Model-Based Exploration in Monitored Markov Decision Processes

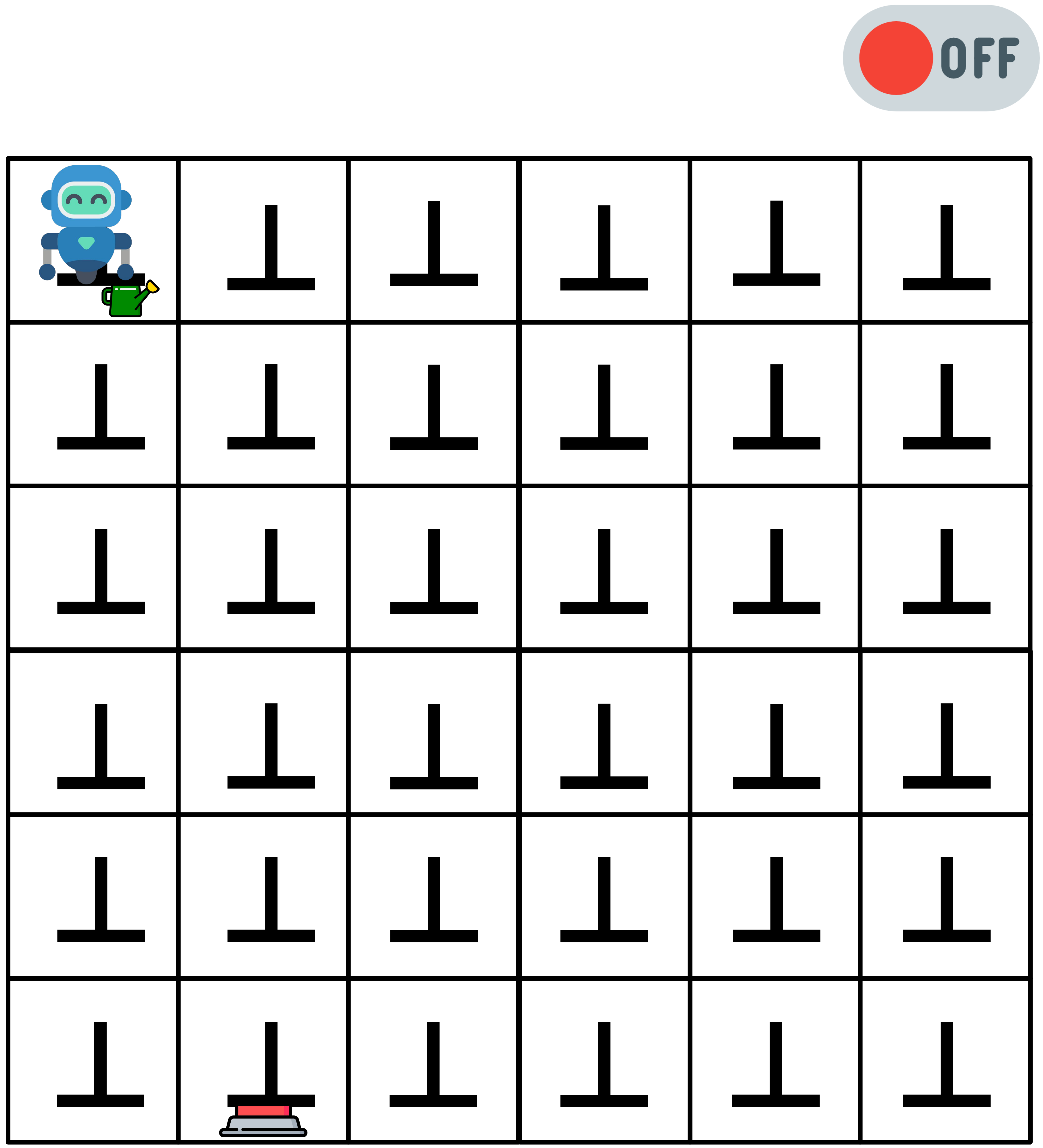
MSc defence seminar

Alireza Kazemipour - August 21st, 2025

What should the agent do?

The agent's supposed to water .

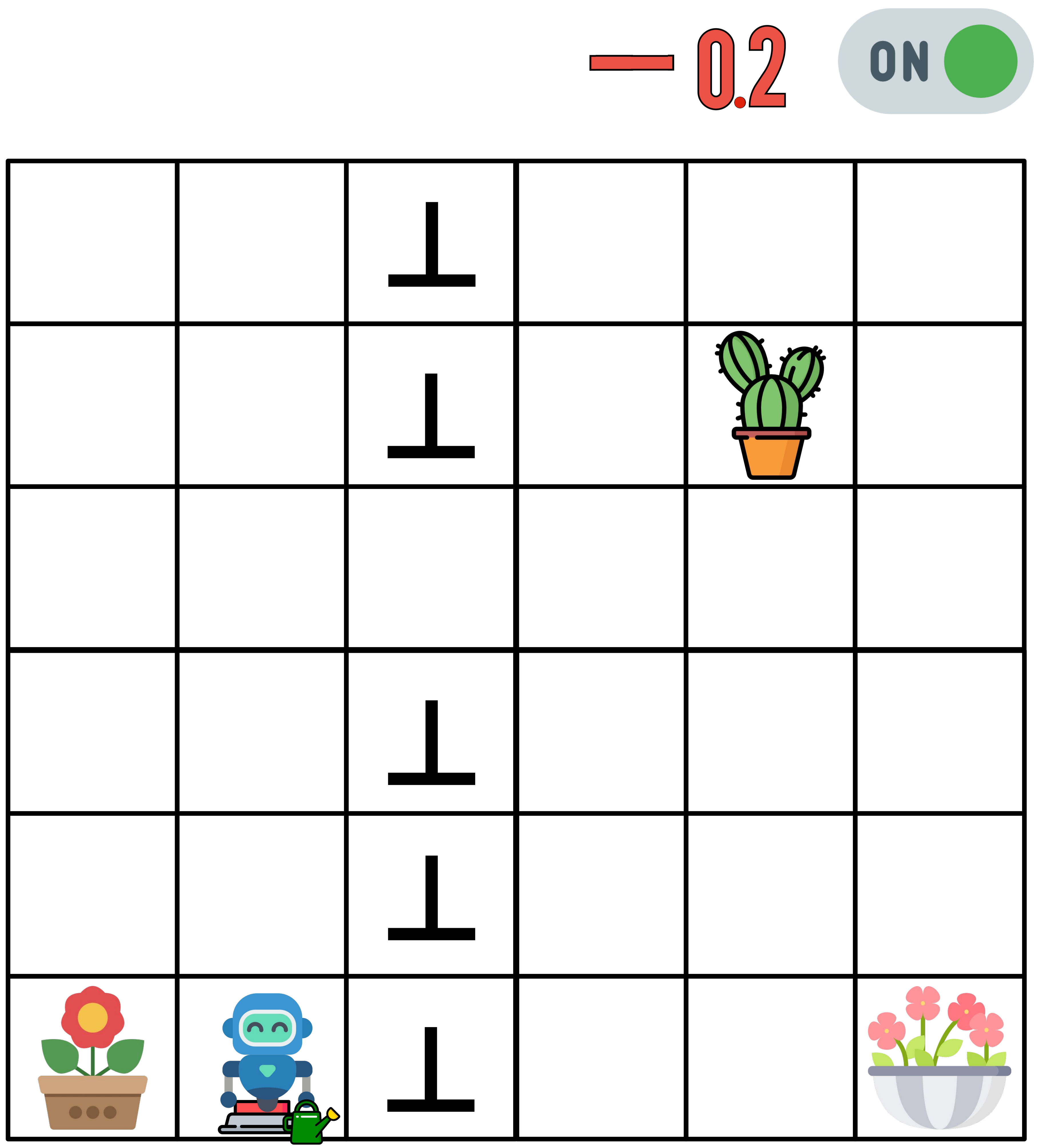
The agent only knows there is at least one  that has the minimum reward a  that has the highest reward.



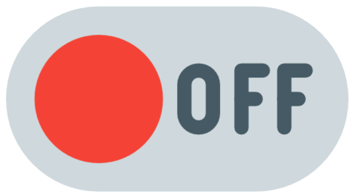
What should the agent do?

The agent's supposed to water .

The agent only knows there is at least one  that has the minimum reward a  that has the highest reward.

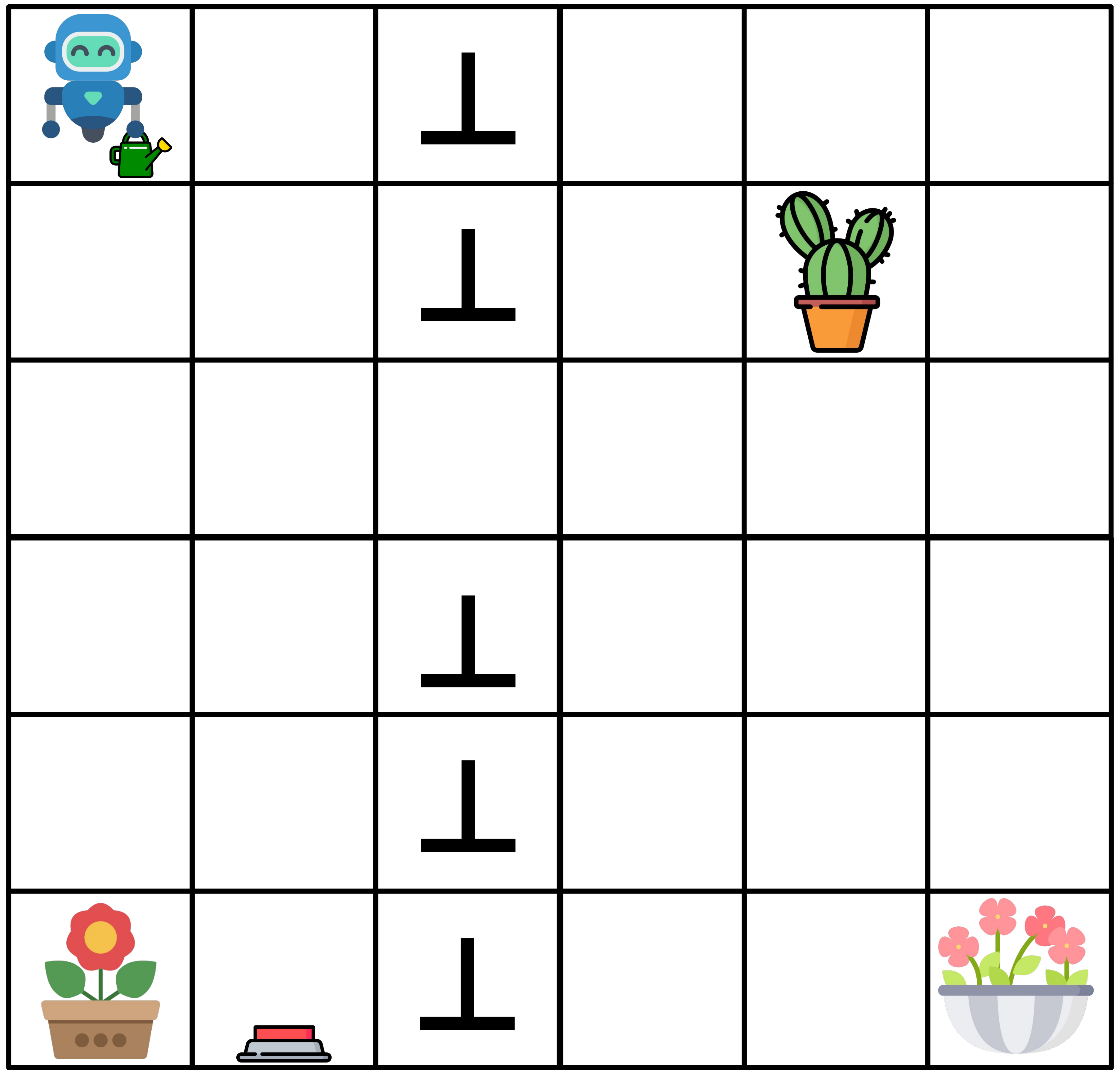


What should the agent do?



The agent's supposed to water .

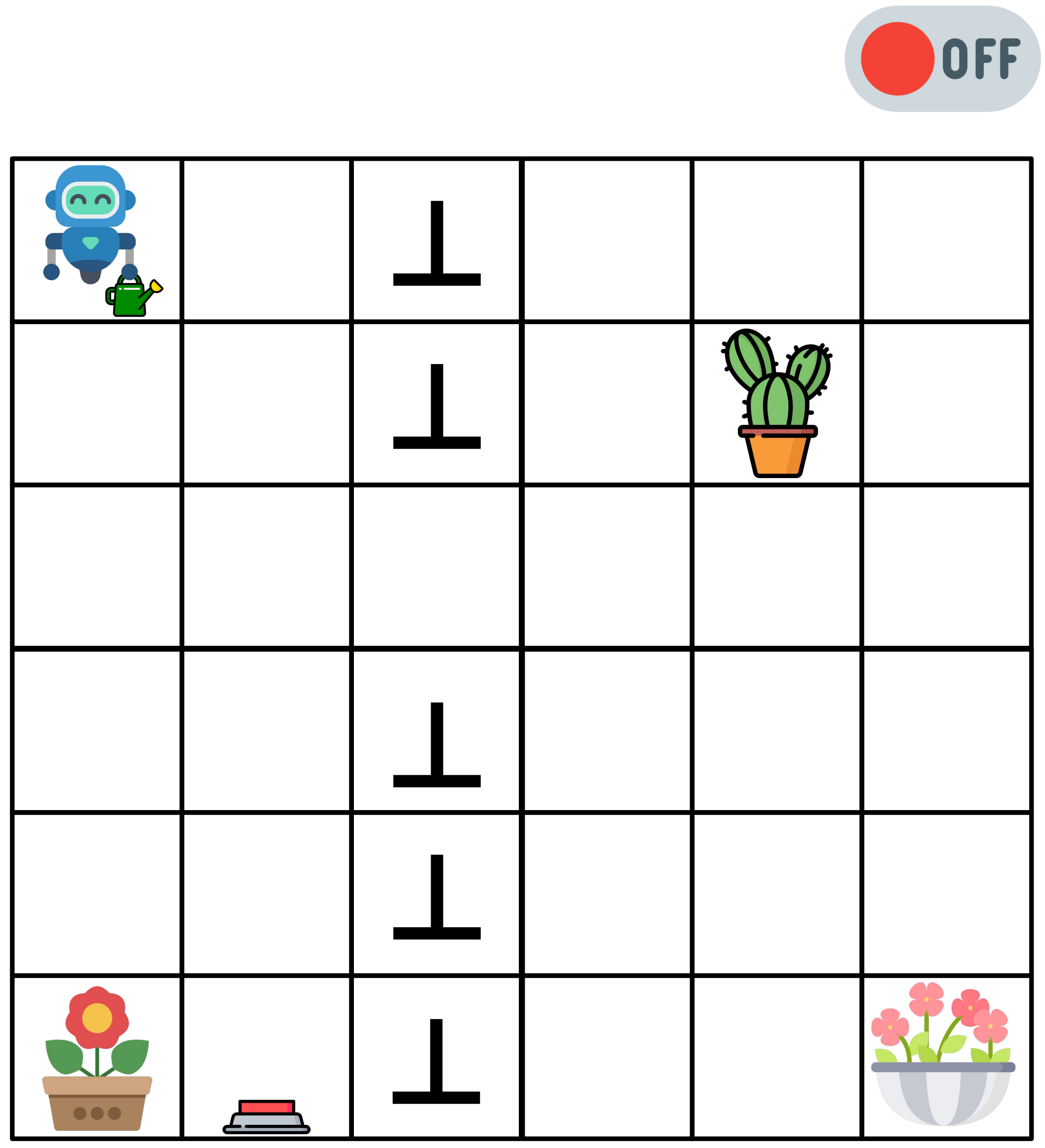
The agent only knows there is at least one  that has the minimum reward a  that has the highest reward.



What should the agent do?

The agent's supposed to water .

The agent only knows there is at least one  that has the minimum reward a  that has the highest reward.

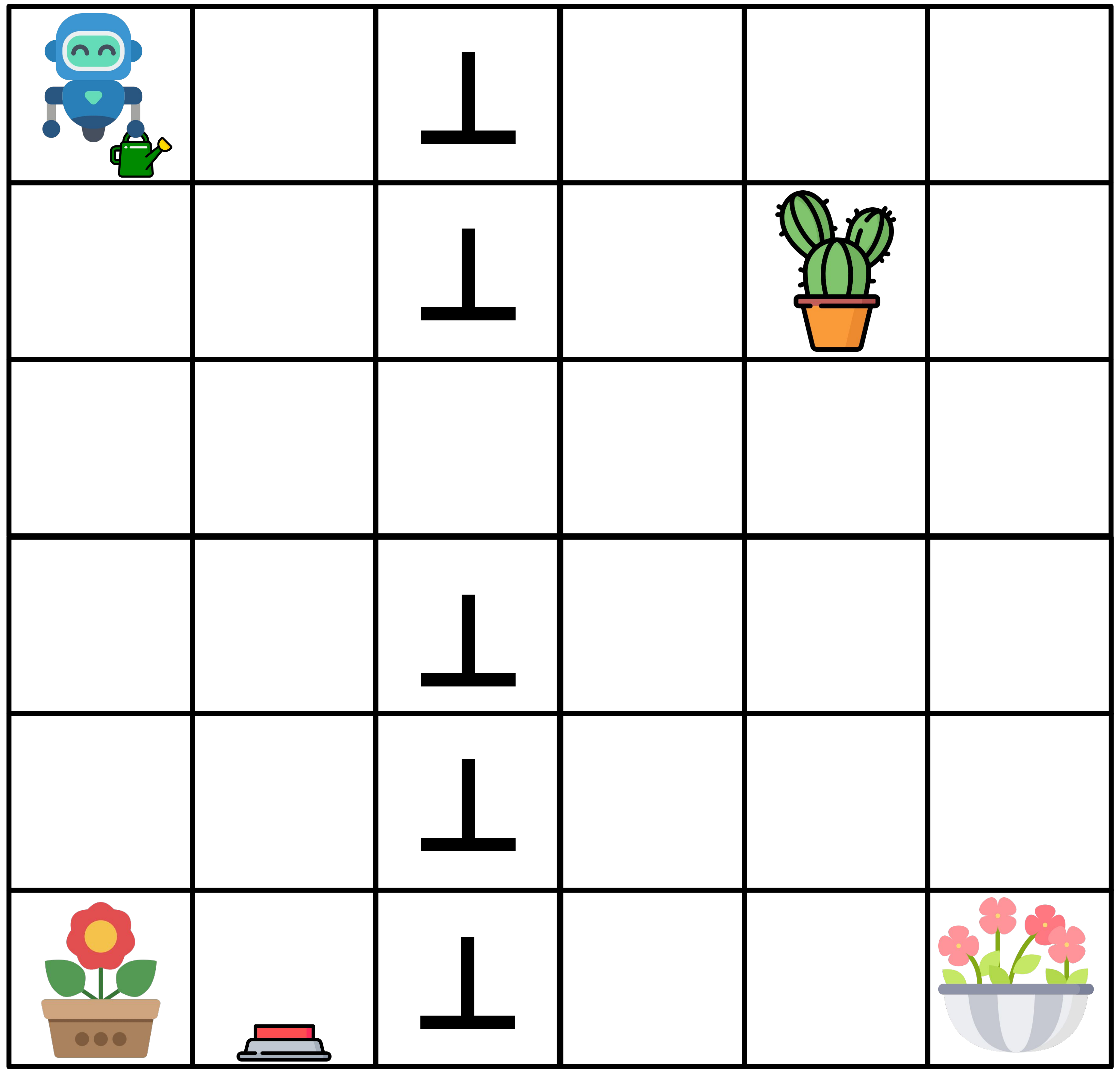


I'm going to answer the question in the slide's title in this talk.

What should the agent do?

Breaking the overarching question into subproblems

1. How to detect ⊥ cells from all the others?
2. How to deal with ⊥ cells?
3. Can the agent be efficient in watering  while not impacting (1) and (2)?



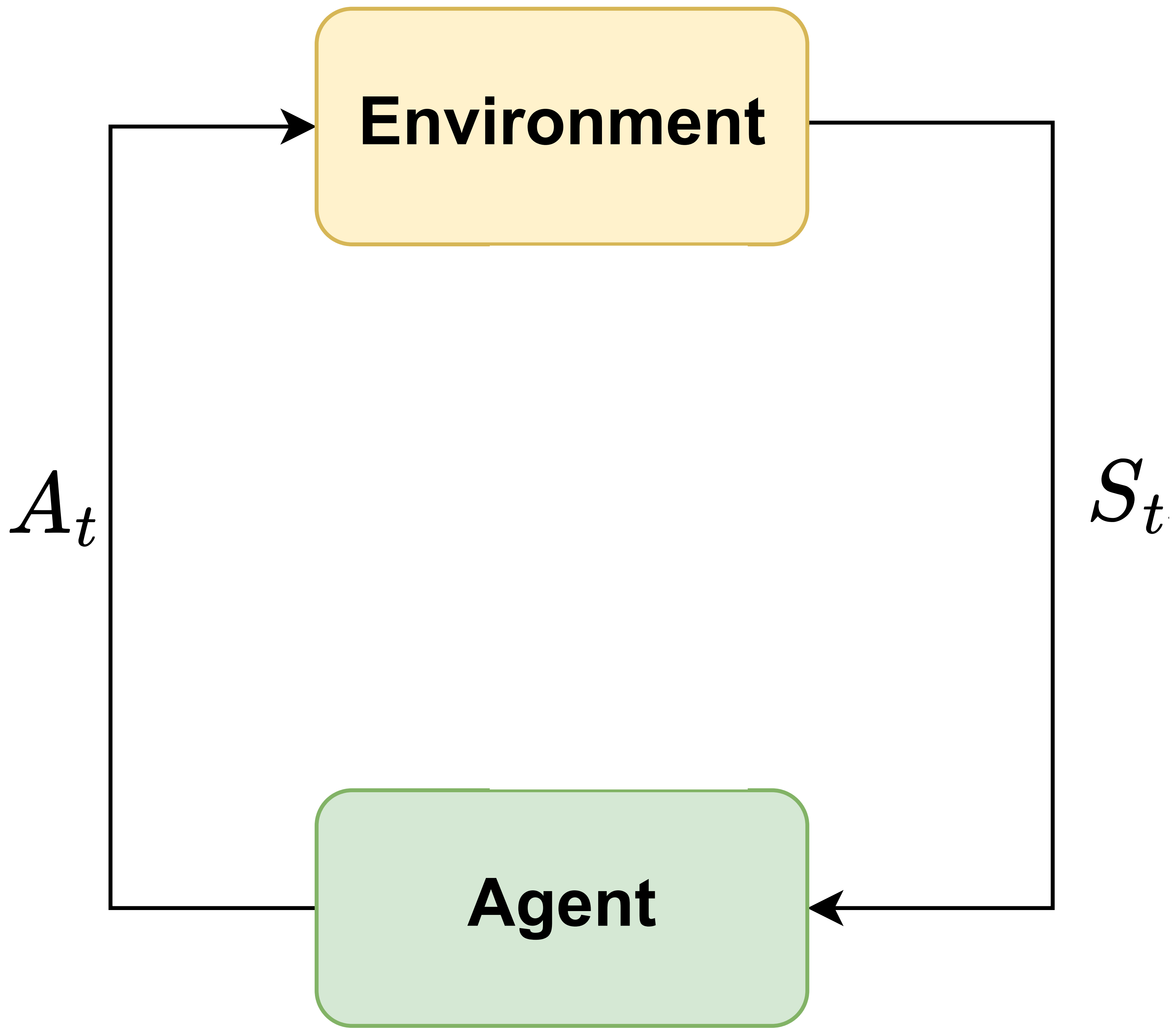
This talk

- Review:
  - Markov Decision Processes
  - Model-Based Interval Estimation with Exploration Bonus (MBIE-EB)
- Problem setting:
  - Monitored Markov Decision Processes
- My proposed solution: Monitored MBIE-EB
  - Theoretical performance
  - Empirical performance
- List of contributions
- Future work
- Acknowledgement

Review

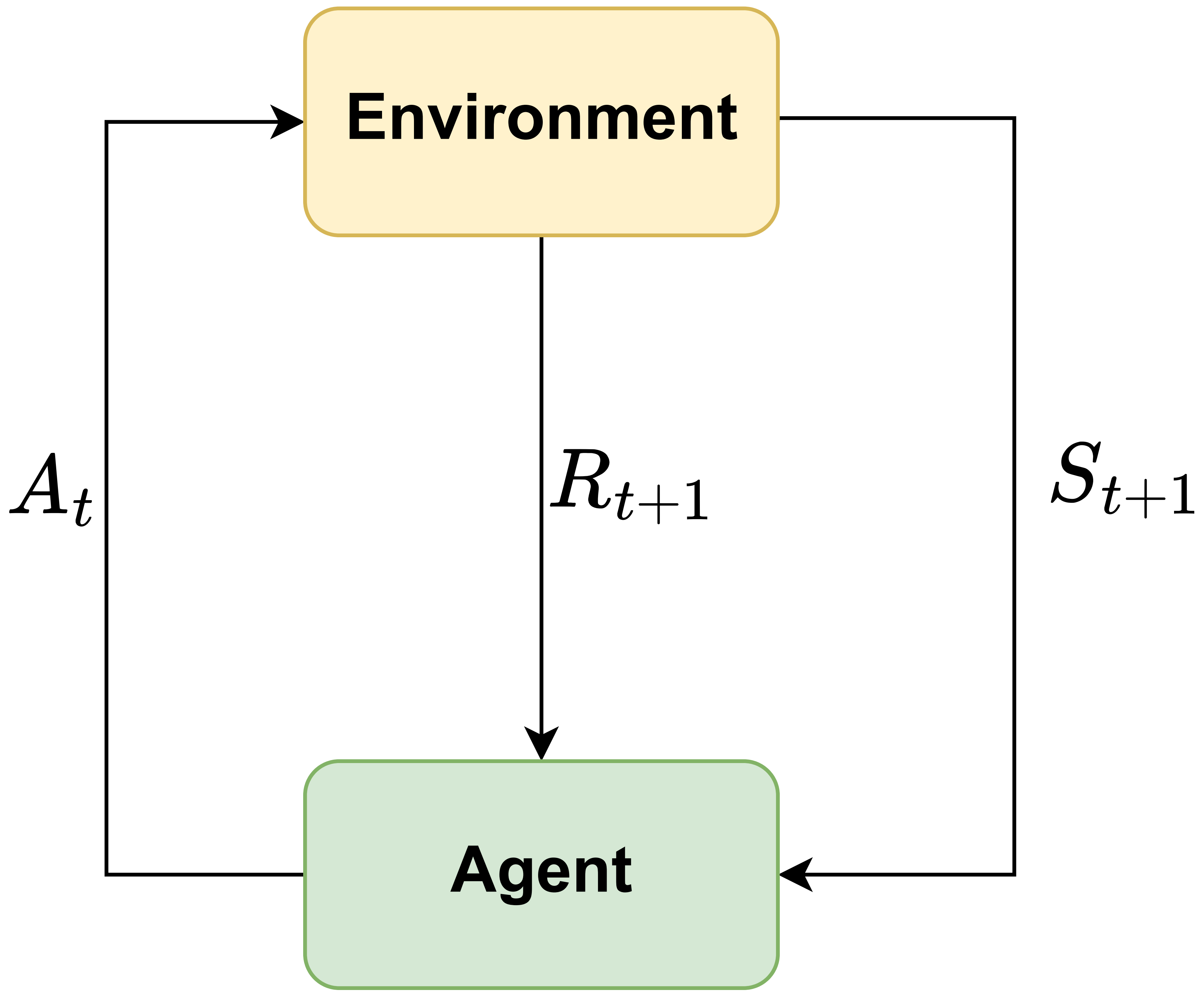
Markov decision processes (MDPs)

A typical mathematical model of interaction in RL



Markov decision processes (MDPs)

A typical mathematical model of interaction in RL



Markov decision processes (MDPs)

A typical mathematical model of interaction in RL

The goal: $\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t R_{t+1}\right]$

A finite MDP: $\langle \mathcal{S}, \mathcal{A}, r, p, \gamma \rangle$

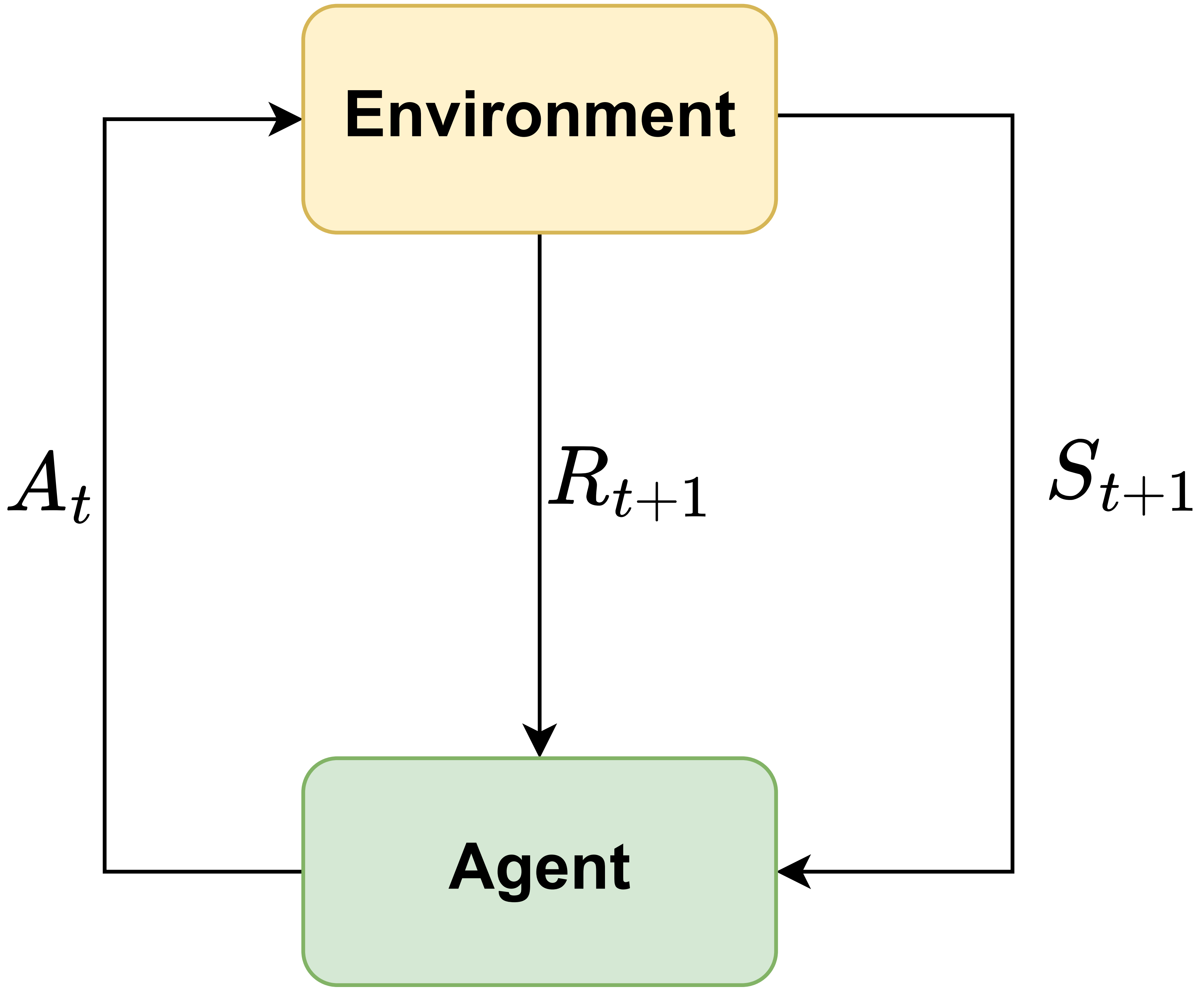
$\mathcal{S}$ is the state space

$\mathcal{A}$ is the action space

r is the expected immediate reward

p is the transition dynamics

$0 \leq \gamma < 1$



Markov decision processes (MDPs)

How to maximize the expected discounted return using models?

Follow

$$Q^*(s,a) = r(s,a) + \gamma \sum_{s'} p\left(s' \middle| s, a\right) V^*(s').$$

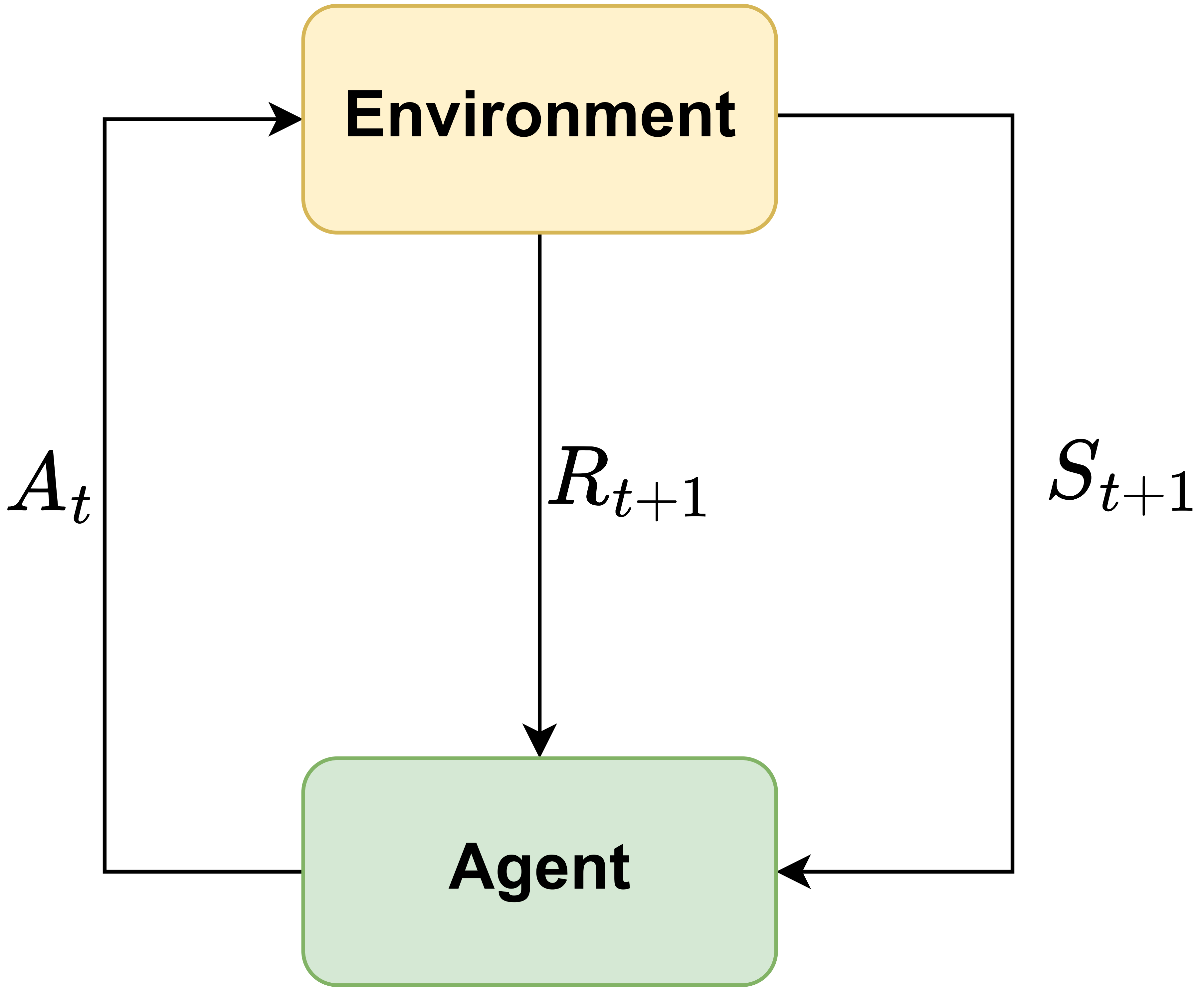
The model-based learning's challenge:

We know sample estimates $\hat{r}$, and $\hat{p}$,
but we don't know the true r and p! 😭

One solution:

Using measures on how uncertain we are about $\hat{r}$, and $\hat{p}$.

If we are confident about the quality of sample estimates, then we are golden.



Measuring uncertainty

- Suppose you have n samples. Then

$$\texttt{distance}\left(\text{empirical mean, true mean}\right) \leq \frac{\beta}{\sqrt{n}}$$

- If you particularly have n Bernoulli samples. Then

$$\texttt{distance}\left(\text{empirical mean, true mean}\right) \leq \frac{\beta}{n}$$

for sufficiently large value of β.

$\frac{\beta}{\sqrt{n}}$, and $\frac{\beta}{n}$ measure the **uncertainty**.

Let $\hat{Q}$ denote the action-value functions we get using $\hat{r}$, and $\hat{p}$.

Model-based interval estimation with exploration bonus (MBIE-EB)

There is an algorithm called **MBIE-EB**[1] that is greedy with respect to:

$$\hat{Q}(s, a)$$

1. A. Strehl, et al. "An analysis of model-based Interval Estimation for Markov Decision Processes," (Journal of Computer and System Sciences '08)

Model-based interval estimation with exploration bonus (MBIE-EB)

There is an algorithm called **MBIE-EB**[1] that is greedy with respect to:

$$\hat{Q}(s,a) + \underbrace{\frac{\beta_1}{\sqrt{n}}}_{\text{uncertainty of } \hat{r}}$$

1. A. Strehl, et al. "An analysis of model-based Interval Estimation for Markov Decision Processes," (Journal of Computer and System Sciences '08)

Model-based interval estimation with exploration bonus (MBIE-EB)

There is an algorithm called **MBIE-EB**[1] that is greedy with respect to:

$$\hat{Q}(s,a) + \underbrace{\frac{\beta_1}{\sqrt{n}}}_{\text{uncertainty of } \hat{r}} + \underbrace{\frac{\beta_2}{\sqrt{n}}}_{\text{uncertainty of } \hat{p}}$$

n: number visits to (s, a)

1. A. Strehl, et al. "An analysis of model-based Interval Estimation for Markov Decision Processes," (Journal of Computer and System Sciences '08)

Model-Based interval estimation with exploration bonus (MBIE-EB)

There is an algorithm called **MBIE-EB** that is greedy with respect to:

$$\hat{Q}(s,a) + \underbrace{\frac{\beta_1}{\sqrt{n}}}_{\text{uncertainty of } \hat{r}} + \underbrace{\frac{\beta_2}{\sqrt{n}}}_{\text{uncertainty of } \hat{p}}$$

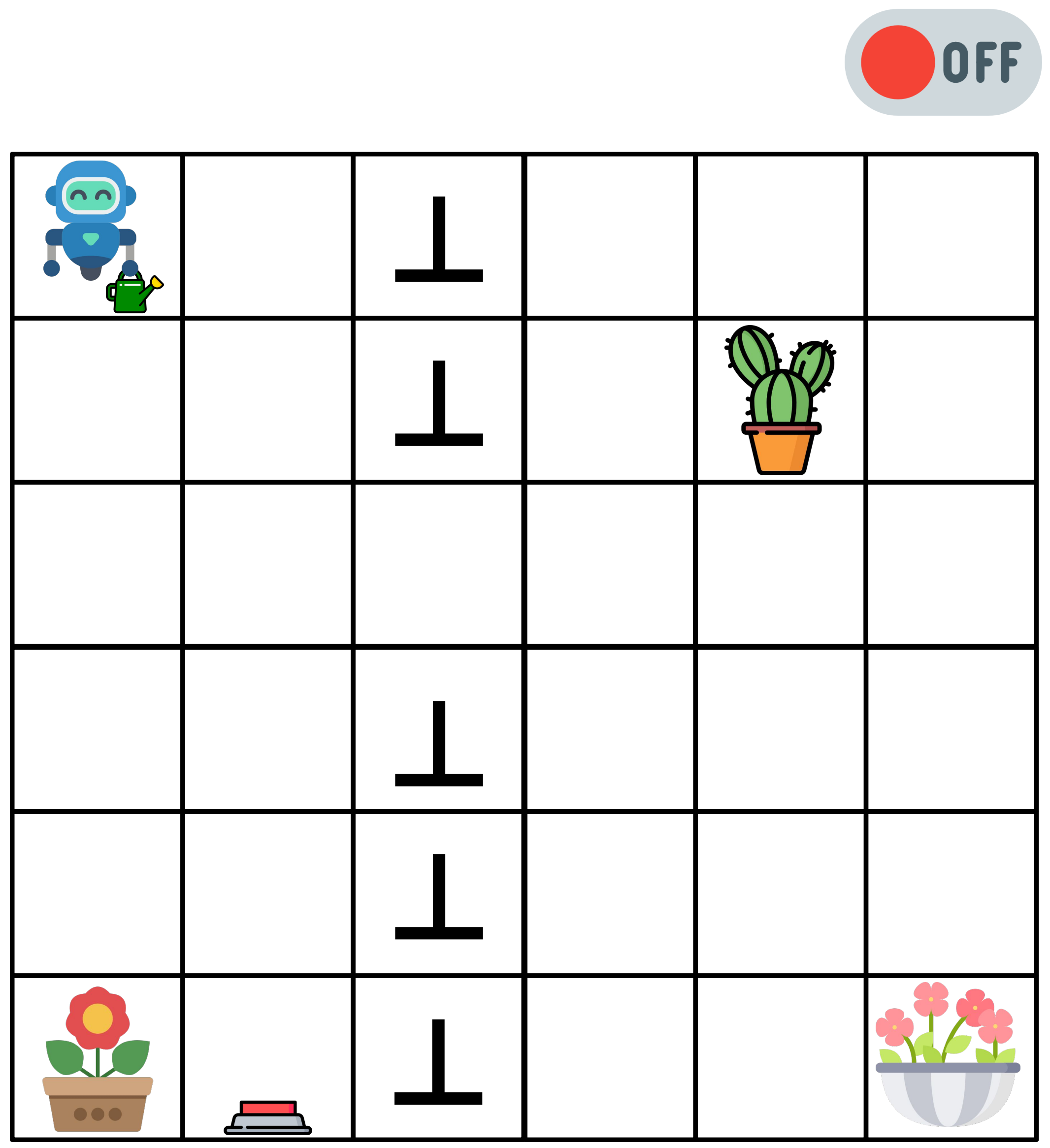
number visits to (s, a)

MBIE-EB is also efficient since it finds an ϵ-optimal policy in following number of time steps:

$$\widetilde{O}\left(\frac{|\mathcal{S}|^2|\mathcal{A}|}{\epsilon^3(1-\gamma)^6}\right)$$

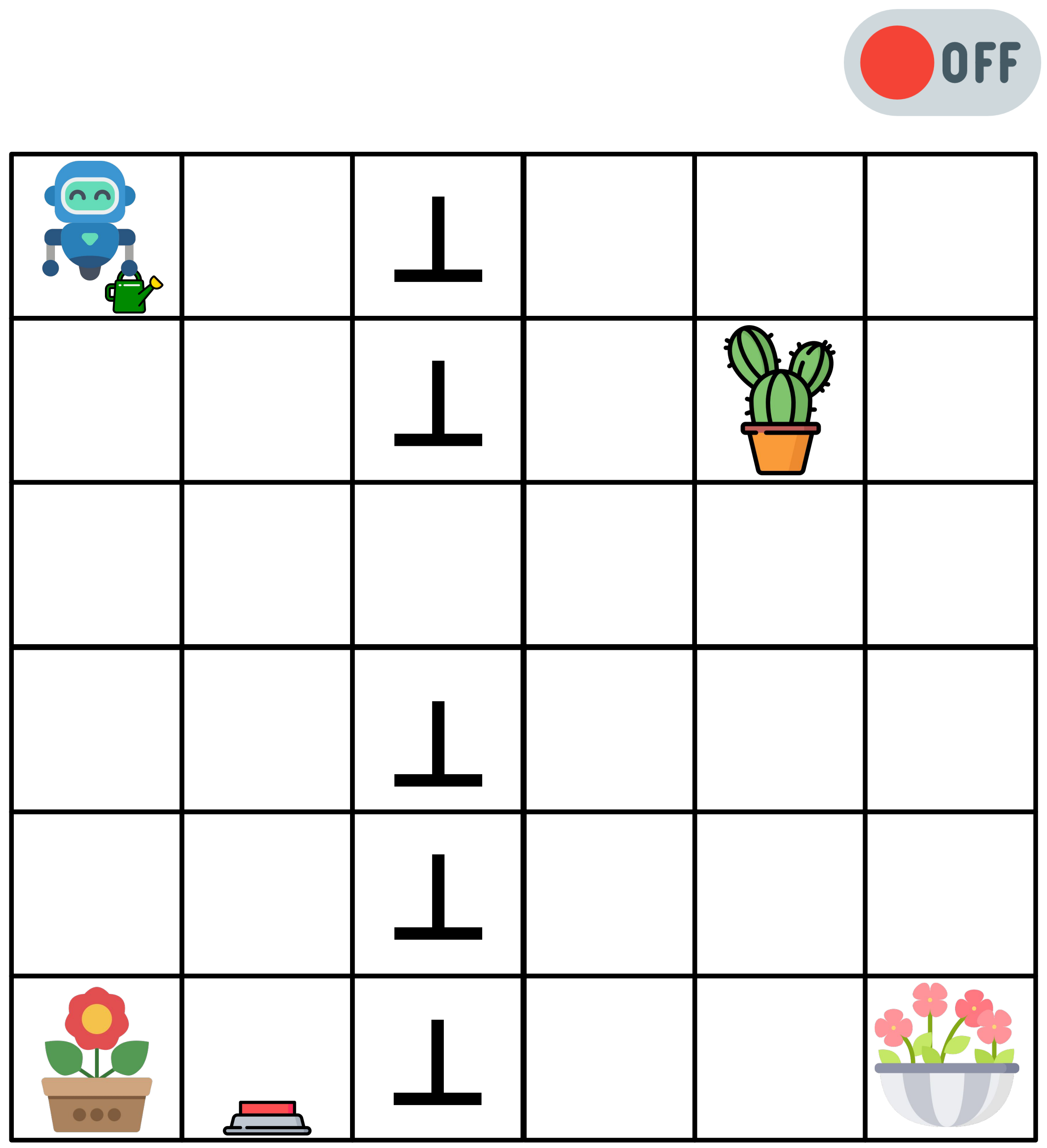
Takeaways so far:

1. A good algorithm like MBIE-EB uses **bonuses** as measures of **uncertainty**
2. **We're interested in solving**



Problem setting

MDPs cannot model



But Monitored MDPs[1] can!

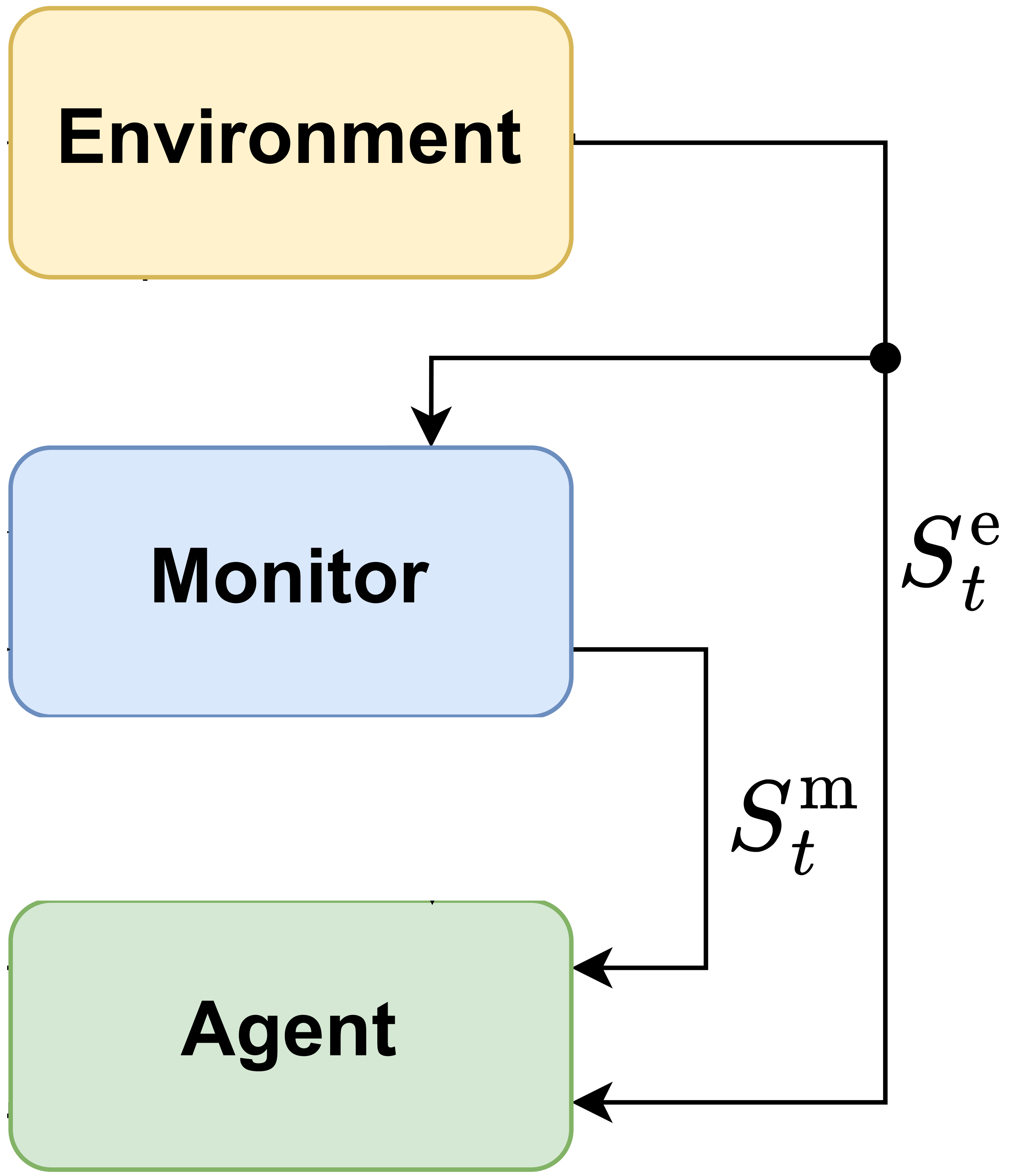
1. S. Parisi, et al. "Monitored Markov Decision Processes," (AAMAS '24)

Monitored Markov decision processes (Mon-MDPs)

An extension of MDPs to cover partial observability of rewards

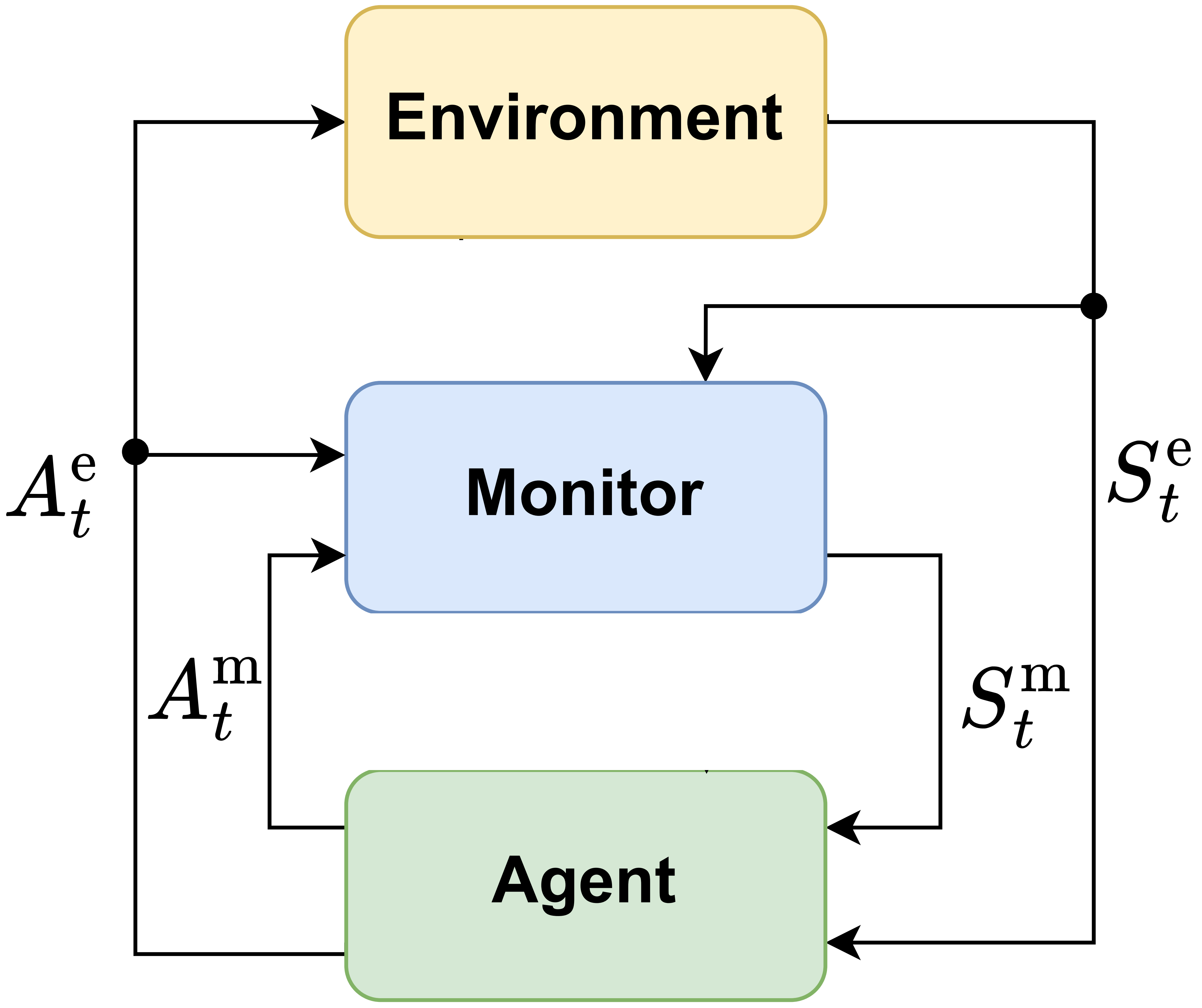
Monitored Markov decision processes (Mon-MDPs)

An extension of MDPs to cover partial observability of rewards



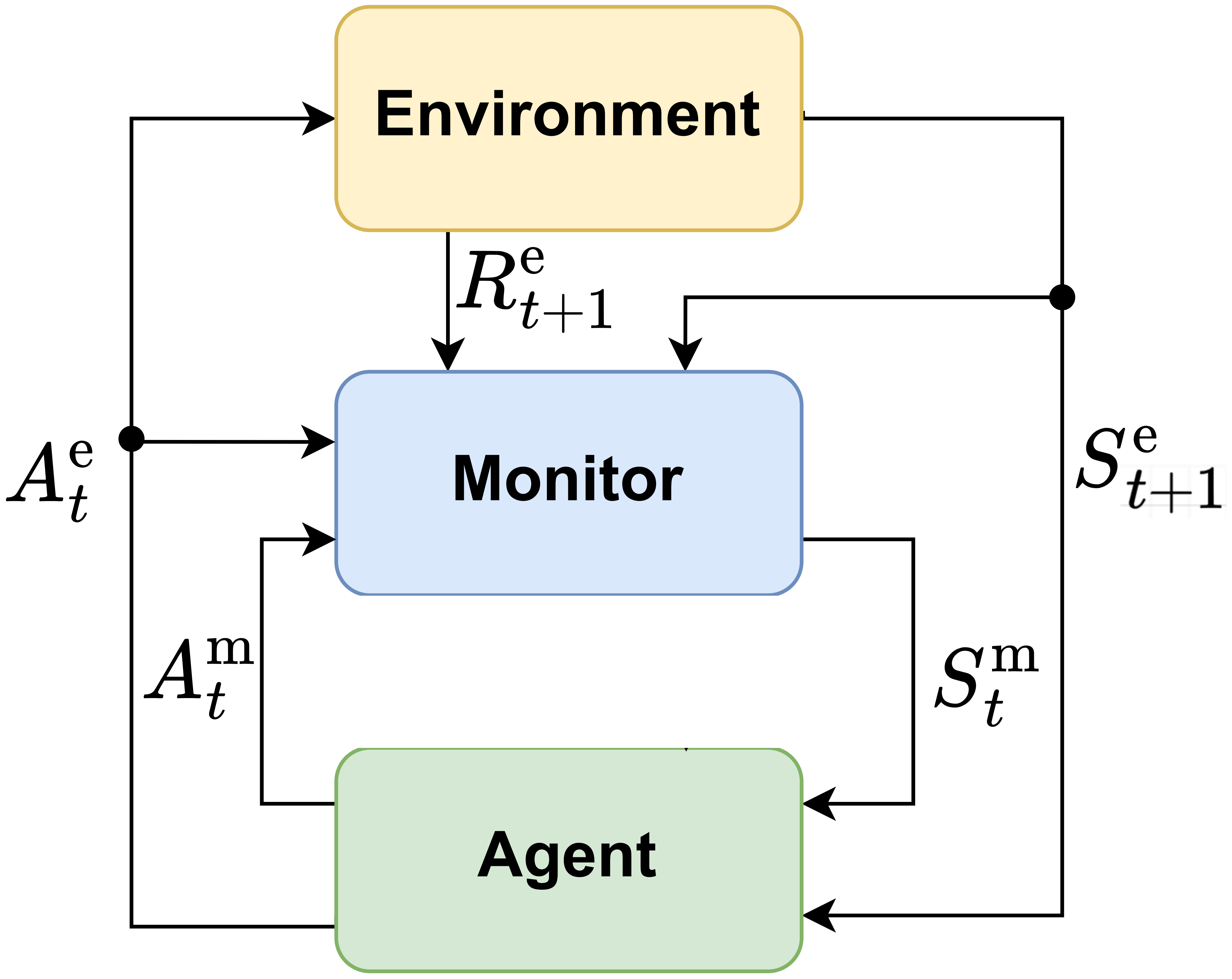
Monitored Markov decision processes (Mon-MDPs)

An extension of MDPs to cover partial observability of rewards



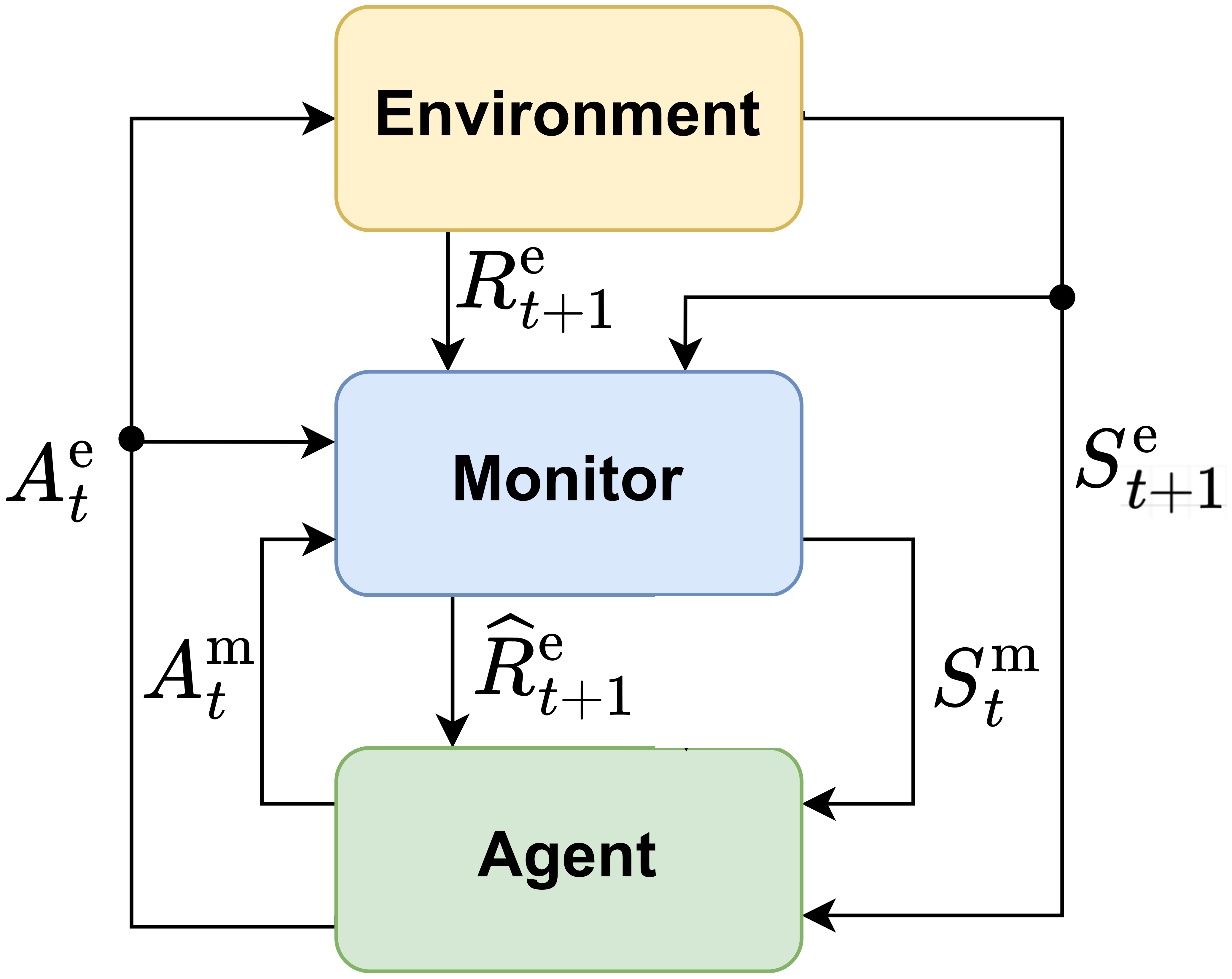
Monitored Markov decision processes (Mon-MDPs)

An extension of MDPs to cover partial observability of rewards



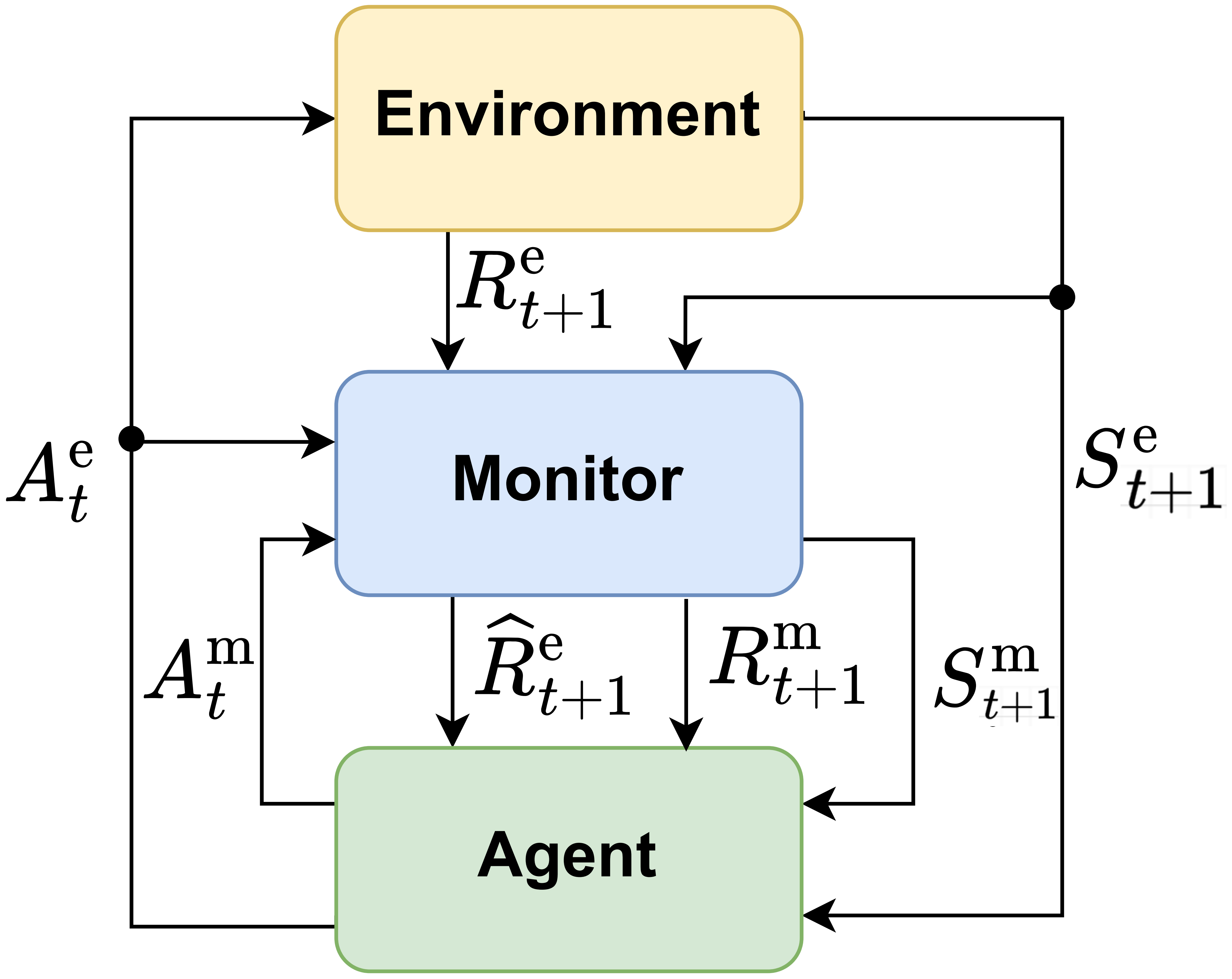
Monitored Markov decision processes (Mon-MDPs)

An extension of MDPs to cover partial observability of rewards



Monitored Markov decision processes (Mon-MDPs)

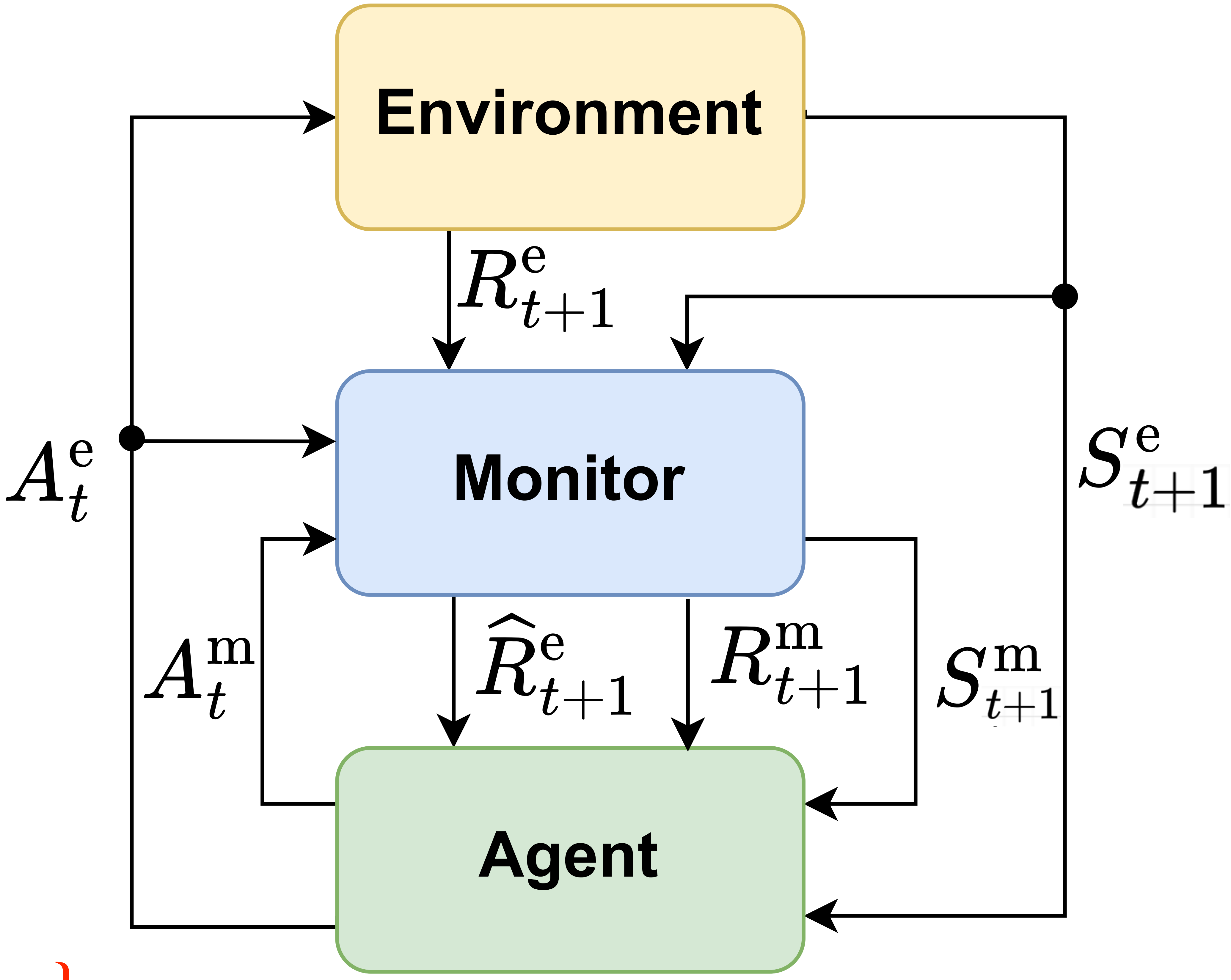
An extension of MDPs to cover partial observability of rewards



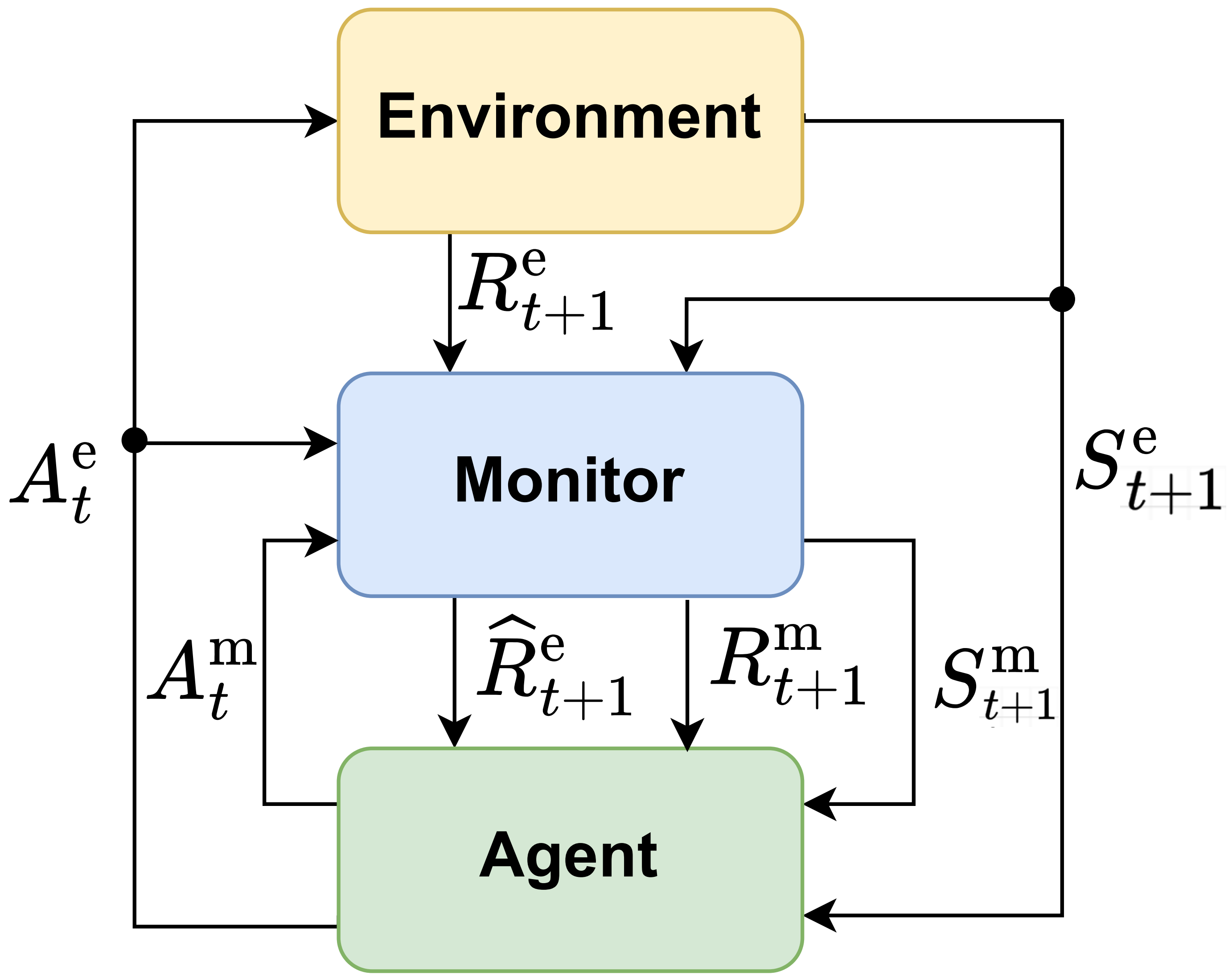
Monitored Markov decision processes (Mon-MDPs)

An extension of MDPs to cover partial observability of rewards

- The goal: $\max_{\pi} \mathbb{E}_{\pi}\left[\sum_{t=0}^{\infty} \gamma^t \left(R^{\mathrm{e}}_{t+1} + R^{\mathrm{m}}_{t+1}\right)\right]$
- A finite Mon-MDP: $\langle \mathcal{S}, \mathcal{A}, r, p, f^{\mathrm{m}}, \gamma \rangle$
- $\mathcal{S} := \mathcal{S}^{\mathrm{e}} \times \mathcal{S}^{\mathrm{m}}, \quad \mathcal{A} := \mathcal{A}^{\mathrm{e}} \times \mathcal{A}^{\mathrm{m}}$
- r is the joint mean reward
- p is the joint transition dynamics
- Monitor function $\widehat{R}^{\mathrm{e}}_{t+1} \sim f^{\mathrm{m}}$, and $\widehat{R}^{\mathrm{e}}_{t+1} \in \mathbb{R} \cup \{ \perp \}$
- $0 \leq \gamma < 1$



Assumption

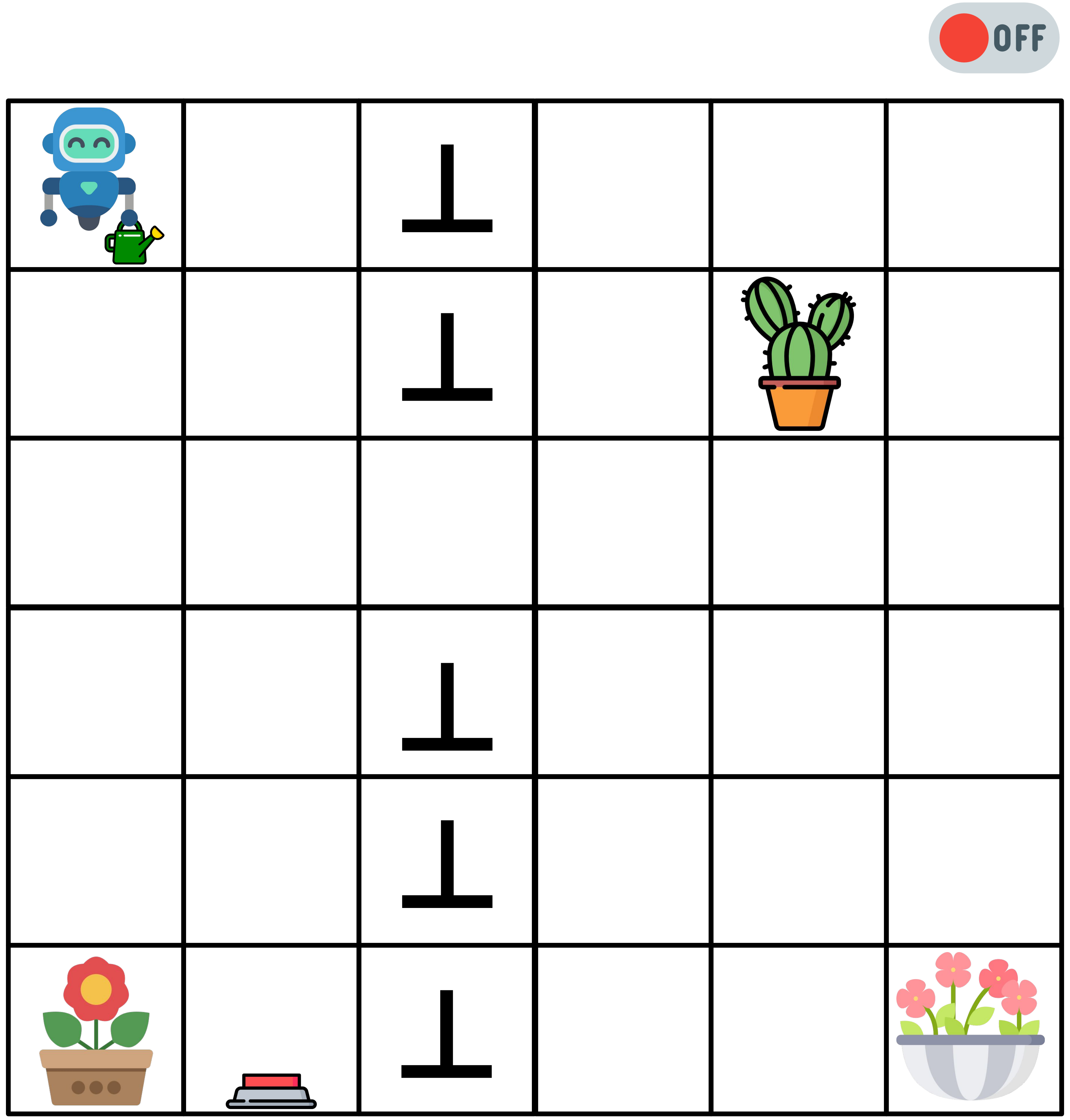


Truthfulness: The monitor doesn't change the underlying reward:

$$\widehat{R}^{\mathrm{e}}_{t+1} \in \{R^{\mathrm{e}}_{t+1}, \perp\}$$

Bottleneck - An example of a Mon-MDP

Suppose the button activates a monitoring system



Bottleneck - An example of a Mon-MDP

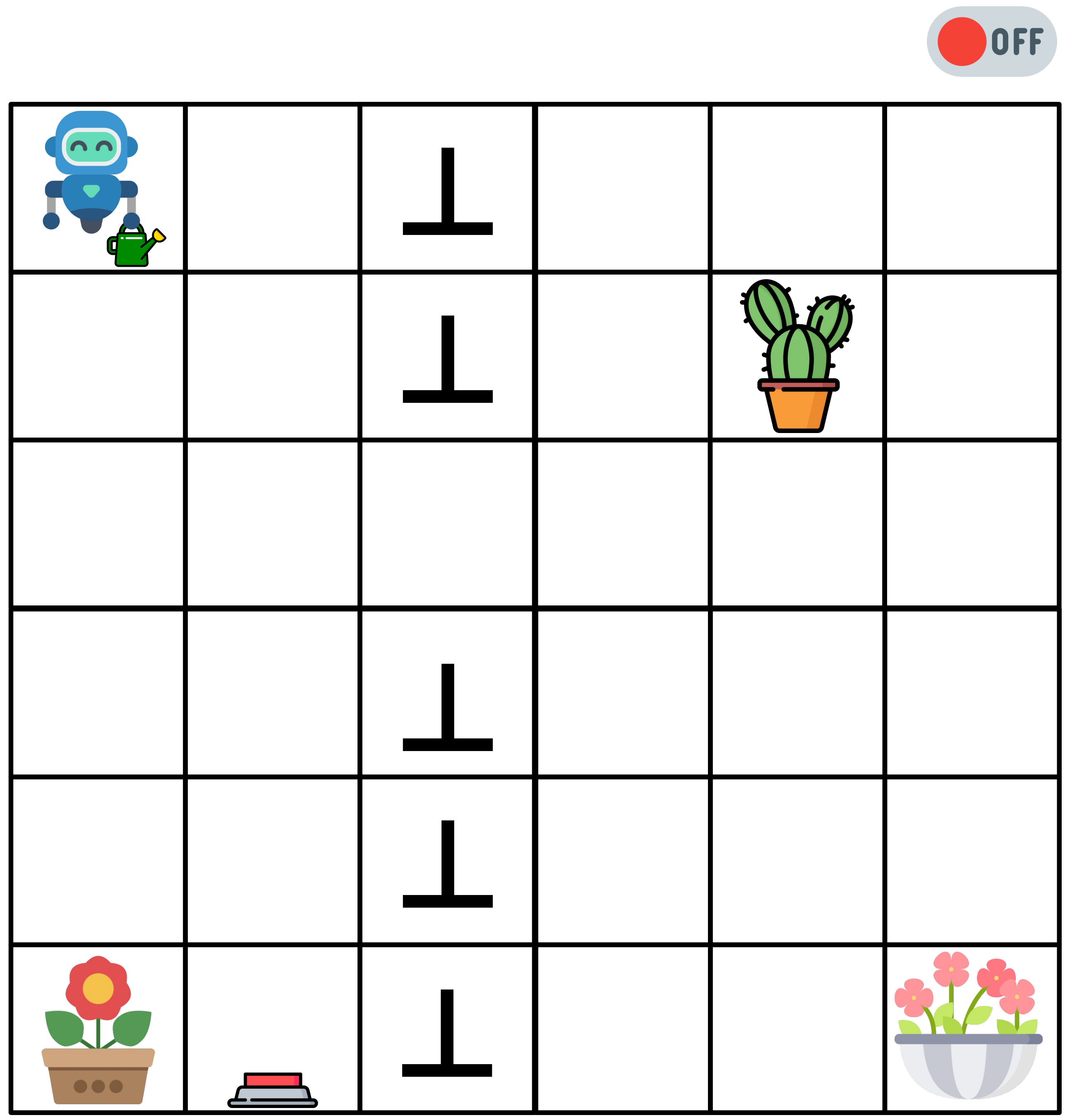
Suppose the button activates a monitoring system

$\mathcal{S}^{\mathrm{m}} := \{\mathrm{OFF}, \ \mathrm{ON}\}, \quad \mathcal{A}^{m} := \{\text{NO-OP}\}$

Let X_t be random uniform and $0 \leq \rho \leq 1$

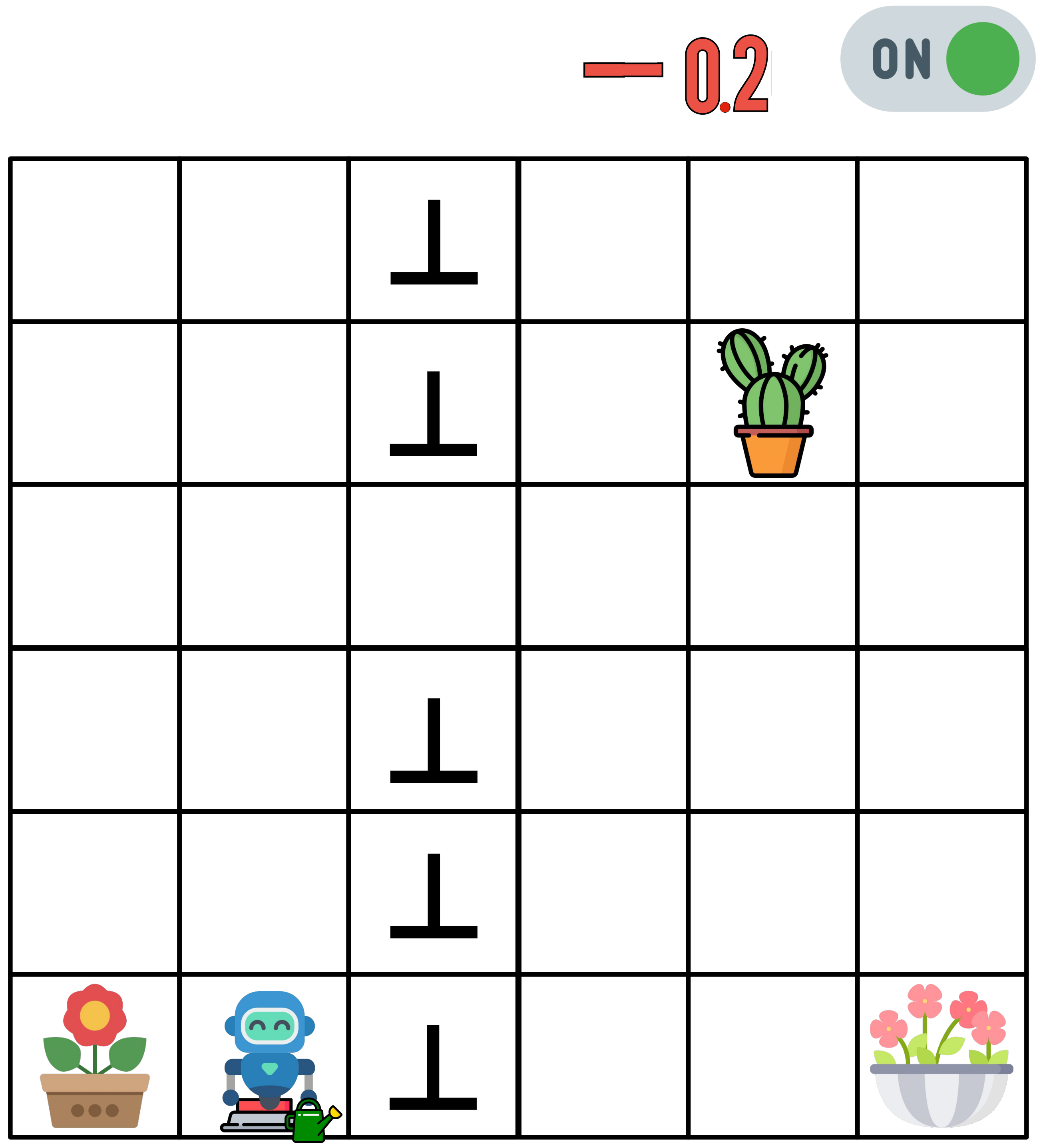
$$\widehat{R}^{\mathrm{e}}_{t+1} := \begin{cases} R^{\mathrm{e}}_{t+1}, & \text{if } X_t \leq \rho \text{ and } S^{\mathrm{m}}_t = \mathrm{ON}; \\ \bot, & \text{Otherwise} \end{cases}$$

$$S^{\mathrm{m}}_{t+1} := \begin{cases} \mathrm{ON}, & \text{if } S^{\mathrm{m}}_t = \mathrm{OFF} \text{ and } S^{\mathrm{e}}_t = \text{"B-CELL" and } A^{\mathrm{e}}_t = \downarrow; \\ \mathrm{OFF}, & \text{if } S^{\mathrm{m}}_t = \mathrm{ON} \text{ and } S^{\mathrm{e}}_t = \text{"B-CELL" and } A^{\mathrm{e}}_t = \downarrow; \\ S^{\mathrm{m}}_t, & \text{Otherwise} \end{cases}$$



Bottleneck - An example of a Mon-MDP

Suppose the button activates a monitoring system



$\mathcal{S}^{\mathrm{m}} := \{\mathrm{OFF},\ \mathrm{ON}\}, \quad \mathcal{A}^{m} := \{\text{NO-OP}\}$

Let X_t be random uniform and $0 \leq \rho \leq 1$

$$\widehat{R}^{\mathrm{e}}_{t+1} := \begin{cases} R^{\mathrm{e}}_{t+1}, & \text{if } X_t \leq \rho \text{ and } S^{\mathrm{m}}_t = \mathrm{ON}; \\ \bot\,, & \text{Otherwise} \end{cases}$$

$$S^{\mathrm{m}}_{t+1} := \begin{cases} \mathrm{ON}, & \text{if } S^{\mathrm{m}}_t = \mathrm{OFF} \text{ and } S^{\mathrm{e}}_t = \text{"B-CELL" and } A^{\mathrm{e}}_t = \downarrow\,; \\ \mathrm{OFF}, & \text{if } S^{\mathrm{m}}_t = \mathrm{ON} \text{ and } S^{\mathrm{e}}_t = \text{"B-CELL" and } A^{\mathrm{e}}_t = \downarrow\,; \\ S^{\mathrm{m}}_t, & \text{Otherwise} \end{cases}$$

$$R^{\mathrm{m}}_{t+1} := \begin{cases} -0.2, & \text{if } S^{\mathrm{m}}_t = \mathrm{ON}; \\ 0, & \text{Otherwise} \end{cases}$$

Solution

Our research questions

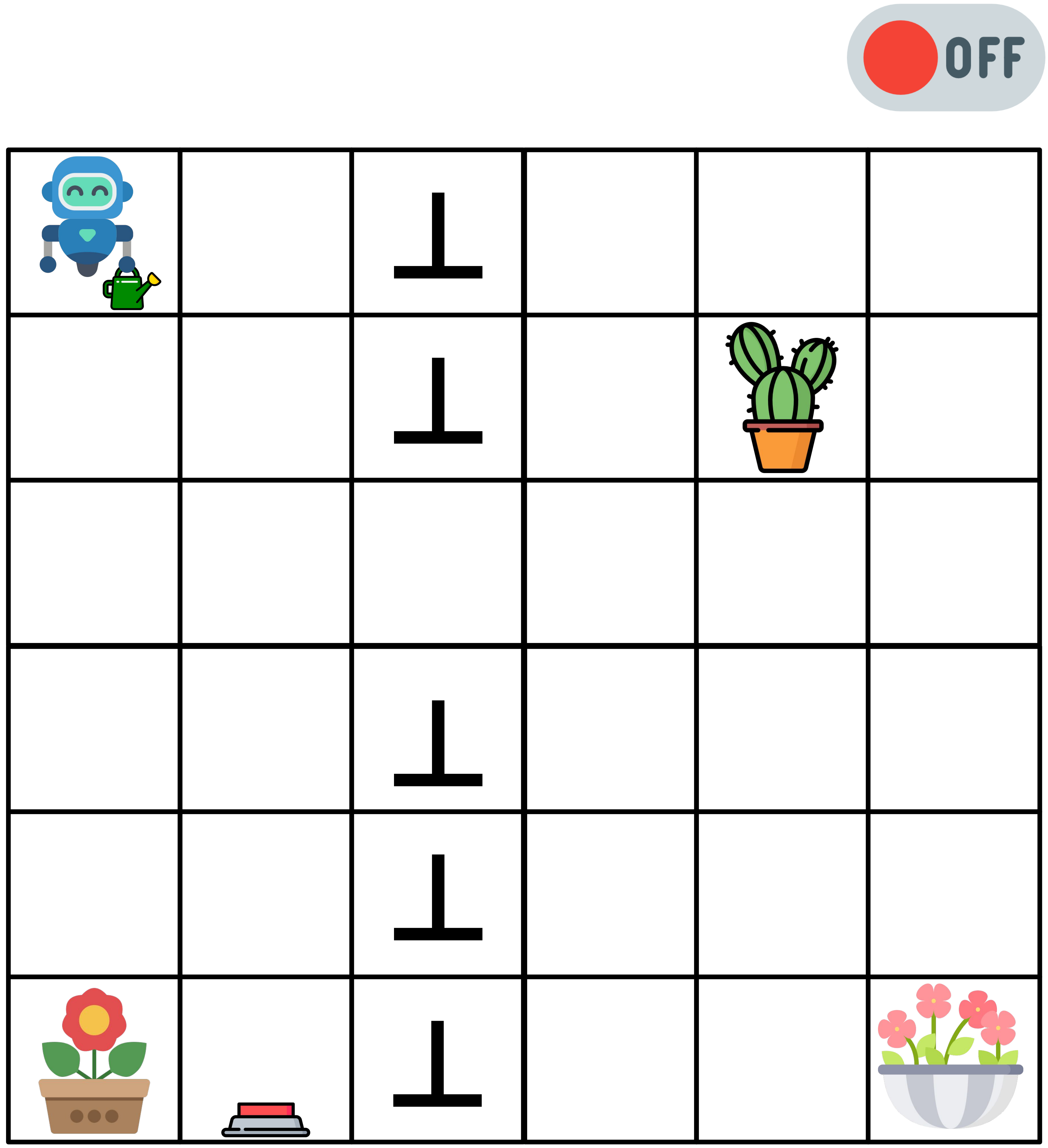
Review

1. How to detect ⊥ cells from all the others?

2. How to deal with ⊥ cells?

3. Can the agent be efficient in watering 

   while not impacting (1) and (2)?

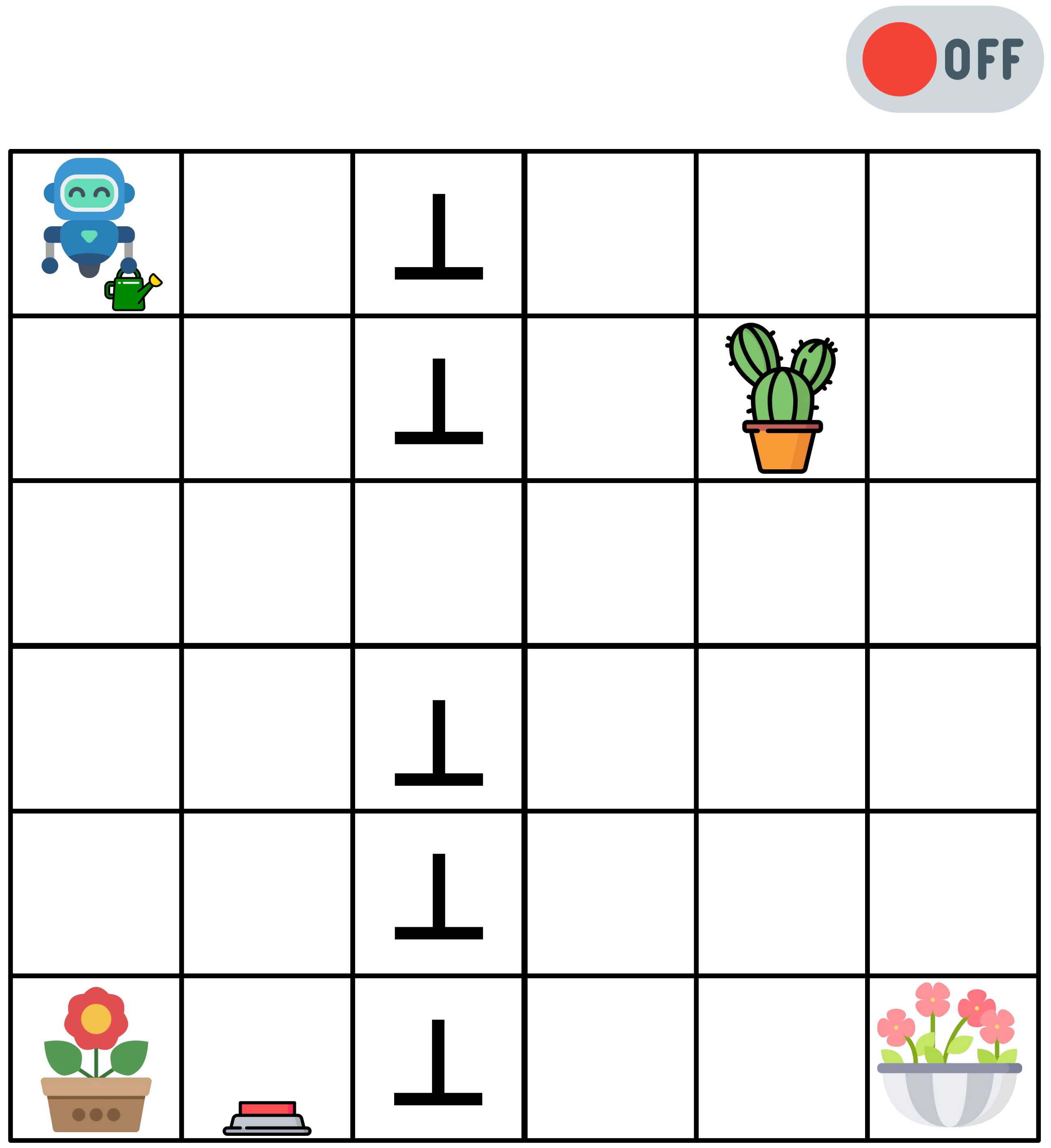


1- How to detect true ⊥ cells?

1- How to detect true ⊥ cells?

Explore to observe rewards

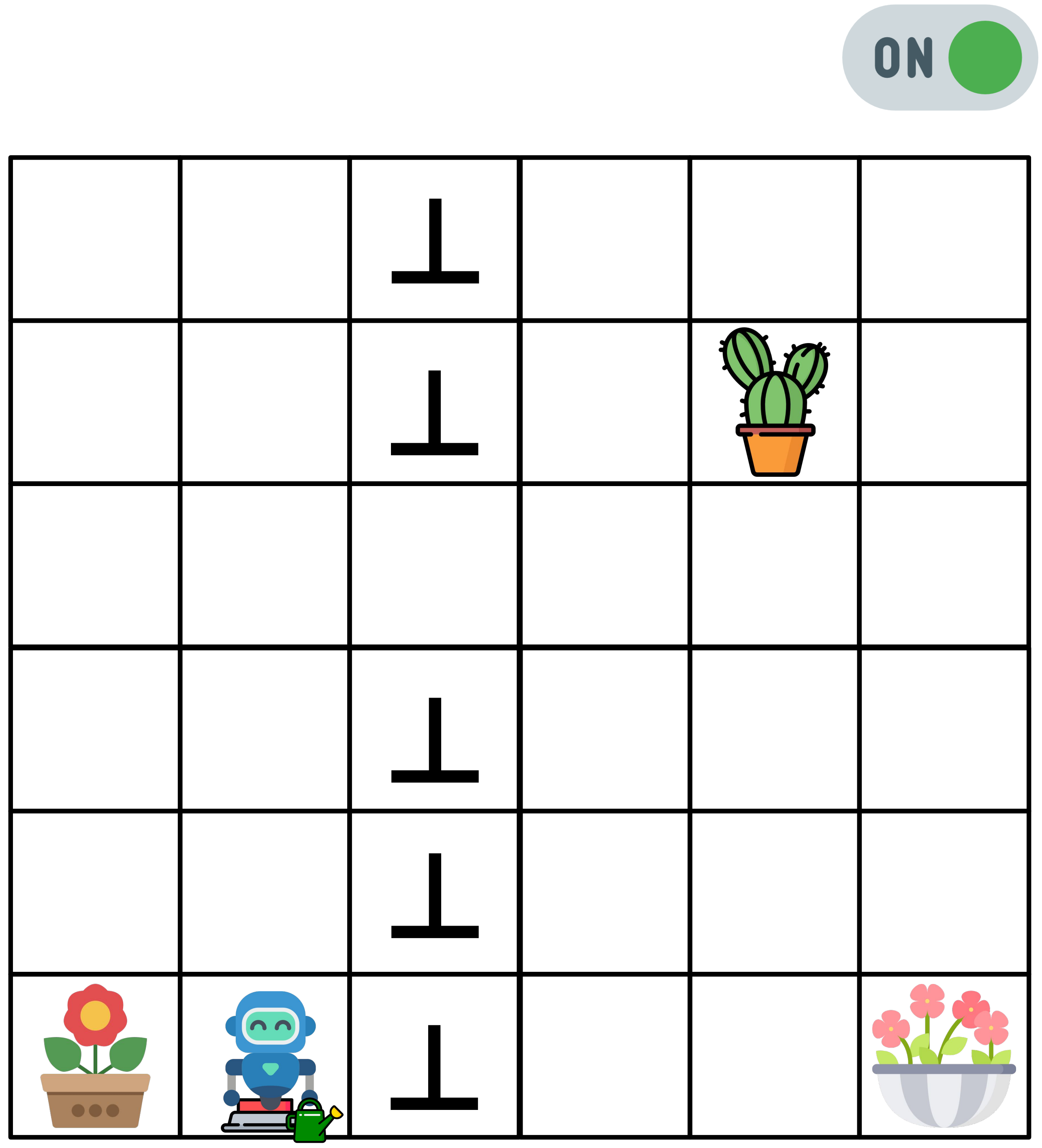
$$\widetilde{R}_{t+1} = \begin{cases} 1 & \text{if the action led to observing the reward in a state that the reward hasn't been observed before} \\ 0 & \text{otherwise} \end{cases}$$



1- How to detect true ⊥ cells?

Explore to observe rewards

$$\widetilde{R}_{t+1} = \begin{cases} 1 & \text{if the action led to observing the reward in a state that the reward hasn't been observed before} \\ 0 & \text{otherwise} \end{cases}$$

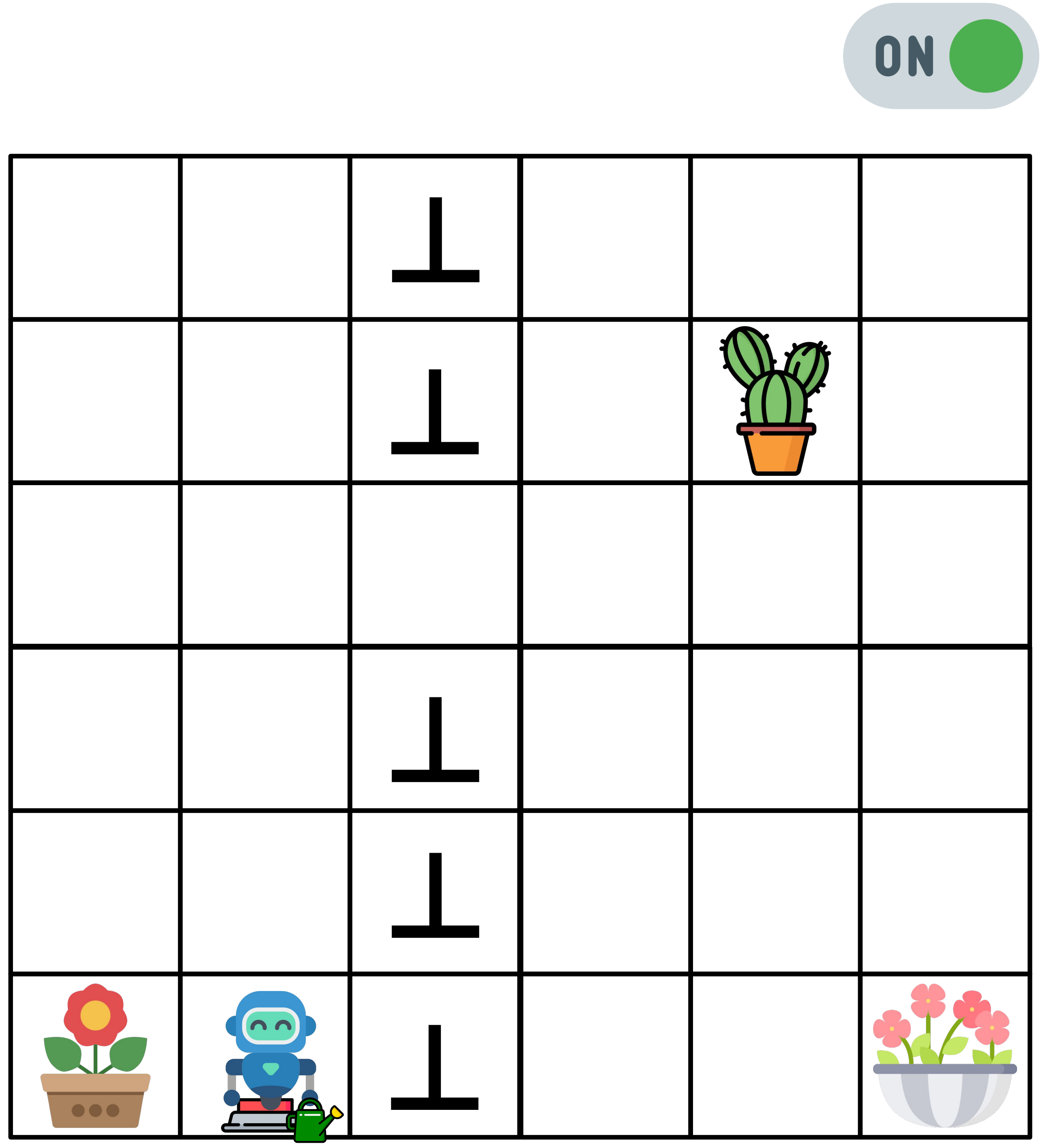


1- How to detect true ⊥ cells?

Explore to observe rewards

$$\widetilde{R}_{t+1} = \begin{cases} 1 & \text{if the action led to observing the reward in a state that the reward hasn't been observed before} \\ 0 & \text{otherwise} \end{cases}$$

$\widetilde{R}_{t+1}$ is Bernoulli.



Measuring uncertainty

Review

- Suppose you have n samples. Then

$$\texttt{distance}\left(\text{empirical mean, true mean}\right) \leq \frac{\beta}{\sqrt{n}}$$

- If you particularly have n Bernoulli samples. Then

$$\texttt{distance}\left(\text{empirical mean, true mean}\right) \leq \frac{\beta}{n}$$

for sufficiently large value of β

$\frac{\beta}{\sqrt{n}}$, and $\frac{\beta}{n}$ measure the **uncertainty**.

1- How to detect true ⊥ cells?

Explore to observe rewards

$$\widetilde{R}_{t+1} = \begin{cases} 1 & \text{if the action led to observing the reward in a state that the reward hasn't been observed before} \\ 0 & \text{otherwise} \end{cases}$$

$\widetilde{R}_{t+1}$ is Bernoulli.

1- How to detect true ⊥ cells?

Explore to observe rewards

$$\widetilde{R}_{t+1} = \begin{cases} 1 & \text{if the action led to observing the reward in a state that the reward hasn't been observed before} \\ 0 & \text{otherwise} \end{cases}$$

$\widetilde{R}_{t+1}$ is Bernoulli.

Be greedy w.r.t $\widetilde{Q}(s,a) = \underbrace{g\left(\frac{\beta_1}{n}\right)}_{\text{uncertainty for } \tilde{r}}$

A function

1- How to detect true ⊥ cells?

Explore to observe rewards

$$\widetilde{R}_{t+1} = \begin{cases} 1 & \text{if the action led to observing the reward in a state that the reward hasn't been observed before} \\ 0 & \text{otherwise} \end{cases}$$

$\widetilde{R}_{t+1}$ is Bernoulli.

number visits to (s, a)

$$\text{Be greedy w.r.t } \widetilde{Q}(s,a) = \underbrace{g\left(\frac{\beta_1}{n}\right)}_{\text{uncertainty for } \tilde{r}} + \gamma \sum_{s'} \hat{p}\left(s' \middle| s, a\right) \widetilde{V}(s') + \underbrace{\frac{\beta_2}{\sqrt{n}}}_{\text{uncertainty for } \hat{p}}$$

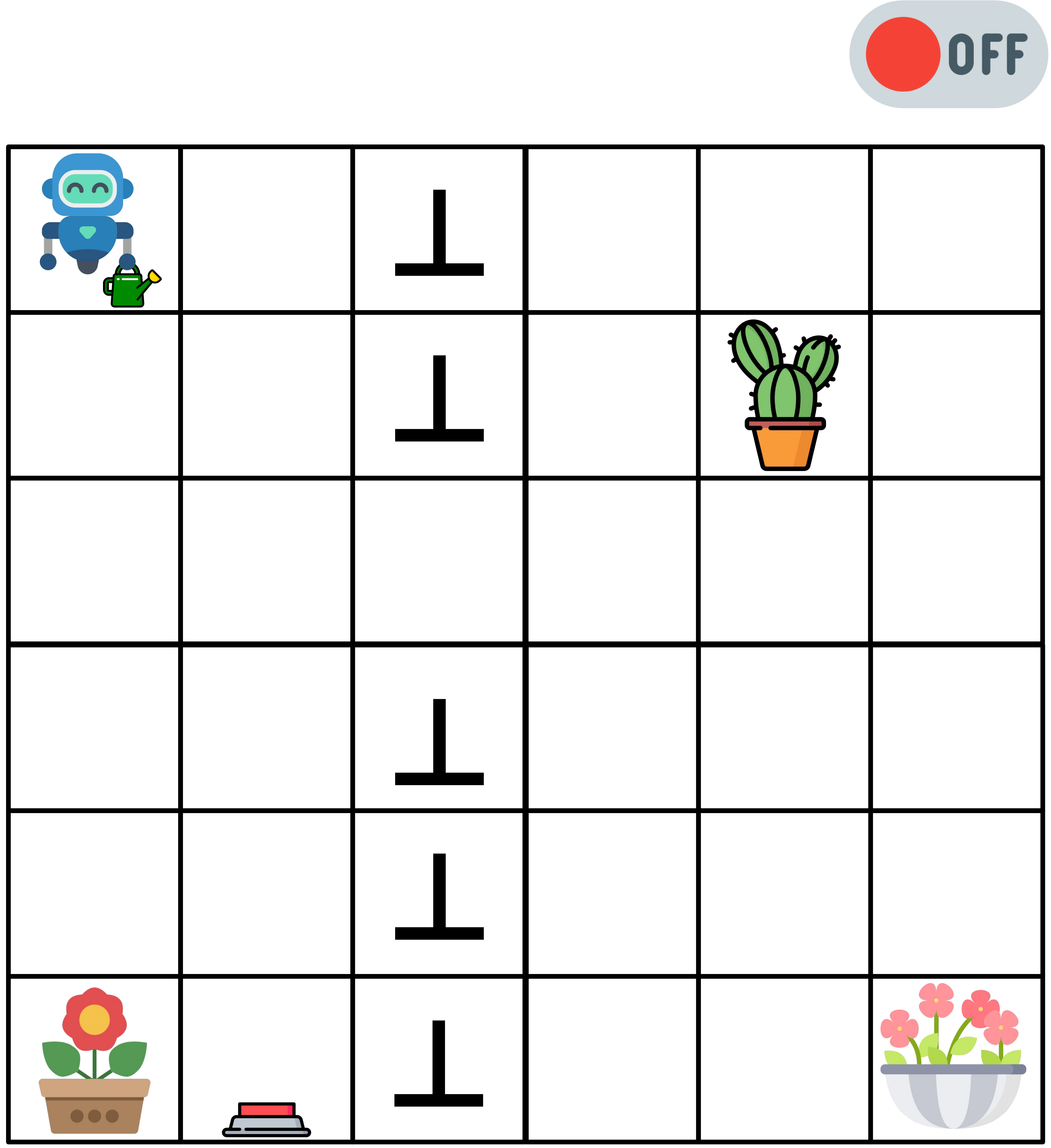
A function

Our research questions

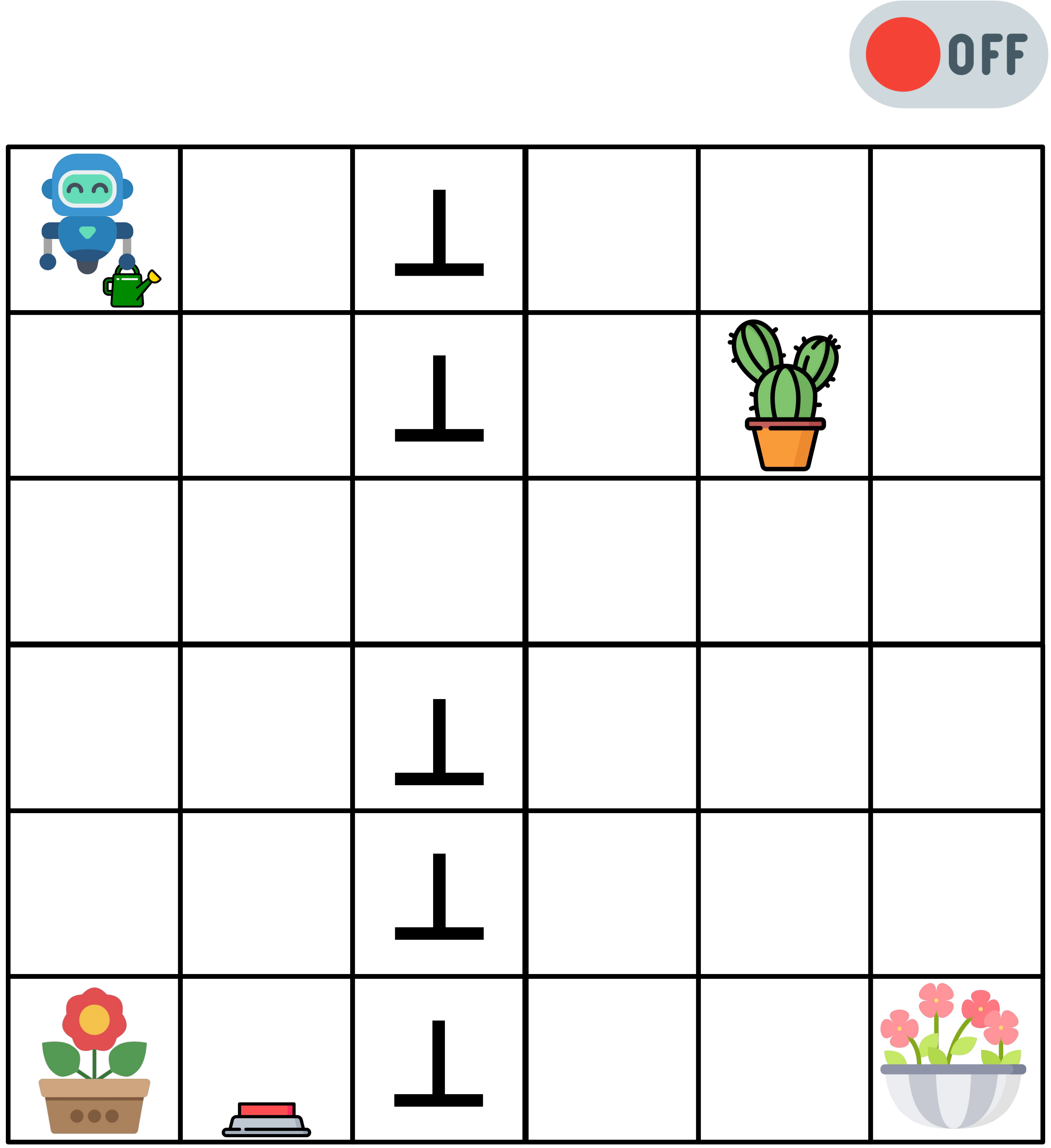
Review

1. How to detect ⊥ cells from all the others? ✅
2. How to deal with ⊥ cells?
3. Can the agent be efficient in watering

   while not impacting (1) and (2)?

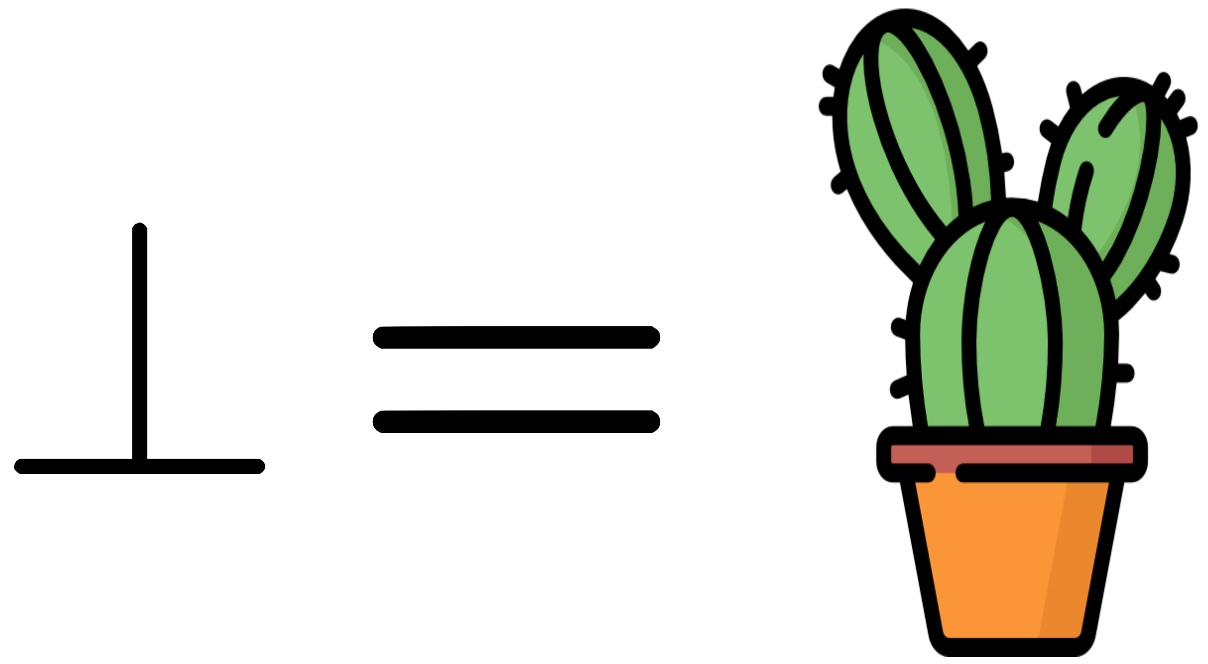


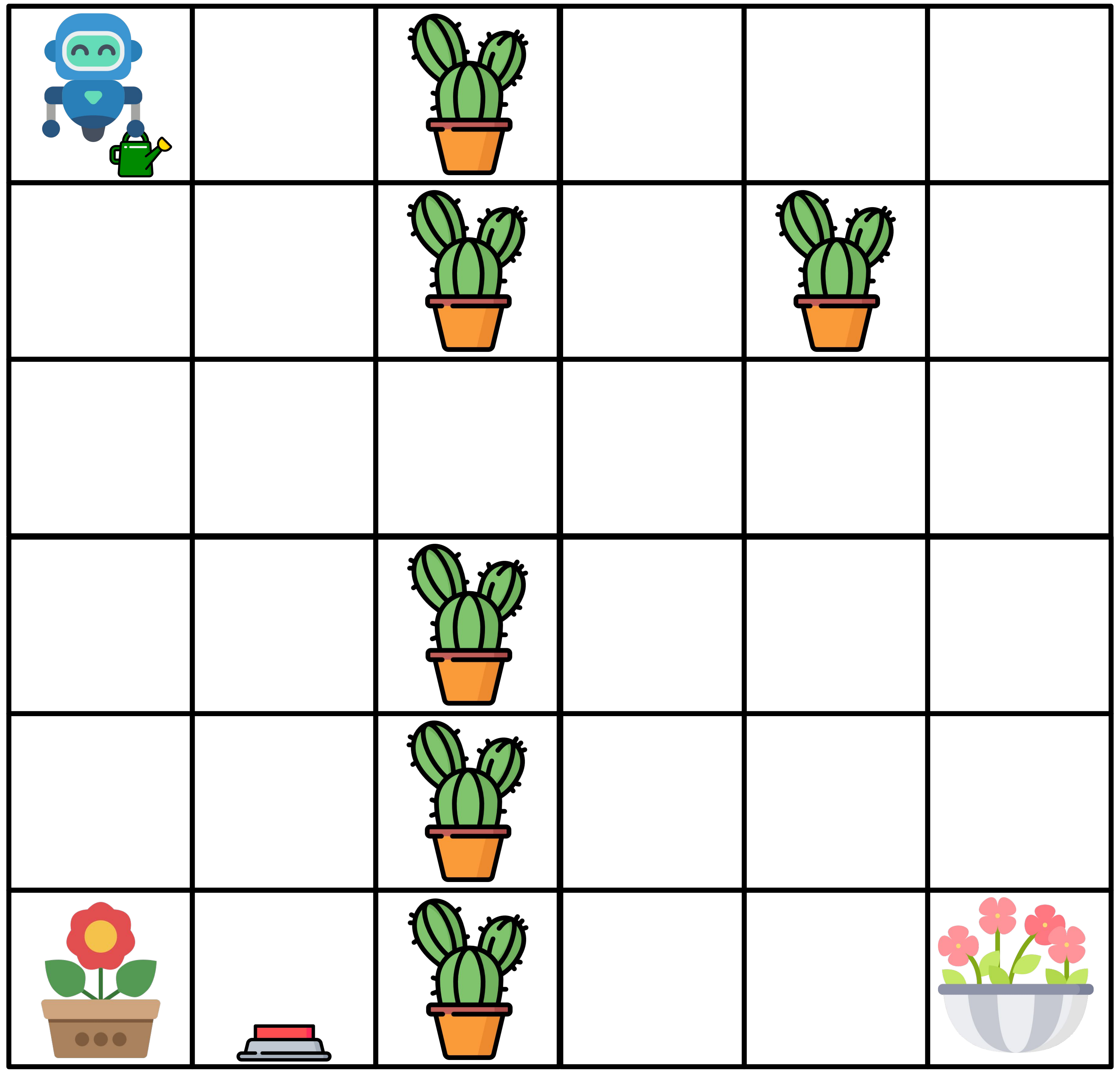
2- How to deal with ⊥ cells?



2- How to deal with ⊥ cells?

Be pessimistic about them





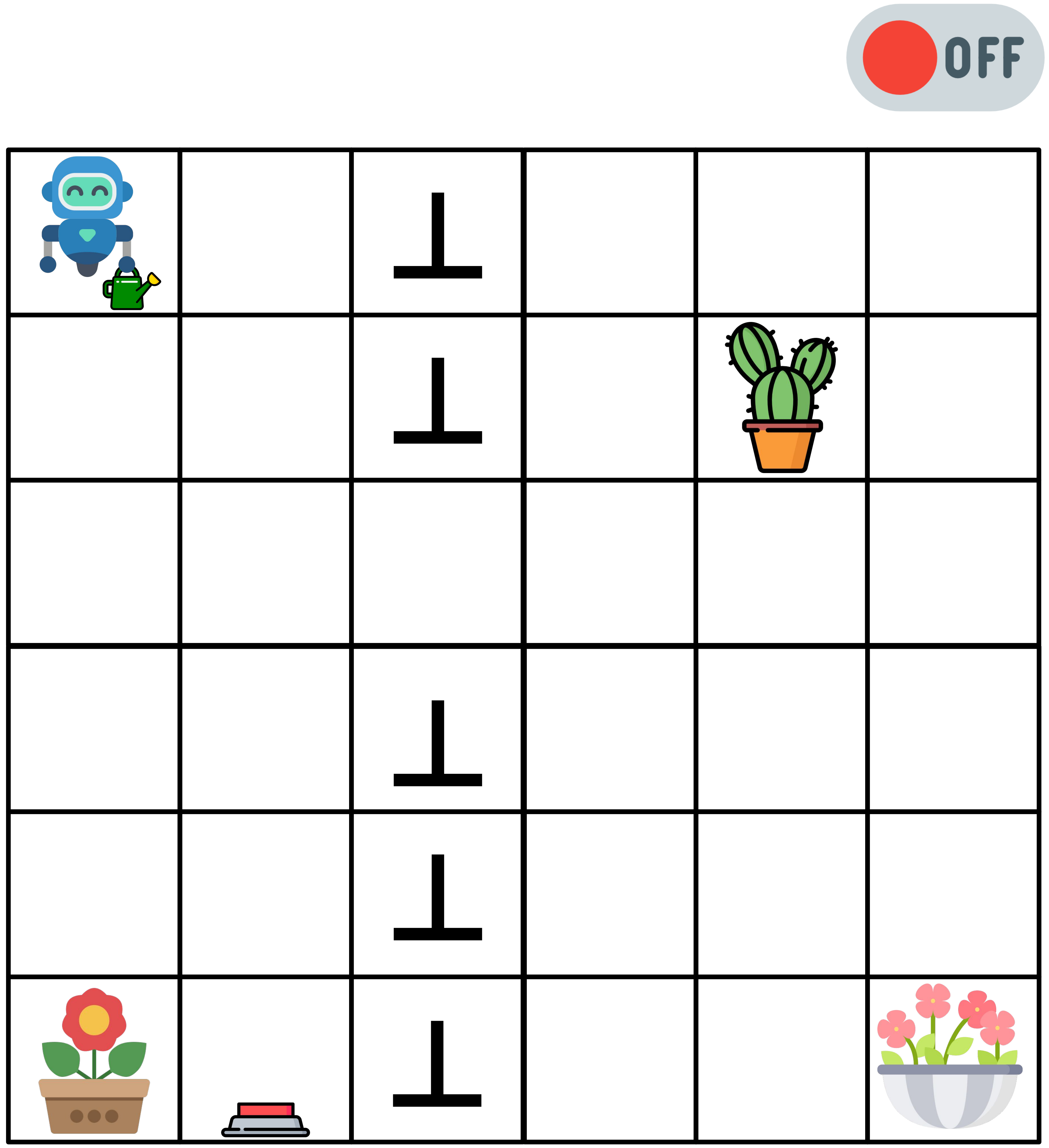
Our research questions

Review

1. How to detect ⊥ cells from all the others? ✅

2. How to deal with ⊥ cells? ✅

3. Can the agent be efficient in watering  while not impacting (1) and (2)?



3- Can the agent be efficient?

3- Can the agent be efficient?

Use MBIE-EB

$\frac{\beta}{\sqrt{n}}$, and $\frac{\beta}{n}$ measure the **uncertainty.**

3- Can the agent be efficient?

Use MBIE-EB

$$\hat{Q}(s, a) = \hat{r}^{\mathrm{e}}\left(s^{\mathrm{e}}, a^{\mathrm{e}}\right) + \underbrace{\frac{\beta_1}{\sqrt{n_1}}}_{\text{uncertainty of } r^{\mathrm{e}}}$$

number of times the env reward is observed

3- Can the agent be efficient?

Use MBIE-EB

$$\hat{Q}(s,a) = \hat{r}^{\mathrm{e}}\left(s^{\mathrm{e}}, a^{\mathrm{e}}\right) + \underbrace{\frac{\beta_1}{\sqrt{n_1}}}_{\text{uncertainty of } r^{\mathrm{e}}} + \hat{r}^{\mathrm{m}}\left(s^{\mathrm{m}}, a^{\mathrm{m}}\right) + \underbrace{\frac{\beta_2}{\sqrt{n_2}}}_{\text{uncertainty of } r^{\mathrm{m}}}$$

number of times the env reward is observed

number of times the mon reward is observed

3- Can the agent be efficient?

Use MBIE-EB

$$\hat{Q}(s,a) = \hat{r}^{\mathrm{e}}\left(s^{\mathrm{e}},a^{\mathrm{e}}\right) + \underbrace{\frac{\beta_1}{\sqrt{n_1}}}_{\text{uncertainty of } r^{\mathrm{e}}} + \hat{r}^{\mathrm{m}}\left(s^{\mathrm{m}},a^{\mathrm{m}}\right) + \underbrace{\frac{\beta_2}{\sqrt{n_2}}}_{\text{uncertainty of } r^{\mathrm{m}}} + \gamma\sum_{s'}\hat{p}\left(s' \middle| s,a\right)\hat{V}(s') + \underbrace{\frac{\beta_3}{\sqrt{n_3}}}_{\text{uncertainty of } p}$$

- n_1: number of times the env reward is observed
- n_2: number of times the mon reward is observed
- n_3: number of visits to (s,a)

3- Can the agent be efficient?

Use MBIE-EB

$$\hat{Q}(s,a) = \hat{r}^{\mathrm{e}}\left(s^{\mathrm{e}}, a^{\mathrm{e}}\right) + \underbrace{\frac{\beta_1}{\sqrt{n_1}}}_{\text{uncertainty of } r^{\mathrm{e}}} + \hat{r}^{\mathrm{m}}\left(s^{\mathrm{m}}, a^{\mathrm{m}}\right) + \underbrace{\frac{\beta_2}{\sqrt{n_2}}}_{\text{uncertainty of } r^{\mathrm{m}}} + \gamma \sum_{s'} \hat{p}\left(s' \middle| s, a\right) \hat{V}(s') + \underbrace{\frac{\beta_3}{\sqrt{n_3}}}_{\text{uncertainty of } p}$$

n_1: number of times the env reward is observed

n_2: number of times the mon reward is observed

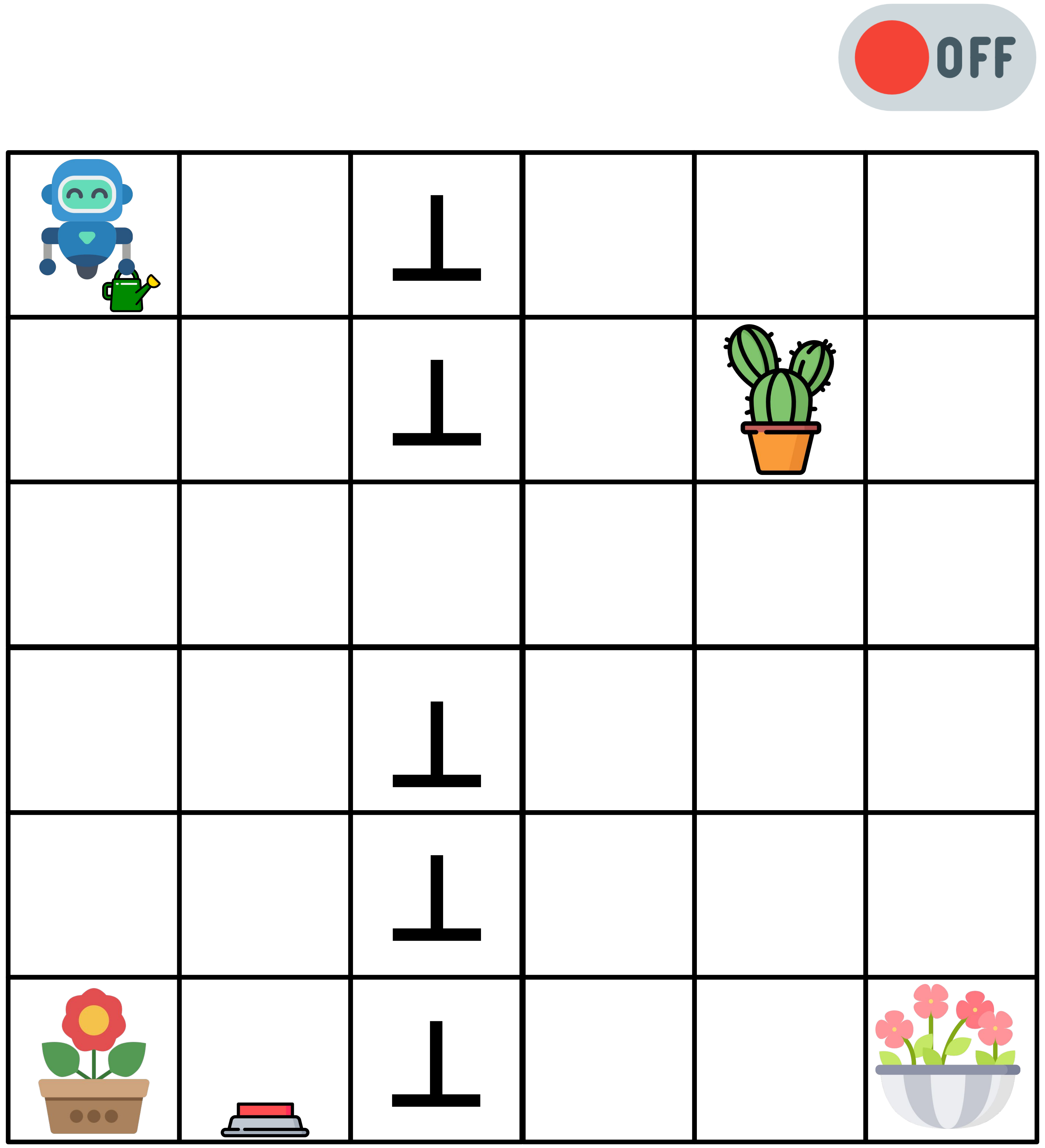
n_3: number of visits to (s, a)

If n_1 was zero (due to unobservability), use  instead.

Our research questions

Review

1. How to detect ⊥ cells from all the others? ✅
2. How to deal with ⊥ cells? ✅
3. Can the agent be efficient in watering 💐 while not impacting (1) and (2)? ✅



Our research questions

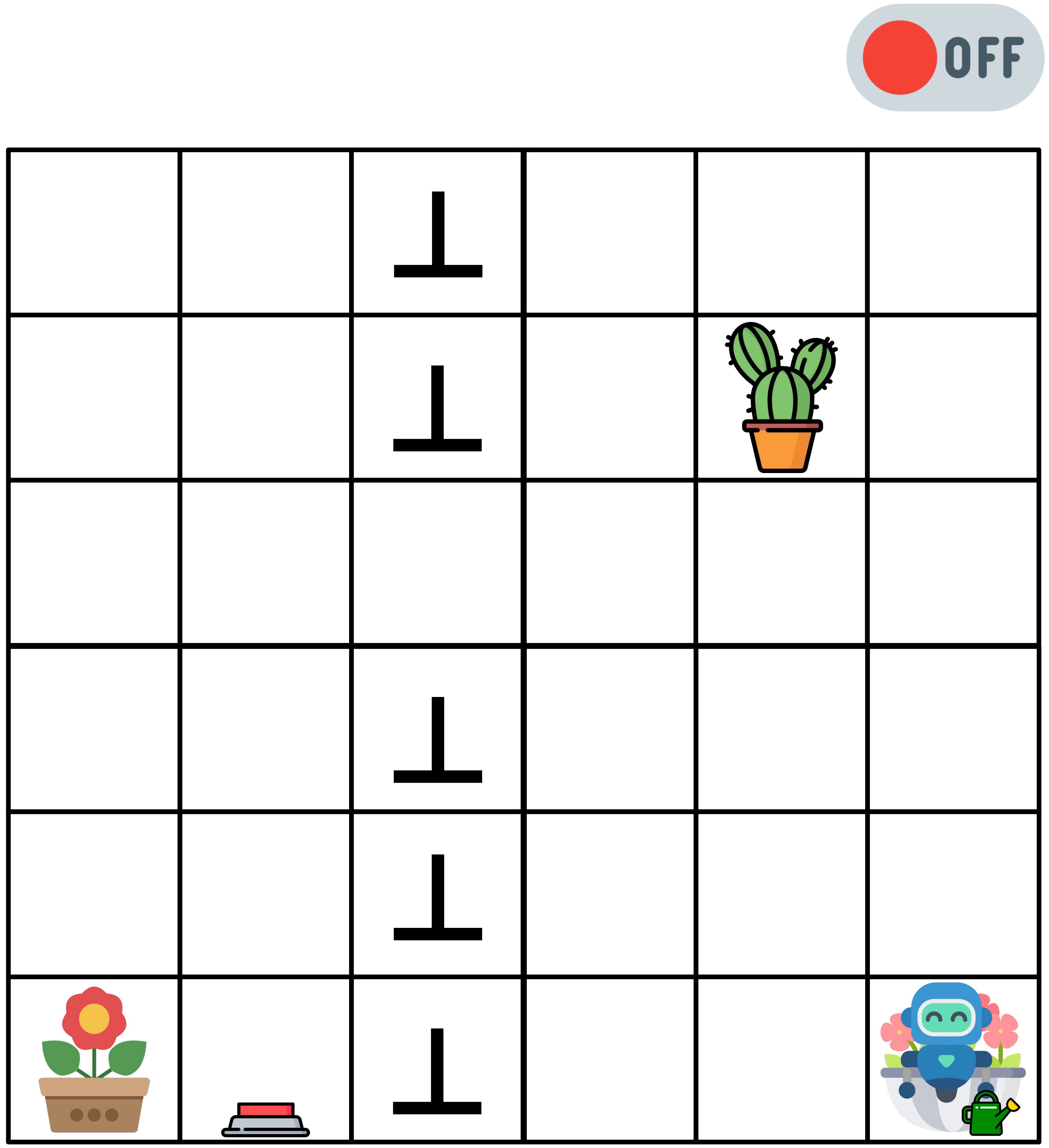
Review

1. How to detect ⊥ cells from all the others? ✅

2. How to deal with ⊥ cells? ✅

3. Can the agent be efficient in watering 🪴

   while not impacting (1) and (2)? ✅



Monitored MBIE-EB's theoretical performance

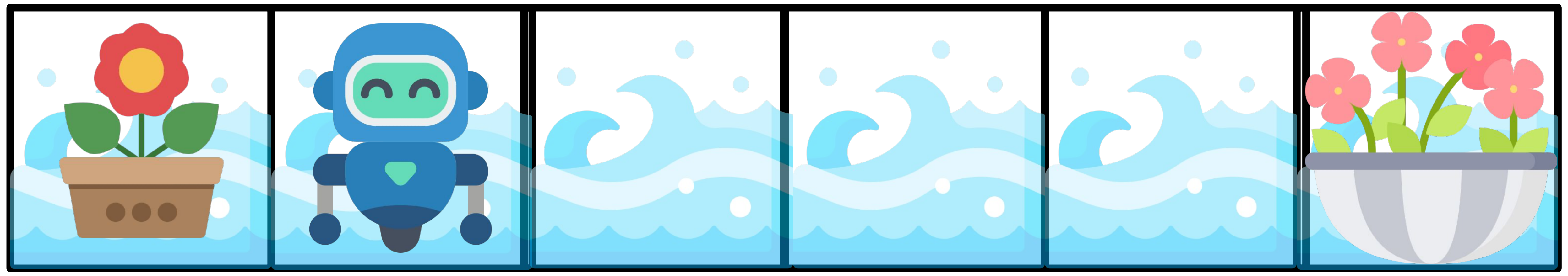
Monitored MBIE-EB's theoretical performance

$$\widetilde{\mathcal{O}}\left(\frac{|\mathcal{S}||\mathcal{A}|}{\rho(1-\gamma)^6\epsilon^3}\right)$$

Monitored MBIE-EB's empirical performance

Monitored MBIE-EB's empirical performance

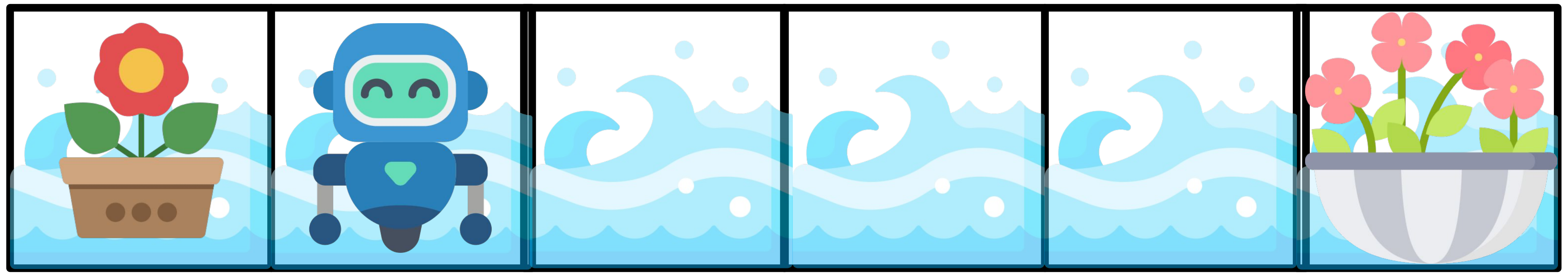
On River Swim

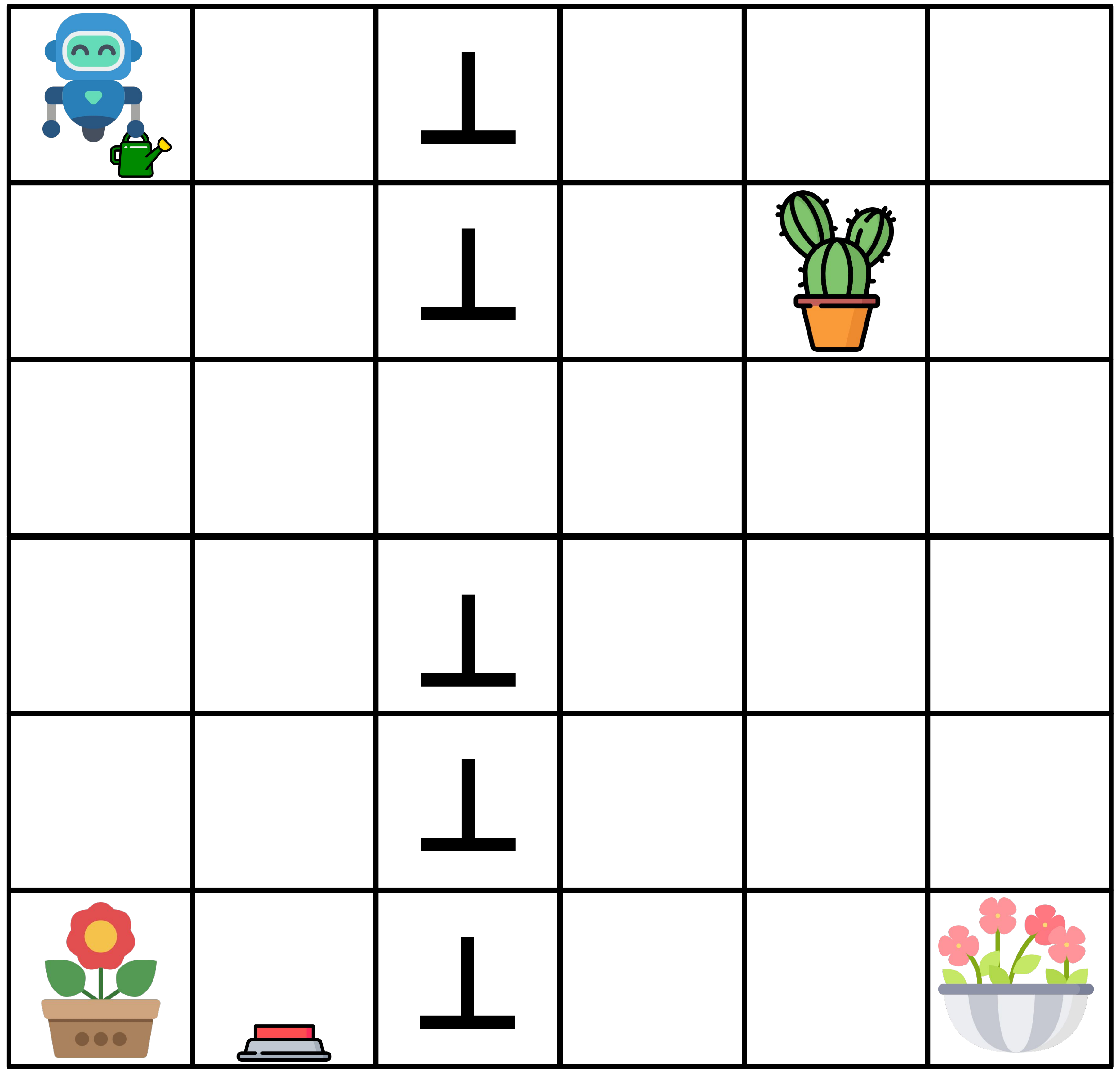


The agent should go to the right but due to stochasticity, it's more likely to move left or stay put. This stochasticity makes the exploration hard.

Monitored MBIE-EB's empirical performance

On River Swim & Bottleneck

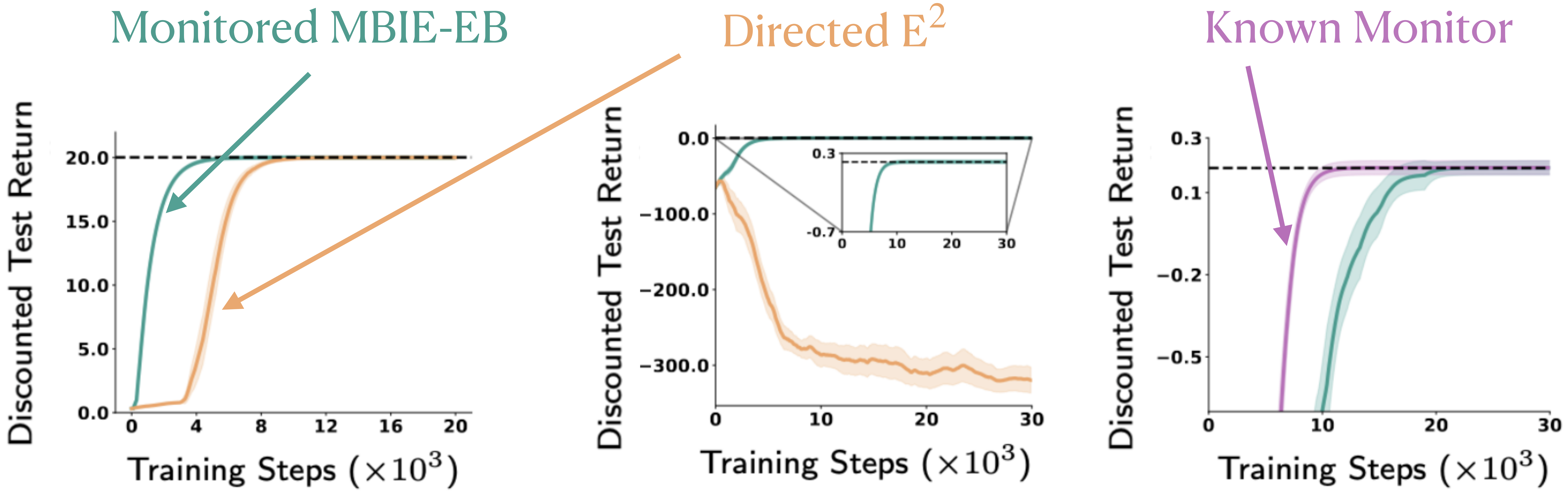




The agent should go to the right but due to stochastic, it's more likely to move left or stay put. This stochasticity makes the exploration hard.

Monitored MBIE-EB's empirical performance

Directed Explore-Exploit (Directed E²) is the state-of-the-art algorithm in Mon-MDPs



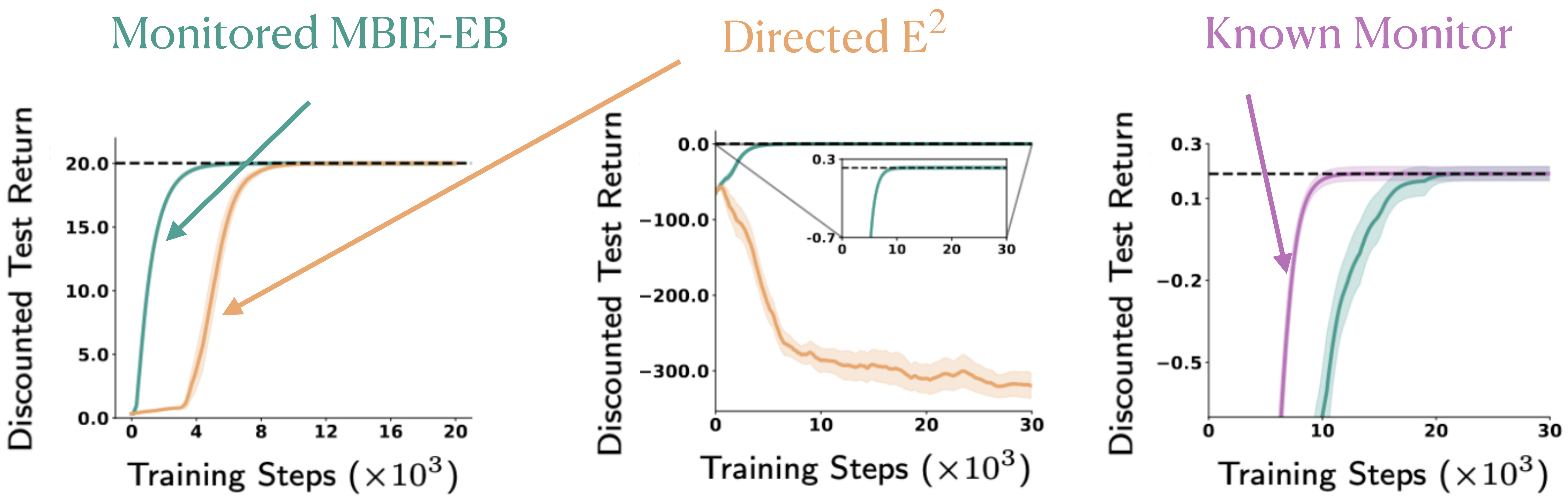
River Swim

Bottleneck w/ $\rho = 5\,\%$

Bottleneck w/ $\rho = 5\,\%$

- Dashed horizontal line is the minimax-optimal discounted return.

Takeaways



River Swim

Bottleneck w/ $\rho = 5\,\%$

Bottleneck w/ $\rho = 5\,\%$

1. Due to being model-based and planning, Monitored MBIE-EB performs well on River Swim.
2. Monitored MBIE-EB is robust against stochastic observability and finds the minimax-optimal policy.
3. Monitored MBIE-EB can leverage prior knowledge about the monitor.

List of Contributions

List of my contributions

- Defining the minimax-optimality in Mon-MDPs replacing the notion of MDPs' optimality.
- Presenting Monitored MBIE-EB, the first model-based minimax-optimal algorithm for Mon-MDPs.
- Proving the polynomial sample complexity of Monitored MBIE-EB.
- Showing the dependence of the Monitored MBIE-EB's sample complexity on ρ in Mon-MDPs is essentially unimprovable.
- Demonstrating the superior performance of Monitored MBIE-EB compared to Directed E^2, the previous state-of-the-art algorithm in Mon-MDPs. We showed more dramatic results when the dynamics of how the agent can or cannot observe the reward is known apriori.

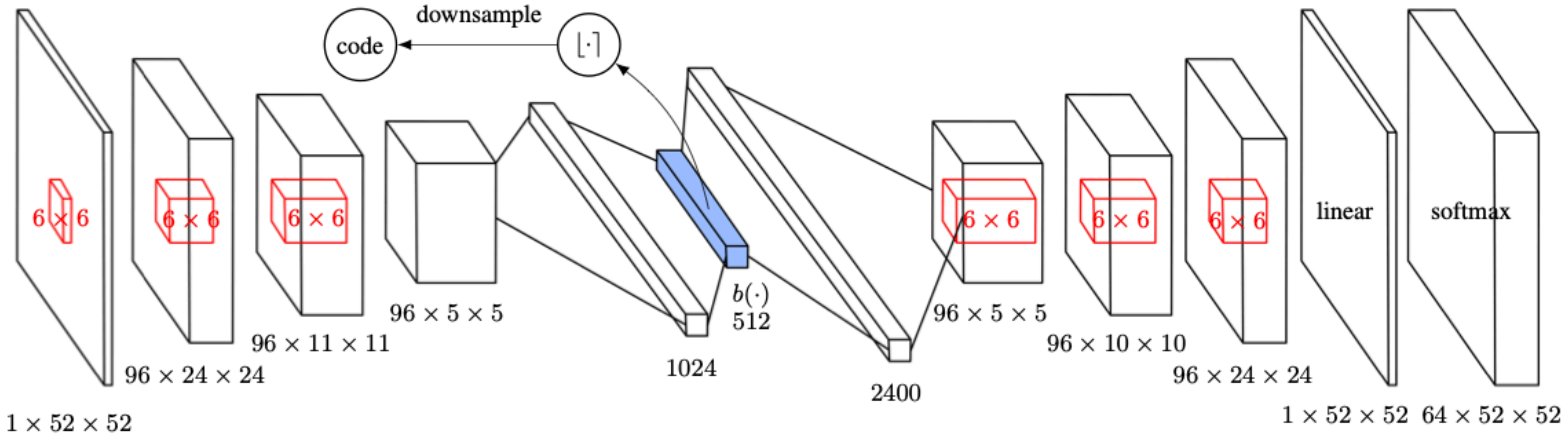
Future work

1- Planning and counts in the latent space

Beyond finite domains

#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning

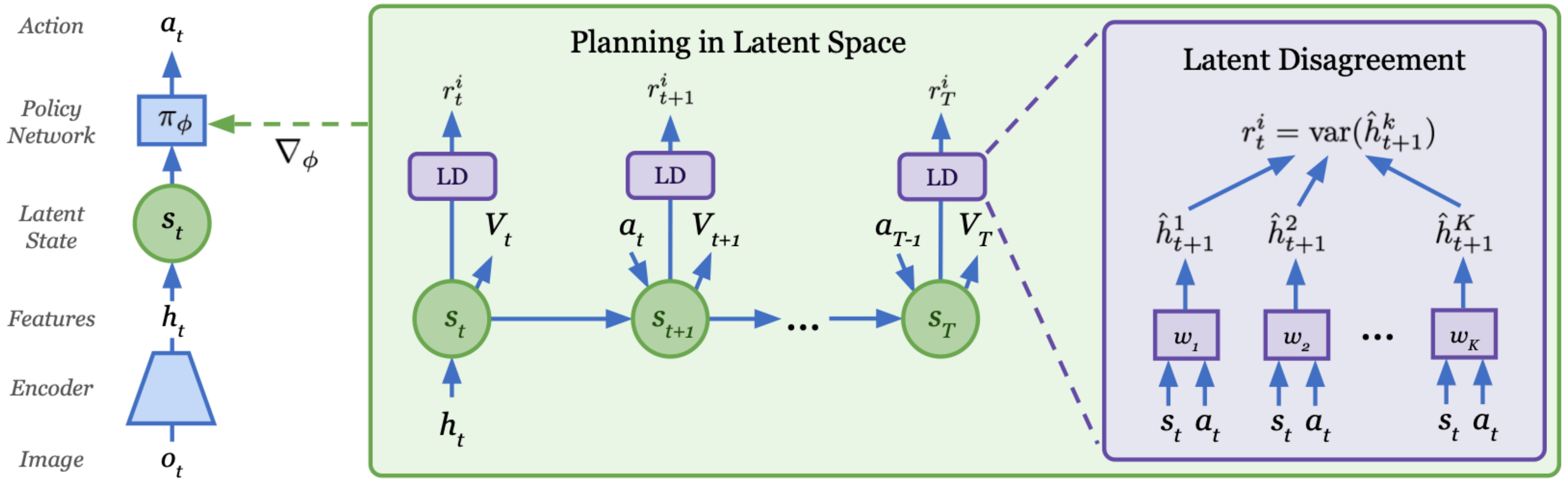
Haoran Tang[1]*, Rein Houthooft[34]*, Davis Foote[2], Adam Stooke[2], Xi Chen[2]†, Yan Duan[2]†, John Schulman[4], Filip De Turck[3], Pieter Abbeel [2]†
[1] UC Berkeley, Department of Mathematics
[2] UC Berkeley, Department of Electrical Engineering and Computer Sciences
[3] Ghent University – imec, Department of Information Technology
[4] OpenAI



(2016)

Planning to Explore via Self-Supervised World Models

Ramanan Sekar [1]* Oleh Rybkin [1]* Kostas Daniilidis [1] Pieter Abbeel [2] Danijar Hafner [34] Deepak Pathak [56]



(2020)

2- Use a better base algorithm

MBIE-EB's upper bound is loose

Our upper bound

$$\widetilde{\mathcal{O}}\left(\frac{|\mathcal{S}||\mathcal{A}|}{\rho(1-\gamma)^6\epsilon^3}\right)$$

PAC Bounds for Discounted MDPs

Tor Lattimore[1] and **Marcus Hutter**[1,2,3]

Research School of Computer Science
[1]Australian National University and [2]ETH Zürich and [3]NICTA
{tor.lattimore,marcus.hutter}@anu.edu.au

$$\widetilde{\Omega}\left(\frac{|\mathcal{S}||\mathcal{A}|}{(1-\gamma)^3\epsilon^2}\right)$$

(2012)

3- Unifying the observation and optimization

A unified algorithm

Near-Optimal Reinforcement Learning in Polynomial Time

MICHAEL KEARNS* mkearns@cis.upenn.edu
Department of Computer and Information Science, University of Pennsylvania, Moore School Building, 200 South 33rd Street, Philadelphia, PA 19104-6389, USA

SATINDER SINGH* satinder.baveja@syntekcapital.com
Syntek Capital, New York, NY 10019, USA

Explicit Explore or Exploit (E^3), (2002)

R-MAX – **A General Polynomial Time Algorithm for Near-Optimal Reinforcement Learning**

Ronen I. Brafman BRAFMAN@CS.BGU.AC.IL
Computer Science Department
Ben-Gurion University
Beer-Sheva, Israel 84105

Moshe Tennenholtz* MOSHE@ROBOTICS.STANFORD.EDU
Computer Science Department
Stanford University
Stanford, CA 94305

R-Max, (2002)

Acknowledgement

I can't thank enough these individuals

- My lovely supervisors: Matt and Mike
- My collaborators: Simone and Monta
- Mike's and Matt's lab: Too many to name individually
- My teachers and all theoreticians throughout history whose contributions has given and will give me insights to study and (ideally) solve problems
- My family
- God ❤️

Thank you! :)