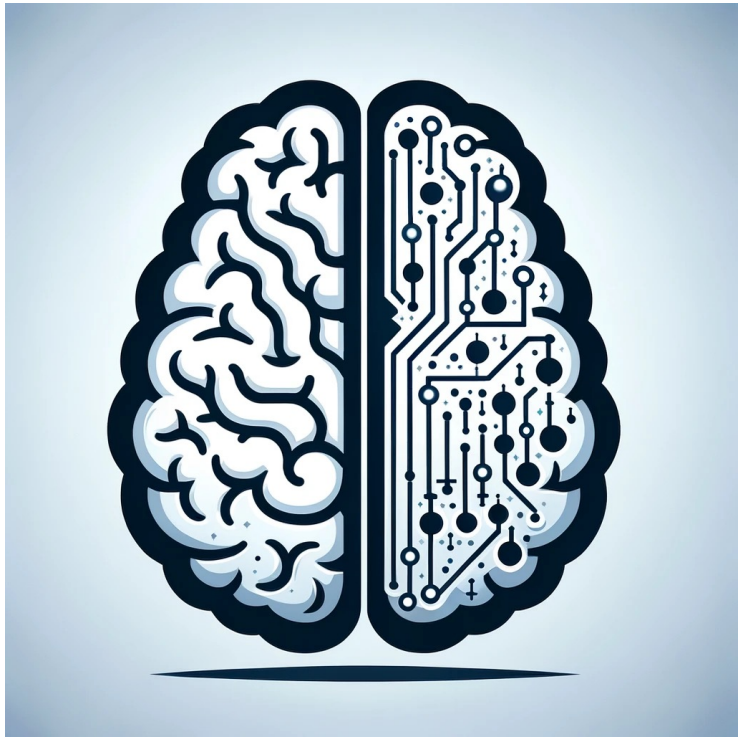
EN 601.473/601.673: Cognitive Artificial Intelligence (CogAI)



Lecture 19: Theory of Mind

Tianmin Shu

Final project

- Presentation dates: April 23 and April 25
- Make sure you have written the title and full names of your team members in your proposal, so that your project will be scheduled and graded correctly
- Mandatory attendance so you can learn others' works and interact with your peers. Let me know if you have any time constraints by the end of the week.
- More instructions to follow on how to prepare for the presentations.

Physical and social reasoning

- Common sense scene understanding, intuitive theories
- Physical reasoning
- Social reasoning

How an 18-month-old child helps another person

Understand other people
1. single-agent behavior
2. multi-agent interactions

Interact with other people

Social Scene Understanding

More advanced topics:
- Recursive reasoning
- Communication
- Moral judgment

Multi-agent Cooperation

Outline

Social Interaction

Theory of Mind

Animacy

Outline

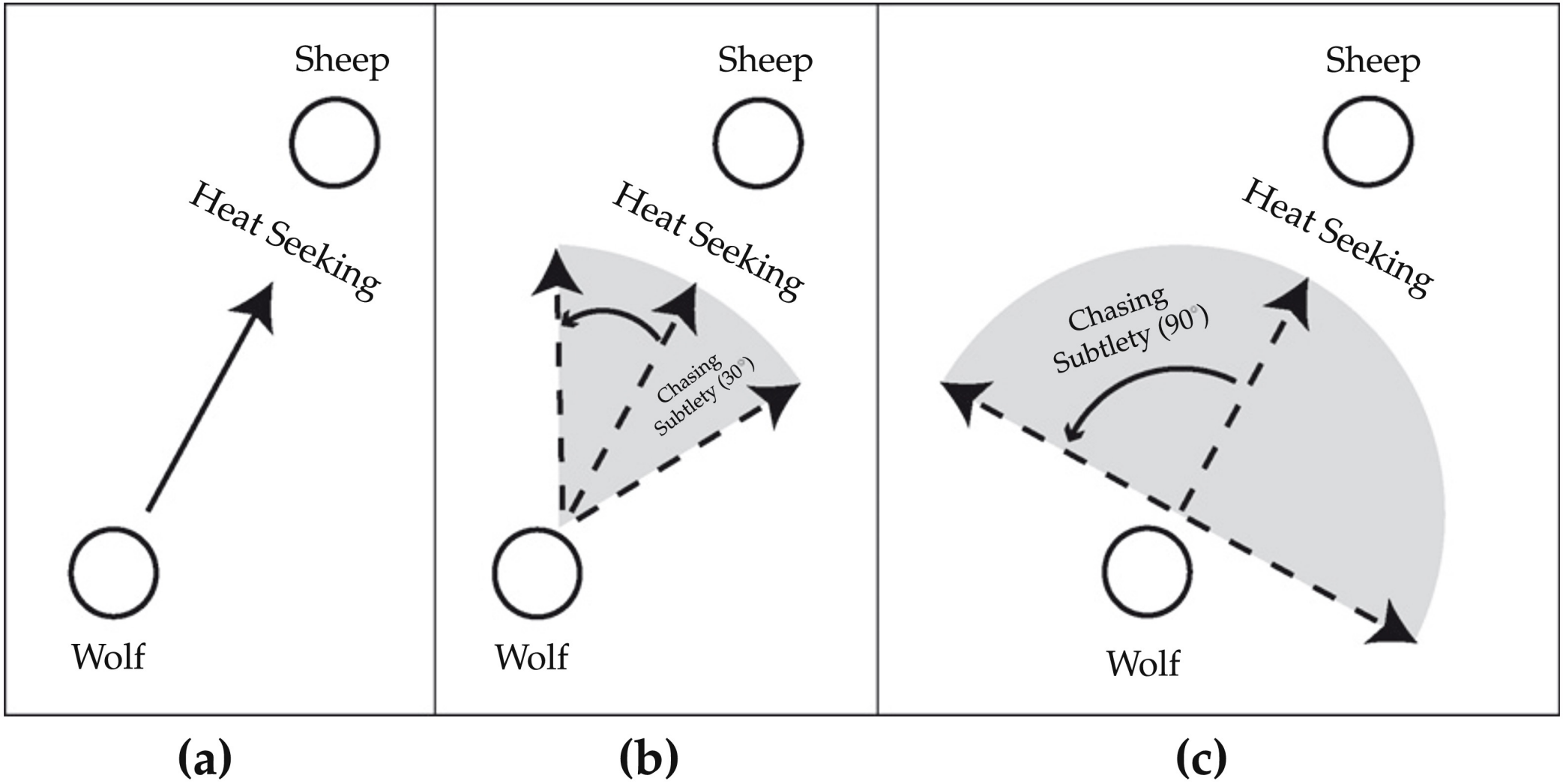
Social Interaction

Theory of Mind

Animacy

Example 2: The psychophysics of chasing

Varying degree of chasing subtlety

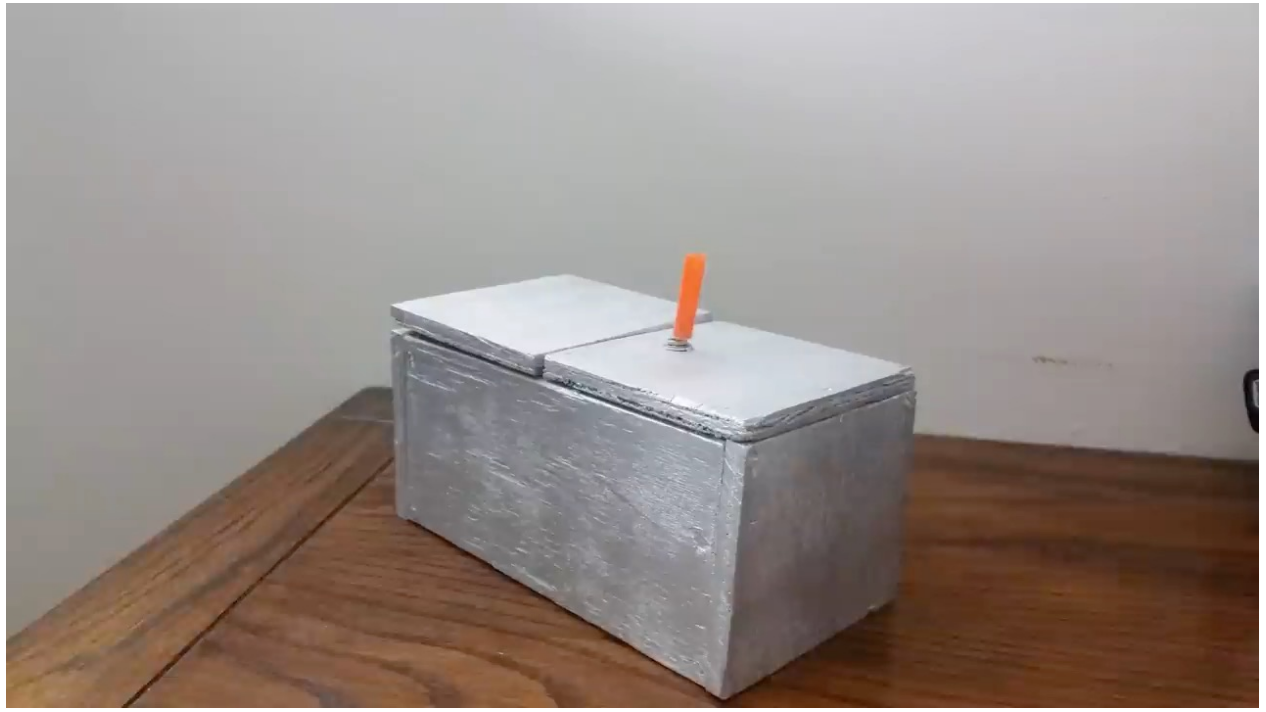


“Useless machines”

Which machine is more like an agent?

Useless machine A

Useless machine B



Need a mentalistic representation!

Outline

Social Interaction

Theory of Mind

Animacy

Theory of Mind

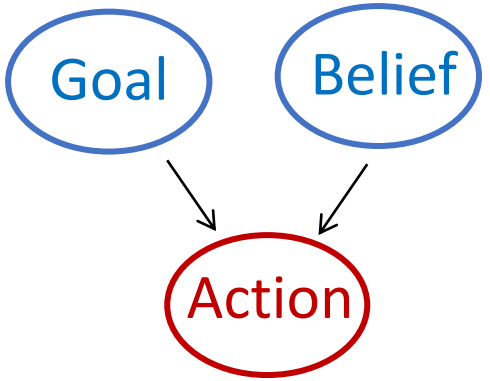
- Basic concepts and evaluation
- Single-agent decision making
- Inverse single-agent decision making

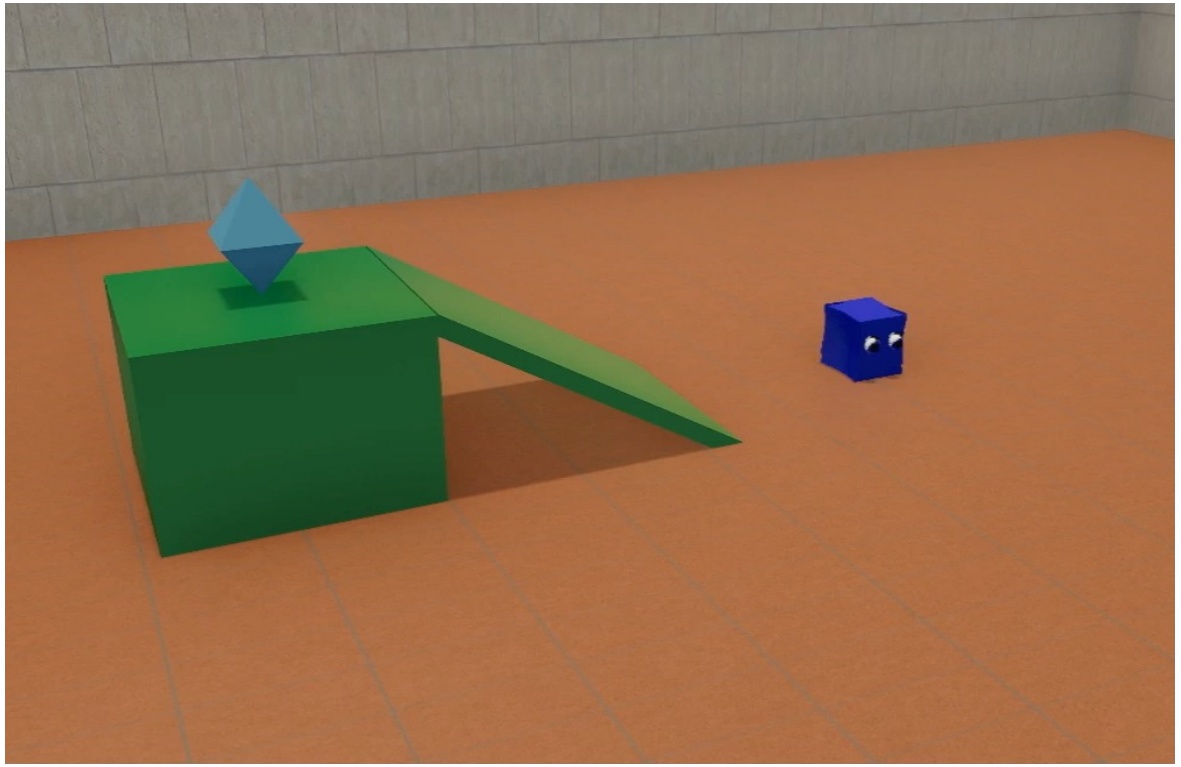
Theory of Mind

- Basic concepts and evaluation
- Single-agent decision making
- Inverse single-agent decision making

Theory of Mind in humans

Theory of Mind: Reasoning about **hidden mental** variables that drive **observable actions**





The agent takes the most ***efficient action*** to reach its ***goal*** (blue diamond), maximizing its ***reward*** and minimizing the ***cost,*** subject to ***constraints***

Basic concepts in Theory of Mind and their evaluation

Inspiration from developmental psychology: what basic concepts + how to evaluate them

https://psych.ubc.ca/news/500-babies-a-new-study-explores-social-development-from-birth-to-age-three/

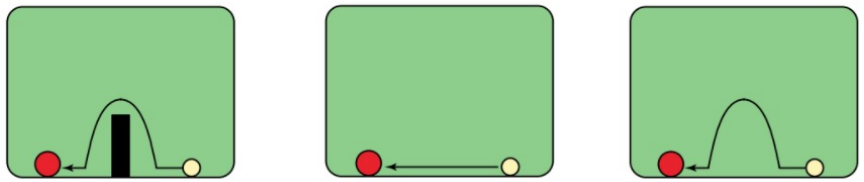
Developmental trajectory

3-6 months ⋯ **9-12 months** ⋯

Agents can change object motion
Agents have goals
Agents act efficiently

Unseen agent can cause visible outcomes
Judge efforts needed to reach a goal
Infer if agents help/hinder
Understand what agents can or cannot see

Familiarization Test (Expected) Test (Surprising)



Trials for probing babies' social common sense

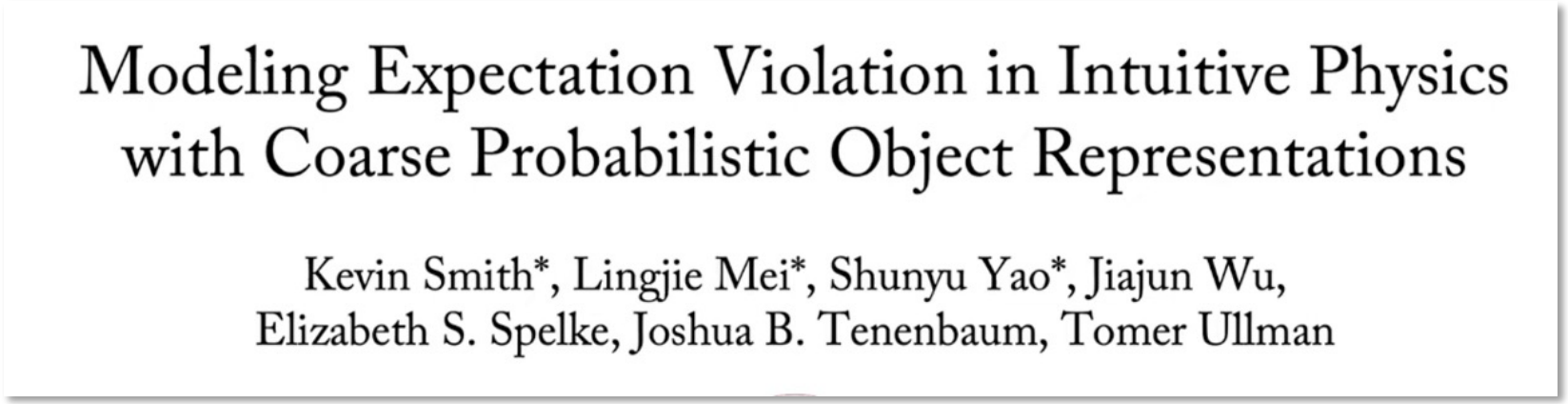
Gergely & Csibra (2003)

How do we measure infants' intelligence?

- Looking time!
- Longer looking time indicates a higher degree of surprise

Violation of Expectation paradigm

- Violation of expectation measured by looking time
- E.g., infants expect physically plausible events and are surprised by violation of physics → machine intuitive physics studies



Modeling Expectation Violation in Intuitive Physics with Coarse Probabilistic Object Representations

Kevin Smith*, Lingjie Mei*, Shunyu Yao*, Jiajun Wu,
Elizabeth S. Spelke, Joshua B. Tenenbaum, Tomer Ullman

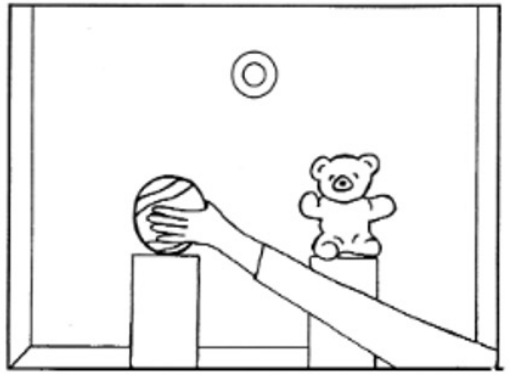
NeurIPS 2019

http://physadept.csail.mit.edu/

* indicates equal contribution

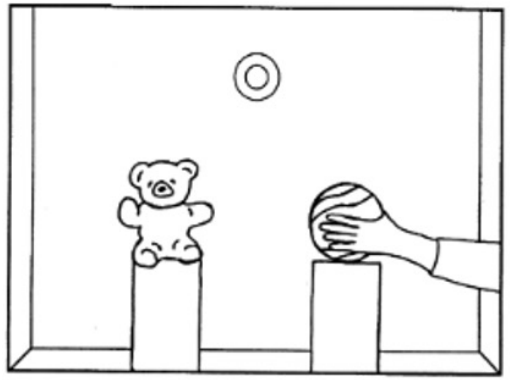
Violation of Expectation paradigm for Theory of Mind

Familiarization

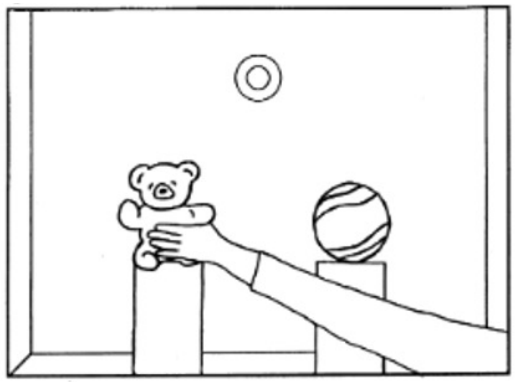


Example from Woodward (1998)

Test

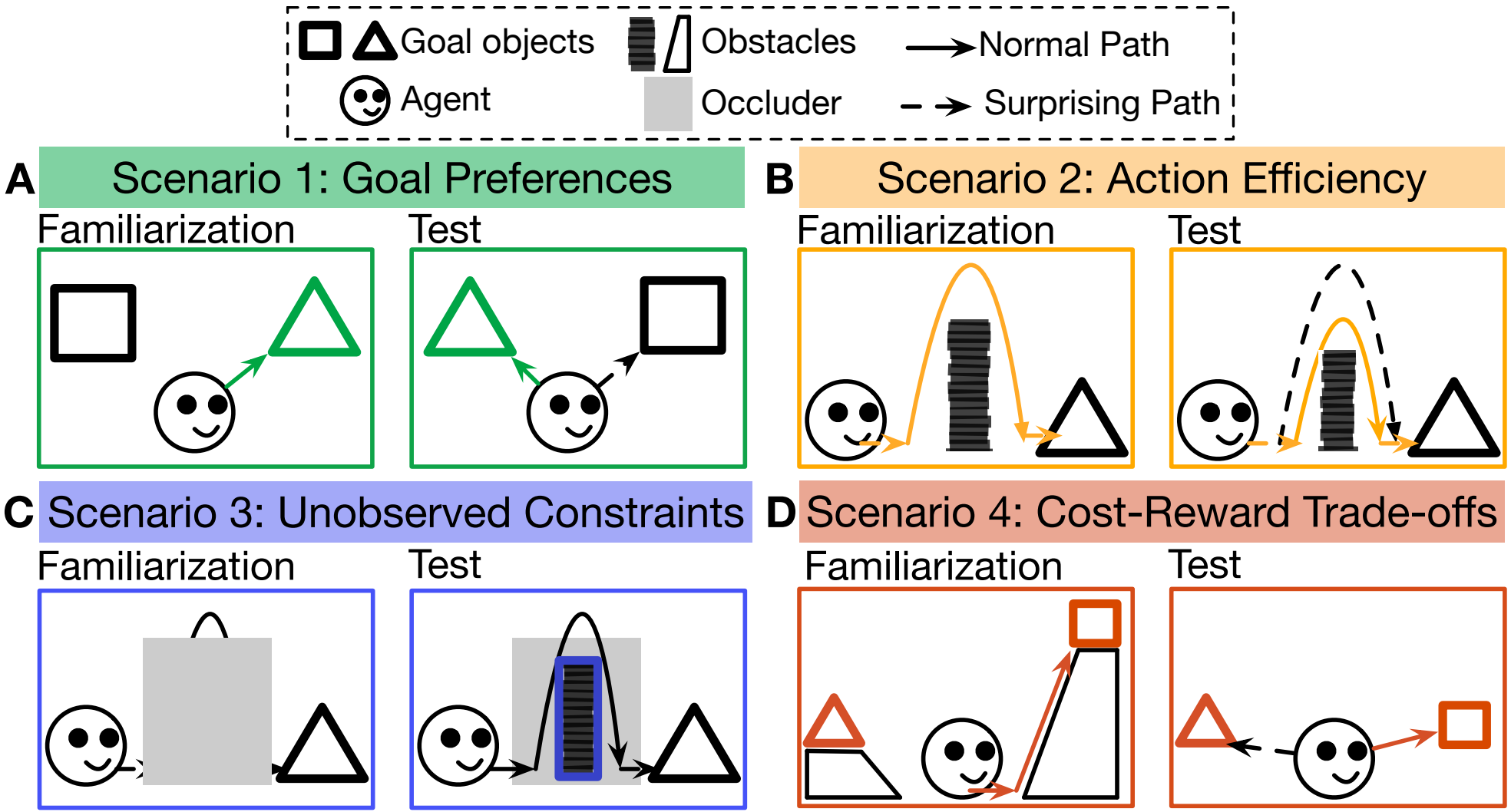


Expected => shorter looking



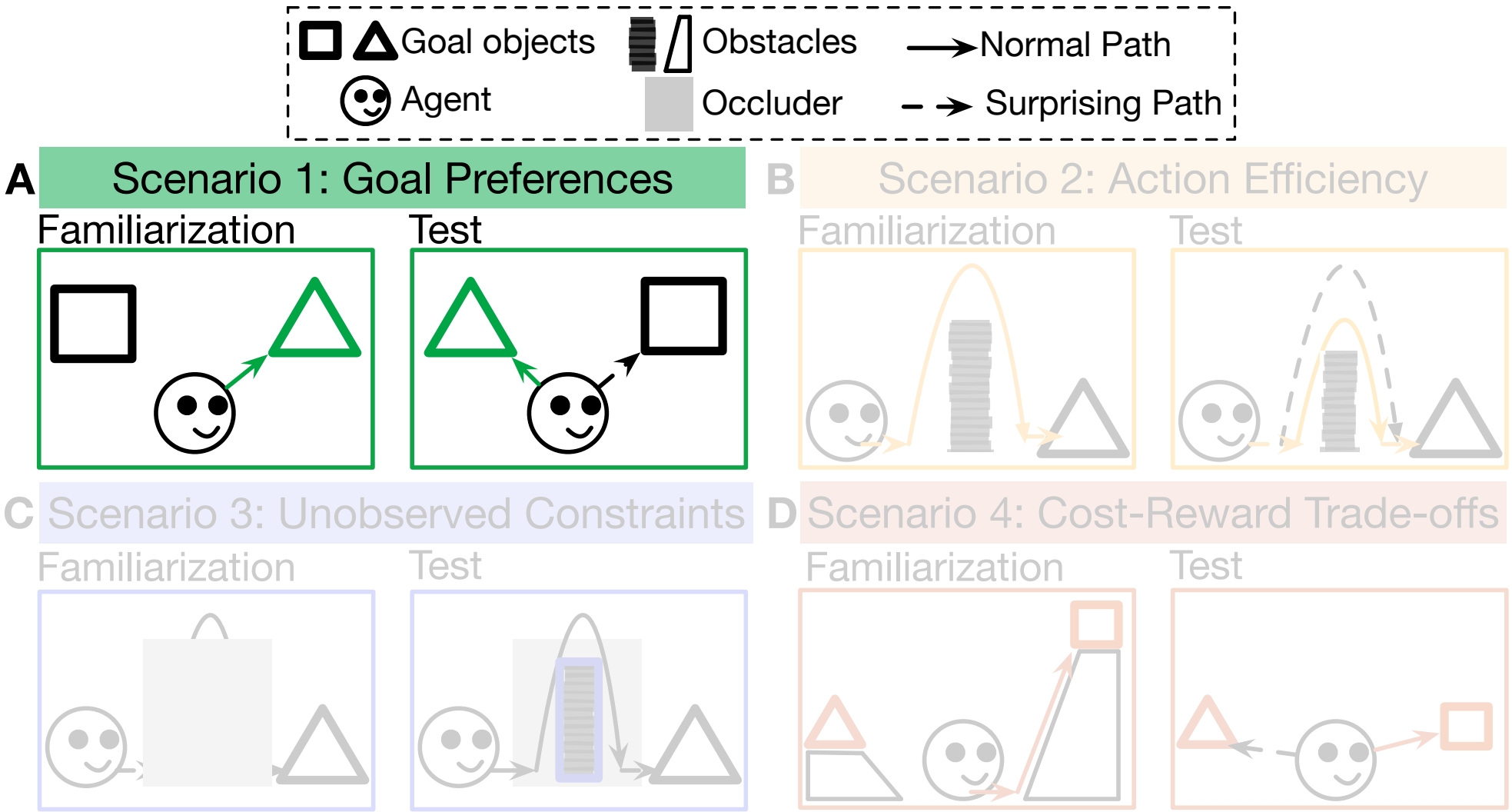
Surprising => longer looking

AGENT: Action, Goal, Efficiency, coNstraint, uTility



Shu et al. AGENT: A Benchmark for Core Psychological Reasoning. *ICML 2021*

AGENT: Action, Goal, Efficiency, coNstraint, uTility



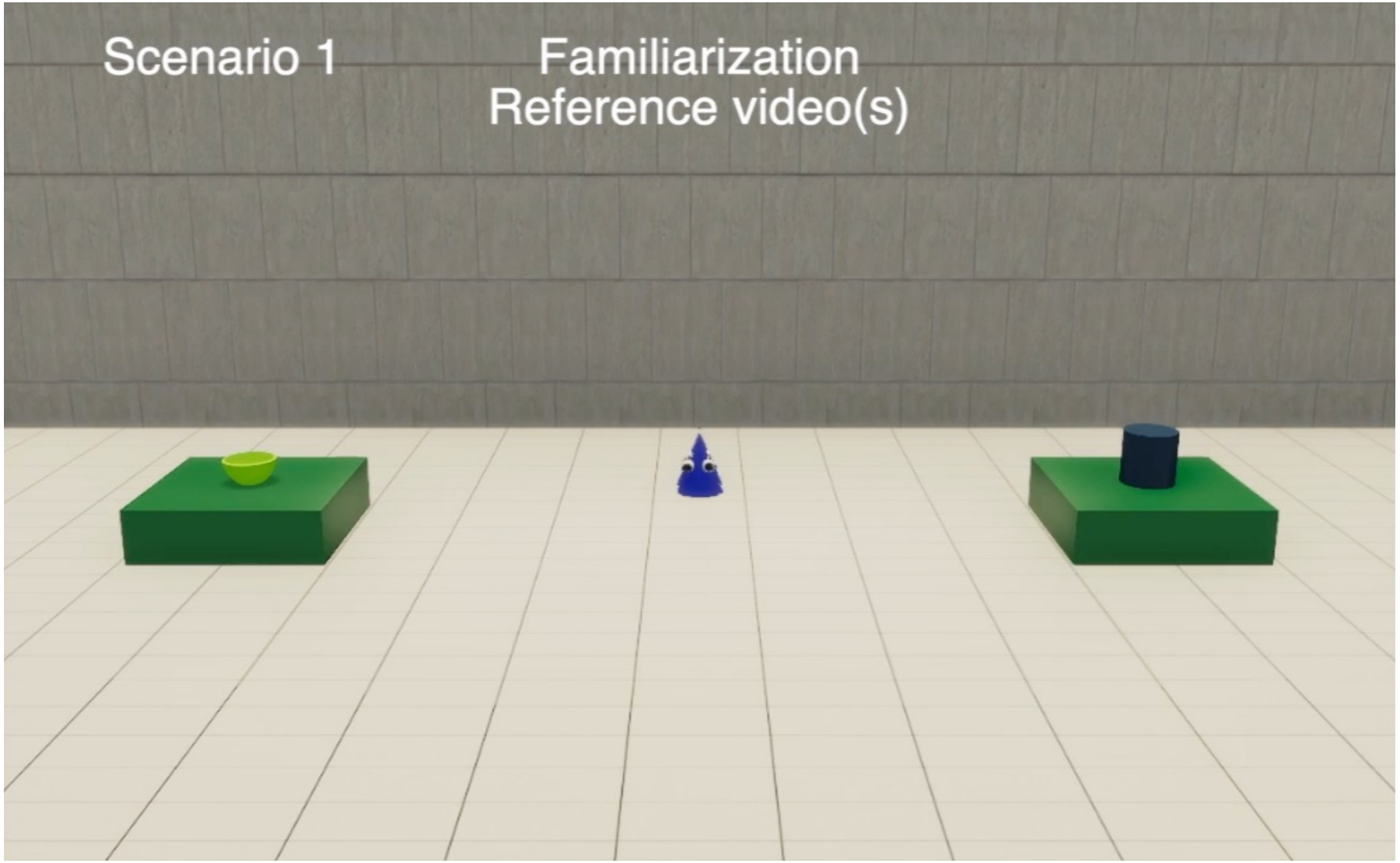
Example trial for scenario 1

Familiarization

Test

Example trial for scenario 1

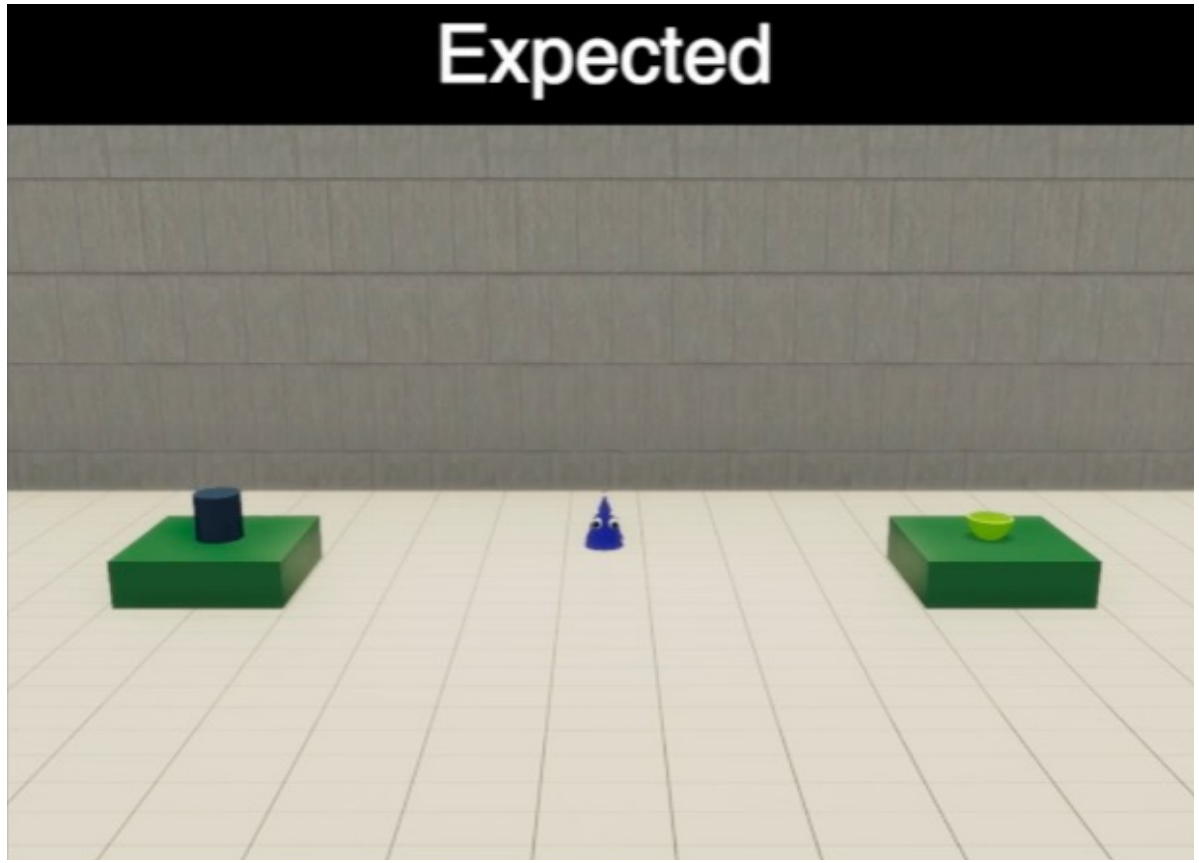
Familiarization | Test

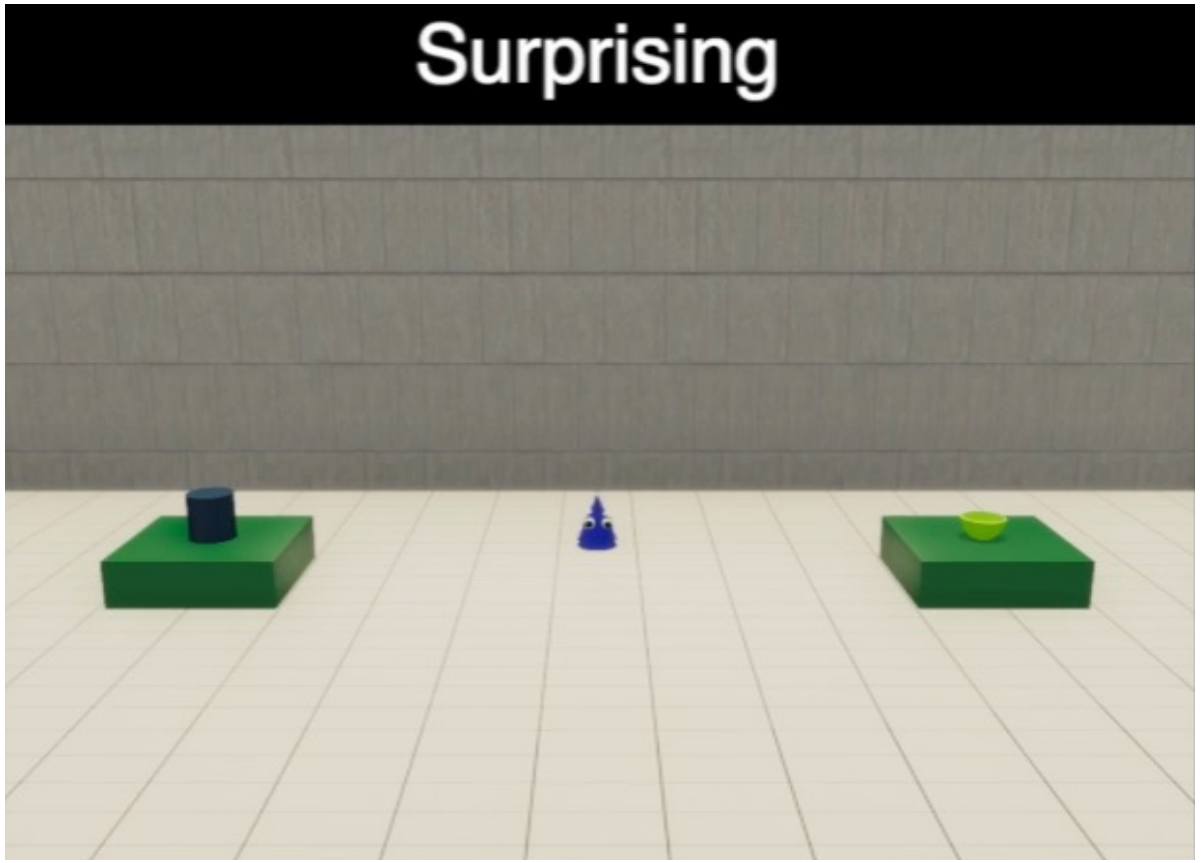


Example trial for scenario 1

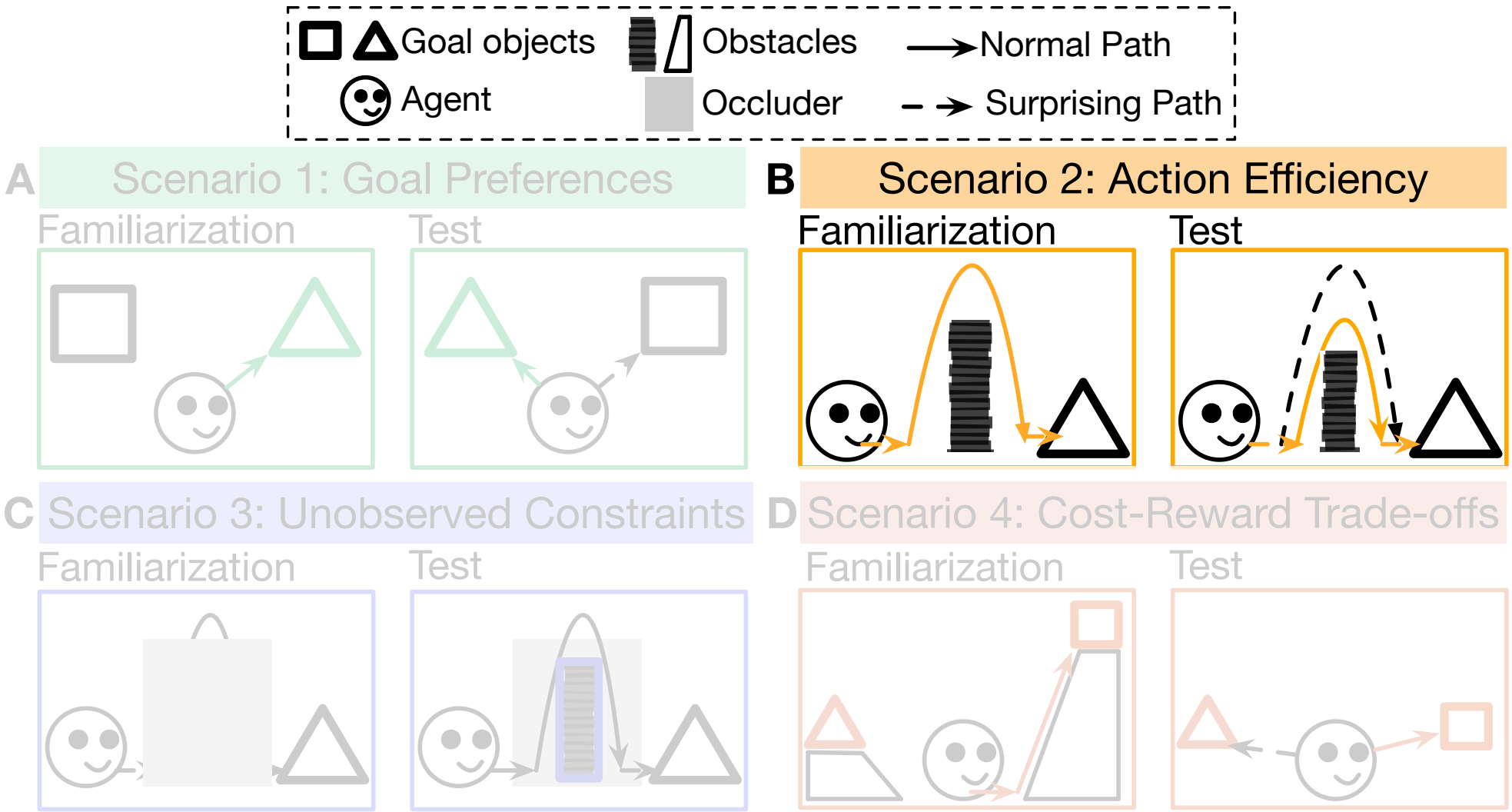
Familiarization

Test





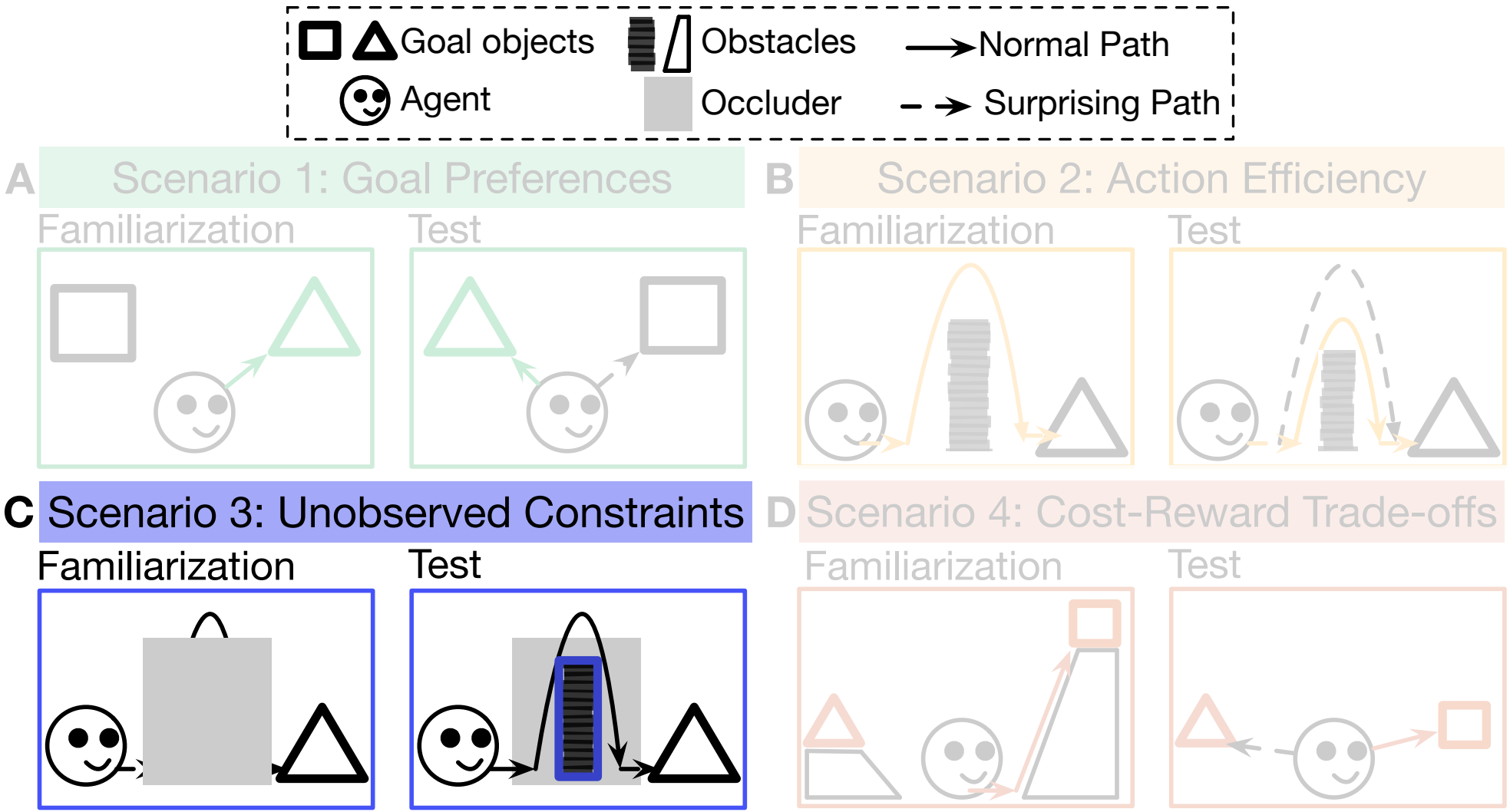
AGENT: Action, Goal, Efficiency, coNstraint, uTility



Type 2.1 Example Familiarization 2x

Type 2.1 Example Familiarization 2x

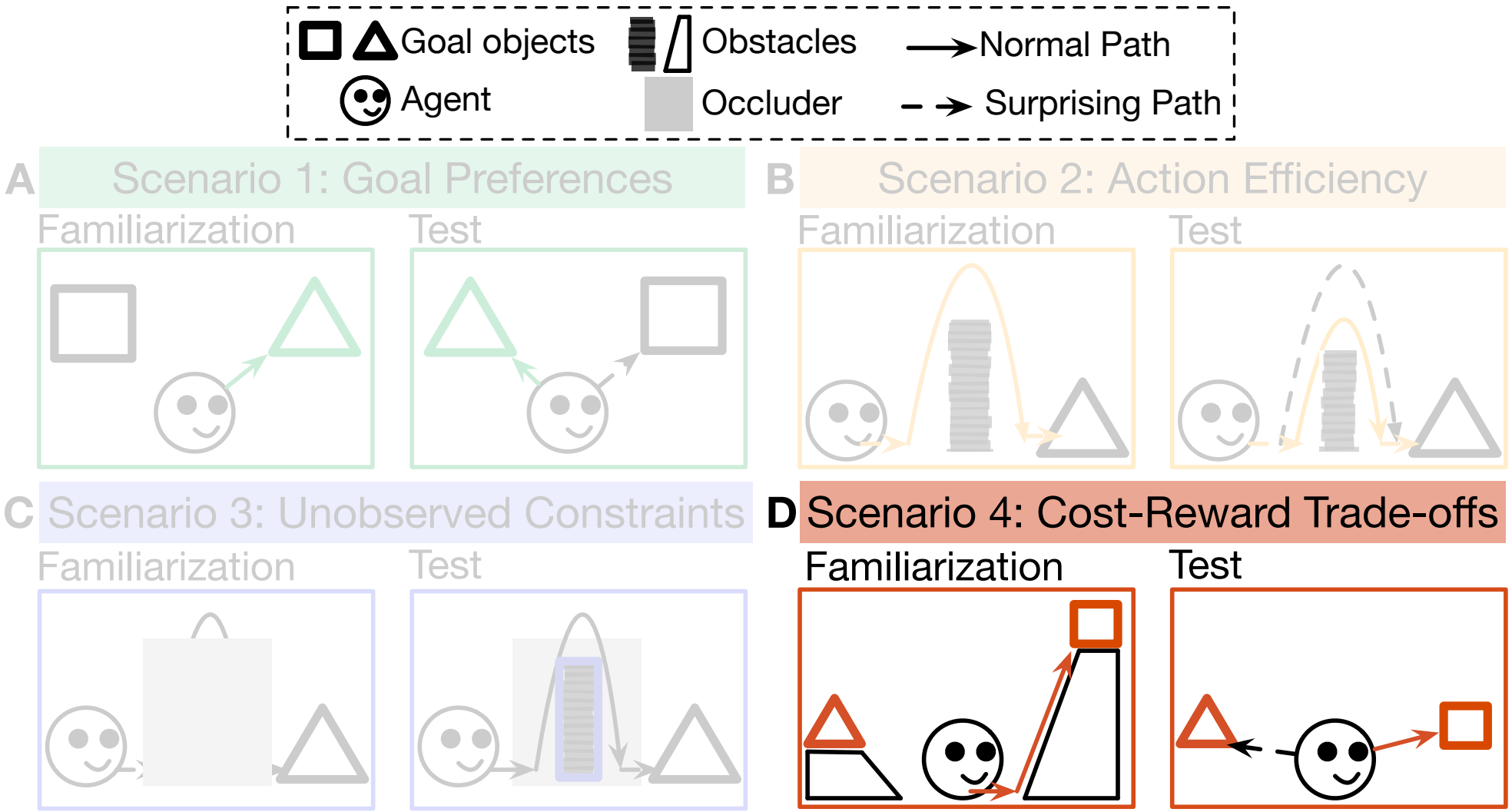
AGENT: Action, Goal, Efficiency, coNstraint, uTility



Type 3.1 Example Familiarization 2x

Type 3.1 Example Familiarization 2x

AGENT: Action, Goal, Efficiency, coNstraint, uTility



Type 4.1 Example

Familiarization

2x

Familiarization 1

Familiarization 2

Medium cost

High cost

Type 4.1 Example

Familiarization

2x

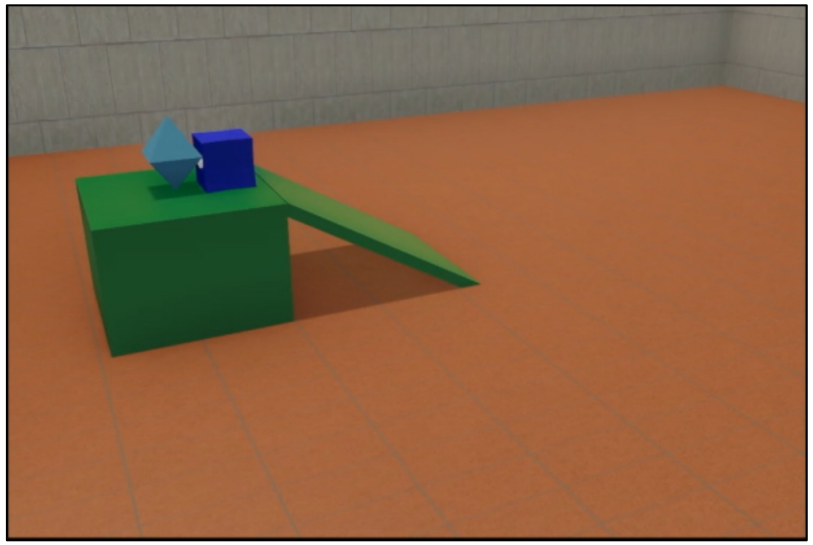
Familiarization 1

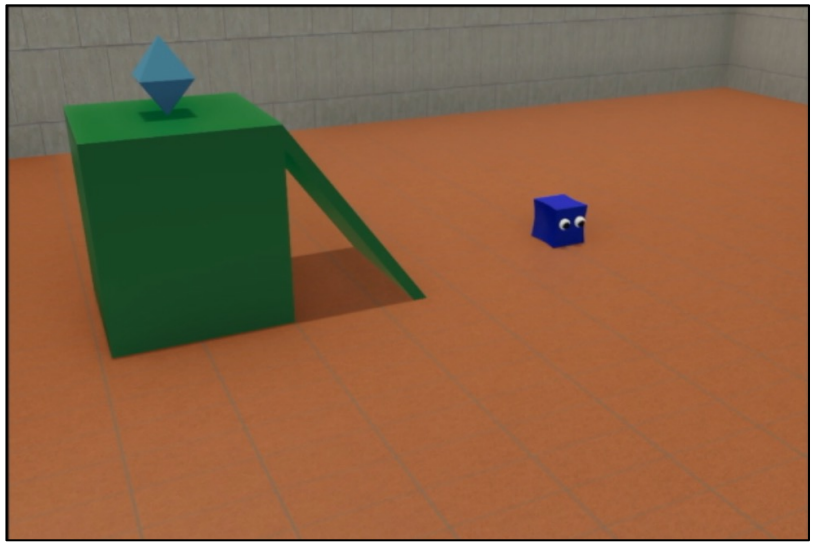
Familiarization 2

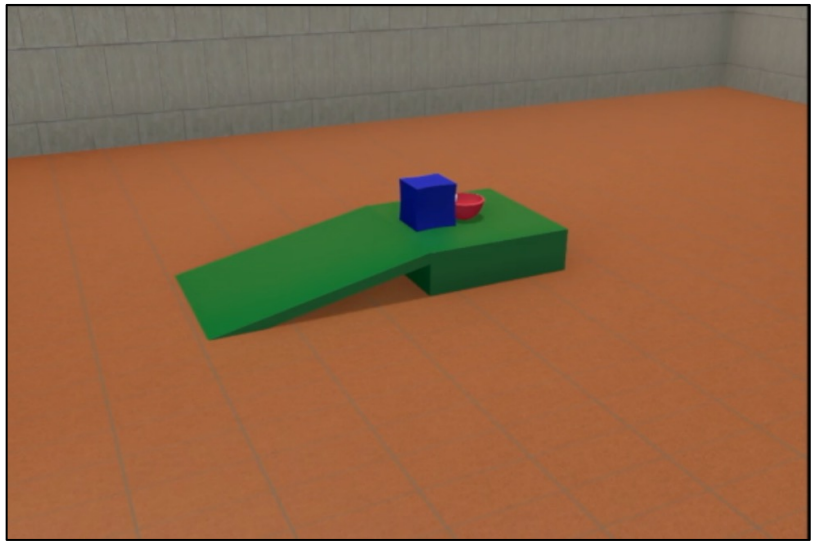
Low cost

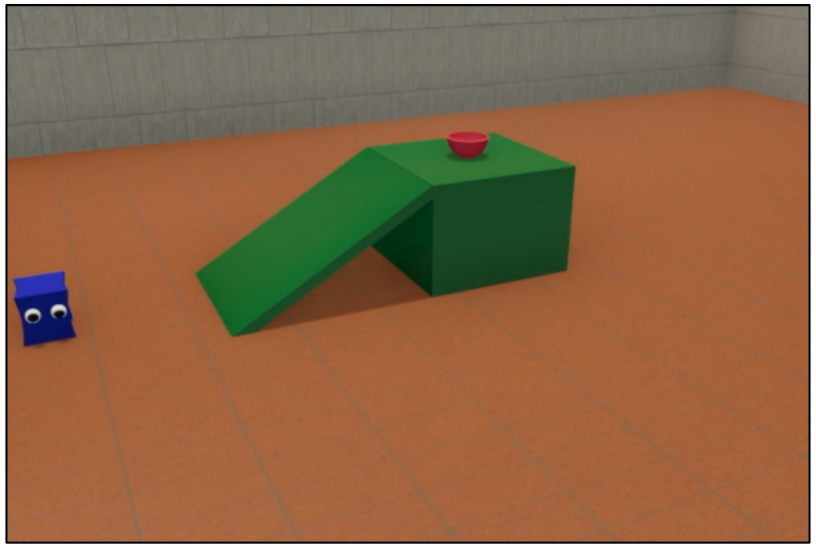
Medium cost

Reward of the blue diamond is higher







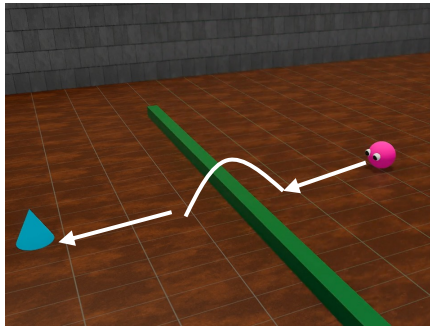


Type 4.1 Example Familiarization 2x

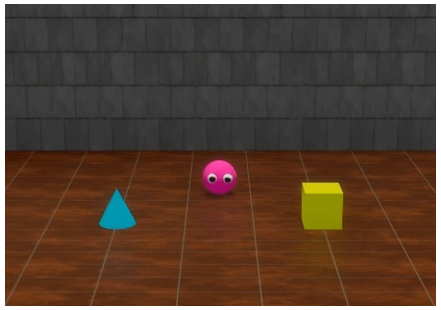
Familiarization 1

Familiarization 2

Evaluation for each trial

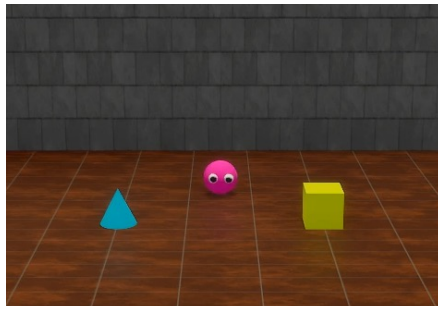


Reference



Test (Expected)

Correct iff <



Test (Surprising)

Model

$S_{expected}$

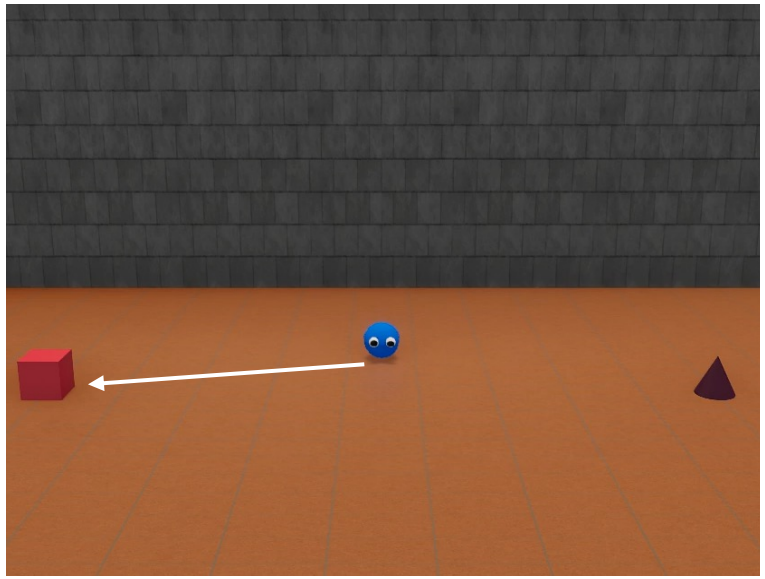
Model

$S_{surprising}$

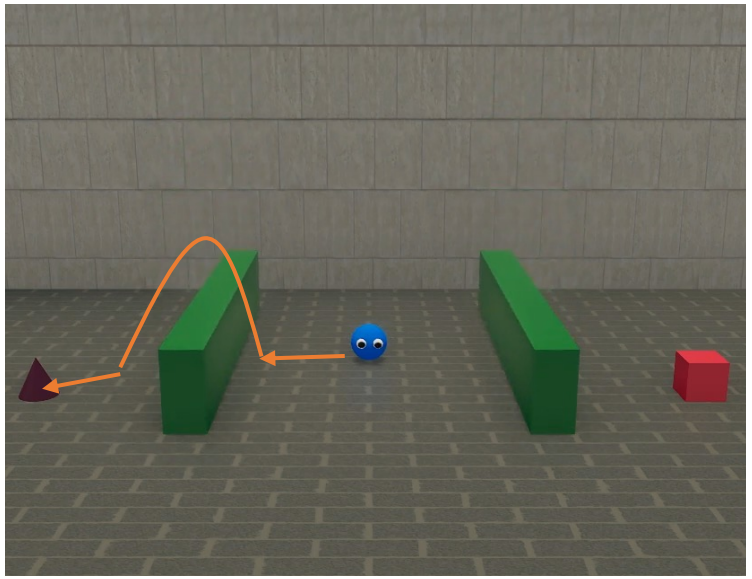
Multiple types of trials for each scenario

Example types in scenario 1

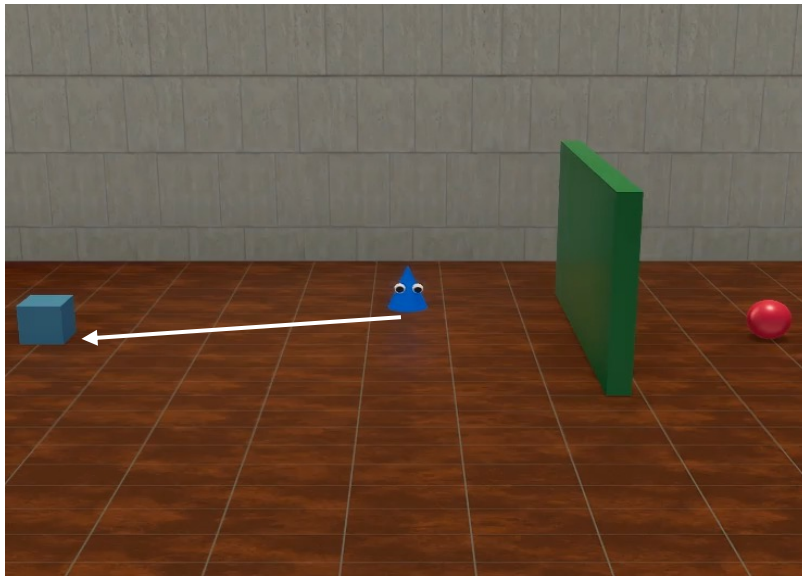
Type 1: no barriers



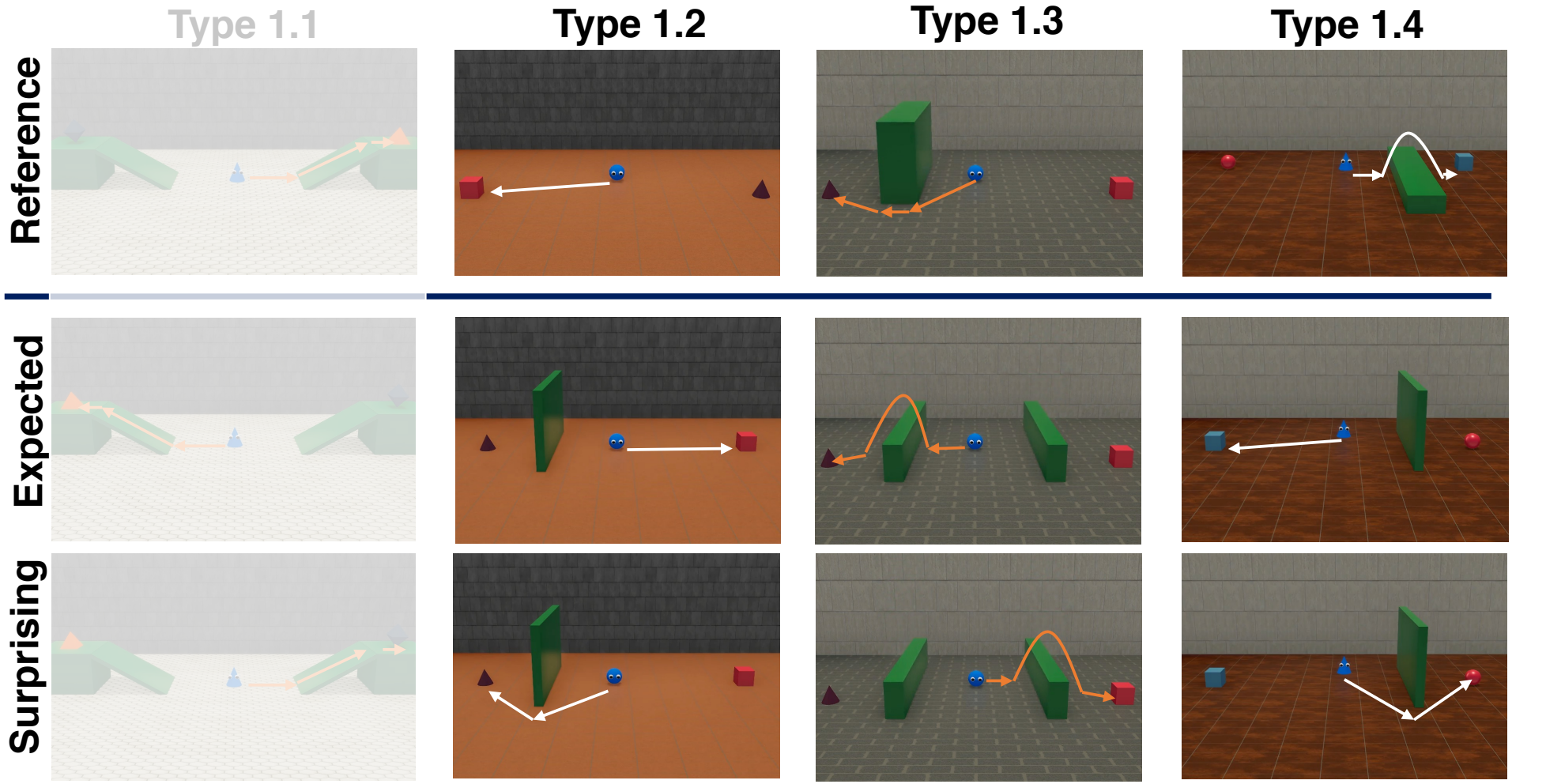
Type 3: barriers between agent and both objects



Type 4: one barrier between agent and one of the objects

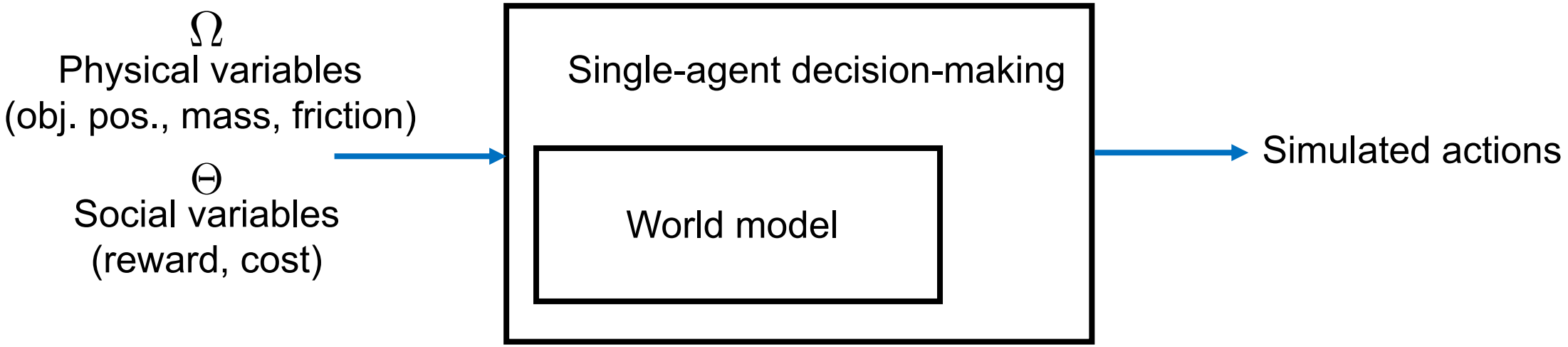


Generalization evaluation: leave one type out

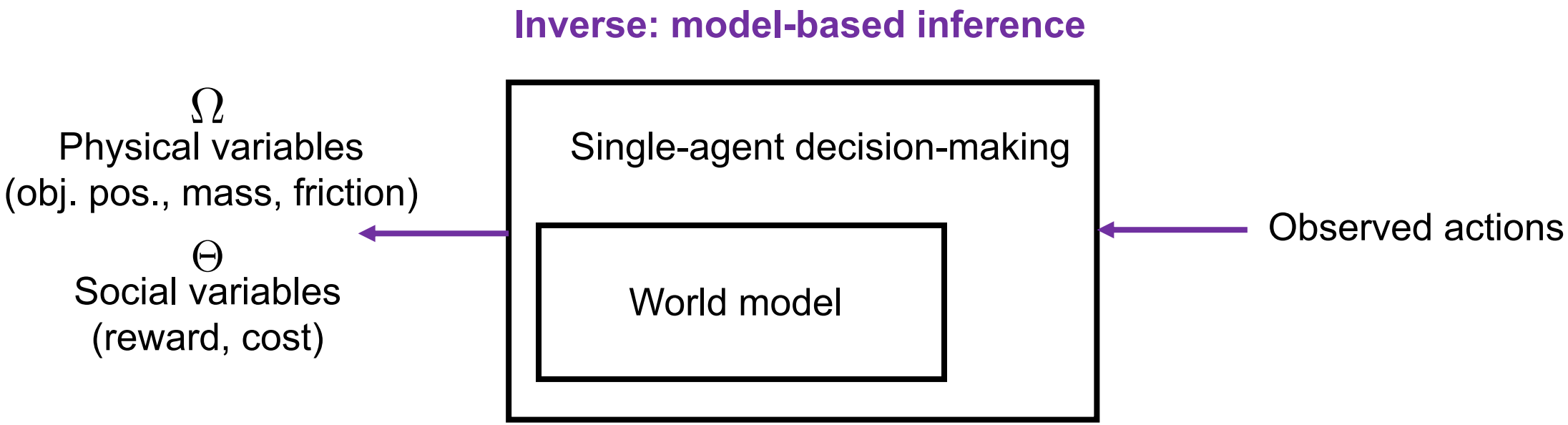


A generative model of single-agent behavior

Forward: simulating human-like behavior



A generative model of single-agent behavior



Theory of Mind

- Basic concepts and evaluation
- Single-agent decision making
- Inverse single-agent decision making

Single-agent decision making

- Markov decision process (MDP)
- Value, policy
- Model-based approaches for solving MDPs
- Model-free RL (Q-learning)

The basic idea: agent and environment

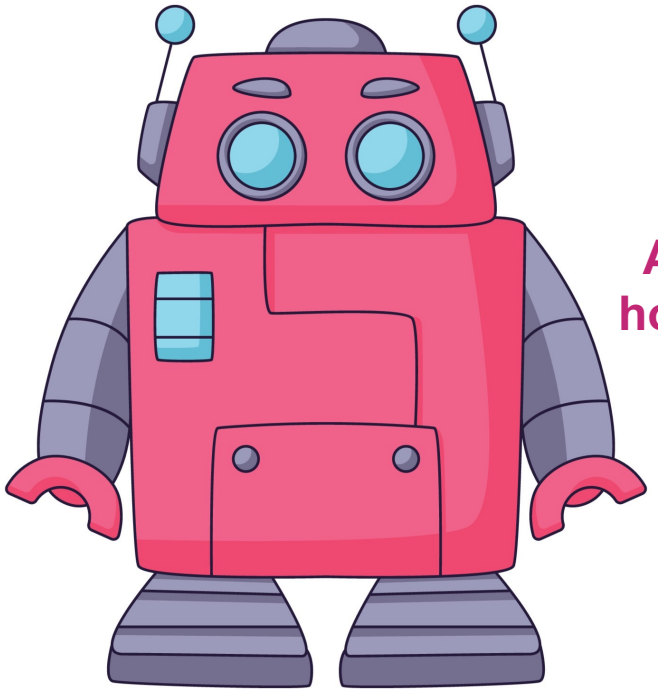
Goal: find the best actions to maximize the total reward

State: current screen

Action:  

Reward: score

Agent



Action

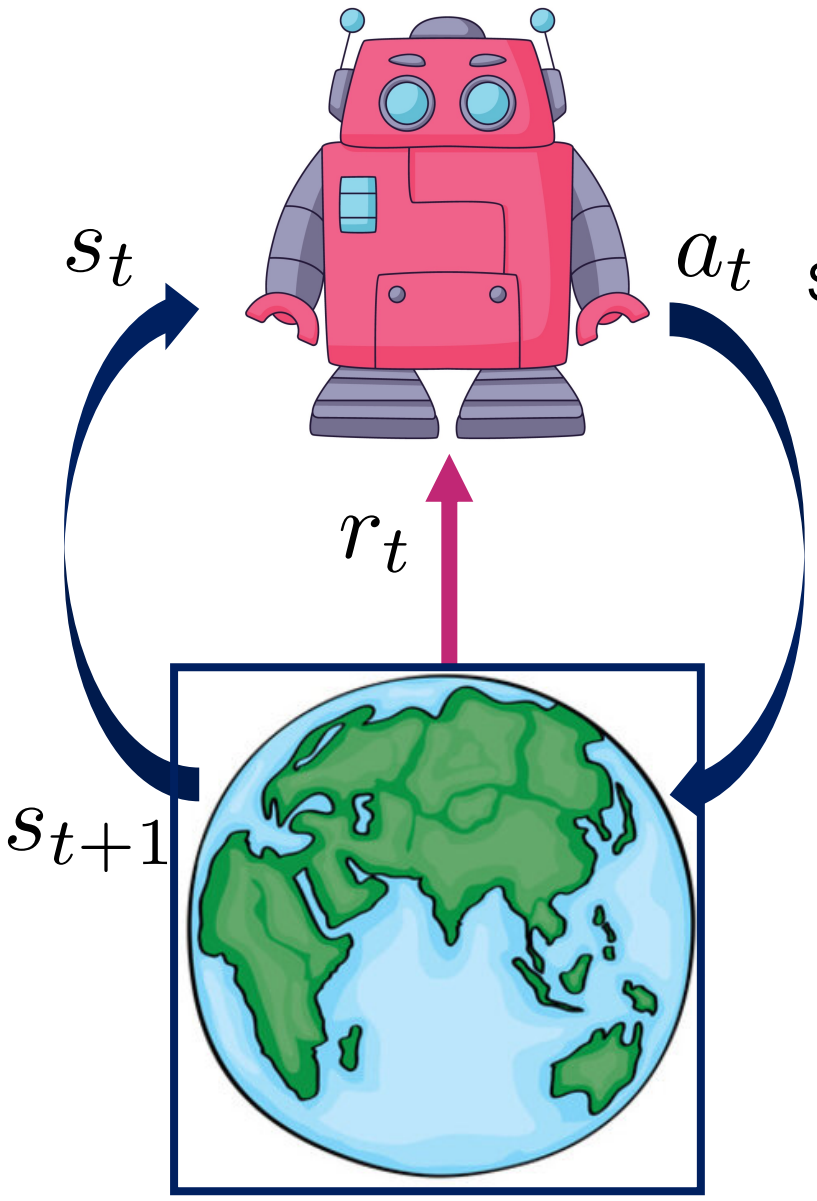
Reward

A scalar value indicating how well the agent is doing

State

Environment

Markov decision process (MDP)



At each time step t

State $s_t \in \mathcal{S}$ ←State space

Action $a_t \in \mathcal{A}$ ←Action space

State transition probabilities (world model) $P(s_{t+1}|s_t, a_t)$

Reward function $r_t = R(s_t, a_t)$

- Deterministic: step on firm ground: stay on the ground
- Stochastic: step on thin ice: may or may not fall into the water

Discounted factor $\gamma \in [0, 1]$

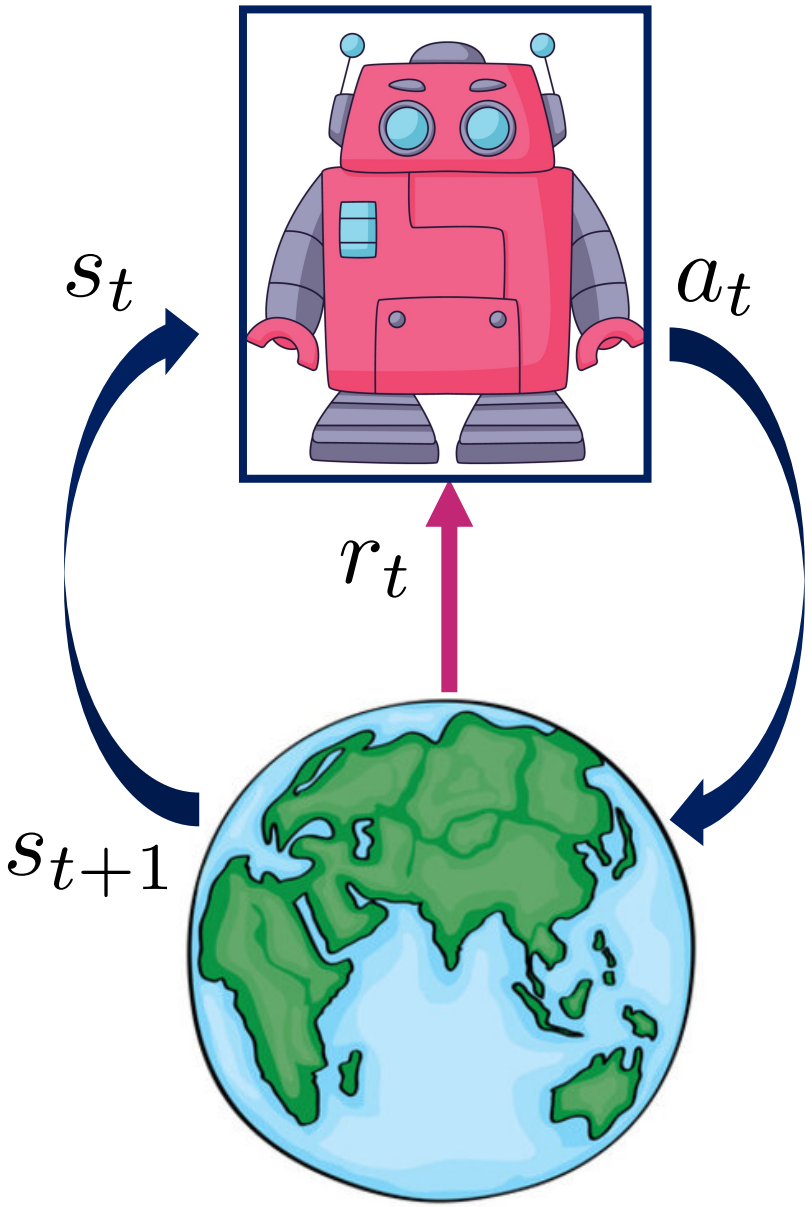
Return G_t The accumulated reward from step *t*

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots = \sum_{k=0}^{\infty} \gamma^k R_{t+k}$$

Why discounting the future rewards? Many reasons, e.g.,

- Account for uncertainty in the future
- Avoid infinite return

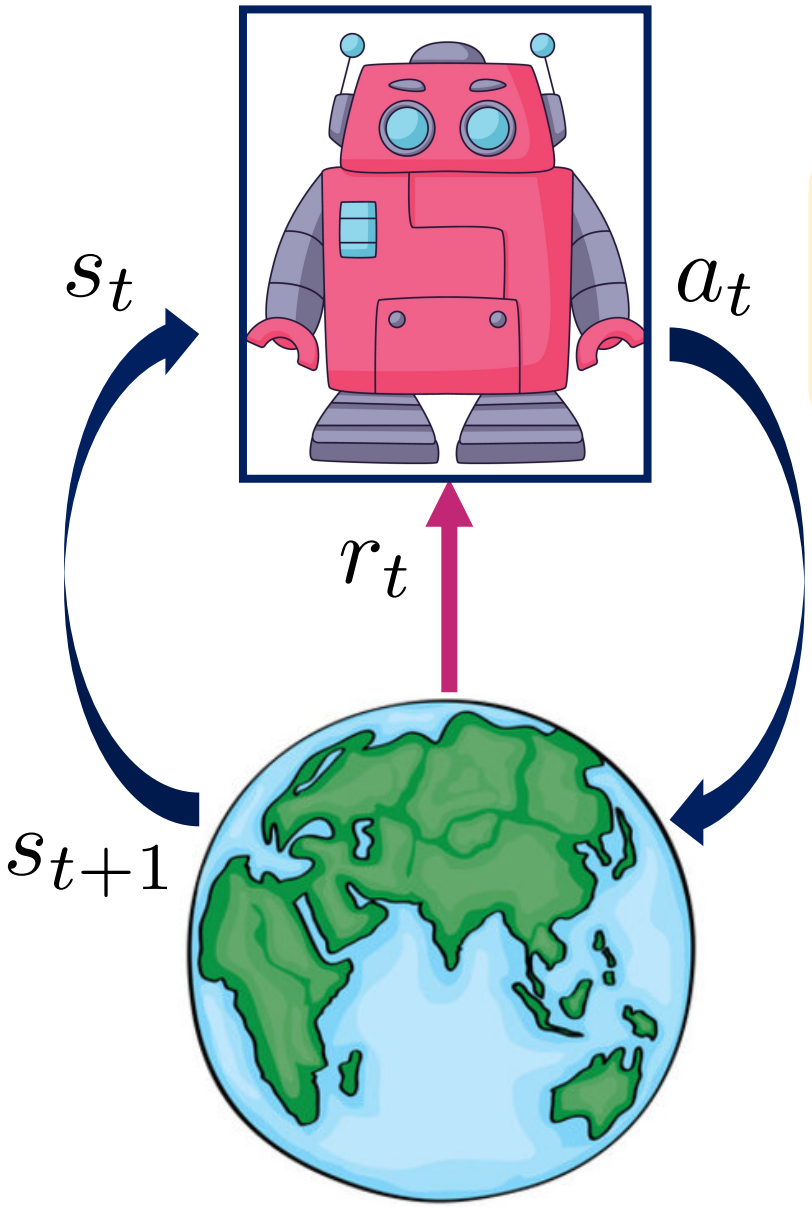
“Brain” of the agent



Policy

Value Function

Policy



A policy π is a distribution of actions given the current state

$$a_t \sim \pi(a_t|s_t)$$

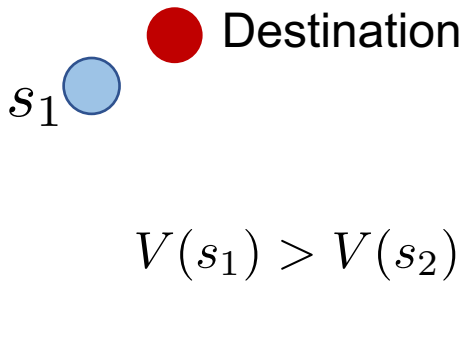
The policy decides what the agent will do in the given environment

$$s_t \rightarrow a_t \overset{r_t}{\rightarrow} s_{t+1} \rightarrow a_{t+1} \overset{r_{t+1}}{\rightarrow} s_{t+2} \rightarrow \cdots$$

Value function

Evaluation of a policy π

$V^{\pi}(s)$ helps estimate how good it is to be in a state



A **state-value** function $V^{\pi}(s)$ is the expected return from the current state *s*, and then following policy π

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t | s_t = s]$$

$s_t \rightarrow a_t \xrightarrow{r_t} s_{t+1} \rightarrow a_{t+1} \xrightarrow{r_{t+1}} s_{t+2} \rightarrow \cdots \; G_t$

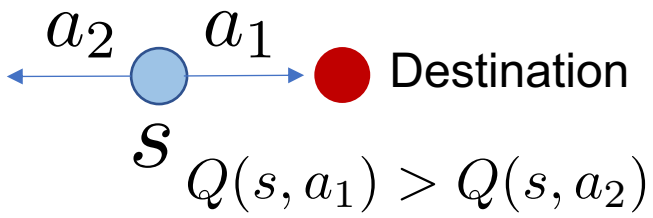
$s_t \rightarrow a_t \xrightarrow{r_t} s_{t+1} \rightarrow a_{t+1} \xrightarrow{r_{t+1}} s_{t+2} \rightarrow \cdots \; G_t$

$s_t \rightarrow a_t \xrightarrow{r_t} s_{t+1} \rightarrow a_{t+1} \xrightarrow{r_{t+1}} s_{t+2} \rightarrow \cdots \; G_t$

$\vdots$ $\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad\qquad$ $\vdots$

Average return of all possible sequences of actions sampled from the policy π

$Q^{\pi}(s, a)$ helps estimate which action in a given state can lead to a higher return in the future

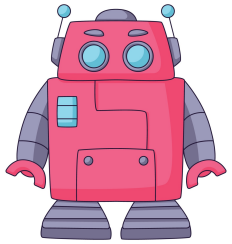


An **action-value** function $Q^{\pi}(s, a)$ is the expected return from the current state *s*, taking the current action *a*, and then following policy π

$$Q^{\pi}(s, a) = \mathbb{E}_{\pi}[G_t | s_t = s, a_t = a]$$

A toy example

An **episode**: move from the initial state to the terminal state

Reward	0	0	+1	0	0
			 (Terminal)		
Policy π	→	→	STOP	→	→
Value Function $V^{\pi}(s)$ $\gamma = 0.9$	0.81	0.9	1	0	0

Bellman equations

Bellman equation for $V^{\pi}(s)$

Immediate reward Discounted value of the next state

$$V^{\pi}(s) = \mathbb{E}_{\pi}\left[r_t + \gamma V^{\pi}(s_{t+1}) \middle| s_t = s\right]$$

Bellman equations

Bellman equation for $V^{\pi}(s)$

$$G_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots$$

Immediate reward | Discounted value of the next state

$$V^{\pi}(s) = \mathbb{E}_{\pi}[G_t | s_t = s]$$

Definition of G_t

$$= \mathbb{E}_{\pi}[r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} + \cdots | s_t = s]$$

Refactorization

$$= \mathbb{E}_{\pi}[r_t + \gamma(r_{t+1} + \gamma r_{t+2} + \cdots) | s_t = s]$$

Definition of G_{t+1}

$$= \mathbb{E}_{\pi}[r_t + \gamma G_{t+1} | s_t = s]$$

Definition of $V^{\pi}(s_{t+1})$

$$= \mathbb{E}_{\pi}[r_t + \gamma V^{\pi}(s_{t+1}) | s_t = s]$$

Bellman equations

Bellman equation for $V^{\pi}(s)$

Immediate reward Discounted value of the next state

$$V^{\pi}(s) = \mathbb{E}_{\pi}[r_t + \gamma V^{\pi}(s_{t+1}) | s_t = s]$$

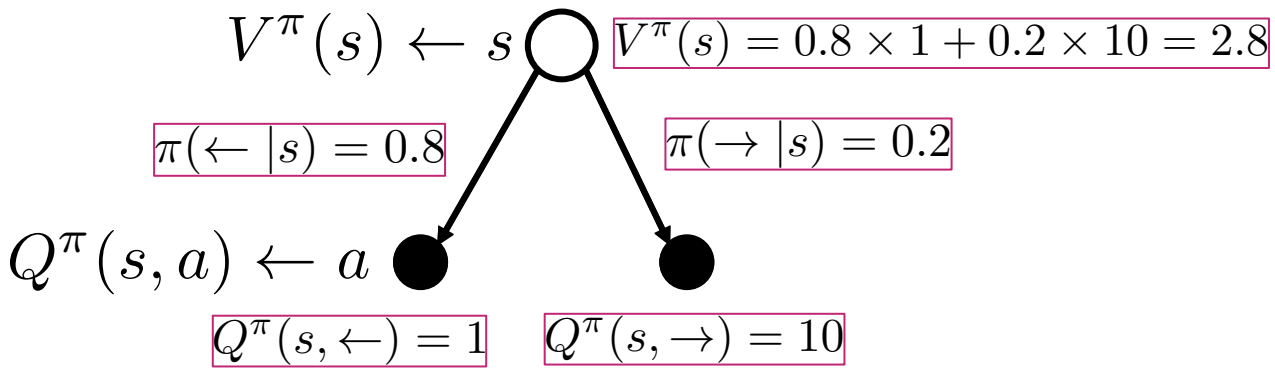
Bellman equation for $Q^{\pi}(s, a)$

$$Q^{\pi}(s, a) = E_{\pi}[r_t + \gamma Q^{\pi}(s_{t+1}, a_{t+1}) | s_t = s, a_t = a]$$

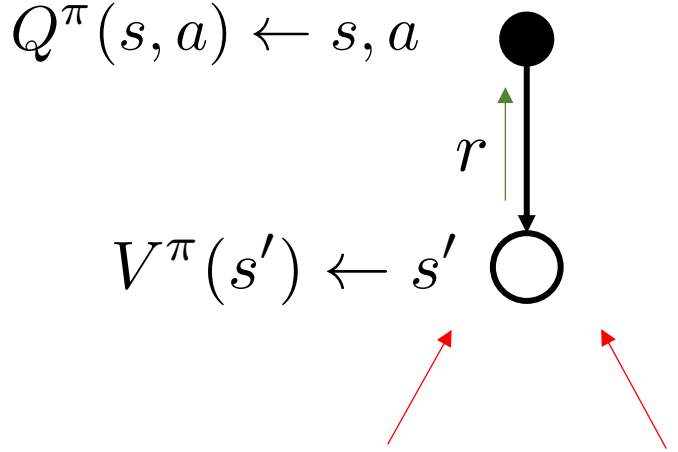
Relationship between $V^{\pi}(s)$ and $Q^{\pi}(s, a)$?

Backup diagrams for Bellman equations

Bellman equation for $V^\pi(s)$



Bellman equation for $Q^\pi(s,a)$



$$Q^\pi(s',a') \leftarrow s',a'$$

$$V^\pi(s) = \sum_{a\in\mathcal{A}} \pi(a|s) Q^\pi(s,a)$$

$$Q^\pi(s,a) = R(s,a) + \gamma V^\pi(s')$$

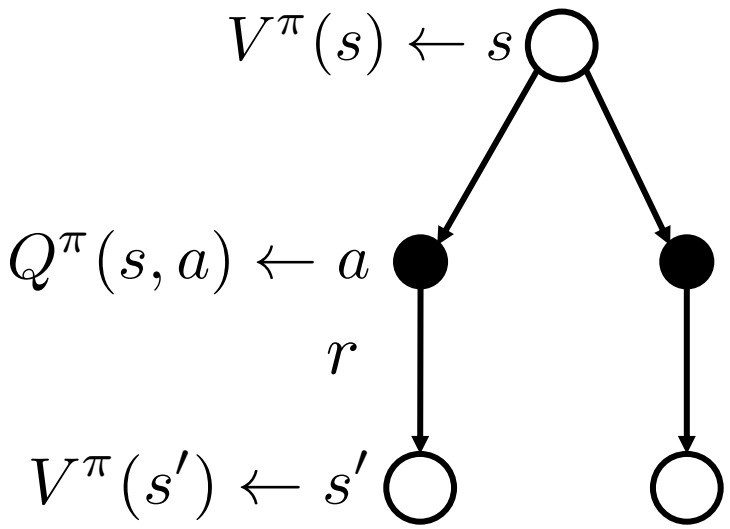
```
def func():
    func()
func()
```

Recursive Call

$$Q^\pi(s,a) = R(s,a) + \gamma \sum_{a'\in\mathcal{A}} \pi(a'|s') Q^\pi(s',a')$$

Backup diagrams for Bellman equations

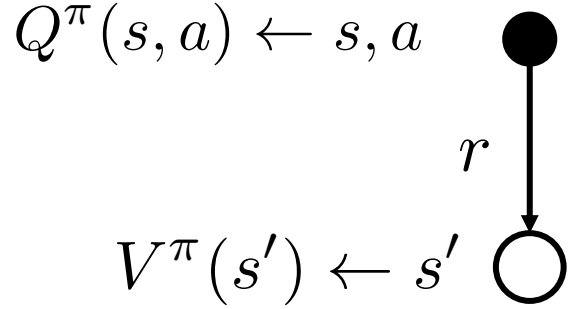
Bellman equation for $V^\pi(s)$



$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) Q^\pi(s,a)$$

$$V^\pi(s) = \sum_{a \in \mathcal{A}} \pi(a|s) \left(R(s,a) + \gamma V^\pi(s')\right)$$

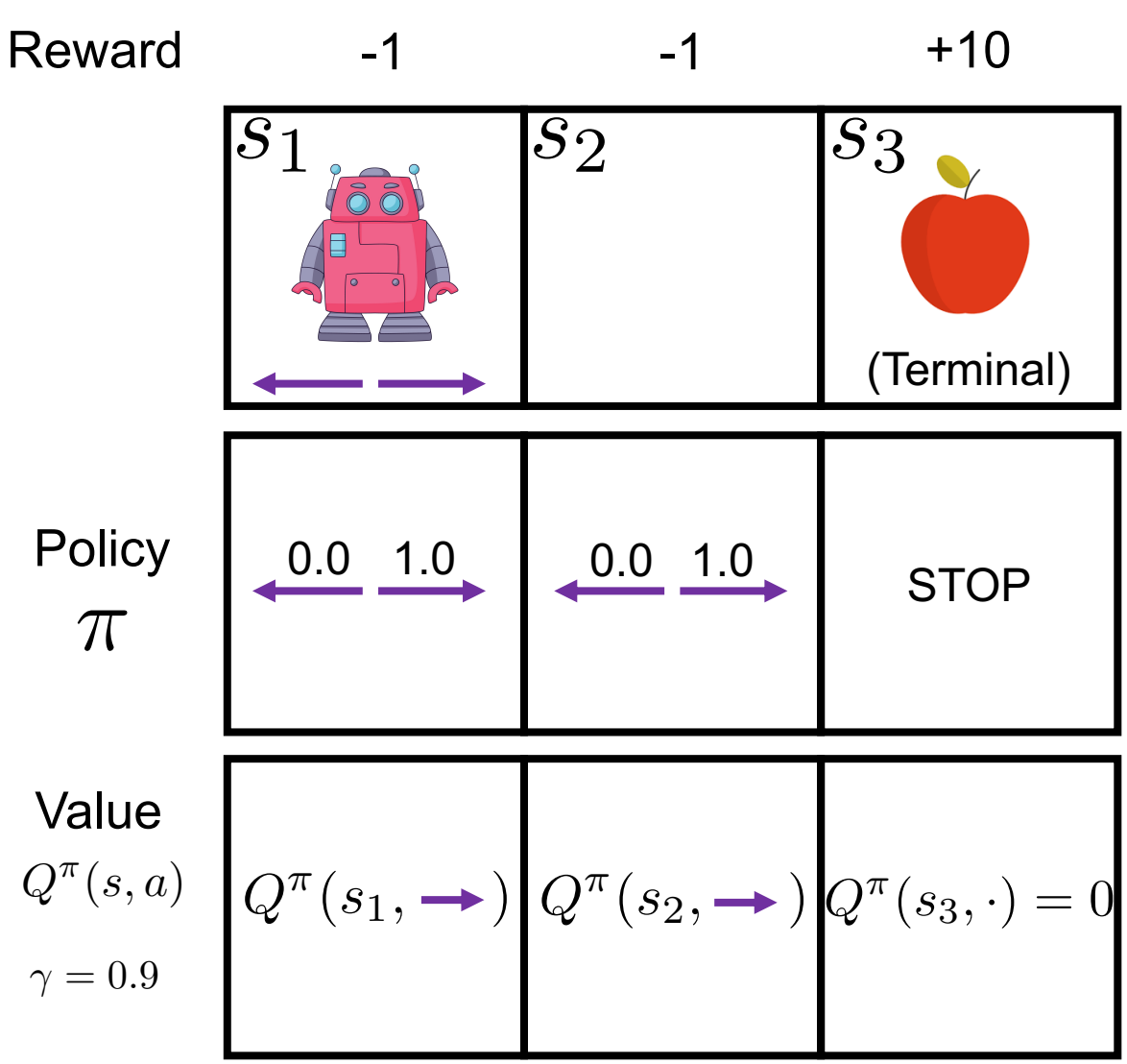
Bellman equation for $Q^\pi(s,a)$



$$Q^\pi(s,a) = R(s,a) + \gamma V^\pi(s')$$

Exercise for you after the lecture

A toy example of computing for value function



$$Q^\pi(s,a) = R(s,a) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') Q^\pi(s',a')$$

$$\begin{aligned} Q^\pi(s_2, \rightarrow) &= R(s_2, \rightarrow) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s_3) Q^\pi(s_3, a') \\ &= 10 + 0.9 \times 0 \\ &= 10 \end{aligned}$$

$$\begin{aligned} Q^\pi(s_1, \rightarrow) &= R(s_1, \rightarrow) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s_2) Q^\pi(s_2, a') \\ &= R(s_1, \rightarrow) + \gamma \left(\cancel{\pi(\leftarrow|s_2) Q^\pi(s_2, \leftarrow)} + \pi(\rightarrow|s_2) Q^\pi(s_2, \rightarrow)\right) \\ &= R(s_1, \rightarrow) + \gamma Q^\pi(s_2, \rightarrow) \\ &= -1 + 0.9 \times 10 \\ &= 8 \end{aligned}$$

Exercise for you after the lecture

What have we learned so far

- MDPs
- Policy
- How to evaluate any arbitrary policy using value functions
- Bellman equations

Immediate reward Discounted value of the next step

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') Q^{\pi}(s',a')$$

- However, the goal of solving an MDP is to find the optimal policy rather than evaluating any arbitrary policy
- Next, we will discuss how to find the optimal policy

Optimal value function

The optimal **state-value** function $V^*(s)$ is the maximum value function over all policies

$$V^*(s) = \max_\pi V^\pi(s)$$

The optimal **action-value** function $Q^*(s,a)$ is the maximum value function over all policies

$$Q^*(s,a) = \max_\pi Q^\pi(s,a)$$

Optimal value function → the best performance for an agent in the environment

Finding an optimal policy

An optimal policy can be found by maximizing **action-value** function

$$\pi^*(a|s) = \begin{cases} 1 & \text{if } a = \arg\max_{a \in \mathcal{A}} Q^*(s,a) \\ 0 & \text{otherwise} \end{cases}$$

$$a_1 = \text{argmax}_{a \in \mathcal{A}} Q^*(s,a)$$

$$\pi^*(a = a_1|s) = 1.0$$

$$Q^*(s,a)\,?$$

Bellman equation for the optimal value function

Bellman equation for $Q^{\pi}(s,a)$

$$Q^{\pi}(s,a) = R(s,a) + \gamma \sum_{a' \in \mathcal{A}} \pi(a'|s') Q^{\pi}(s',a')$$

Expectation over all possible actions
→ expectation value of an arbitrary policy

Bellman equation for $Q^{*}(s,a)$

$$Q^{*}(s,a) = R(s,a) + \gamma \max_{a' \in \mathcal{A}} Q^{*}(s',a')$$

Maximization over all possible actions
→ optimal value of all possible policies

How to solve the Bellman equation for $Q^*(s, a)$

$$Q^*(s, a) = R(s, a) + \gamma \max_{a' \in \mathcal{A}} Q^*(s', a')$$

$Q^*(s_t, a_t)$

$\rightarrow Q^*(s_{t+1}, a_{t+1})$

$\rightarrow Q^*(s_{t+2}, a_{t+2})$

$\rightarrow \ldots$

Many iterative methods:

- Policy iteration
- Value iteration
- Sarsa
- Q-learning

PhD thesis of Watkins (1989)

Chris Watkins

Foundation for more recent RL algorithms, e.g., learning to play Atari games

Model-based approaches

- Assume that we know the state transition probabilities (i.e., the world model)
- Policy improvement
- Policy iteration
- Value iteration

Policy improvement theorem

- V(s) tells us how good it is to following current policy from s. Can we do better with a new policy? I.e.,, take a different action at s.

$$q^{\pi}(s,a) = E[r_{t+1} + \gamma v^{\pi}(s_{t+1}) | s_t = s, a_t = a]$$
$$= \sum_{s',r} p(s', r|s, a)[r + \gamma v^{\pi}(s')]$$

- Policy improvement theorem:
- Given any pair of deterministic policies such that, for all states $s \in \mathcal{S}$

$$q^{\pi}(s, \pi'(s)) \geq v^{\pi}(s)$$

- Then the policy π' must be as good as, or better than, π. That is, it must obtain greater or equal expected return from all states $s \in \mathcal{S}$:

$$v^{\pi'}(s) \geq v^{\pi}(s)$$

Policy improvement

- Constructing a greedy policy, which meets the condition of the policy improvement theorem → as good as or better than the original policy

$$\begin{aligned}\pi'(s) &= \arg\max_a q^\pi(s,a) \\ &= \arg\max_a \sum_{s',r} p(s',r|s,a)[r+\gamma v^\pi(s')]\end{aligned}$$