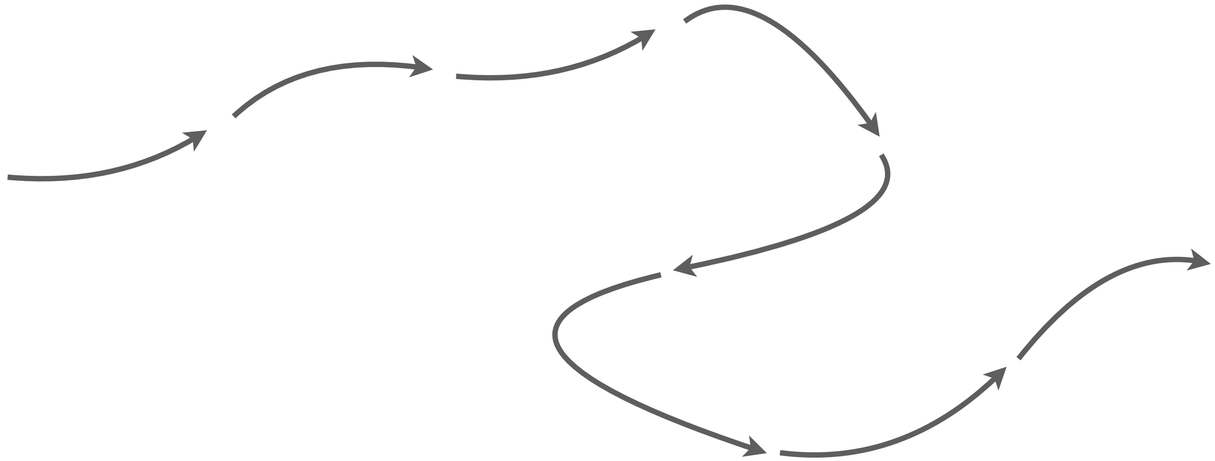
Image Credit: Image-Net

Challenges

Egocentric vision
Active perception
Sparse rewards





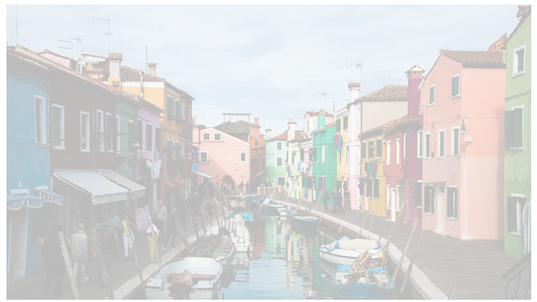


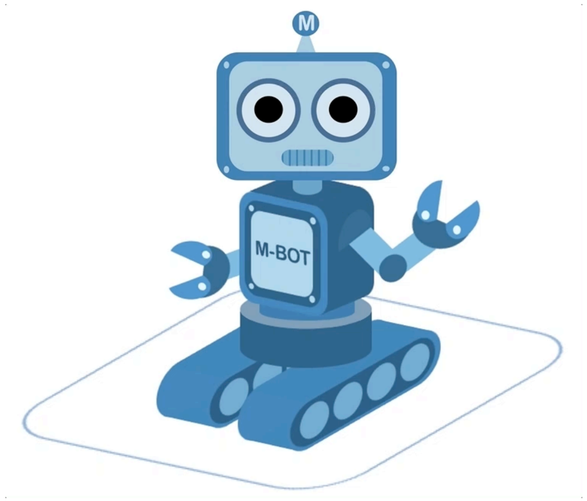
Slide credit: Abhishek Das

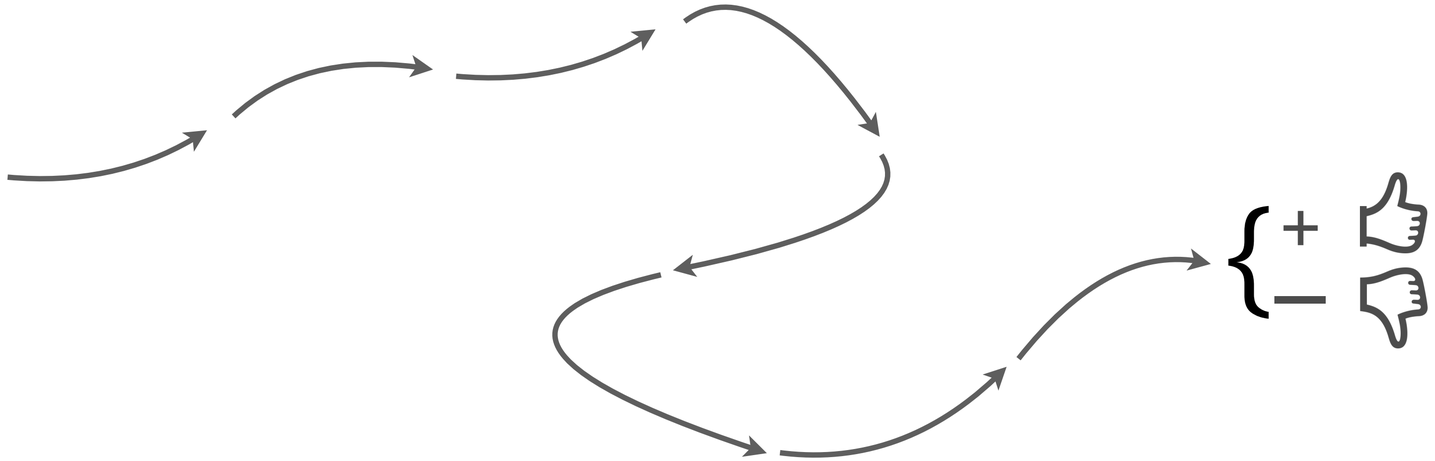
Image Credit: Image-Net

Challenges

Egocentric vision
Active perception
Sparse rewards





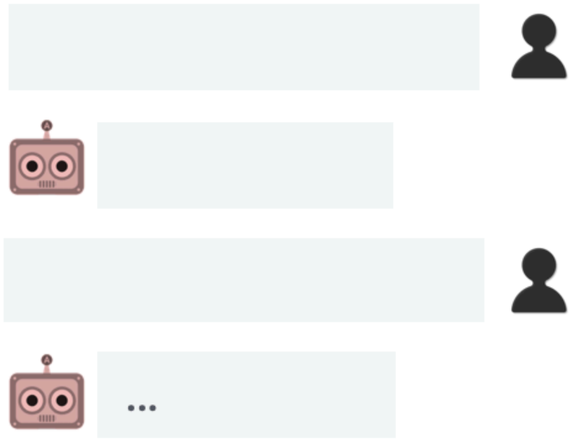


Challenges

Egocentric vision
Active perception
Sparse rewards
Language understanding

Slide credit: Abhishek Das

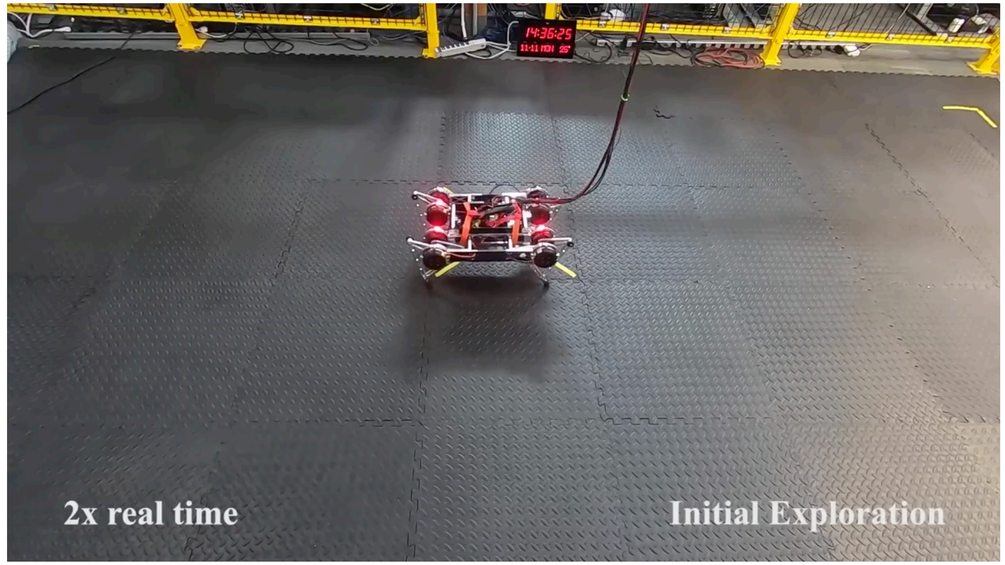
Internet AI → Embodied AI



Slide credit: Abhishek Das

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

Train directly in the real world?



- Slow
- Dangerous
- Resource intensive
- Not easily reproducible

OpenAI et al. 2019
Sun et al. 2021

Habitat-Sim

PointGoal Navigation

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

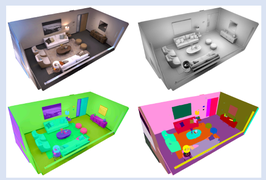
Berkeley UNIVERSITY OF CALIFORNIA 6

Standardizing the Embodied AI “software stack”

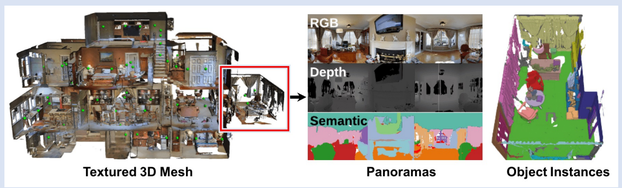
Slide credit: Erik Wijmans

Standardizing the Embodied AI “software stack”

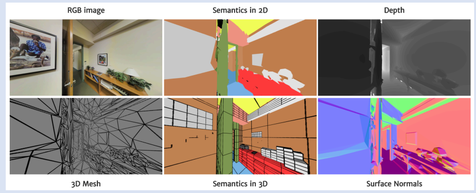
Datasets



Replica (Straub et al., 2019)



Matterport3D (Chang et al., 2017)

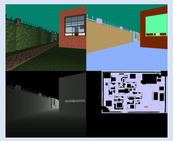


2D-3D-S (Armeni et al., 2017)

Slide credit: Erik Wijmans

Standardizing the Embodied AI “software stack”

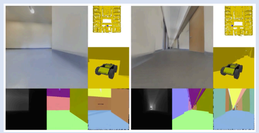
Simulators



House3D
(Wu et al., 2017)

AI2-THOR
(Kolve et al., 2017)

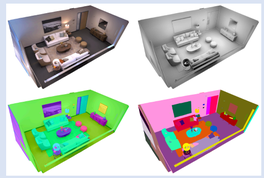
MINOS
(Savva et al., 2017)



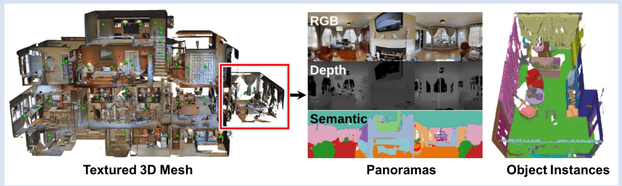
Gibson
(Zamir et al., 2018)

CHALET
(Yan et al., 2018)

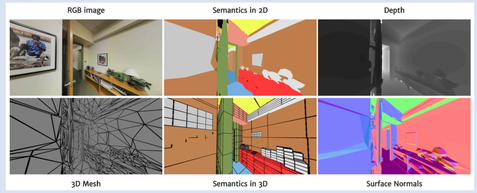
Datasets



Replica (Straub et al., 2019)



Matterport3D (Chang et al., 2017)

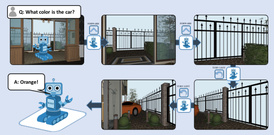


2D-3D-S (Armeni et al., 2017)

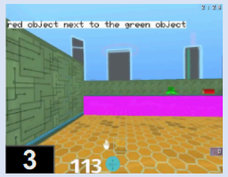
Slide credit: Erik Wijmans

Standardizing the Embodied AI “software stack”

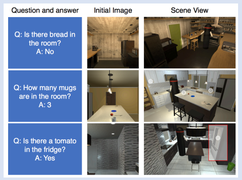
Tasks



EmbodiedQA
(Das et al., 2018)



Language grounding
(Hill et al., 2017)

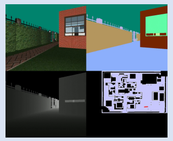


Interactive QA
(Gordon et al., 2018)

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

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

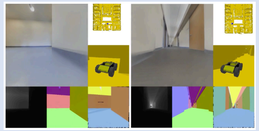
Simulators



House3D
(Wu et al., 2017)

AI2-THOR
(Kolve et al., 2017)

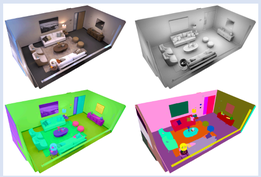
MINOS
(Savva et al., 2017)



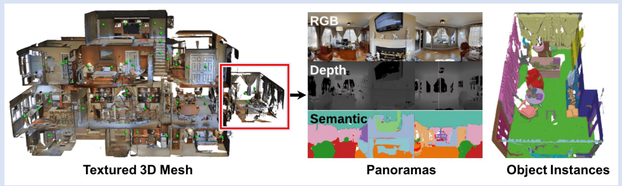
Gibson
(Zamir et al., 2018)

CHALET
(Yan et al., 2018)

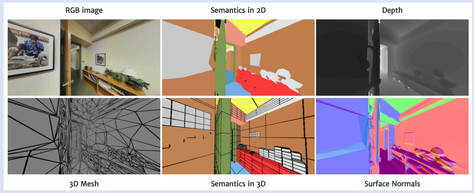
Datasets



Replica (Straub et al., 2019)



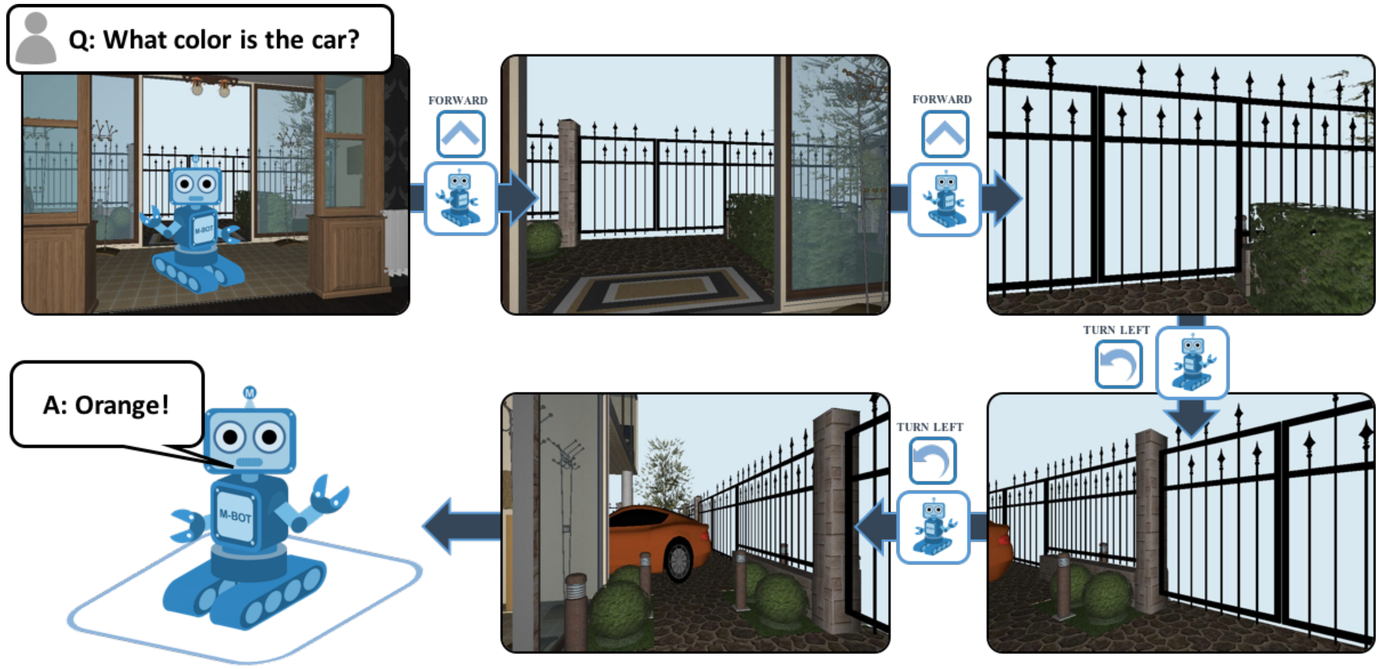
Matterport3D (Chang et al., 2017)



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

Slide credit: Erik Wijmans

Standardizing the Embodied AI “software stack”



EmbodiedQA
(Das et al., 2018)

Slide credit: Erik Wijmans

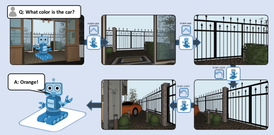
Standardizing the Embodied AI “software stack”

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

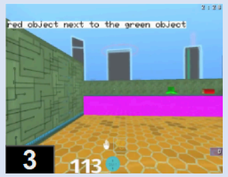
Slide credit: Erik Wijmans

Standardizing the Embodied AI “software stack”

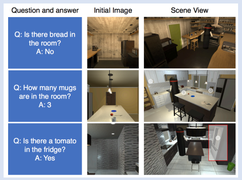
Tasks



EmbodiedQA
(Das et al., 2018)



Language grounding
(Hill et al., 2017)

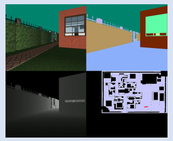


Interactive QA
(Gordon et al., 2018)

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

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

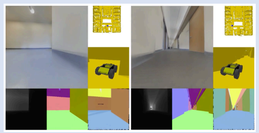
Simulators



House3D
(Wu et al., 2017)

AI2-THOR
(Kolve et al., 2017)

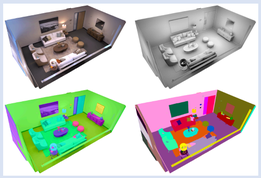
MINOS
(Savva et al., 2017)



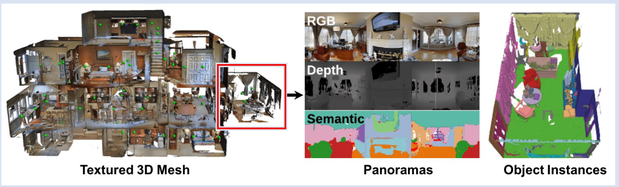
Gibson
(Zamir et al., 2018)

CHALET
(Yan et al., 2018)

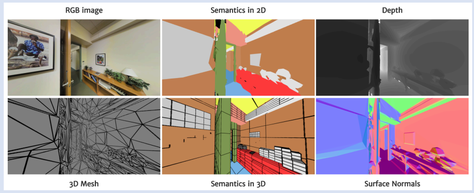
Datasets



Replica (Straub et al., 2019)



Matterport3D (Chang et al., 2017)



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

Habitat Platform

Habitat-Lab

Habitat-Sim

Generic Dataset Support

Slide credit: Erik Wijmans

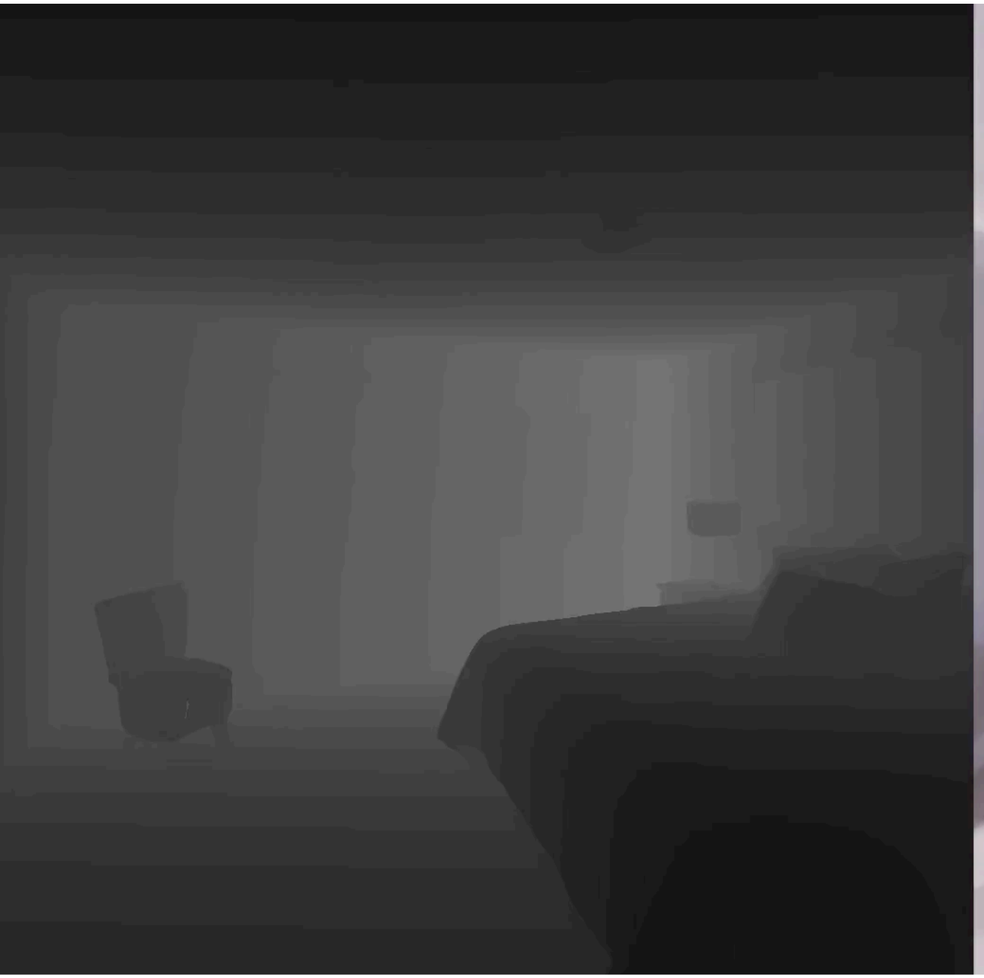
Habitat-Sim Demo

https://habitatonweb.cloudcv.org:8000/

Source: Ram Ramrakhya

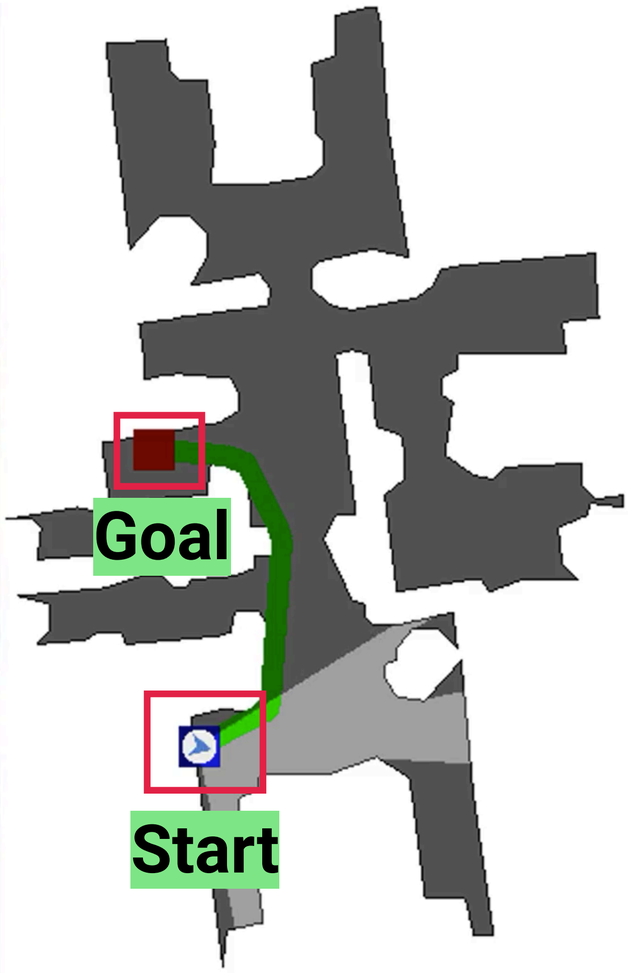
PointGoal Navigation

PointGoal Navigation



Depth

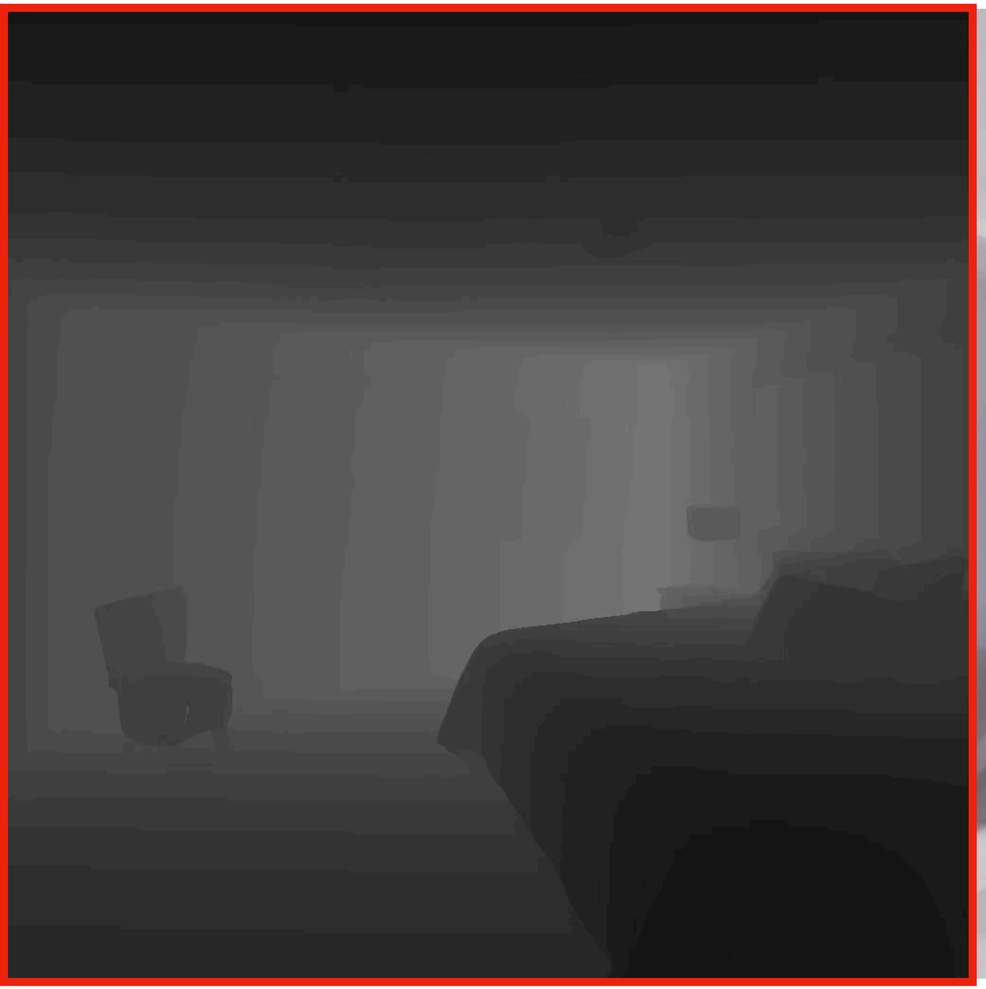
RGB and GPS+Compass



Top Down Map

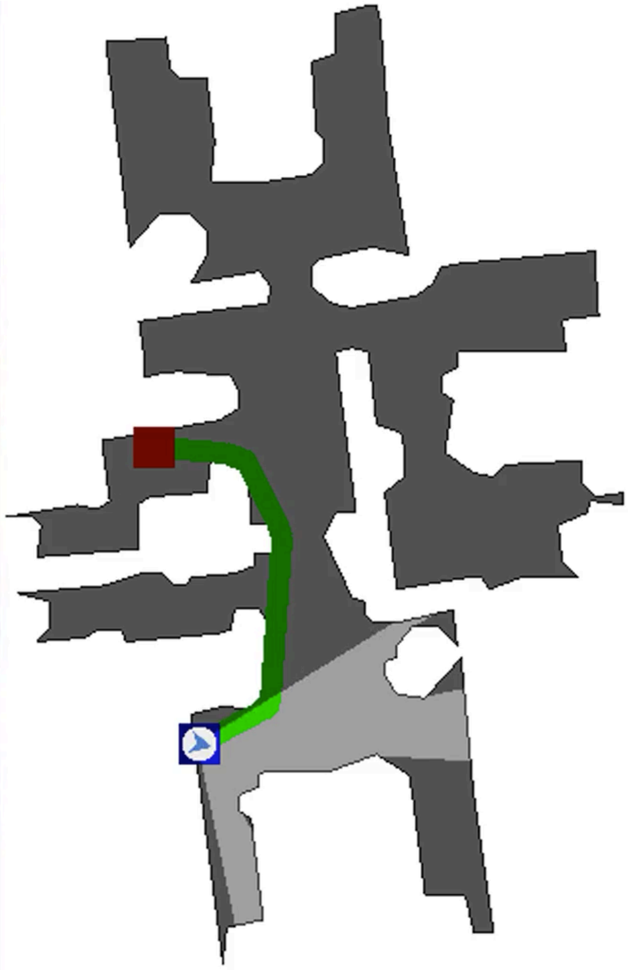
Slide credit: Erik Wijmans

PointGoal Navigation



Depth

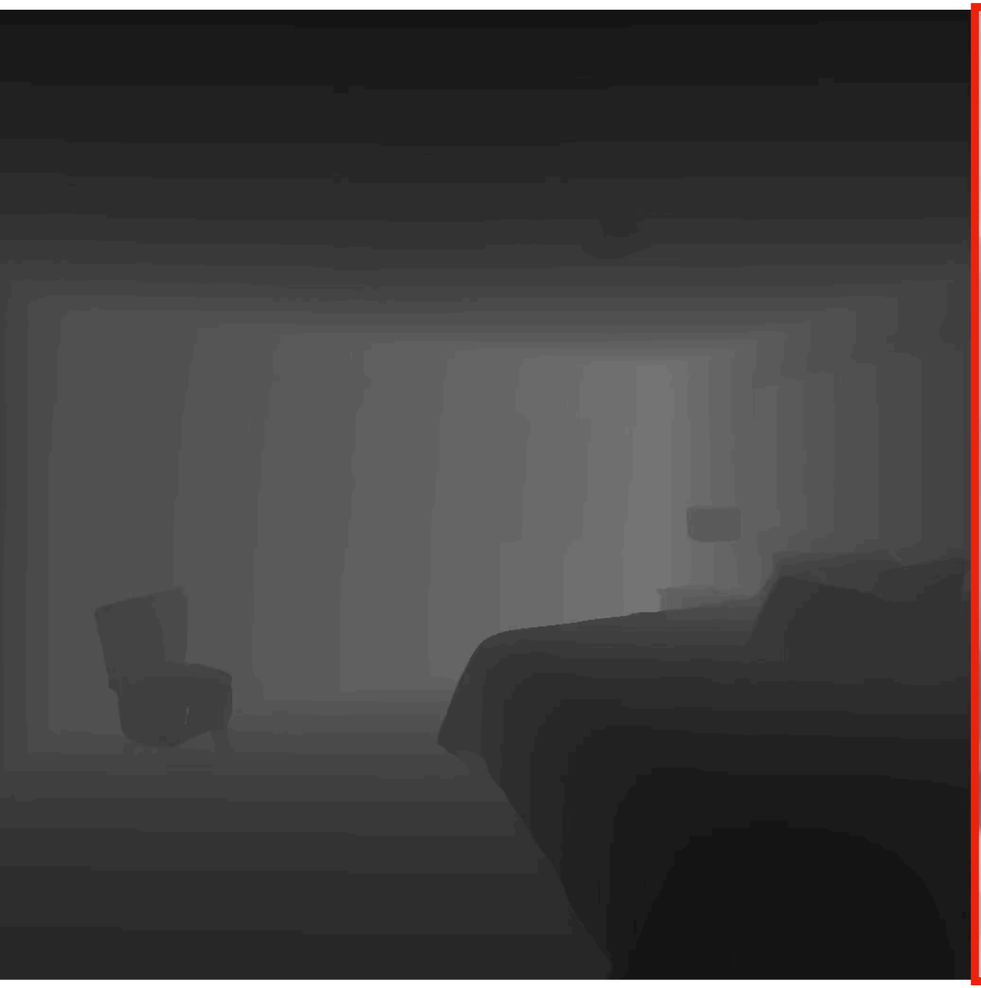
RGB and GPS+Compass



Top Down Map

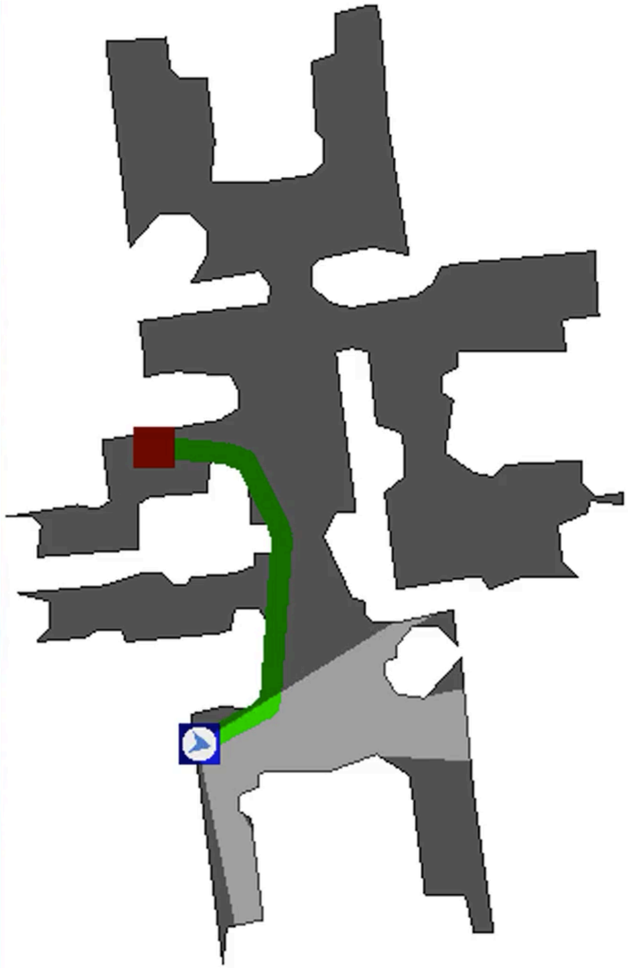
Slide credit: Erik Wijmans

PointGoal Navigation



Depth

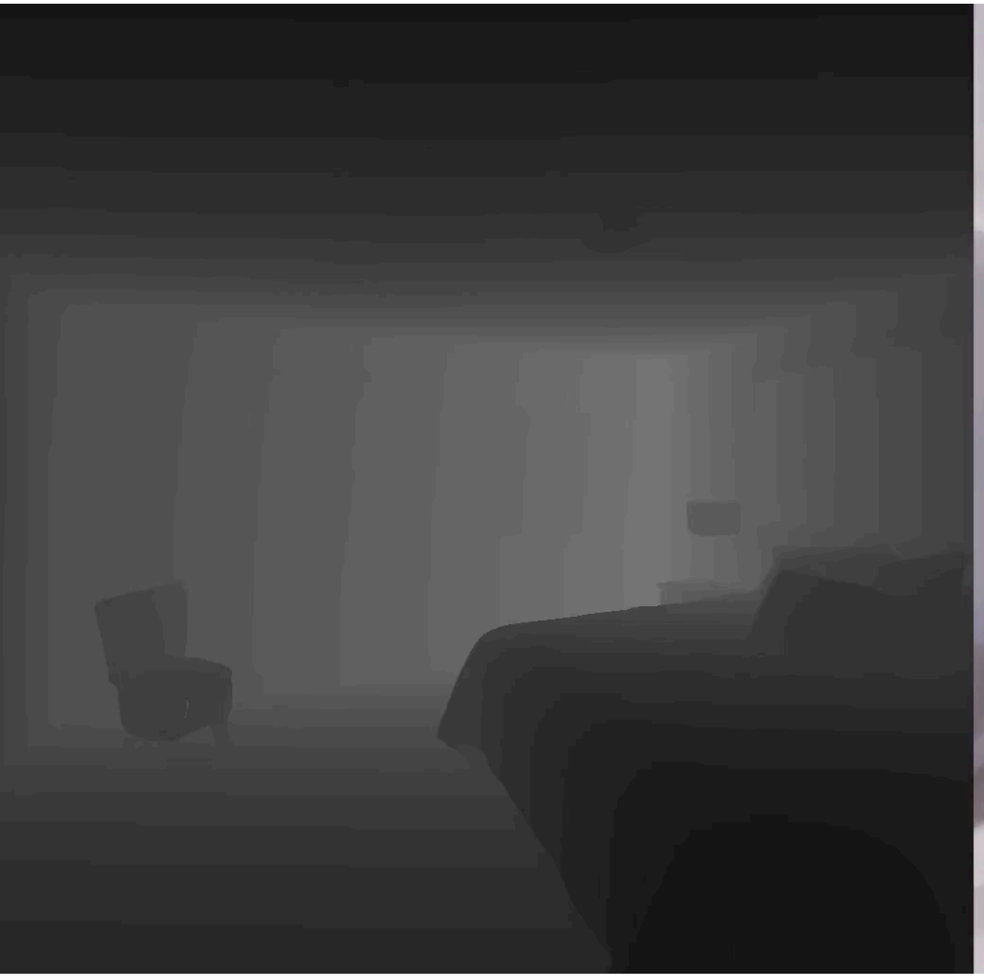
RGB and GPS+Compass



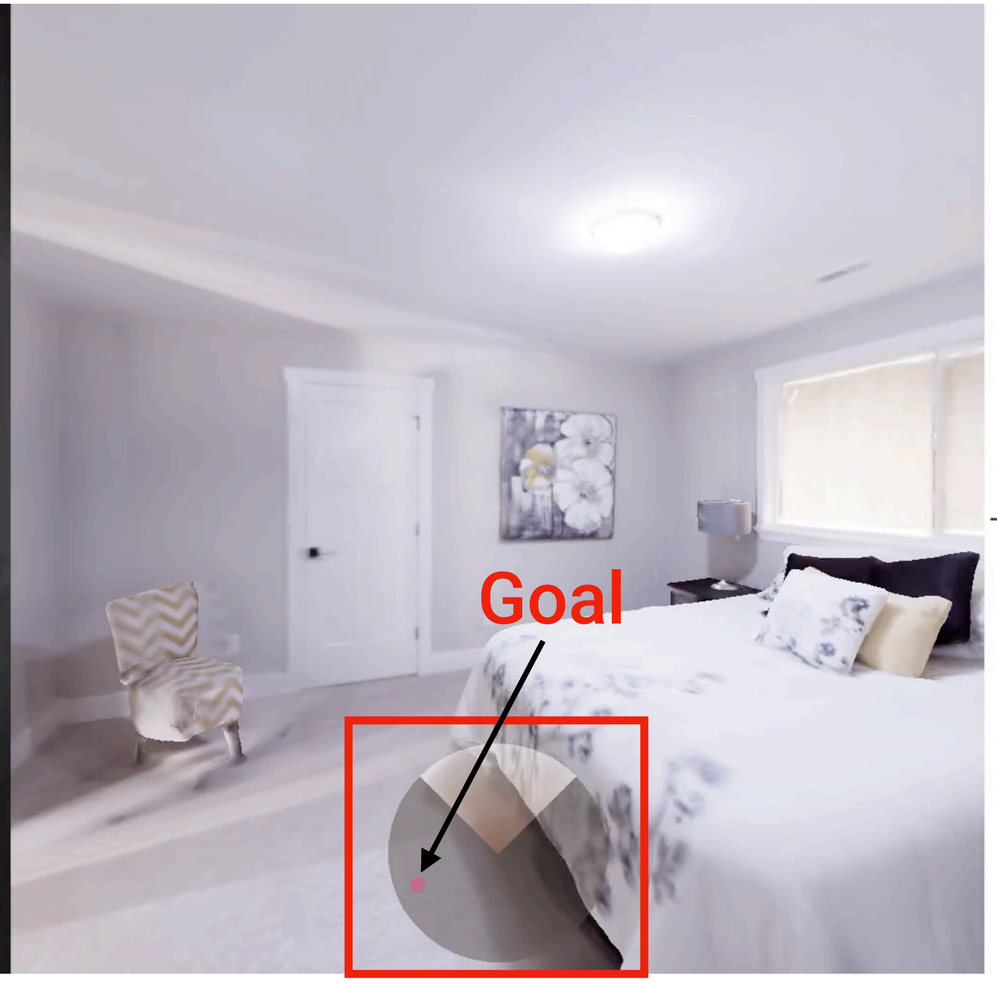
Top Down Map

Slide credit: Erik Wijmans

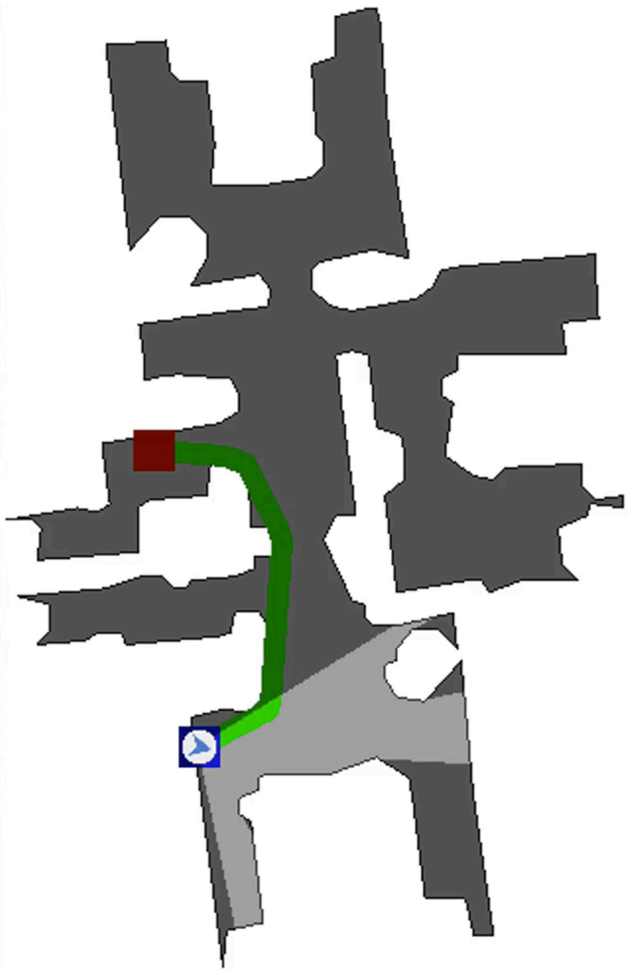
PointGoal Navigation



Depth



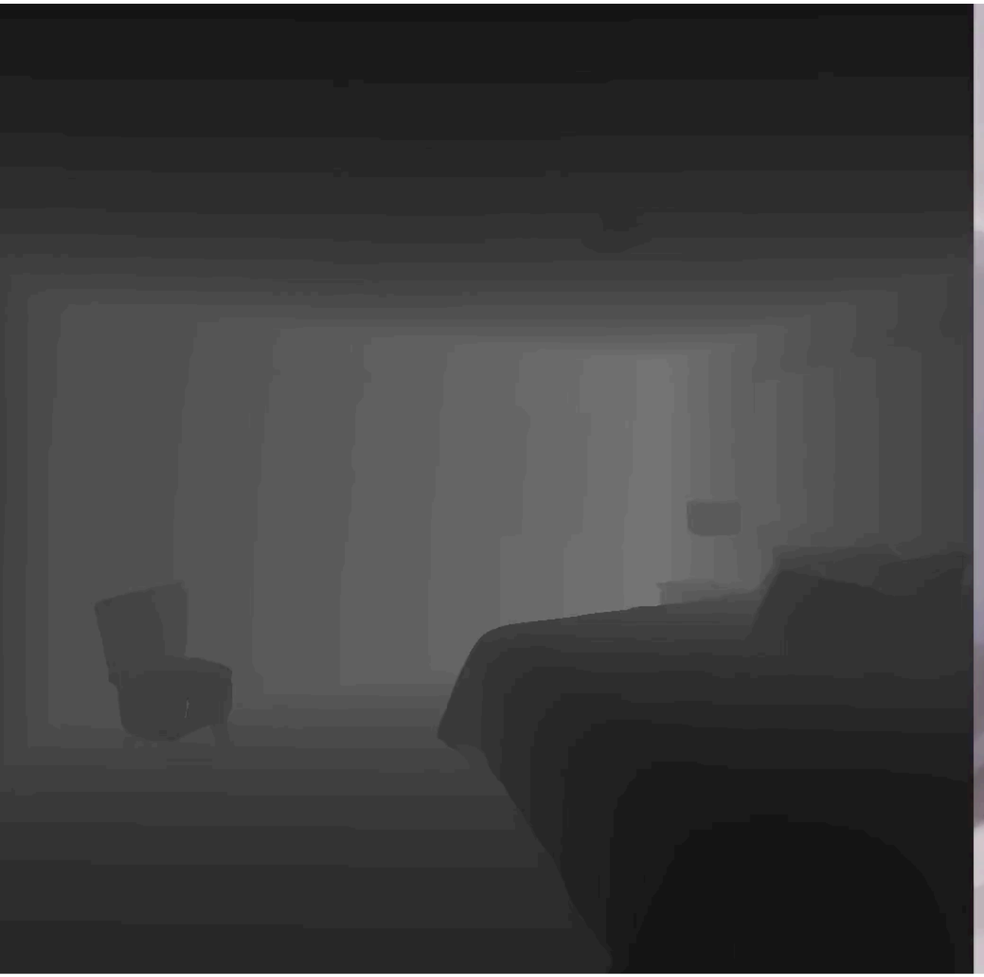
RGB and GPS+Compass



Top Down Map

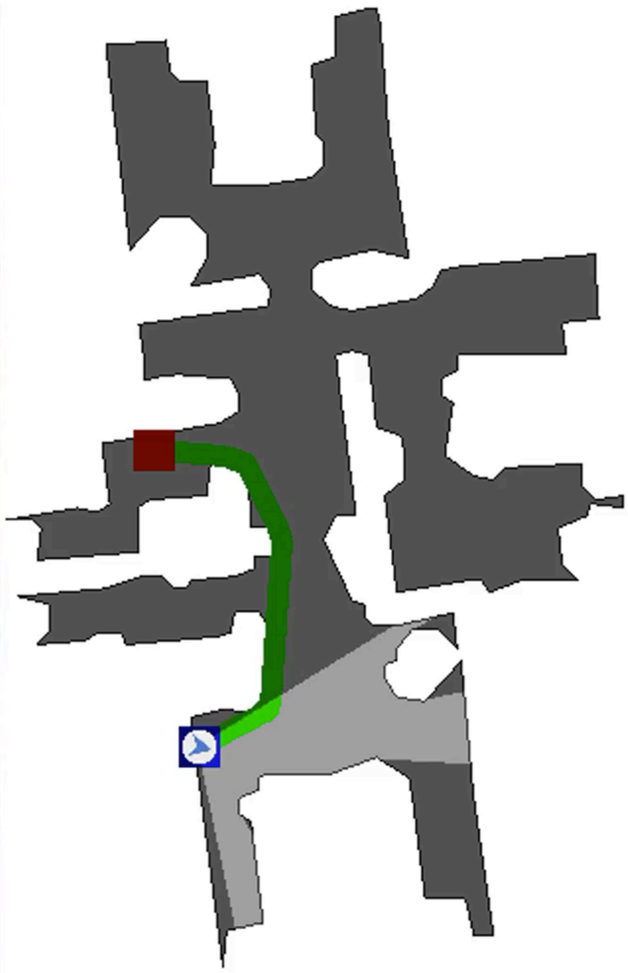
Slide credit: Erik Wijmans

PointGoal Navigation



Depth

RGB and GPS+Compass



Top Down Map

Slide credit: Erik Wijmans

Agent and Model Design

Slide credit: Erik Wijmans

Agent and Model Design

Sim

Real

Slide credit: Erik Wijmans

Agent and Model Design

- 0.6m tall cylinder with 0.2m radius

Slide credit: Erik Wijmans

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

Slide credit: Erik Wijmans

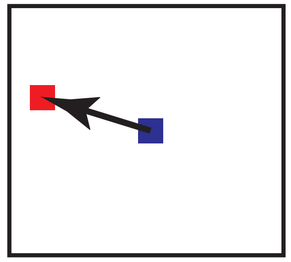
Agent and Model Design

Slide credit: Erik Wijmans

Agent and Model Design

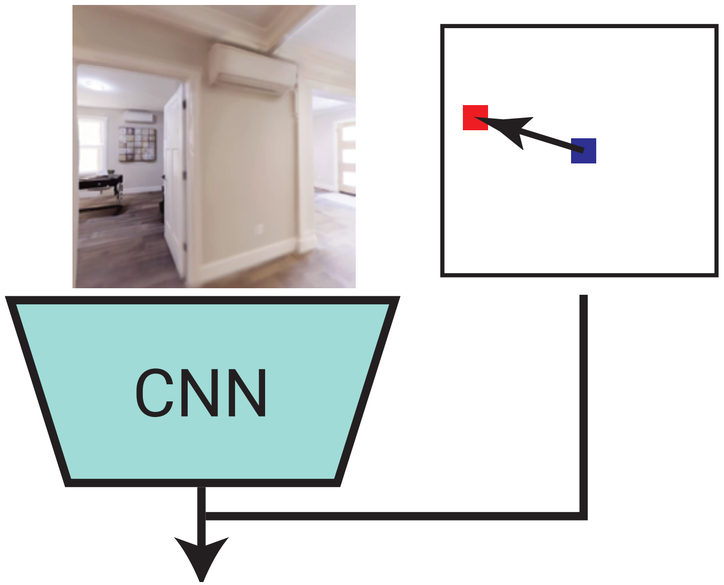
Slide credit: Erik Wijmans

Agent and Model Design



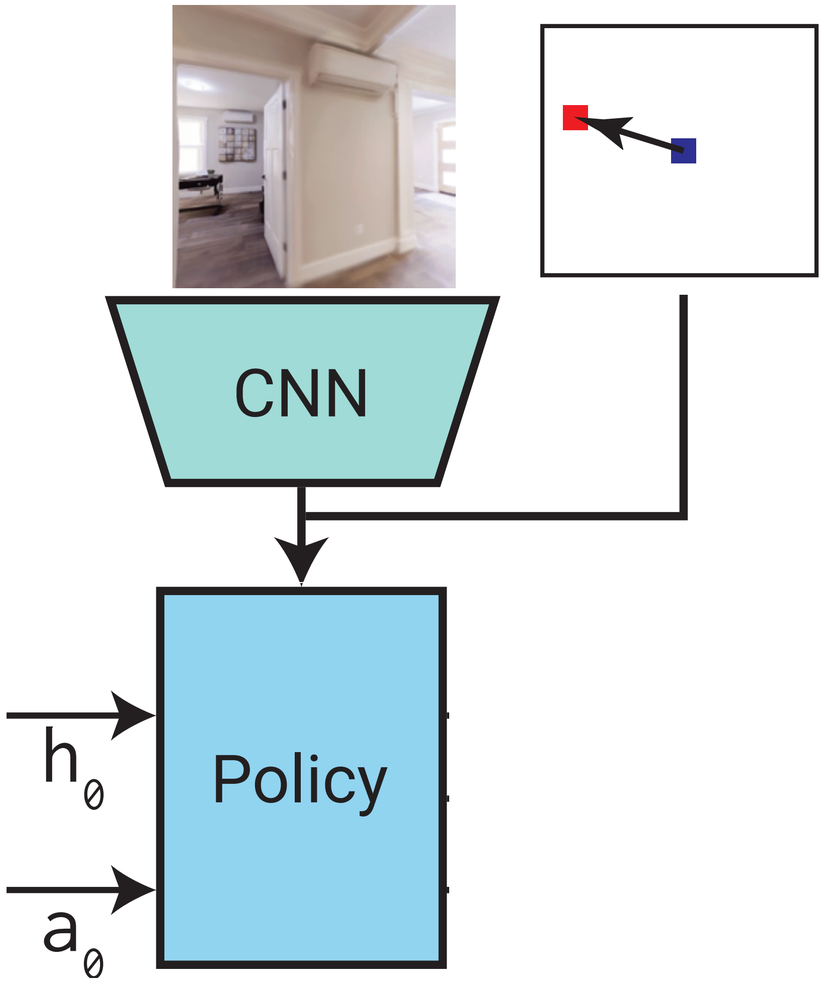
Slide credit: Erik Wijmans

Agent and Model Design



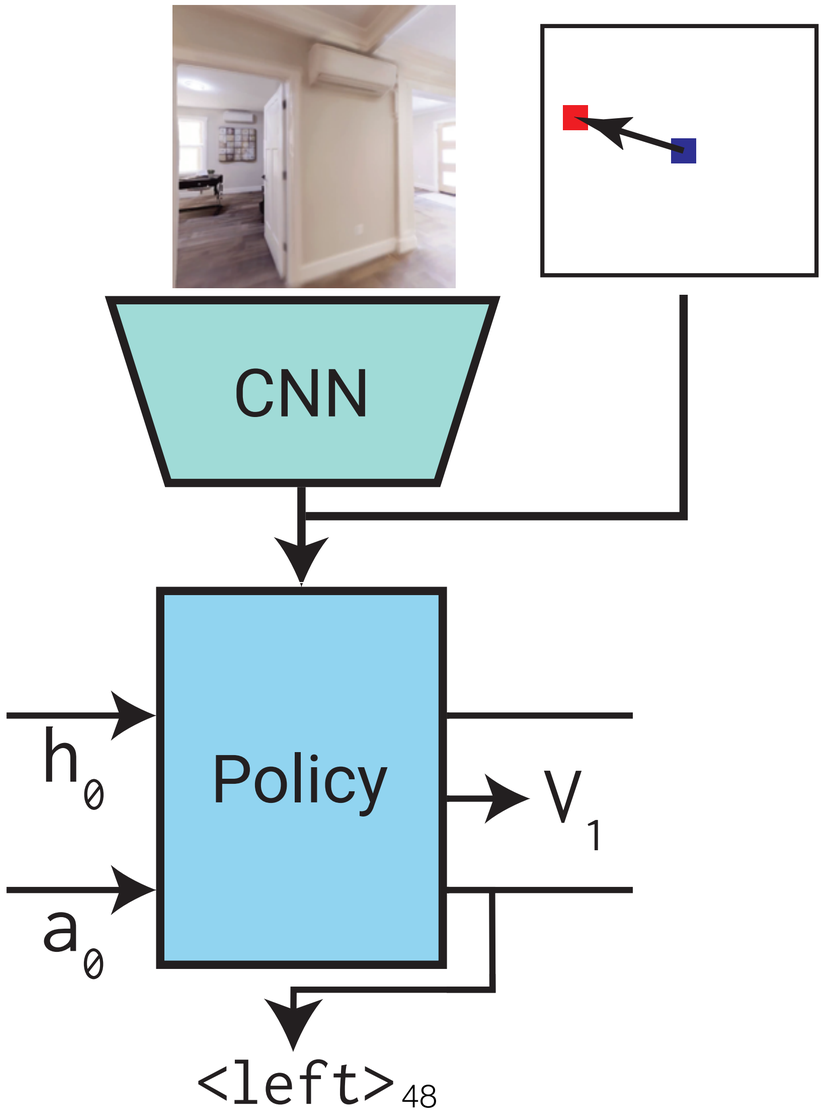
Slide credit: Erik Wijmans

Agent and Model Design



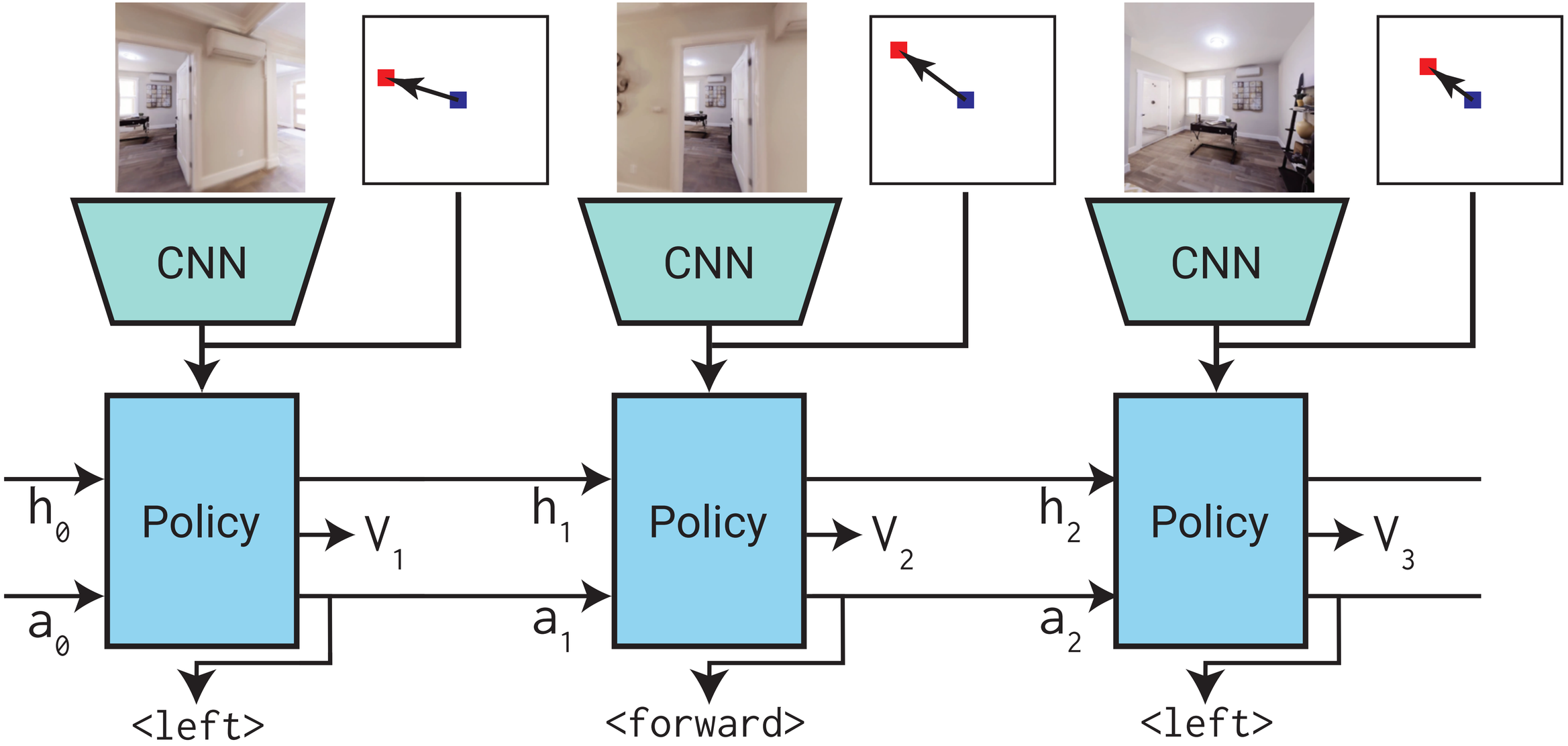
Slide credit: Erik Wijmans

Agent and Model Design



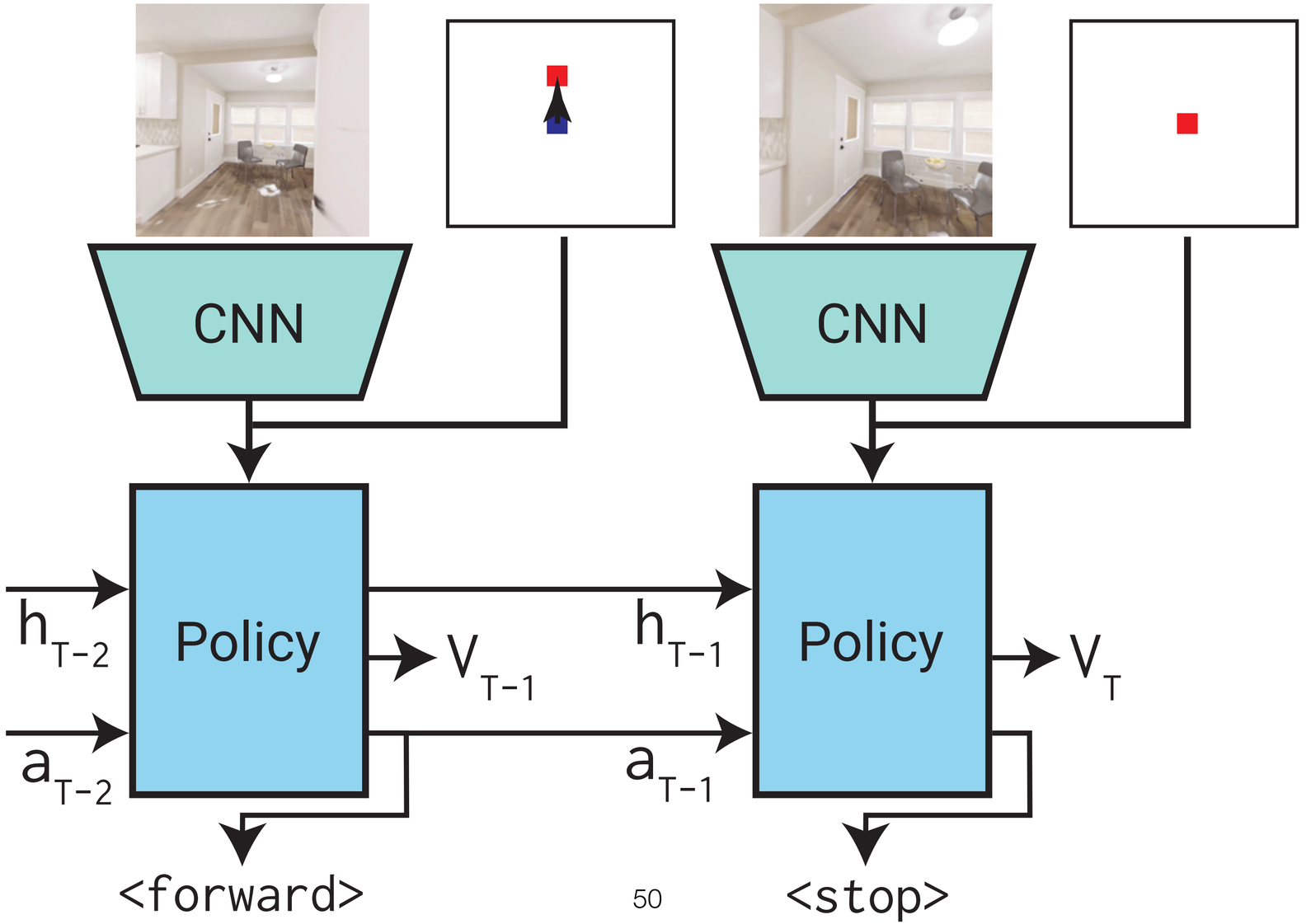
Slide credit: Erik Wijmans

Agent and Model Design



Slide credit: Erik Wijmans

Agent and Model Design



Slide credit: Erik Wijmans

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

- Policy Gradient
  - Advantage Actor Critic (A2C)
  - Trust Region Policy Optimization (TRPO)
  - Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation
  - Sim2Real Transfer
  - Robot2Robot Transfer

Outline

- Policy Gradient
  - Advantage Actor Critic (A2C)
  - Trust Region Policy Optimization (TRPO)
  - Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation
  - Sim2Real Transfer
  - Robot2Robot Transfer

Advantage Actor Critic (A2C)

Advantage Actor Critic (A2C)

Given a policy: $\pi_\theta(a_t \mid s_t)$

Advantage Actor Critic (A2C)

Given a policy: $\pi_\theta(a_t \mid s_t)$

Advantage function: $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$

Advantage Actor Critic (A2C)

Given a policy: $\pi_\theta(a_t \mid s_t)$

Advantage function: $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$

$A(s_t, a_t) > 0$ 👍

Action we chose was better than expected return

$A(s_t, a_t) < 0$ 👎

Action we chose was worse than expected return

Advantage Actor Critic (A2C)

Given a policy: $\pi_\theta(a_t \mid s_t)$

Advantage function: $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$

$A(s_t, a_t) > 0$ 👍

Action we chose was better than expected return

$A(s_t, a_t) < 0$ 👎

Action we chose was worse than expected return

Objective: Maximize

$$\mathcal{J}^{A2C}(\theta) = A(s_t, a_t) \cdot \log \pi_\theta(a_t \mid s_t)$$

Advantage Actor Critic (A2C)

Given a policy: $\pi_\theta(a_t \mid s_t)$

Advantage function: $A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$

$A(s_t, a_t) > 0$ 👍

Action we chose was better than expected return

$A(s_t, a_t) < 0$ 👎

Action we chose was worse than expected return

Objective: Maximize

$$\mathcal{J}^{A2C}(\theta) = A(s_t, a_t) \cdot \log \pi_\theta(a_t \mid s_t)$$

What is the problem with this?

Outline

- Policy Gradient
  - Advantage Actor Critic (A2C)
  - Trust Region Policy Optimization (TRPO)
  - Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation
  - Sim2Real Transfer
  - Robot2Robot Transfer

Trust Region Policy Optimization (TRPO)

Policy:

$$\pi_\theta(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Trust Region Policy Optimization (TRPO)

Policy:

$$\pi_\theta(a_t \mid s_t) \quad \pi_{\theta_{old}}(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Trust Region Policy Optimization (TRPO)

Policy:

$$\pi_\theta(a_t \mid s_t) \quad \pi_{\theta_{old}}(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$$

Trust Region Policy Optimization (TRPO)

Policy:

$$\pi_\theta(a_t \mid s_t) \quad \pi_{\theta_{old}}(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} > 1$$

Action is more likely under current policy than for old policy

Trust Region Policy Optimization (TRPO)

Policy:

$$\pi_\theta(a_t \mid s_t) \quad \pi_{\theta_{old}}(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} \quad < 1$$

Action is more unlikely under current policy than for old policy

Trust Region Policy Optimization (TRPO)

Policy:

$$\pi_\theta(a_t \mid s_t) \quad \pi_{\theta_{old}}(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$$

Surrogate Objective: Maximize

$$\mathcal{J}^{TRPO}(\theta) = A(s_t, a_t) \cdot r_t(\theta)$$

Policy Gradient Objective: Maximize

$$\mathcal{J}^{A2C}(\theta) = A(s_t, a_t) \cdot \log \pi_\theta(a_t \mid s_t)$$

Trust Region Policy Optimization (TRPO)

Policy:

$$\pi_\theta(a_t \mid s_t) \quad \pi_{\theta_{old}}(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$$

Surrogate Objective: Maximize

$$\mathcal{J}^{TRPO}(\theta) = A(s_t, a_t) \cdot r_t(\theta)$$

KL Divergence:

$$KL[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]$$

Trust Region Policy Optimization (TRPO)

Policy:

$$\pi_\theta(a_t \mid s_t) \quad \pi_{\theta_{old}}(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$$

KL Divergence:

$$KL[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]$$

Objective: Maximize

$$\mathcal{J}^{TRPO}(\theta) = A(s_t, a_t) \cdot r_t(\theta) - \beta KL[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]$$

Outline

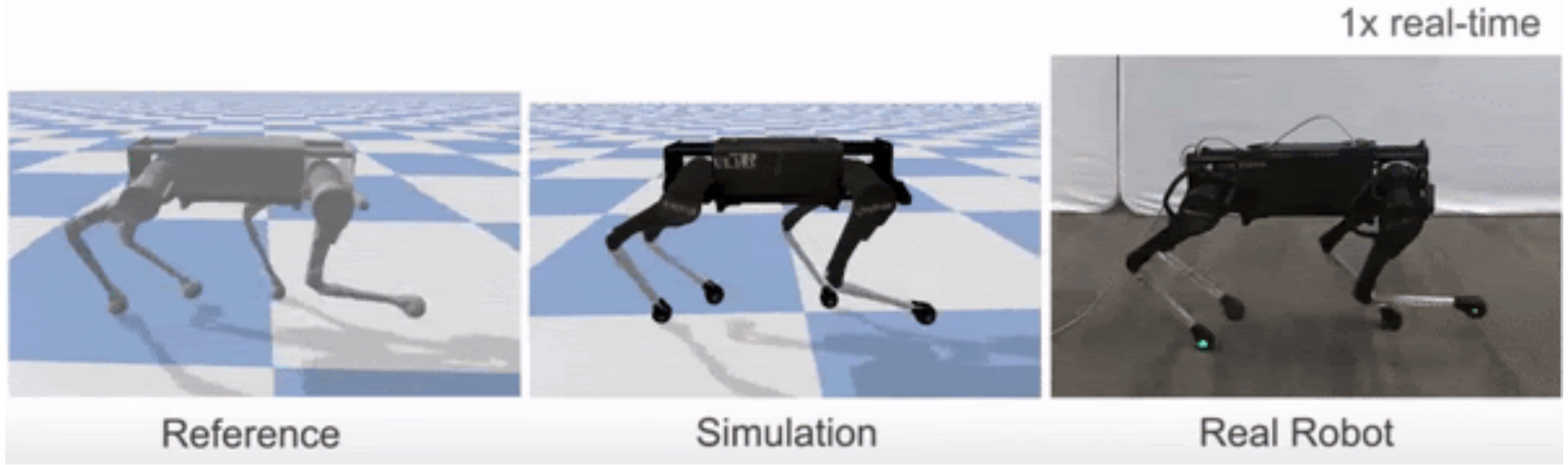
- Policy Gradient
  - Advantage Actor Critic (A2C)
  - Trust Region Policy Optimization (TRPO)
  - Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation
  - Sim2Real Transfer
  - Robot2Robot Transfer

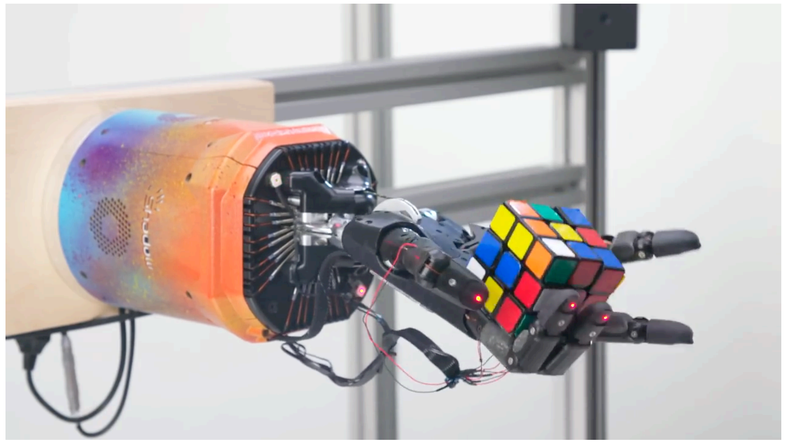
Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO)

OpenAI Five

AlphaStar: Mastering the Real-Time Strategy Game StarCraft II





Proximal Policy Optimization (PPO)

Policy:

$$\pi_\theta(a_t \mid s_t) \quad \pi_{\theta_{old}}(a_t \mid s_t)$$

Advantage function:

$$A(s_t, a_t) = Q(s_t, a_t) - V(s_t)$$

Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)}$$

KL Divergence:

$$KL[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_\theta(\cdot \mid s_t)]$$

Proximal Policy Optimization (PPO)

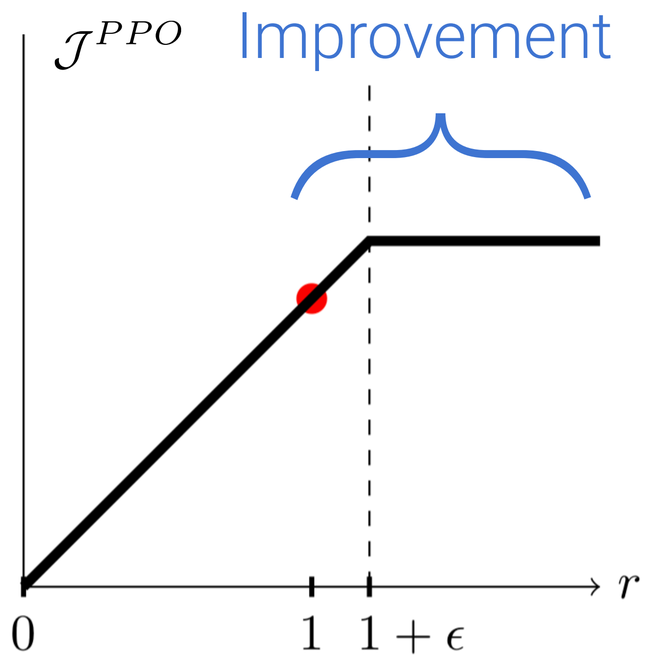
$$\mathcal{J}^{TRPO}(\theta) = A(s_t, a_t) \cdot r_t(\theta) - \beta KL[\pi_{\theta_{old}}(\cdot \mid s_t), \pi_{\theta}(\cdot \mid s_t)]$$

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

Proximal Policy Optimization (PPO)

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

$A(s_t, a_t) > 0$



Recall:

$A(s_t, a_t) > 0$ + 👍

Action we chose was better than expected return

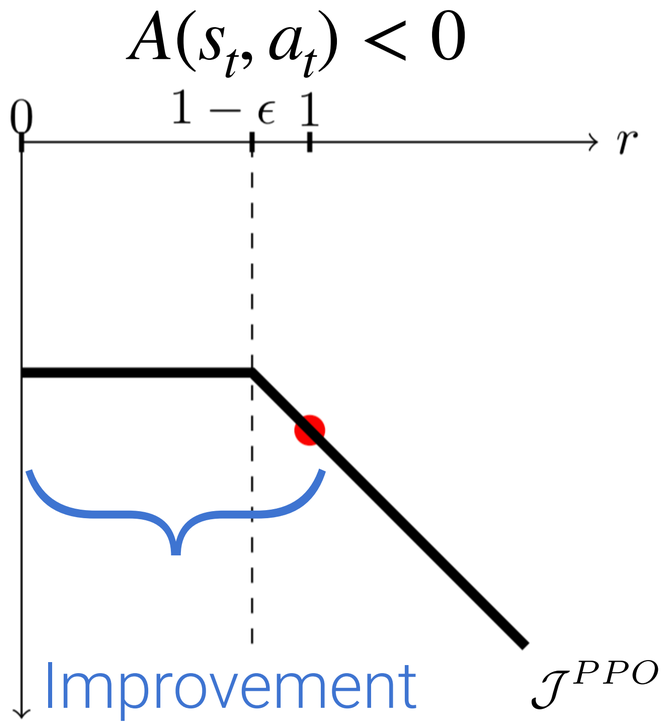
Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} > 1$$

Action is much more likely under current policy than for old policy

Proximal Policy Optimization (PPO)

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



Recall:

$$A(s_t, a_t) < 0$$ – 👎

Action we chose was worse than expected return

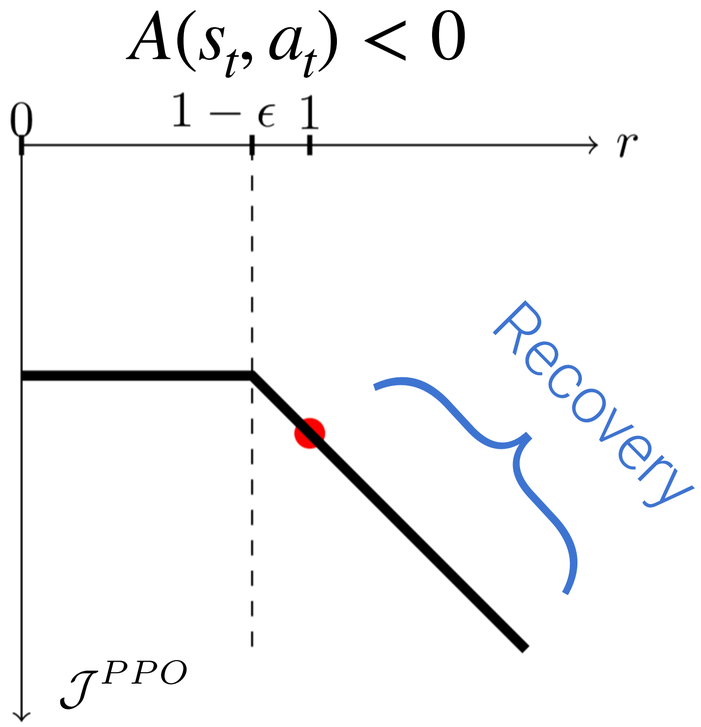
Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} < 1$$

Action is much more unlikely under current policy than for old policy

Proximal Policy Optimization (PPO)

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



Recall:

$$A(s_t, a_t) < 0 \quad - \text{👎}$$

Action we chose was worse than expected return

Probability Ratio:

$$r_t(\theta) = \frac{\pi_\theta(a_t \mid s_t)}{\pi_{\theta_{old}}(a_t \mid s_t)} > 1$$

Action is much more likely under current policy than for old policy

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

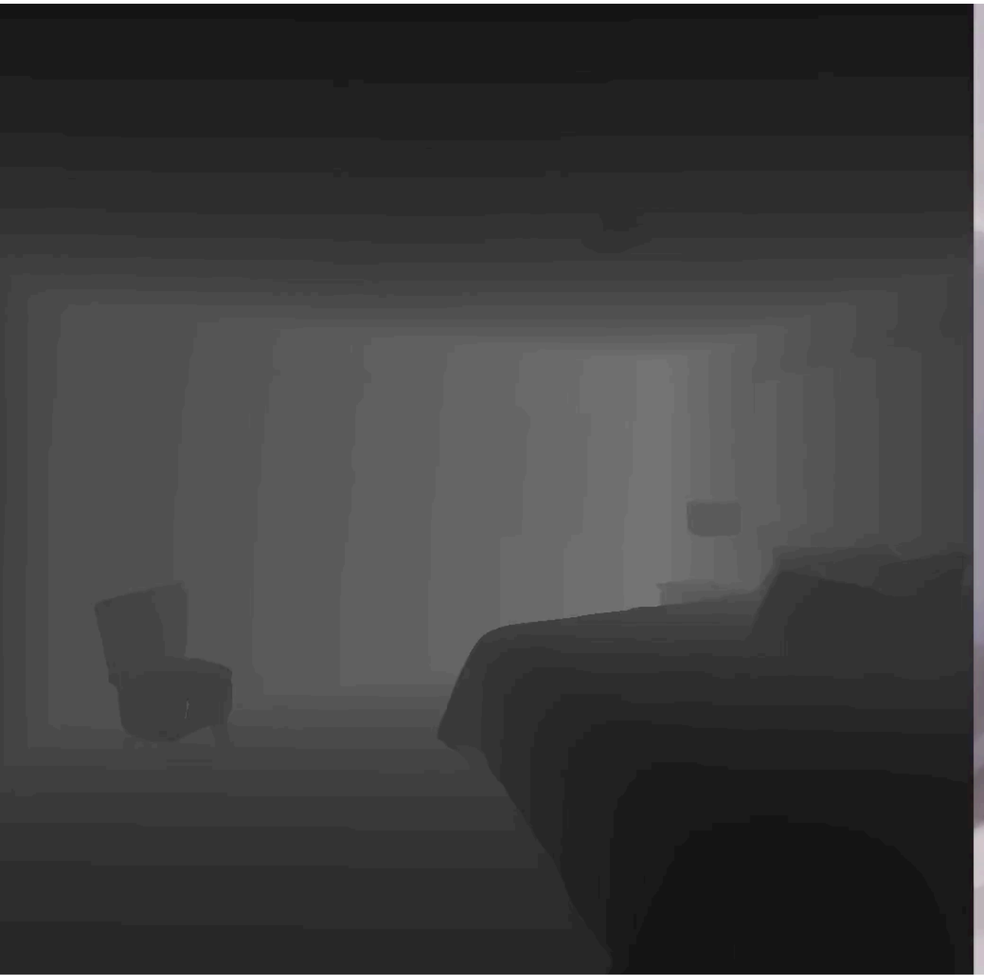
- Collect a set of trajectories using current policy
- For a few epochs (typically 2 or 4)
  - Sample mini batches from rollout (typically 2 or 4)
    - Update the policy via step of PPO objective
- Repeat

Slide credit: Erik Wijmans

Outline

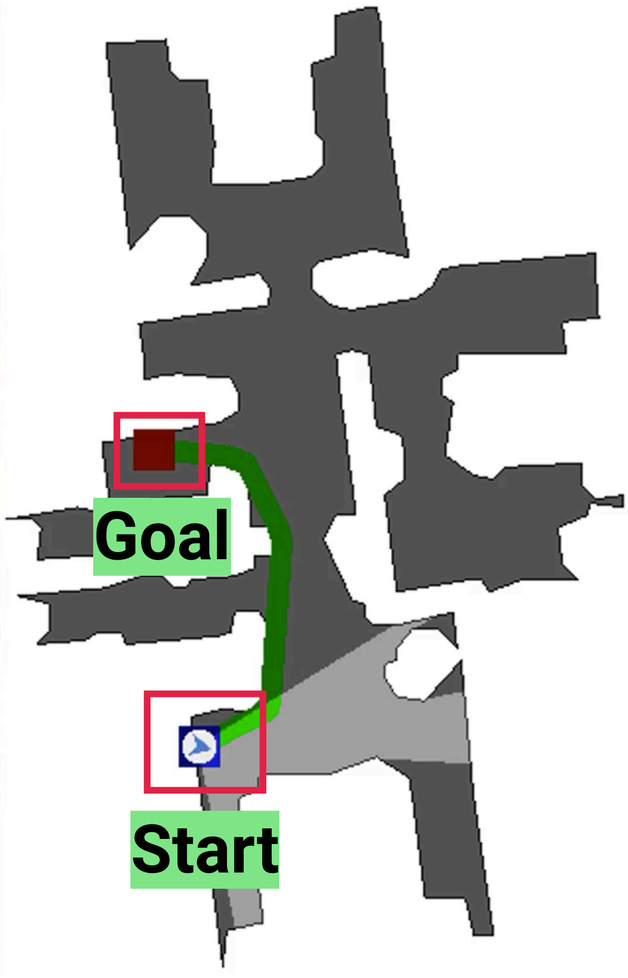
- Policy Gradient
  - Trust Region Policy Optimization (TRPO)
  - Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation
  - Sim2Real Transfer
  - Robot2Robot Transfer

PointGoal Navigation Results

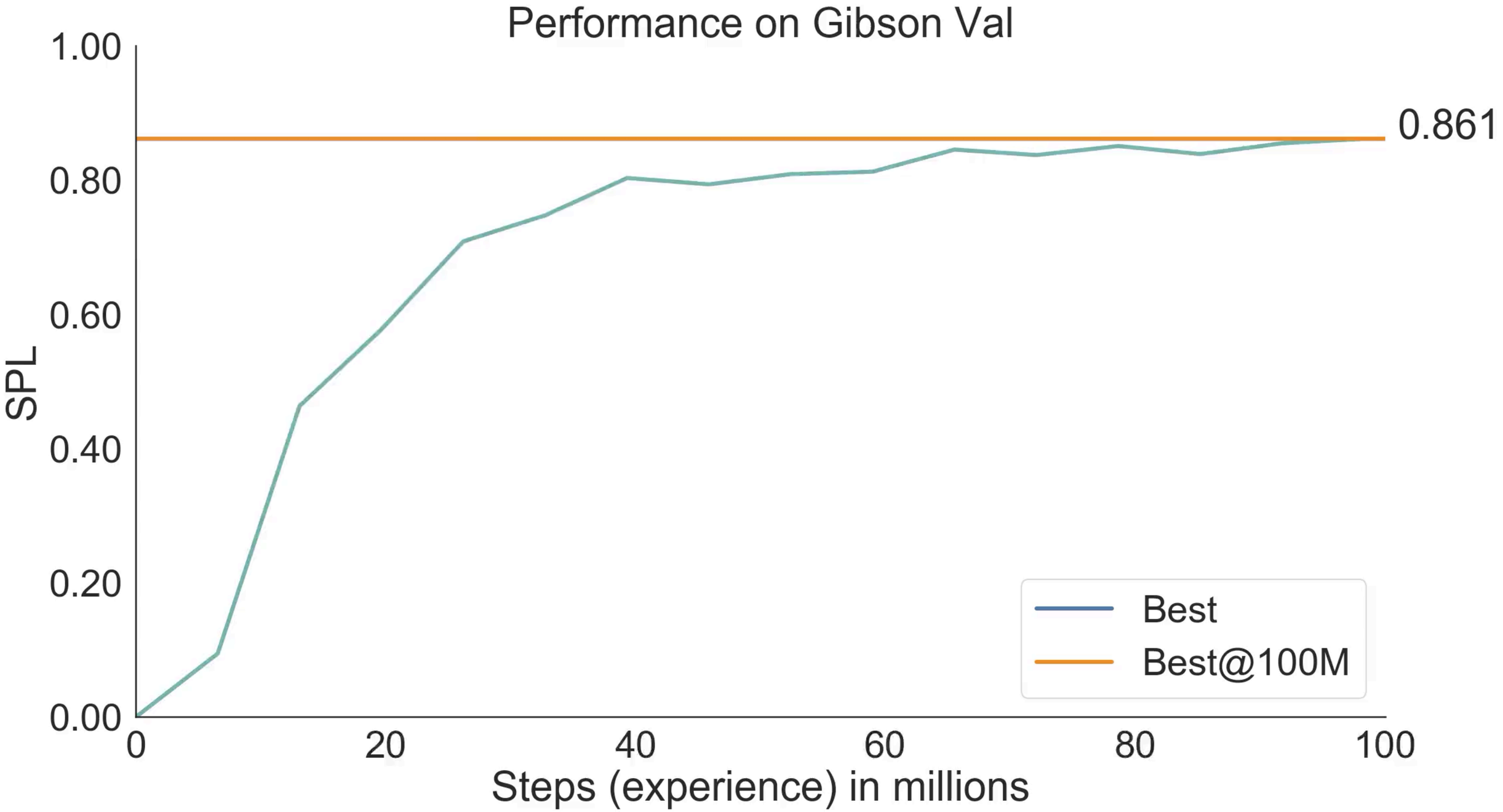


Depth

RGB and GPS+Compass



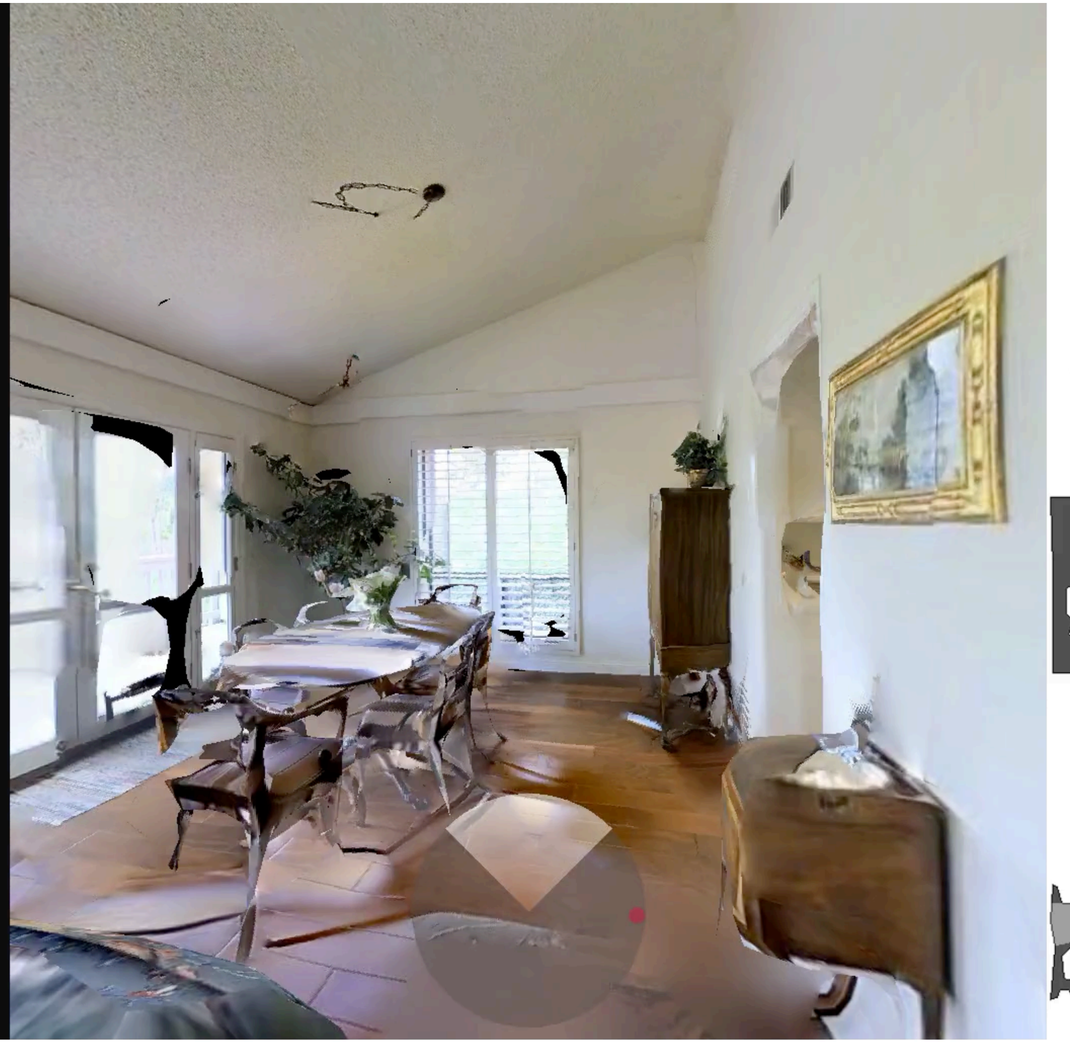
Top Down Map



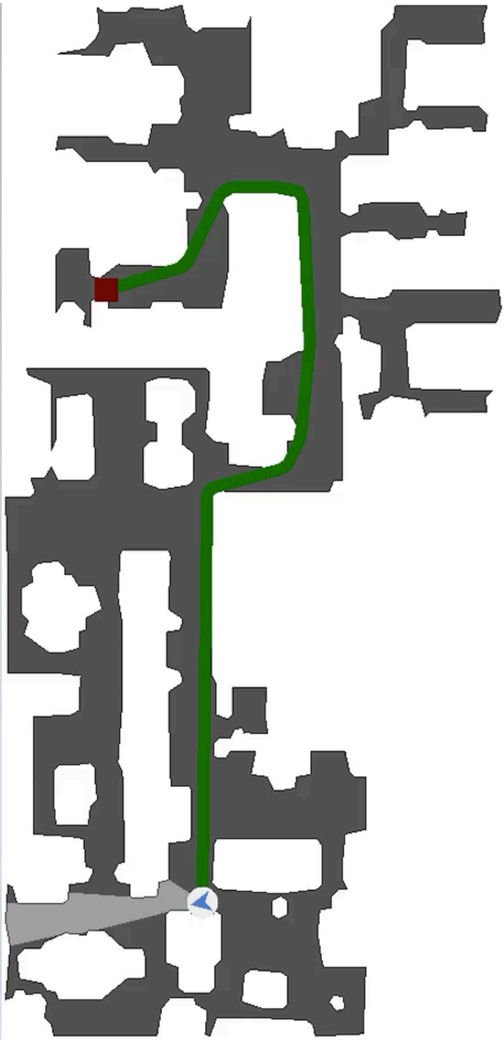
Wijmans et al. ICLR 2020

Qualitative Results

Depth



RGB and GPS+Compass

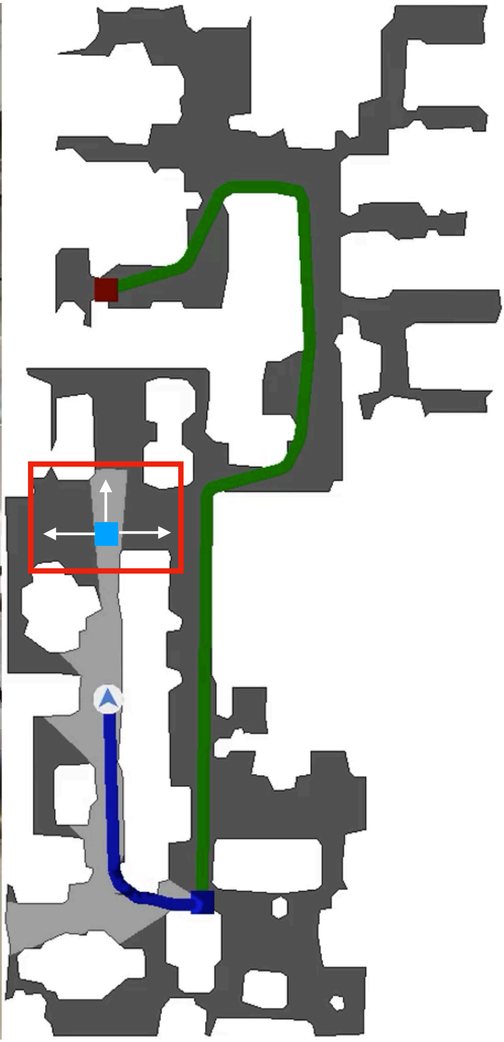


Top Down Map

Slide credit: Erik Wijmans

Depth

RGB and GPS+Compass

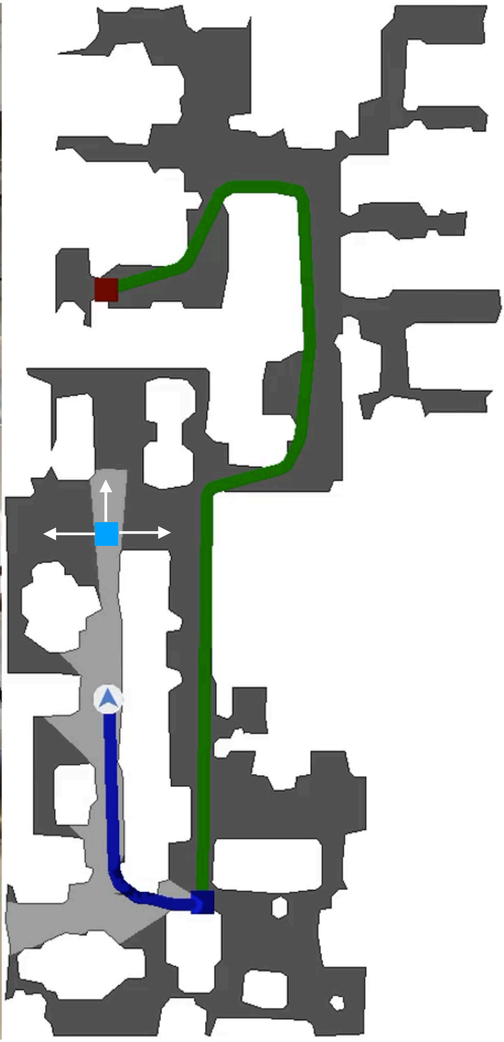


Top Down Map

Slide credit: Erik Wijmans

Depth

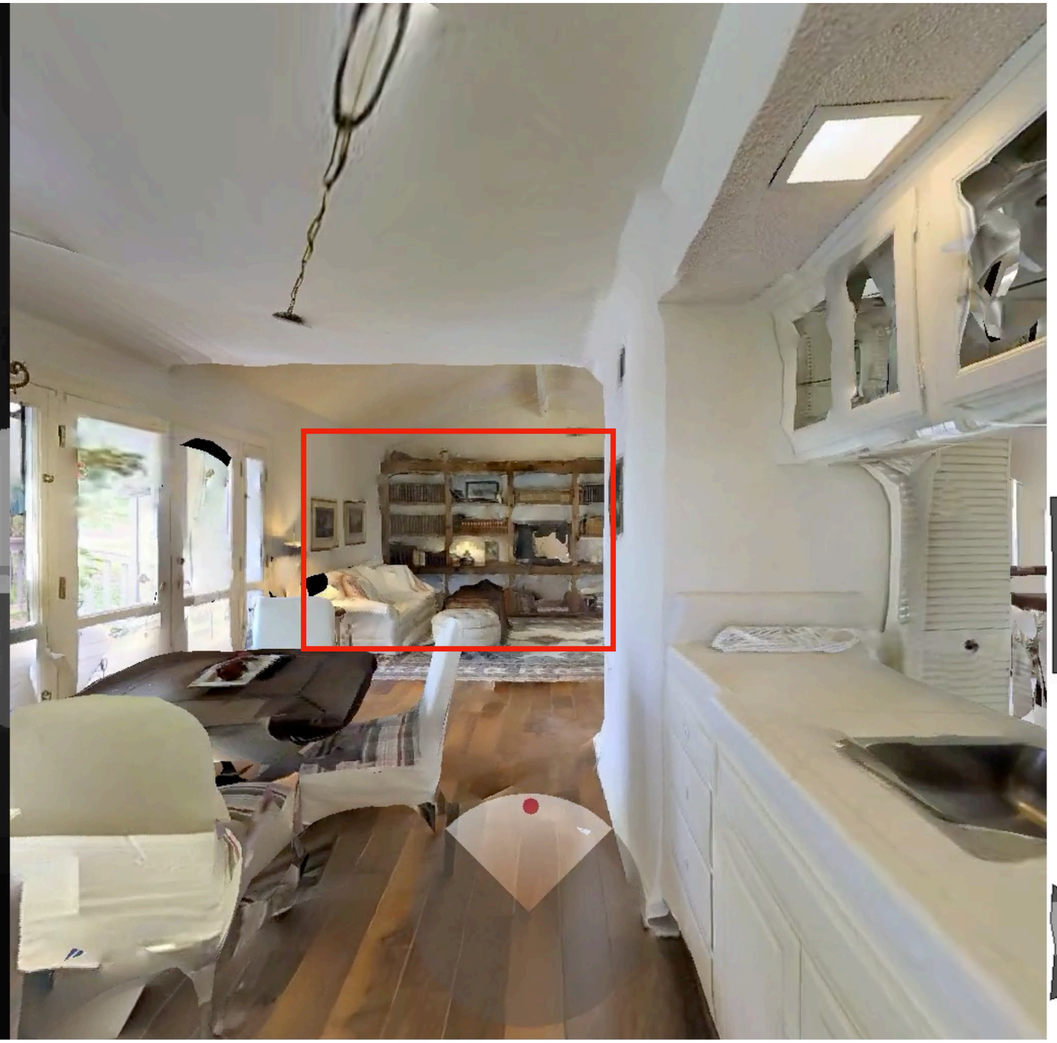
RGB and GPS+Compass



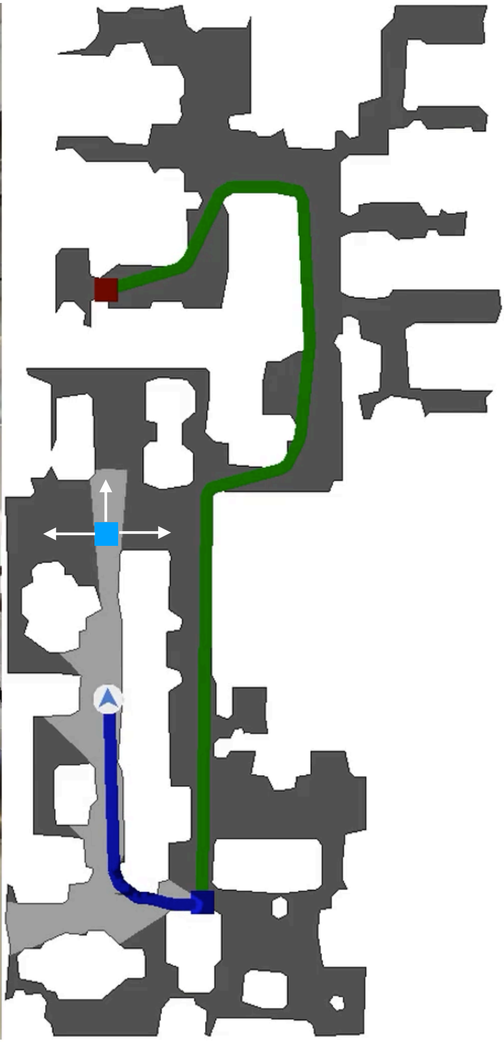
Top Down Map

Slide credit: Erik Wijmans

Depth

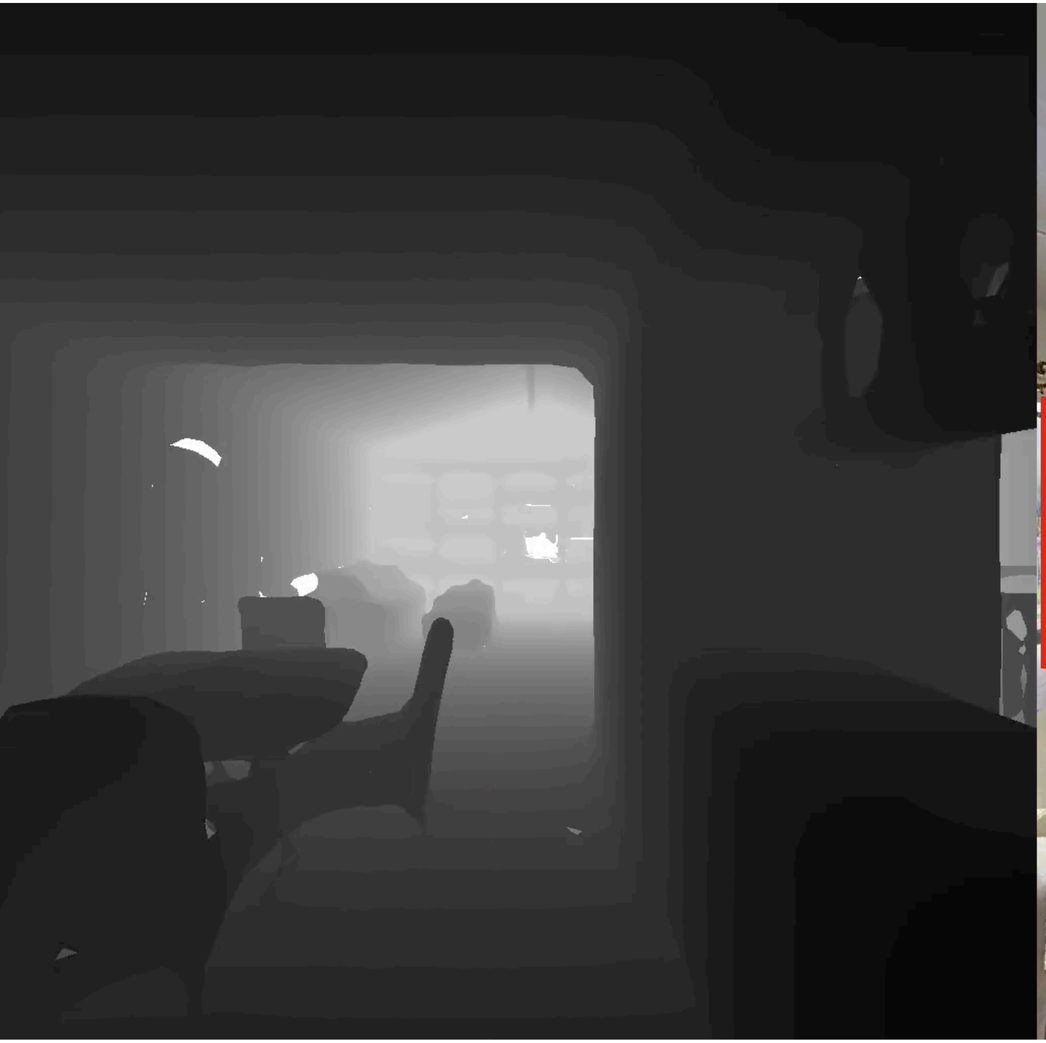


RGB and GPS+Compass



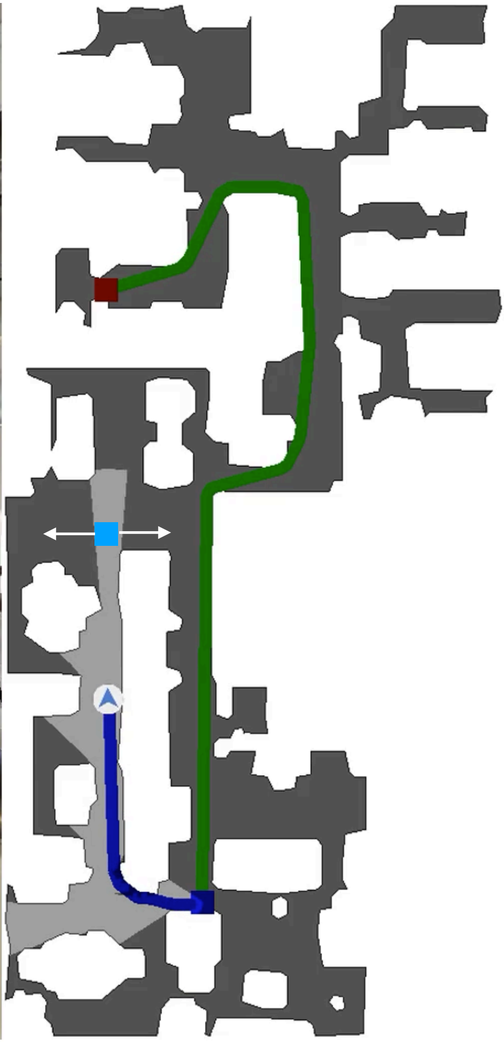
Top Down Map

Slide credit: Erik Wijmans



Depth

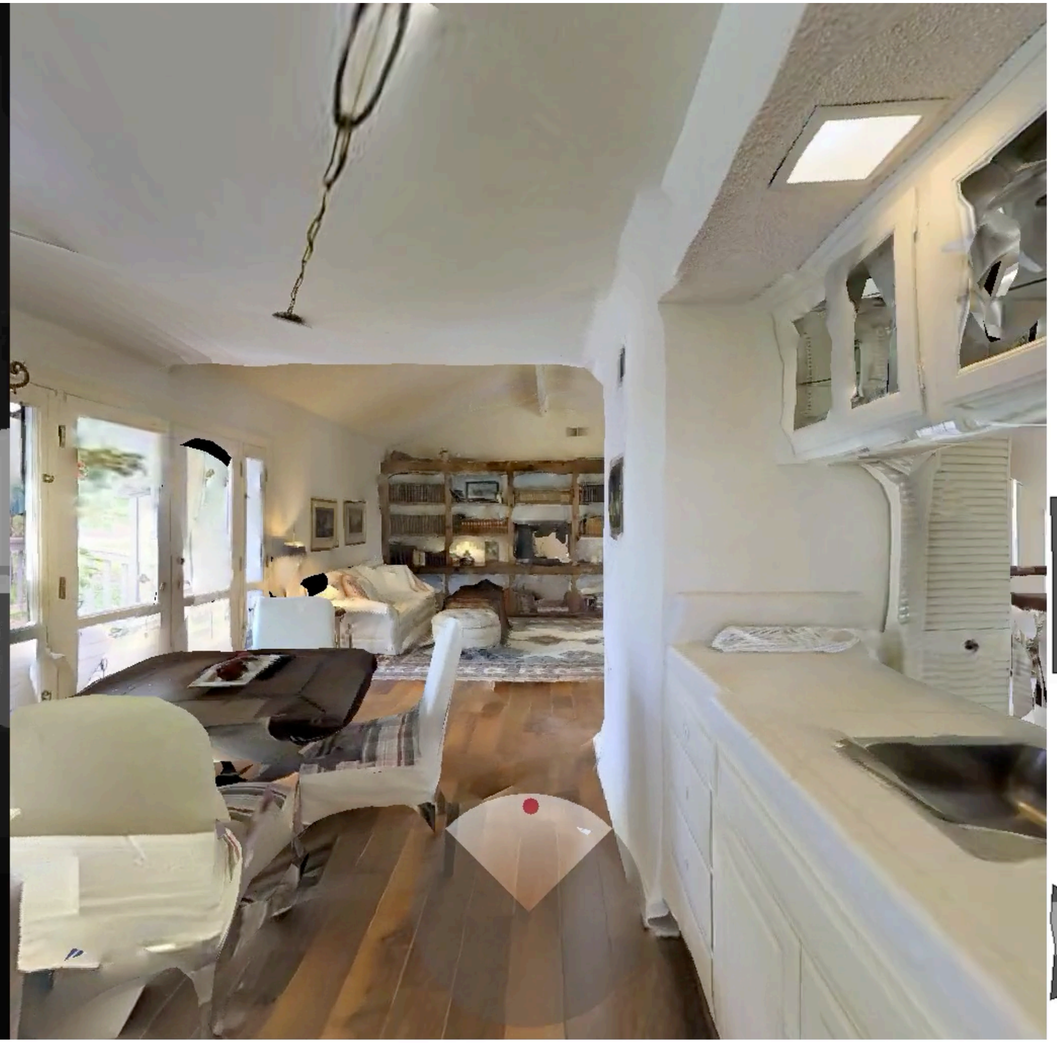
RGB and GPS+Compass



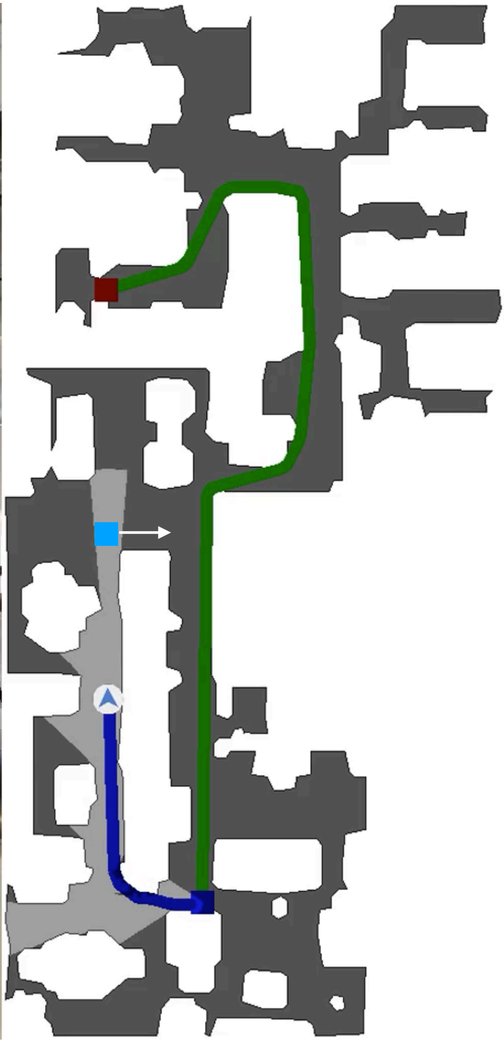
Top Down Map

Slide credit: Erik Wijmans

Depth



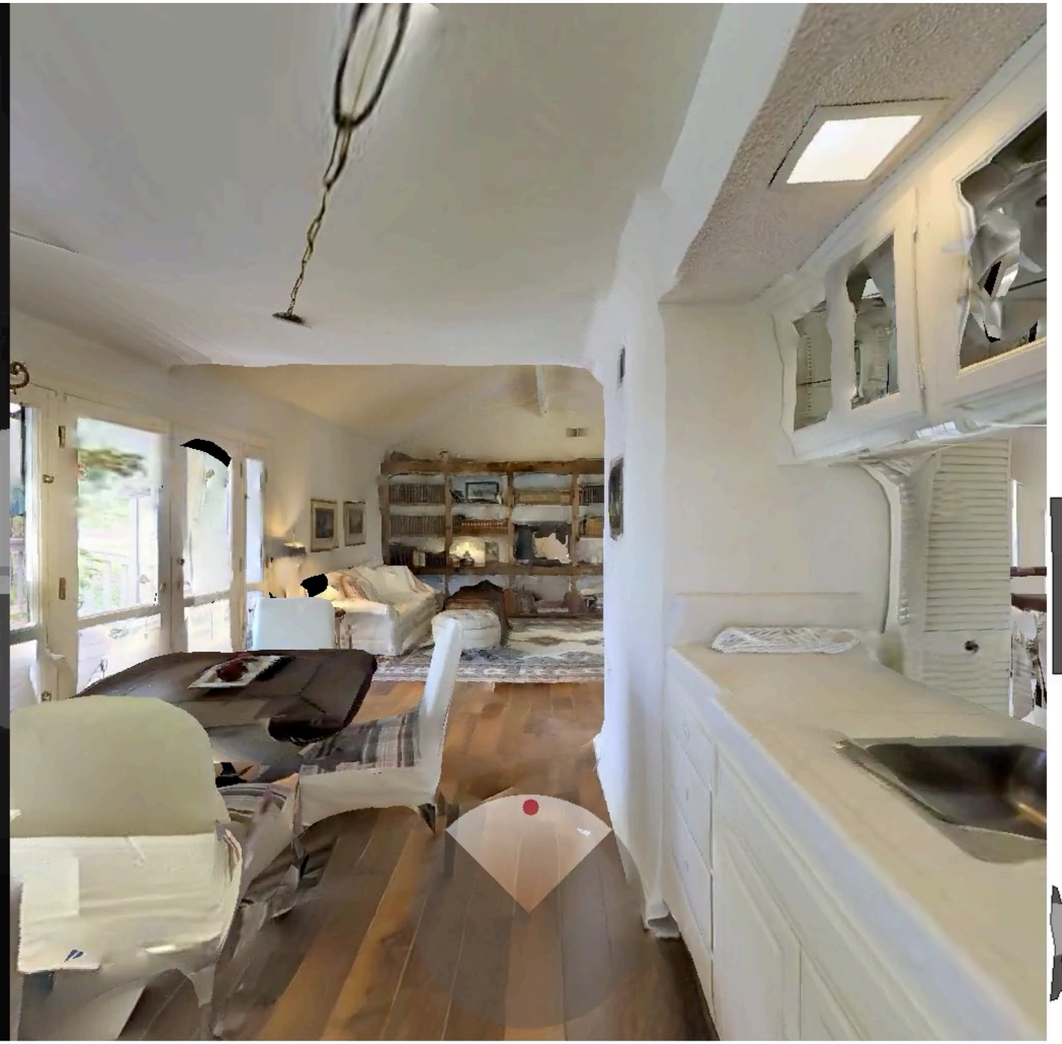
RGB and GPS+Compass



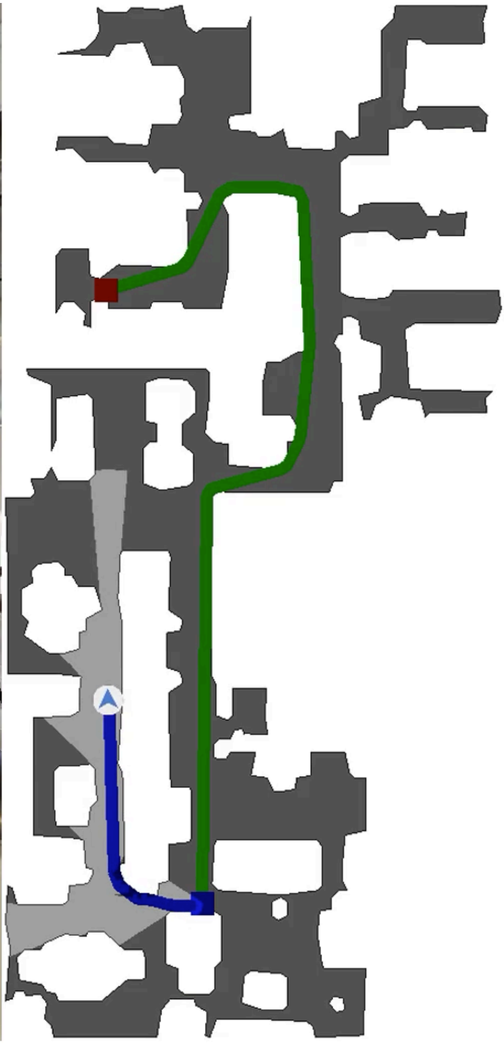
Top Down Map

Slide credit: Erik Wijmans

Depth



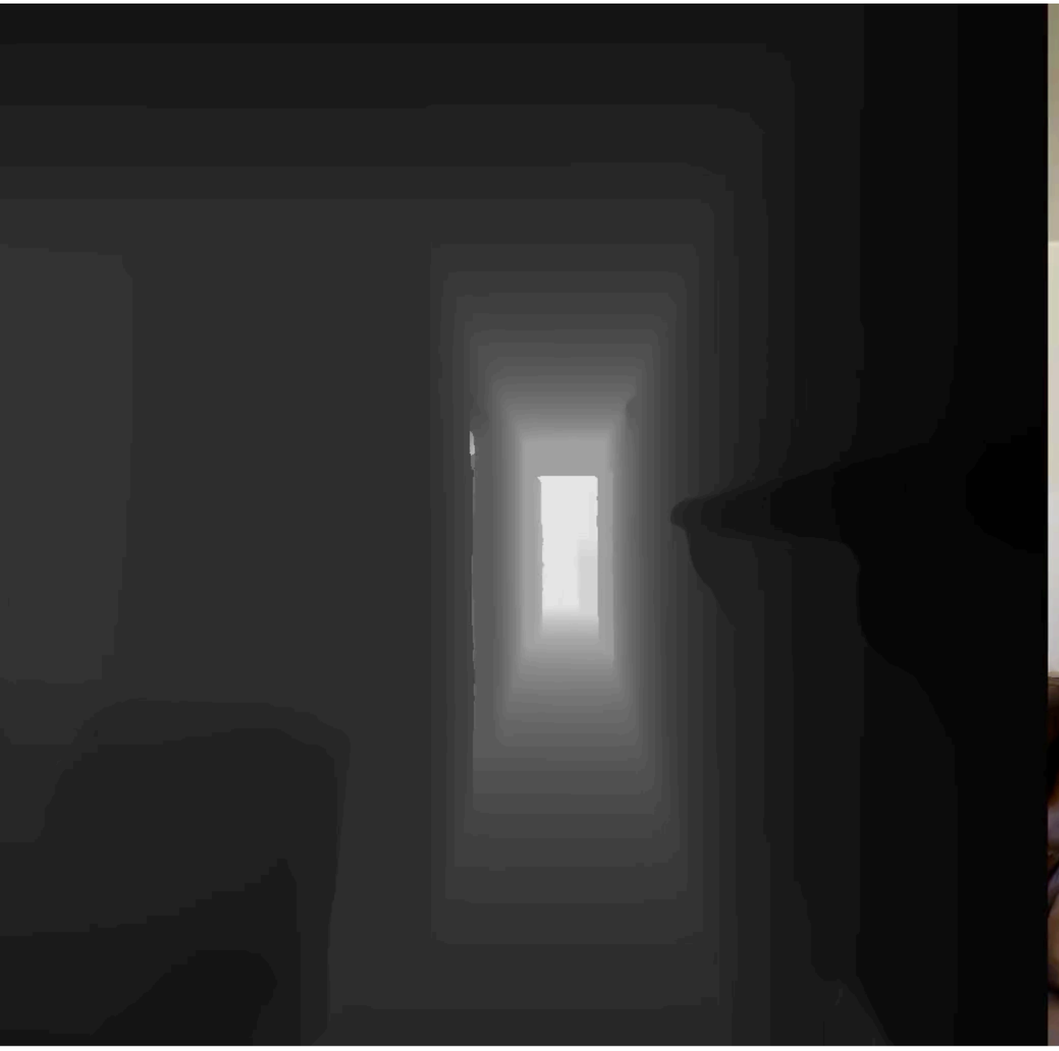
RGB and GPS+Compass



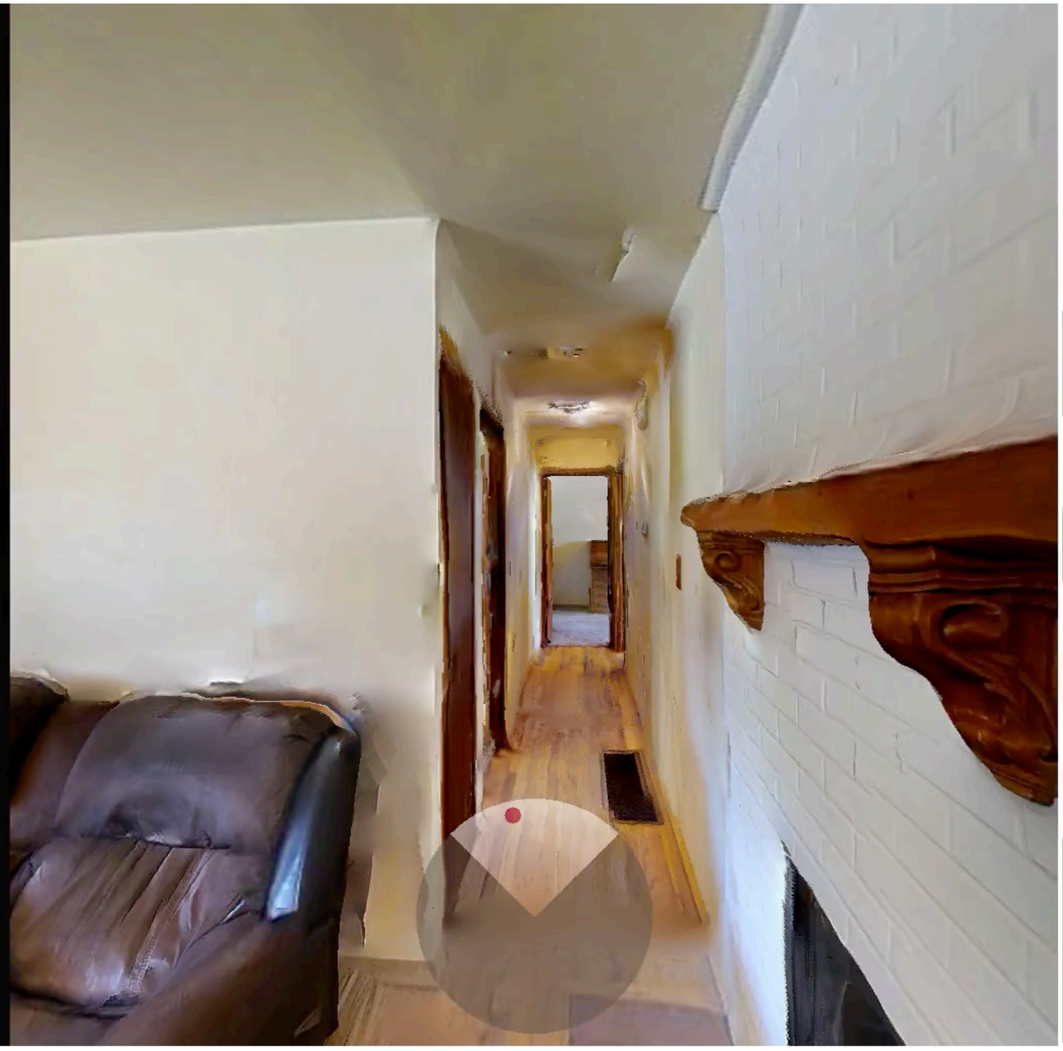
Top Down Map

Slide credit: Erik Wijmans

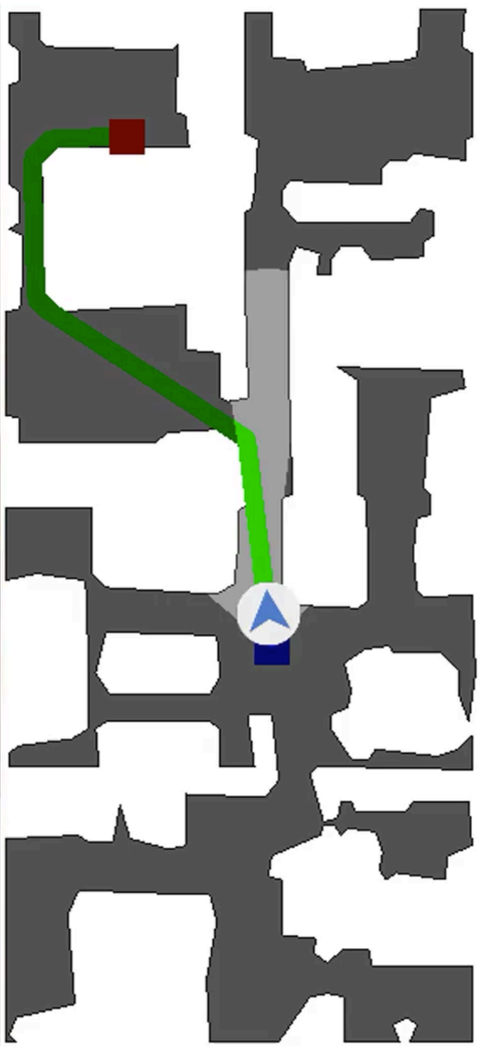
Backtracking



Depth

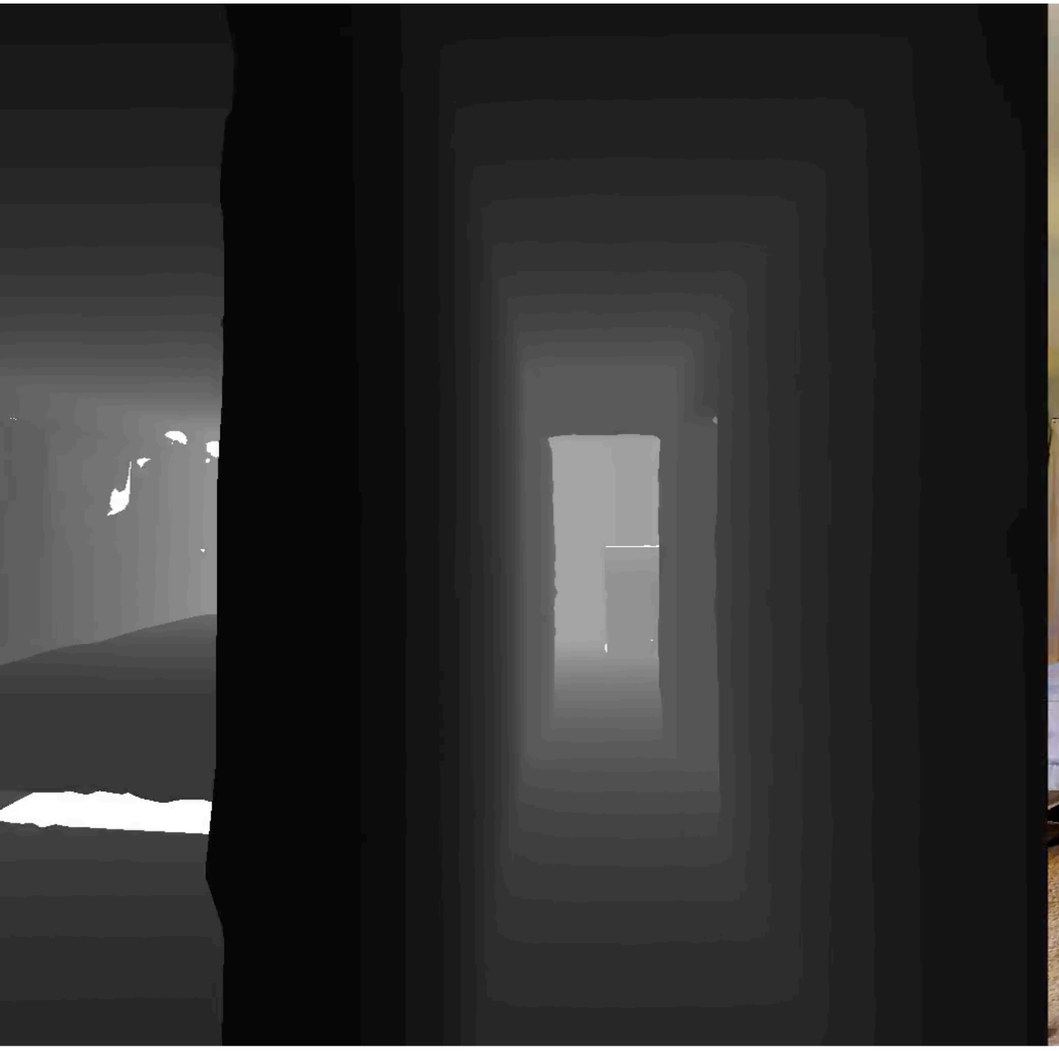


RGB and GPS+Compass



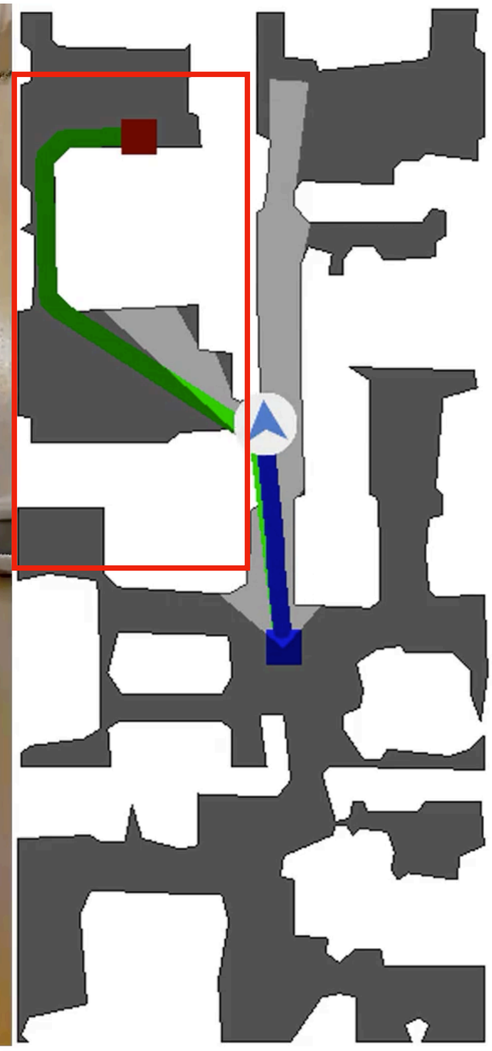
Top Down Map

Slide credit: Erik Wijmans



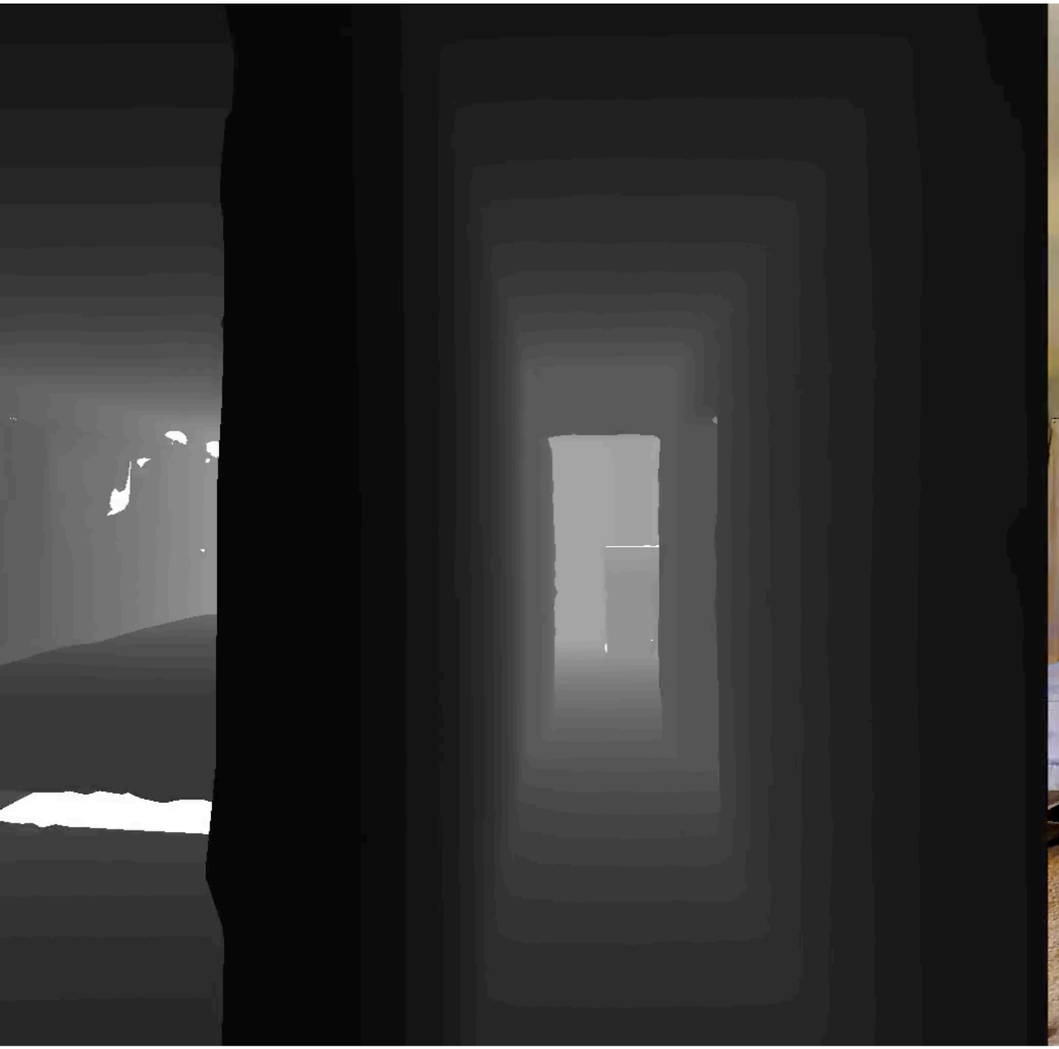
Depth

RGB and GPS+Compass



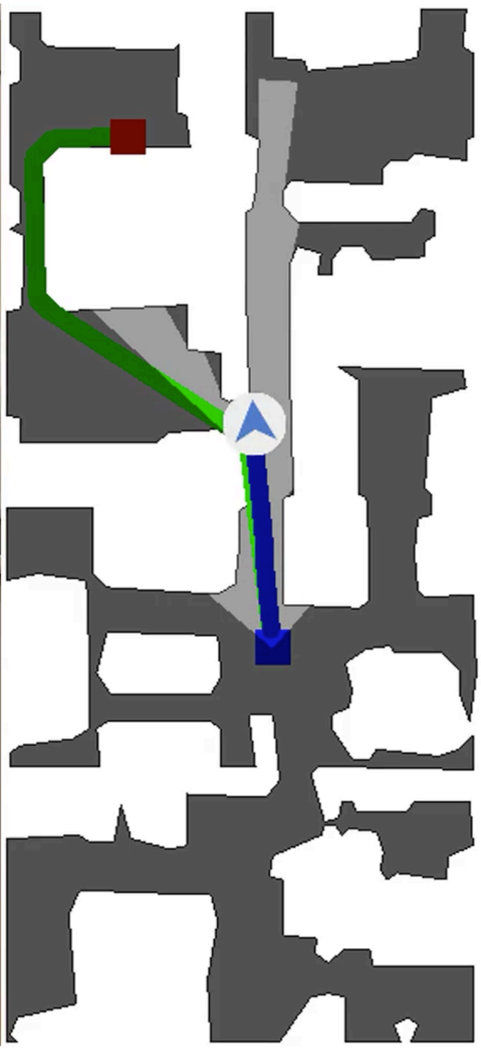
Top Down Map

Slide credit: Erik Wijmans



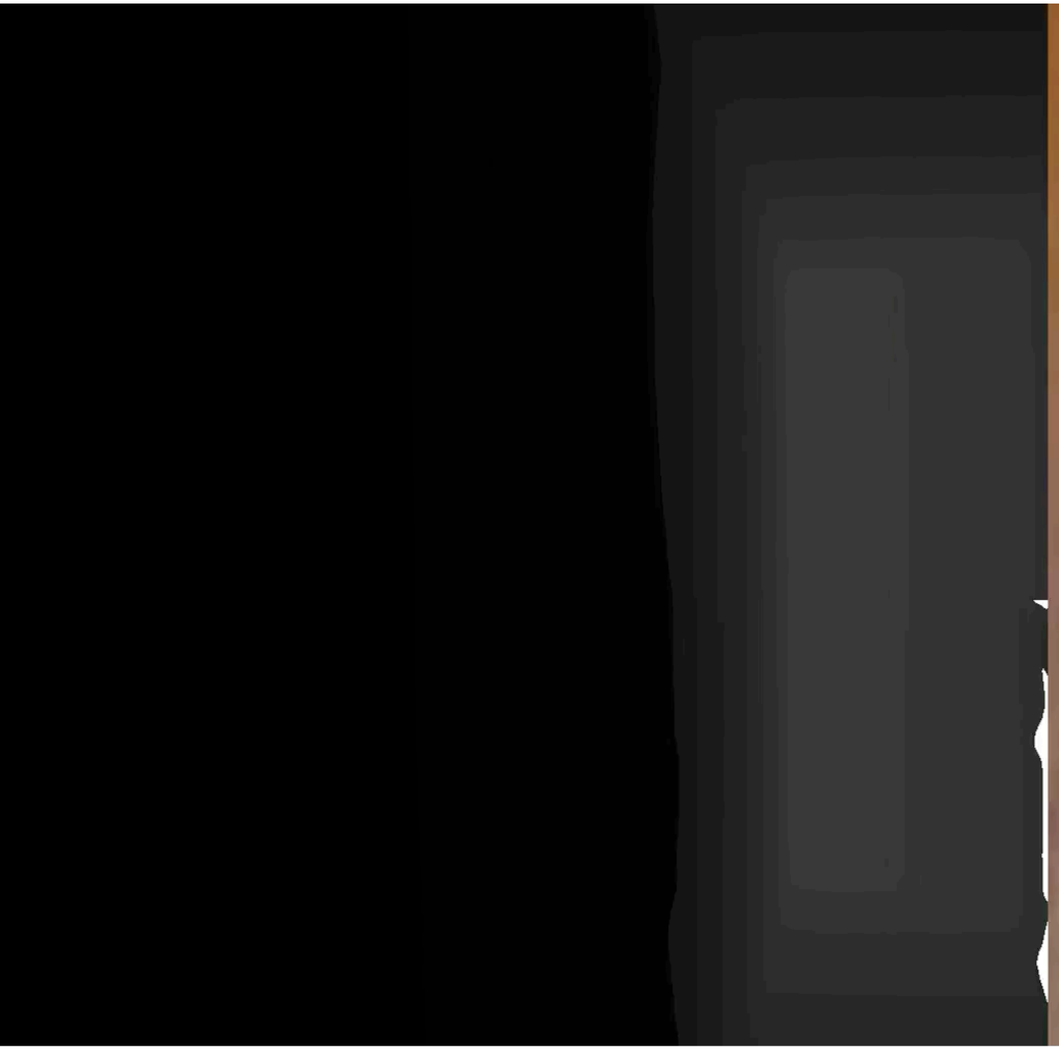
Depth

RGB and GPS+Compass

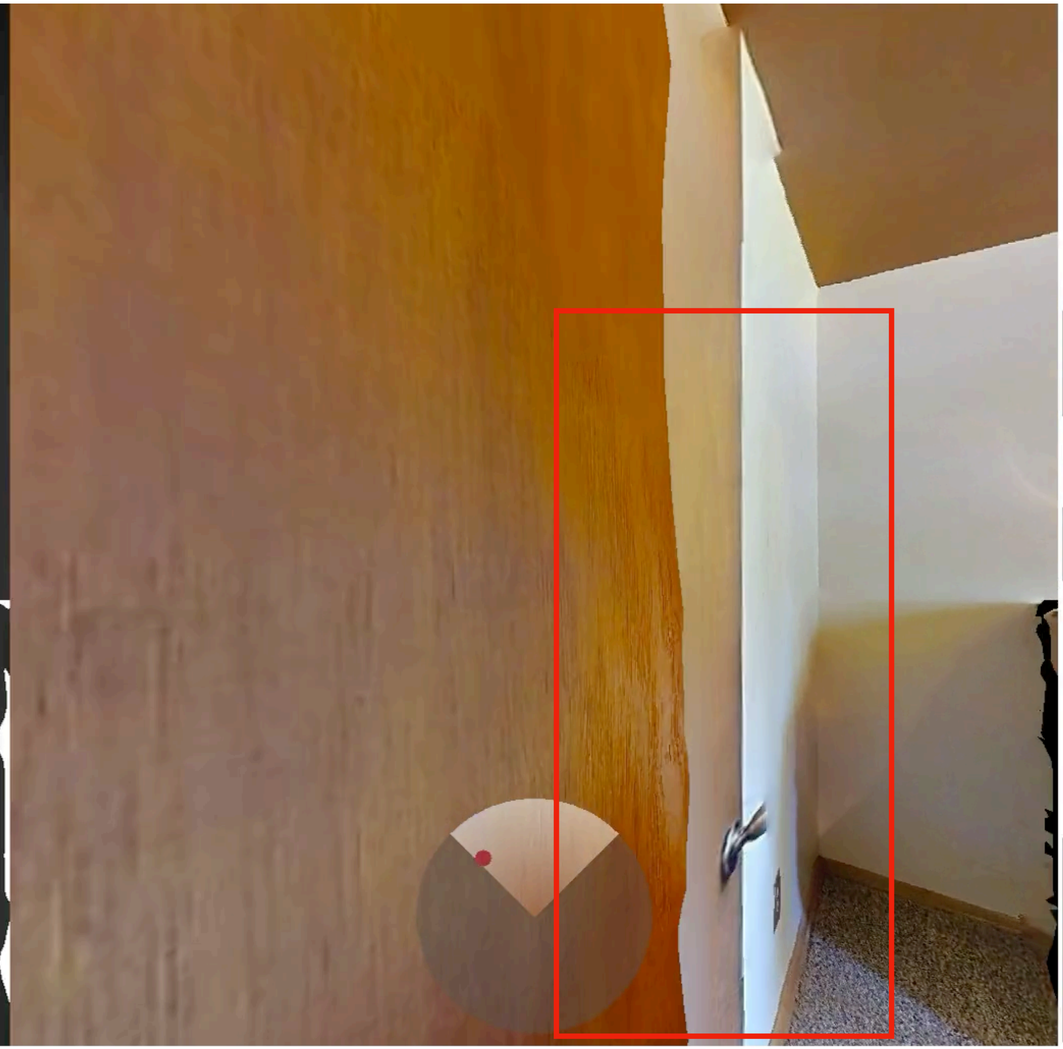


Top Down Map

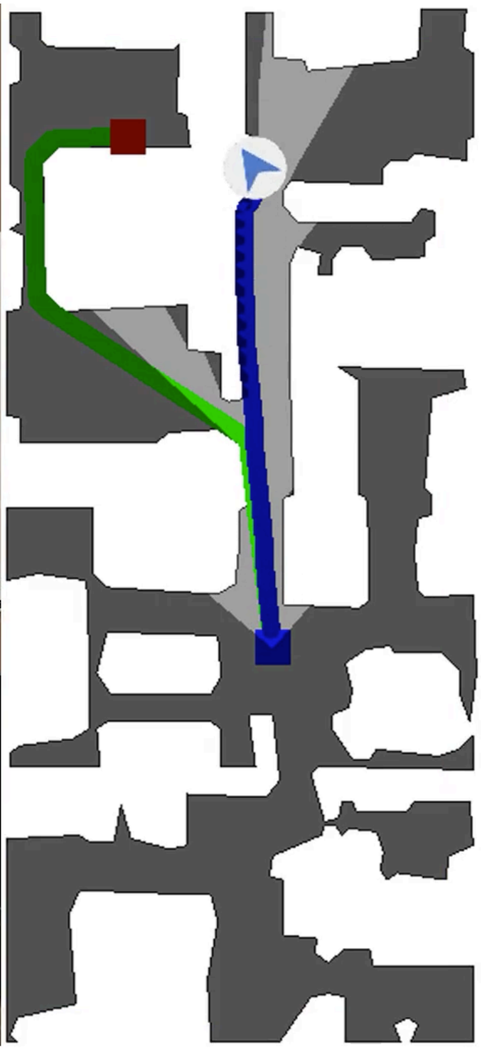
Slide credit: Erik Wijmans



Depth

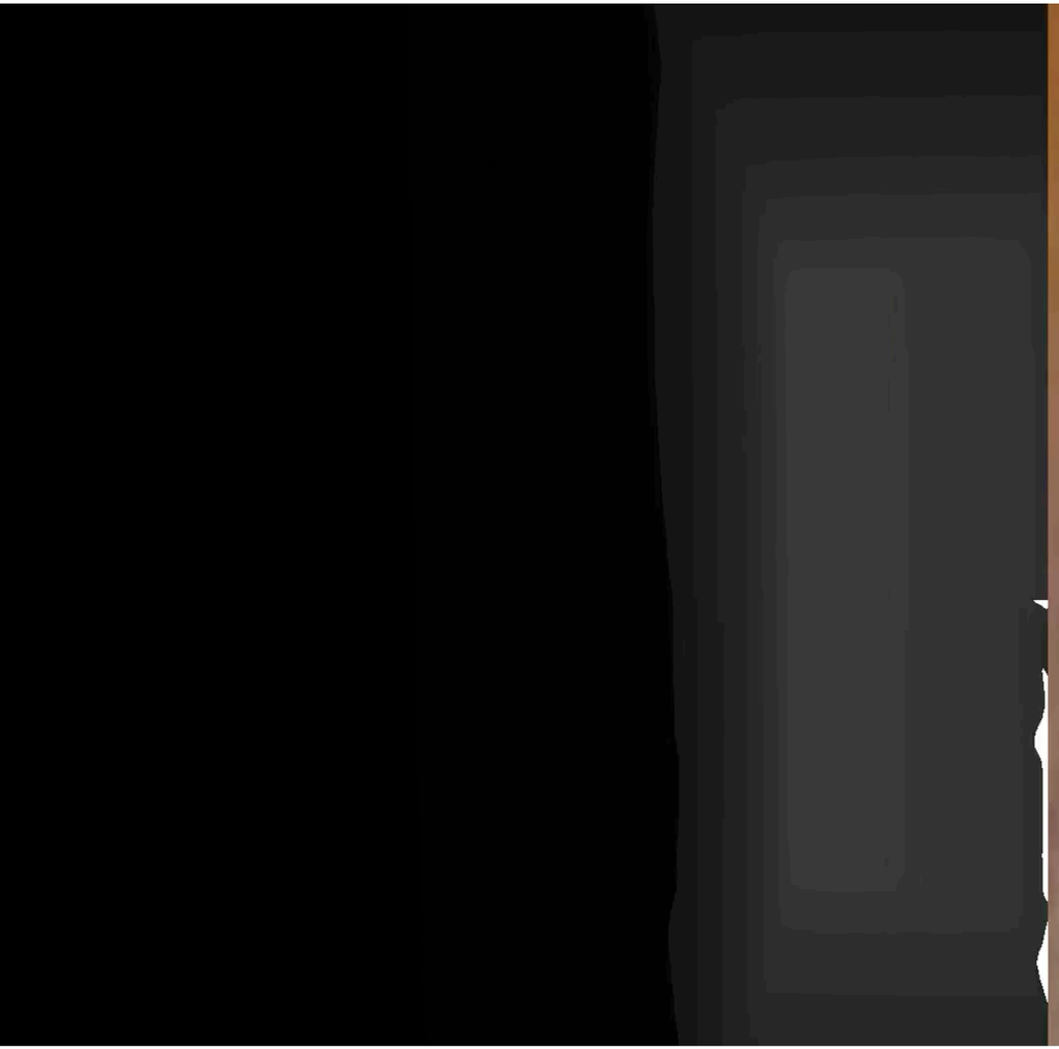


RGB and GPS+Compass

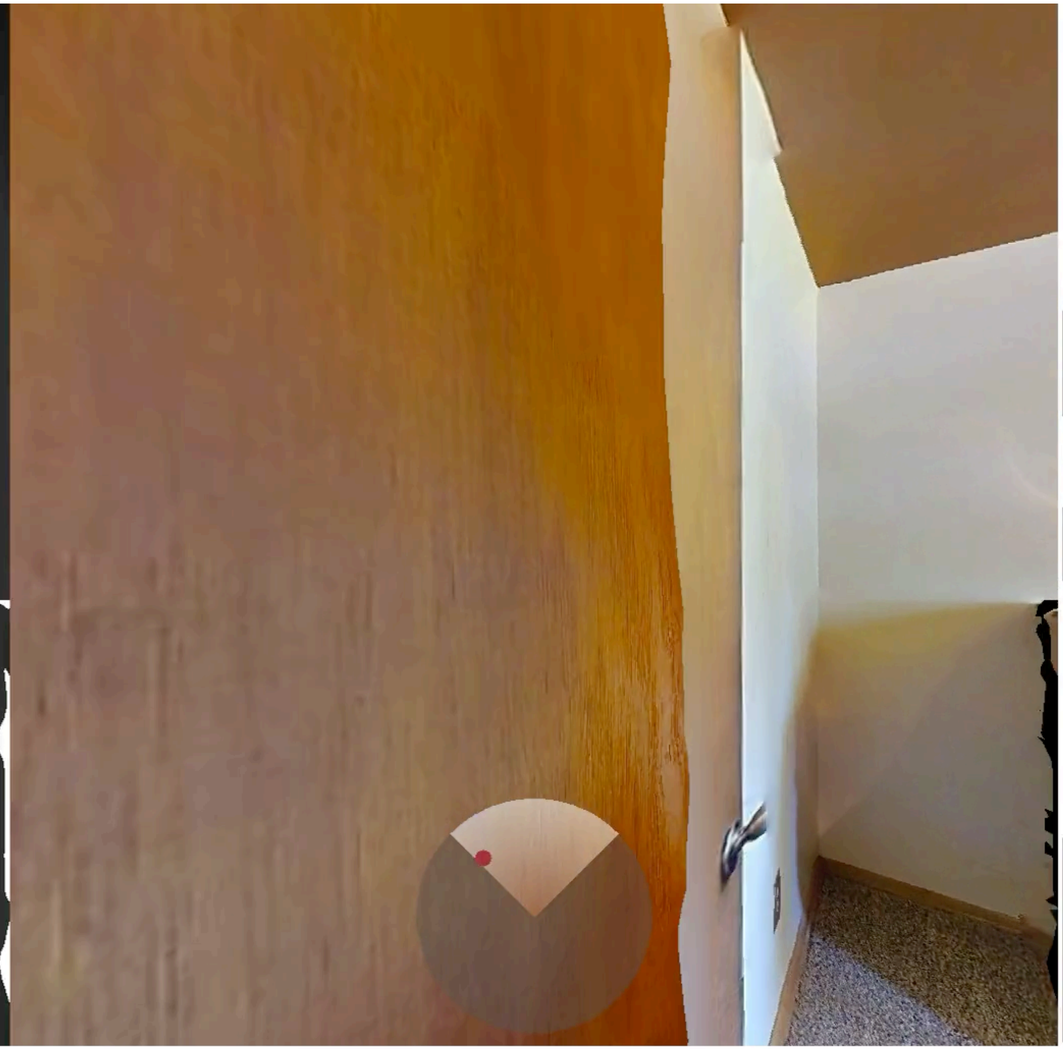


Top Down Map

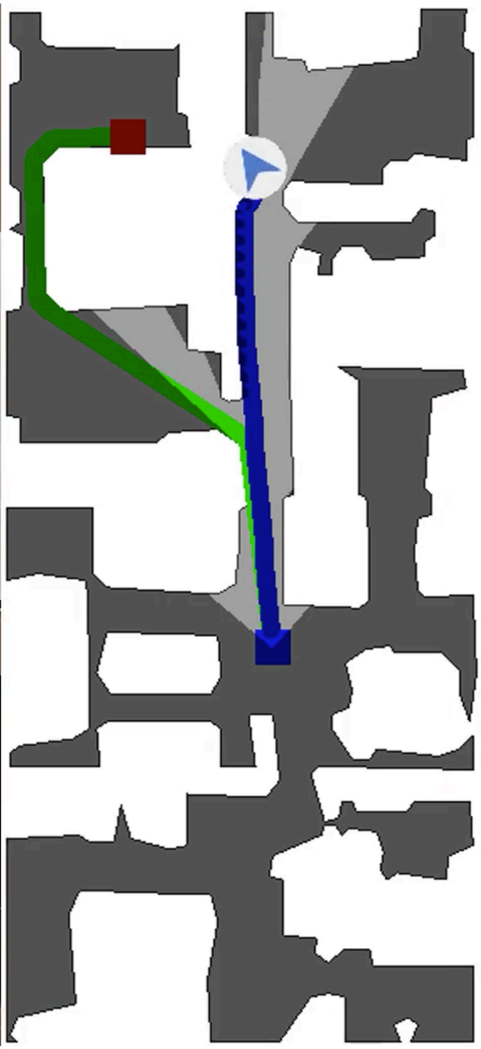
Slide credit: Erik Wijmans



Depth



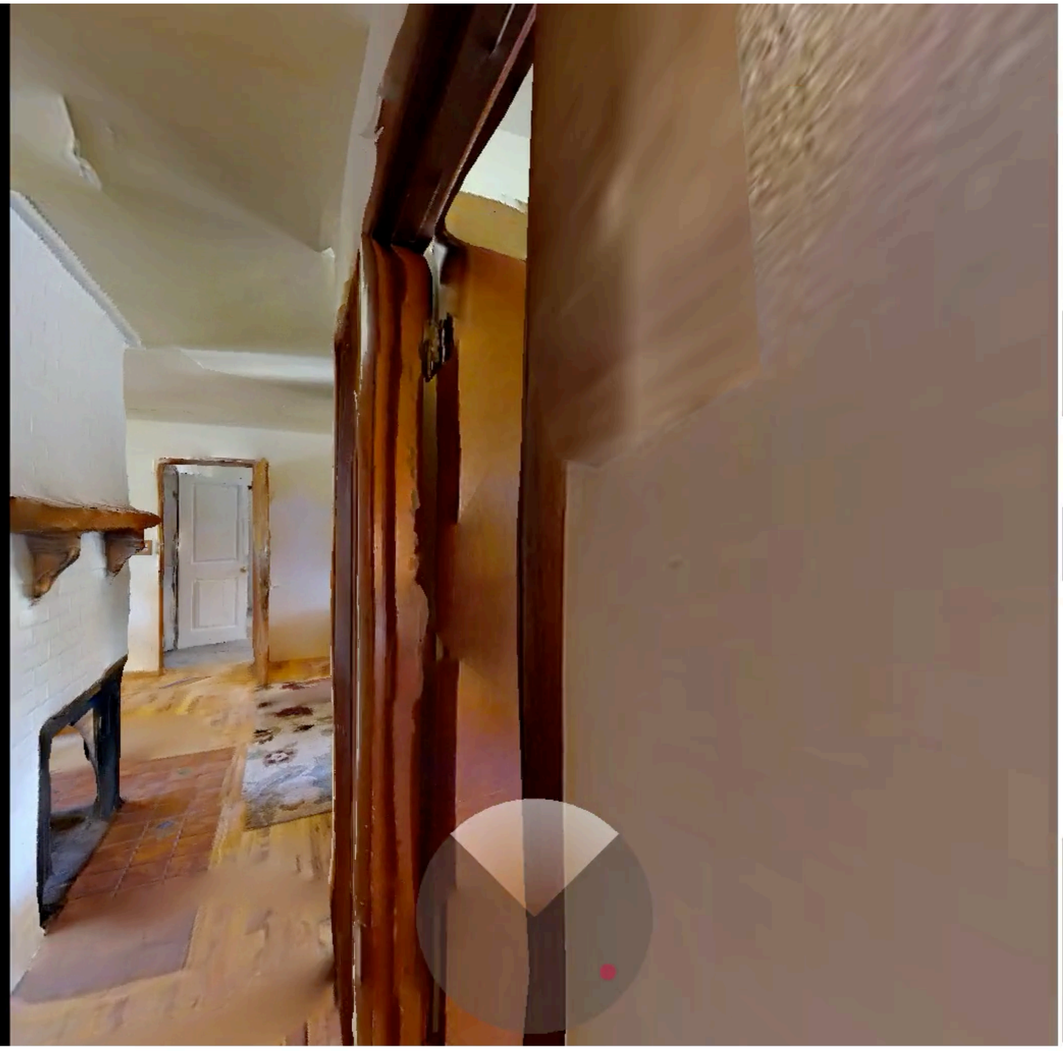
RGB and GPS+Compass



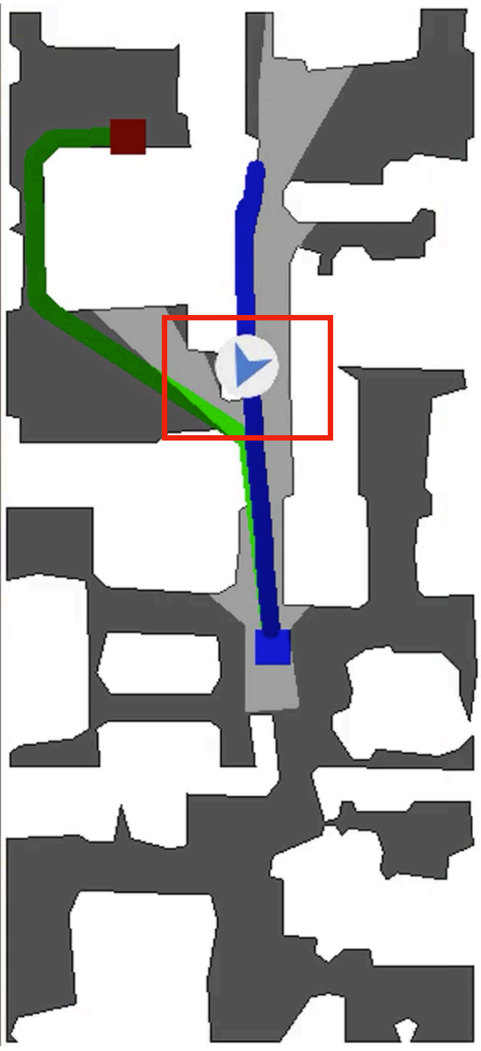
Top Down Map

Slide credit: Erik Wijmans

Depth



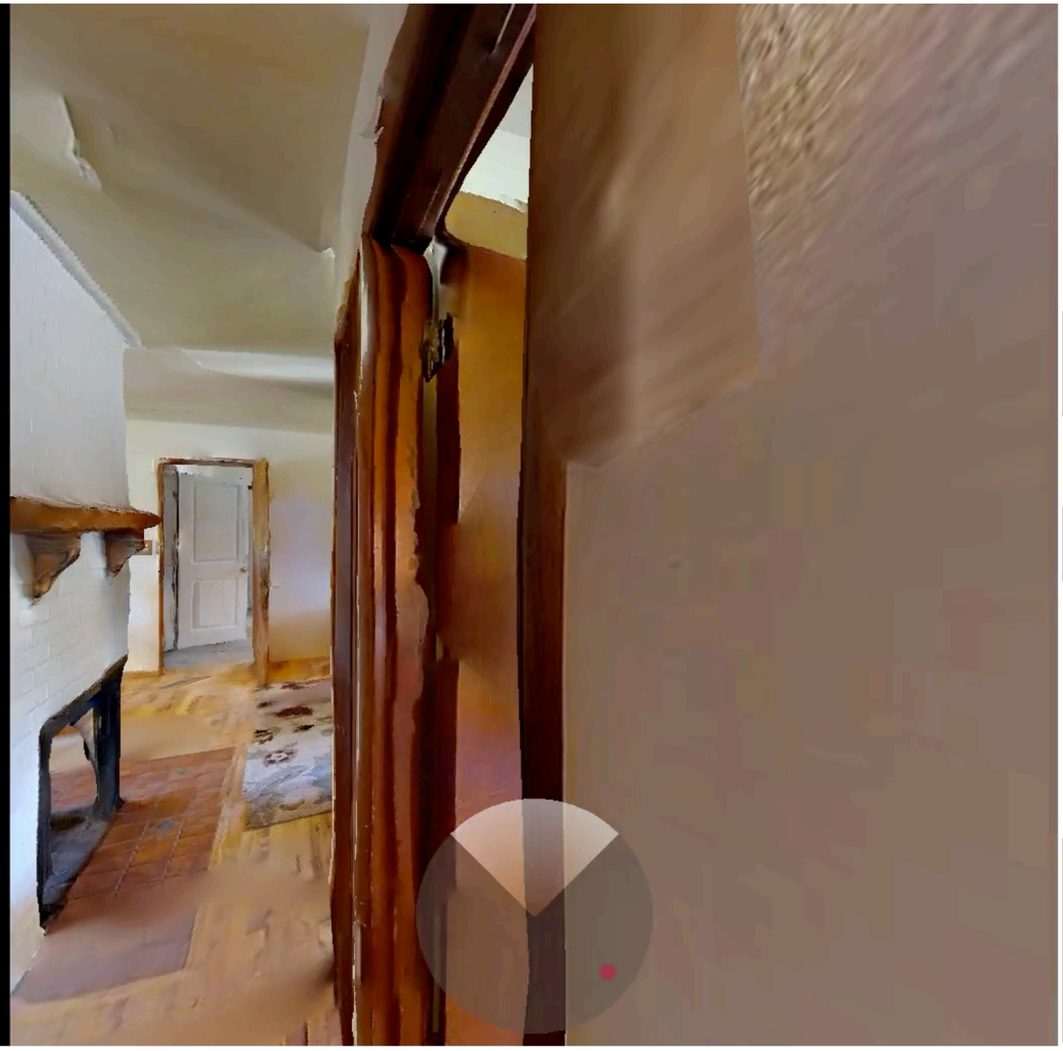
RGB and GPS+Compass



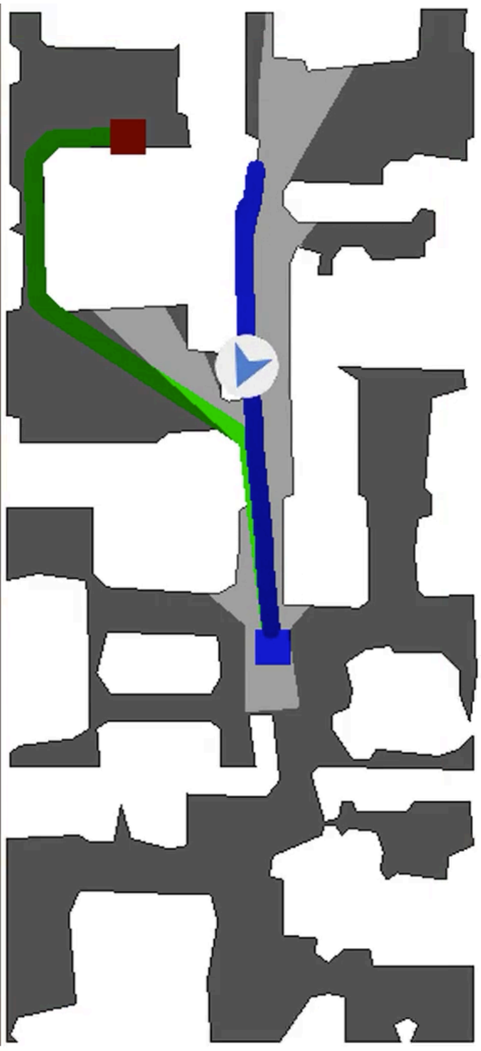
Top Down Map

Slide credit: Erik Wijmans

Depth



RGB and GPS+Compass



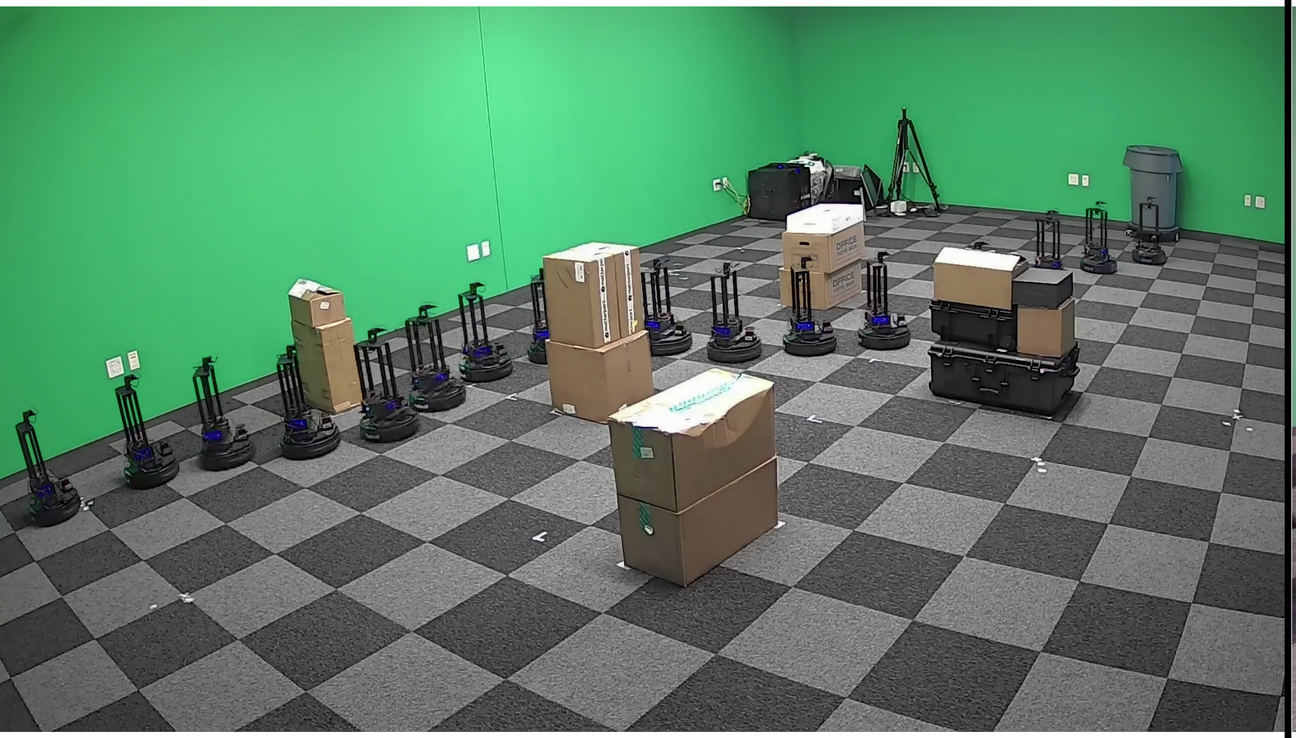
Top Down Map

Slide credit: Erik Wijmans

Does progress in simulation translate to progress on real robots?

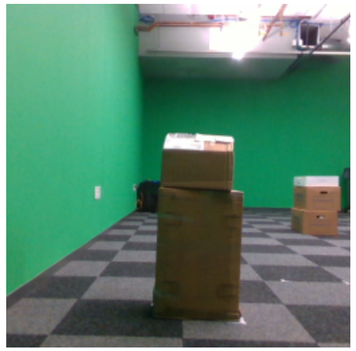
Virtualizing reality

Real World

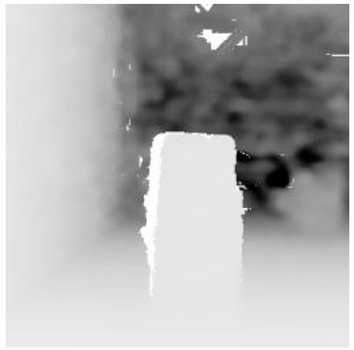


Robot

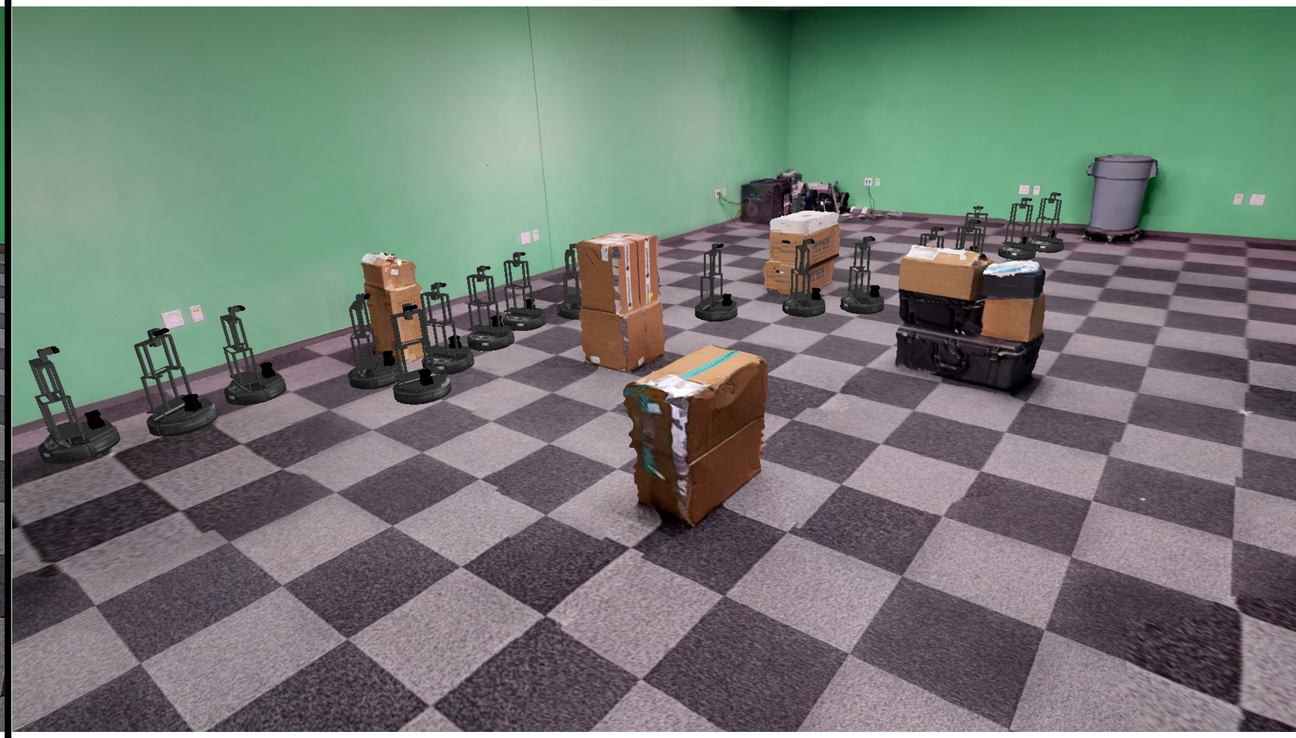
RGB



Depth



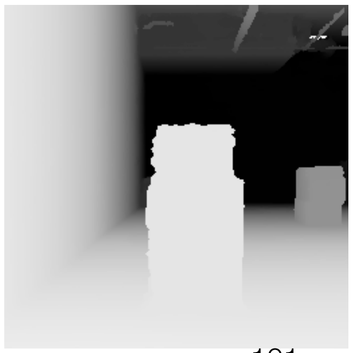
Simulation



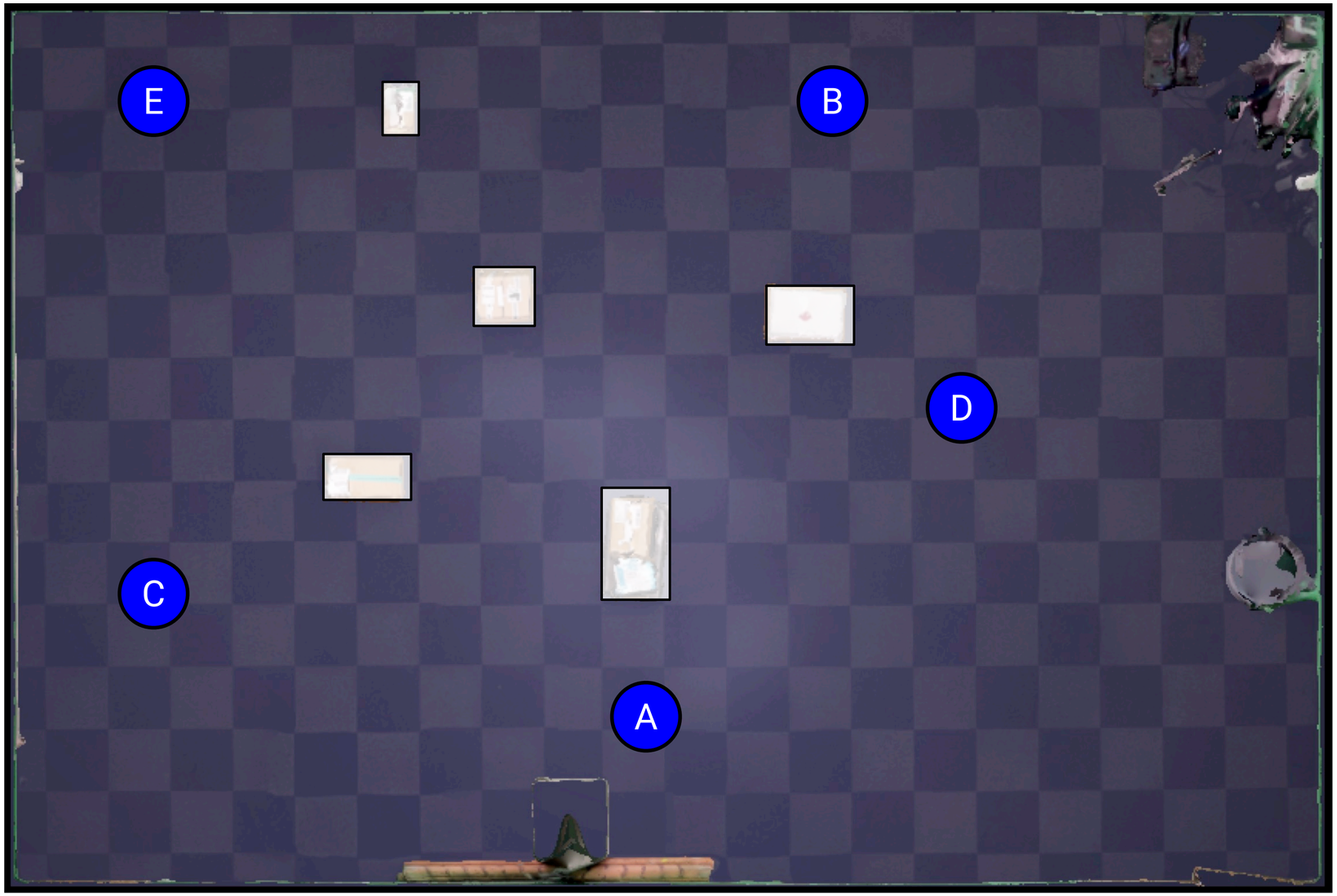
Robot

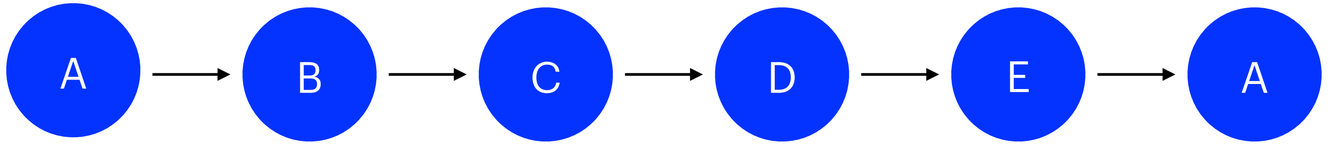
RGB

Depth



LoCoBot





We compare the performance of RL models across 810 identical experiments in parallel across simulation and reality

Simulation

Real World

Kadian*, Truong* et al. RA-L + IROS '20

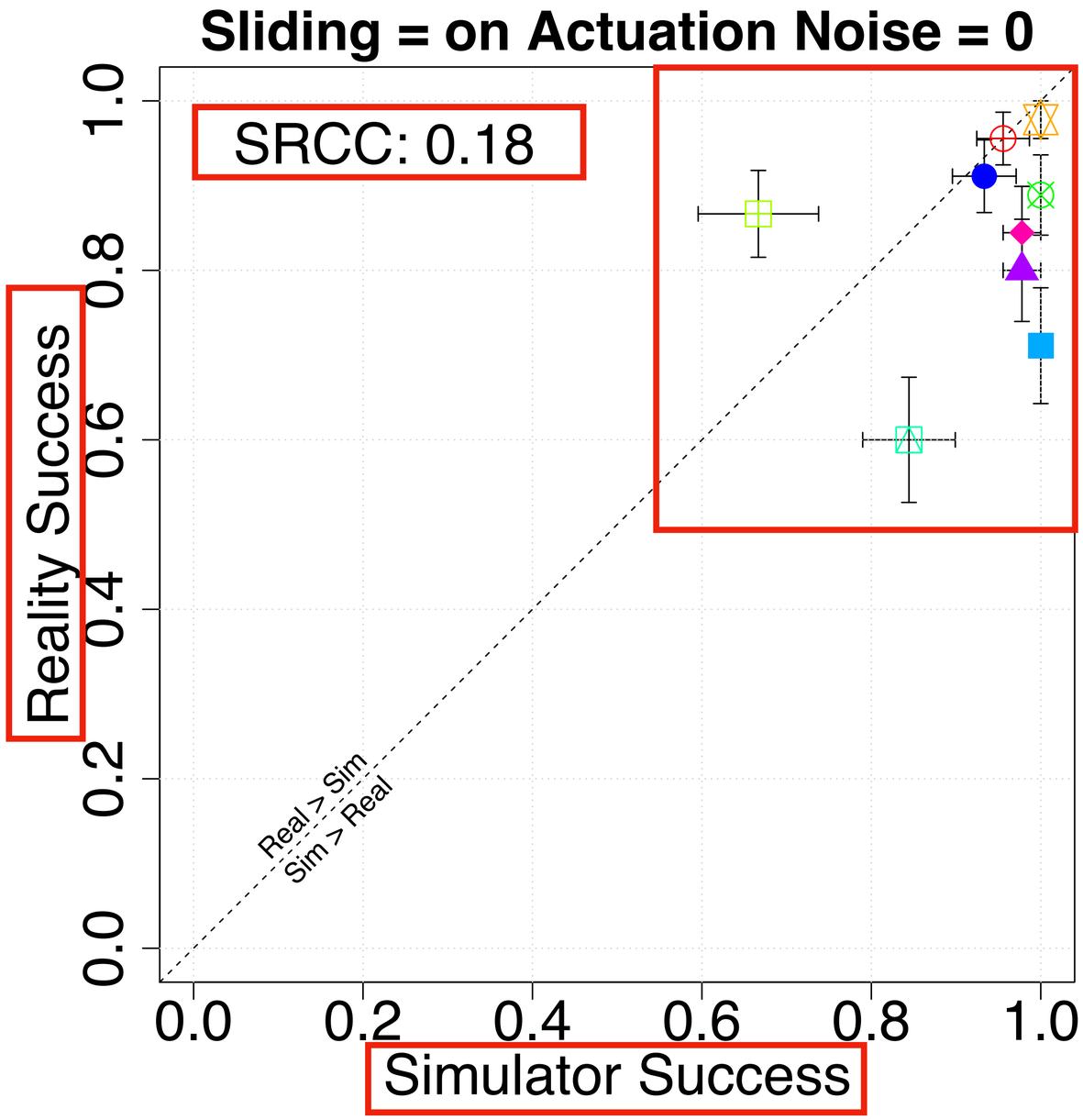
We compare the performance of RL models across 810 identical experiments in parallel across simulation and reality

Simulation

Real World

Kadian*, Truong* et al. RA-L + IROS '20

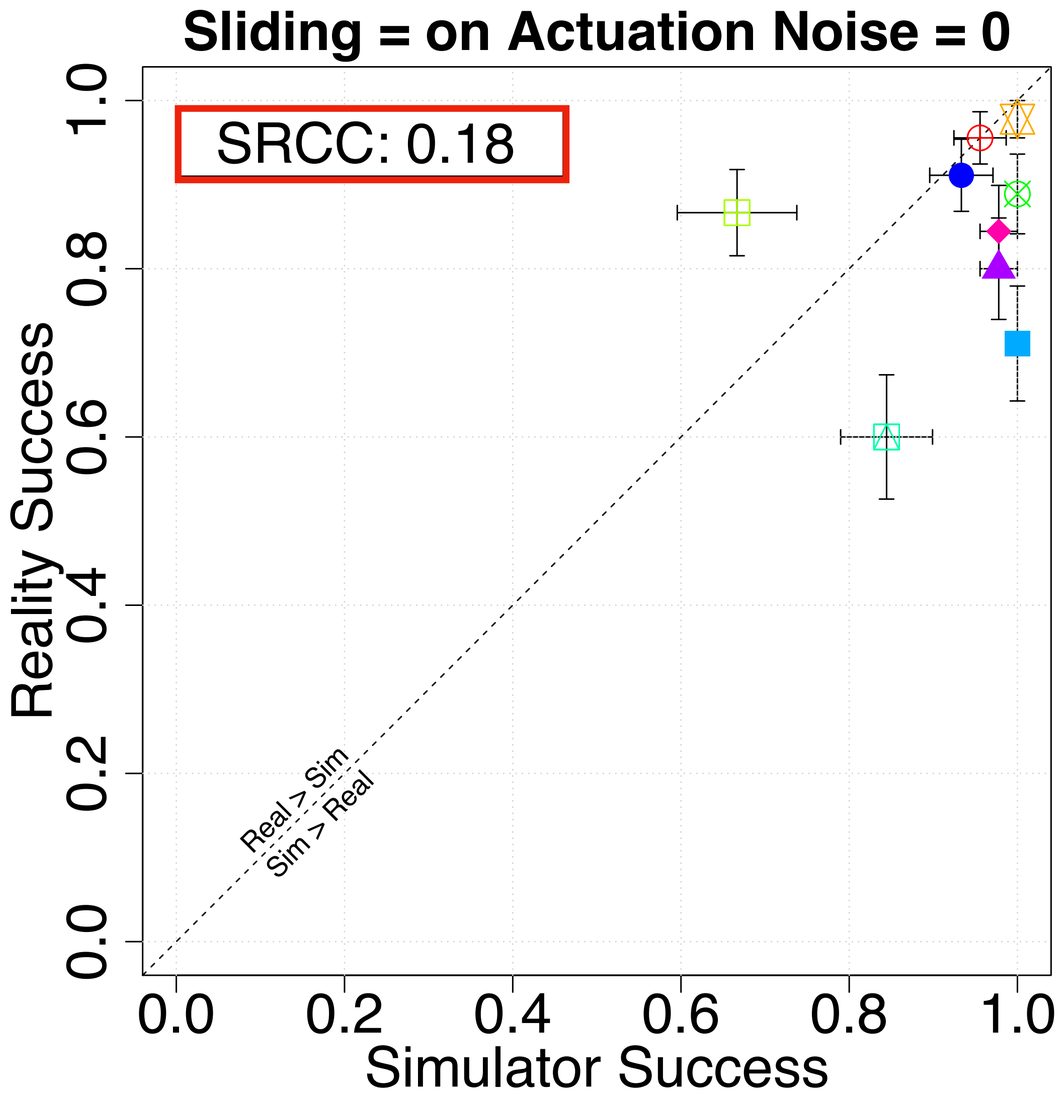
Sim-vs-Real Correlation Coefficient (SRCC)



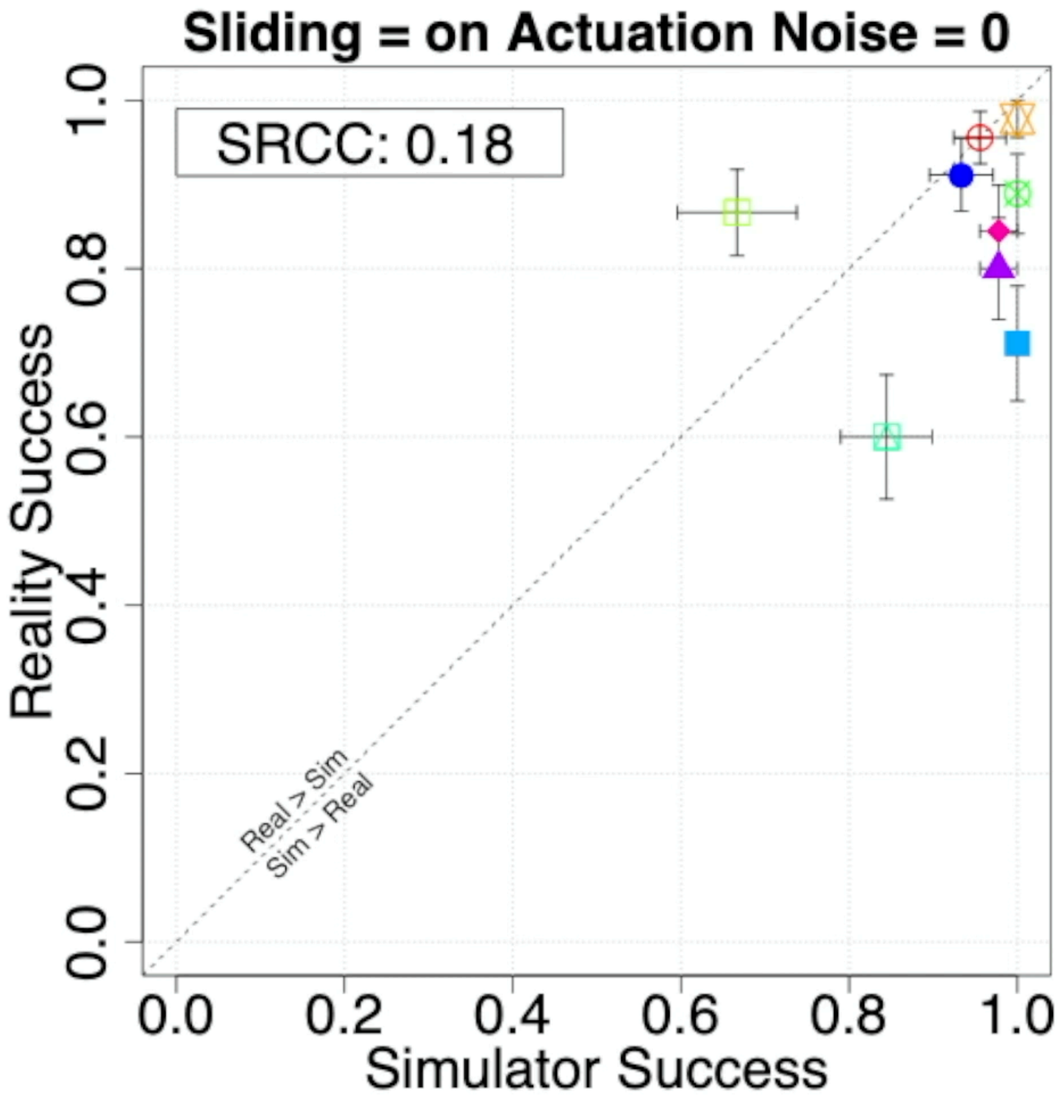
$$r_{xy} = \frac{\sum_{i=1}^{n}(x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum_{i=1}^{n}(x_i - \bar{x})^2}\sqrt{\sum_{i=1}^{n}(y_i - \bar{y})^2}}$$

- Depth – Train(sliding=off, noise=0.5)
- Depth – Train(sliding=off, noise=1.0)
- Predicted depth – Train(sliding=off, noise=0.5)
- Predicted depth – Train(sliding=off, noise=1.0)
- RGB – Train(sliding=off, noise=0.5)
- RGB – Train(sliding=off, noise=1.0)
- Depth – Train(sliding=on, noise=0)
- Predicted depth – Train(sliding=on, noise=0)
- RGB – Train(sliding=on, noise=0)

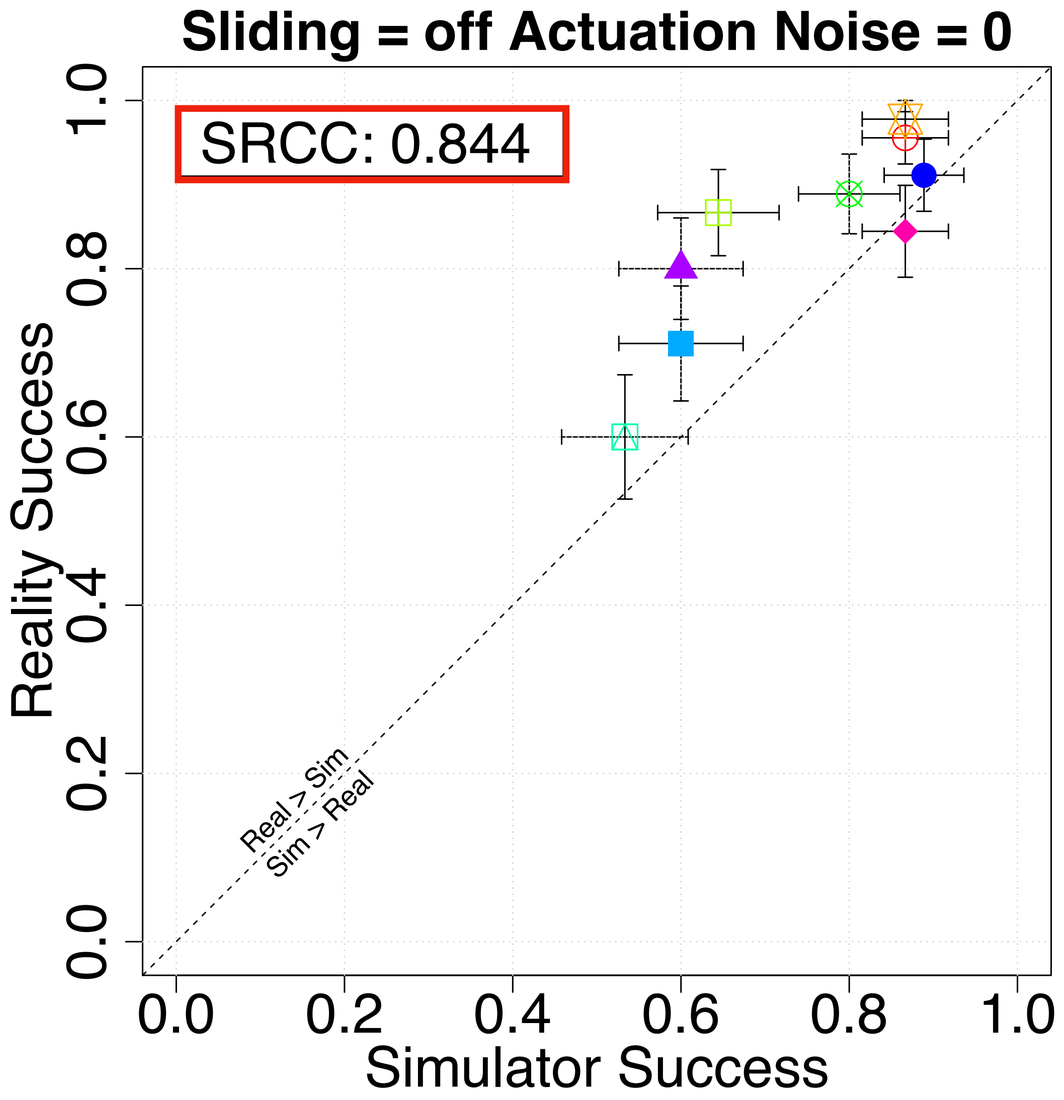
SRCC with original 2019 AI Habitat Challenge simulation settings



Search over simulation parameters to **optimize** SRCC



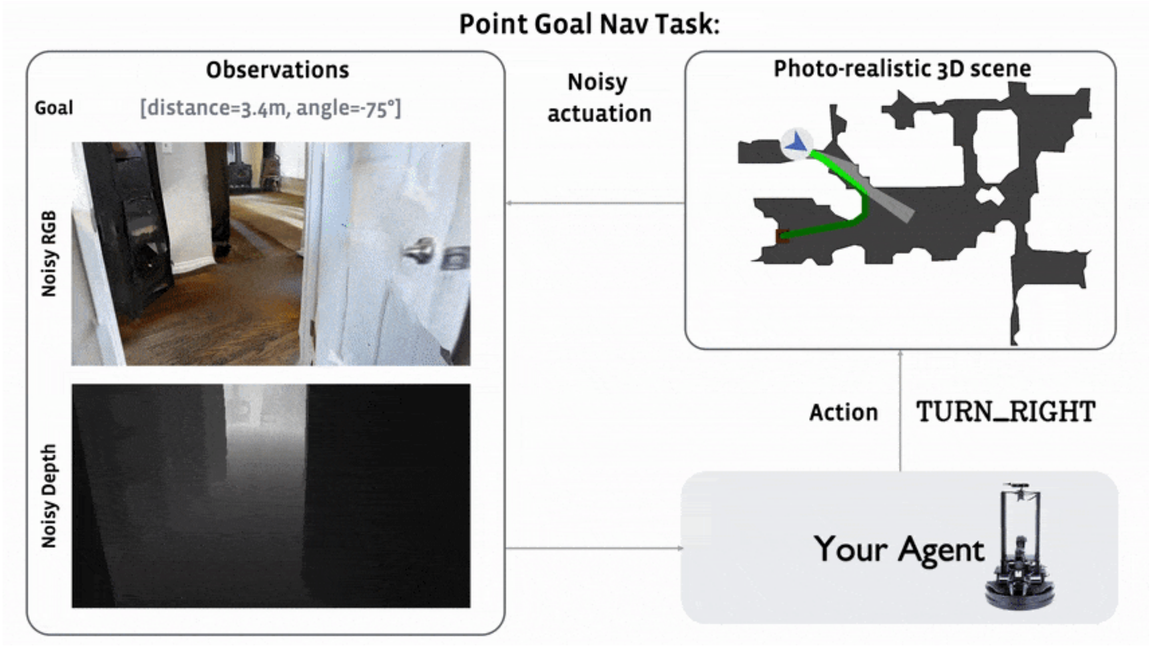
SRCC with **optimized** simulation settings

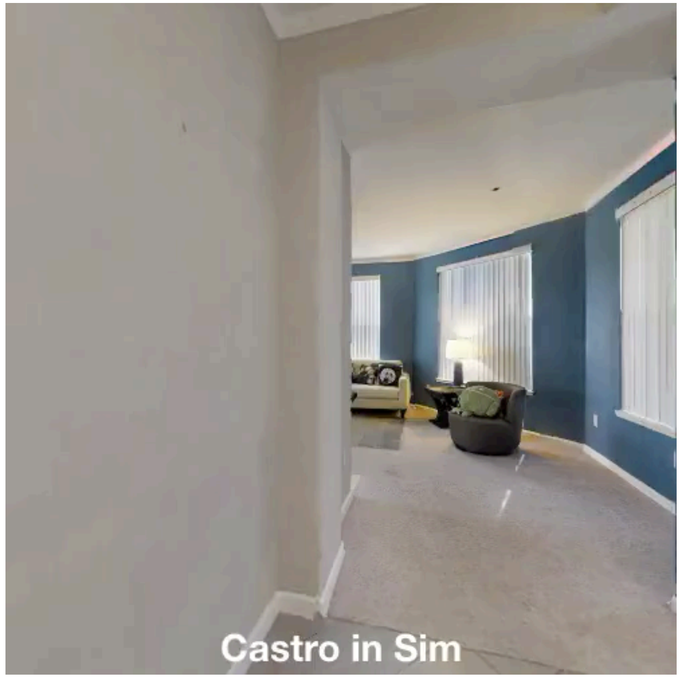


Embodied AI Workshop

CVPR 2020

CVPR 2021



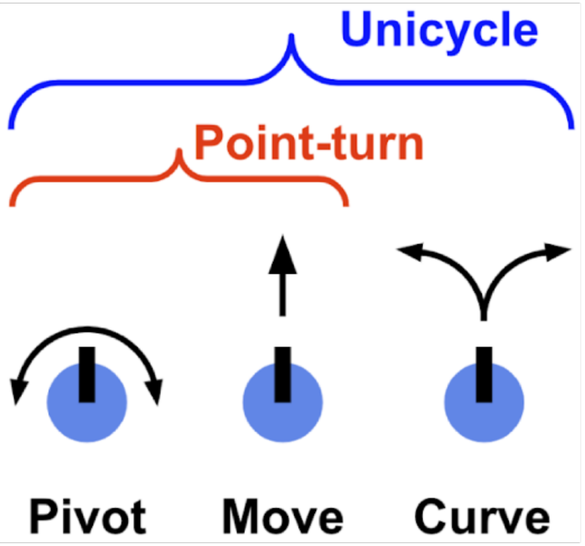


[9] Xia et al. 2018

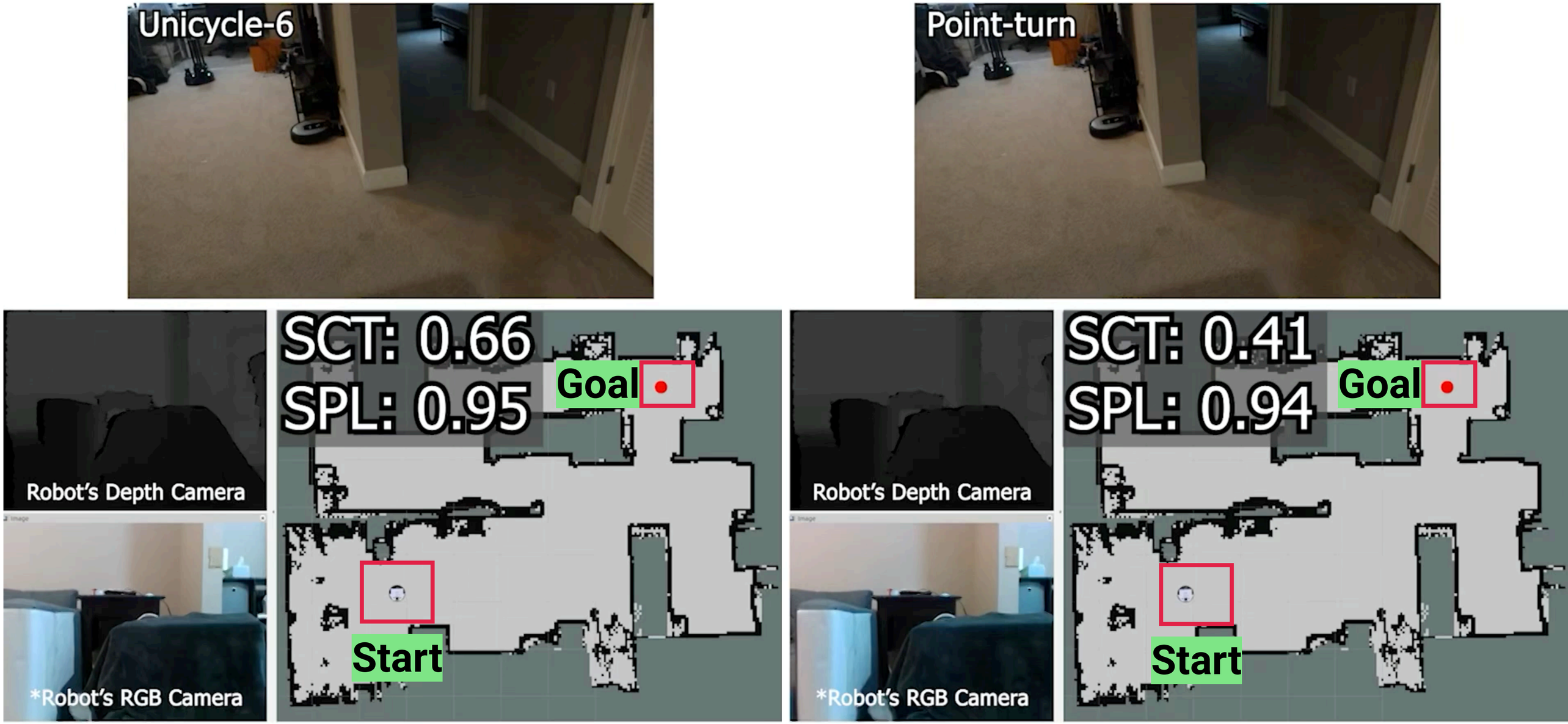

Goal
20x

Leveraging LoCoBot's dynamics

- Real robots have more complex dynamics
- Most real robots (e.g., LoCoBot, Fetch, TurtleBot) move and turn simultaneously
  - i.e., Unicycle-cart dynamics
- Unicycle navigates faster than point-turn under identical maximum velocities



Point-turn vs. Unicycle dynamics

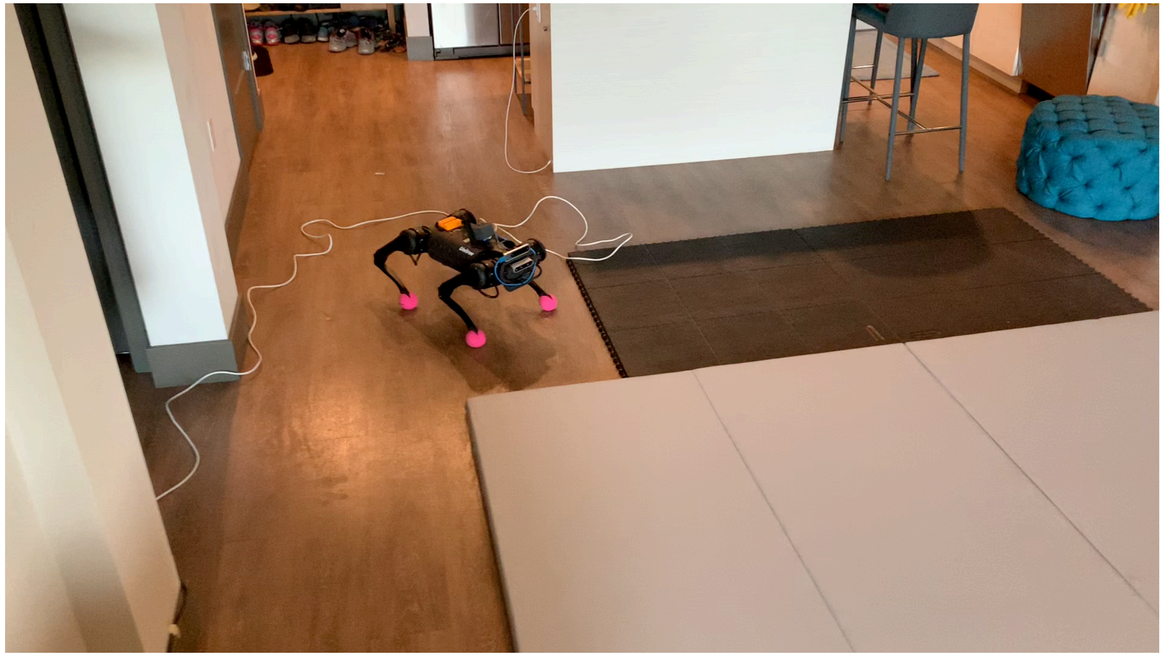


Yokoyama et al. IROS '21

Outline

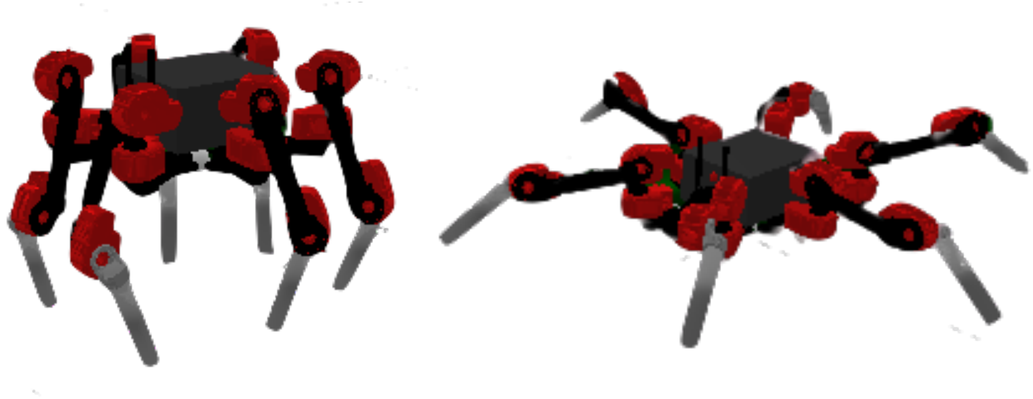
- Policy Gradient
  - Trust Region Policy Optimization (TRPO)
  - Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation
  - Sim2Real Transfer
  - Robot2Robot Transfer

Legged robots are well-suited for navigating indoor environments



Robot-to-Robot generalization

[15] Kumar et al. 2021



How can we learn a single navigation policy for multiple complex, legged robots?

How can we learn a single navigation policy for multiple complex, legged robots?

Reality

Simulation

A1

Train

How can we learn a single navigation policy for multiple complex, legged robots?

Reality

Simulation

A1

AlienGo

Train

How can we learn a single navigation policy for multiple complex, legged robots?

Reality

Simulation

A1 AlienGo Daisy

Train

How can we generalize to new, previously unseen robots?

A1 AlienGo Daisy

Train

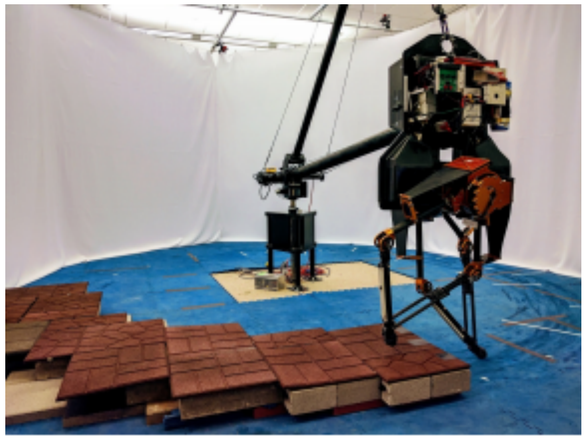
How can we generalize to new, previously unseen robots?

How can we generalize to new, previously unseen robots?

Reinforcement learning for navigation

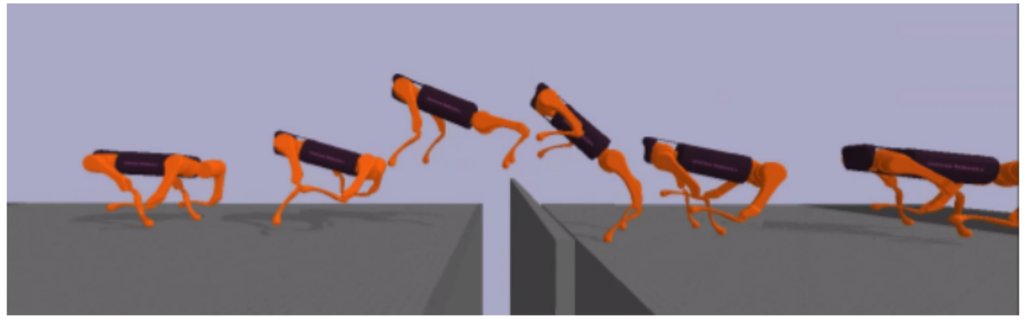
Blind locomotion

[16] Lee et al. 2020

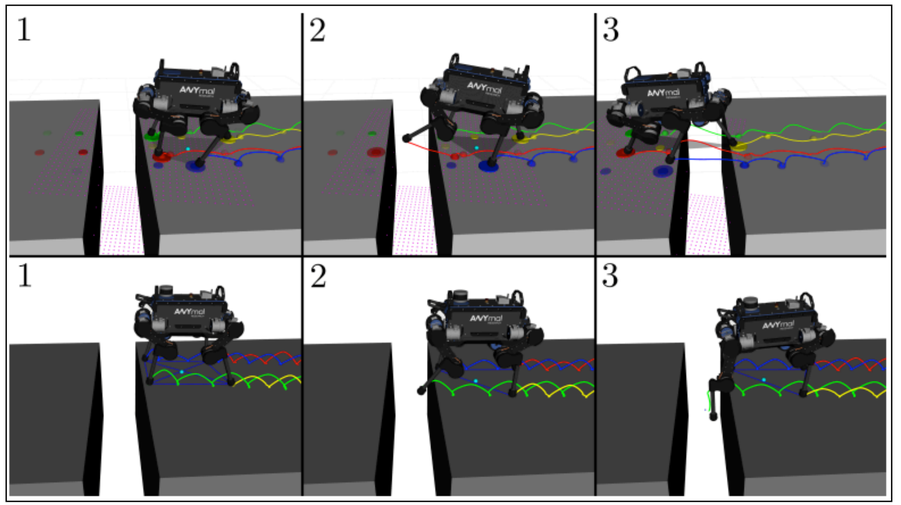


[17] Martin et al. 2017

Expensive sensors (LiDAR)



[18] Iscen et al. 2021

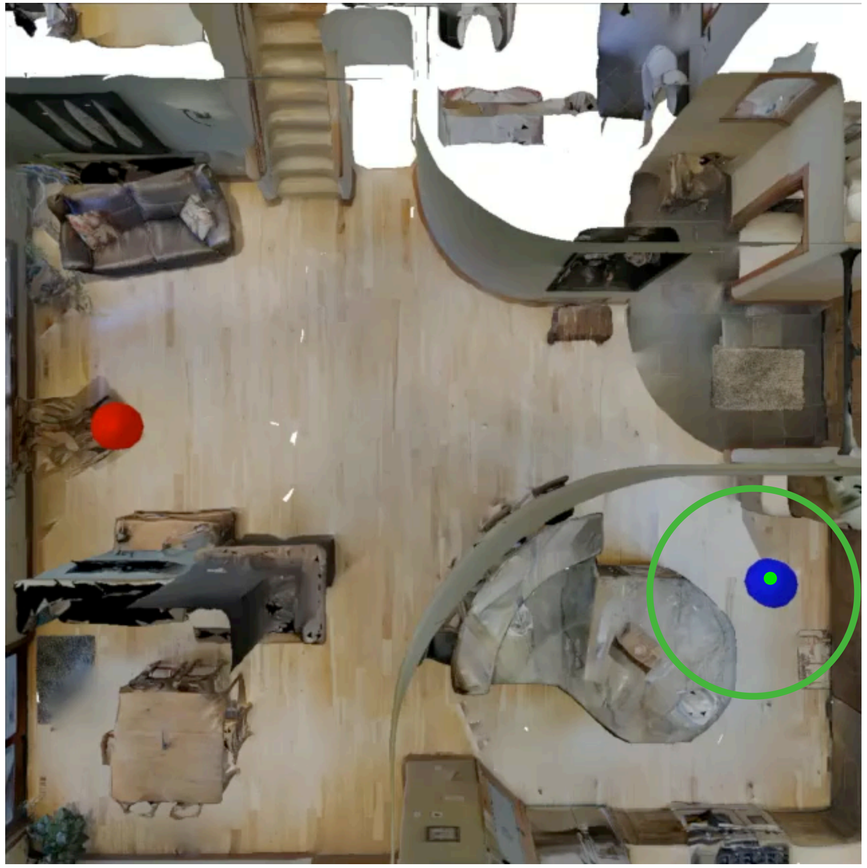


[19] Tsounis, Alge et al. 2020

Idealized spherical agent policy

Egocentric RGB

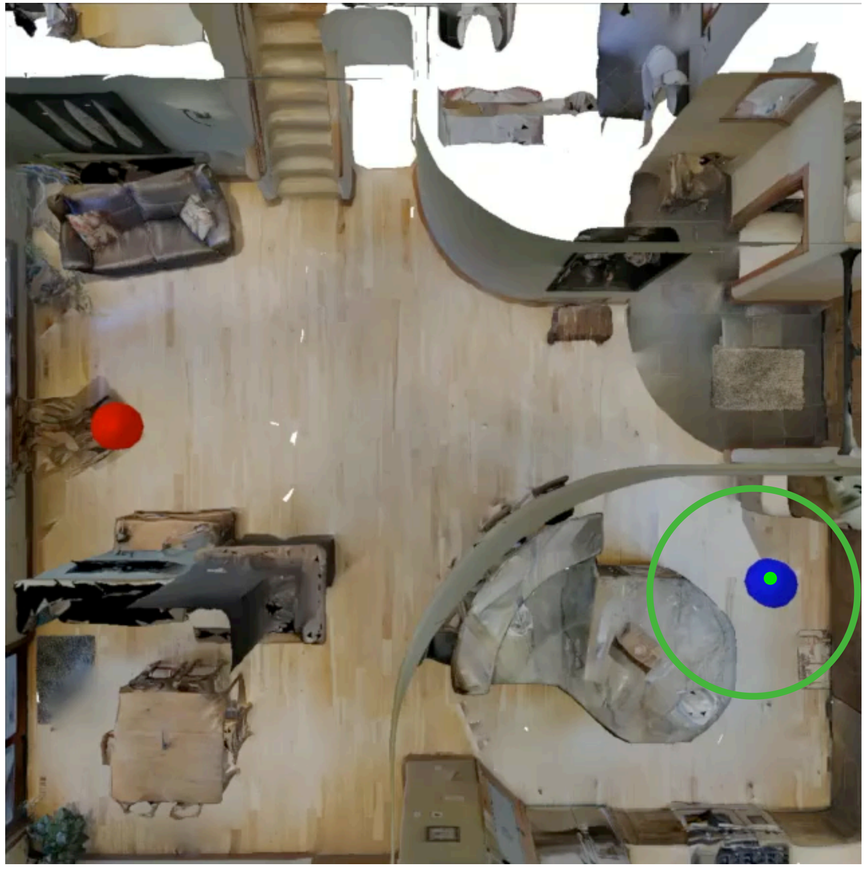
Egocentric Depth



Idealized spherical agent policy

Egocentric RGB

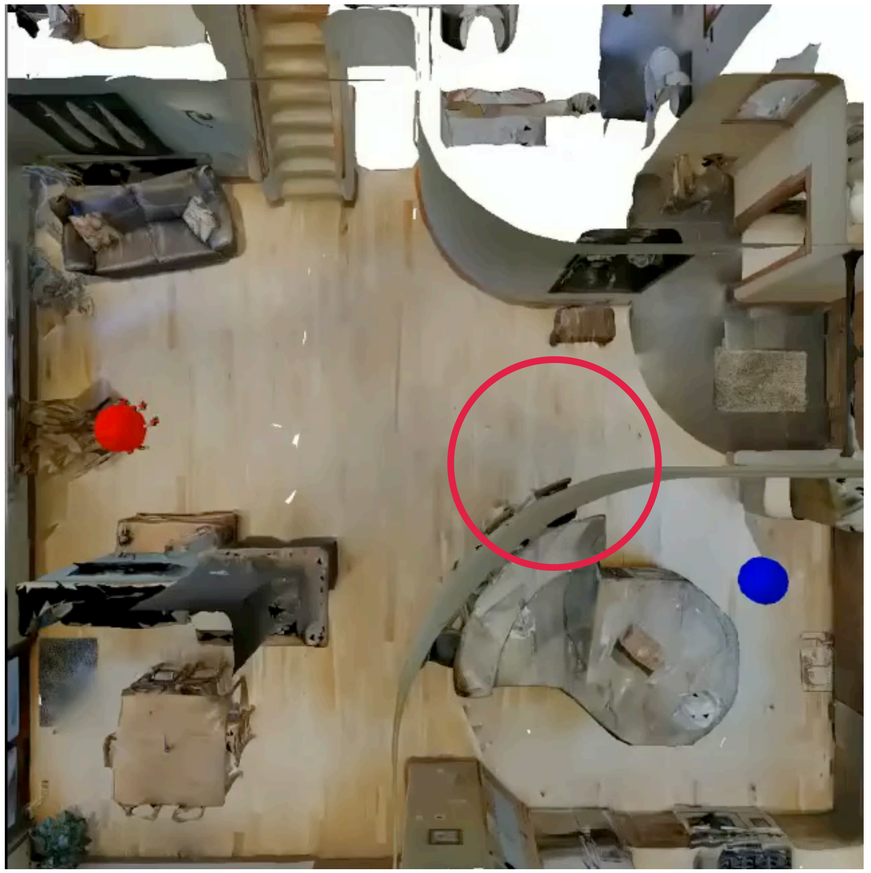
Egocentric Depth



Transferred to Daisy

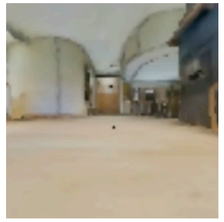
Egocentric RGB

Egocentric Depth



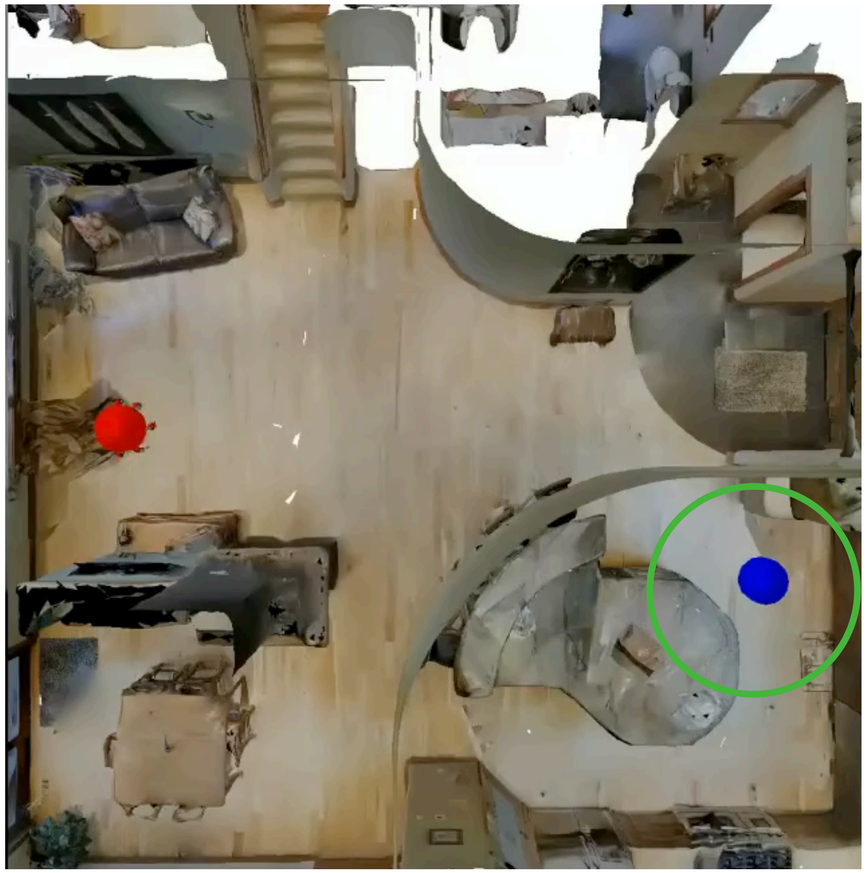
Daisy gets stuck around an obstacle

Policy trained directly on Daisy

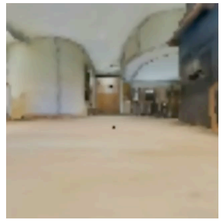


Egocentric RGB

Egocentric Depth

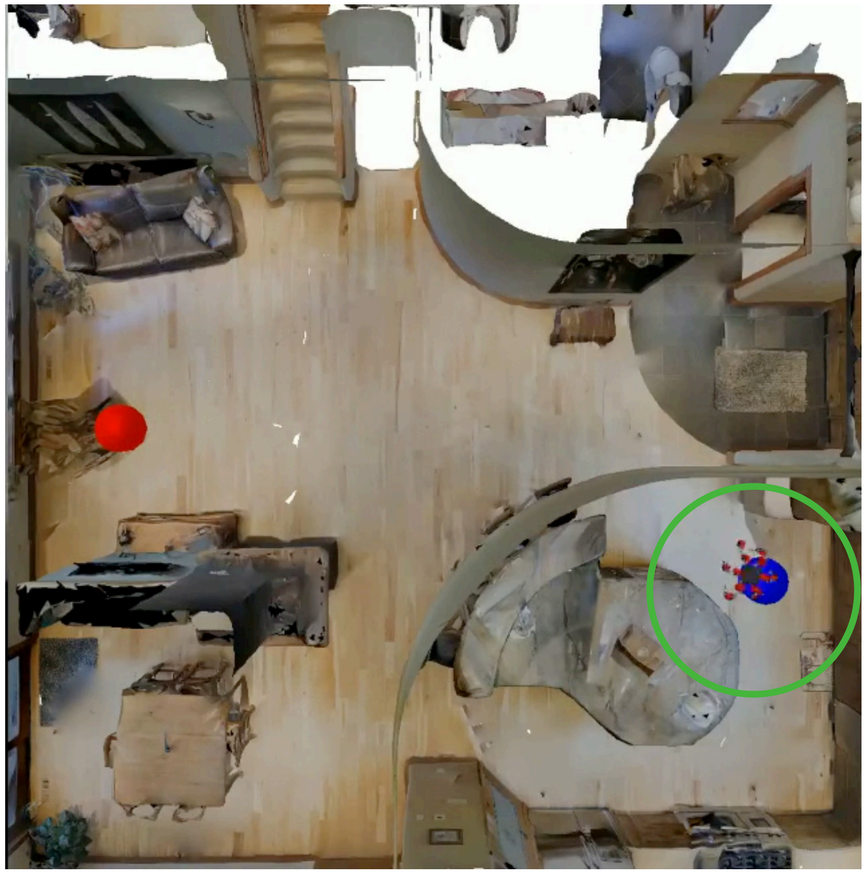


Policy trained directly on Daisy

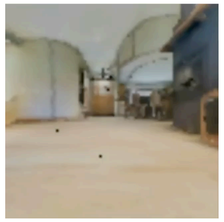


Egocentric RGB

Egocentric Depth

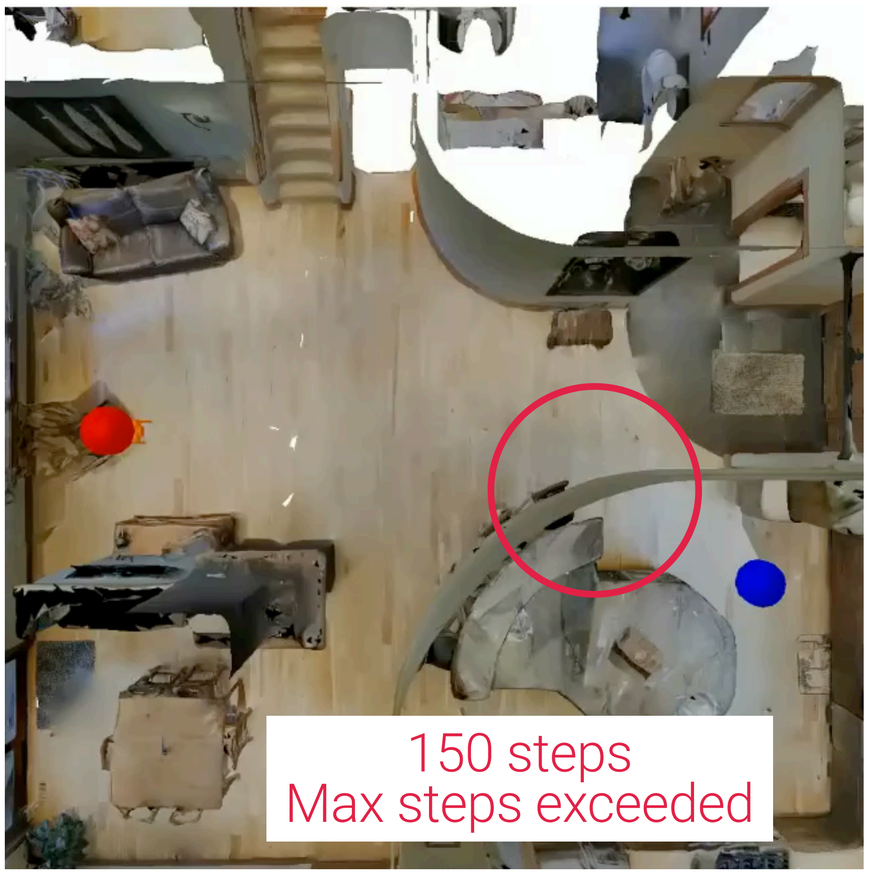


Transferred to AlienGo



Egocentric RGB

Egocentric Depth

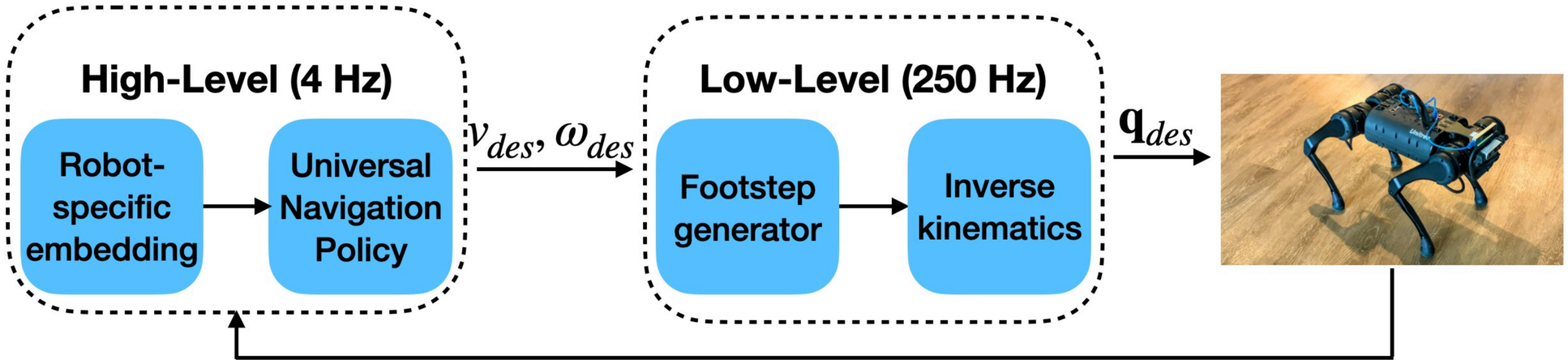


Fails to transfer to a new robot

Dynamics-aware hierarchical visual navigation policy

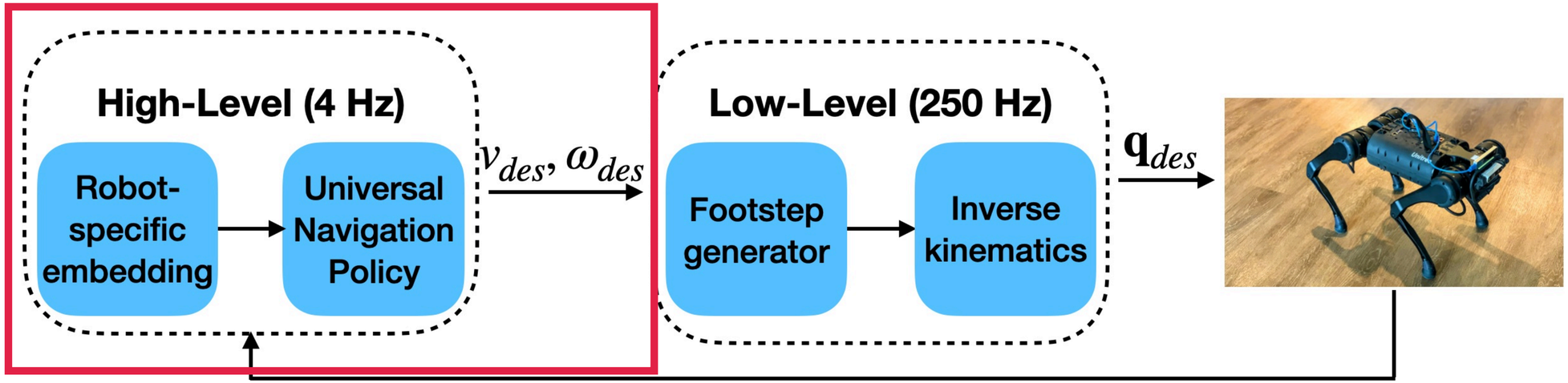
Dynamics-aware hierarchical visual navigation policy

1) Hierarchical navigation policy:



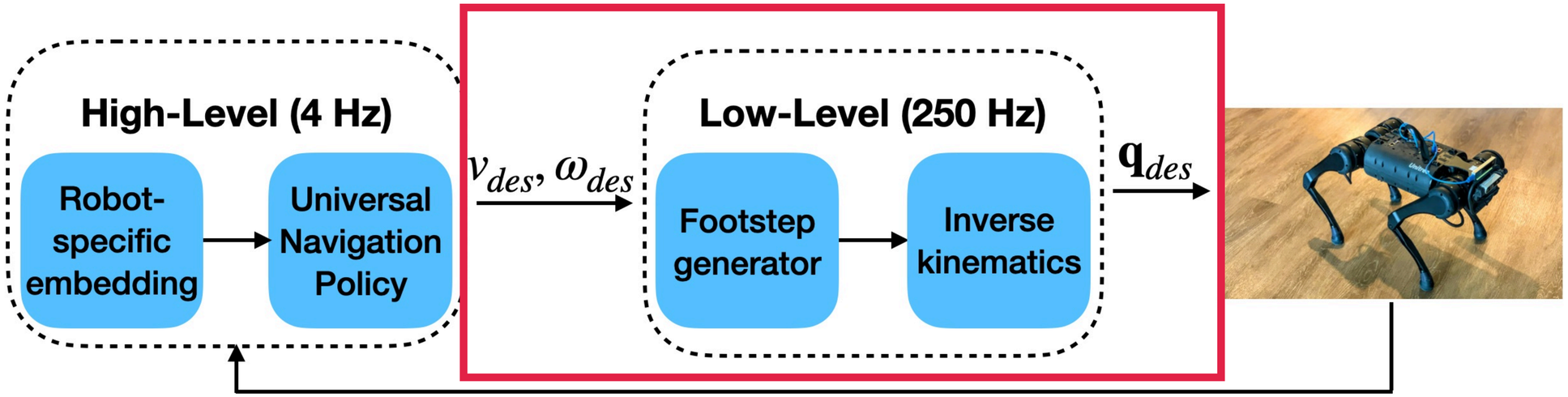
Dynamics-aware hierarchical visual navigation policy

1) Hierarchical navigation policy:
   — High-level policy that reasons about the center of mass motion

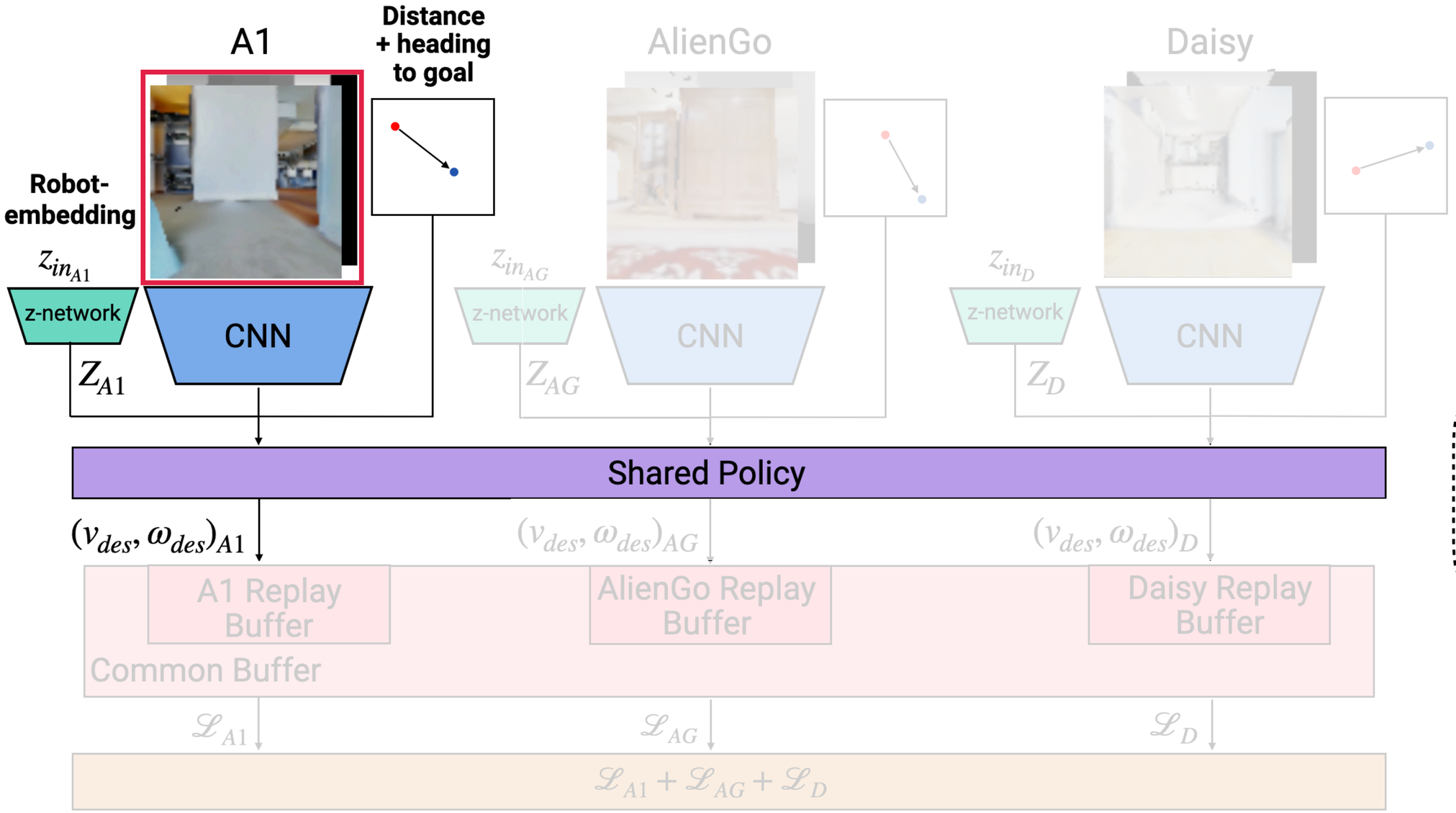


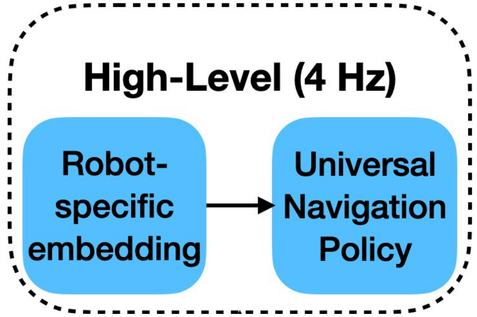
Dynamics-aware hierarchical visual navigation policy

1) Hierarchical navigation policy:
   - High-level policy that reasons about the center of mass motion
   - Low-level policy that converts high-level commands into desired footsteps

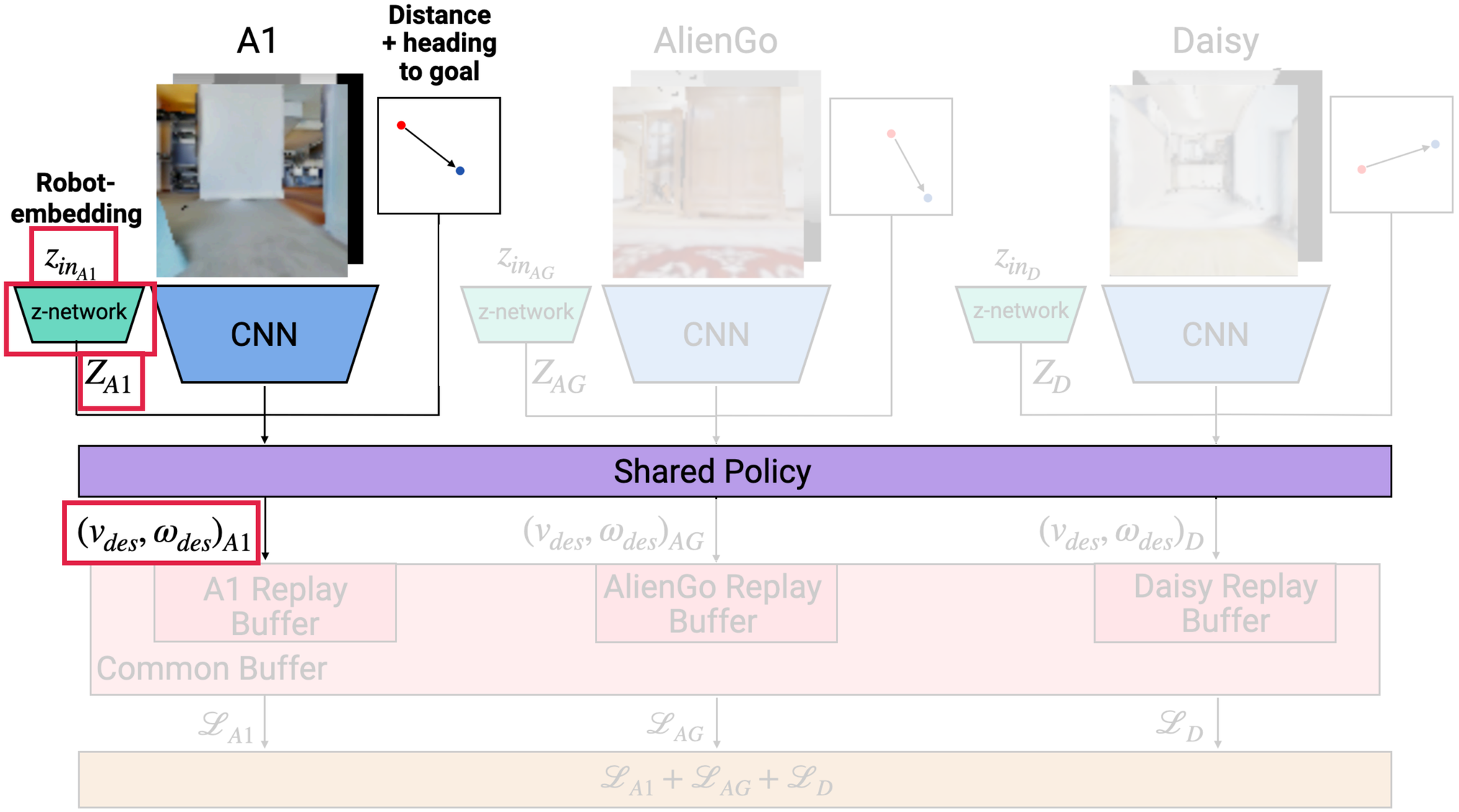


2) Learn a universal navigation policy across multiple robots
— Learns robot-specific embeddings (z)



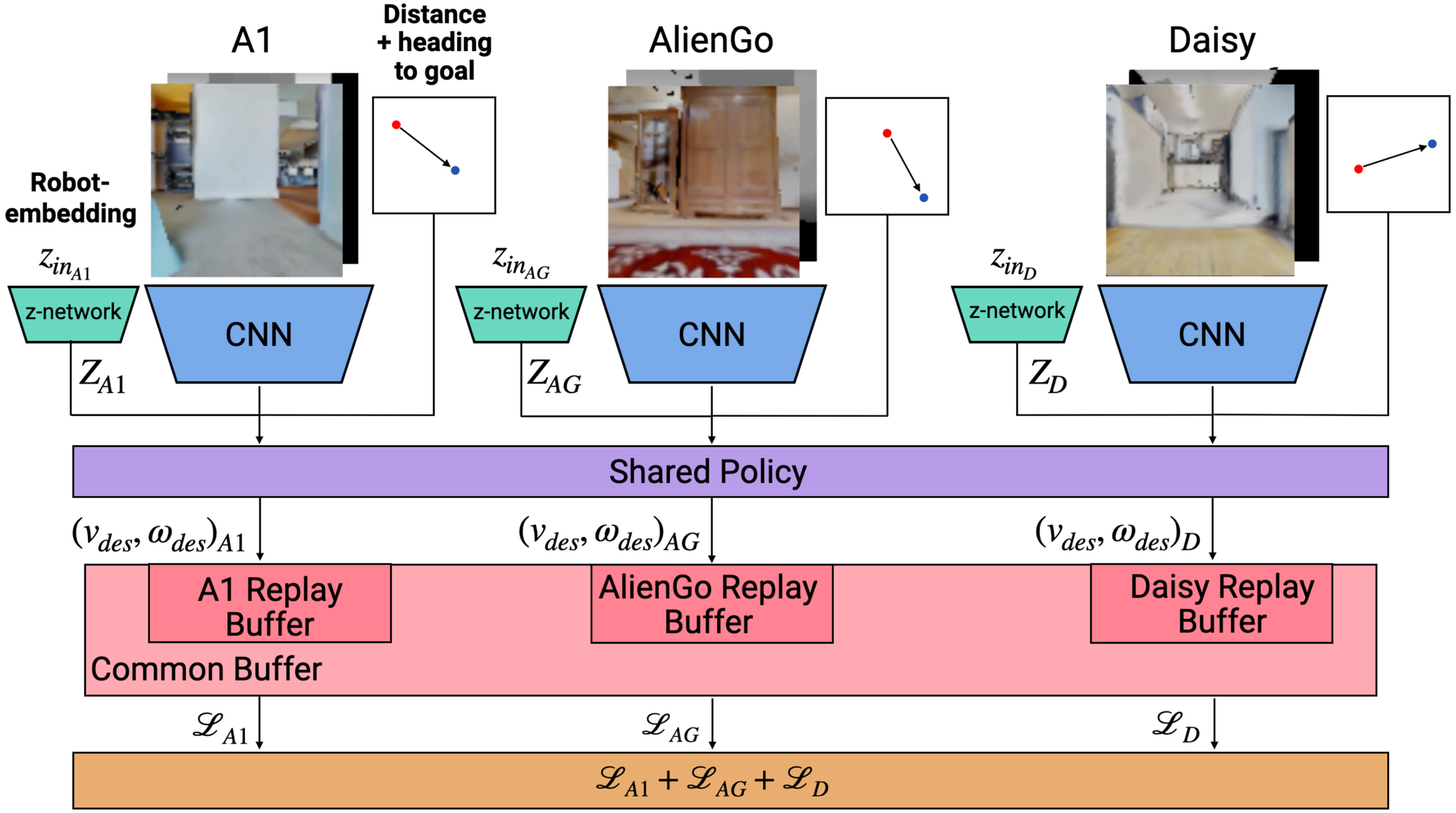


2) Learn a universal navigation policy across multiple robots
— Learns robot-specific embeddings (z)



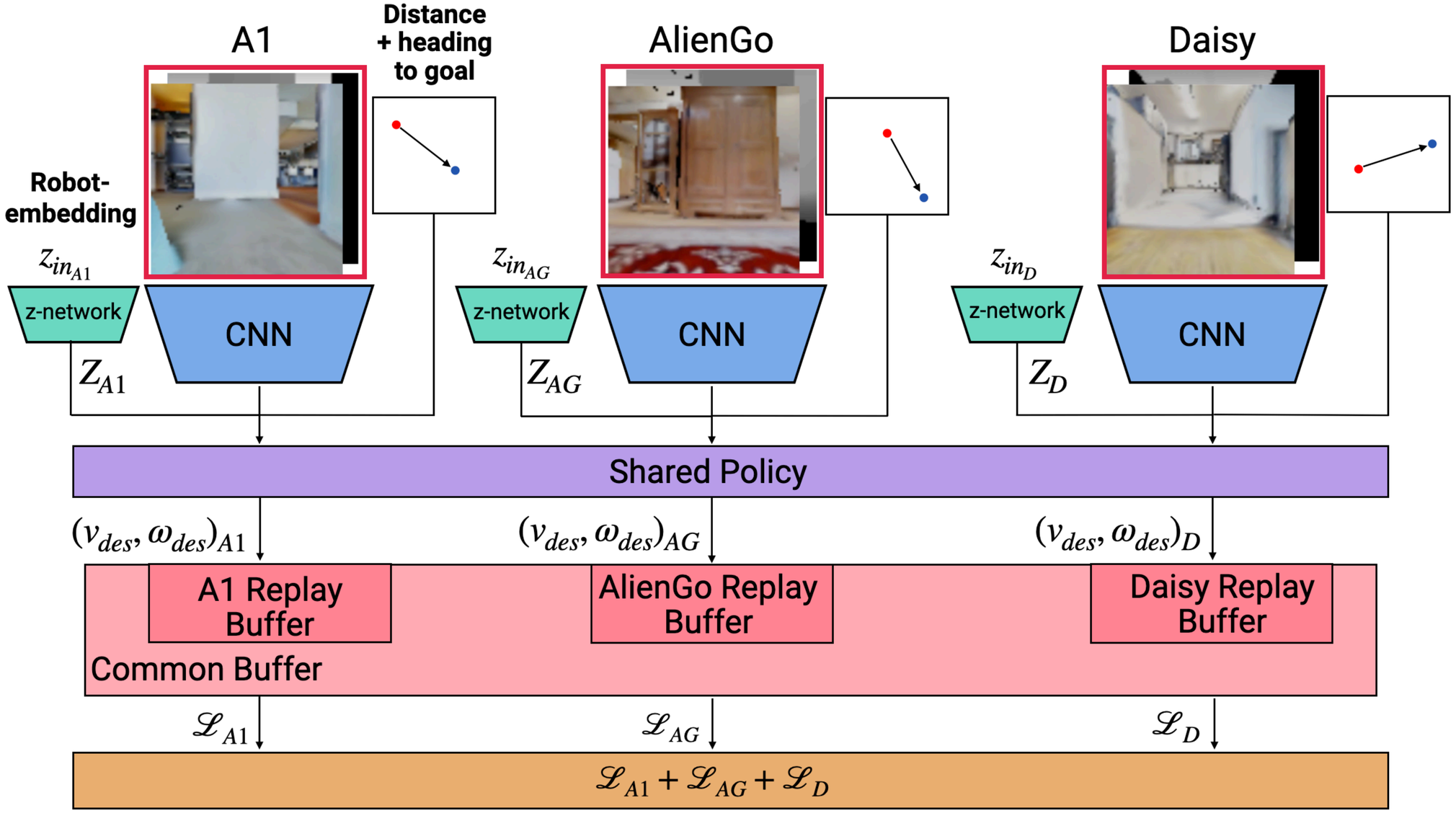
2) Learn a universal navigation policy across multiple robots

- Learns robot-specific embeddings (z)
- Shares data across multiple robots during learning



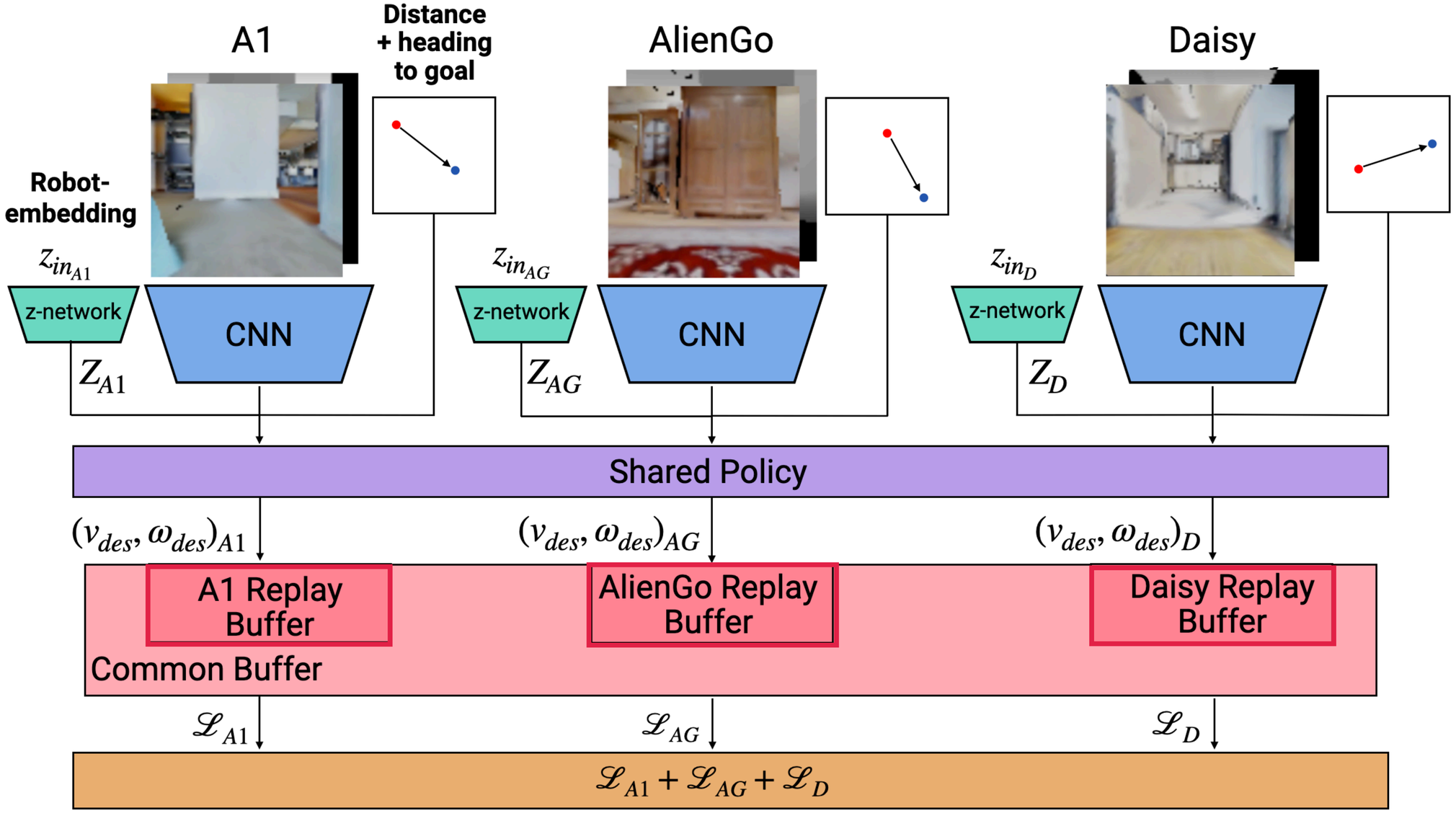
2) Learn a universal navigation policy across multiple robots

- Learns robot-specific embeddings (z)
- Shares data across multiple robots during learning



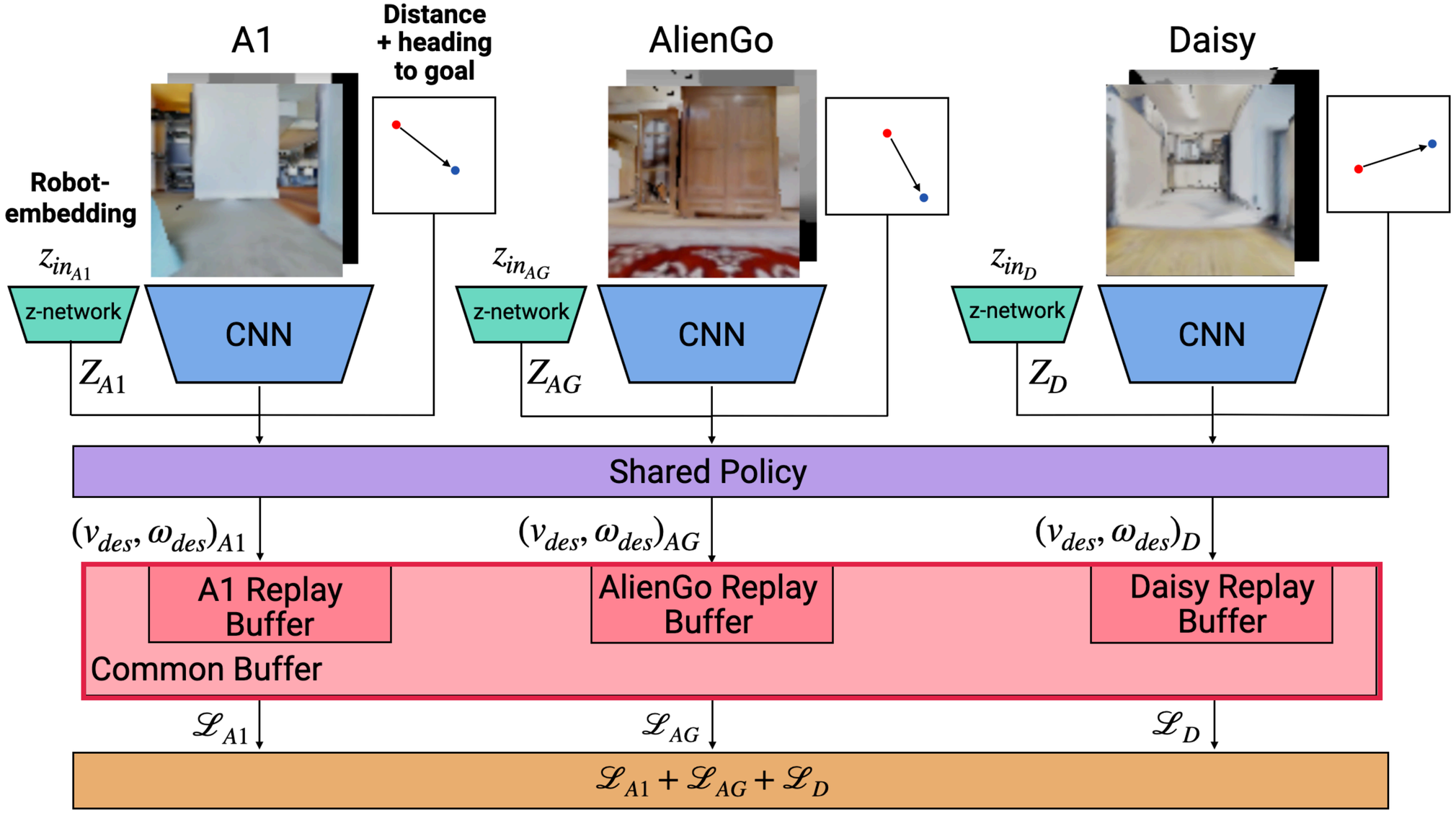
2) Learn a universal navigation policy across multiple robots

- Learns robot-specific embeddings (z)
- Shares data across multiple robots during learning



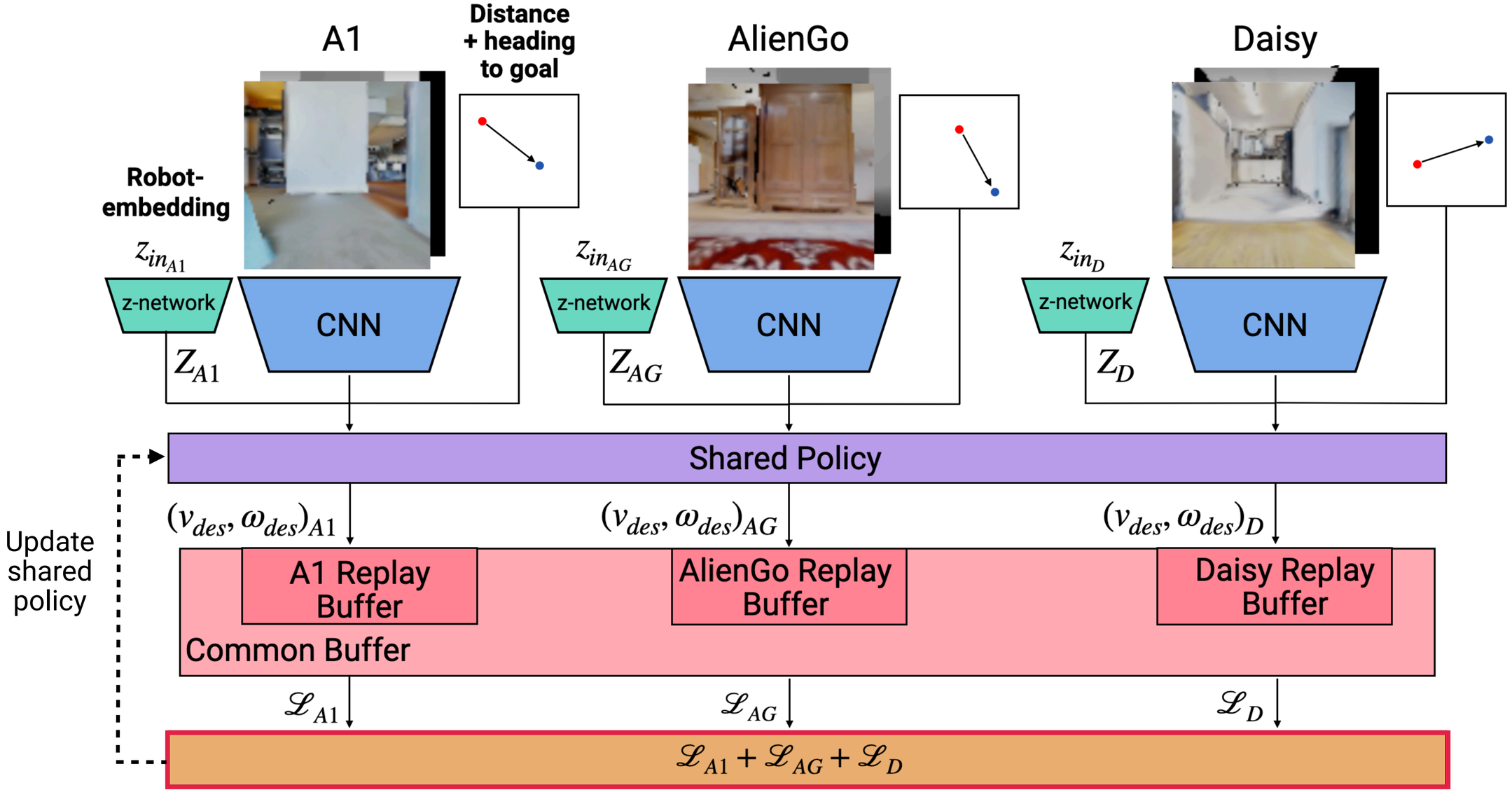
2) Learn a universal navigation policy across multiple robots

- Learns robot-specific embeddings (z)
- Shares data across multiple robots during learning



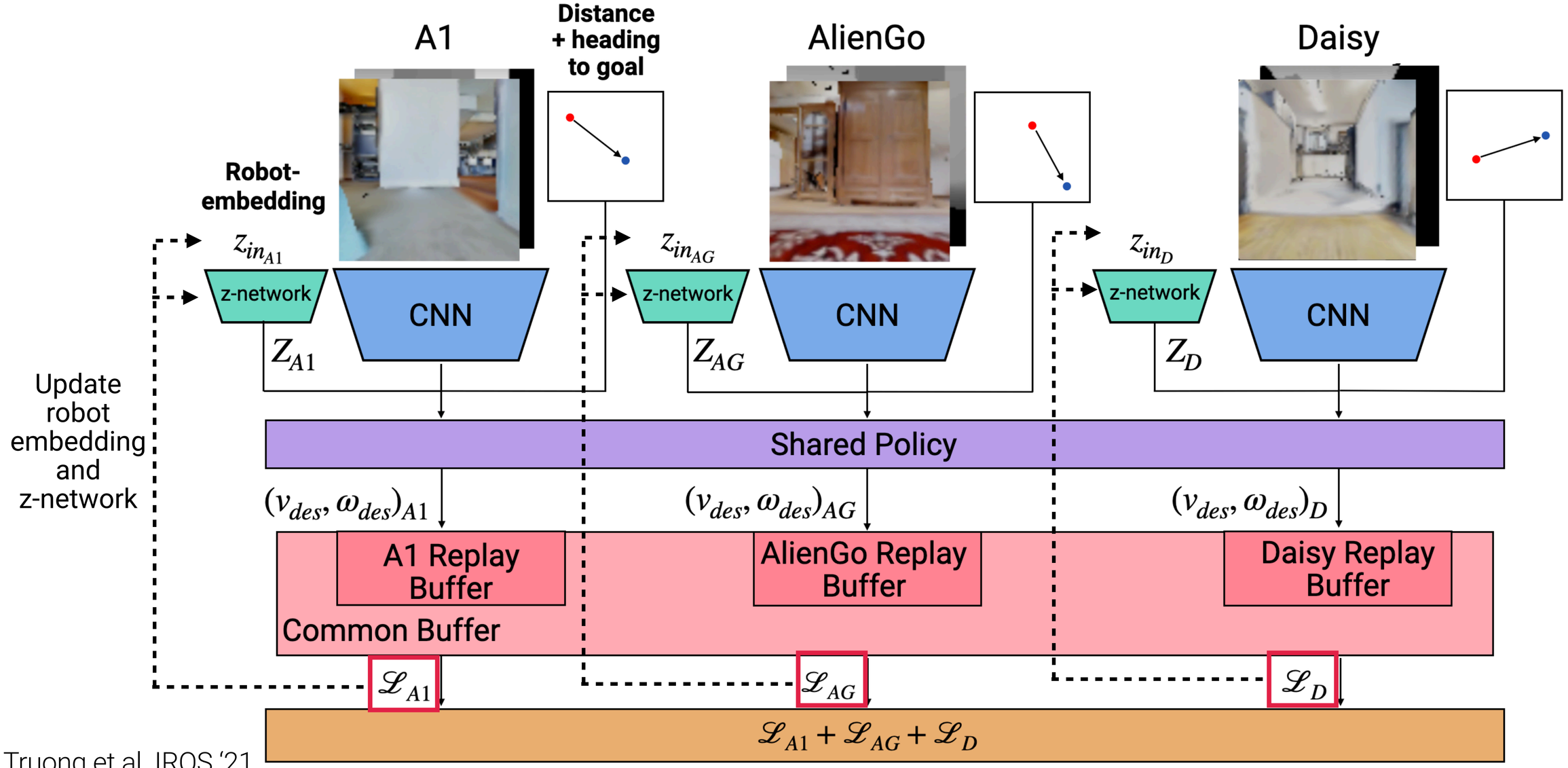
2) Learn a universal navigation policy across multiple robots

- Learns robot-specific embeddings (z)
- Shares data across multiple robots during learning



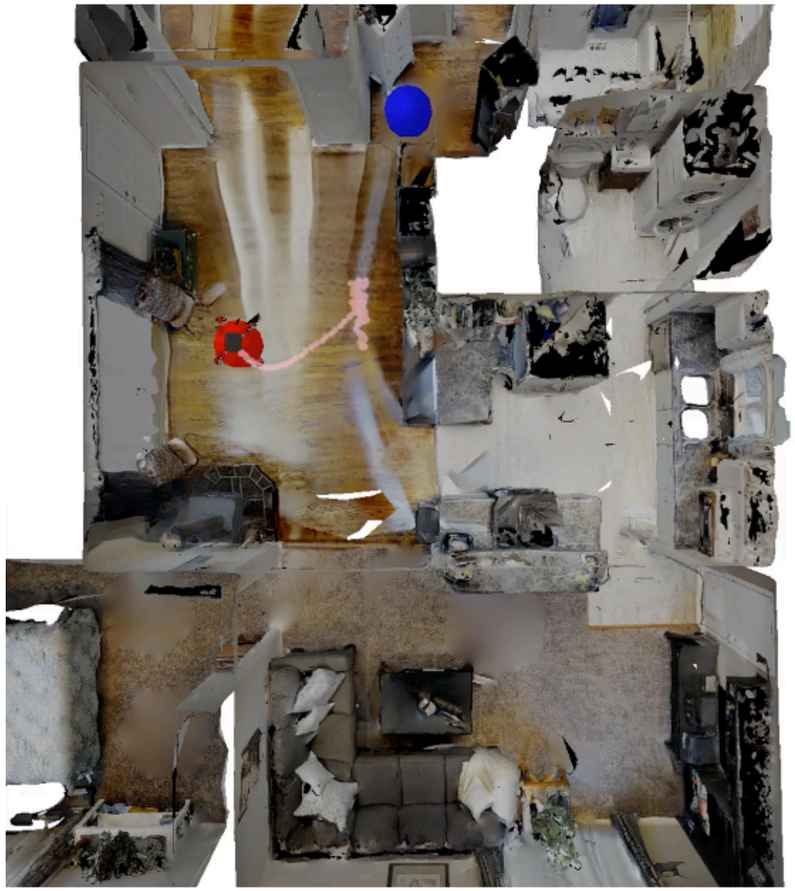
2) Learn a universal navigation policy across multiple robots

- Learns robot-specific embeddings (z)
- Shares data across multiple robots during learning

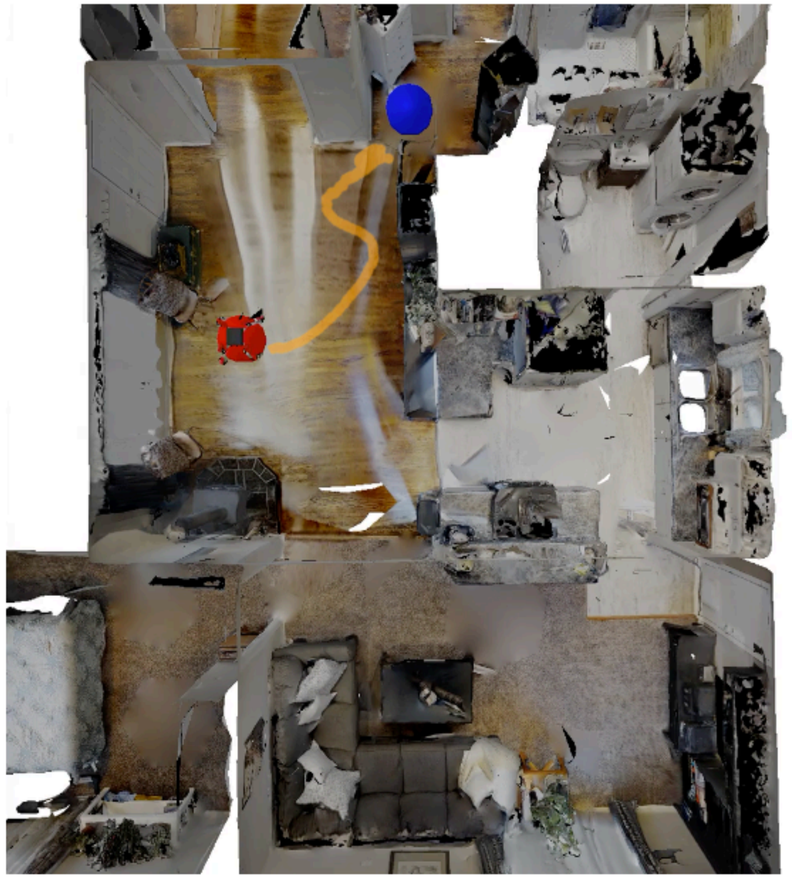


Truong et al. IROS '21

3) At test-time, we search for the optimal robot-embedding (z)



Z = 1.0

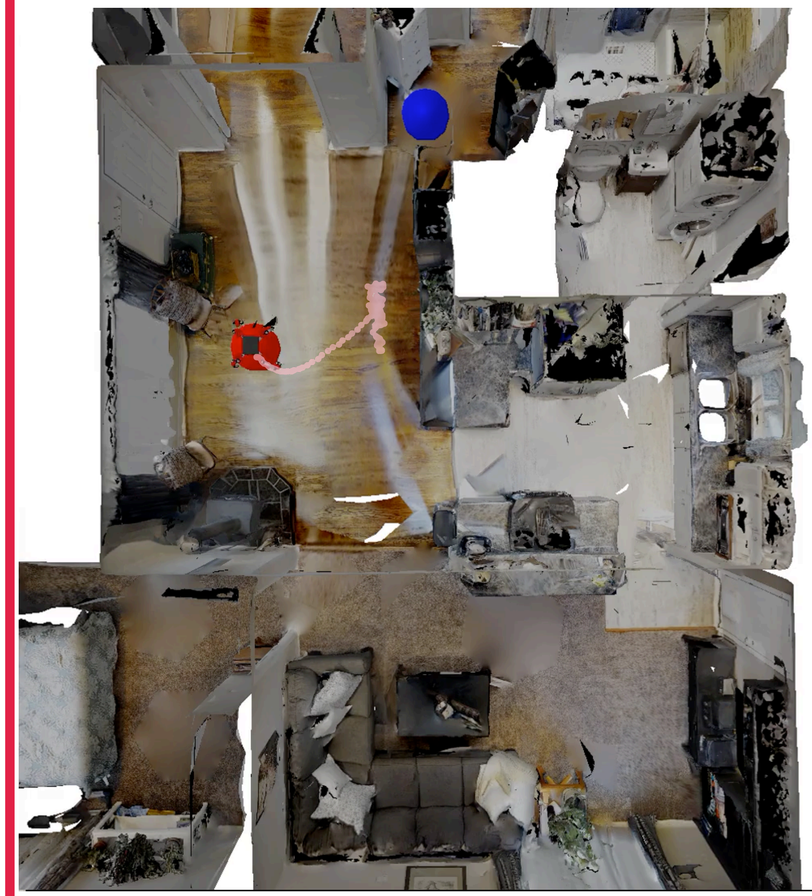


Z = -0.5

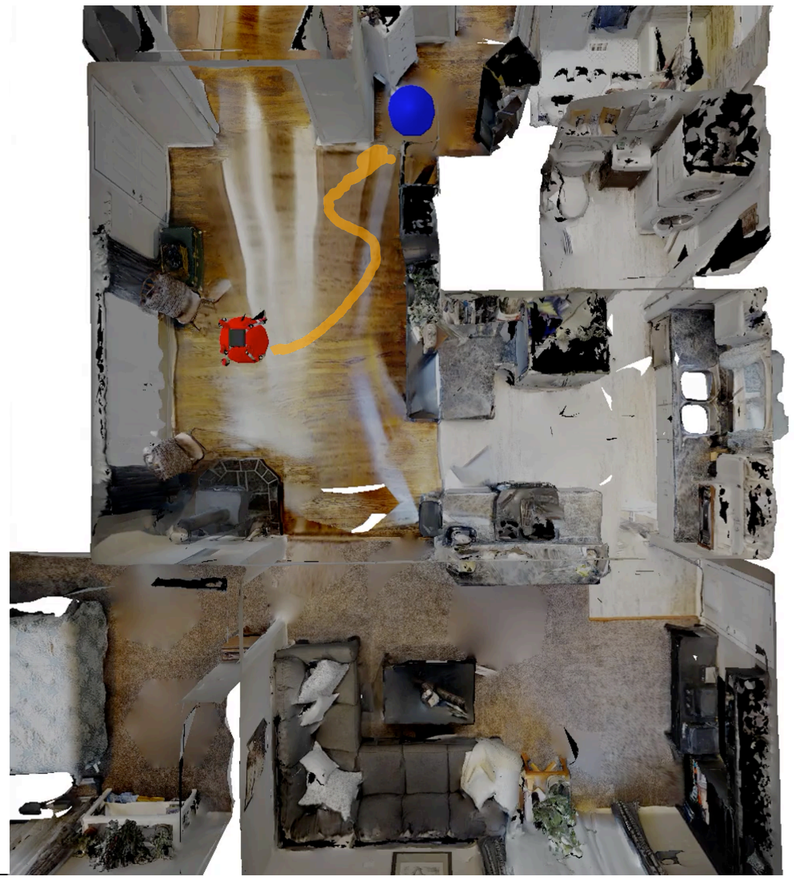
Z = -0.7

3) At test-time, we search for the optimal robot-embedding (z)

— The learned embedding captures different dynamical properties



Z = 1.0



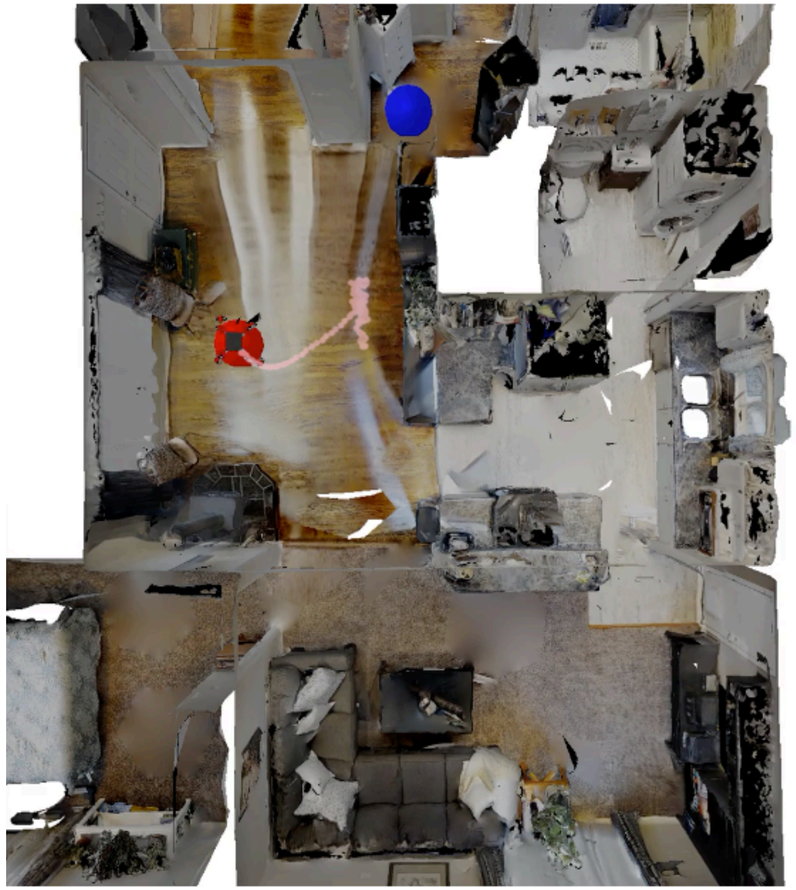
Z = -0.5

Z = -0.7

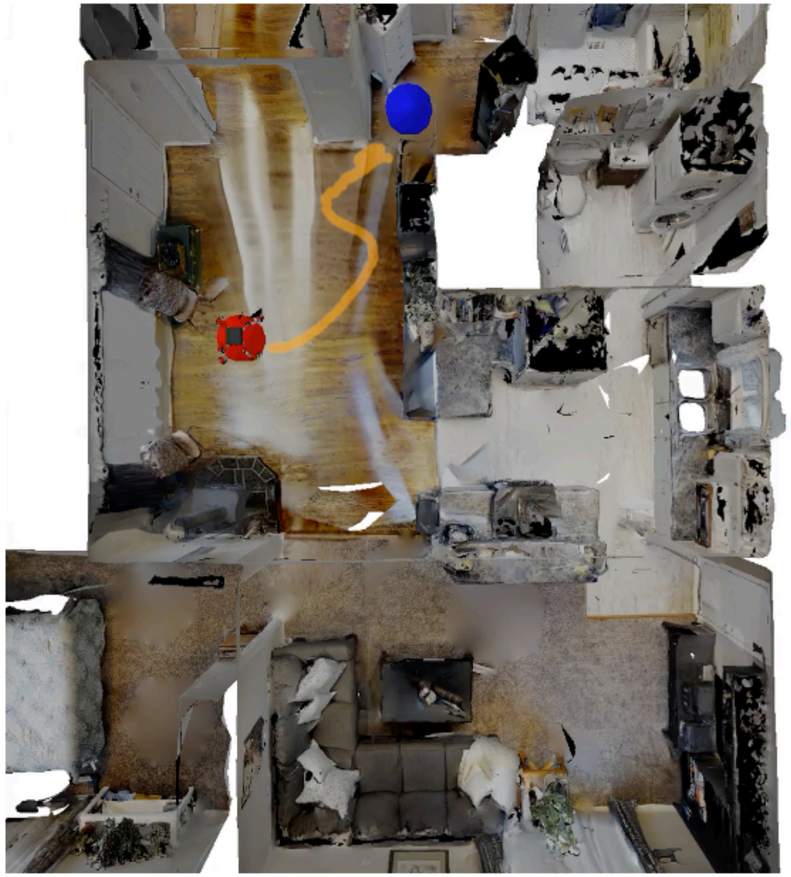
3) At test-time, we search for the optimal robot-embedding (z)

– The learned embedding captures different dynamical properties

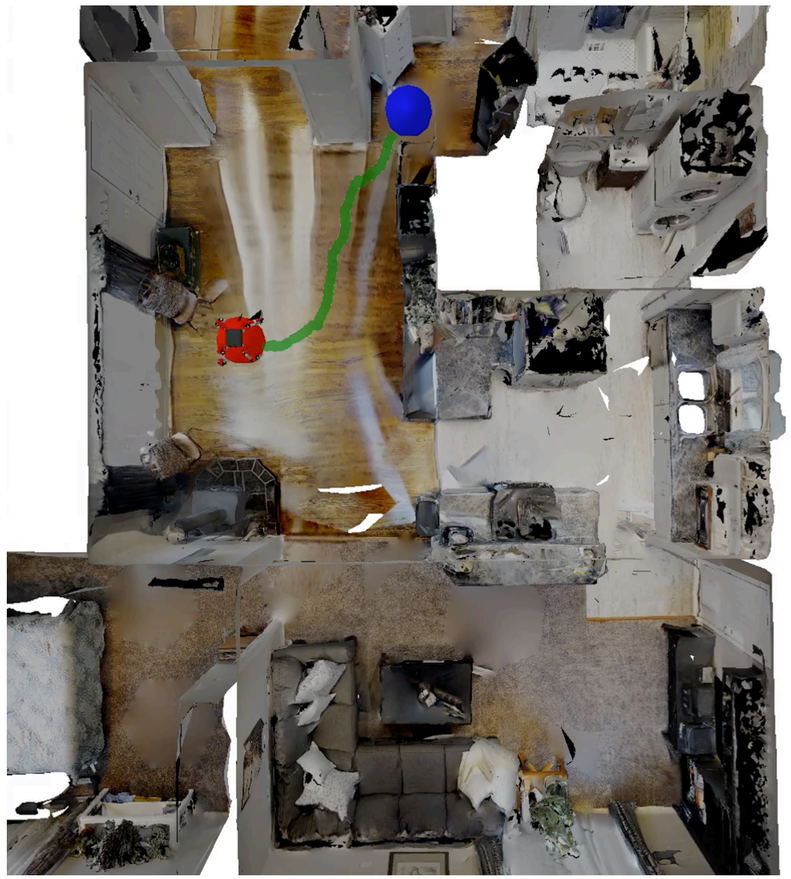
– Shows promising results new robots (4-legged Daisy)



Z = 1.0



Z = -0.5



Z = -0.7

Robot Experiments

We evaluate our approach on 5 robots in simulation, and one real-world quadruped robot (A1).

Robot Experiments

— 3 train robots (in new environments)

A1

AlienGo

Daisy

Robot Experiments

— 3 train robots (in new environments)

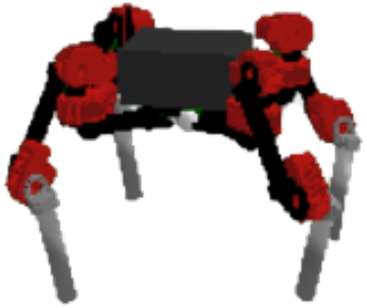
A1

AlienGo

Daisy

— 2 previously unseen robots in simulation and one real-world quadruped (in new envs)

Laikago



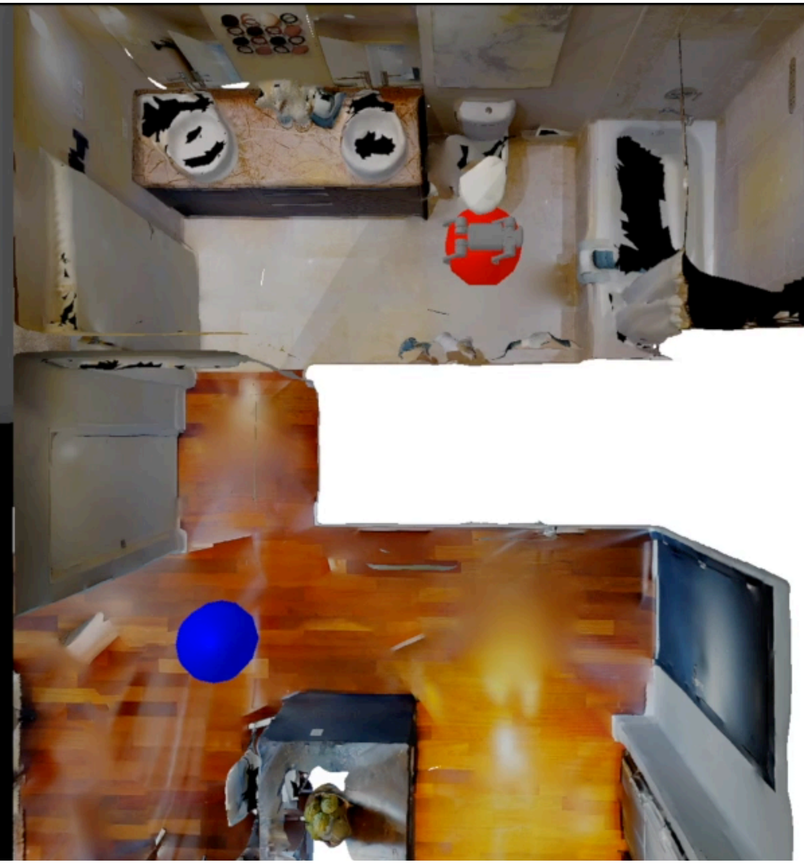
4 Legged Daisy

A1-Real

A1 (train robot, new environment)

Egocentric RGB

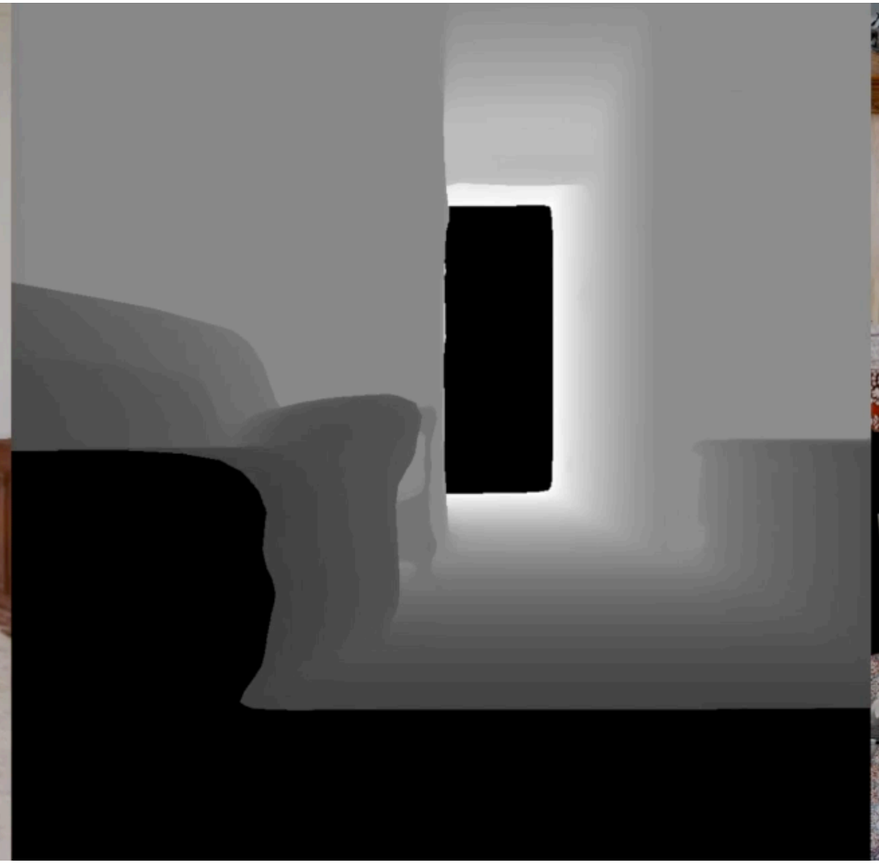
Egocentric Depth



Topdown View
(not available to robot)

AlienGo (train robot, new environment)

Egocentric RGB



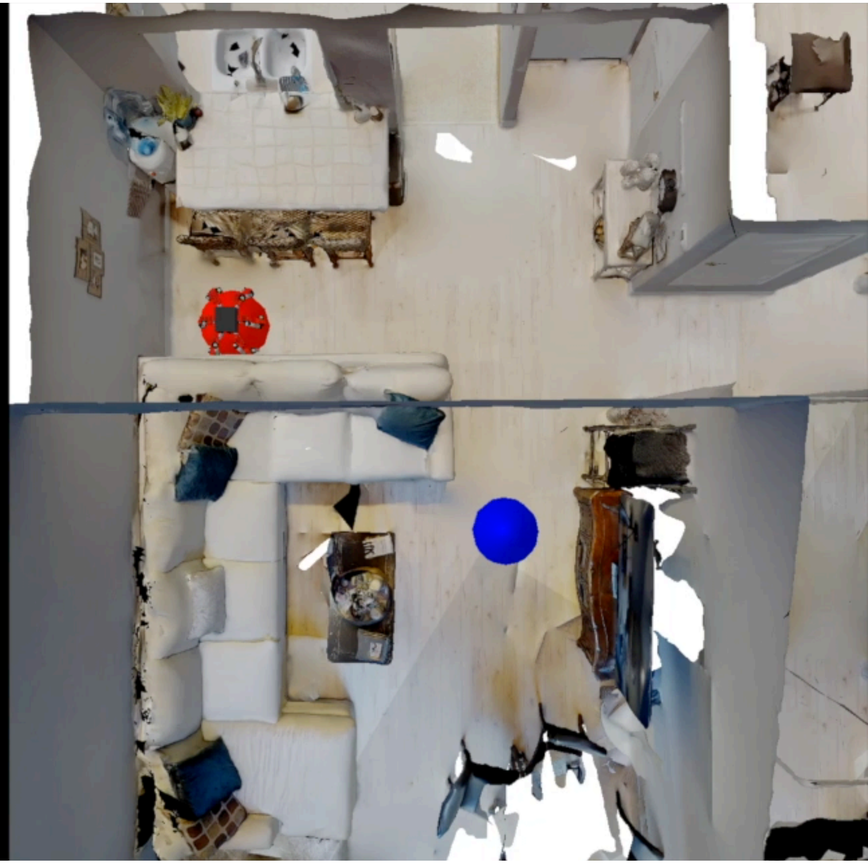
Egocentric Depth

Topdown View
(not available to robot)

Daisy (train robot, new environment)

Egocentric RGB

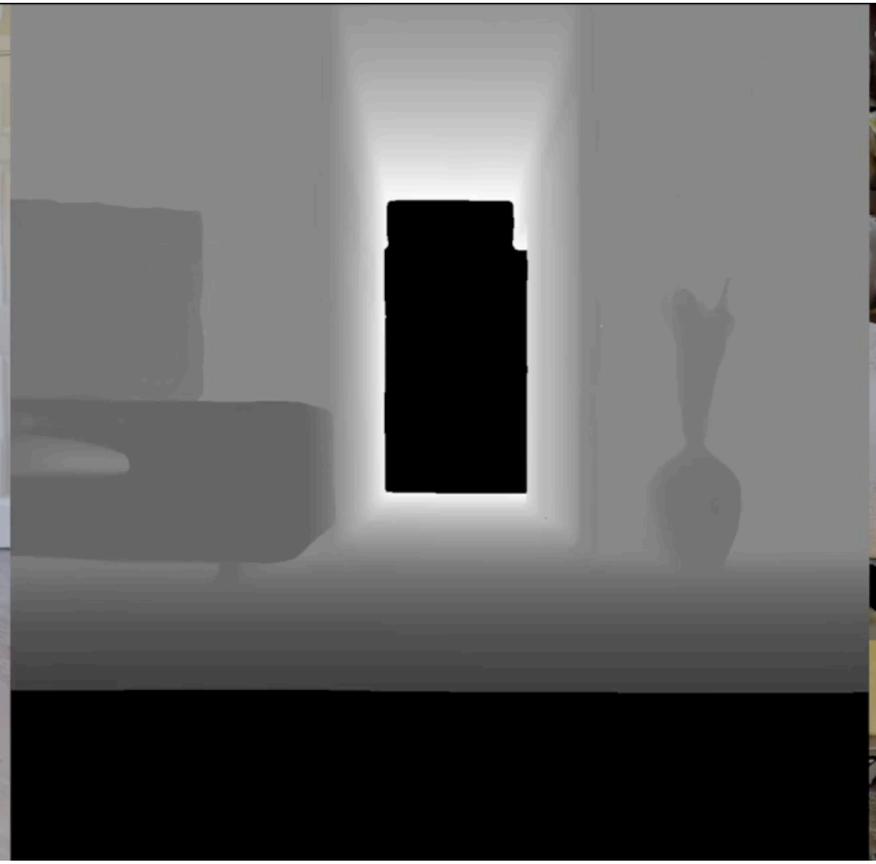
Egocentric Depth



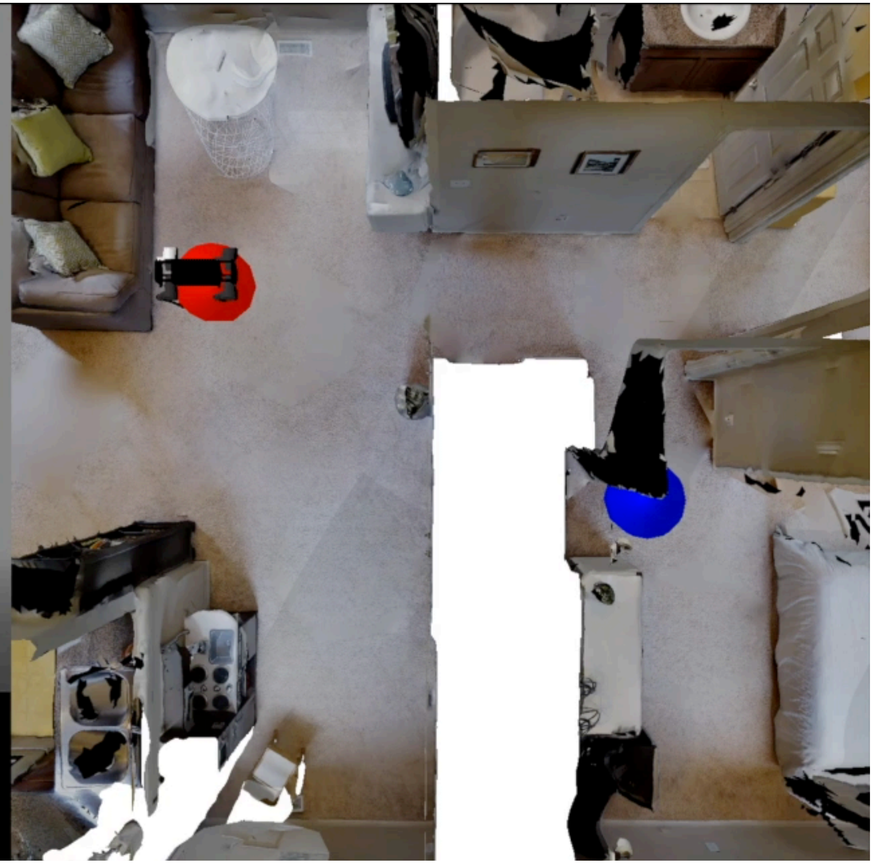
Topdown View
(not available to robot)

Laikago (unseen robot, new environment)

Egocentric RGB

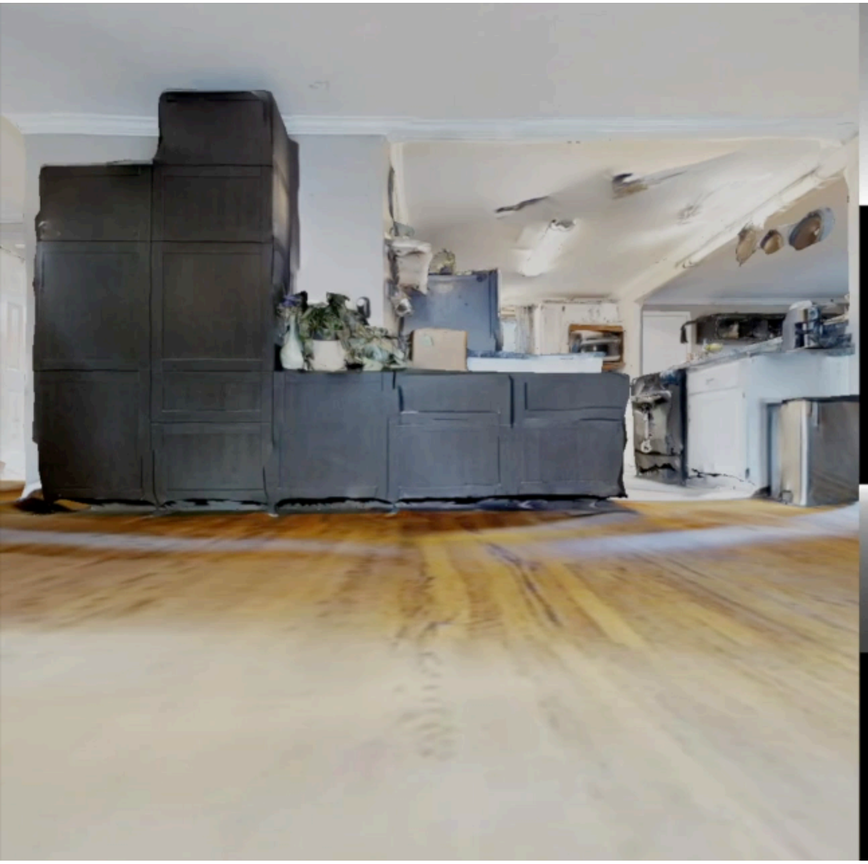


Egocentric Depth

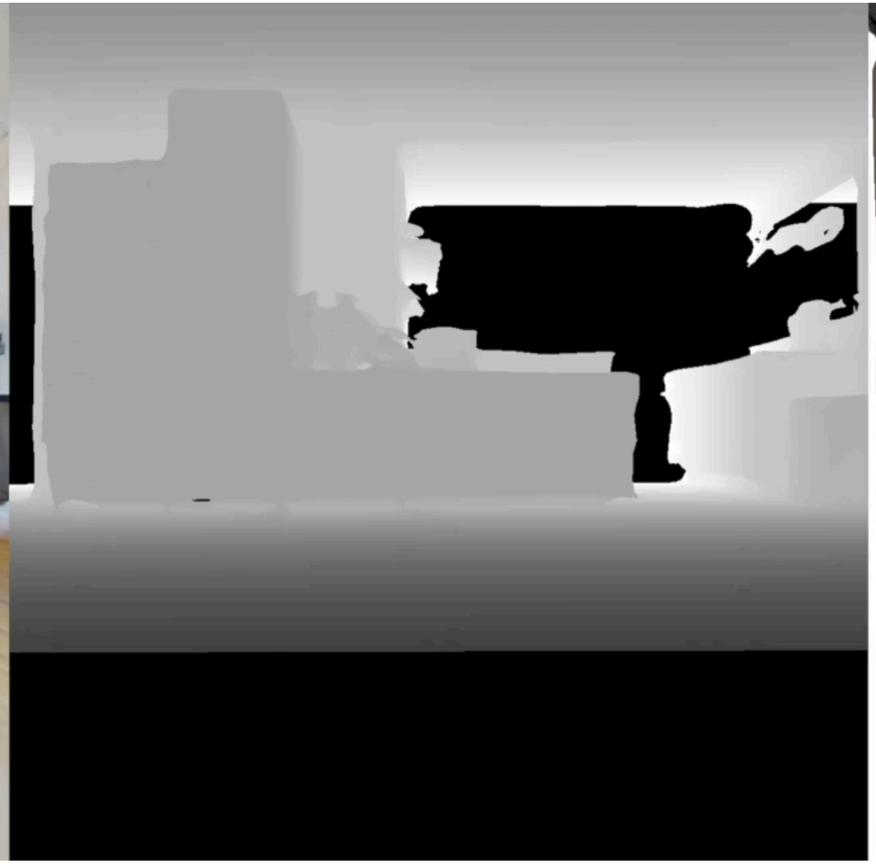


Topdown View
(not available to robot)

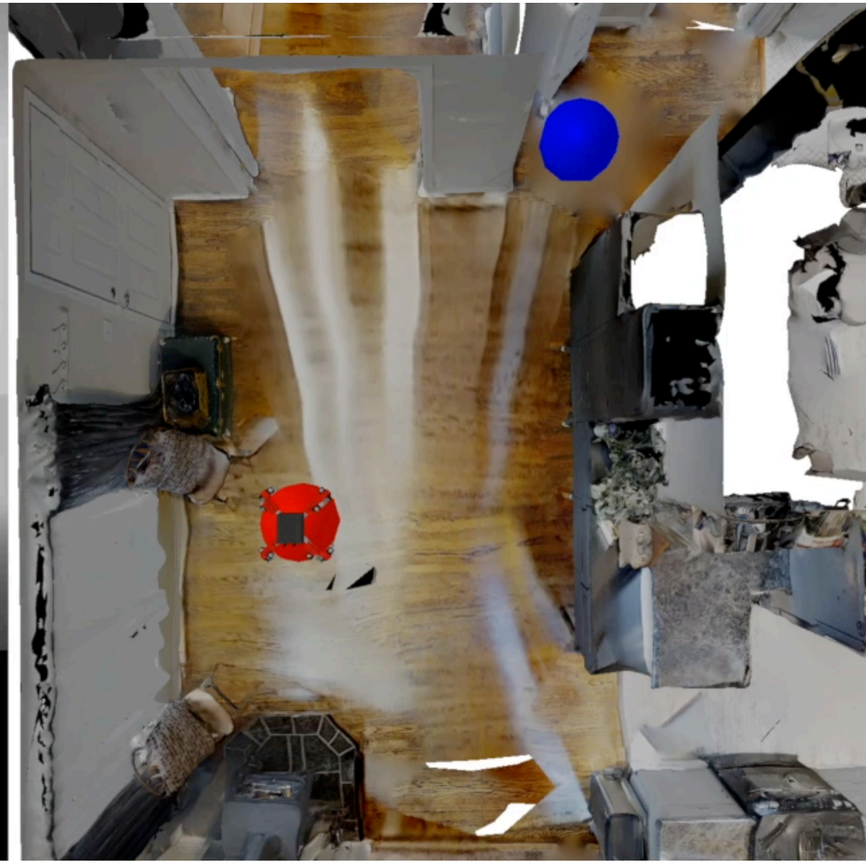
4 Legged Daisy (unseen robot, new environment)



Egocentric RGB



Egocentric Depth



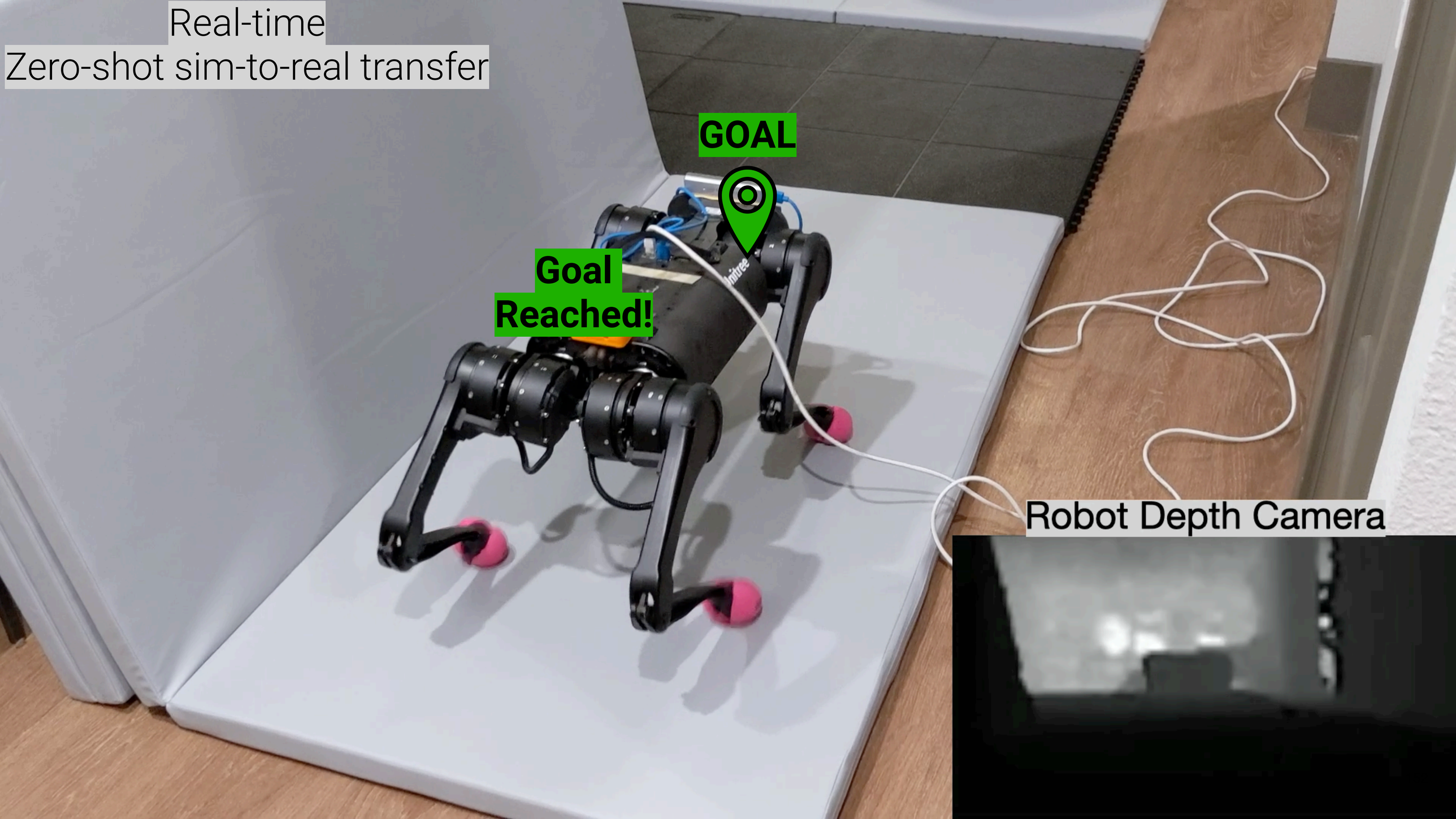
Topdown View
(not available to robot)

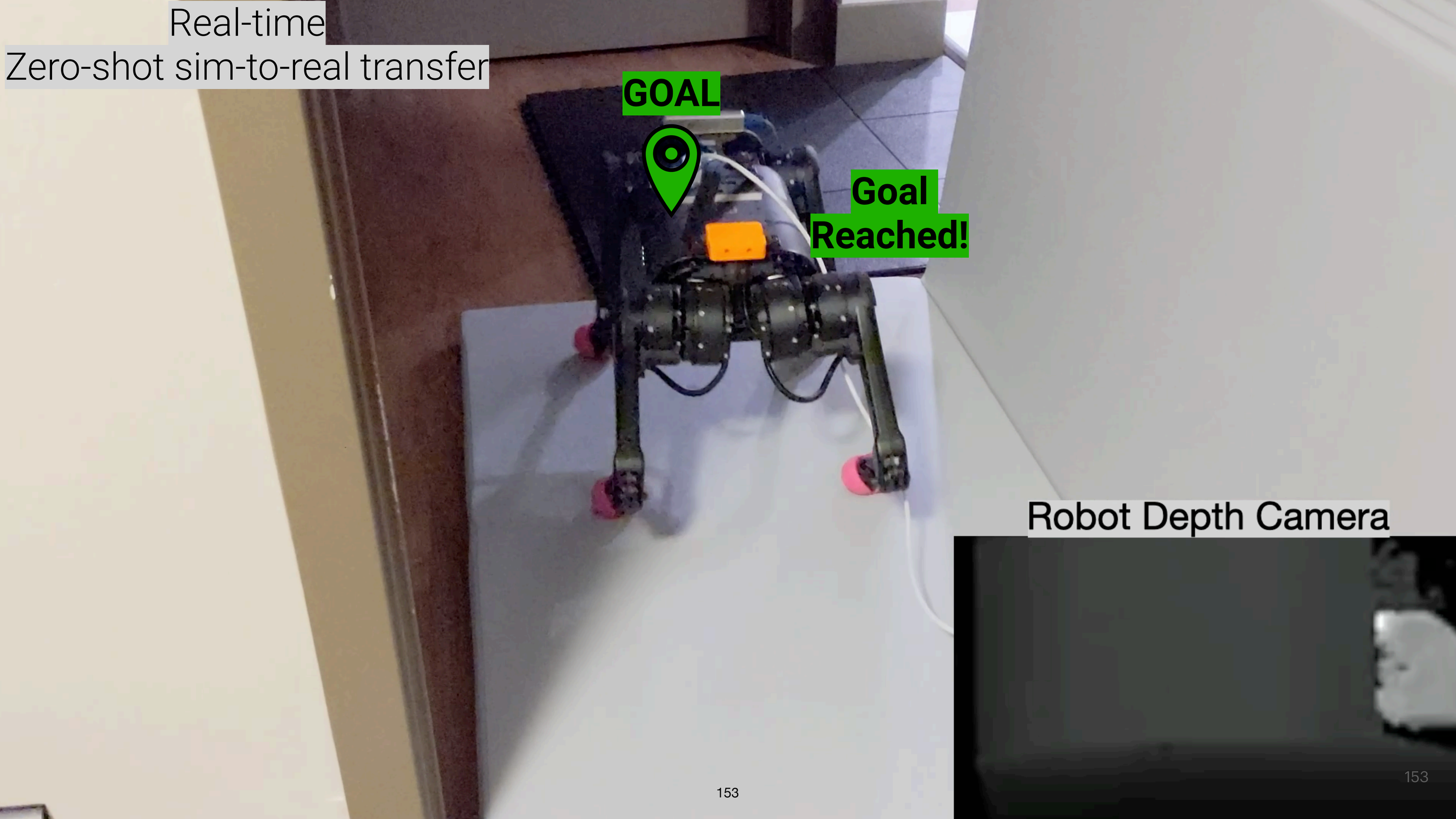
Real-time
Zero-shot sim-to-real transfer

GOAL

Goal Reached!

Robot Depth Camera


Real-time
Zero-shot sim-to-real transfer
GOAL
Goal
Reached!
Robot Depth Camera


Real-time
Zero-shot sim-to-real transfer
GOAL
Goal
Reached!
Robot Depth Camera

153

Thank you!

Questions?