

CS 4803 / 7643: Deep Learning

Topics:

- Application: PointGoal Navigation
- Trust Region Policy Optimization (TRPO)
- Proximal Policy Optimization (PPO)

Erik Wijmans
Georgia Tech

Who Am I?



Erik Wijmans

3rd year PhD student at GT
Advisors: Dhruv Batra and
Irfan Essa

Research Interests

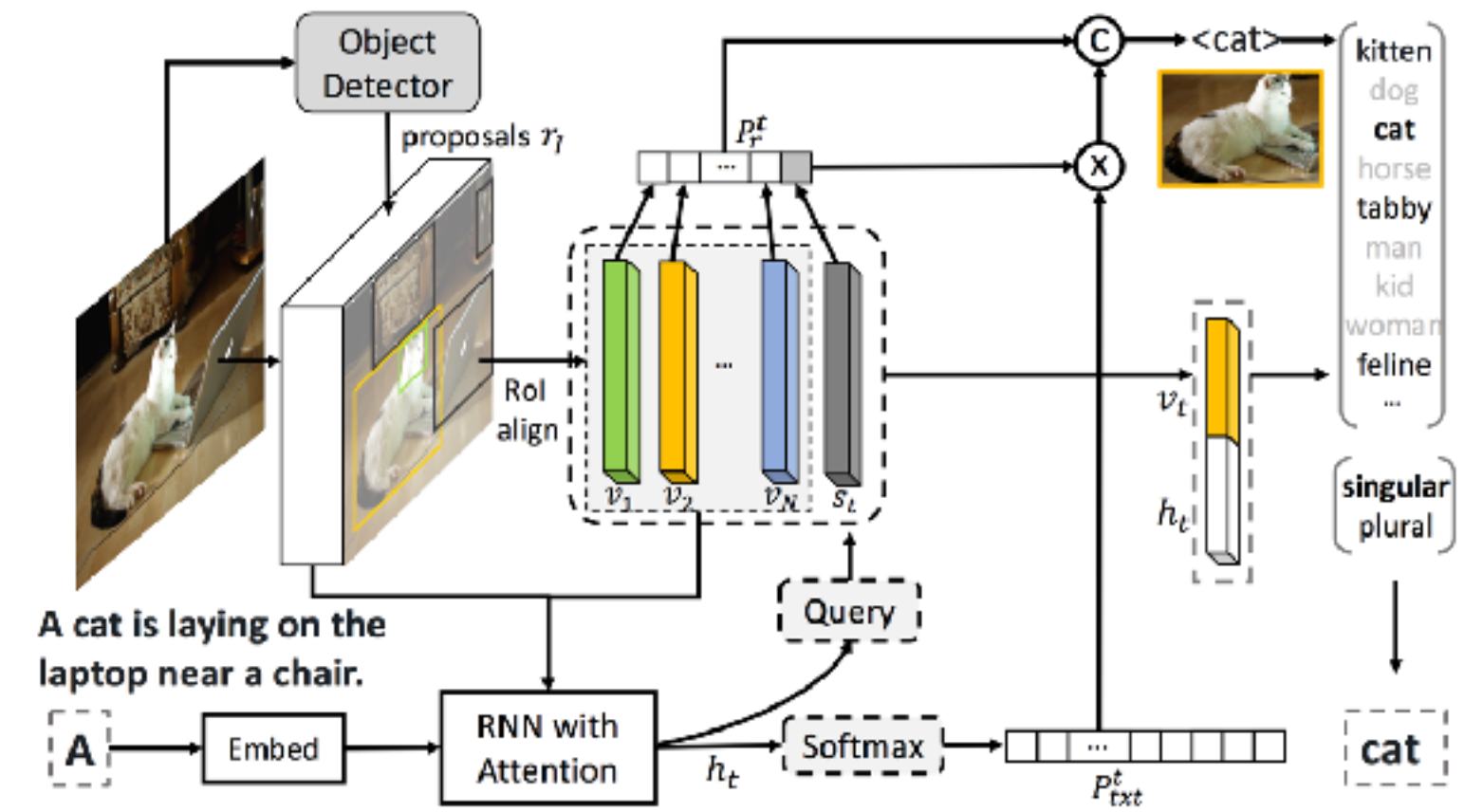
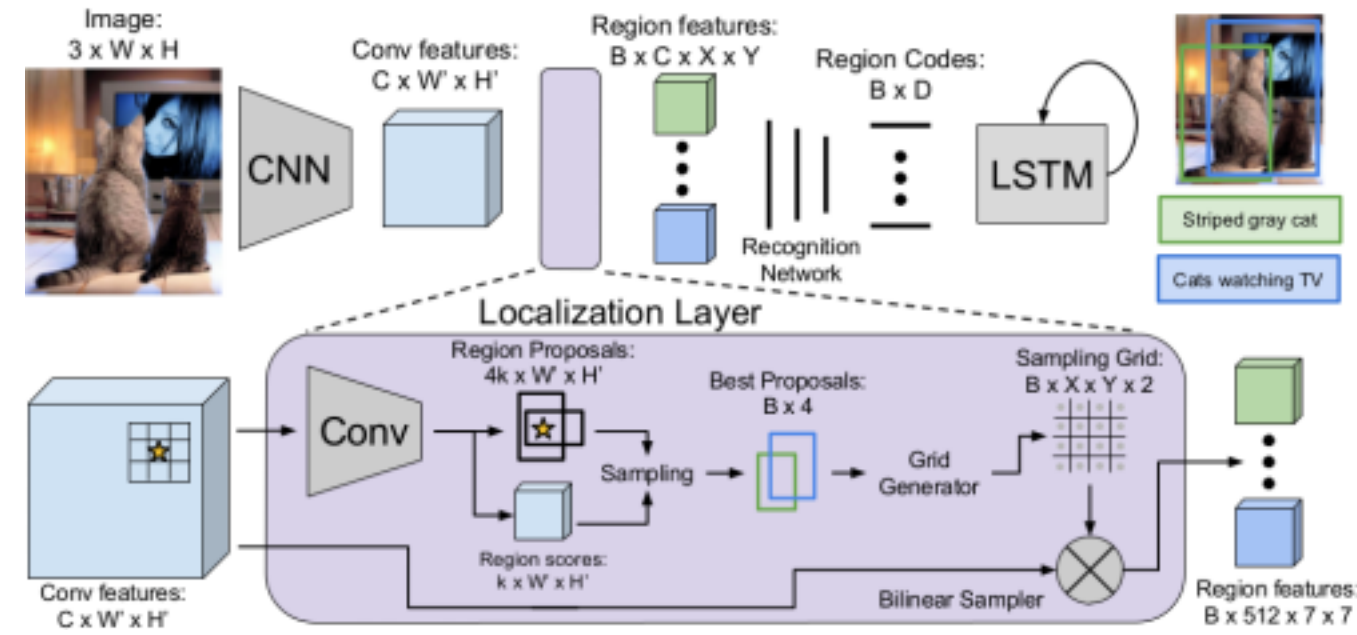
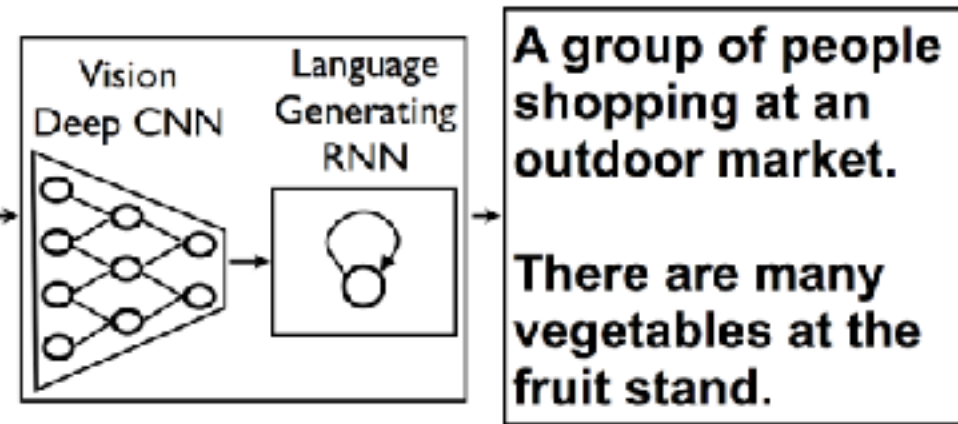
- Computer Vision
- Visual Navigation
- Embodied AI (virtual robots)
- Simulation to reality transfer

Lecture plan/motivation

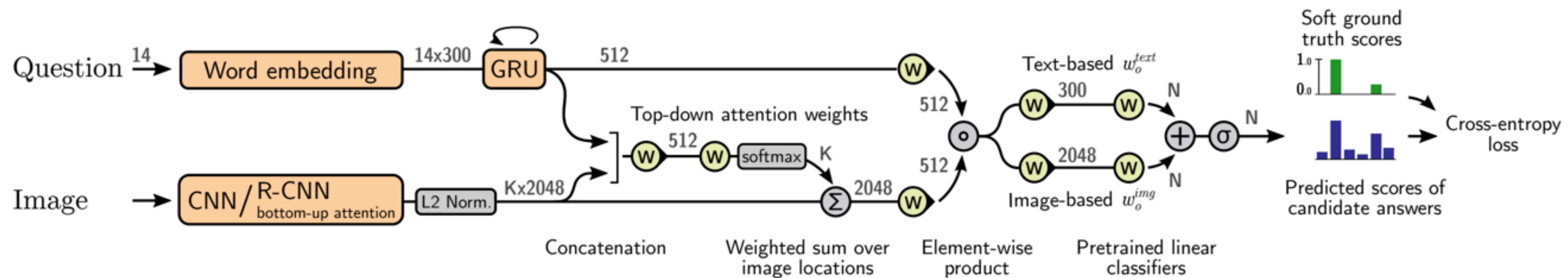
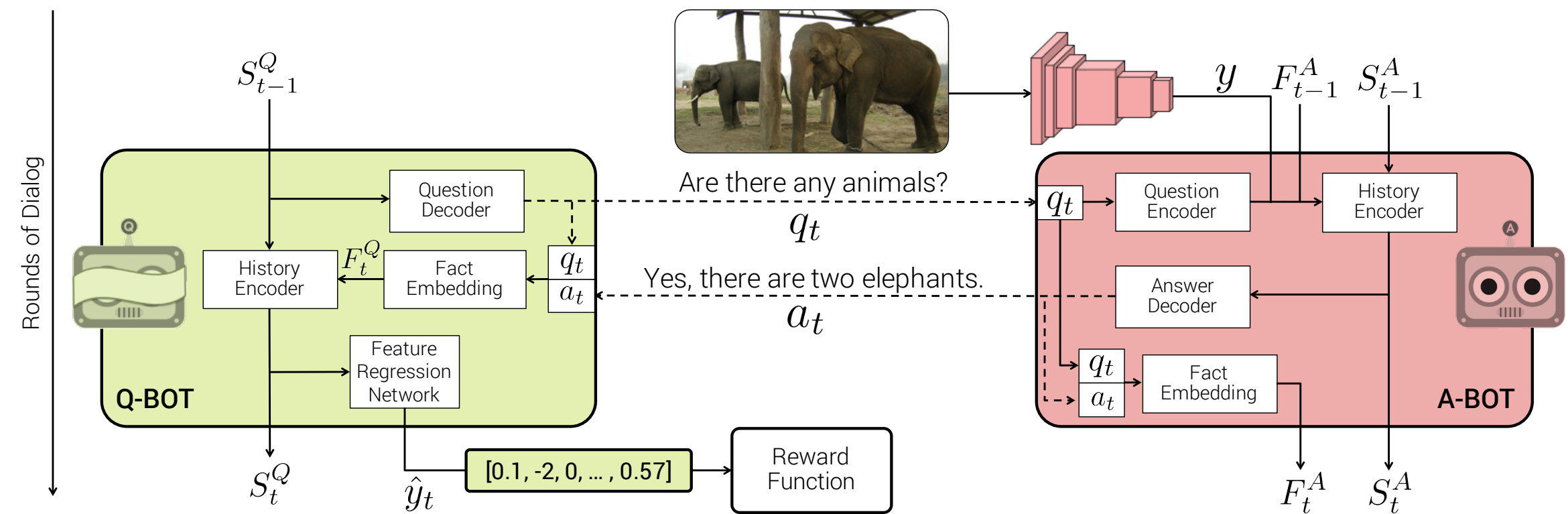
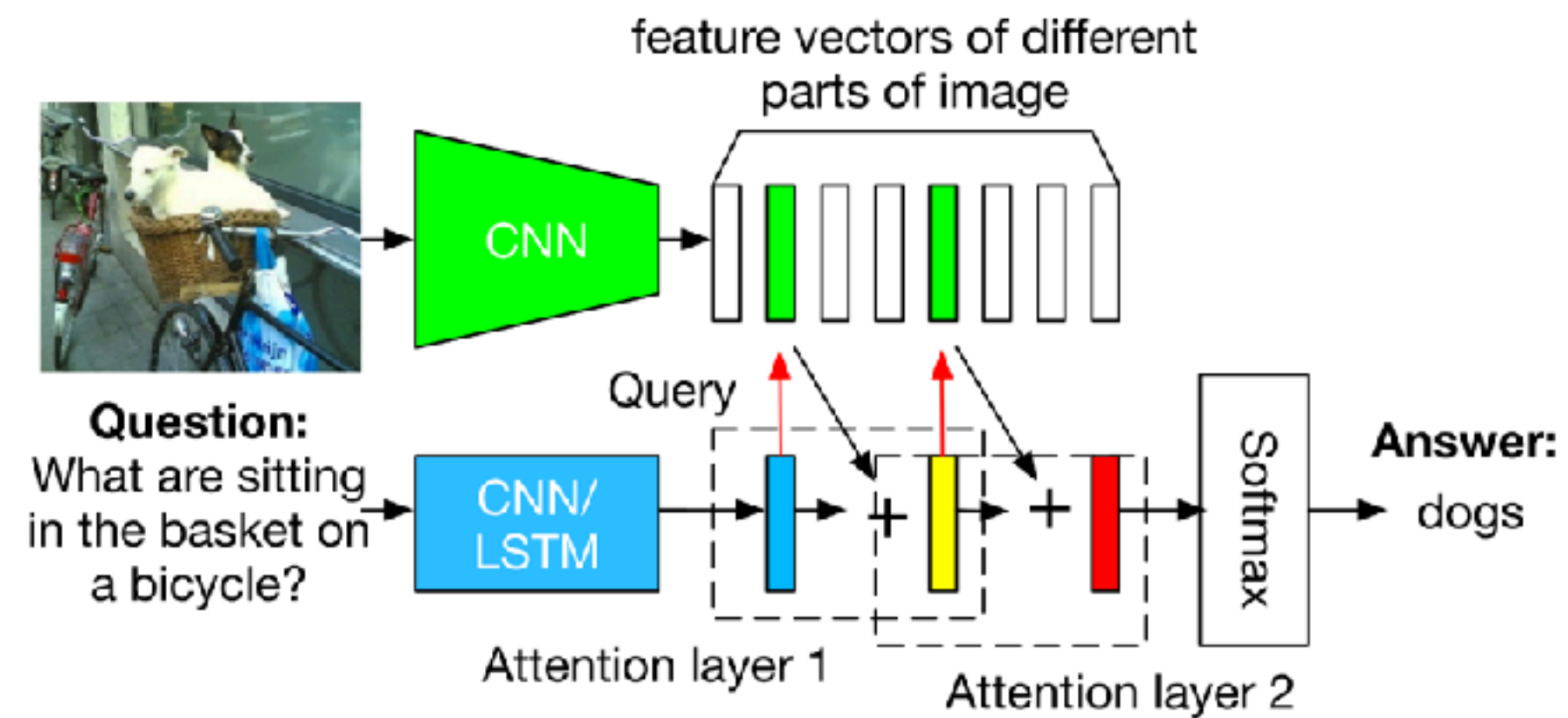
- Combine CNNs, RNNs (LSTMs), and RL together – all things you have learned about in this course – through a task called PointGoal Navigation
- Introduce more advanced RL – TRPO and PPO
- Show results using PPO on PointGoal Navigation

State-of-the-Art Visual Recognition

State-of-the-Art Visual Recognition



State-of-the-Art Visual Recognition



State-of-the-Art Visual Recognition

Applications

Applications



Applications

Physical agent



Applications

Physical agent
capable of taking
actions in the world



Applications

Physical agent
capable of taking
actions in the world

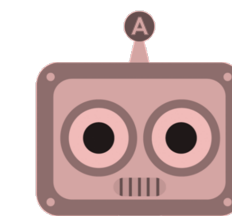


Applications

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

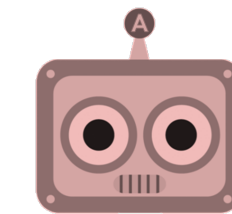


Is there smoke in any room around you?



Yes, in one room

Go there and look for people

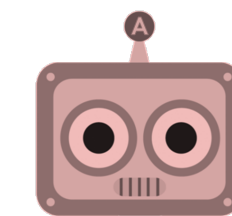


...

Applications

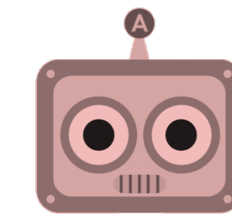


Is there smoke in any room around you?



Yes, in one room

Go there and look for people

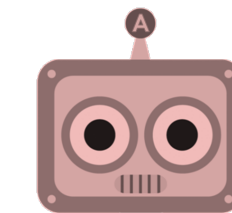


...

Applications

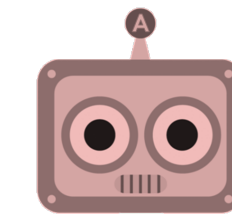


Is there smoke in any room around you?



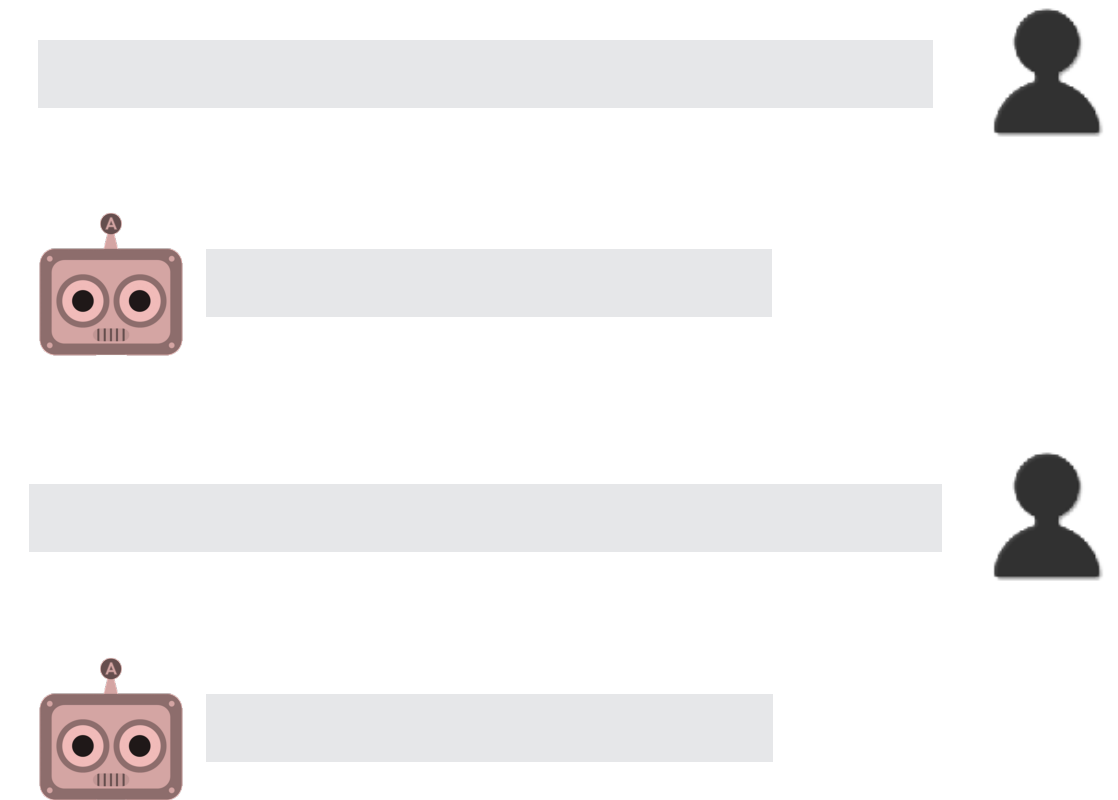
Yes, in one room

Go there and look for people



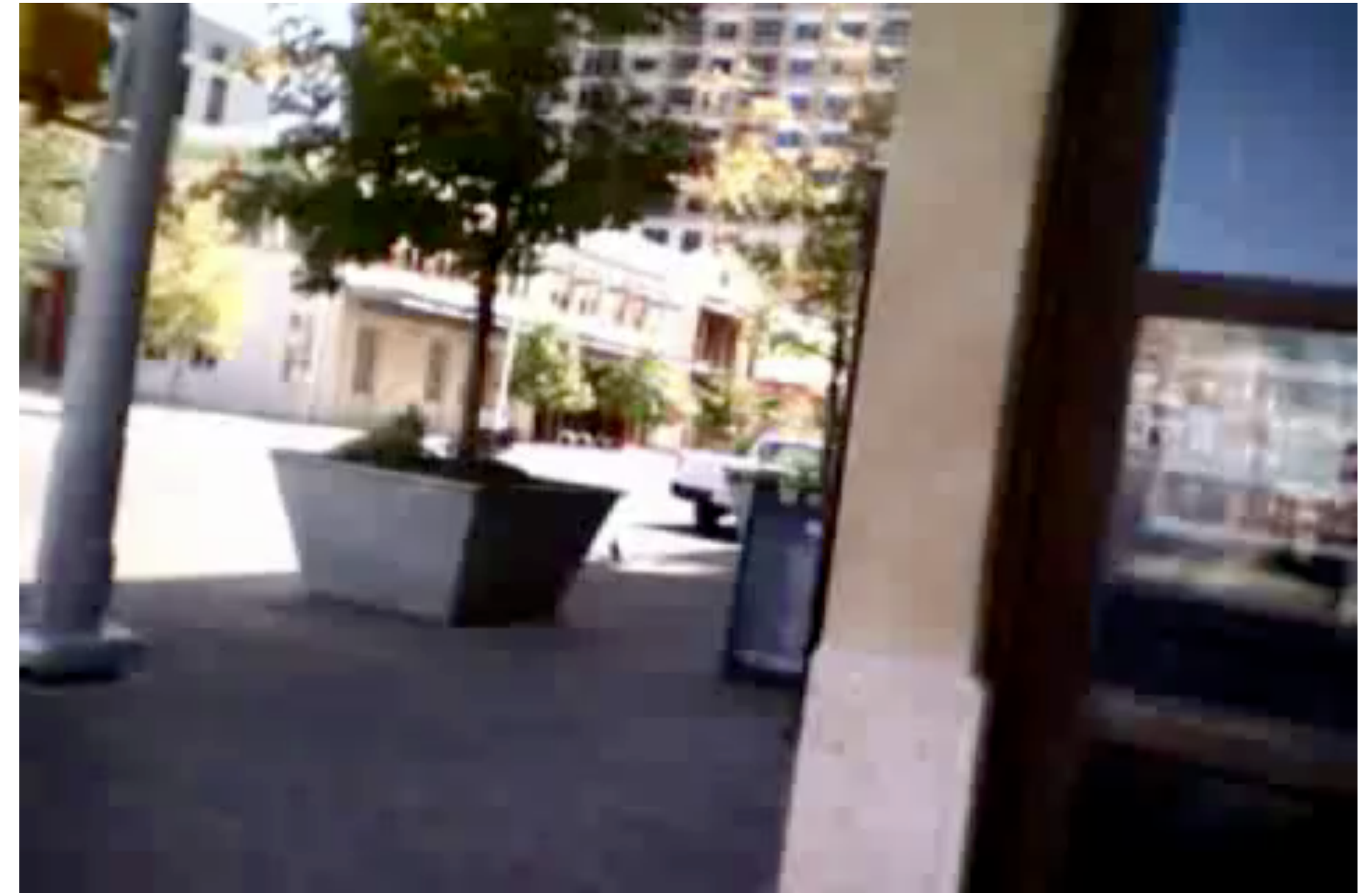
...

Challenges



Challenges

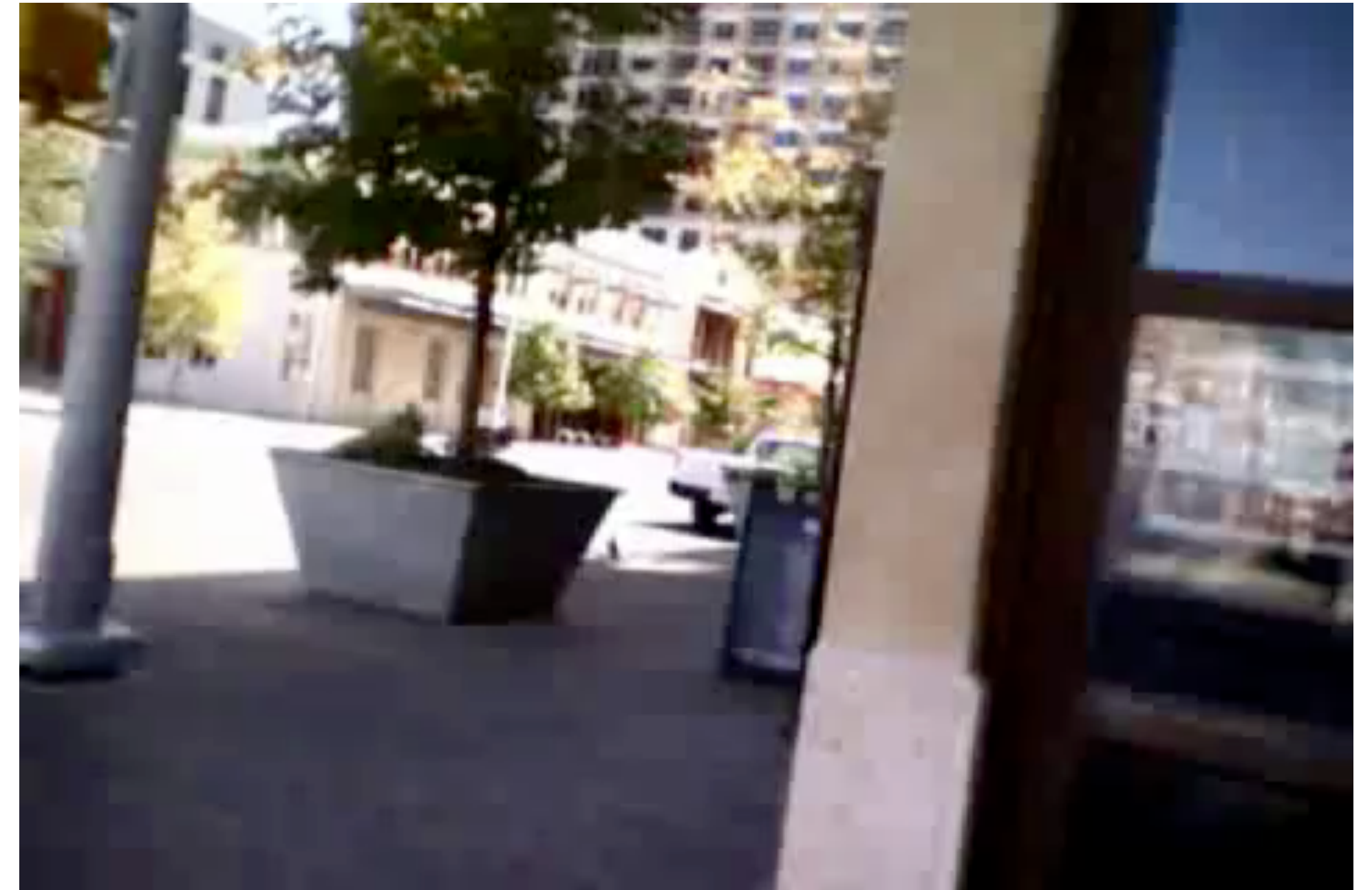
Egocentric vision



No access to well-composed, curated images

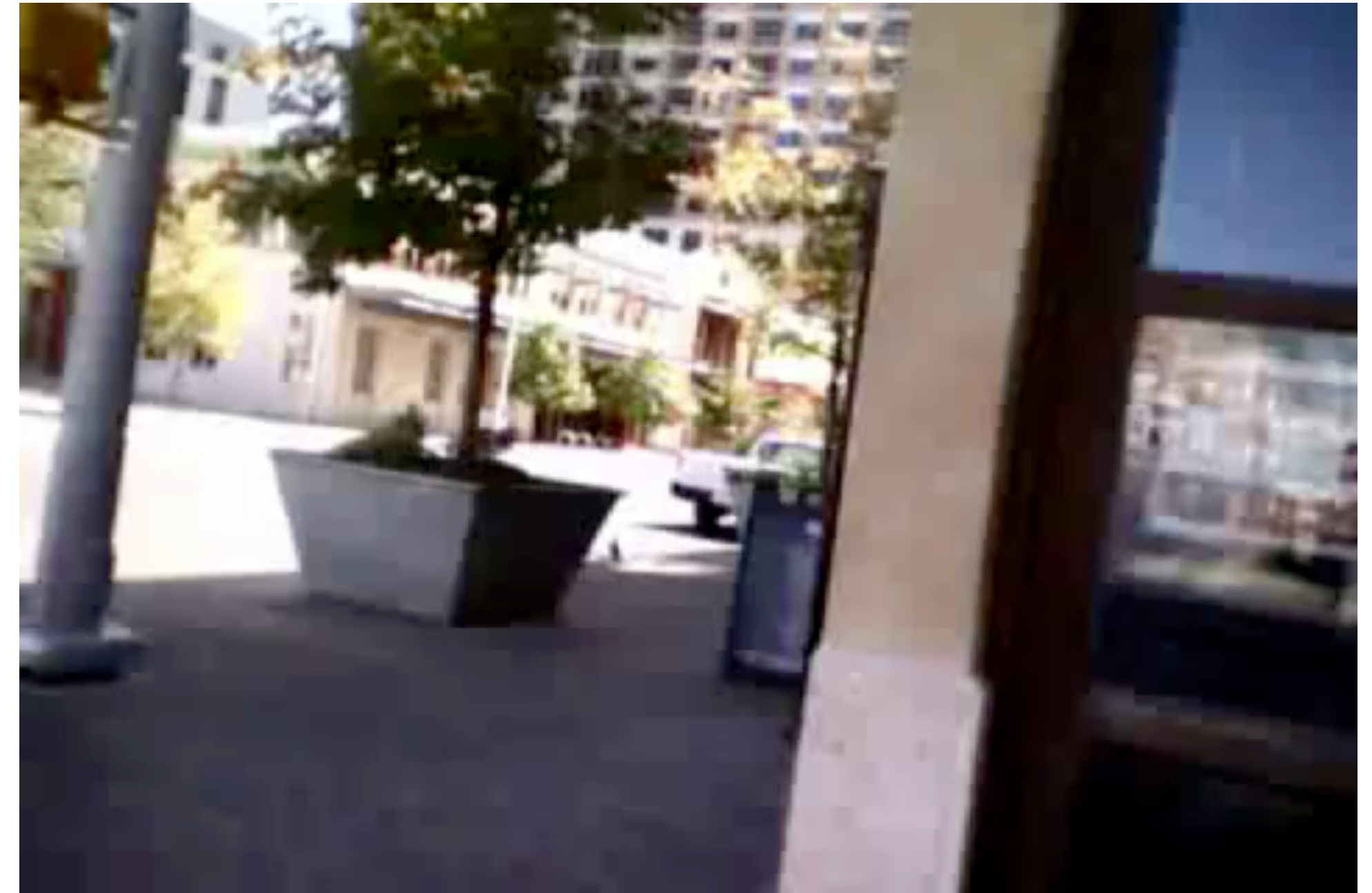
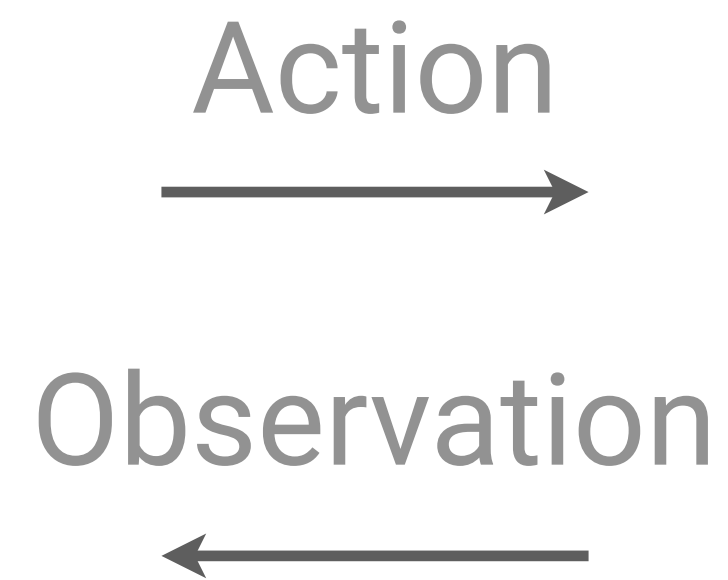
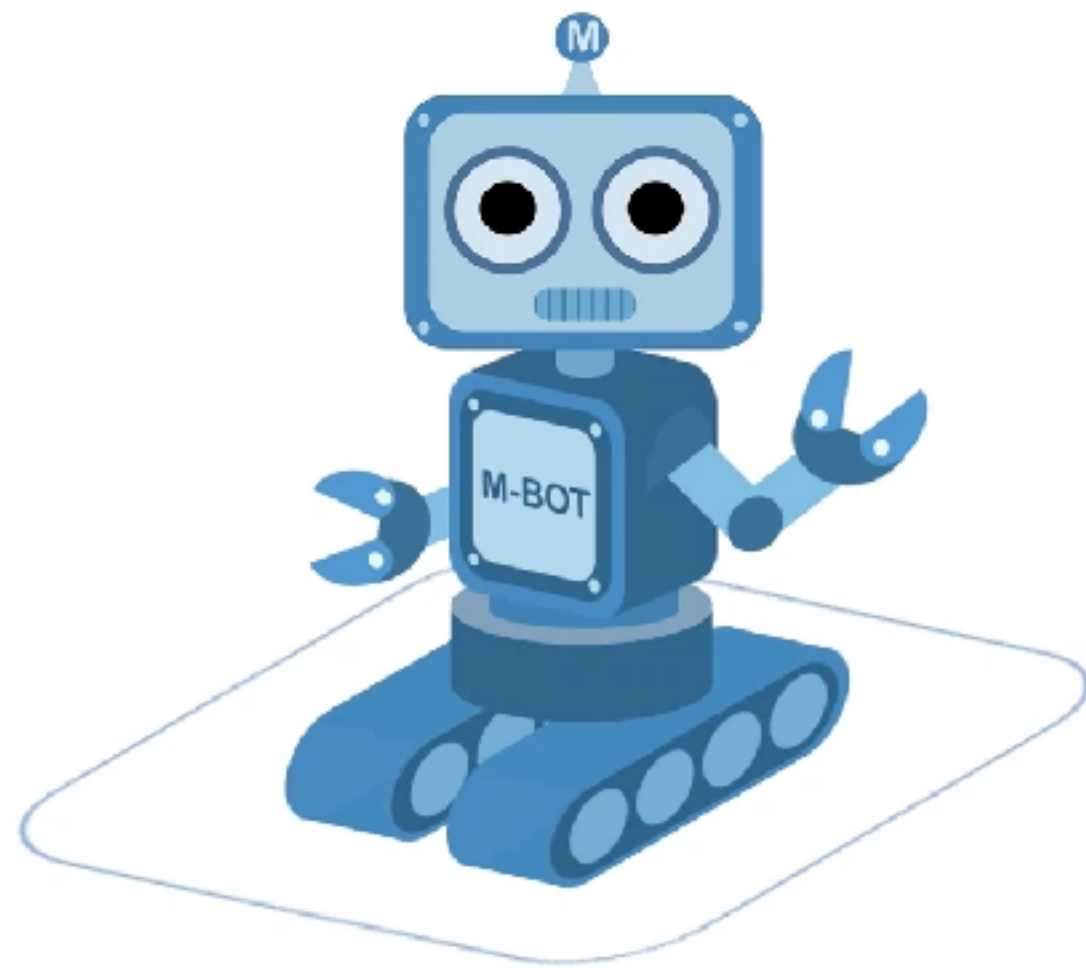
Challenges

Egocentric vision



Challenges

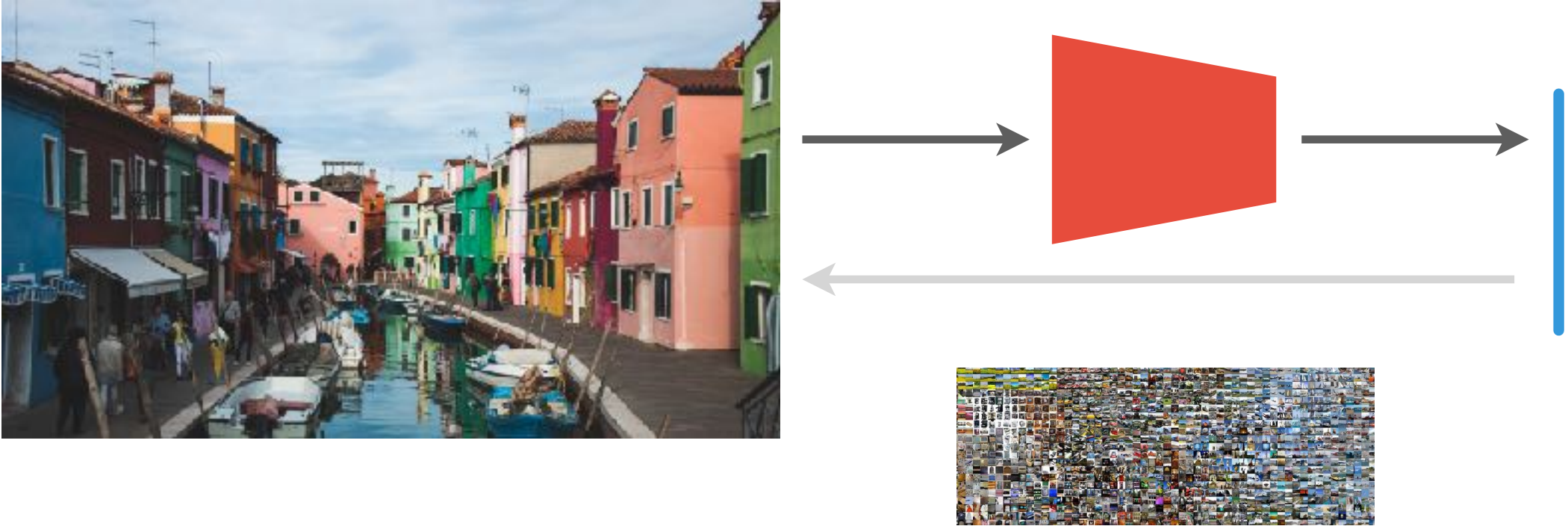
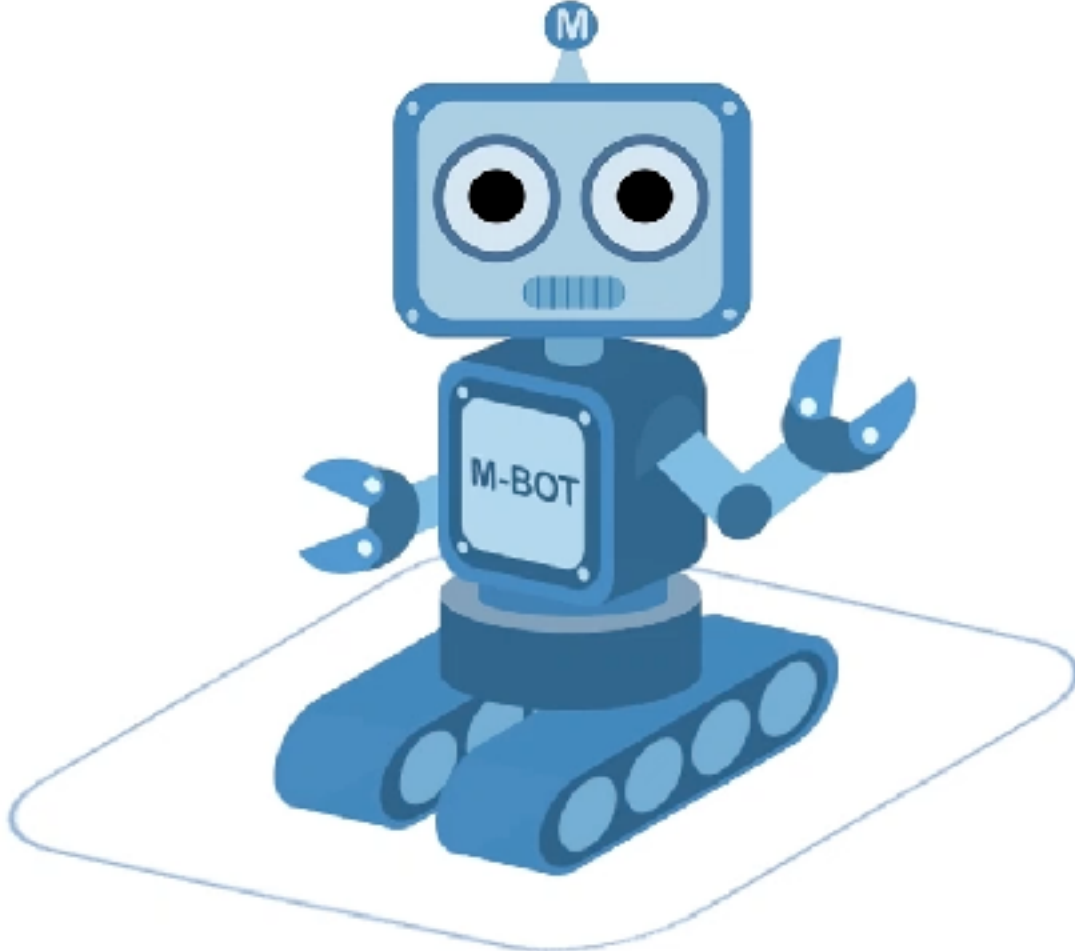
Egocentric vision
Active perception



Agent controls incoming data distribution

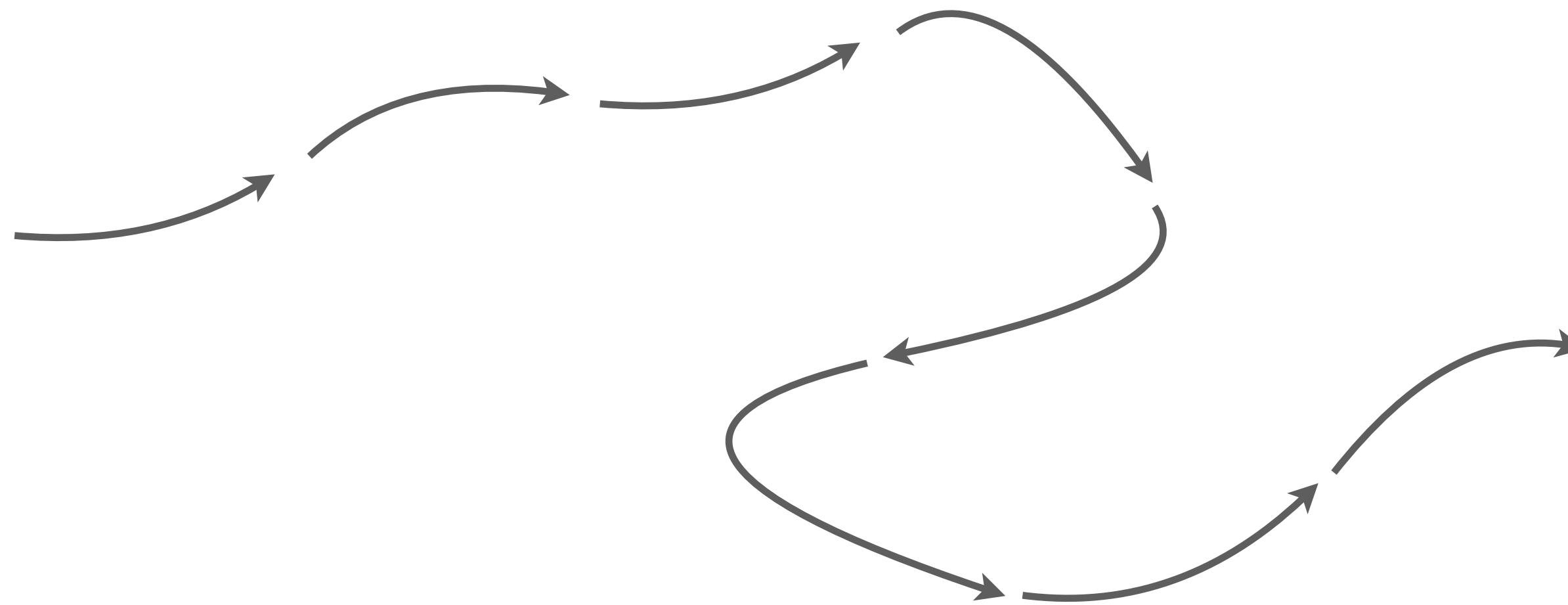
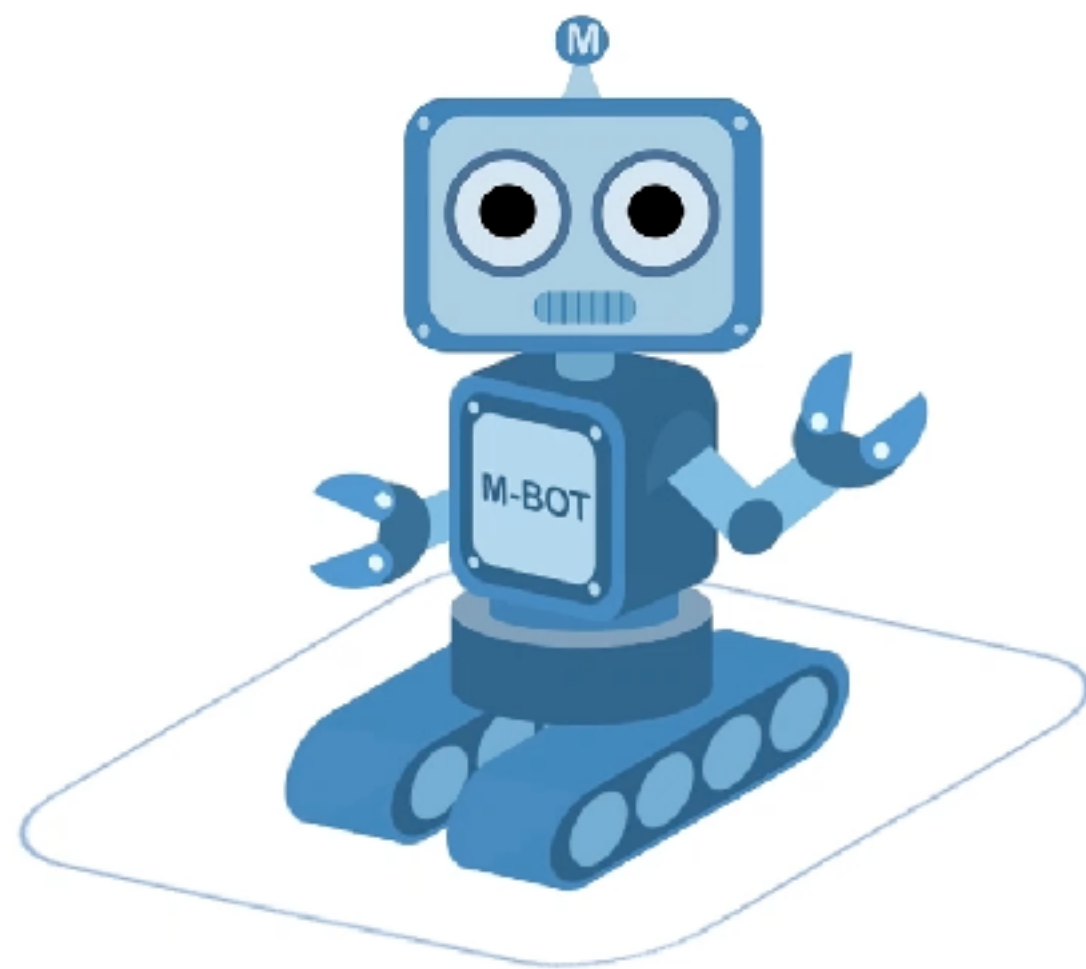
Challenges

Egocentric vision
Active perception
Sparse rewards



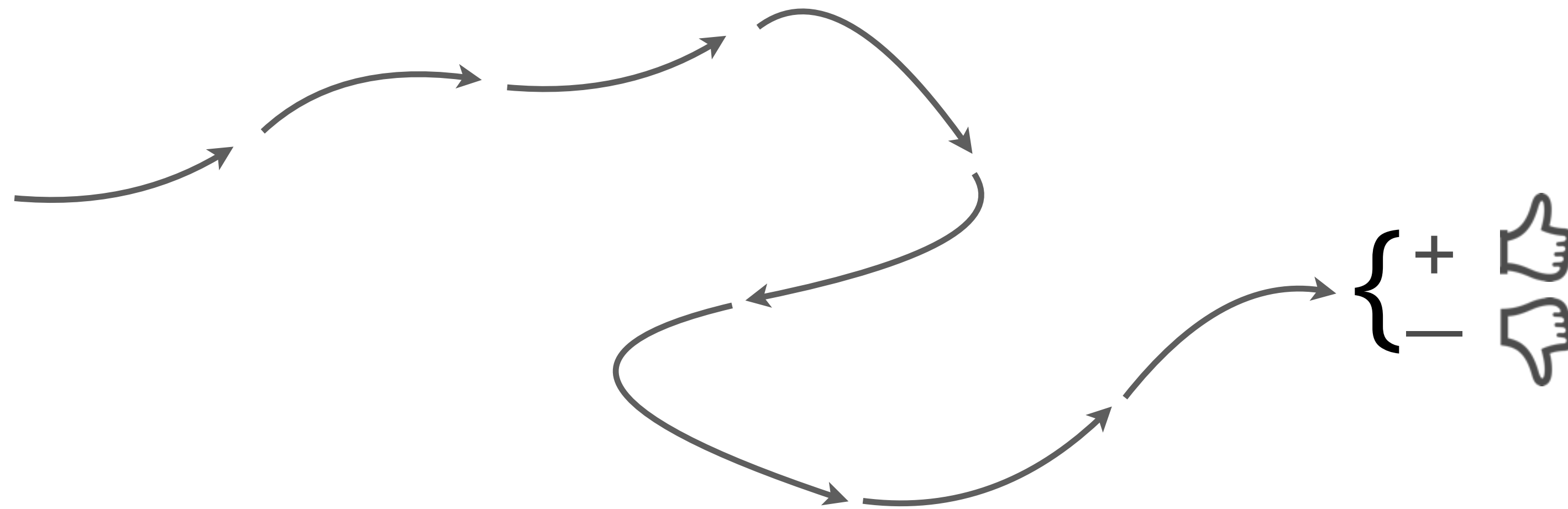
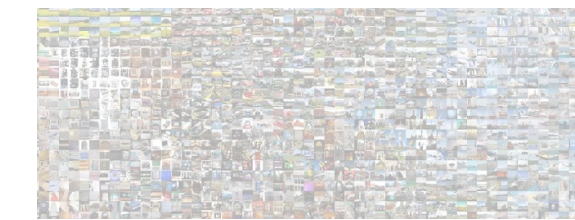
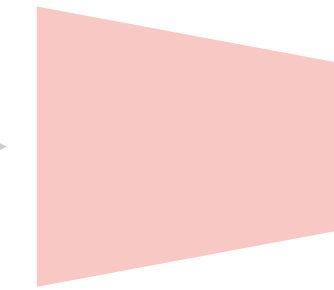
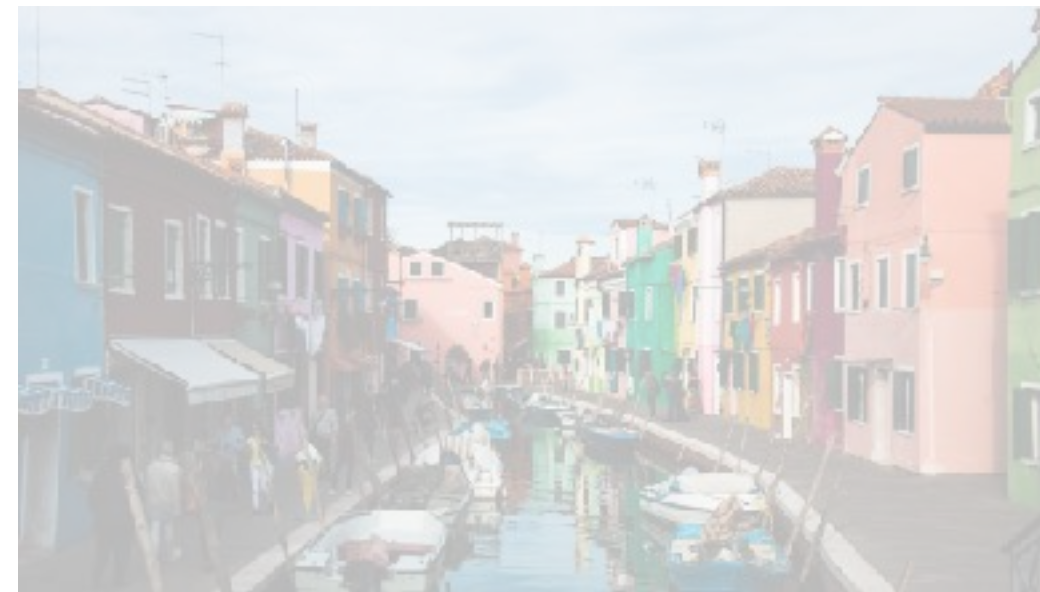
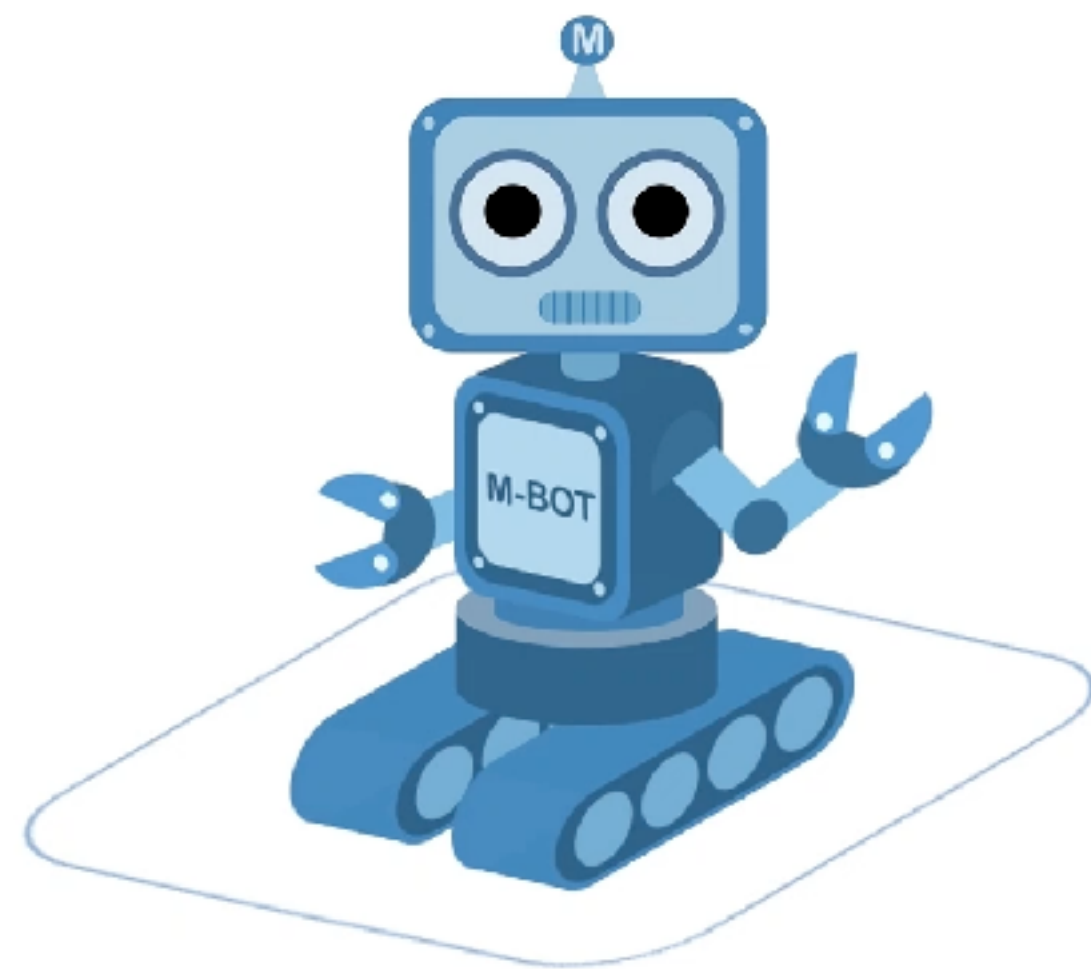
Challenges

Egocentric vision
Active perception
Sparse rewards



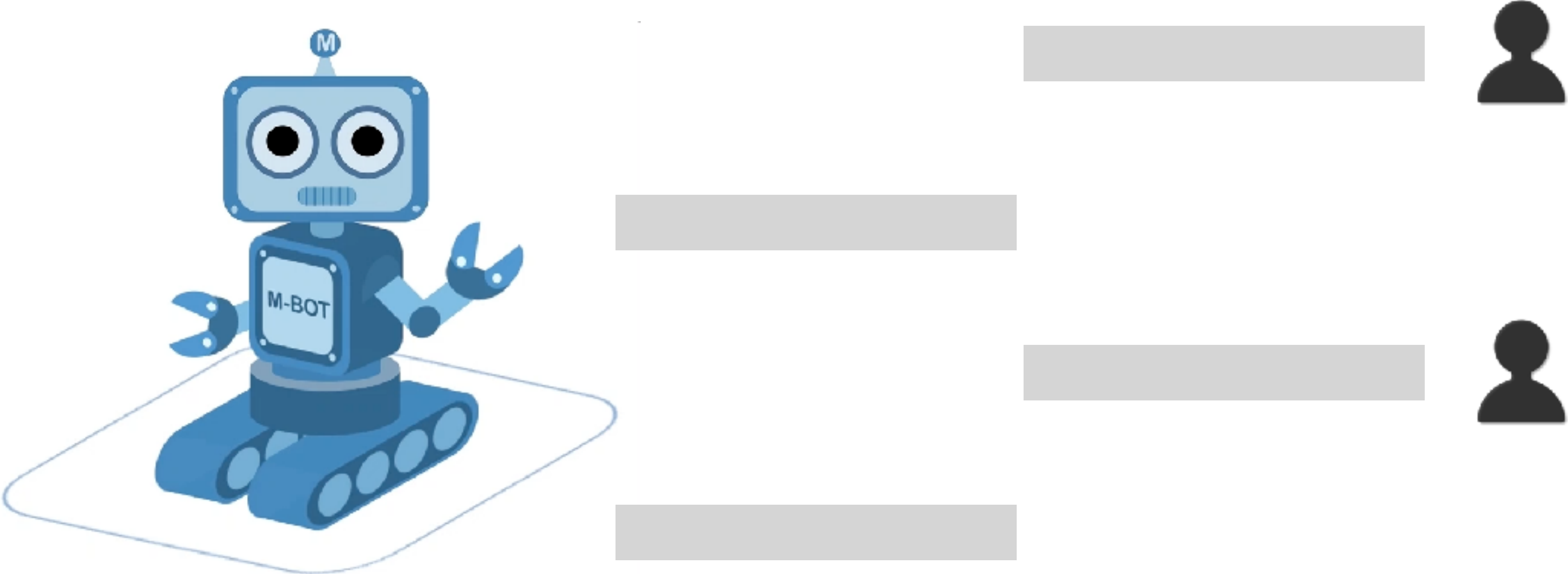
Challenges

Egocentric vision
Active perception
Sparse rewards

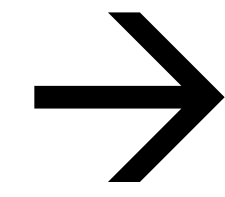


Challenges

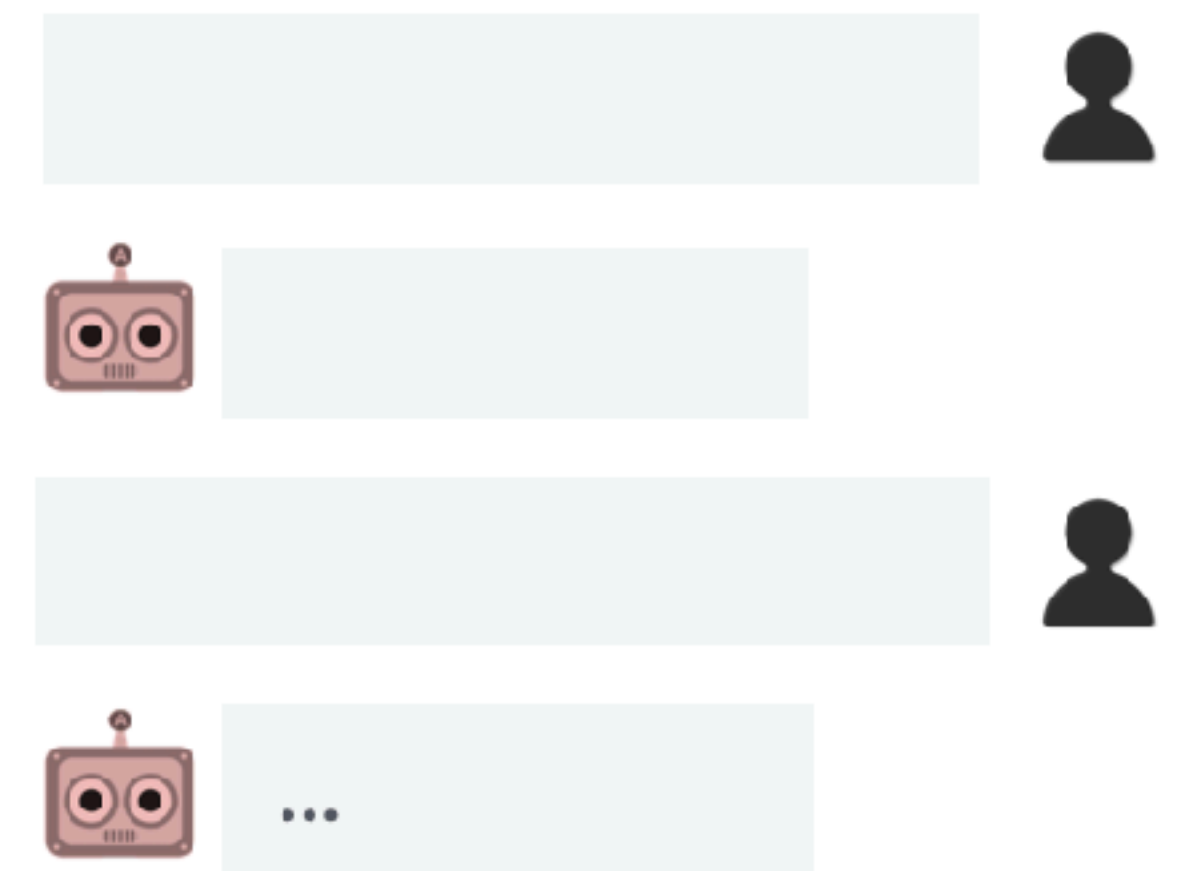
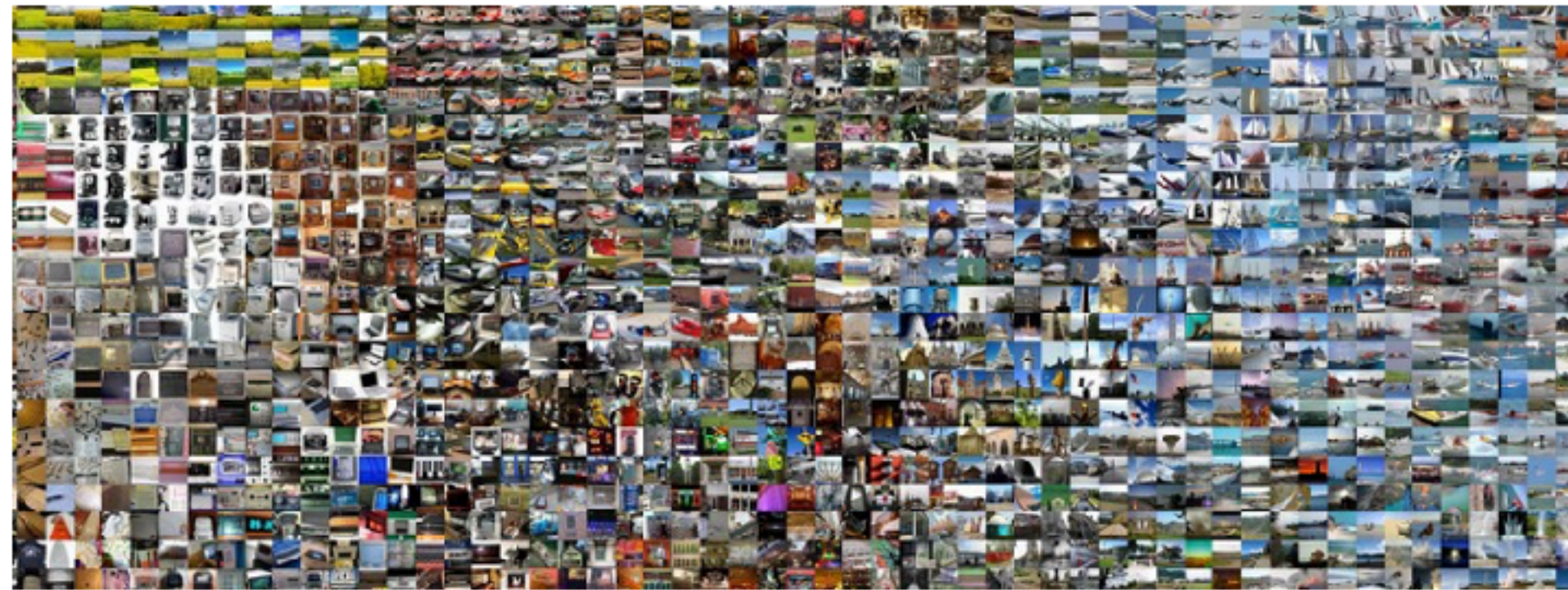
- Egocentric vision
- Active perception
- Sparse rewards
- Language understanding



Internet AI



Embodied AI





Habitat



Manolis Savva^{1,4*}

Abhishek Kadian^{1*}

Oleksandr Maksymets^{1*}

Yili Zhao¹

Erik Wijmans^{1,2,3}

Bhavana Jain¹



Julian Straub²

Jia Liu¹

Vladlen Koltun⁵

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”

Standardizing the Embodied AI “software stack”

Tasks



EmbodiedQA
(Das et al., 2018)

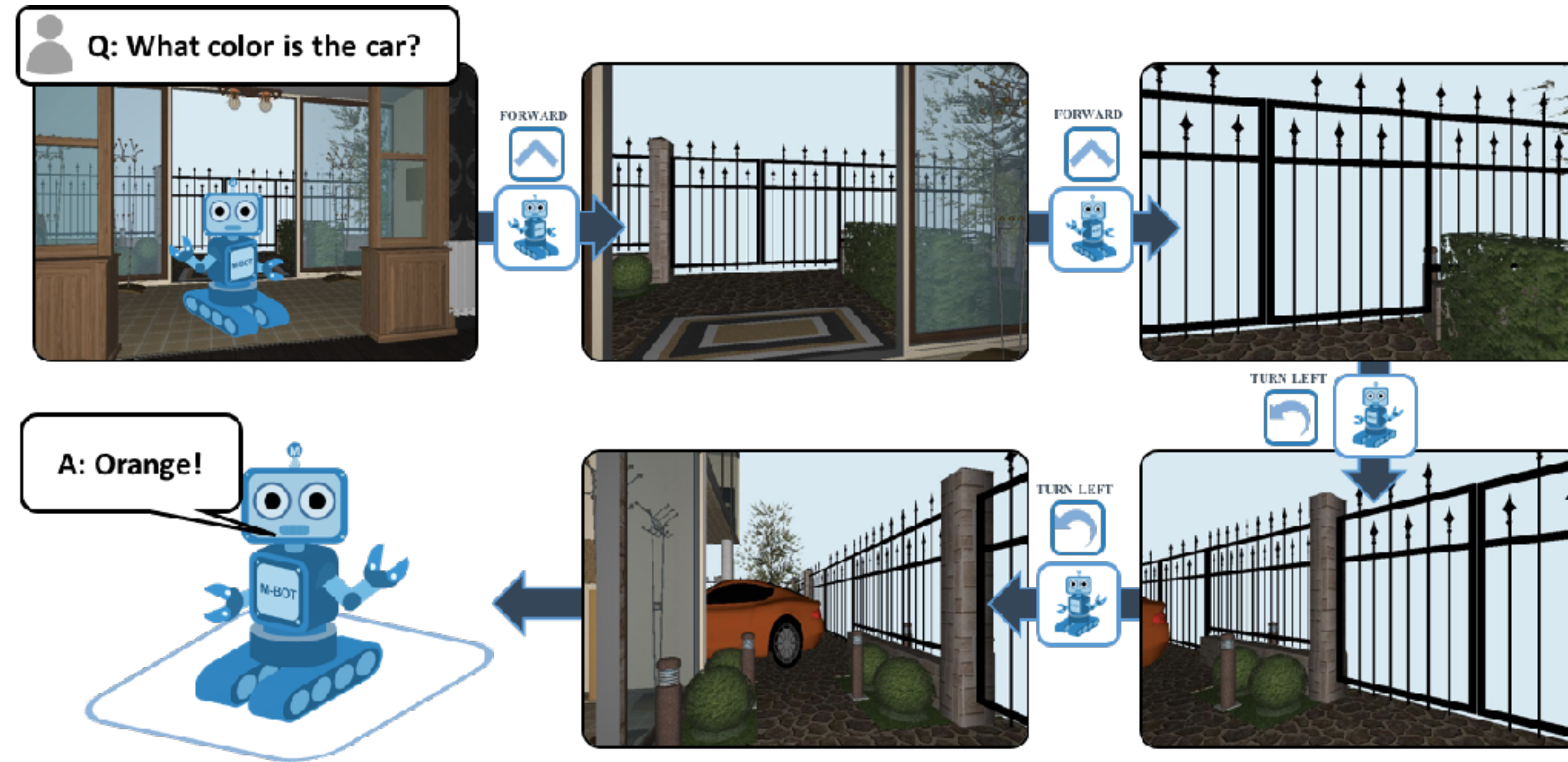
Language grounding
(Hill et al., 2017)

Interactive QA
(Gordon et al., 2018)

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

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

Standardizing the Embodied AI “software stack”



EmbodiedQA
(Das et al., 2018)

Standardizing the Embodied AI “software stack”

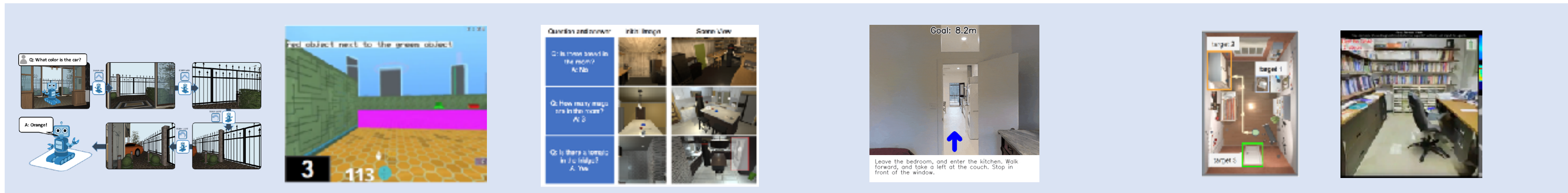


Leave the bedroom, and enter the kitchen. Walk forward, and take a left at the couch. Stop in front of the window.

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

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)

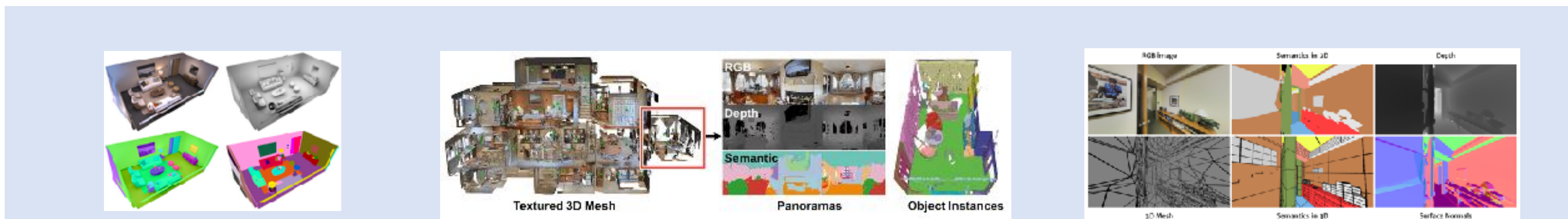
AI2-THOR (Kolve et al., 2017)

MINOS (Savva et al., 2017)

Gibson (Zamir et al., 2018)

CHALET (Yan et al., 2018)

Datasets

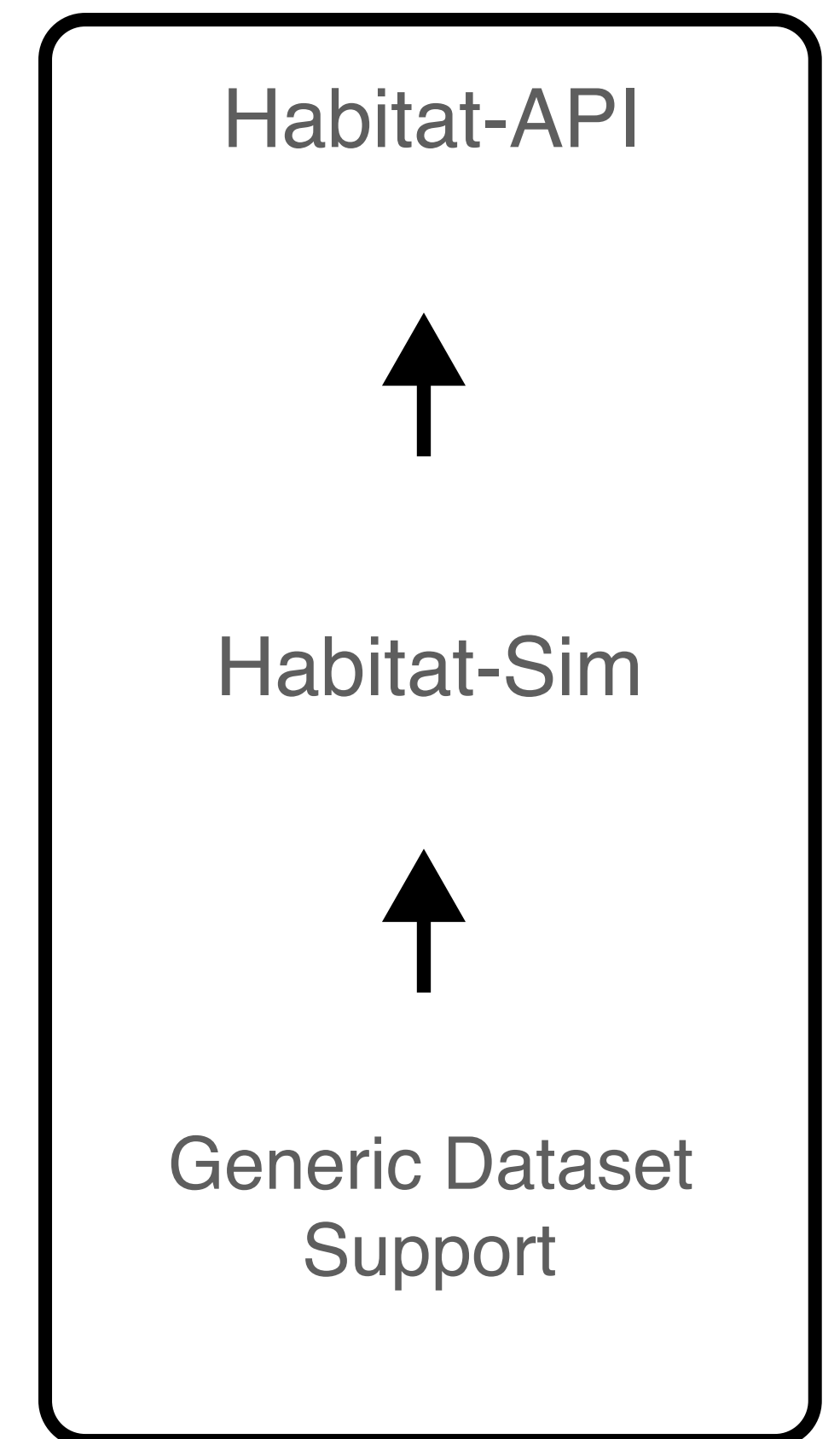


Replica (Straub et al., 2019)

Matterport3D (Chang et al., 2017)

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

Habitat Platform





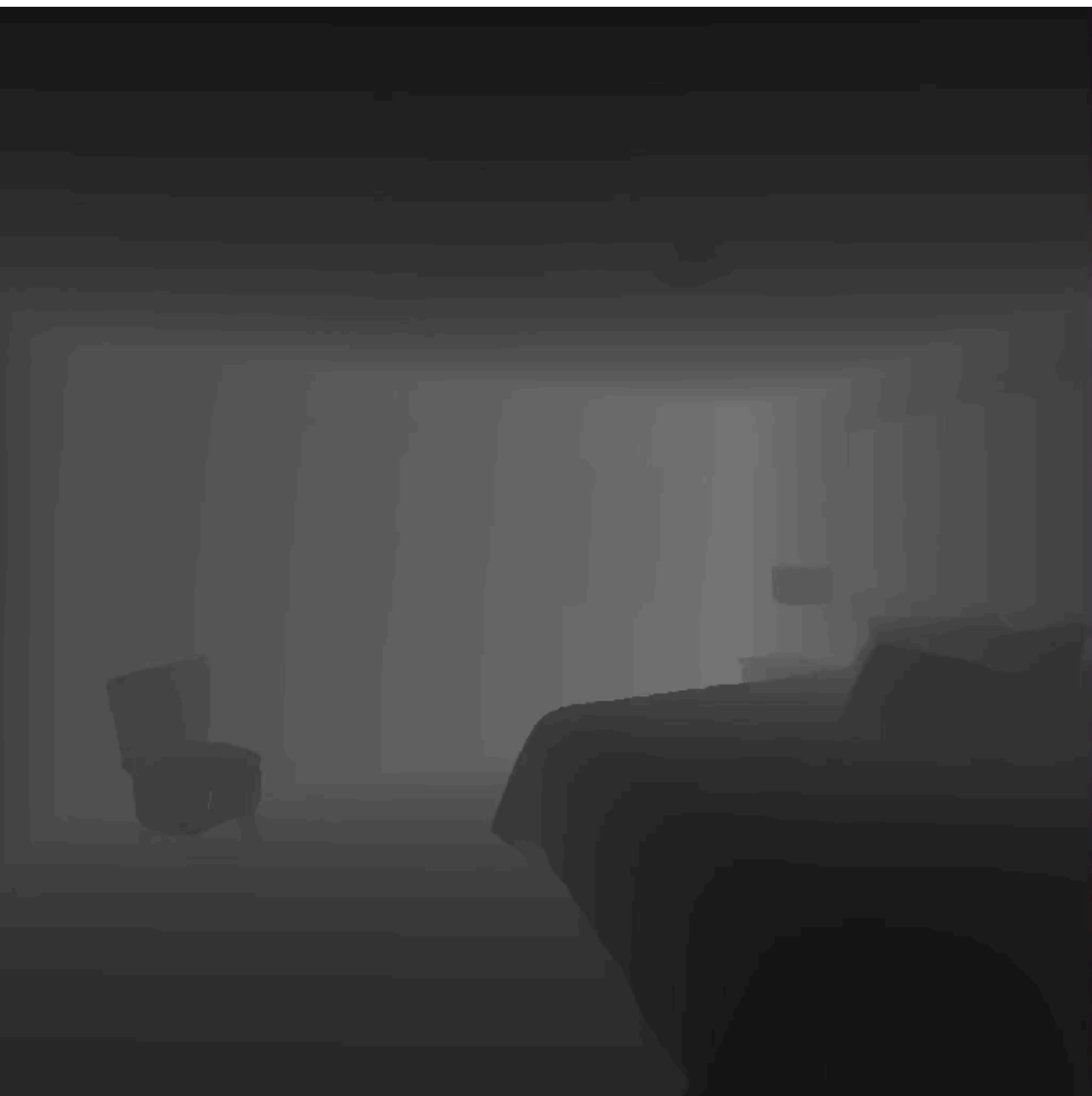
Habitat-Sim Demo



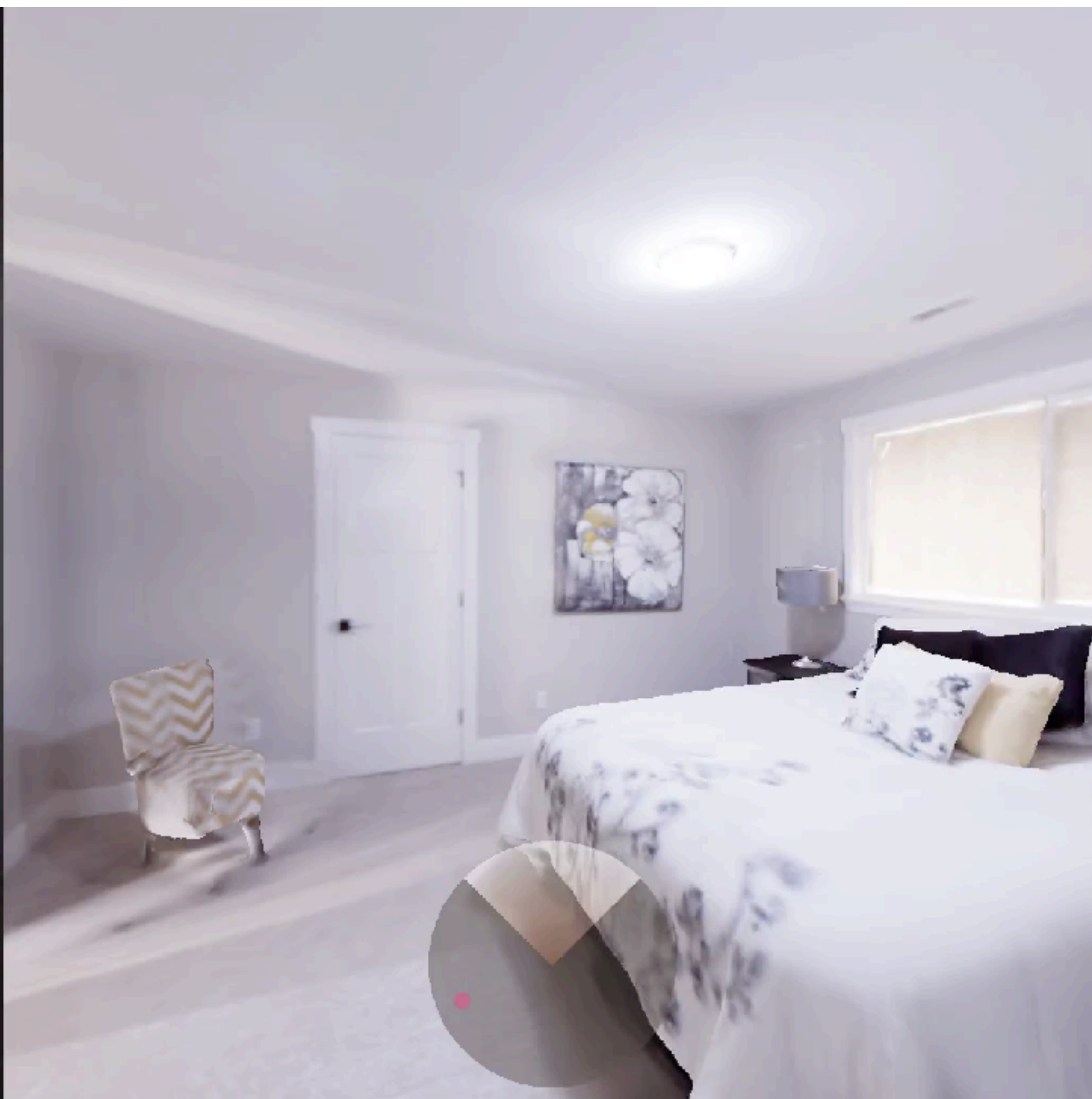
Habitat-API

PointGoal Navigation

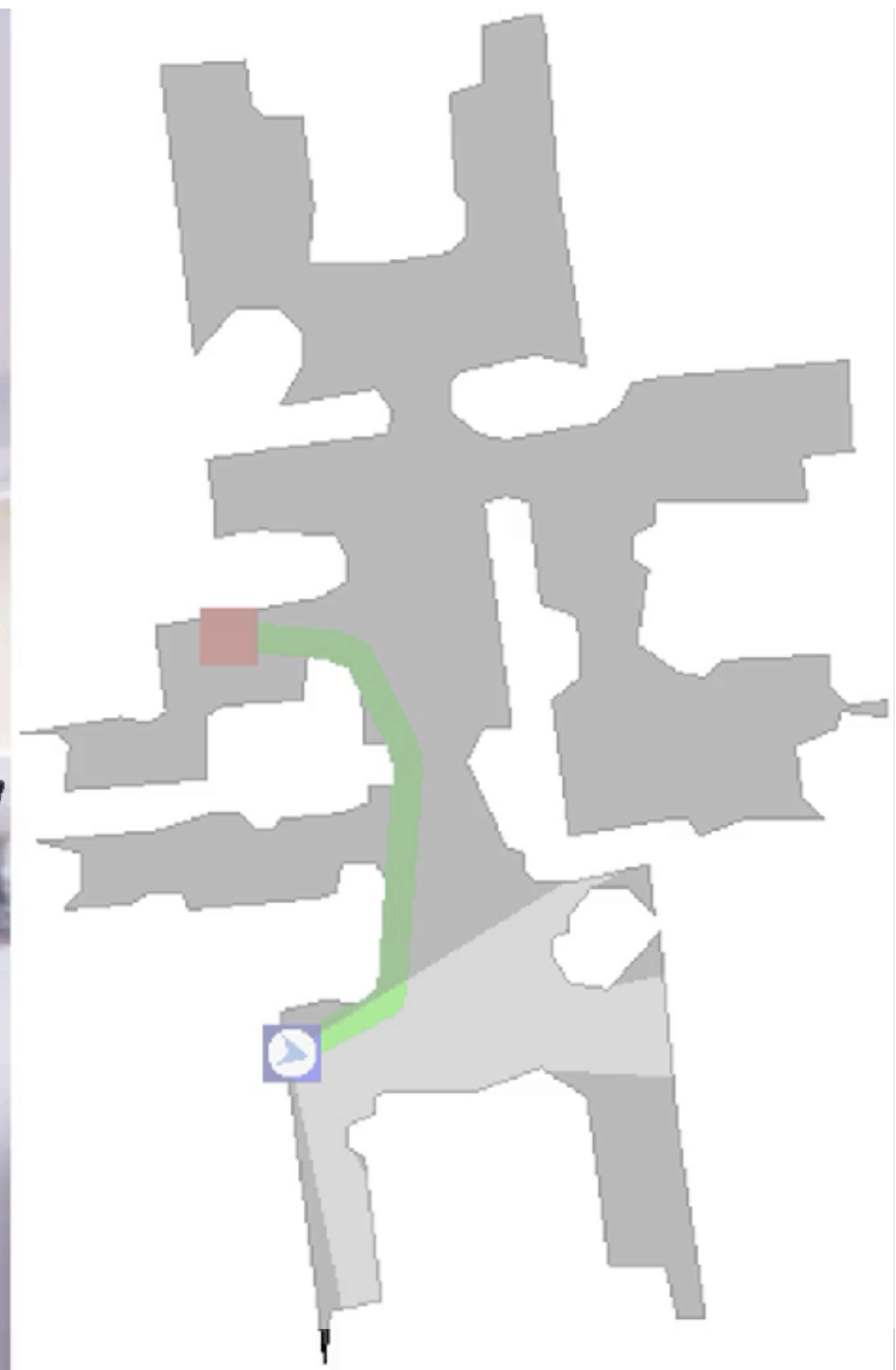
PointGoal Navigation



Depth

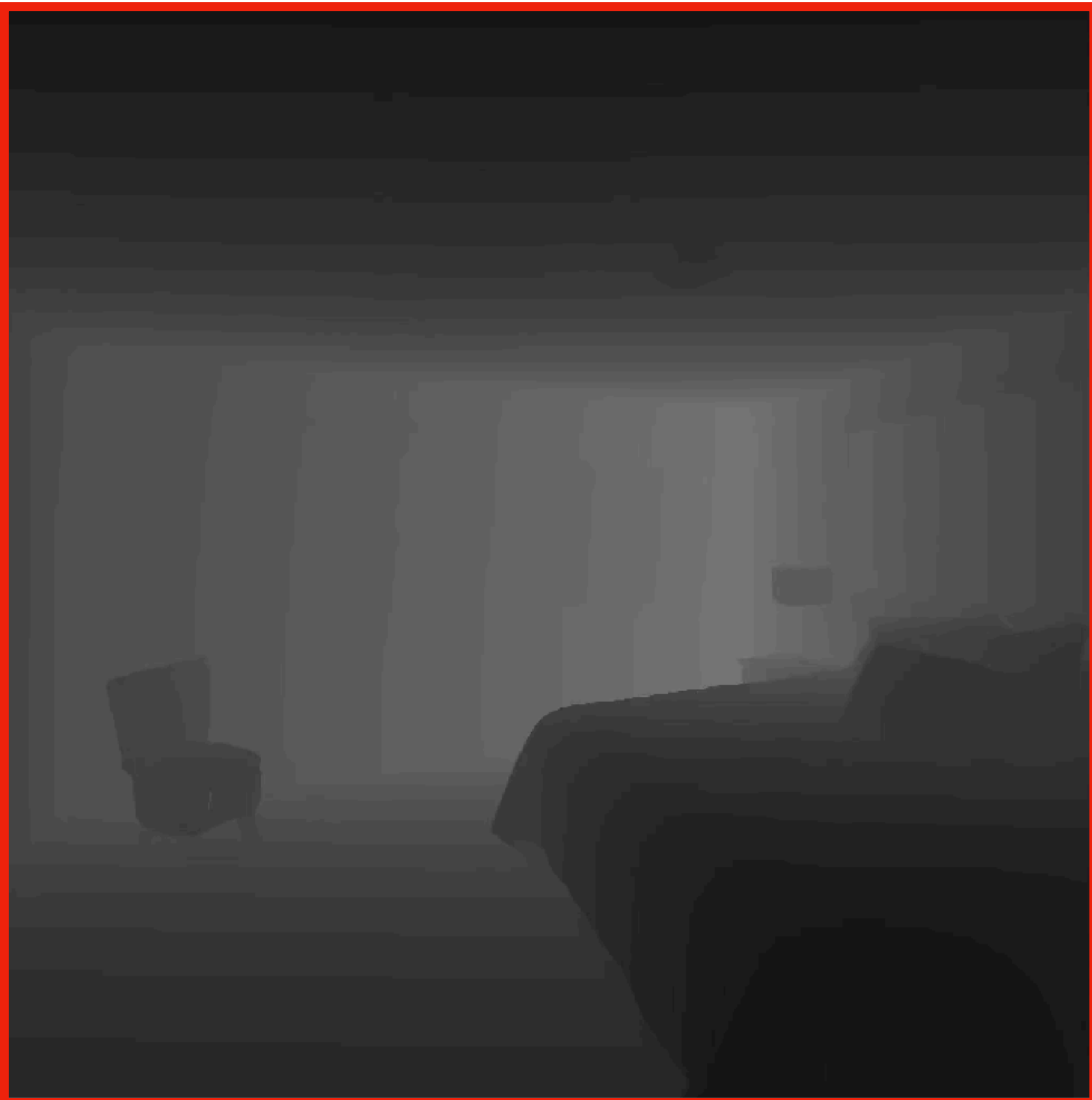


RGB and GPS+Compass



Top Down Map

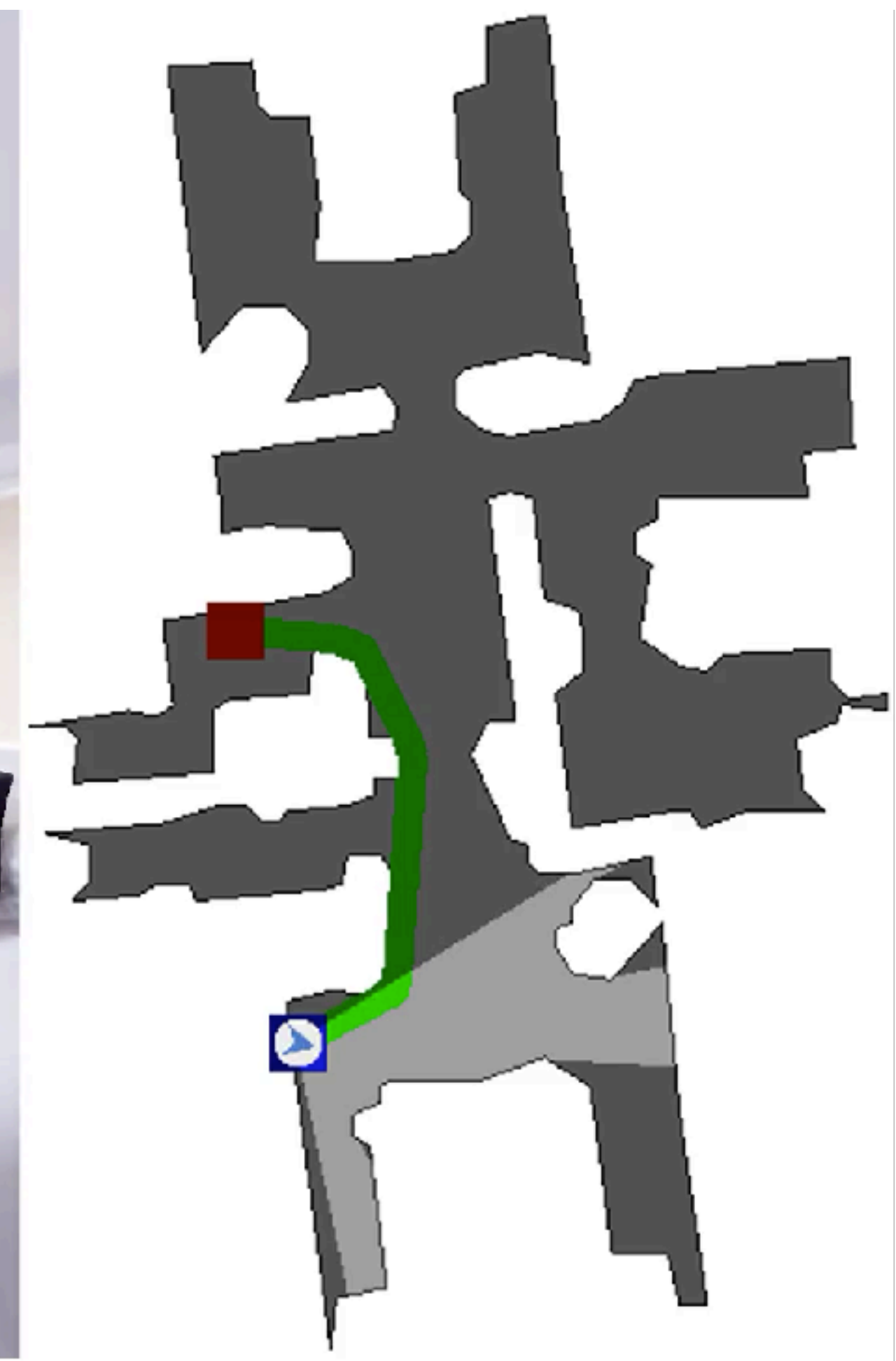
PointGoal Navigation



Depth

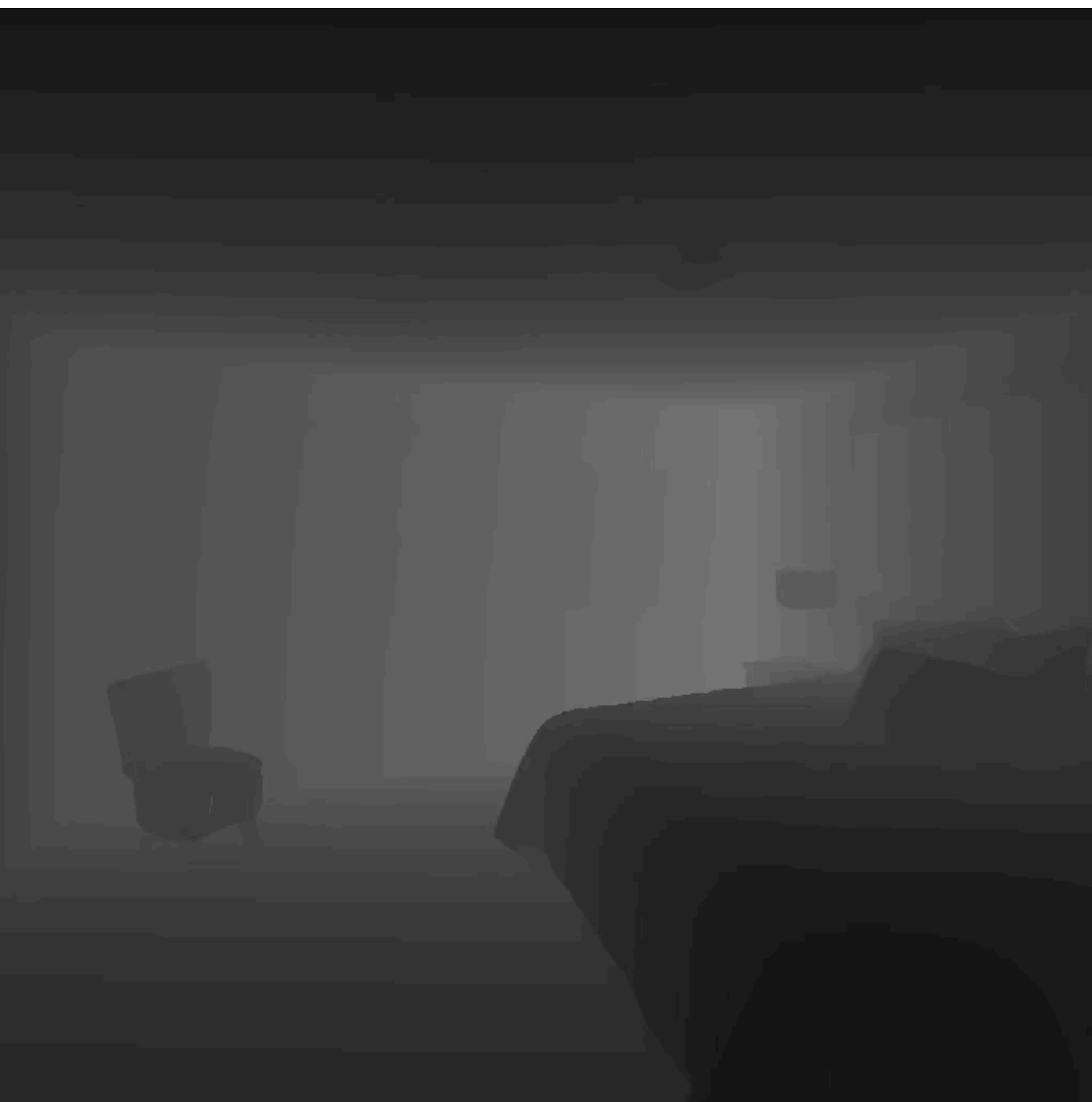


RGB and GPS+Compass



Top Down Map

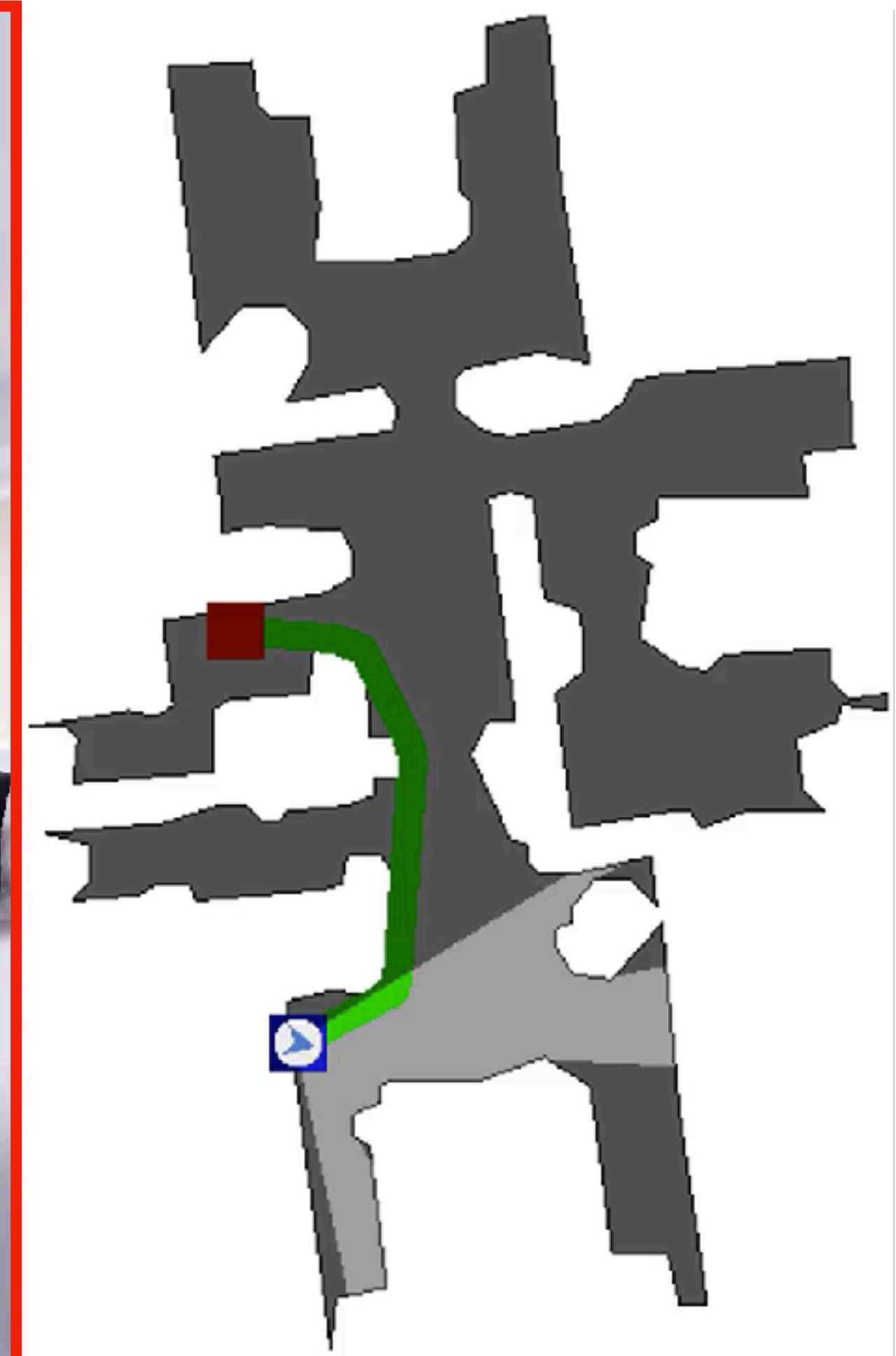
PointGoal Navigation



Depth

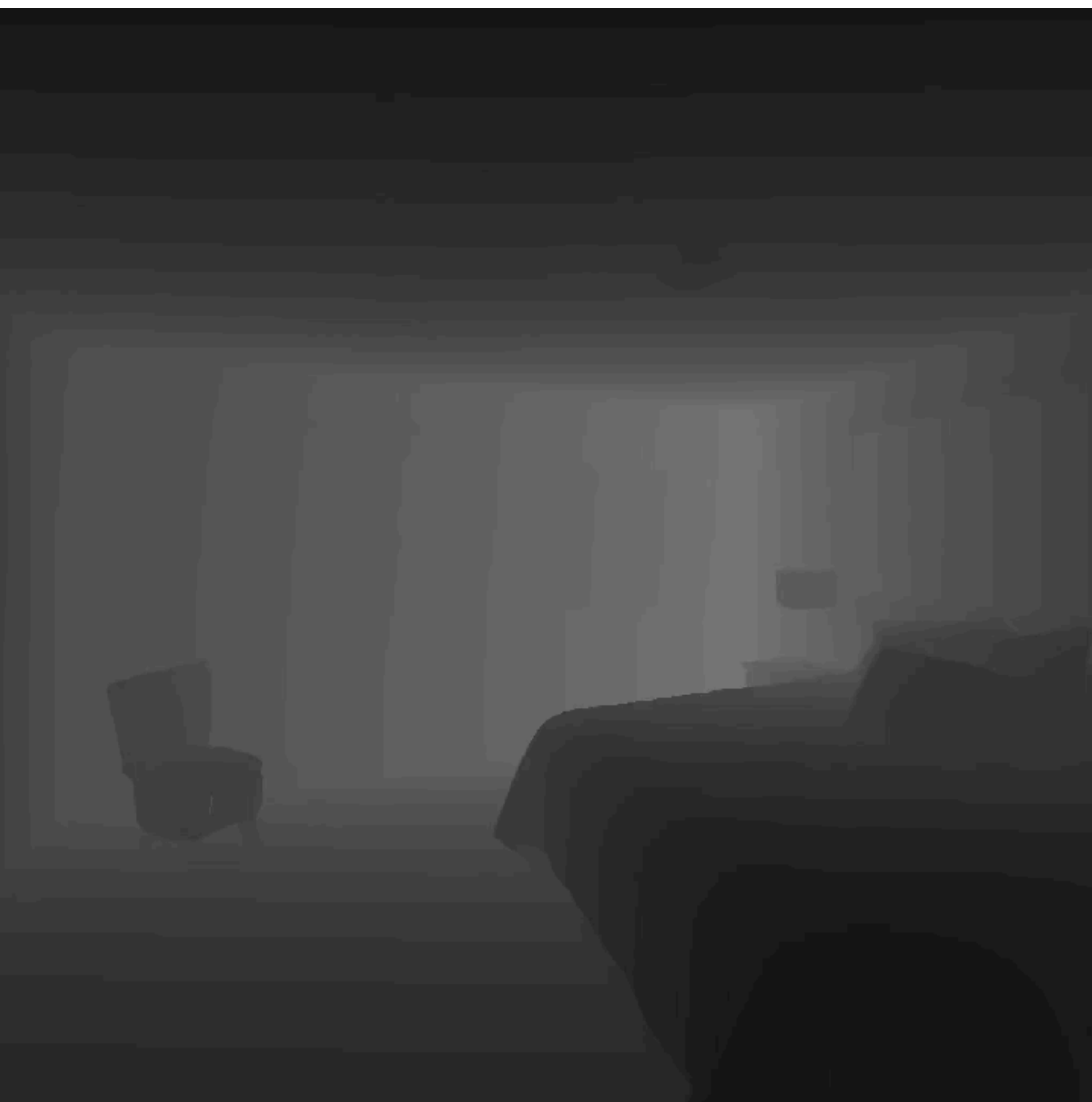


RGB and GPS+Compass

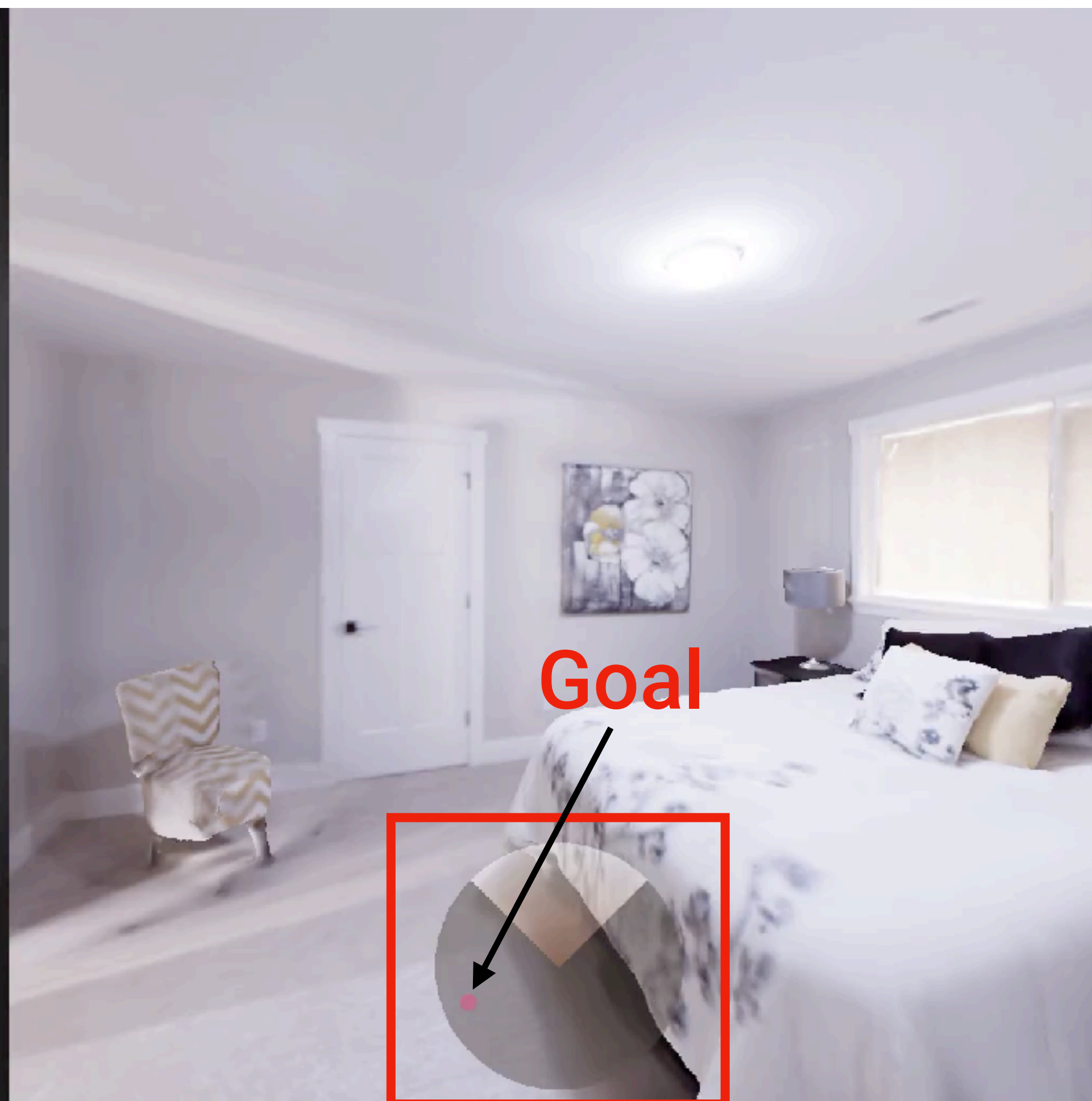


Top Down Map

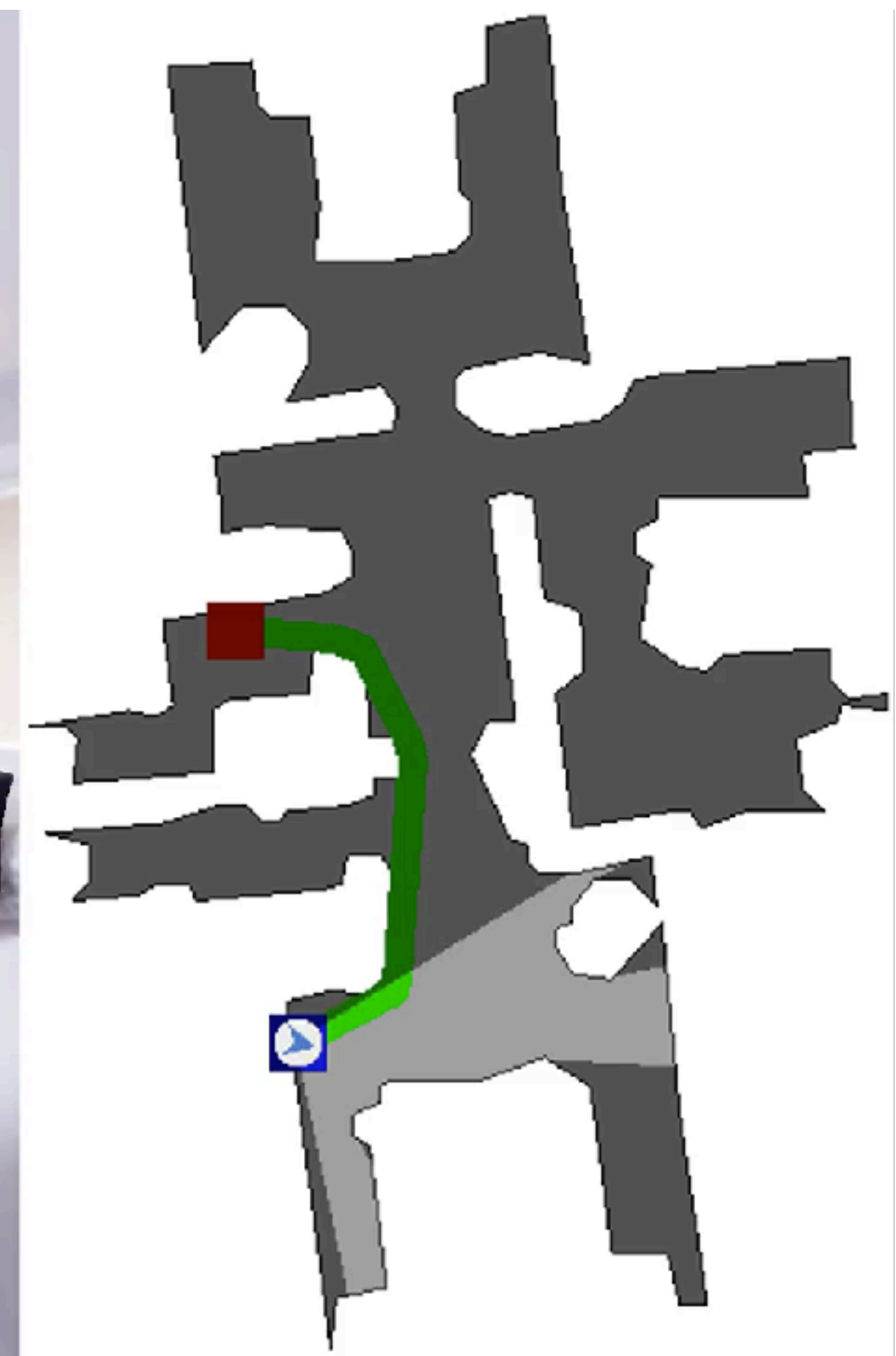
PointGoal Navigation



Depth

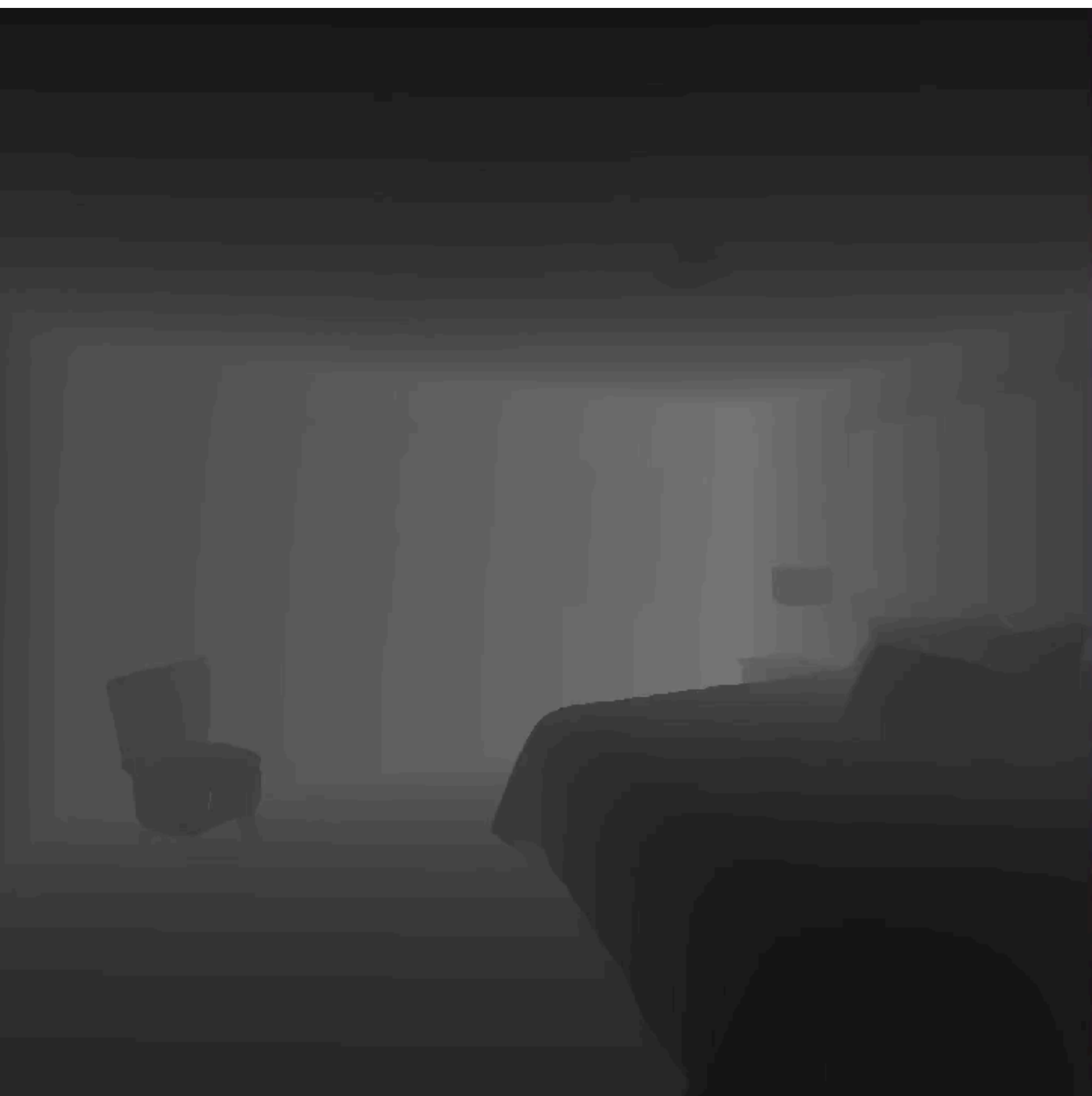


RGB and GPS+Compass



Top Down Map

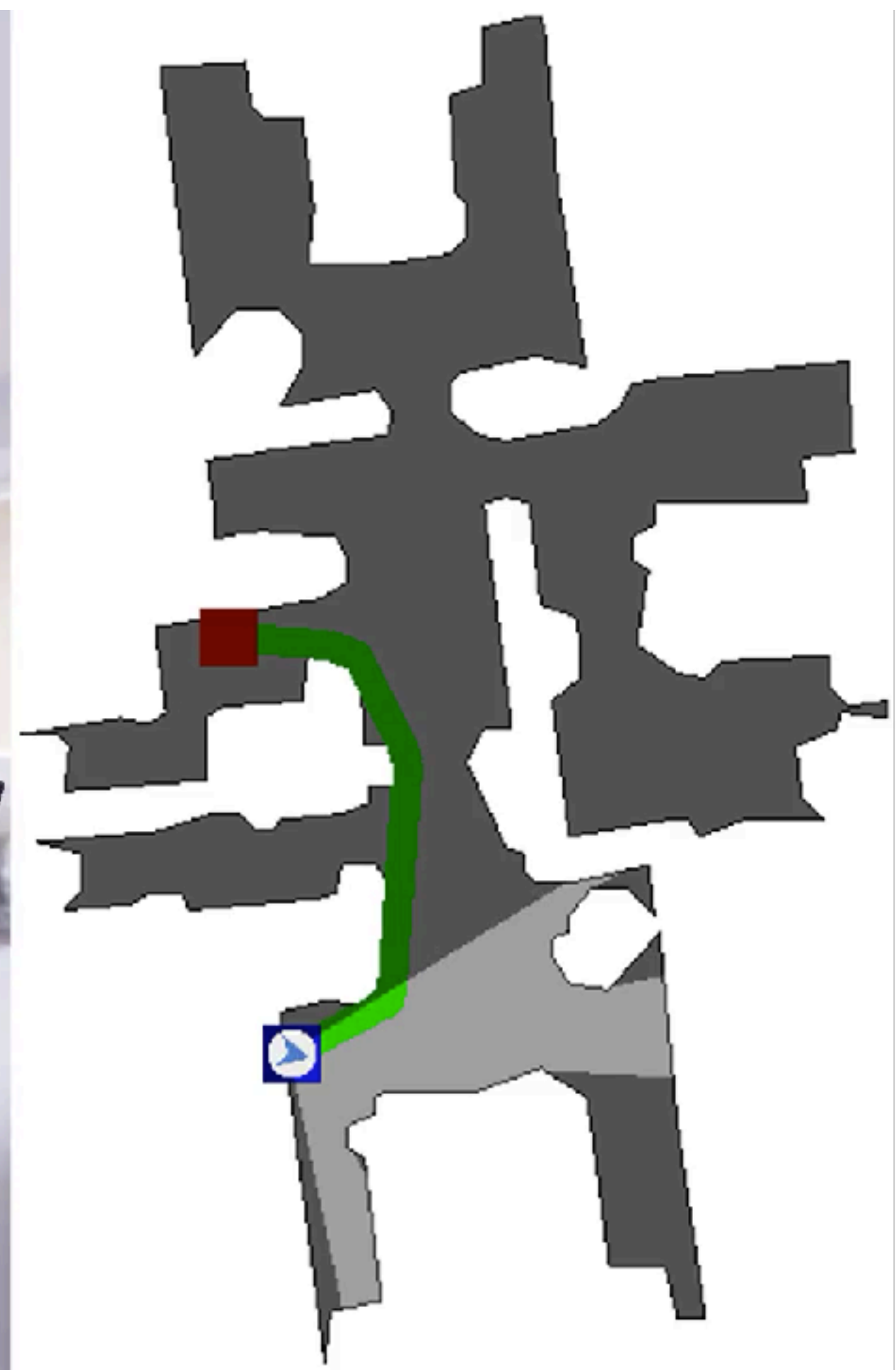
PointGoal Navigation



Depth



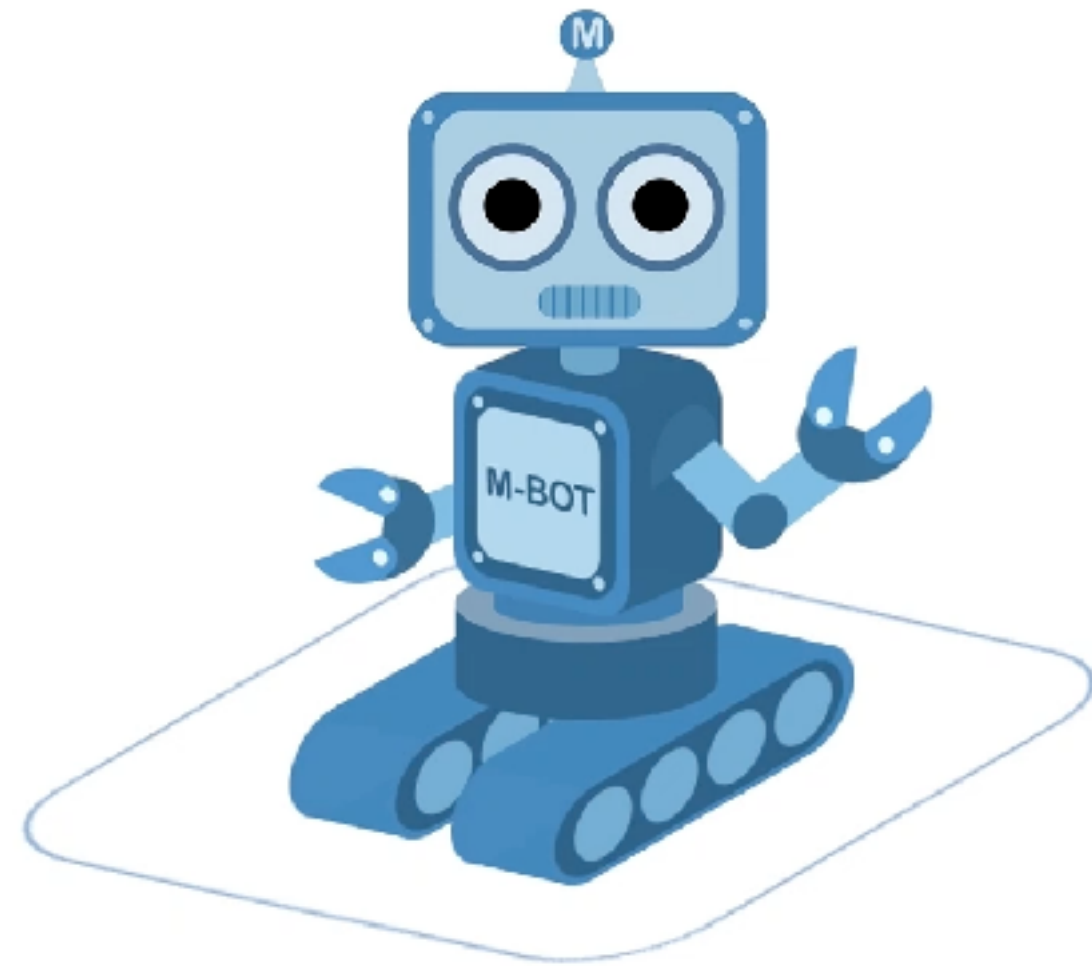
RGB and GPS+Compass



Top Down Map

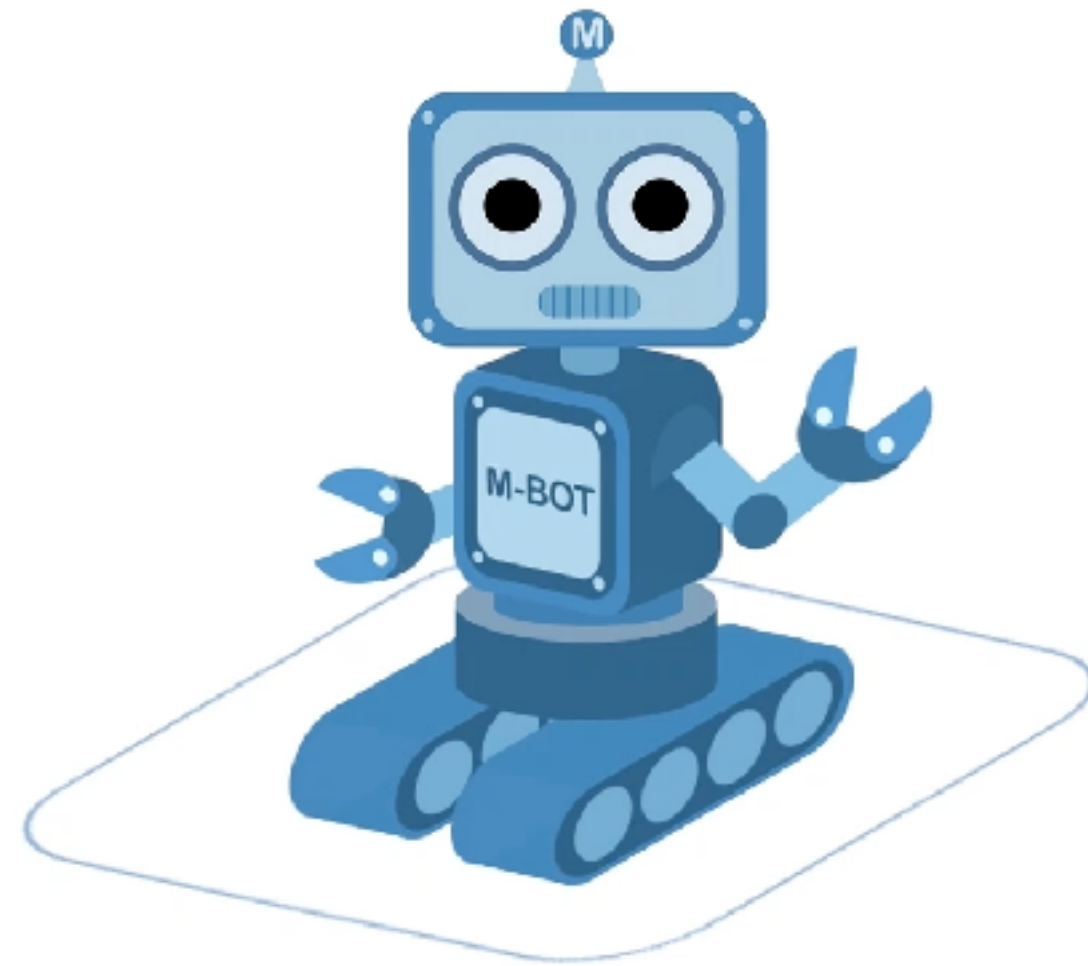
Agent and Model Design

Agent and Model Design

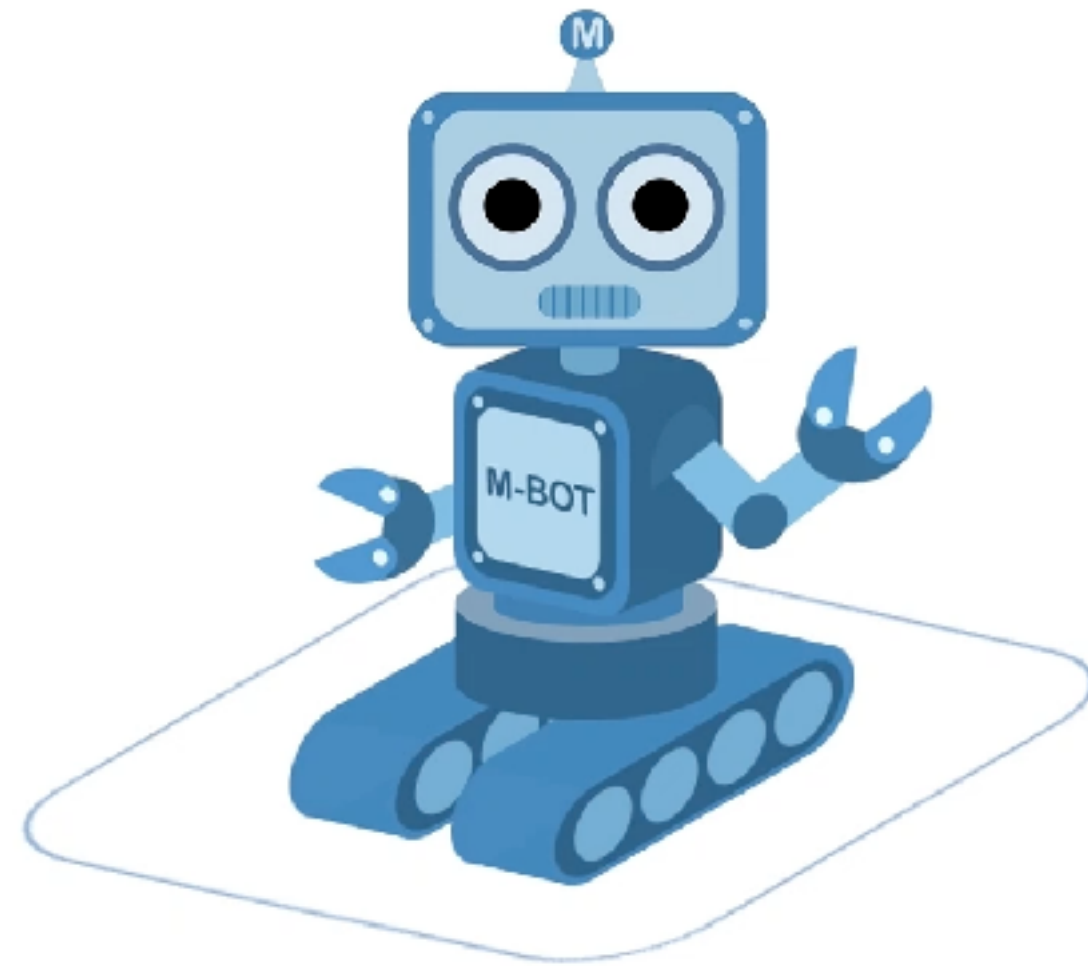


Agent and Model Design

- 1.25m tall cylinder with 0.1m radius

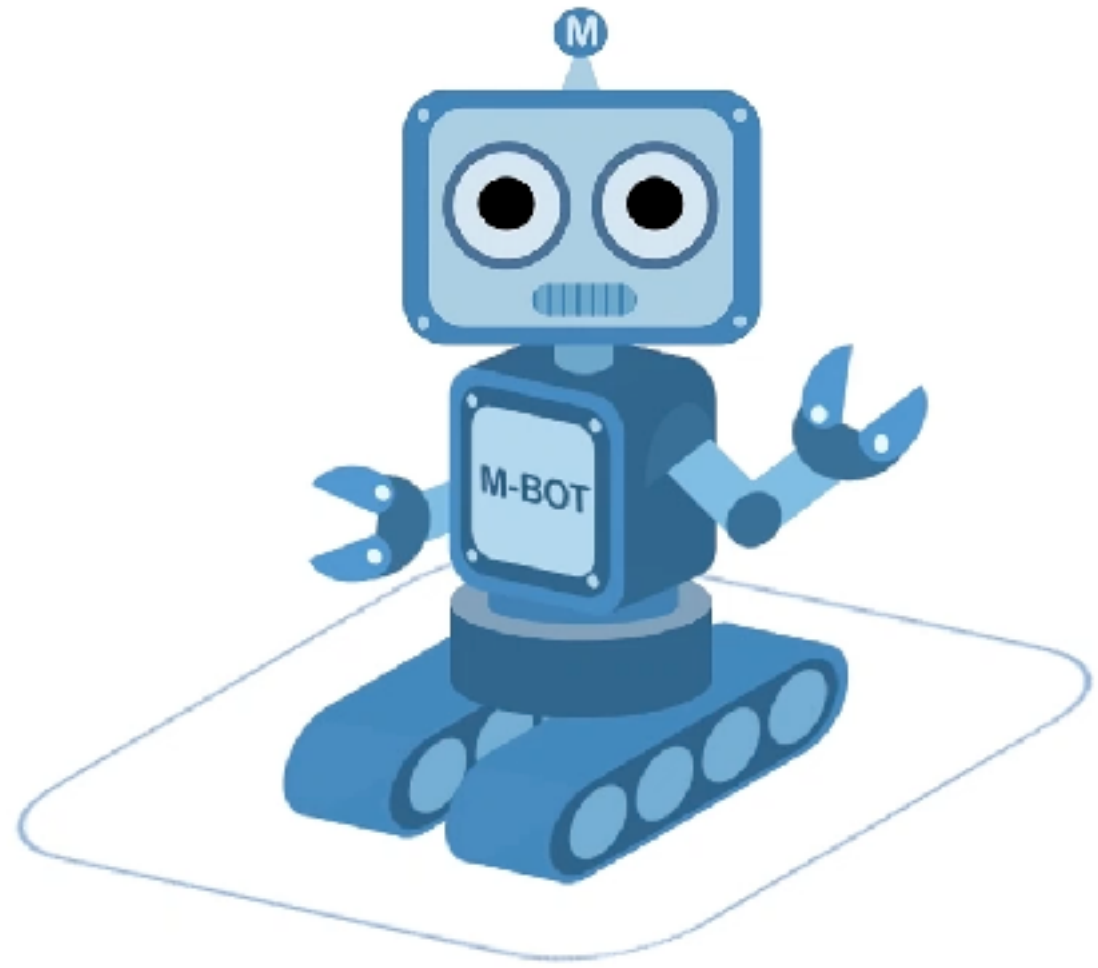


Agent and Model Design

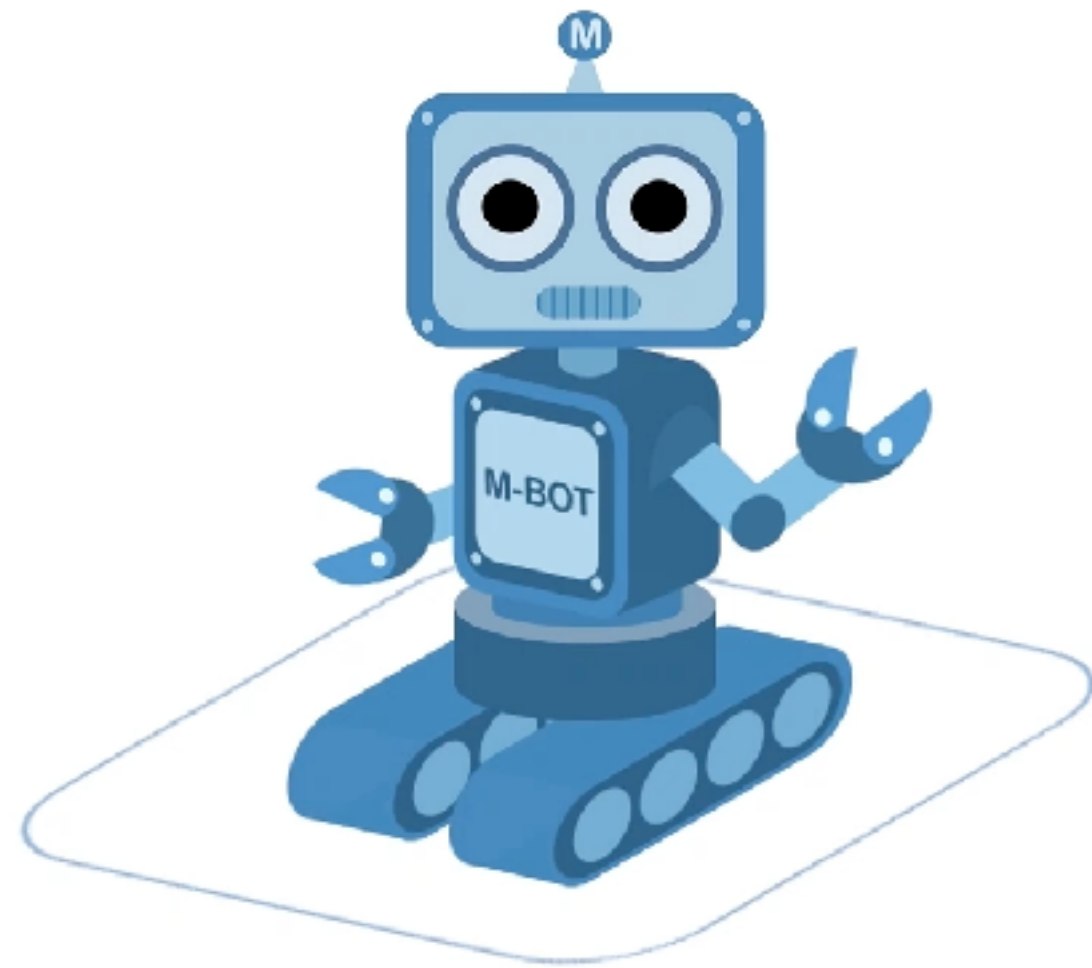
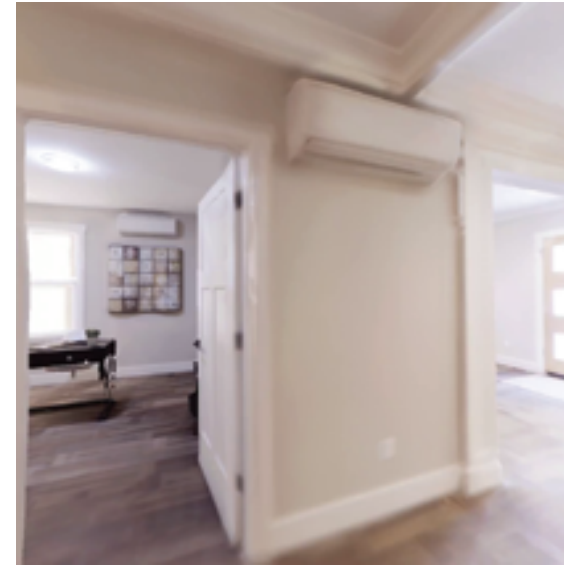


- 1.25m tall cylinder with 0.1m radius
- Actions:
 - <stop>: Indicates the agent believes it has completed the task
 - <forward>: Moves 0.25m forward
 - <left>, <right>: Turn 10 degrees

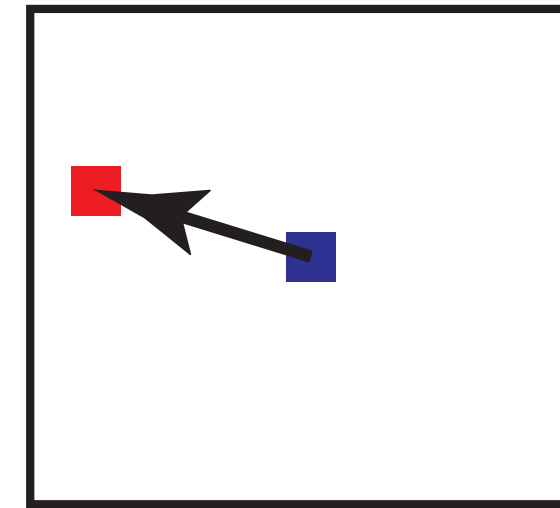
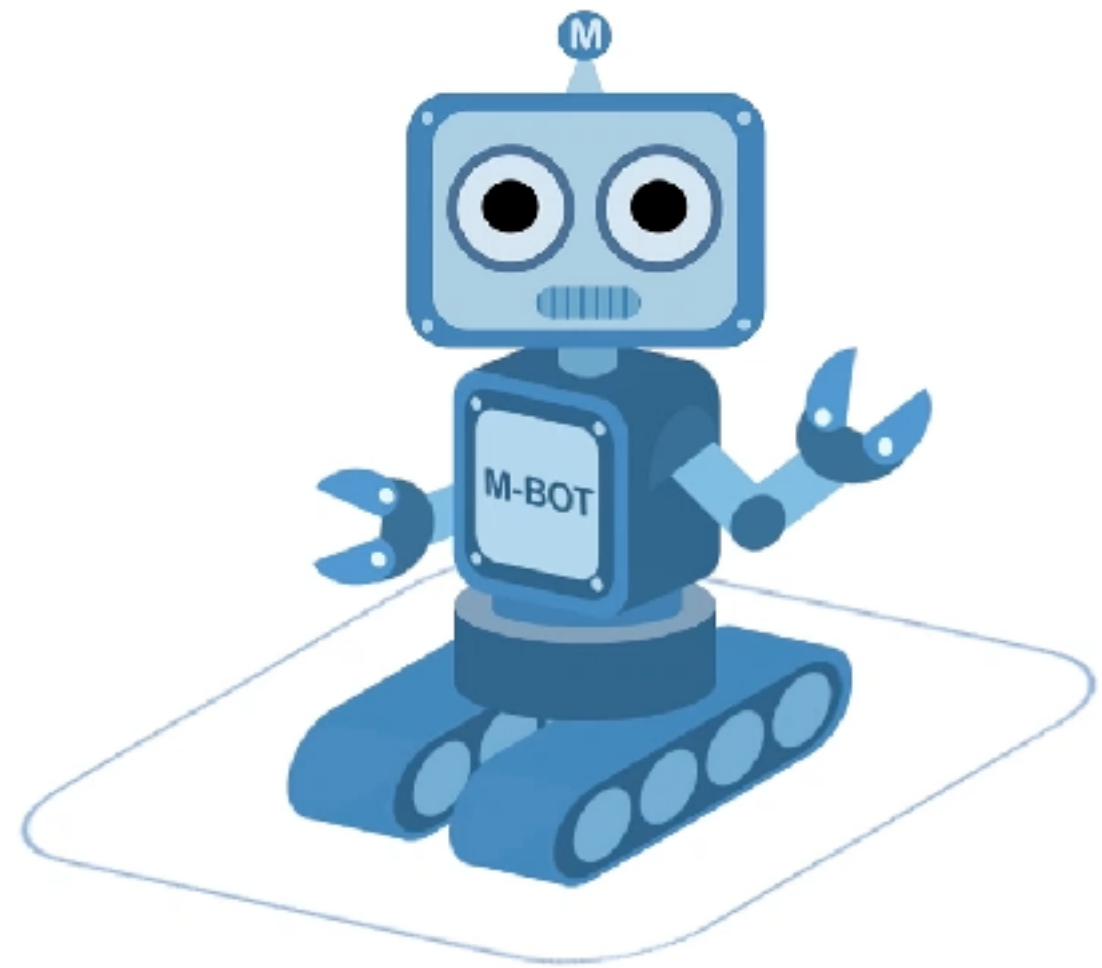
Agent and Model Design



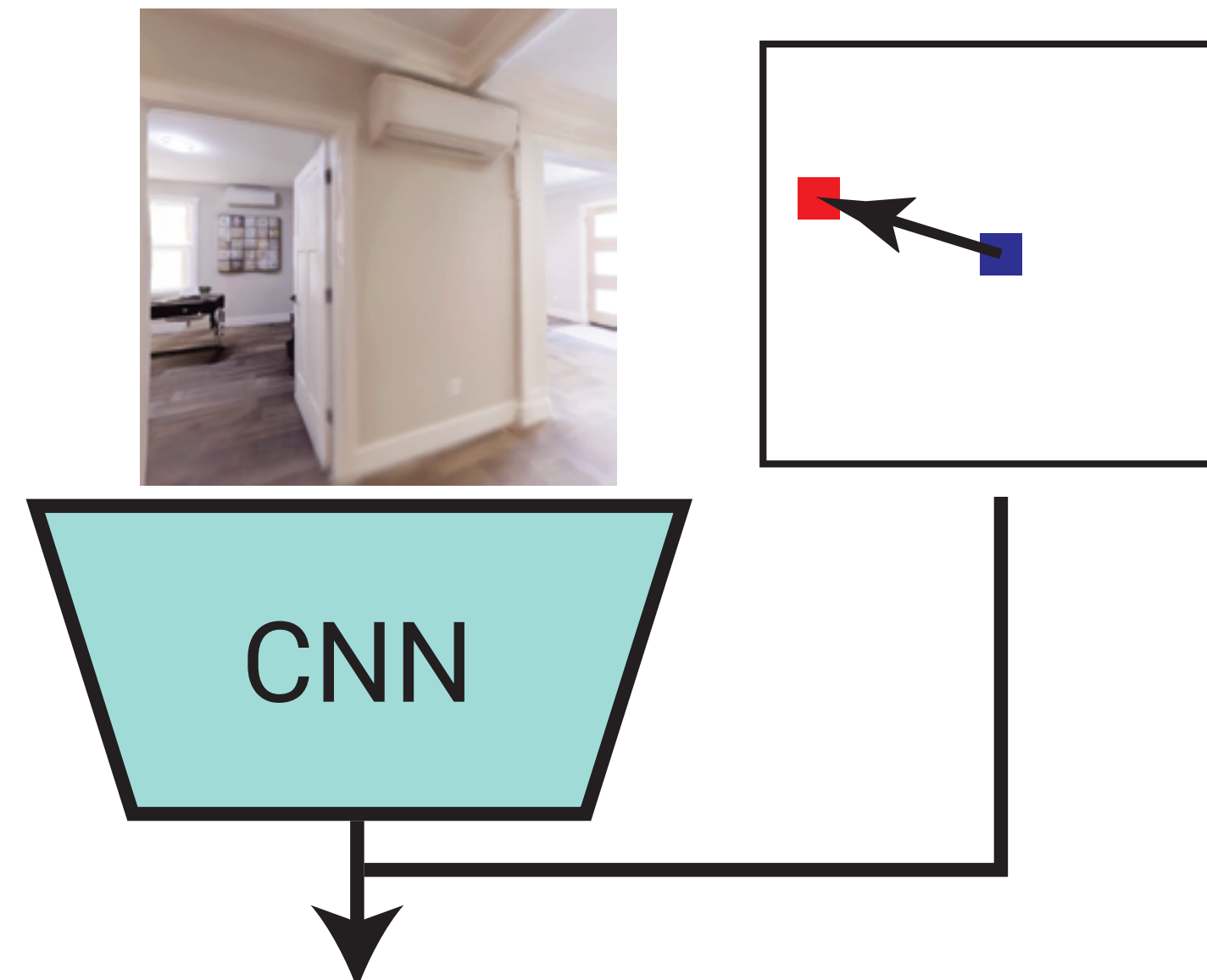
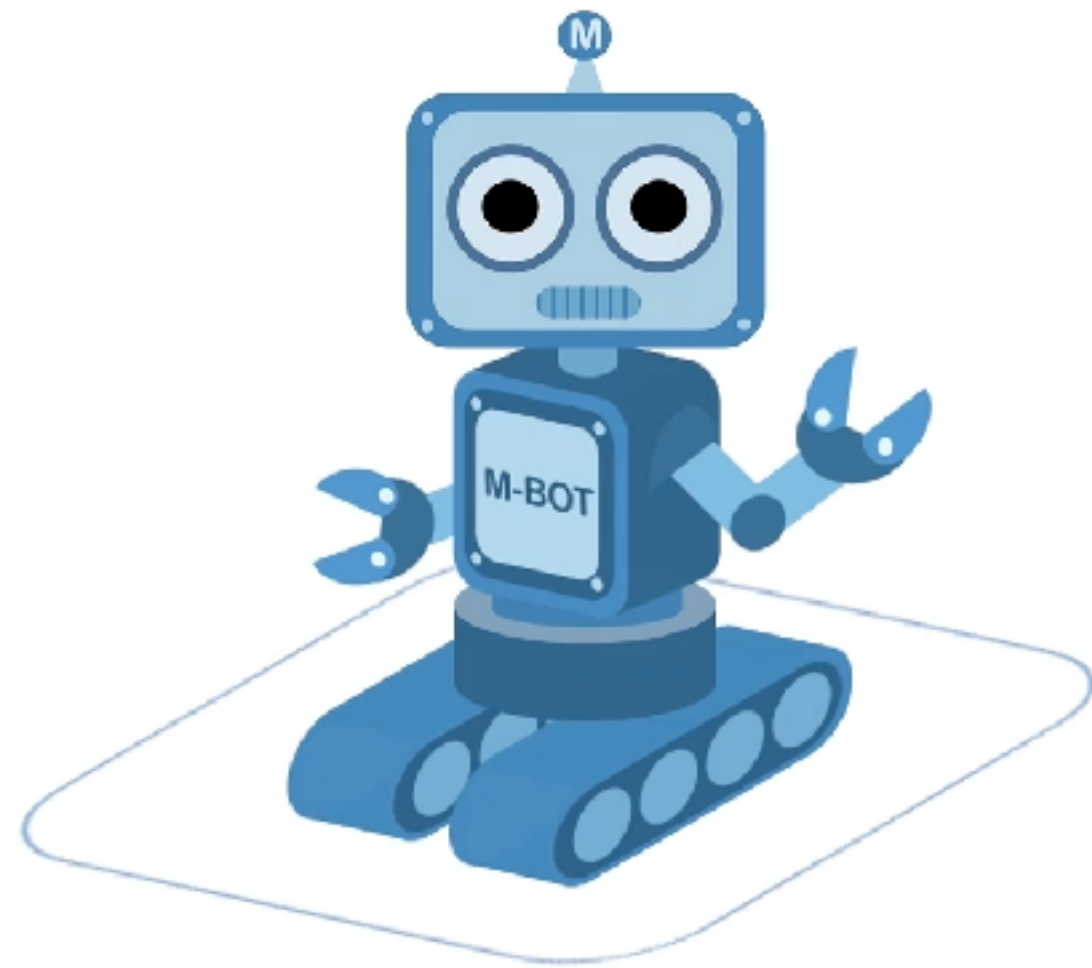
Agent and Model Design



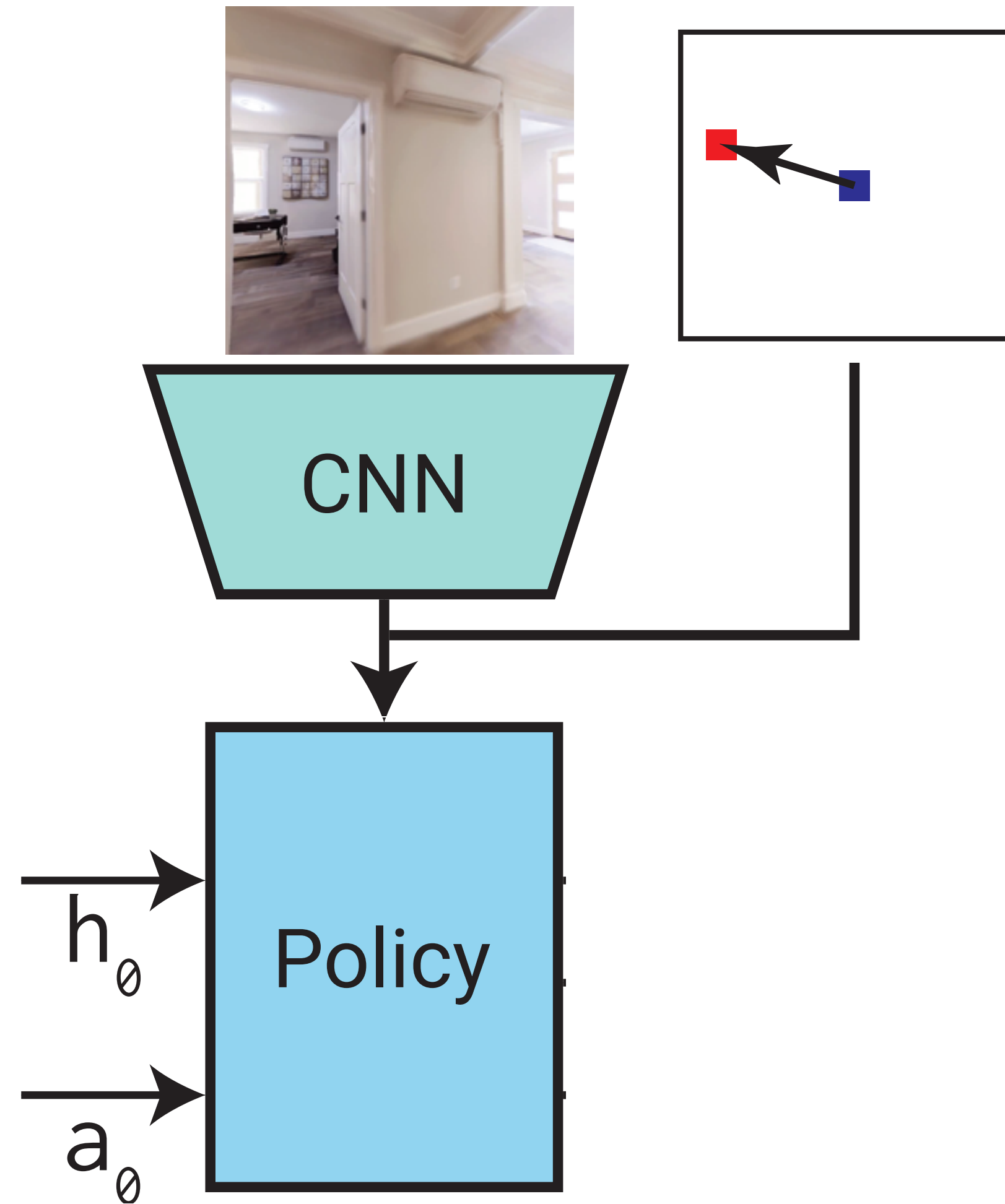
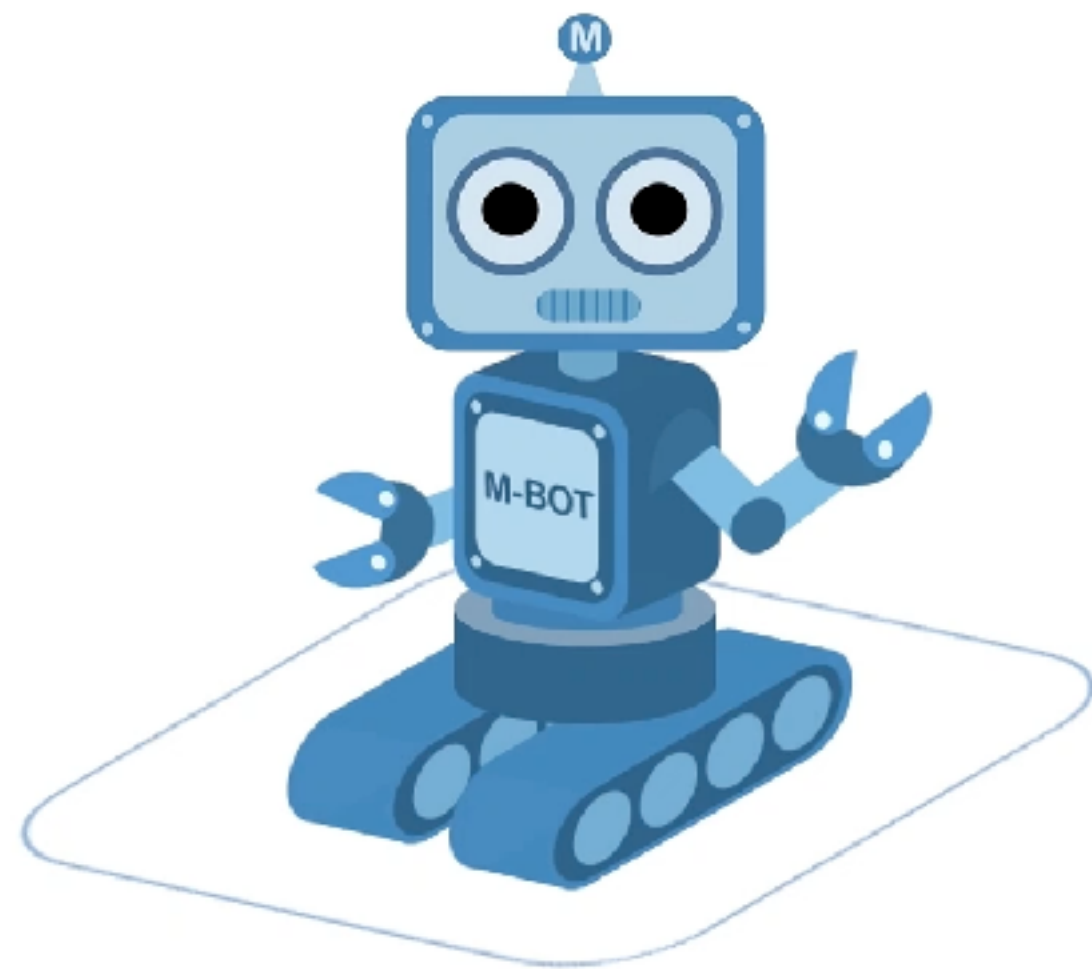
Agent and Model Design



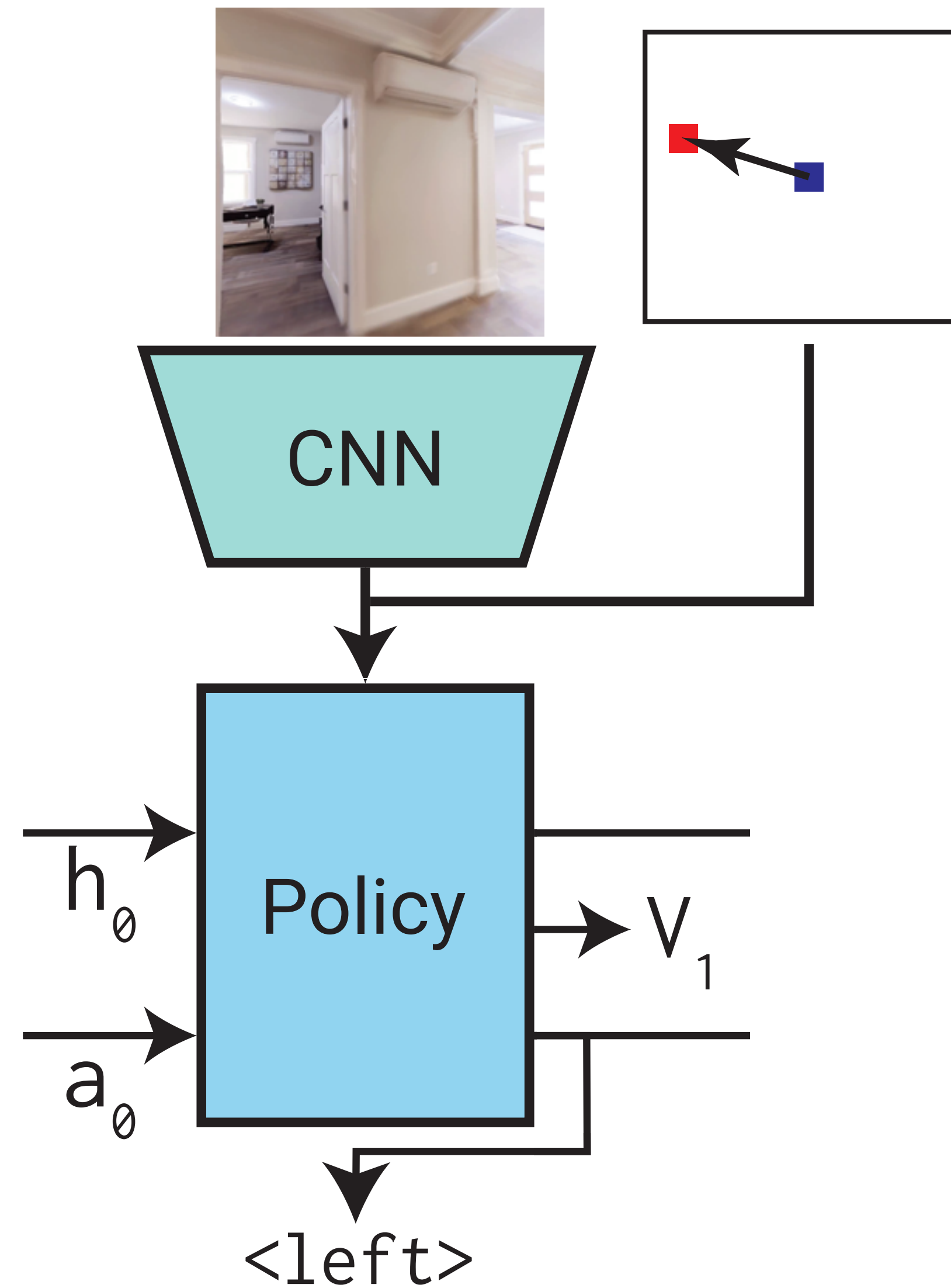
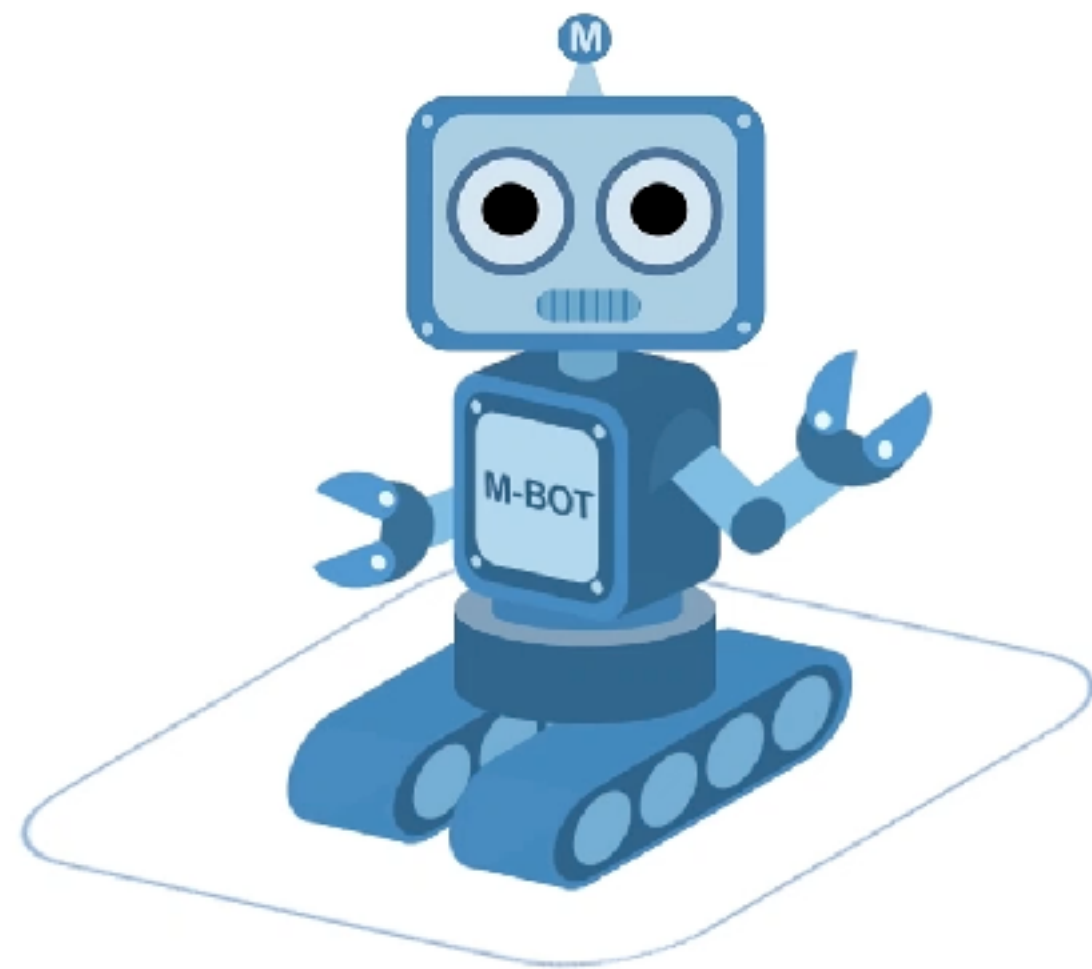
Agent and Model Design



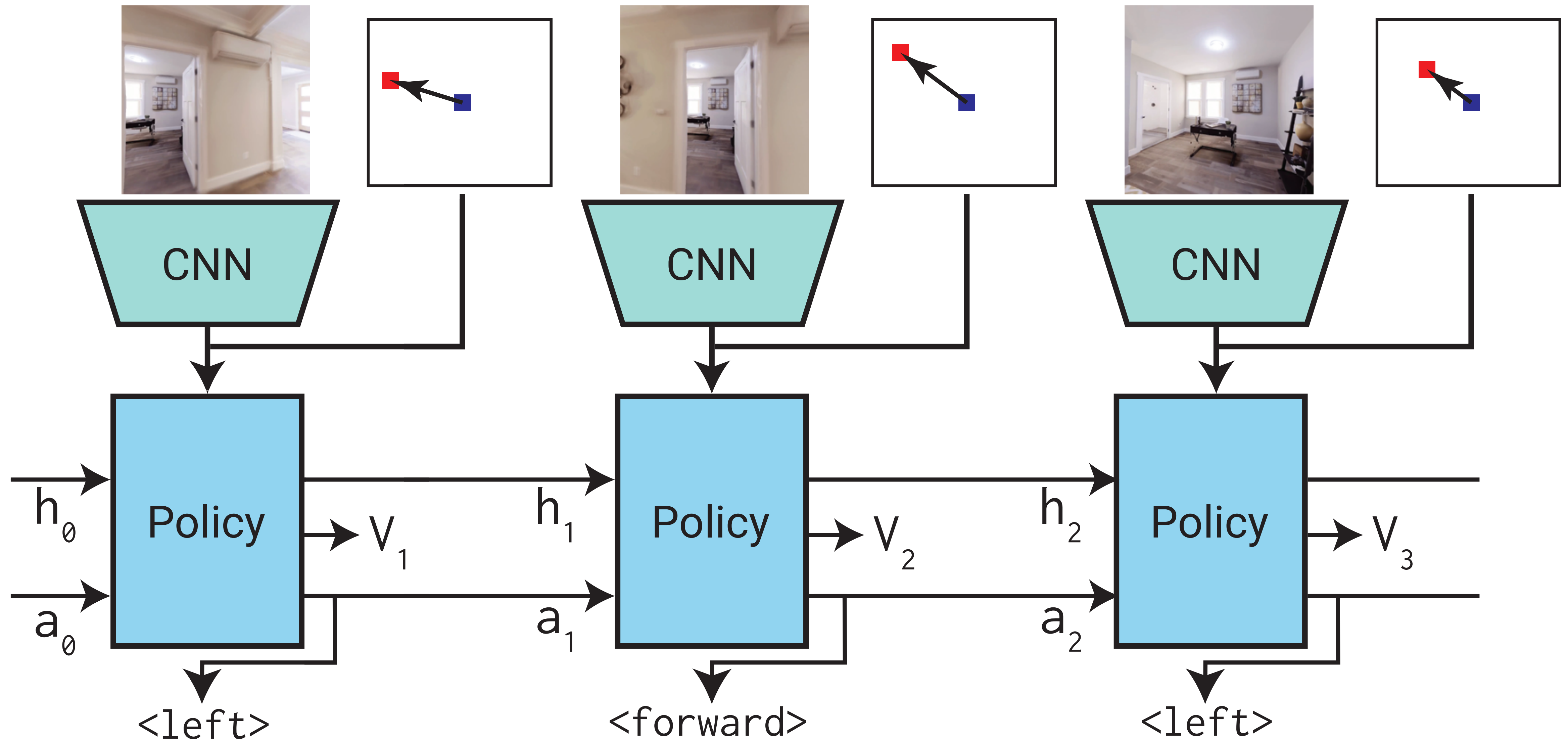
Agent and Model Design



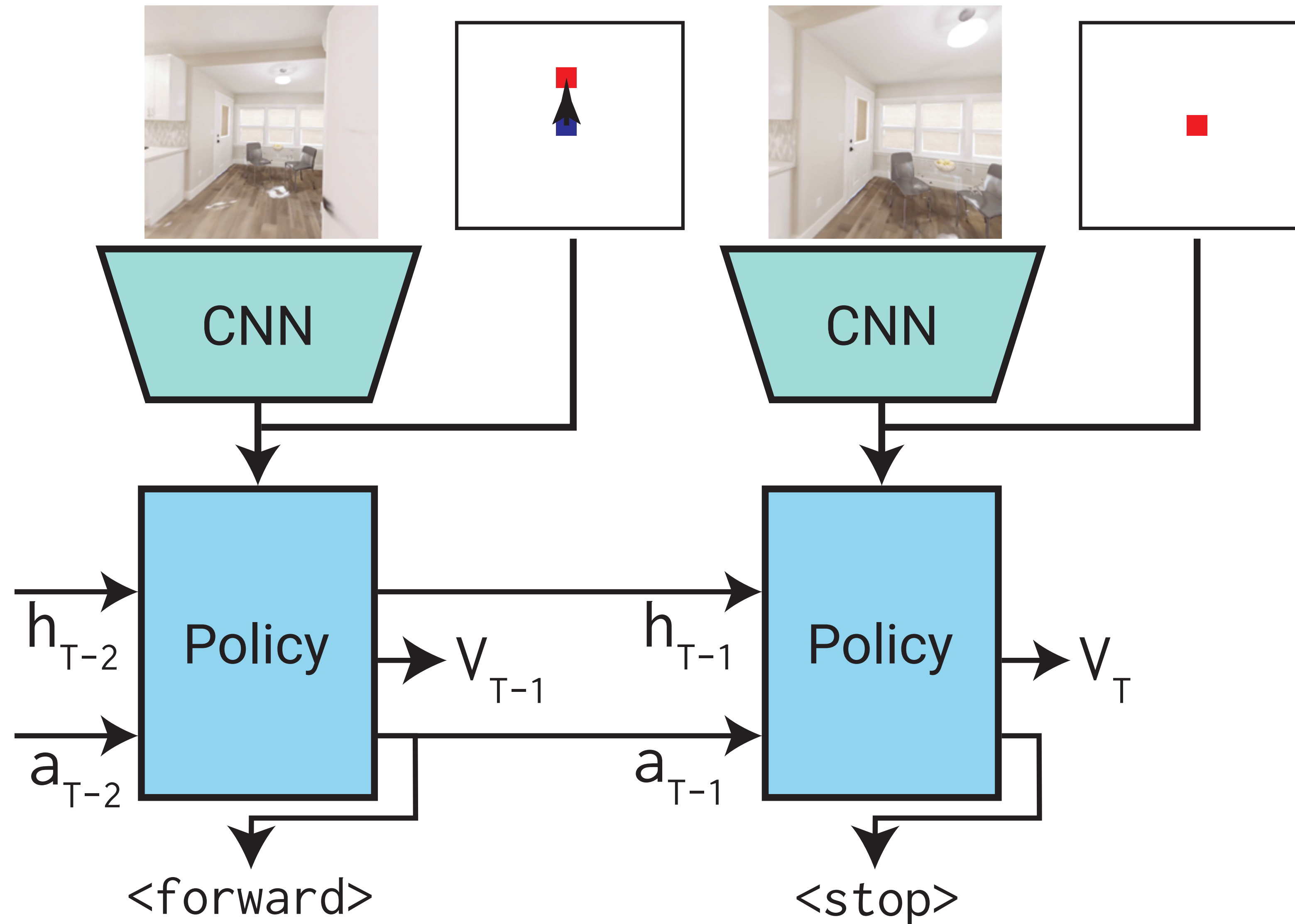
Agent and Model Design



Agent and Model Design

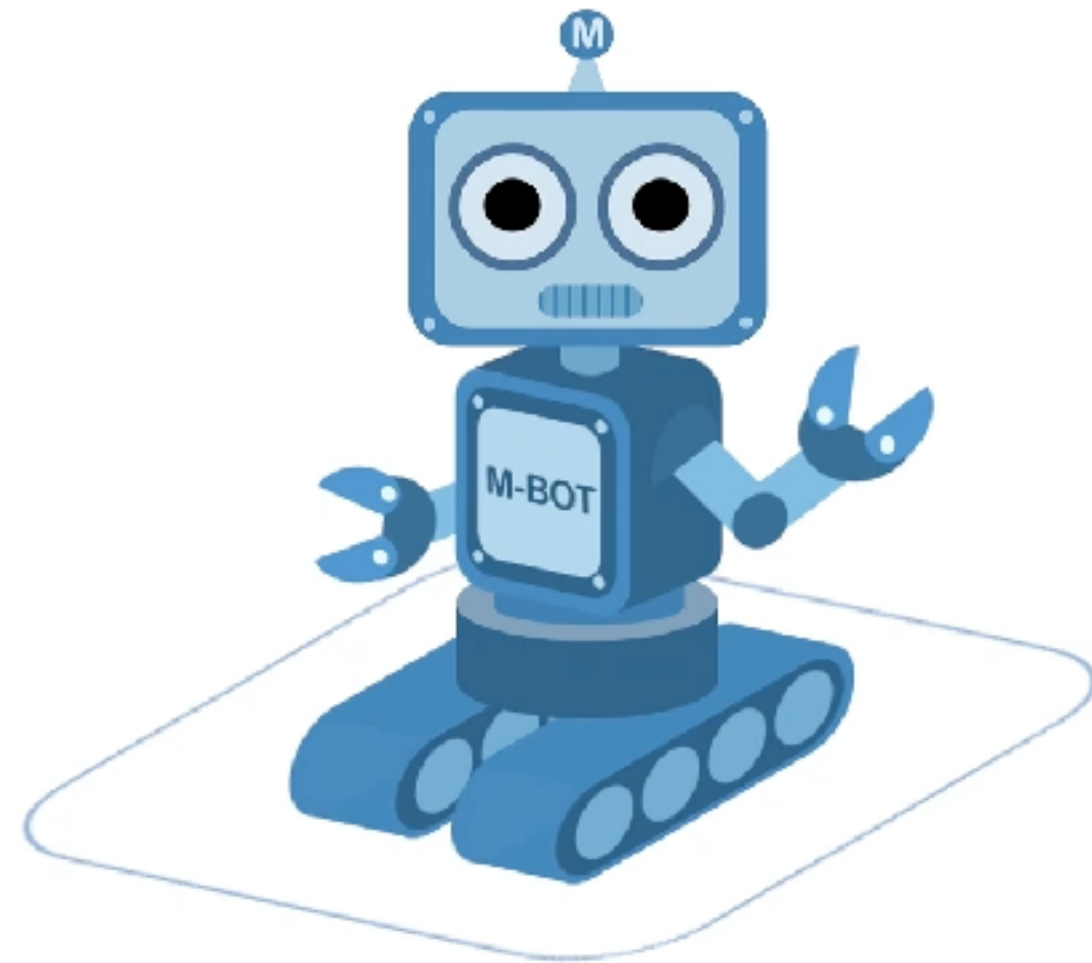


Agent and Model Design

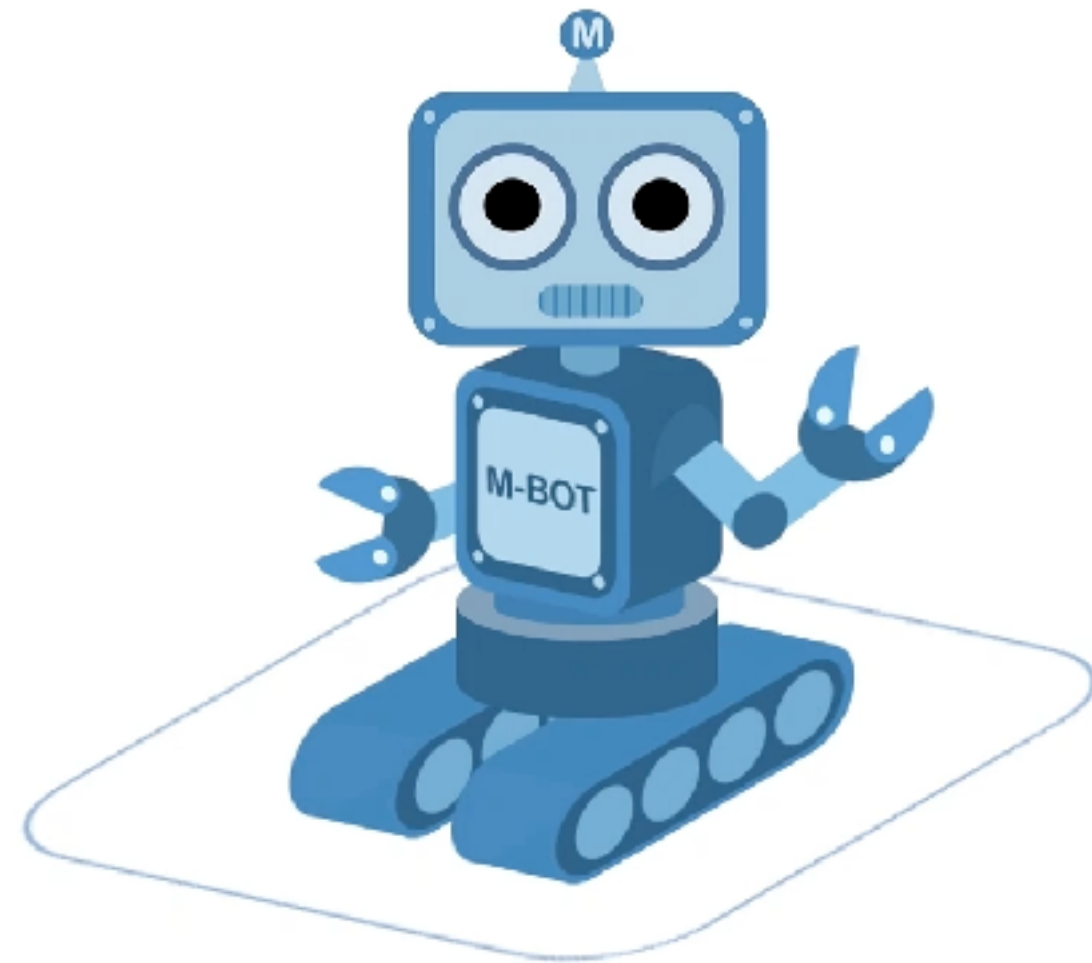


Agent and Model Design

- How do we train this agent?

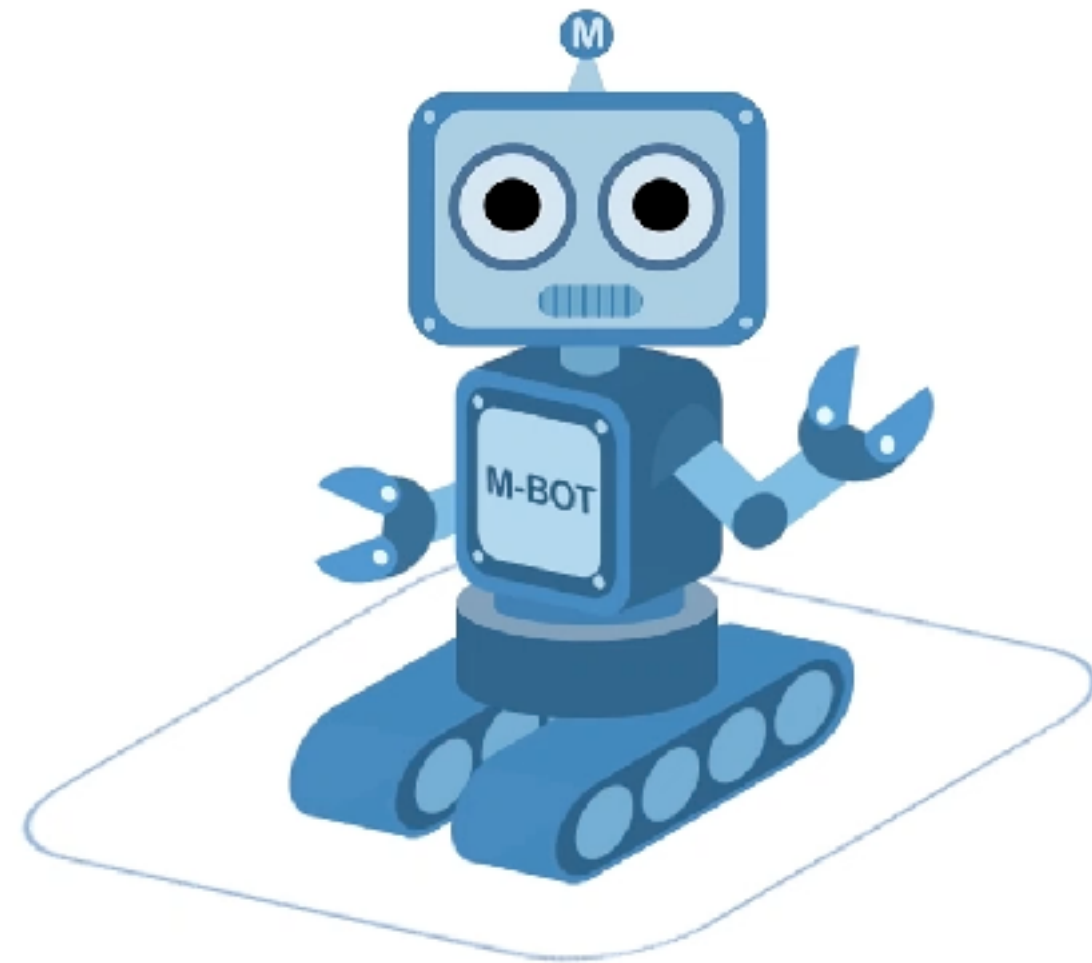


Agent and Model Design



- 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!

Outline

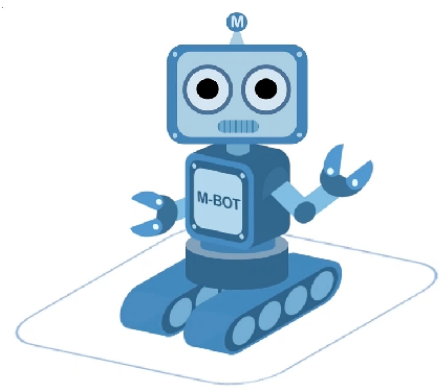
- RL Refresher/Advantage Actor Critic (A2C)
- Trust Region Policy Optimization (TRPO)
- Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation Results

Outline

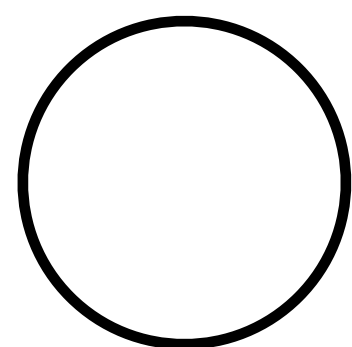
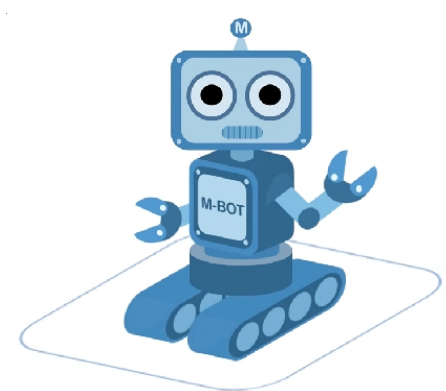
- RL Refresher/Advantage Actor Critic (A2C)
- Trust Region Policy Optimization (TRPO)
- Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation Results

RL Refresher

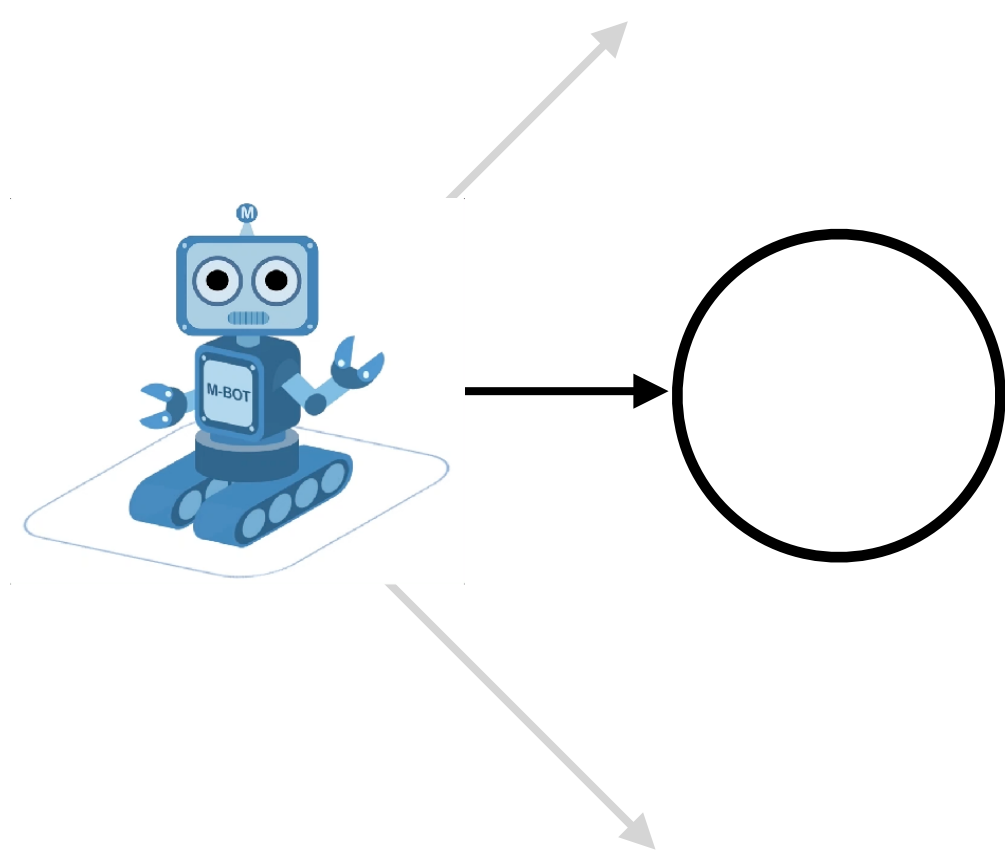
RL Refresher



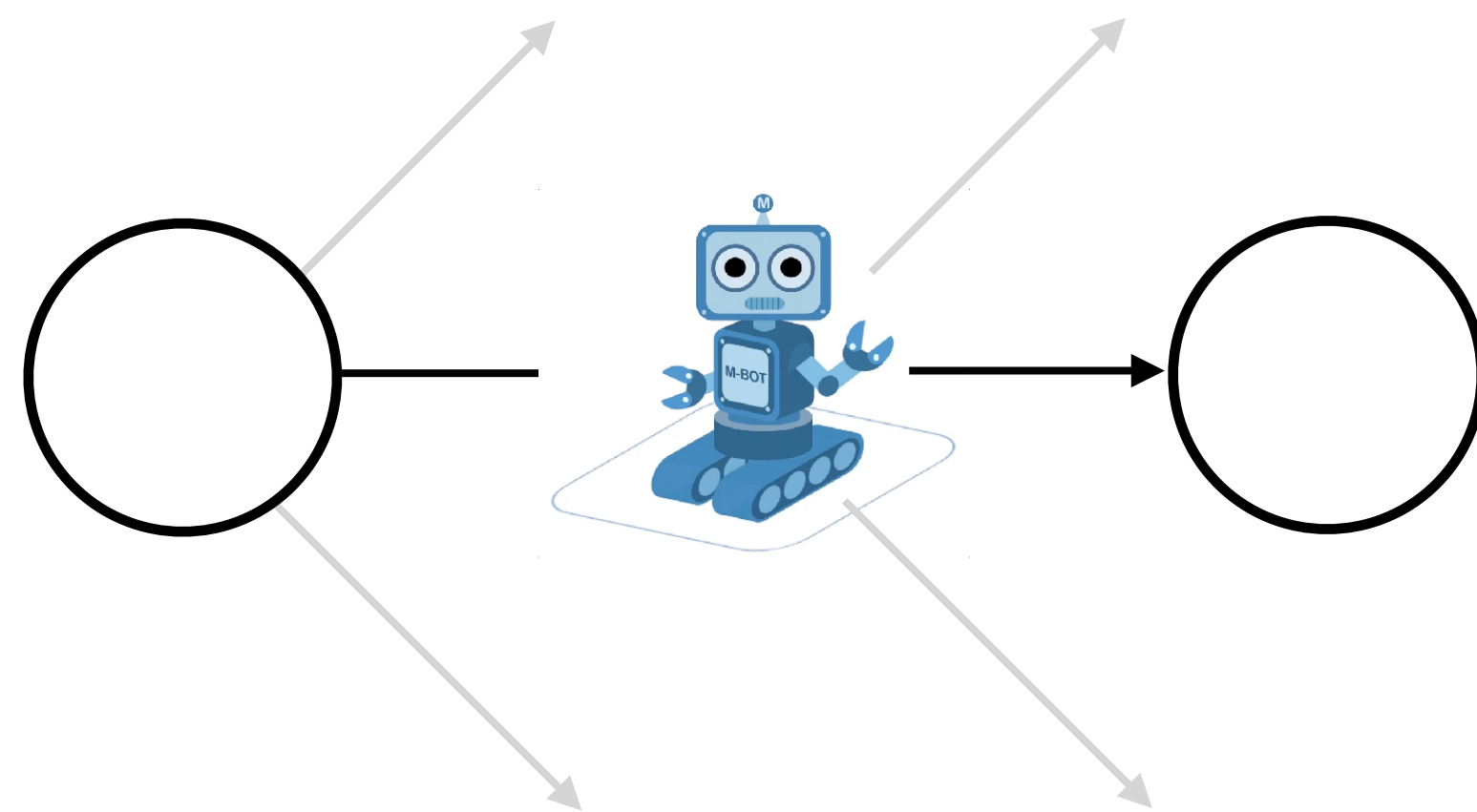
RL Refresher



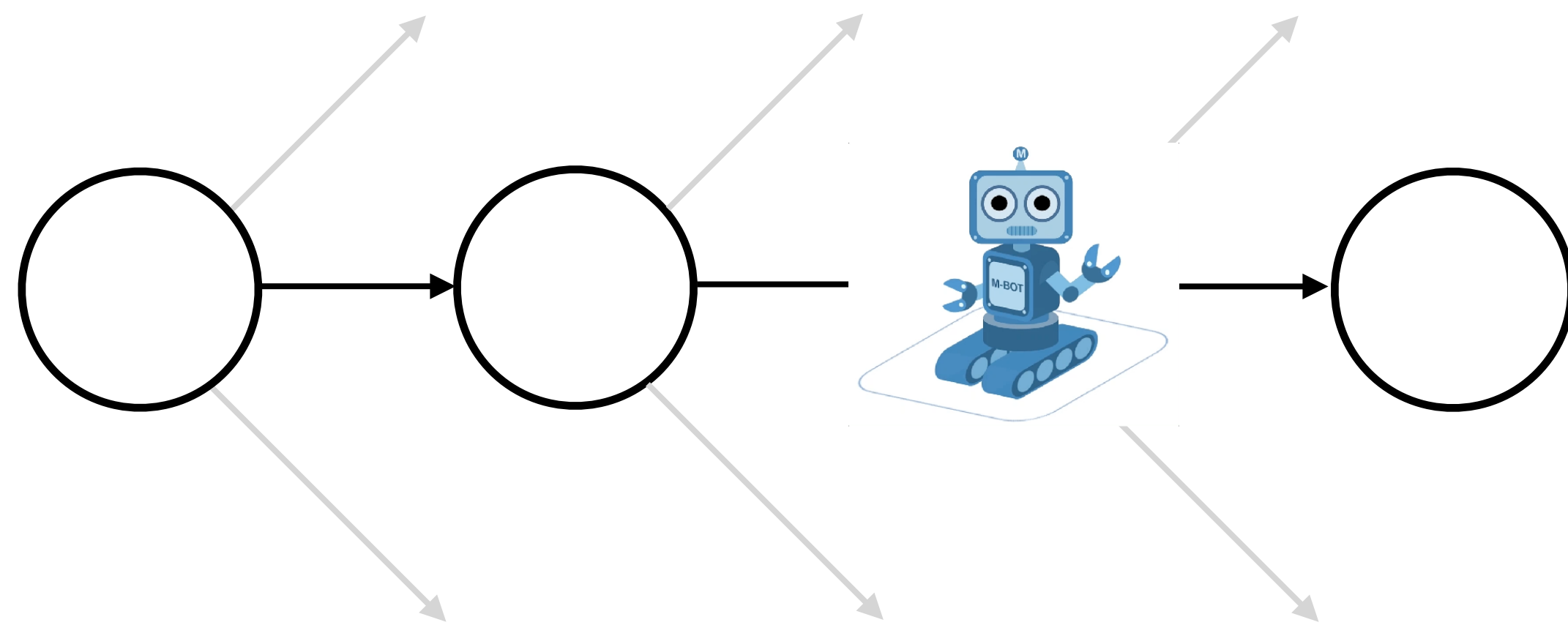
RL Refresher



RL Refresher



RL Refresher



RL Refresher

Objective:

$$\mathbb{E} [\mathcal{R}_T]$$

$$\mathcal{R}_T = \sum_{t=1}^T \mathcal{R}(s_t, a_t)$$

Reinforce

$$\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \mathcal{R}_T$$

Advantage Actor Critic (A2C)

- High variance: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \mathcal{R}_T$

Advantage Actor Critic (A2C)

- High variance: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \mathcal{R}_T$
- Reduce variance with baseline: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - b)$

Advantage Actor Critic (A2C)

- High variance: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \mathcal{R}_T$
- Reduce variance with baseline: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - b)$
- Use value-function as the baseline (A2C):

$$\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - V(s_t))$$

Advantage Actor Critic (A2C)

- High variance: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \mathcal{R}_T$
- Reduce variance with baseline: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - b)$
- Use value-function as the baseline (A2C):

$$\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - V(s_t))$$

Advantage Actor Critic (A2C)

- High variance: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \mathcal{R}_T$
- Reduce variance with baseline: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - b)$
- Use value-function as the baseline (A2C):

$$\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - V(s_t))$$

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

Advantage Actor Critic (A2C)

- High variance: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \mathcal{R}_T$
- Reduce variance with baseline: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - b)$
- Use value-function as the baseline (A2C):

$$\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - V(s_t))$$

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

Advantage Actor Critic (A2C)

- High variance: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot \mathcal{R}_T$
- Reduce variance with baseline: $\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - b)$
- Use value-function as the baseline (A2C):

$$\nabla_{\theta} \log \pi_{\theta}(a_t | s_t) \cdot (\mathcal{R}_T - V(s_t))$$

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

$$A(s_t, a_t) = (\mathcal{R}(s_t, a_t) + V(s_{t+1})) - V(s_t)$$

Advantage Actor Critic (A2C)

- A2C is great, but you can only use each rollout once!

Advantage Actor Critic (A2C)

- A2C is great, but you can only use each rollout once!

Why?

Advantage Actor Critic (A2C)

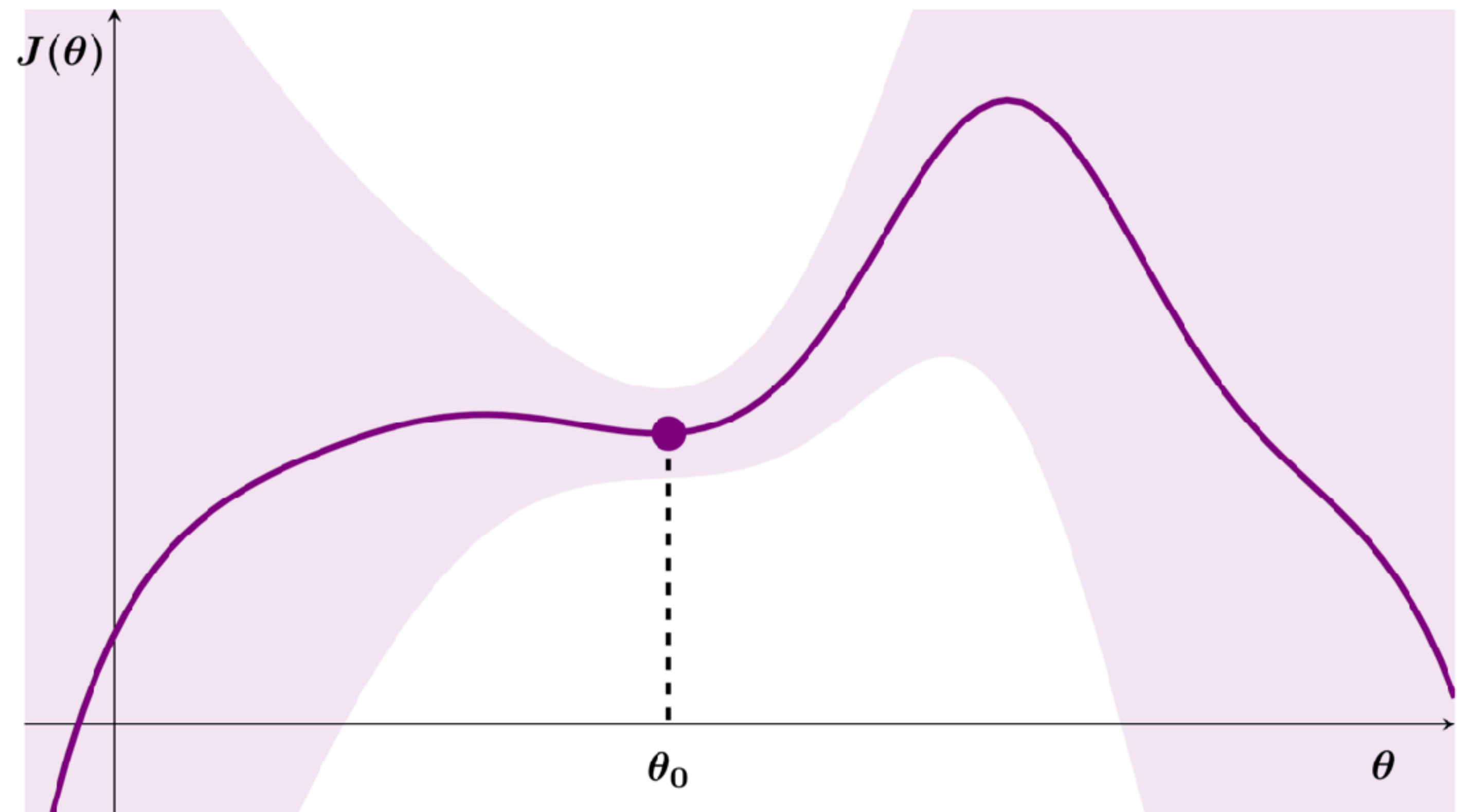
- A2C is great, but you can only use each rollout once!
 - No theoretical grounding to do so

Advantage Actor Critic (A2C)

- Works poorly in-practice

Advantage Actor Critic (A2C)

- Works poorly in-practice



Outline

- RL Refresher/Advantage Actor Critic (A2C)
- Trust Region Policy Optimization (TRPO)
- Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation Results

Trust Region Policy Optimization (TRPO)

A2C Maximizes: $\mathcal{J}^{\text{A2C}}(\theta) = \log \pi_{\theta}(a_t | s_t) A(s_t, a_t)$

Trust Region Policy Optimization (TRPO)

Given a policy: $q(a_t | s_t)$

Trust Region Policy Optimization (TRPO)

Given a policy: $q(a_t | s_t)$

Collect experience and calculate advantage

$$\tau \sim q(a_t | s_t)$$

$$A^q(s_t, a_t) = \mathcal{R}(s_t, a_t) + V^q(s_{t+1}) - V^q(s_t)$$

Trust Region Policy Optimization (TRPO)

Given a policy: $q(a_t | s_t)$

Collect experience and calculate advantage

$$\tau \sim q(a_t | s_t)$$

$$A^q(s_t, a_t) = \mathcal{R}(s_t, a_t) + V^q(s_{t+1}) - V^q(s_t)$$

Maximize:
$$\mathcal{J}(\theta) = \frac{\pi_\theta(a_t | s_t)}{q(a_t | s_t)} \cdot A^q(s_t, a_t)$$

Trust Region Policy Optimization (TRPO)

Maximize:
$$\mathcal{J}(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{q(a_t | s_t)} \cdot A^q(s_t, a_t)$$

Read as: Policy $\pi_{\theta}(a_t | s_t)$ is better than $q(a_t | s_t)$ if it takes good actions ($A^q(s_t, a_t) > 0$) more often and takes bad actions ($A^q(s_t, a_t) < 0$) less often

Trust Region Policy Optimization (TRPO)

Maximize:
$$\mathcal{J}(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{q(a_t | s_t)} \cdot A^q(s_t, a_t)$$

Read as: Policy π_{θ} ($A^q(s_t, a_t) > 0$) more often
Why this objective? good actions ($A^q(s_t, a_t) > 0$) more often
bad actions ($A^q(s_t, a_t) < 0$) less often

Trust Region Policy Optimization (TRPO)

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

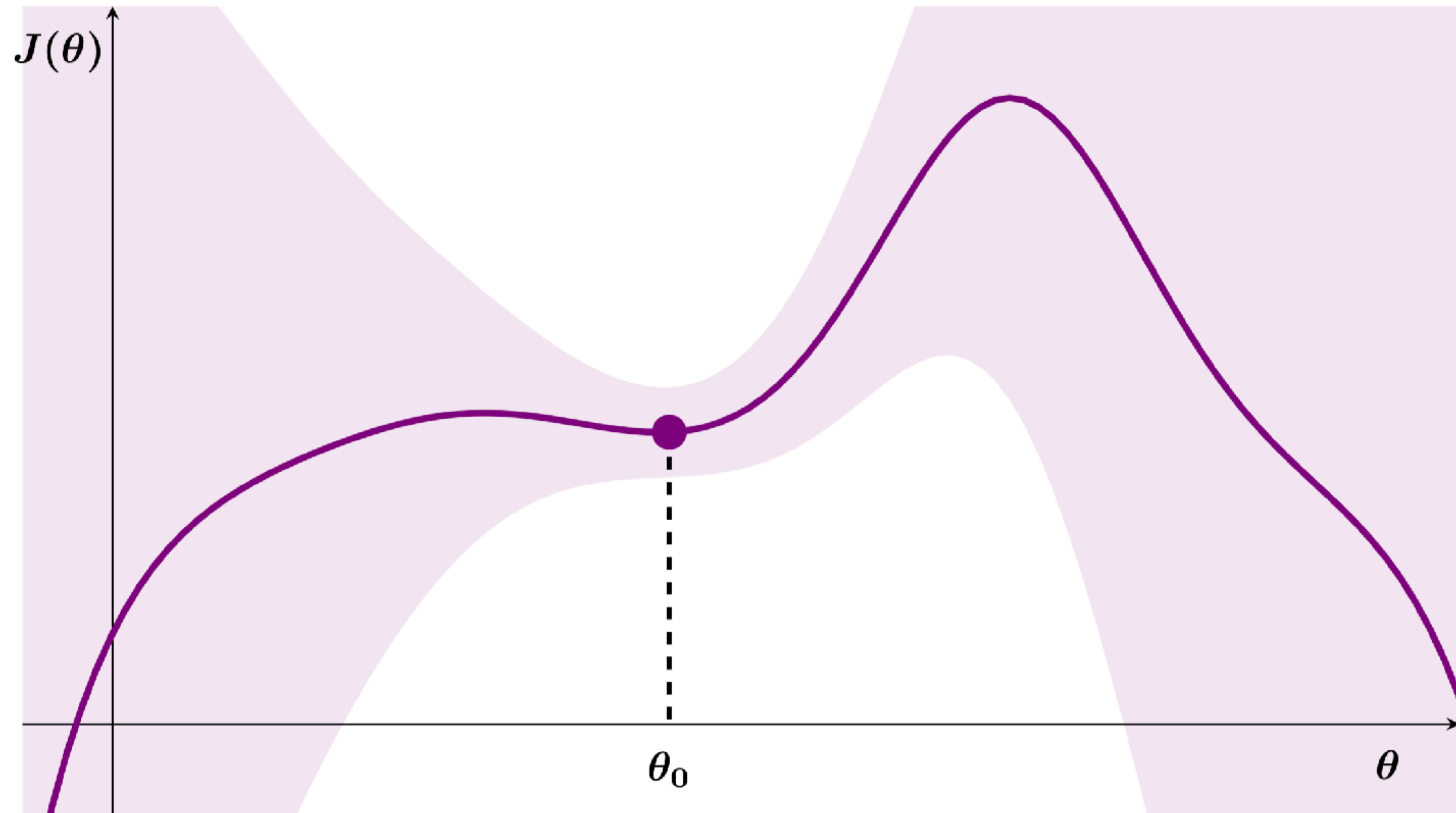
Collect experience and calculate advantage

$$\tau \sim q(a_t | s_t)$$

$$A^q(s_t, a_t) = \mathcal{R}(s_t, a_t) + V^q(s_{t+1}) - V^q(s_t)$$

Maximize:
$$\mathcal{J}(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{q(a_t | s_t)} \cdot A^q(s_t, a_t)$$

Trust Region Policy Optimization (TRPO)



Trust Region Policy Optimization (TRPO)

- Use a *trust-region*!

Trust Region Policy Optimization (TRPO)

- PS 1 problem 1

Trust Region Policy Optimization (TRPO)

- PS 1 problem 1
- In this problem, you showed that the gradient descent update rule

$$\mathbf{w}^{(t+1)} = \mathbf{w}^{(t)} - \eta \nabla f \mathbf{w}^{(t)}$$

can be seen as the minimizer of the affine-lower bound of $f(\mathbf{w})$ subject to a *trust-region*:

$$\underbrace{f(\mathbf{w}^{(t)}) + \langle \mathbf{w} - \mathbf{w}^{(t)}, \nabla f(\mathbf{w}^{(t)}) \rangle}_{\text{affine lower bound to } f(\cdot)} + \underbrace{\frac{\lambda}{2} \cdot \|\mathbf{w} - \mathbf{w}^{(t)}\|^2}_{\text{proximity term}}$$

Trust Region Policy Optimization (TRPO)

$$\mathcal{J}^{TRPO}(\theta) = \underbrace{r_t(\theta) A^q(s_t, a_t)}_{\text{importance-weighted advantage}} - \underbrace{\beta \cdot KL\left(\pi_\theta(a_t | s_t) \parallel q(a_t | s_t)\right)}_{\text{proximity term}}$$
$$r_t(\theta) = \frac{\pi_\theta(a_t | s_t)}{q(a_t | s_t)}$$

Trust Region Policy Optimization (TRPO)

- Advantage
 - Able to perform multiple optimization steps per rollout

Trust Region Policy Optimization (TRPO)

- Advantage
 - Able to perform multiple optimization steps per rollout
- Disadvantage
 - Choosing the correct value for beta is challenging and problem/network dependent

Outline

- RL Refresher/Advantage Actor Critic (A2C)
- Trust Region Policy Optimization (TRPO)
- Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation Results

Proximal Policy Optimization (PPO)

Proximal Policy Optimization (PPO)



OpenAI Five

**AlphaStar: Mastering the
Real-Time Strategy Game
StarCraft II**

Proximal Policy Optimization (PPO)

Given a policy: $q(a_t | s_t)$

Proximal Policy Optimization (PPO)

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

Proximal Policy Optimization (PPO)

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

Objective: Maximize

$$r_t(\theta) = \frac{\pi_{\theta}(a_t | s_t)}{q(a_t | s_t)}$$

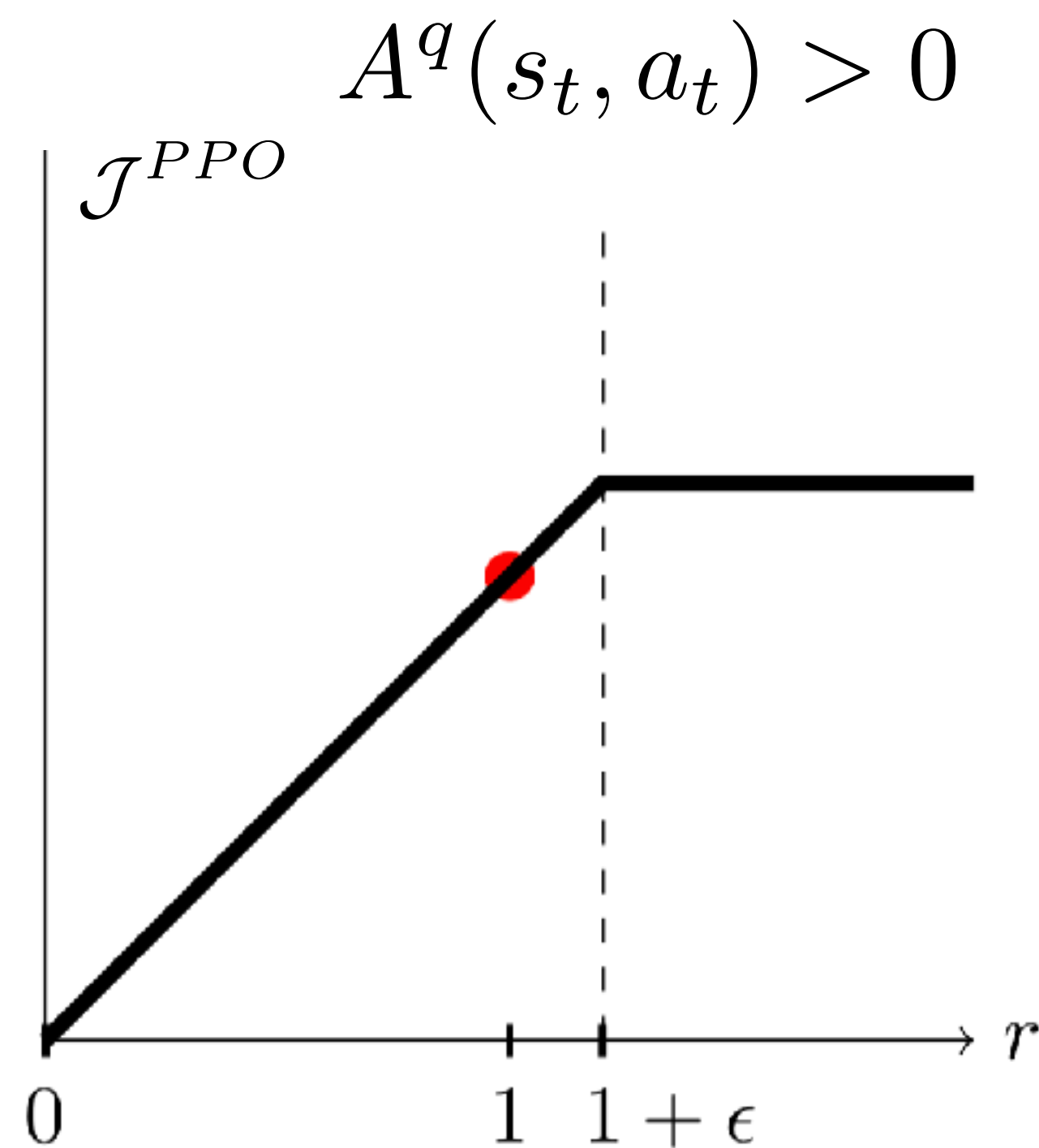
$$\mathcal{J}^{\text{PPO}}(\theta) = A^q(s_t, a_t) \cdot \begin{cases} \min(r_t(\theta), 1 + \epsilon) & \text{if } A^q(s_t, a_t) > 0 \\ \max(r_t(\theta), 1 - \epsilon) & \text{if } A^q(s_t, a_t) < 0 \end{cases}$$

Proximal Policy Optimization (PPO)

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

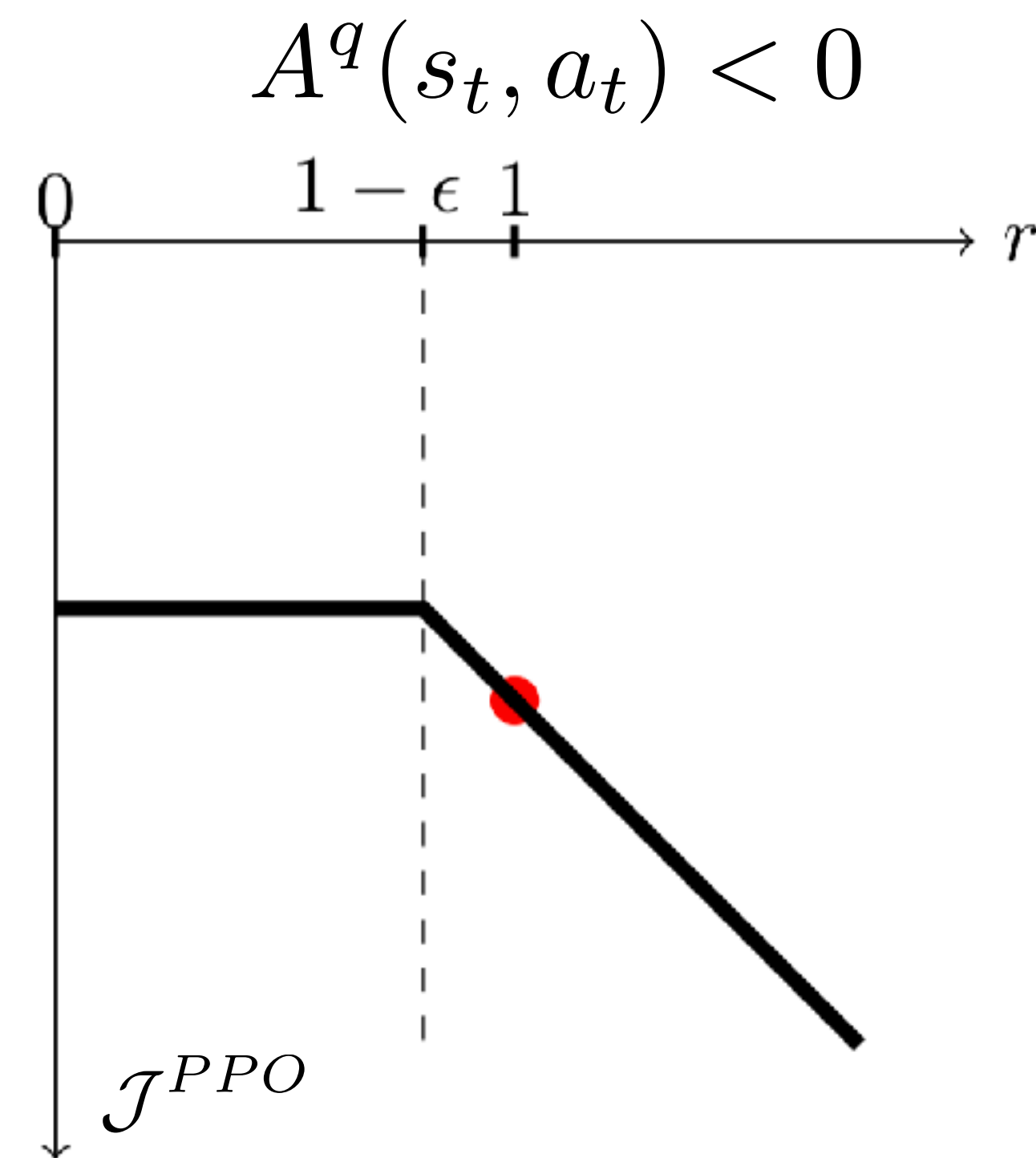
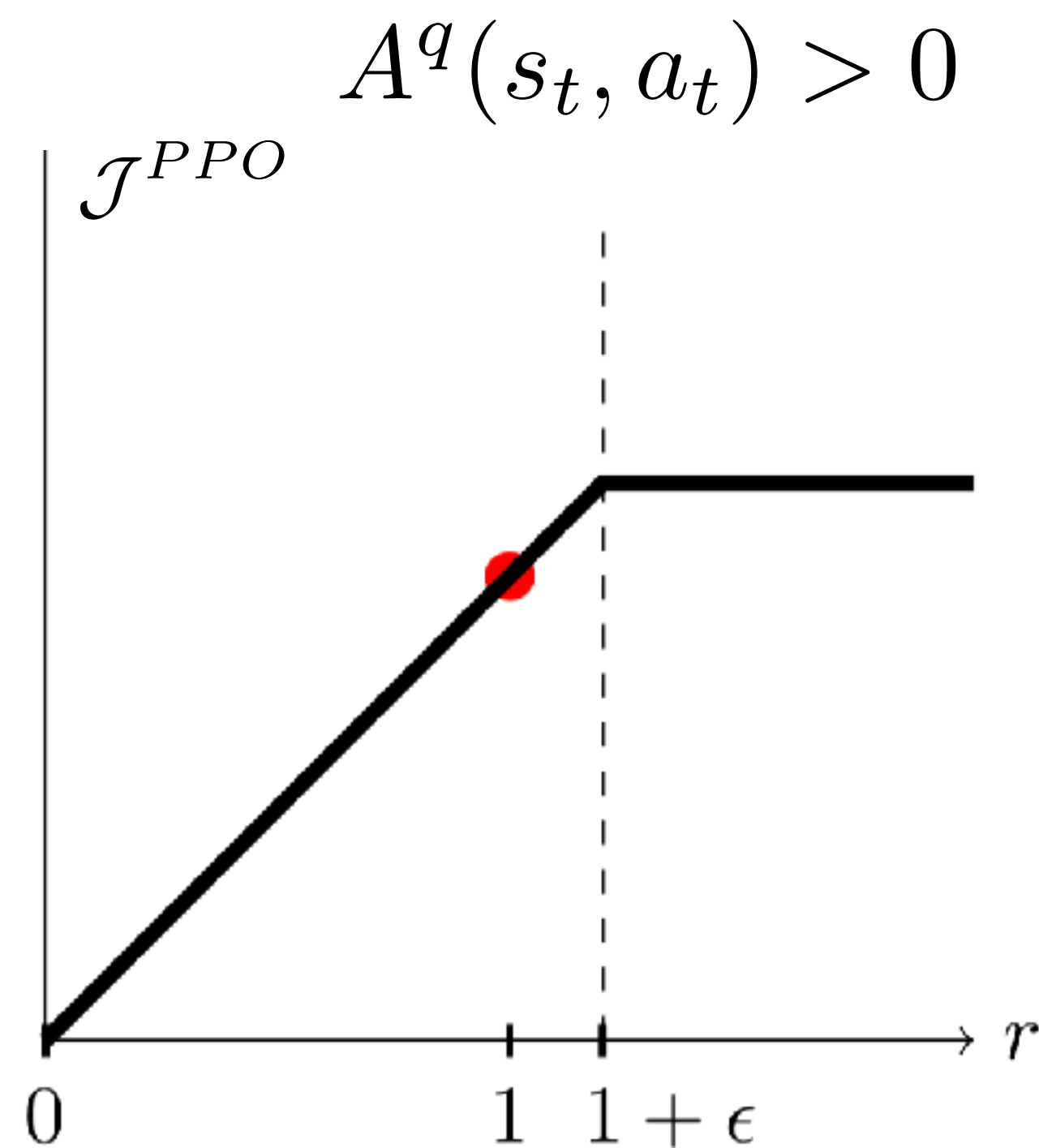
Proximal Policy Optimization (PPO)

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



Proximal Policy Optimization (PPO)

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



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

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

A2C Implementation

1. Collect a set of trajectories using current policy
2. Update policy via step of A2C objective
3. Repeat

PPO/TRPO Implementation

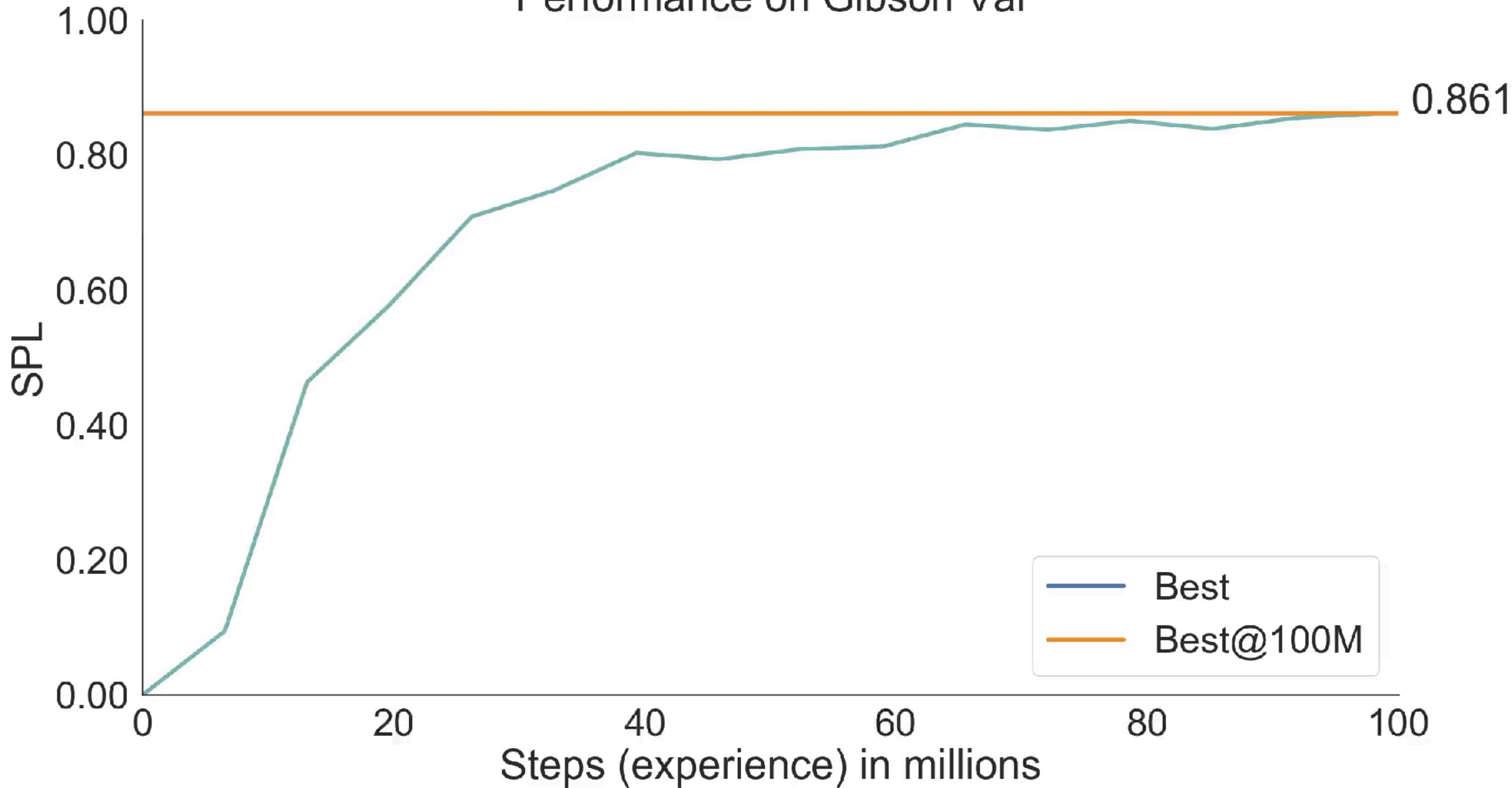
1. Collect a set of trajectories using current policy
2. For a few epochs (typically 2 or 4)
 1. Sample mini batches from rollout (typically 2 or 4)
 1. Update the policy via step of PPO/TRPO objective
3. Repeat

Outline

- RL Refresher/Advantage Actor Critic (A2C)
- Trust Region Policy Optimization (TRPO)
- Proximal Policy Optimization (PPO)
- Application: PointGoal Navigation Results

PointGoal Navigation Results

Performance on Gibson Val



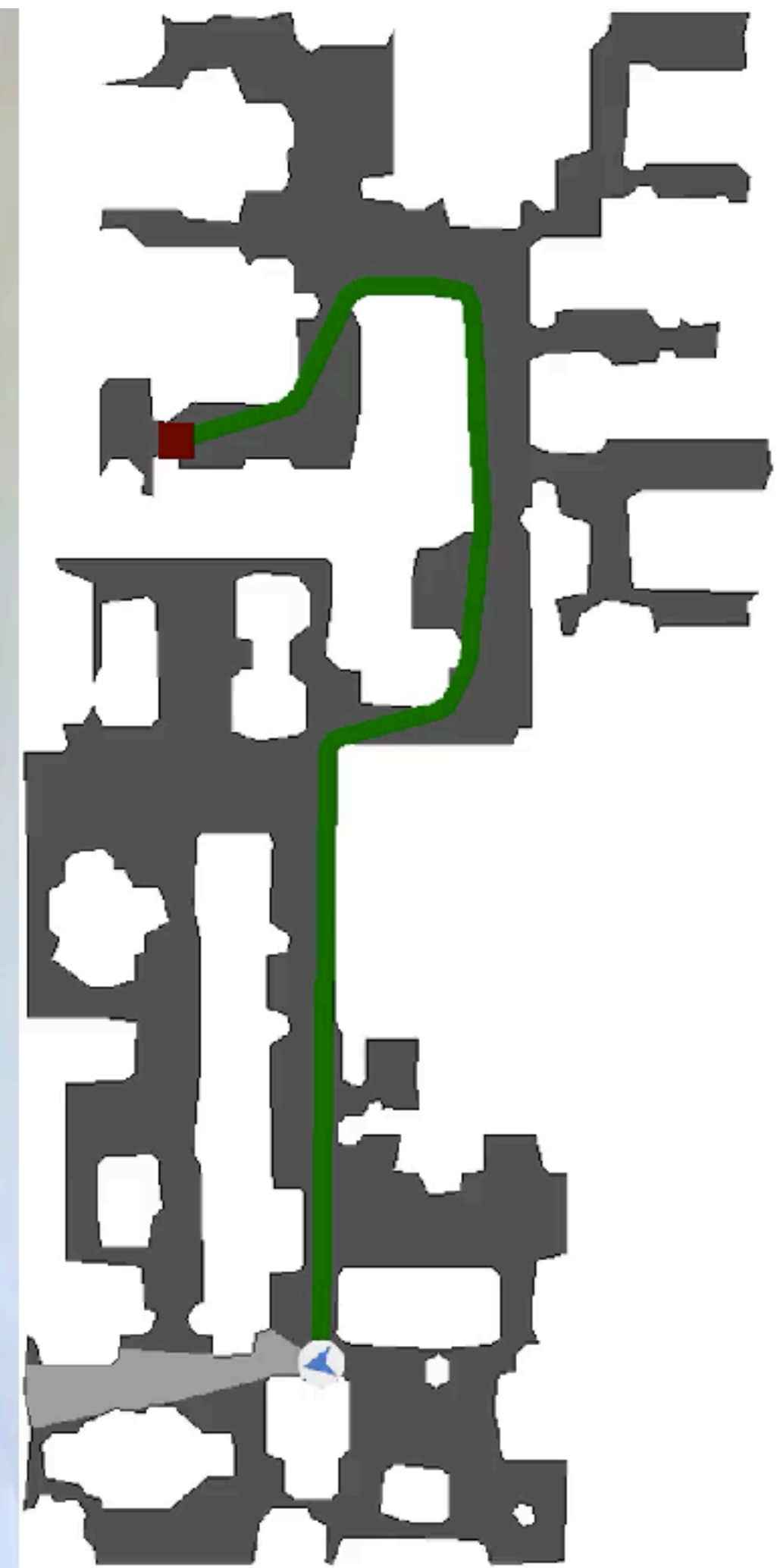
Qualitative Results



Depth



RGB and GPS+Compass



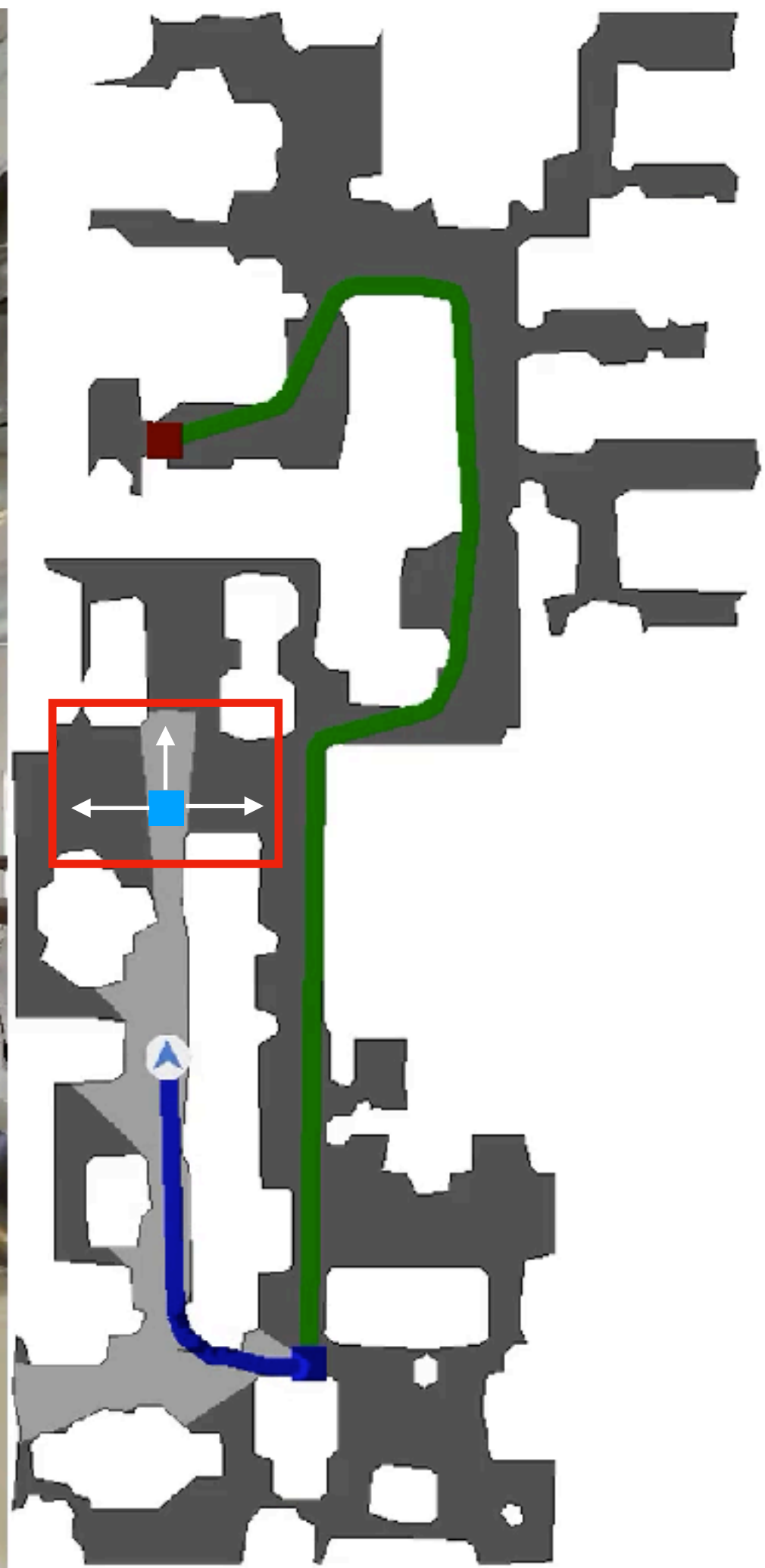
Top Down Map



Depth



RGB and GPS+Compass



Top Down Map



Depth



RGB and GPS+Compass



Top Down Map



Depth



RGB and GPS+Compass



Top Down Map



Depth



RGB and GPS+Compass



Top Down Map



Depth



RGB and GPS+Compass



Top Down Map



Depth



RGB and GPS+Compass

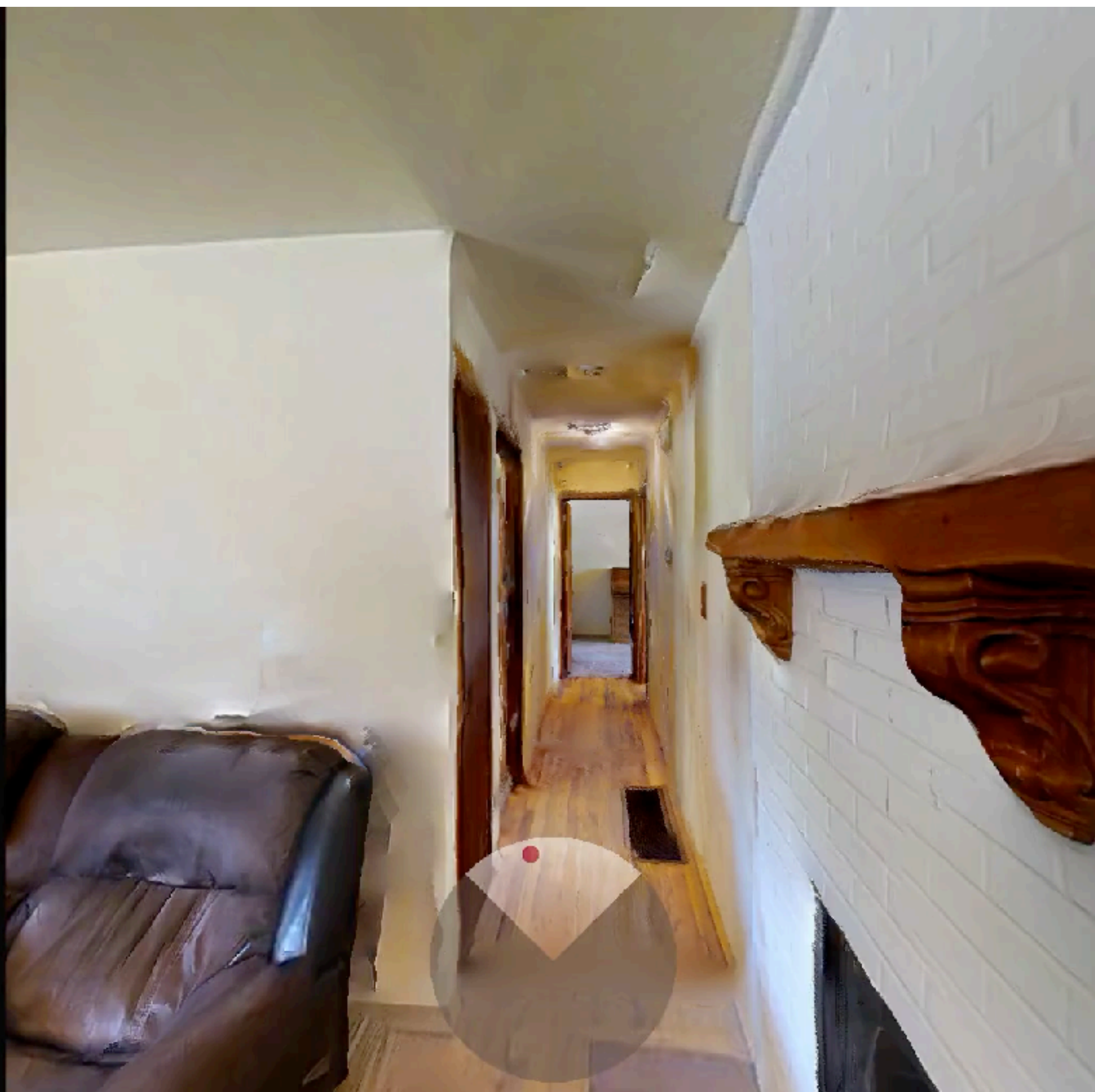


Top Down Map

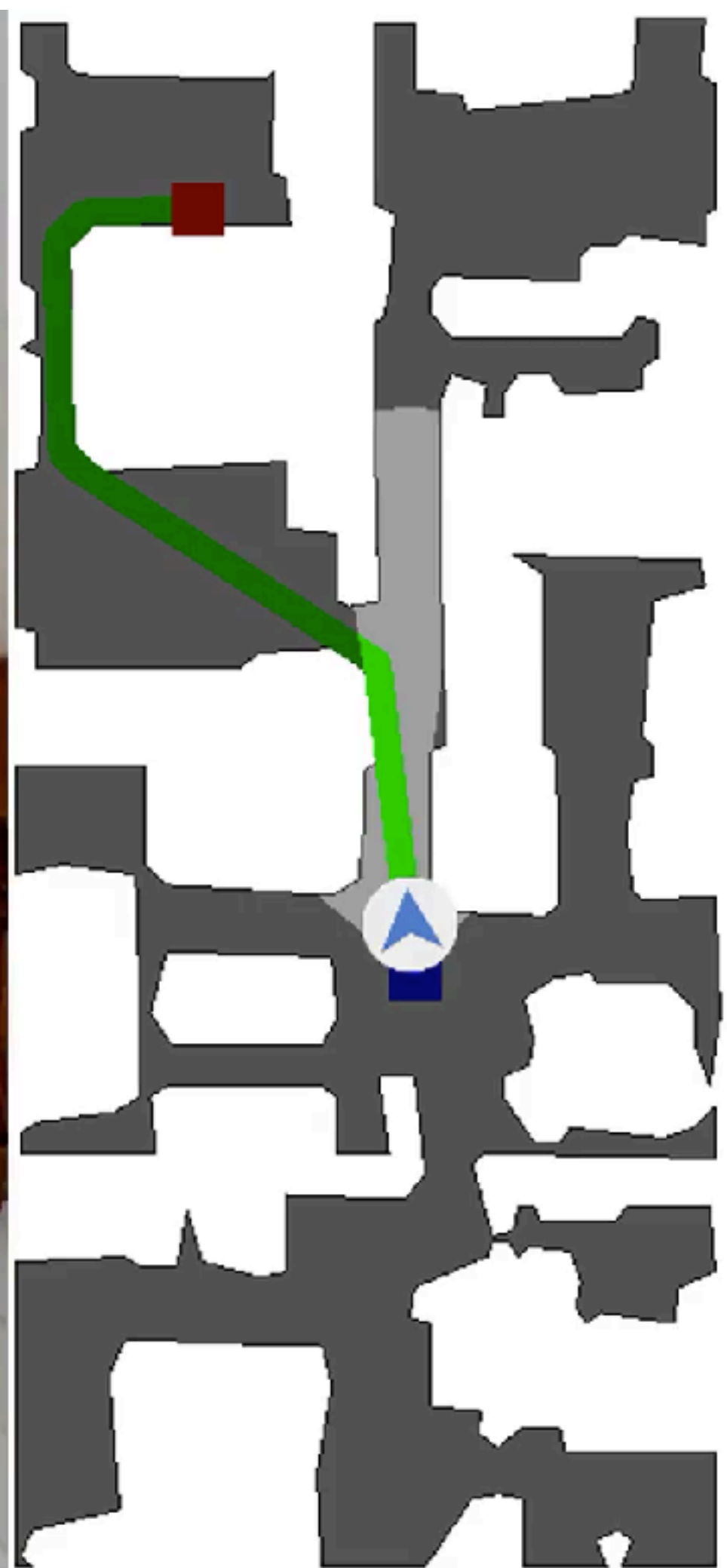
Backtracking



Depth



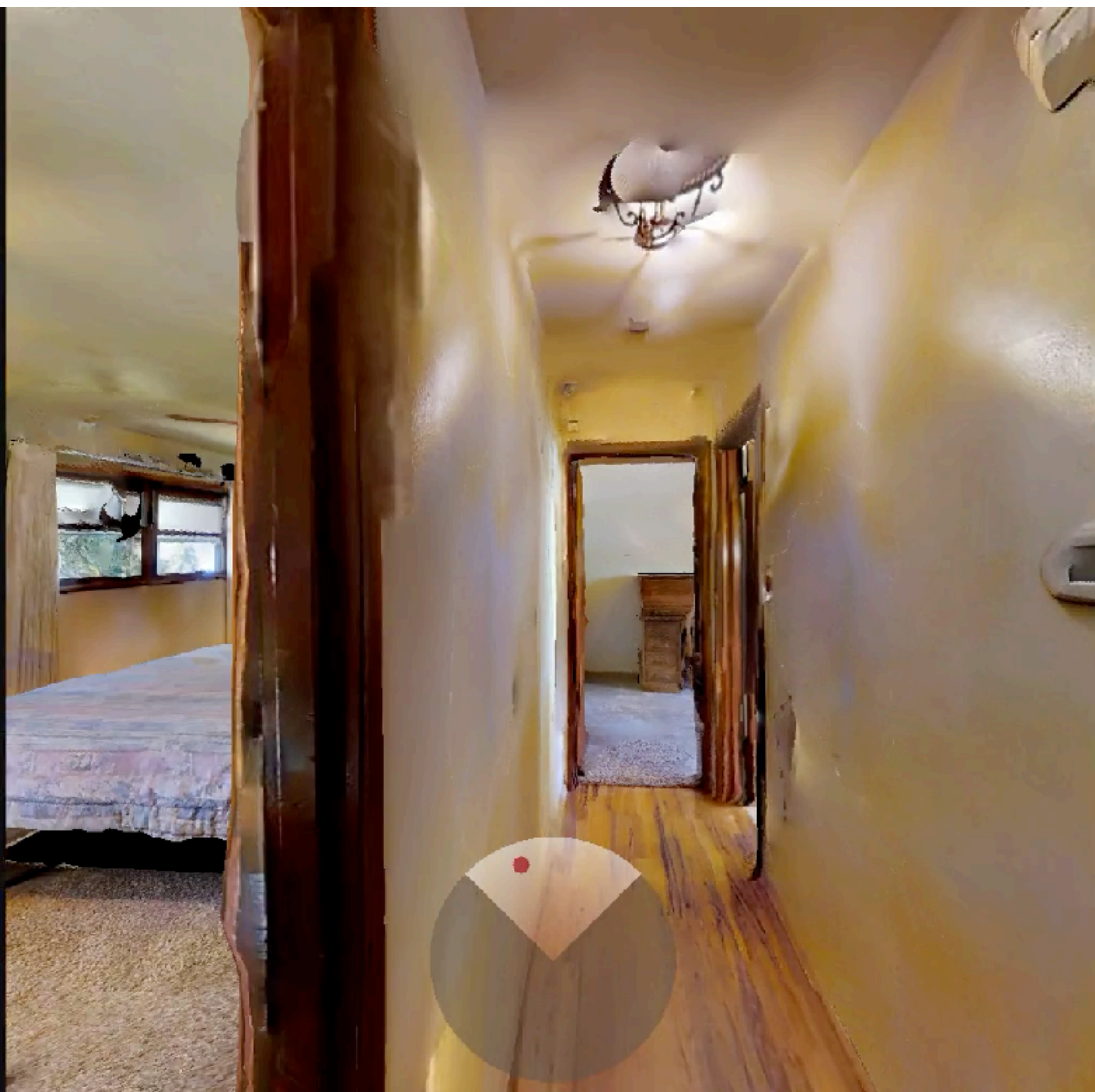
RGB and GPS+Compass



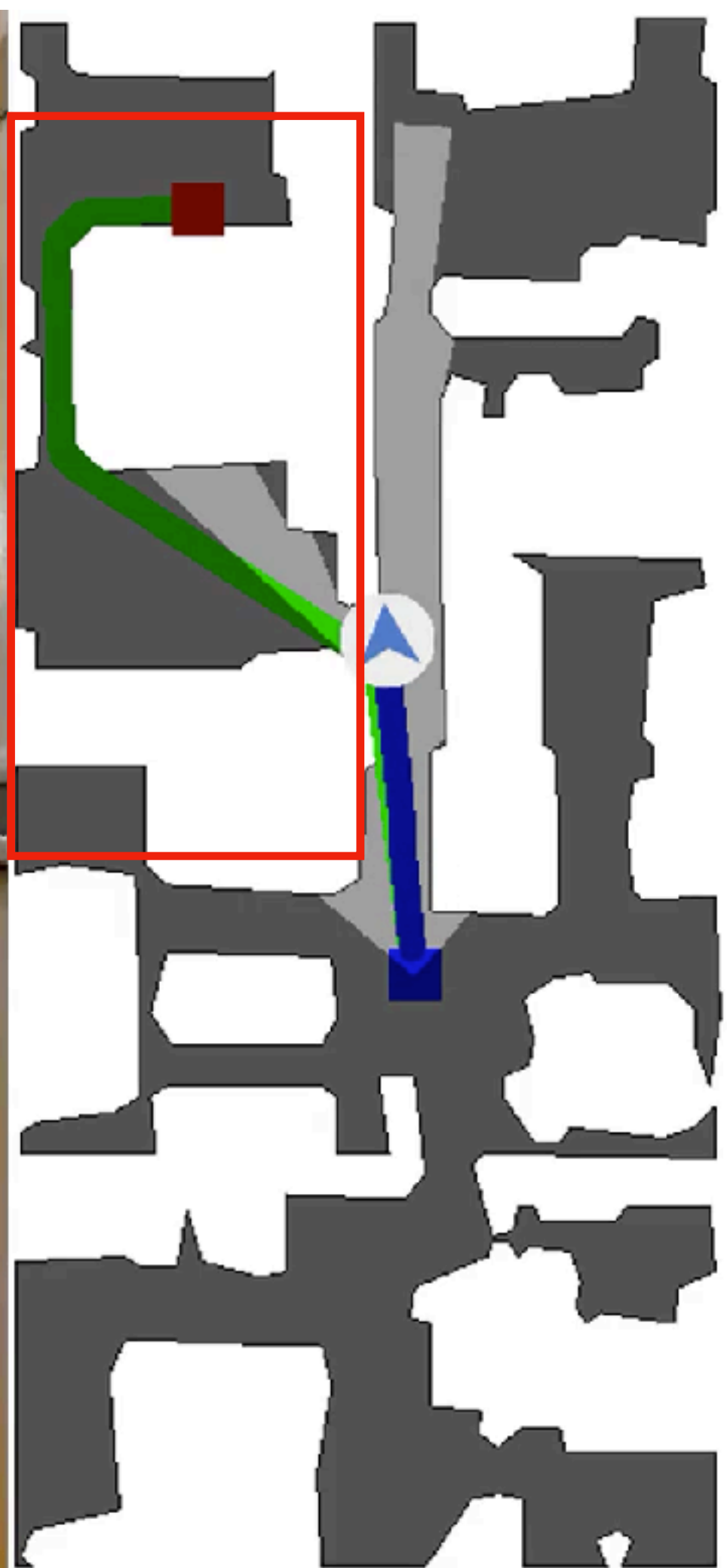
Top Down Map



Depth



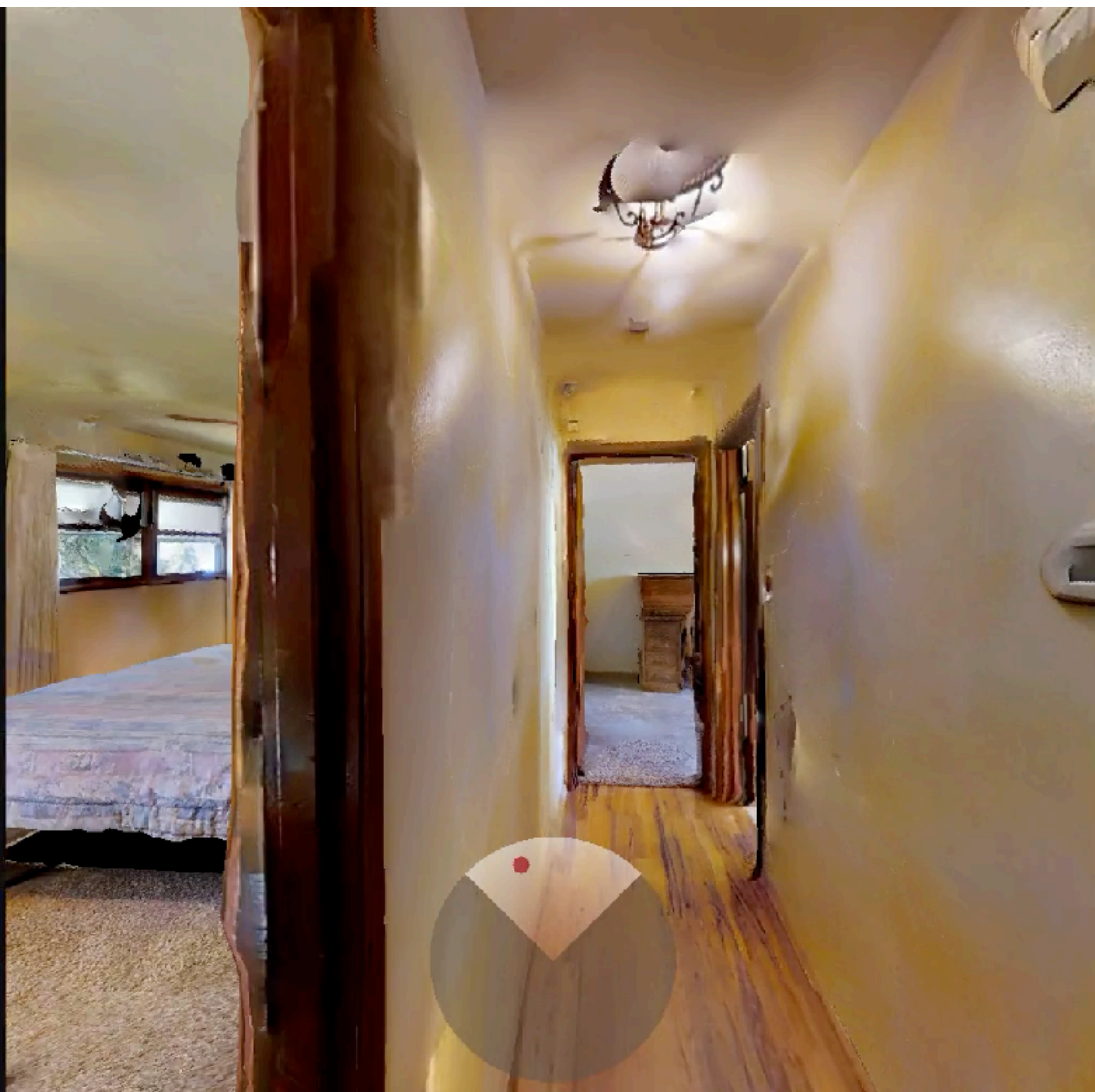
RGB and GPS+Compass



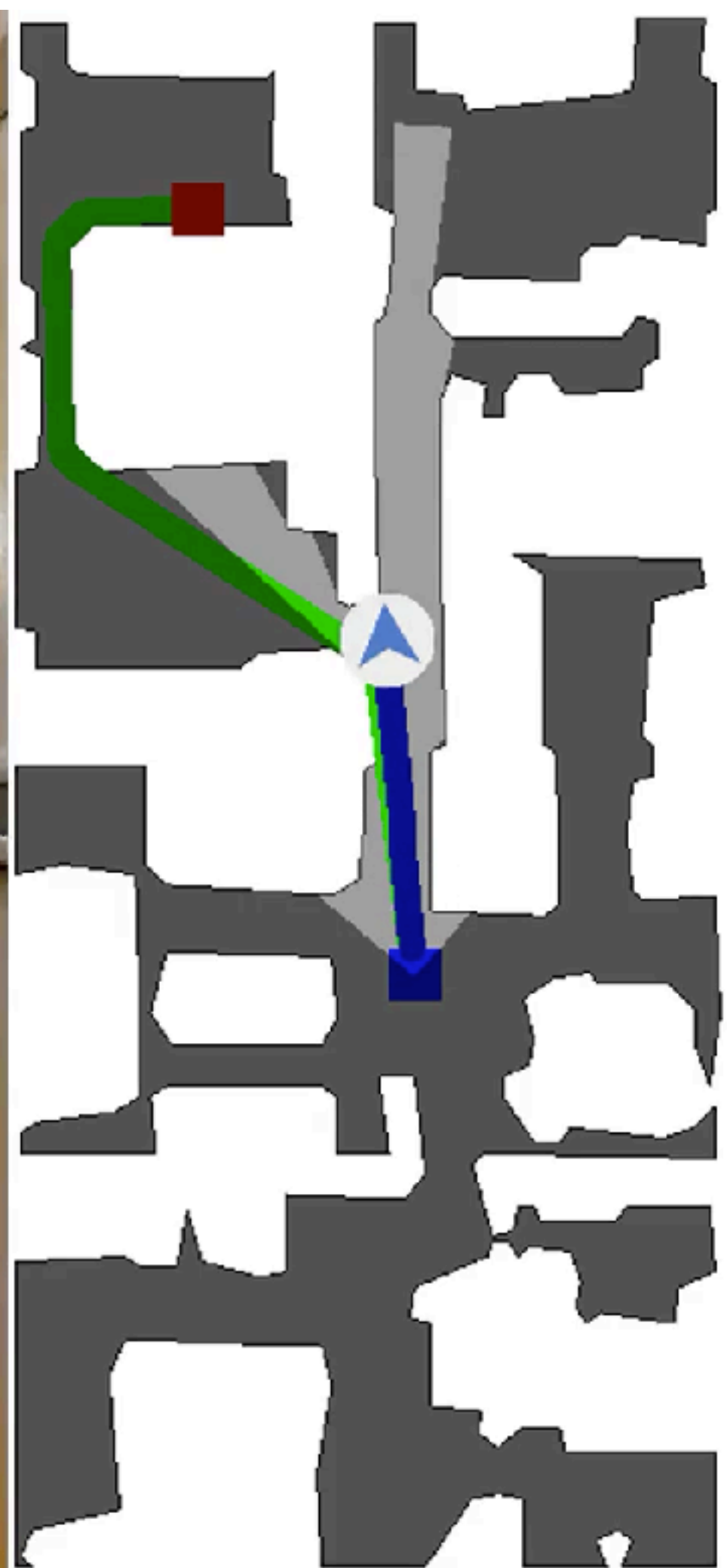
Top Down Map



Depth



RGB and GPS+Compass



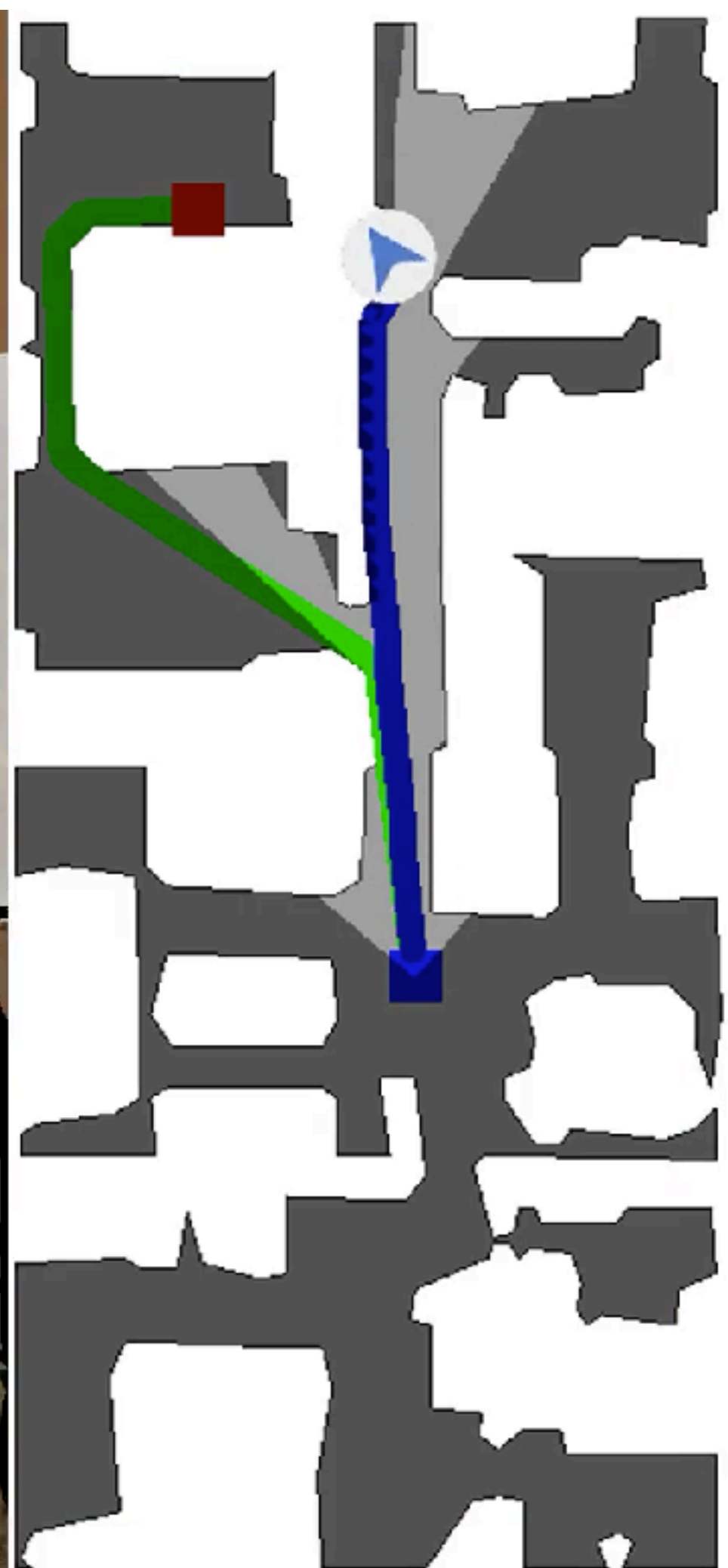
Top Down Map



Depth



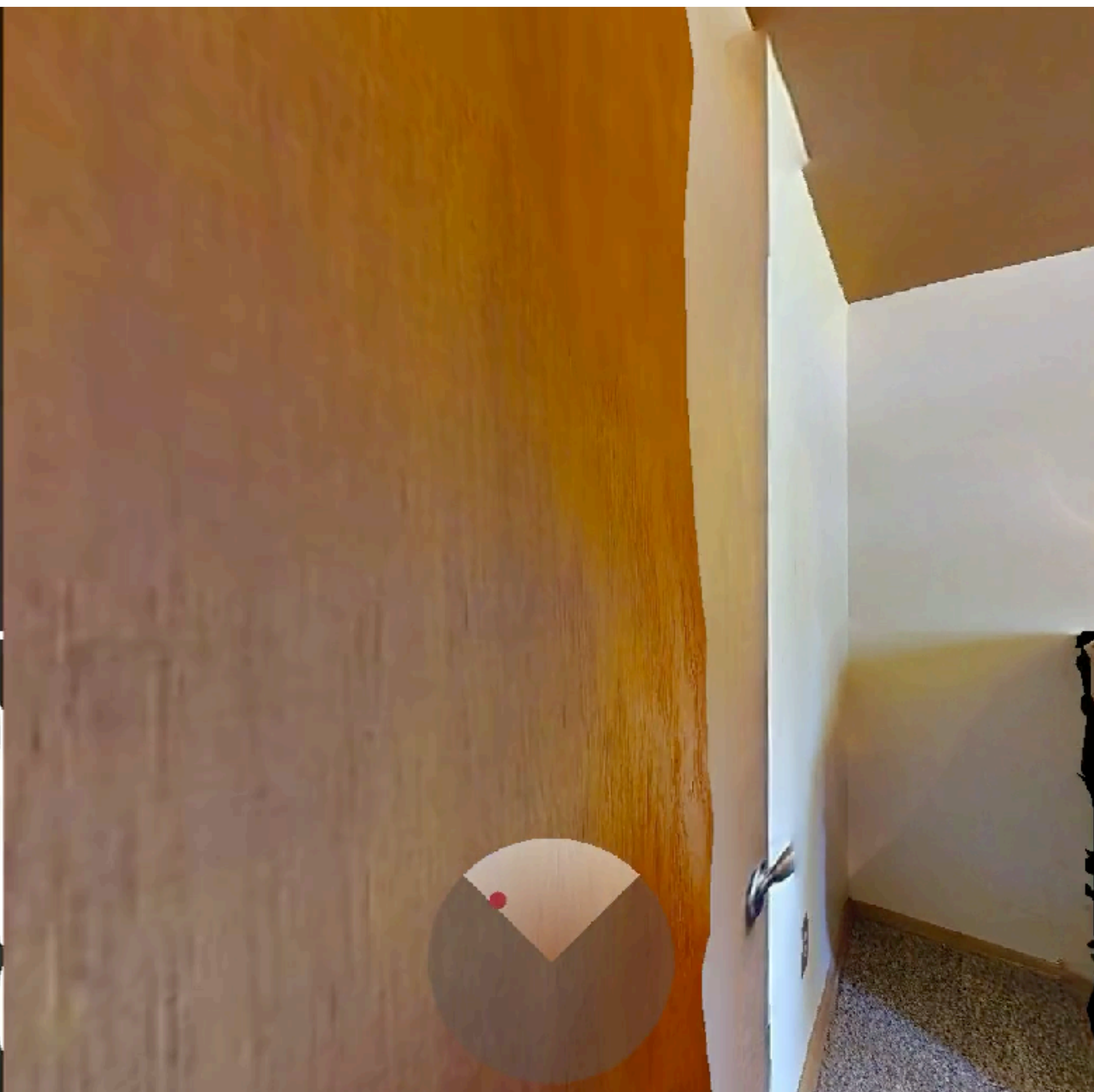
RGB and GPS+Compass



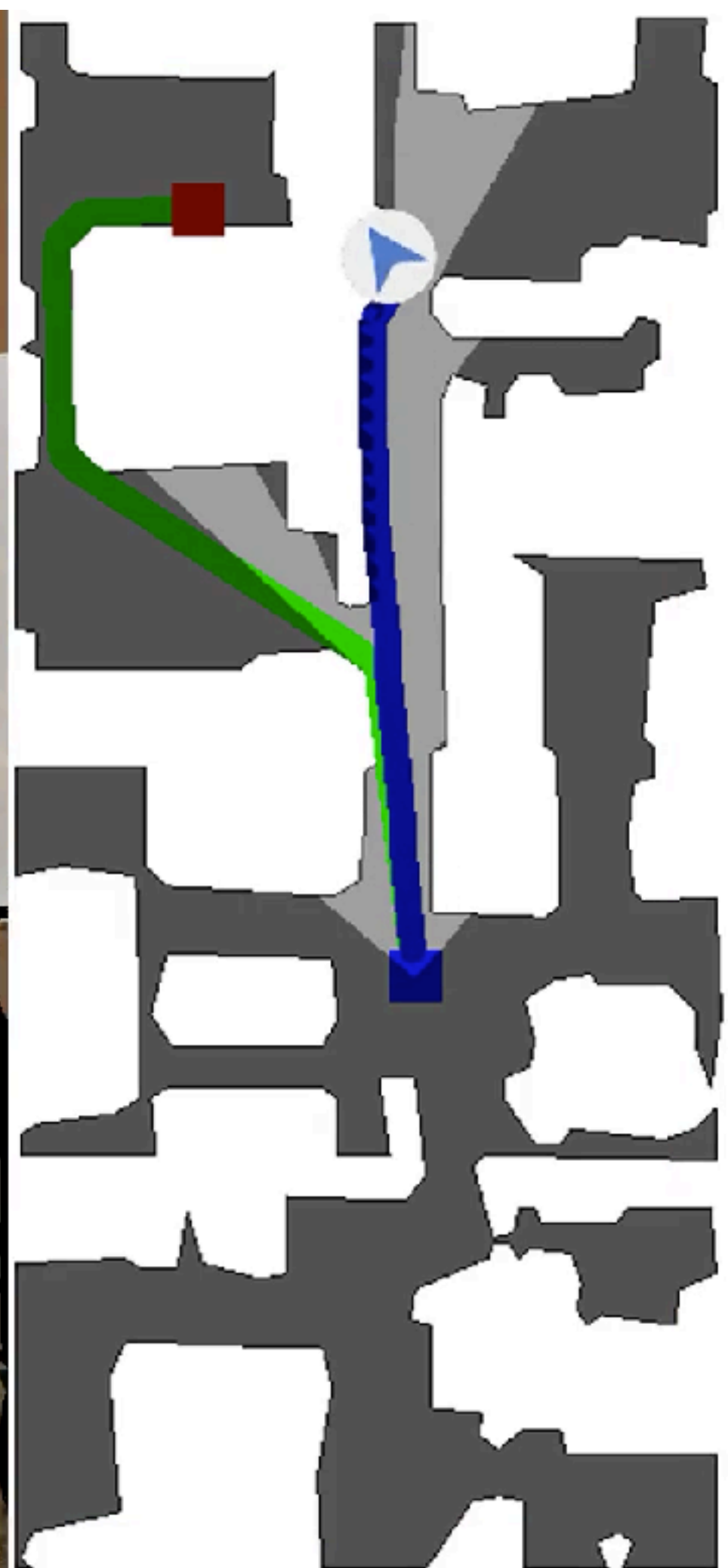
Top Down Map



Depth



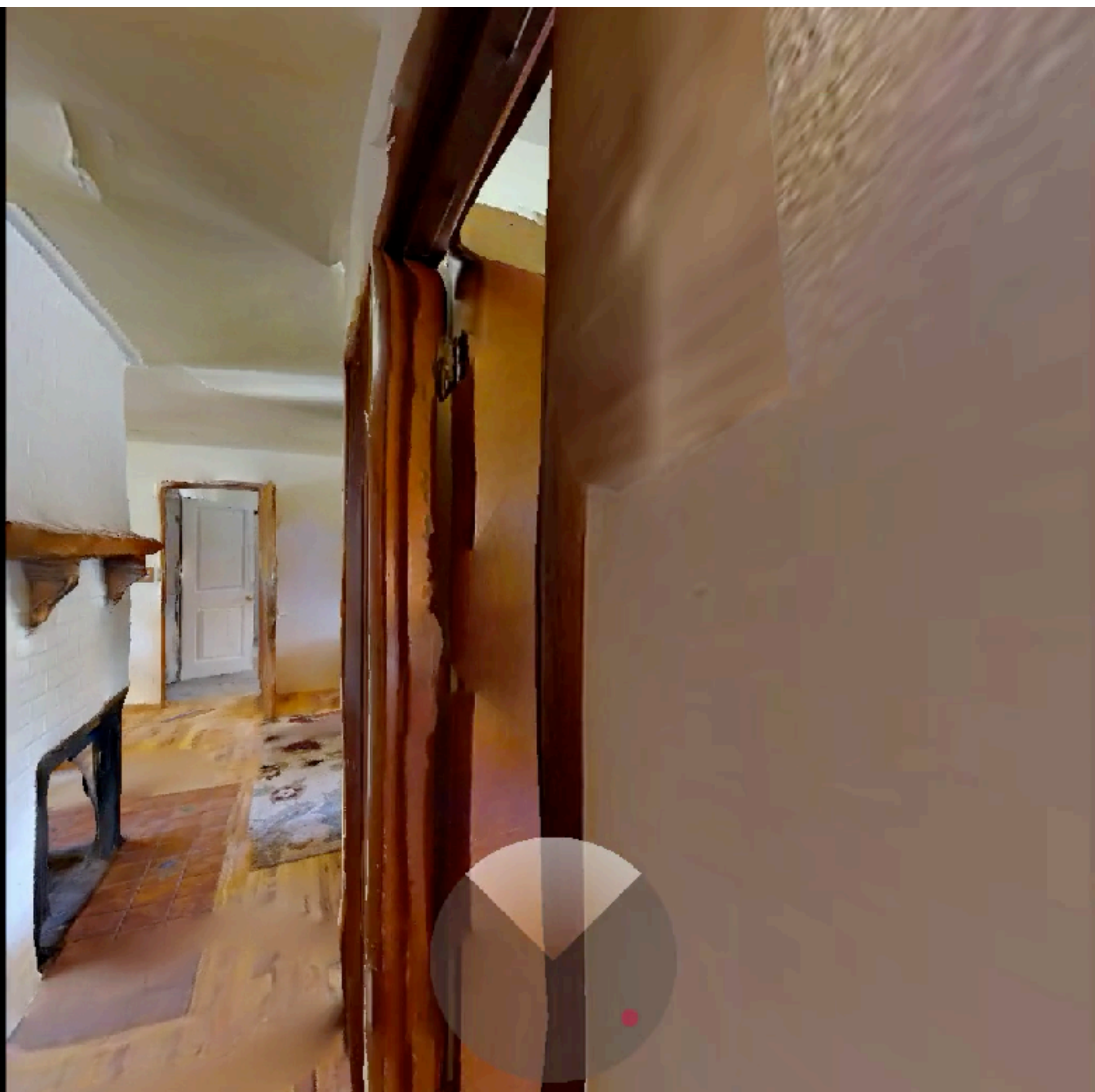
RGB and GPS+Compass



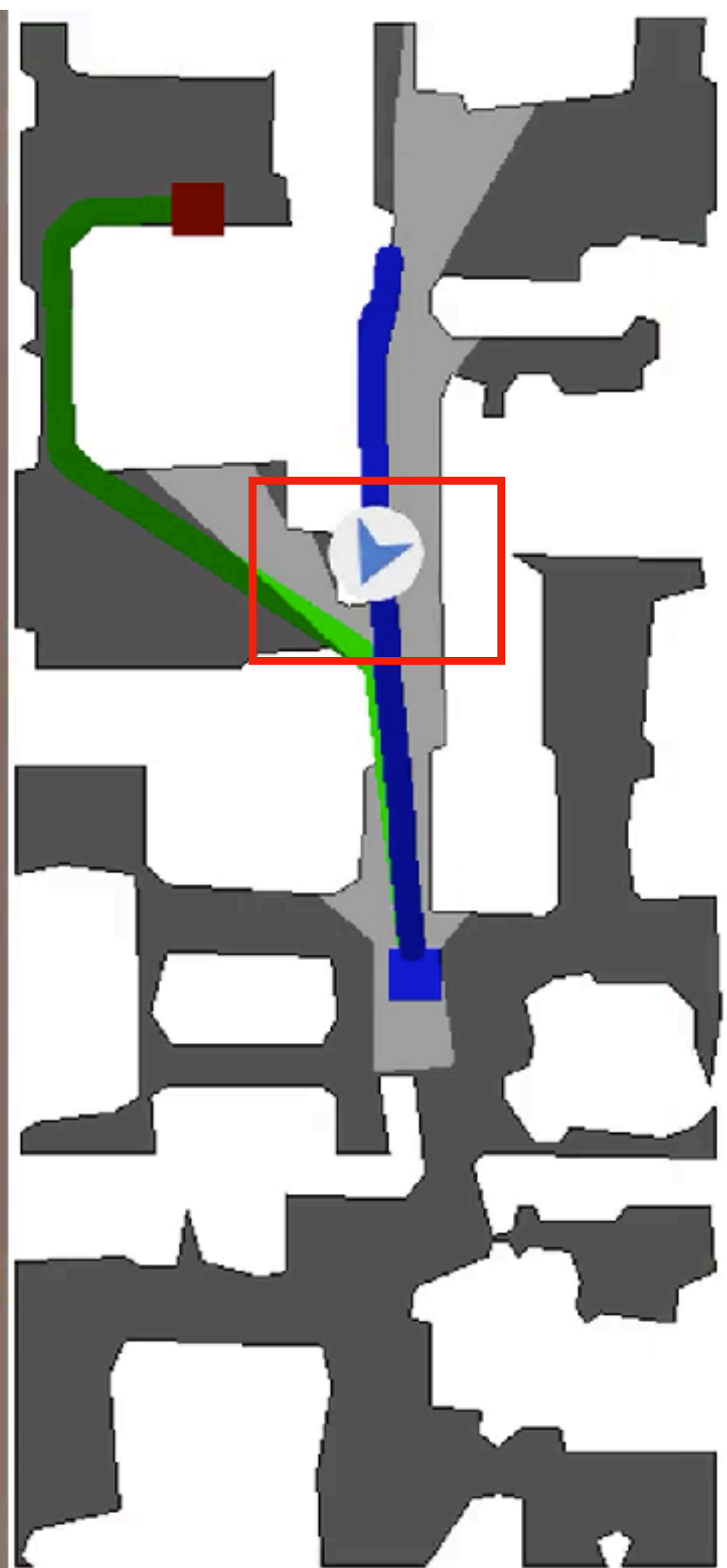
Top Down Map



Depth



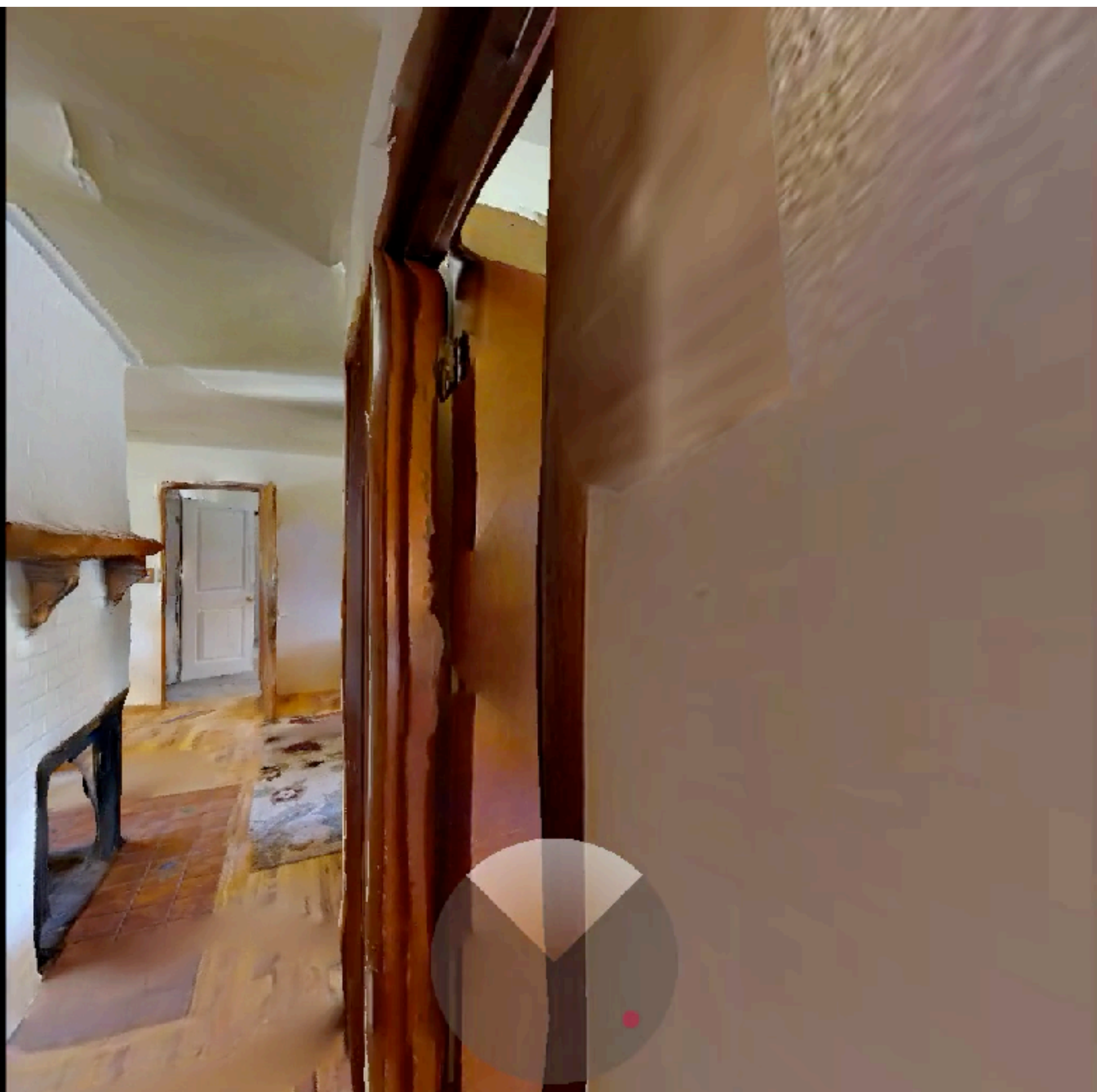
RGB and GPS+Compass



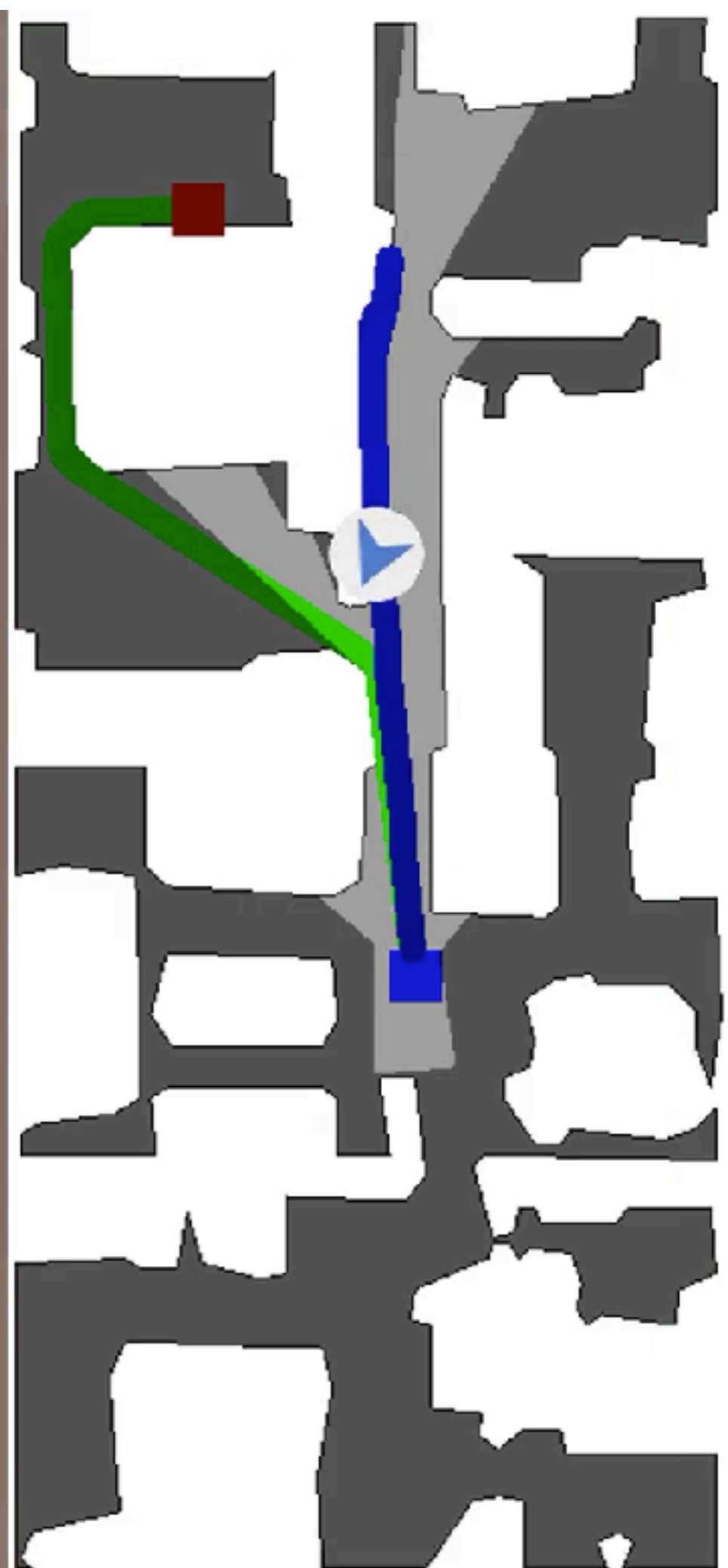
Top Down Map



Depth



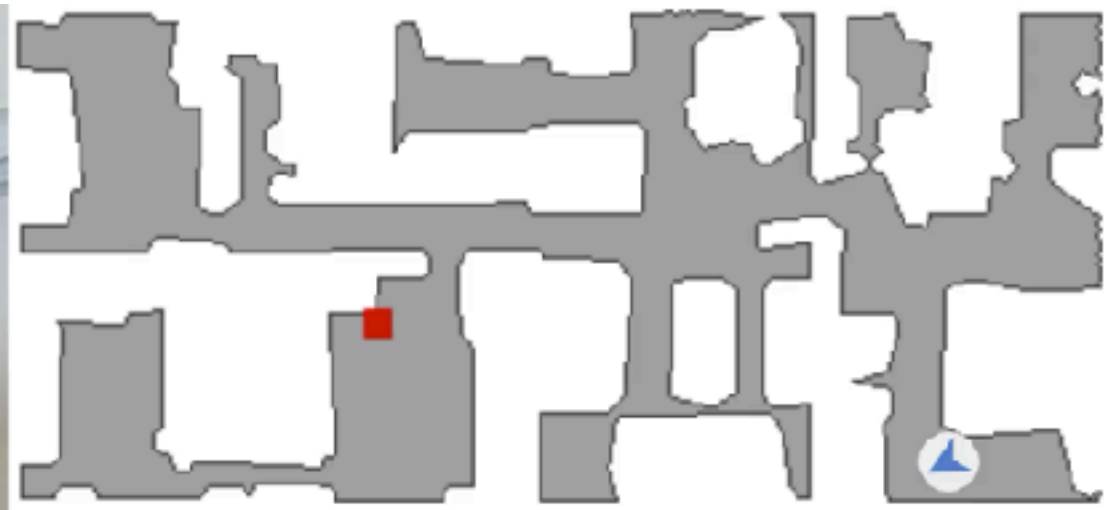
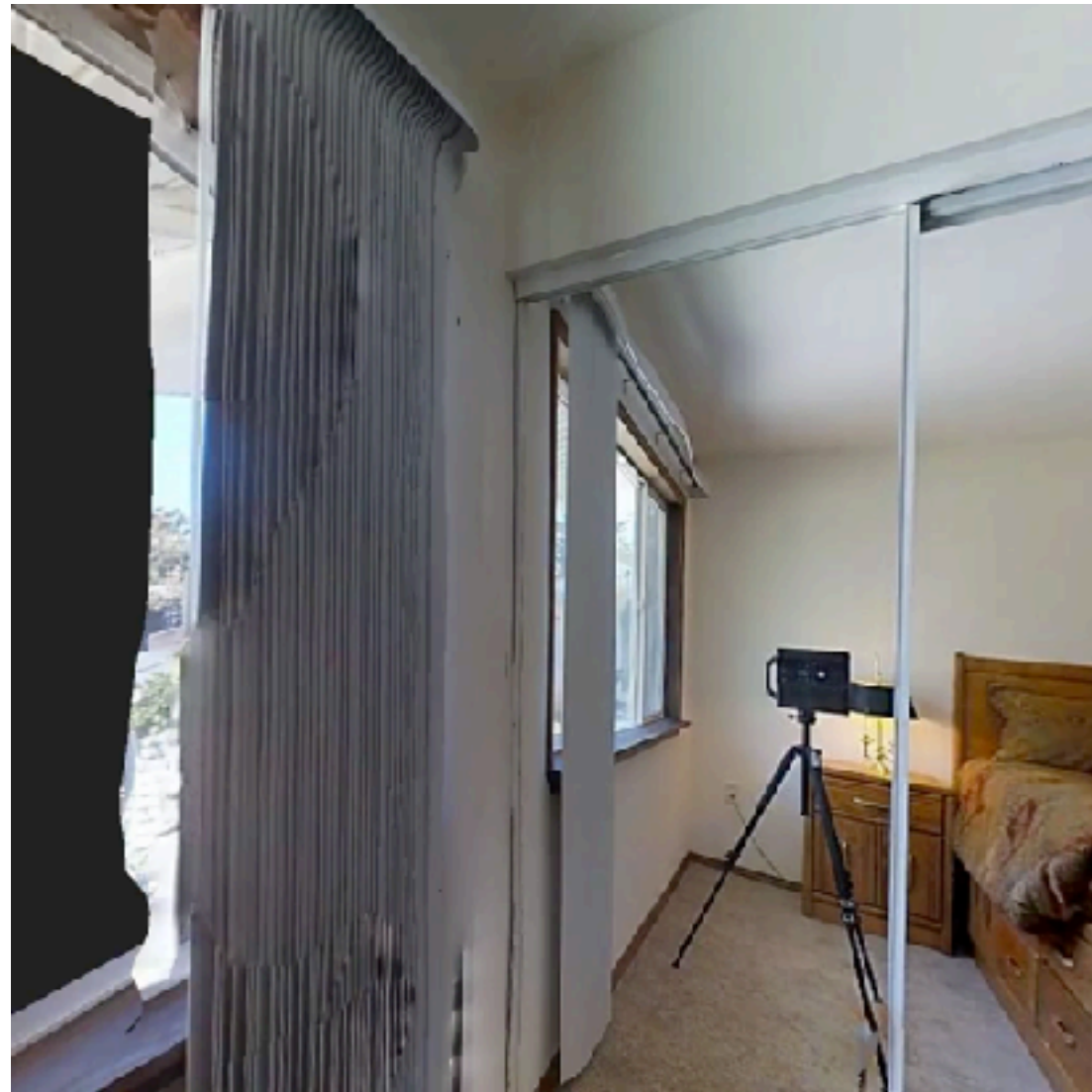
RGB and GPS+Compass



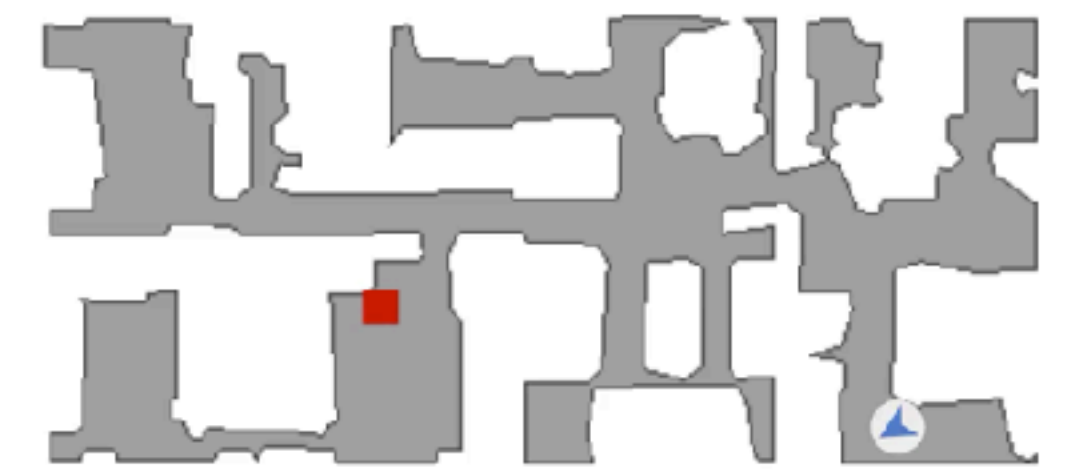
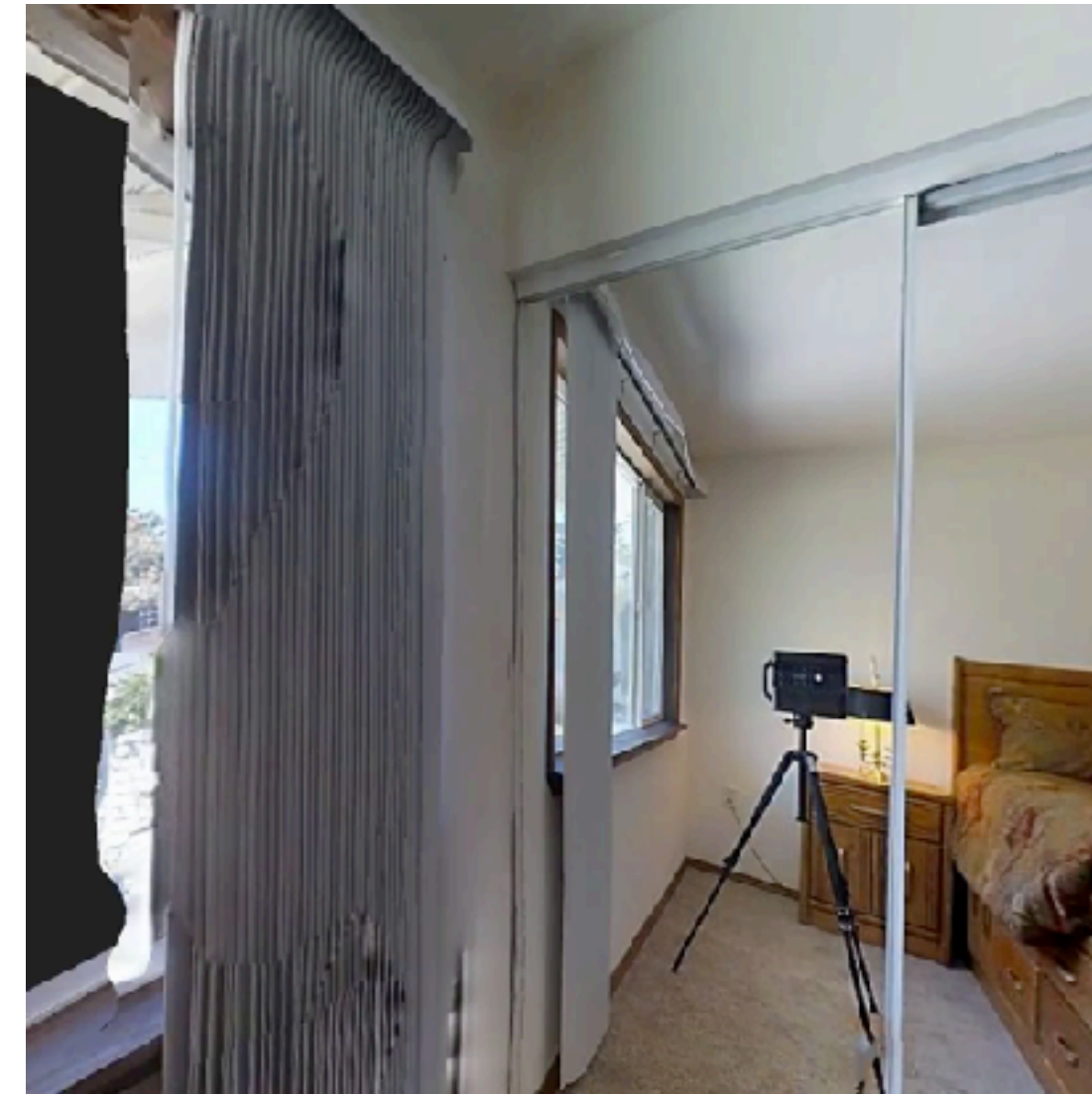
Top Down Map

Visual Turing Test

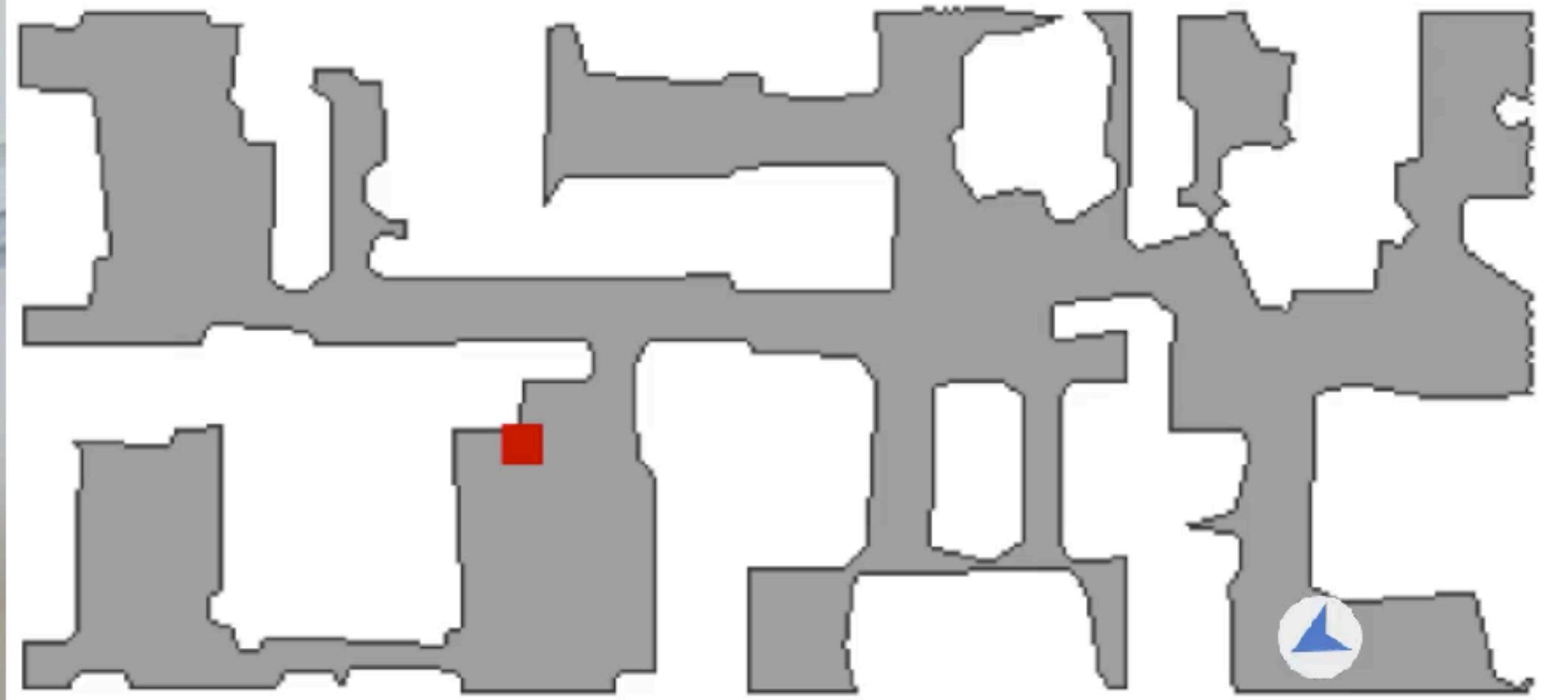
Option 1



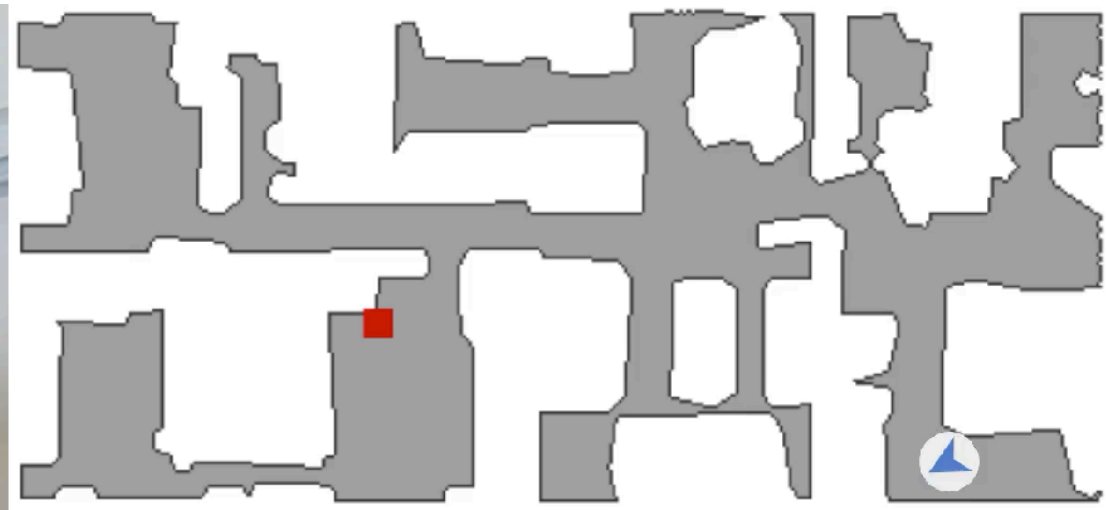
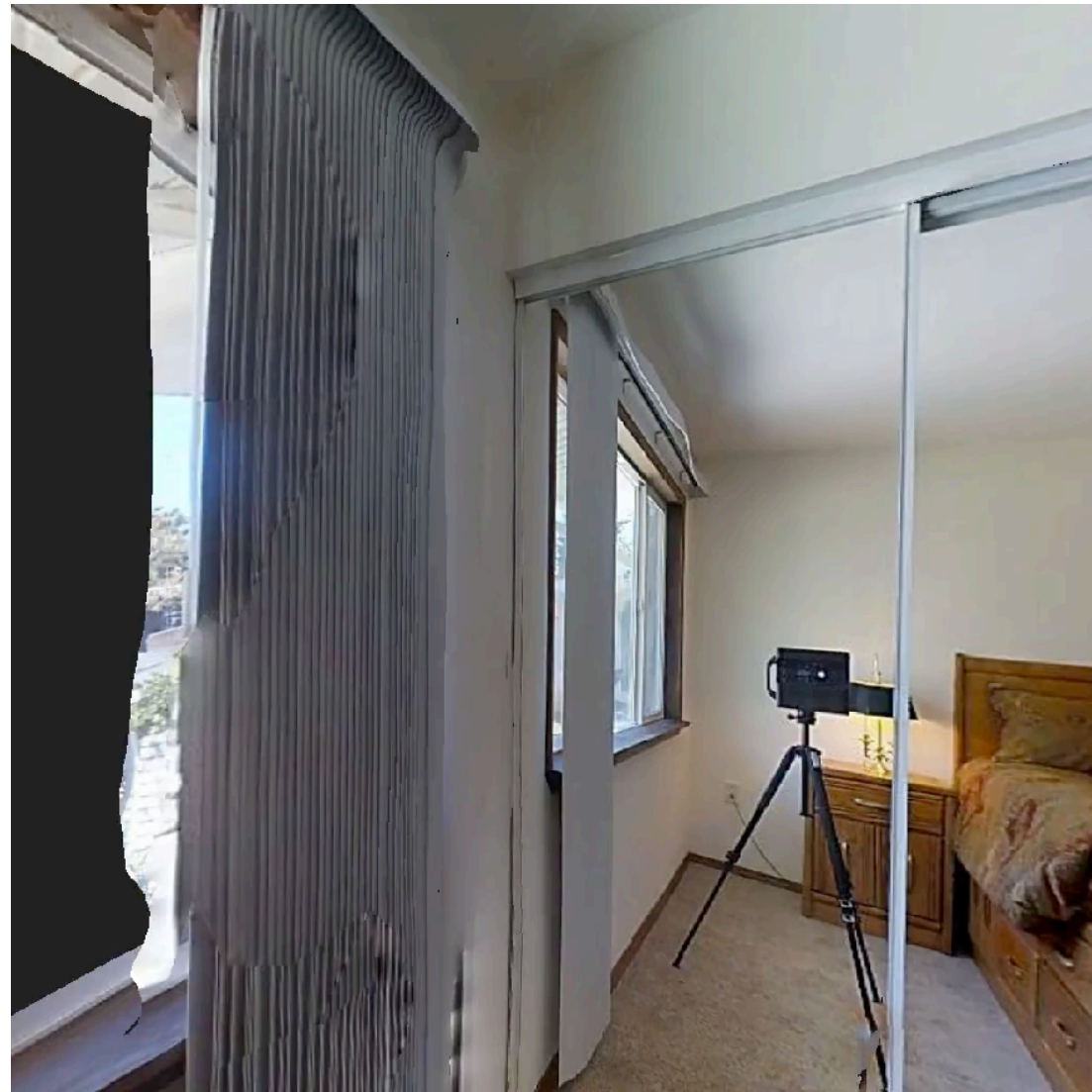
Option 2



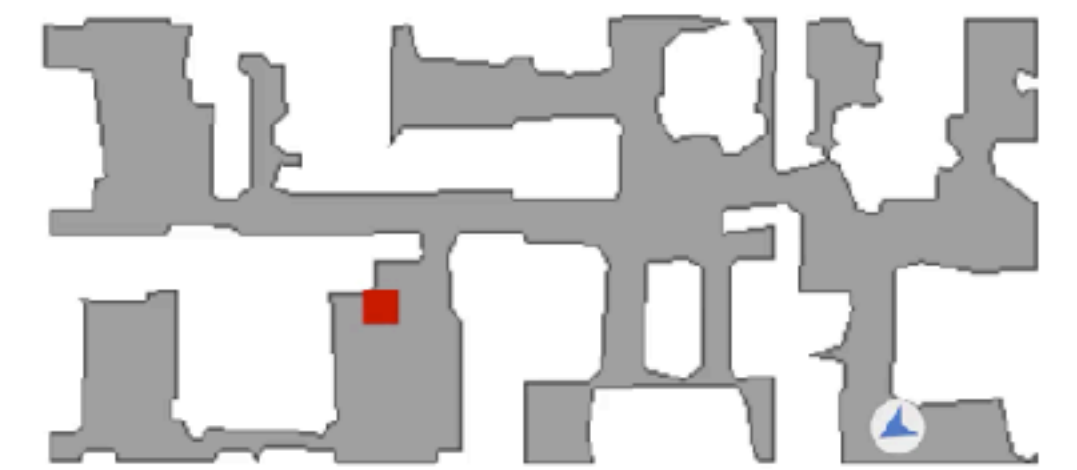
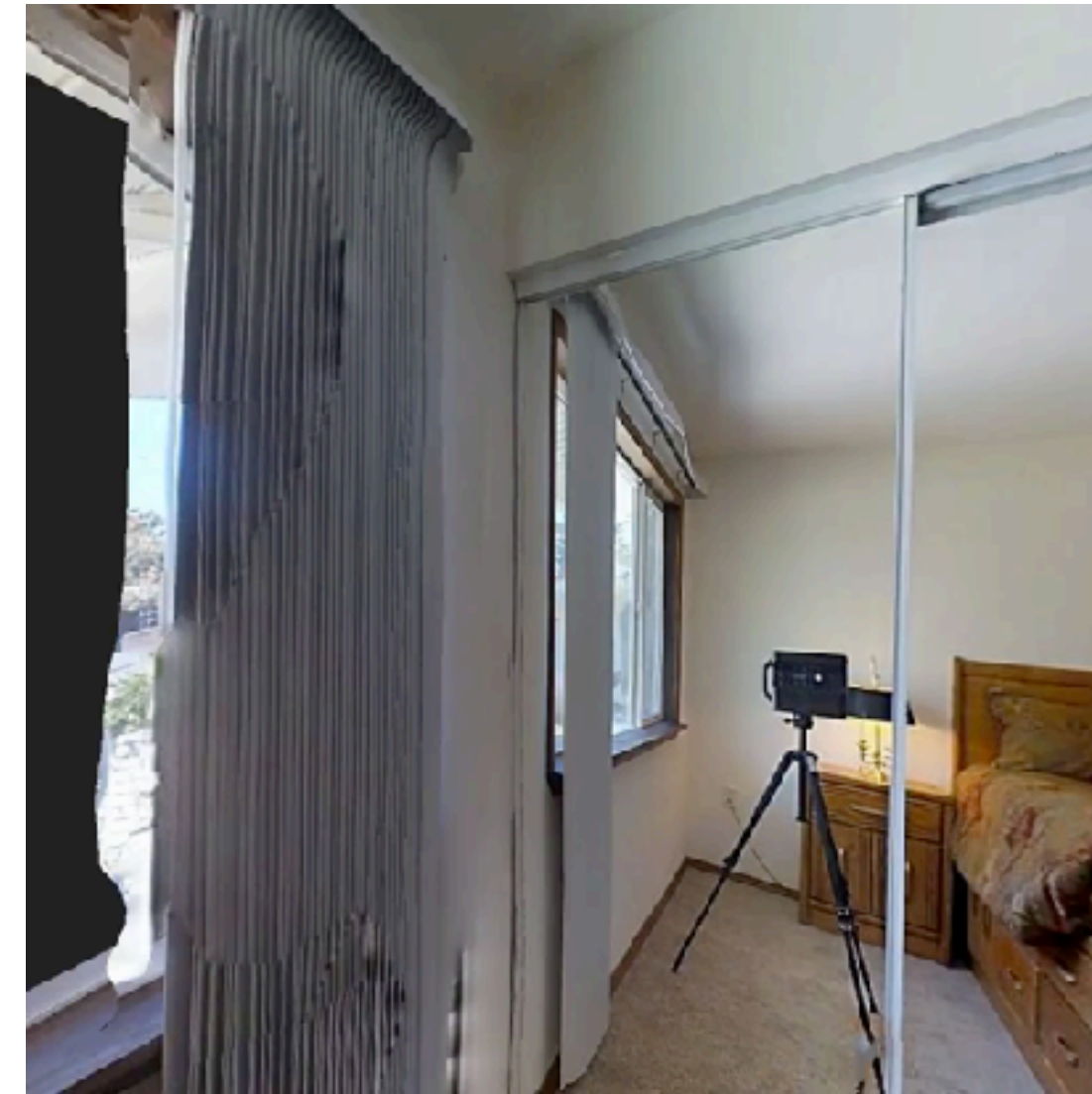
Option 1



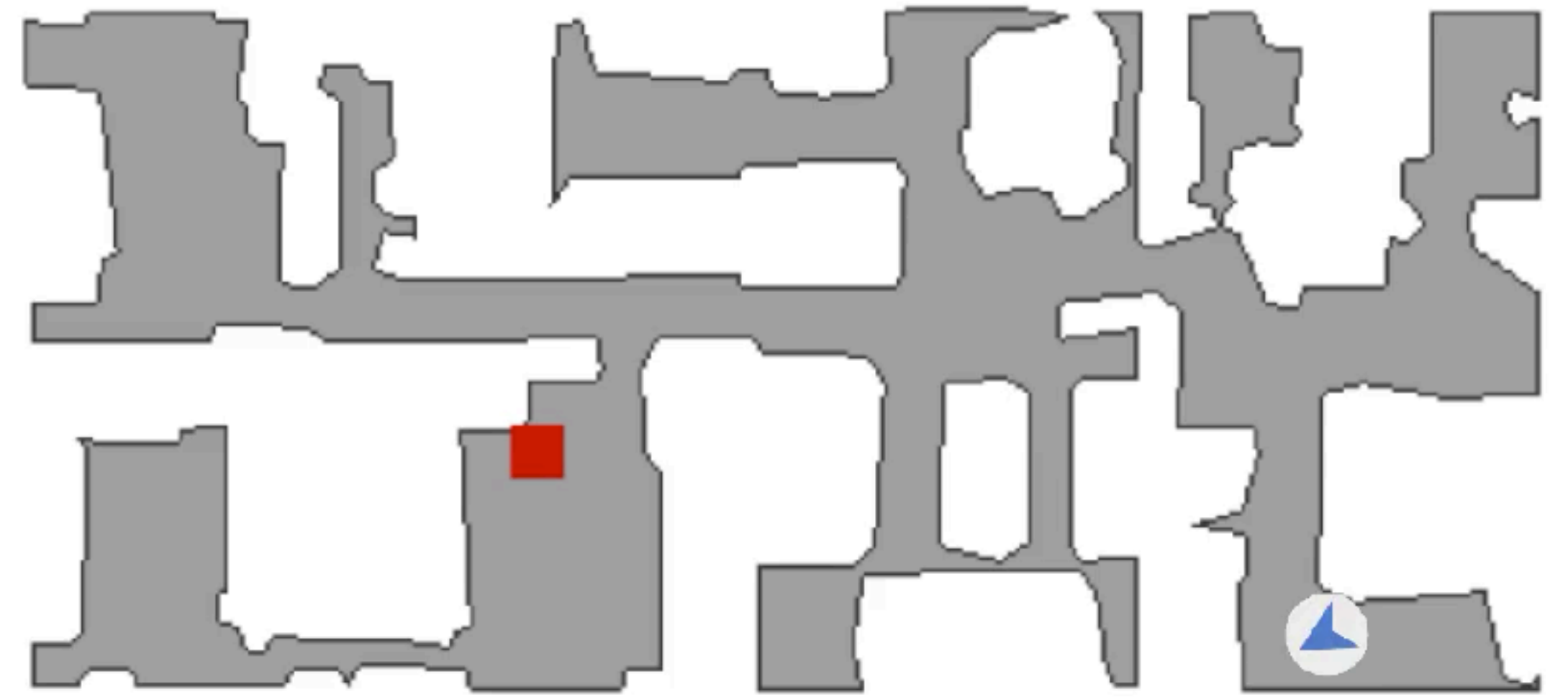
Option 1



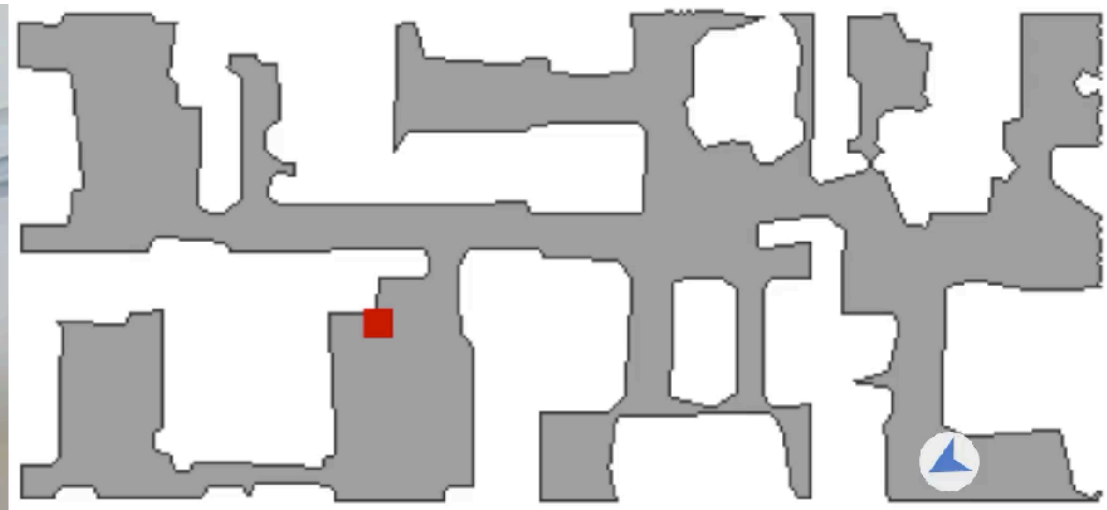
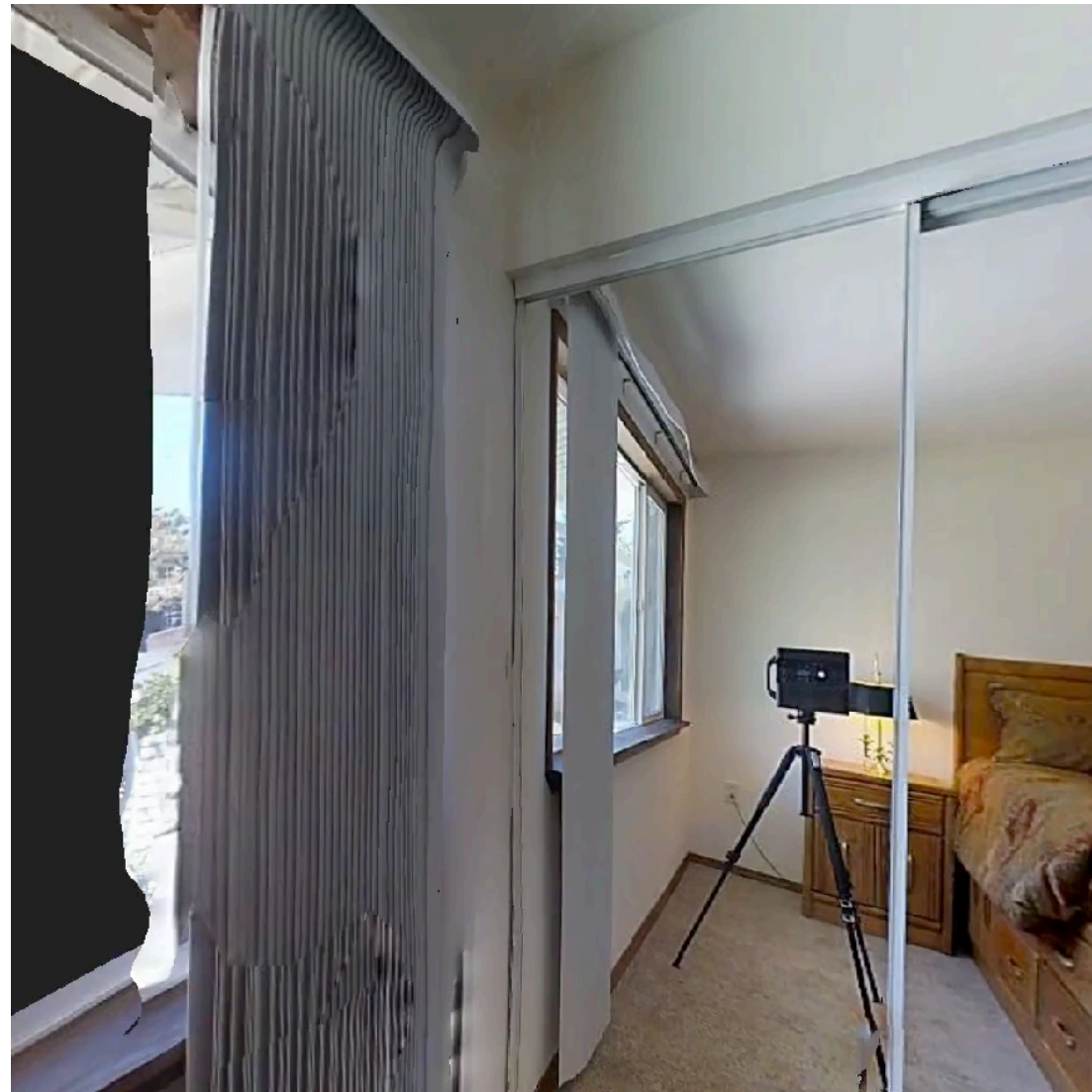
Option 2



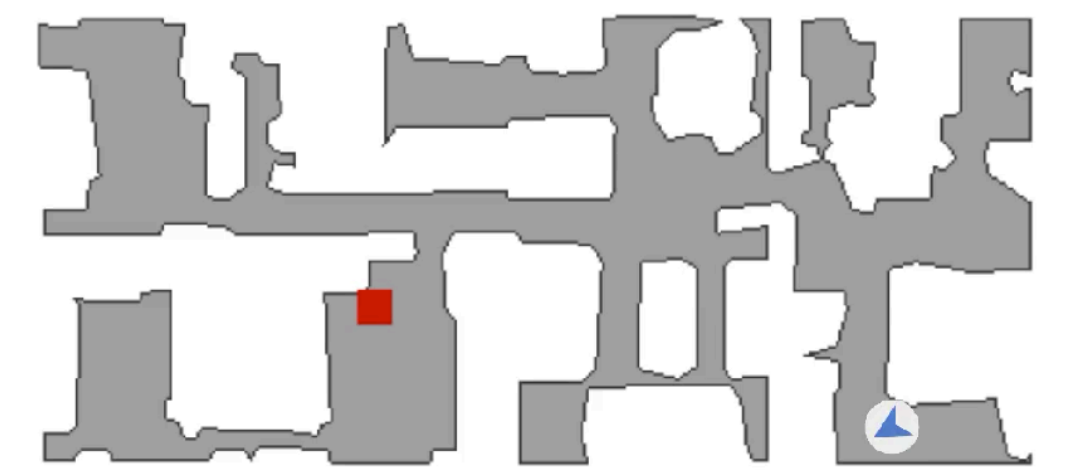
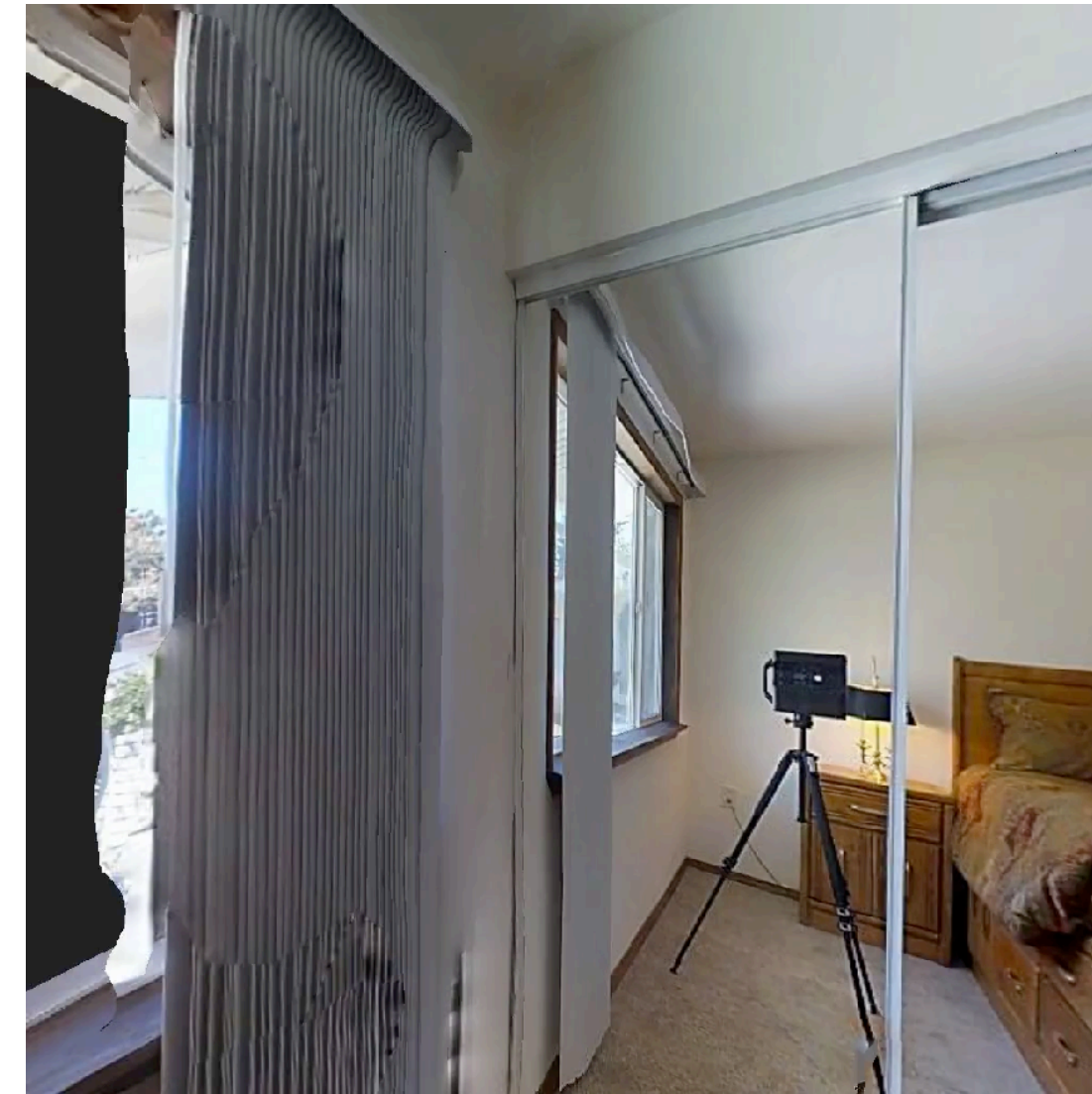
Option 2



Option 1



Option 2



Learned Agent



Shortest Path Oracle

