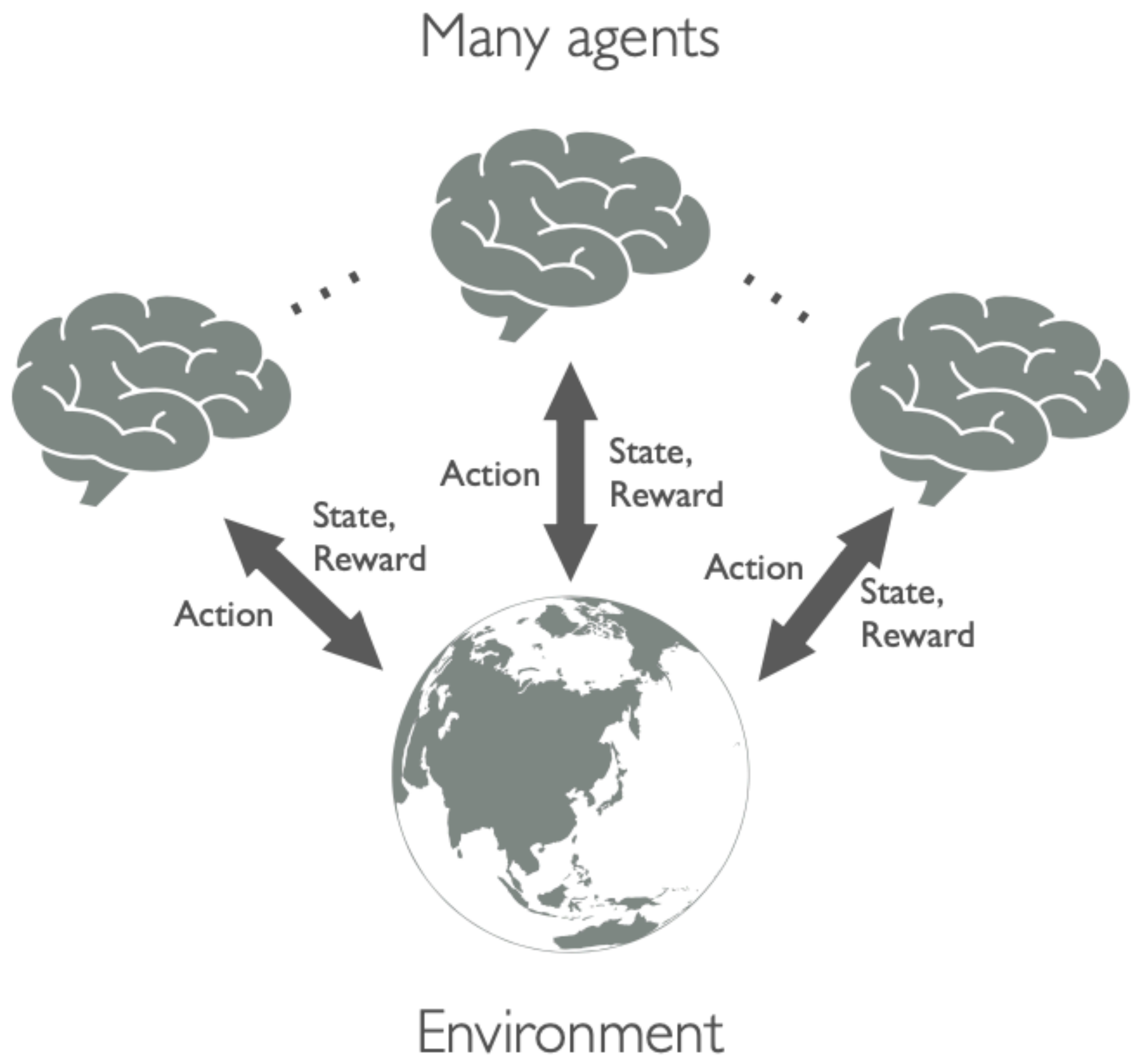
WISCONSIN
UNIVERSITY OF WISCONSIN–MADISON

Safe Multi-Agent Reinforcement Learning in Polynomial Time

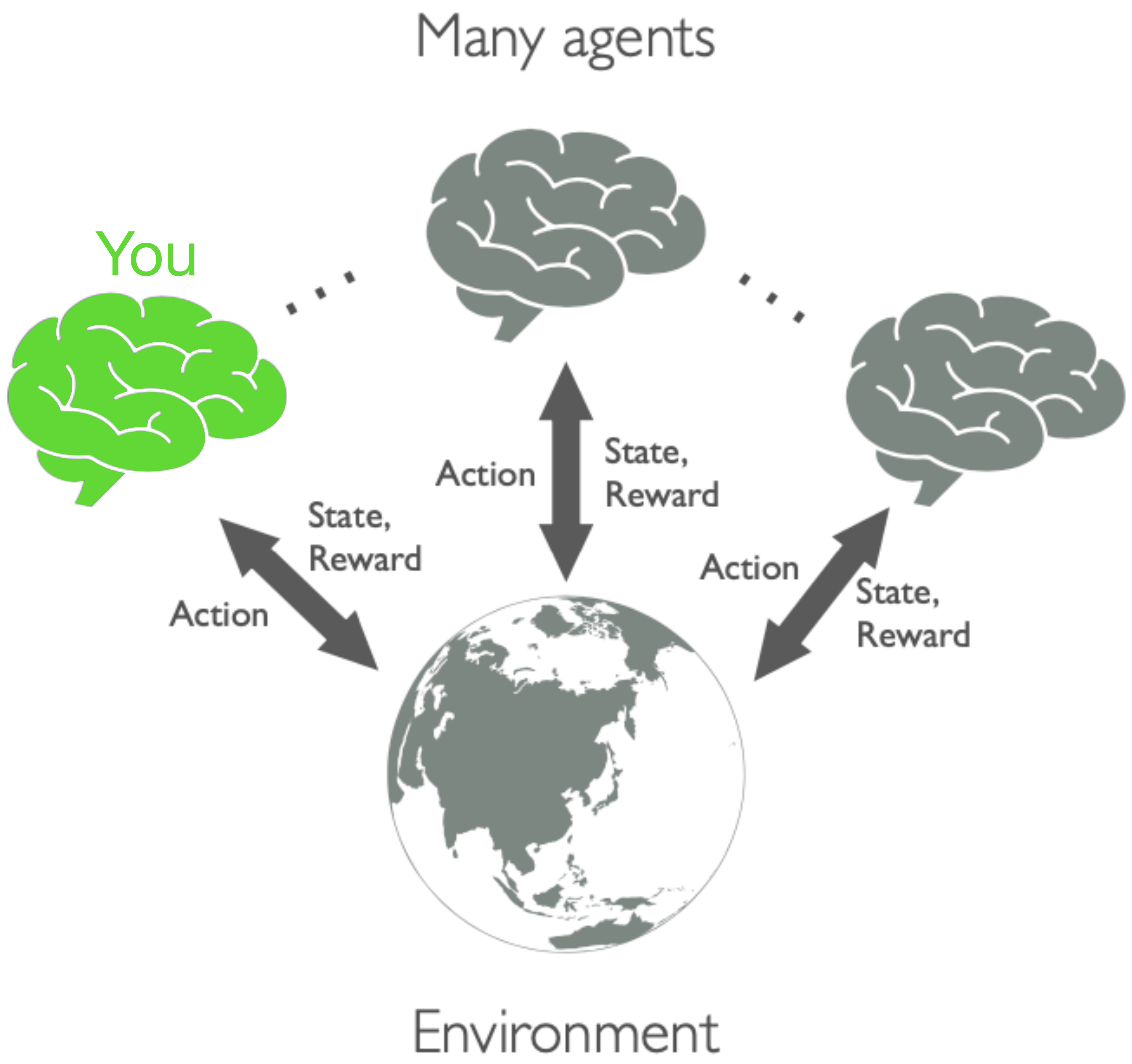
Jeremy McMahan

Safety Concerns

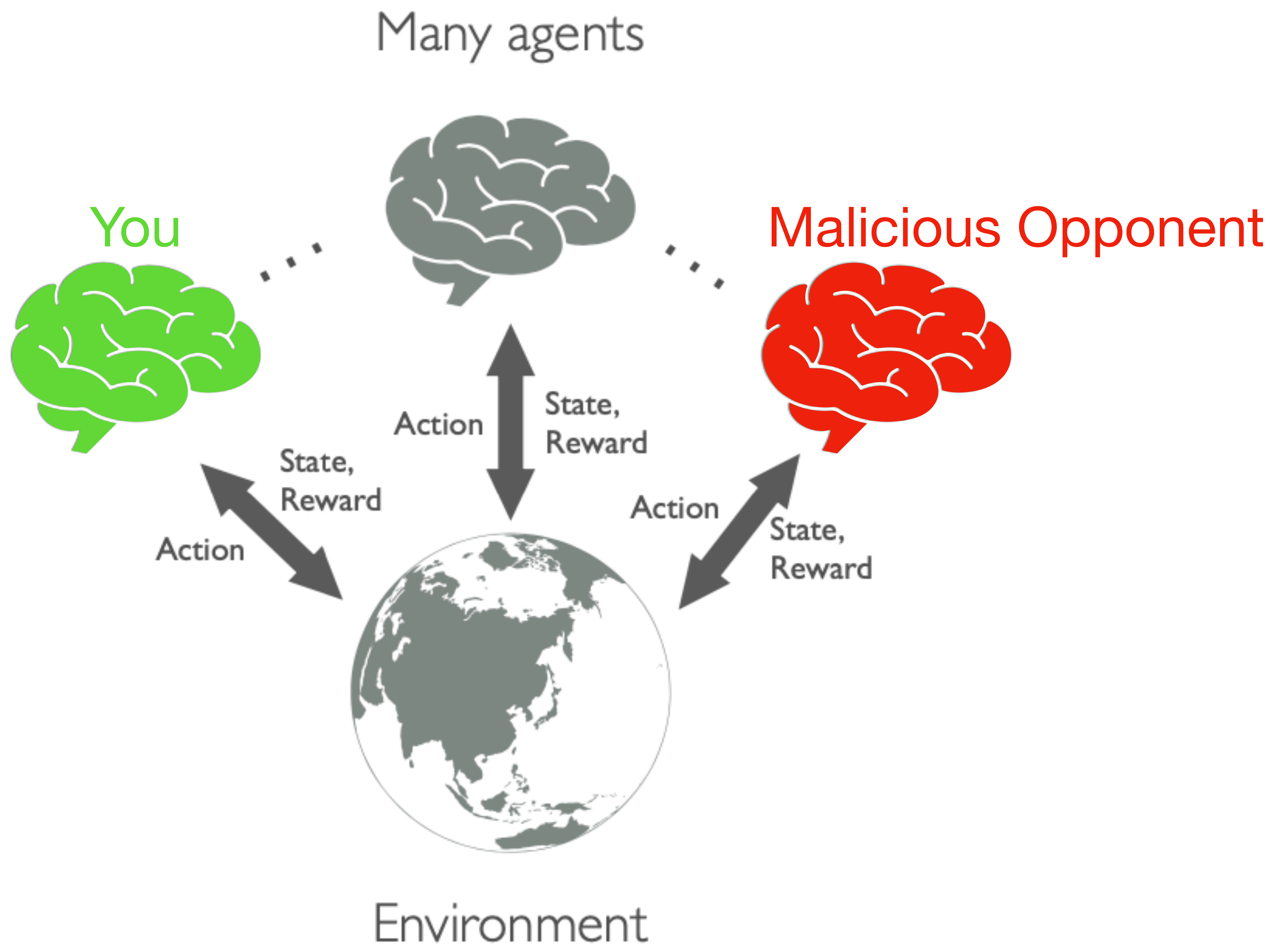
Safety Concerns



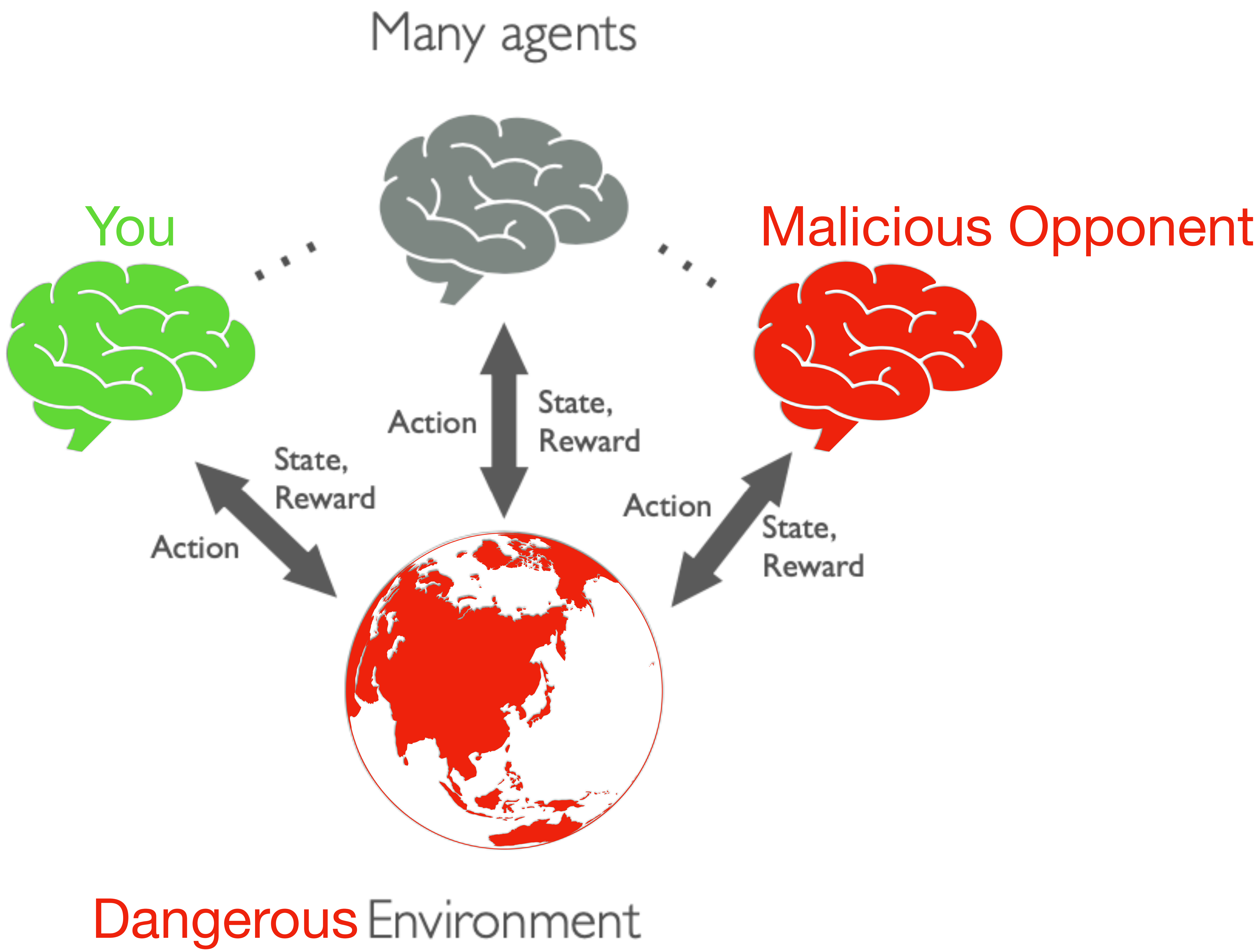
Safety Concerns



Safety Concerns



Safety Concerns



Safety Landscape

Safety Landscape

Safety from **Agents**:

Safety Landscape

Safety from **Agents**:

Adversarial MARL

Safety Landscape

Safety from **Agents**:
Adversarial MARL

Safety from **Environment**:

Safety Landscape

Safety from **Agents**:
Adversarial MARL

Safety from **Environment**:
Constrained MARL

Safety Landscape

Safety from **Agents**:
Adversarial MARL

Safety from **Environment**:
Constrained MARL

Safety Landscape

Safety from **Agents**:
Adversarial MARL

1. Manipulation Attacks

Safety from **Environment**:
Constrained MARL

Safety Landscape

Safety from **Agents**:
Adversarial MARL

1. Manipulation Attacks
2. Misinformation Attacks

Safety from **Environment**:
Constrained MARL

Safety Landscape

Safety from **Agents**:
Adversarial MARL

1. Manipulation Attacks
2. Misinformation Attacks

Safety from **Environment**:
Constrained MARL

1. Anytime Constraints

Safety Landscape

Safety from **Agents**:
Adversarial MARL

1. Manipulation Attacks
2. Misinformation Attacks

Safety from **Environment**:
Constrained MARL

1. Anytime Constraints
2. Single-Constraint FPTAS

Safety Landscape

Safety from **Agents**:
Adversarial MARL

1. Manipulation Attacks
2. Misinformation Attacks

Safety from **Environment**:
Constrained MARL

1. Anytime Constraints
2. Single-Constraint FPTAS
3. Multi-Constraint Bicriteria

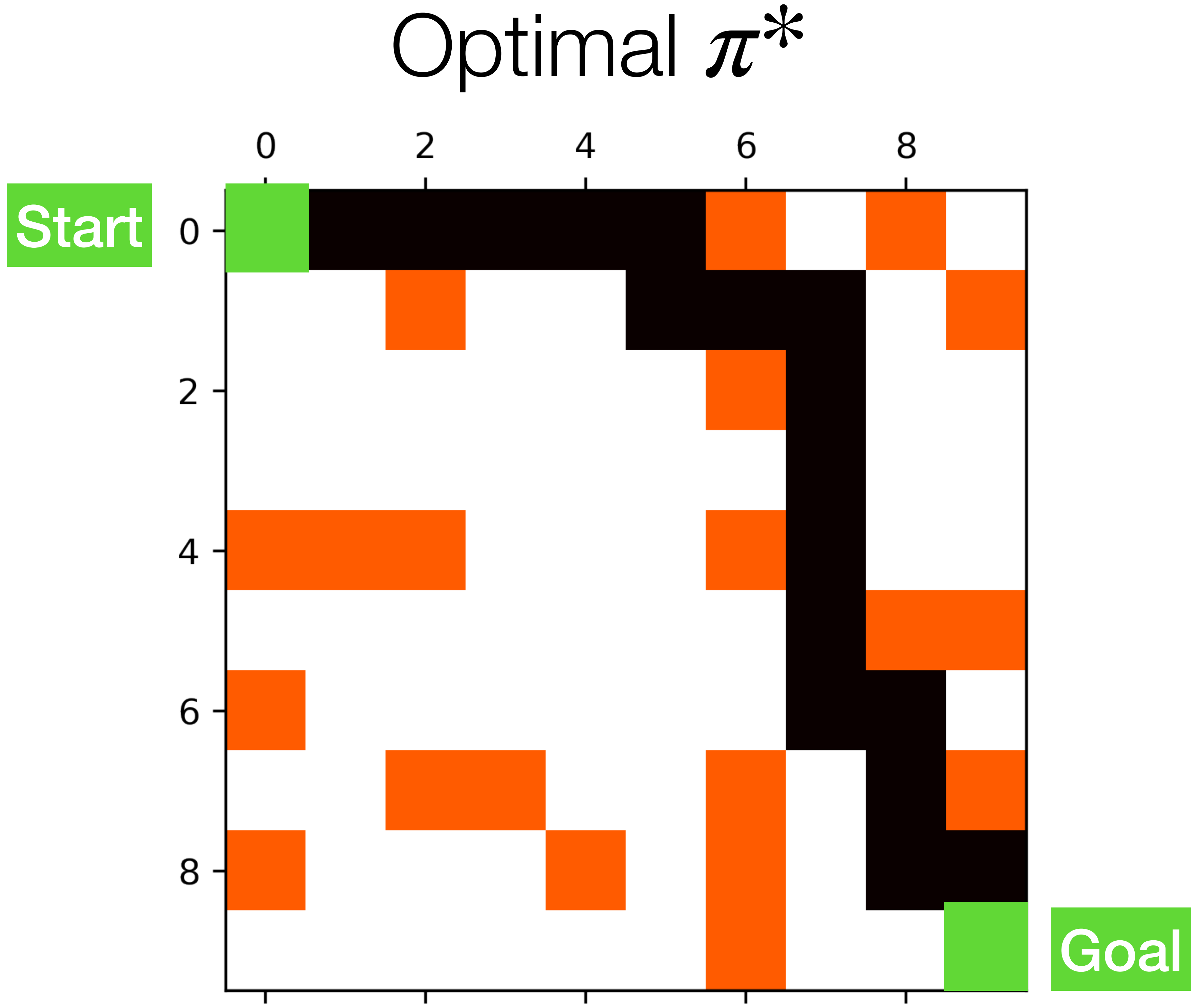
Adversarial MARL

Manipulation Attacks

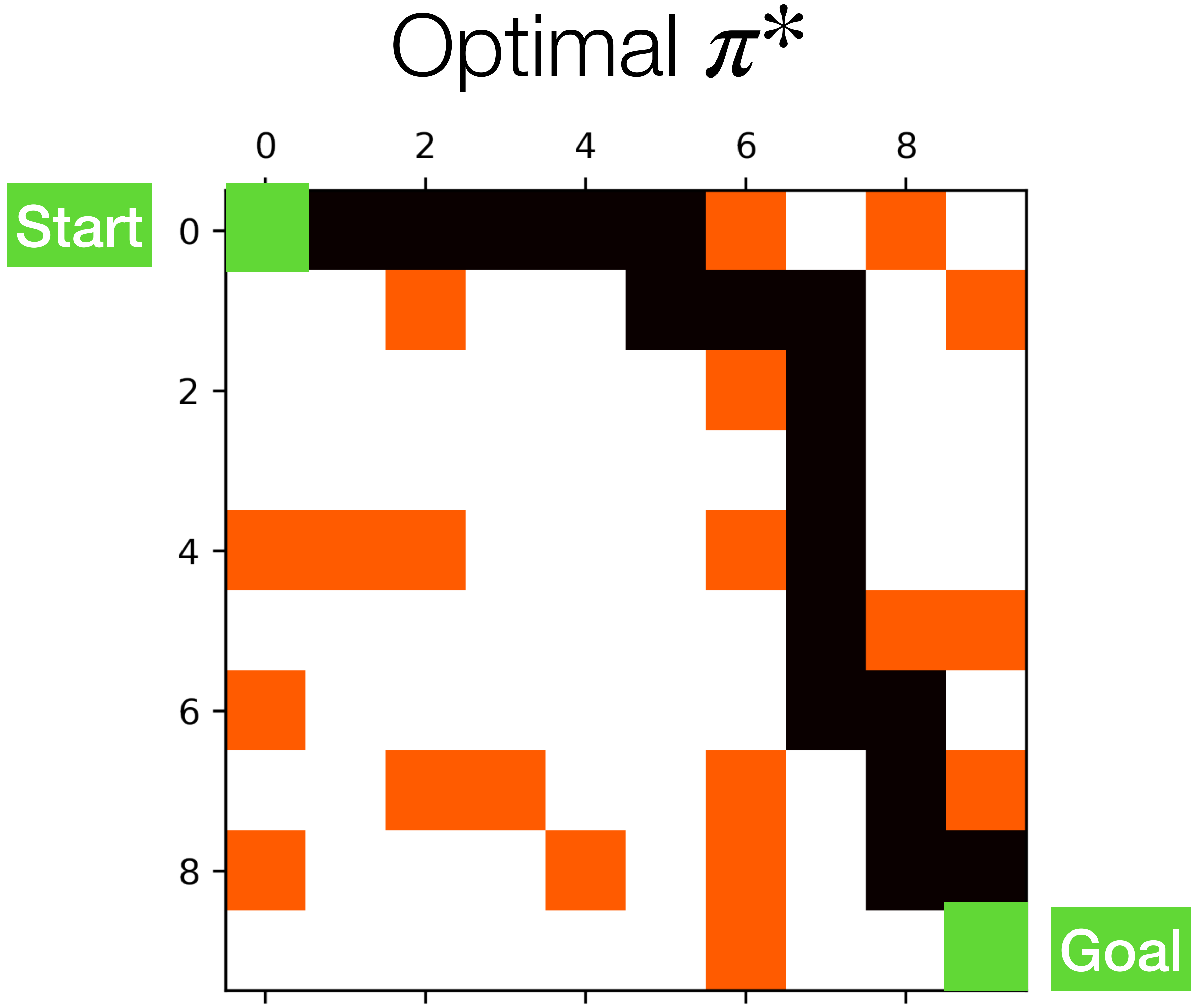
**AAAI 2024*

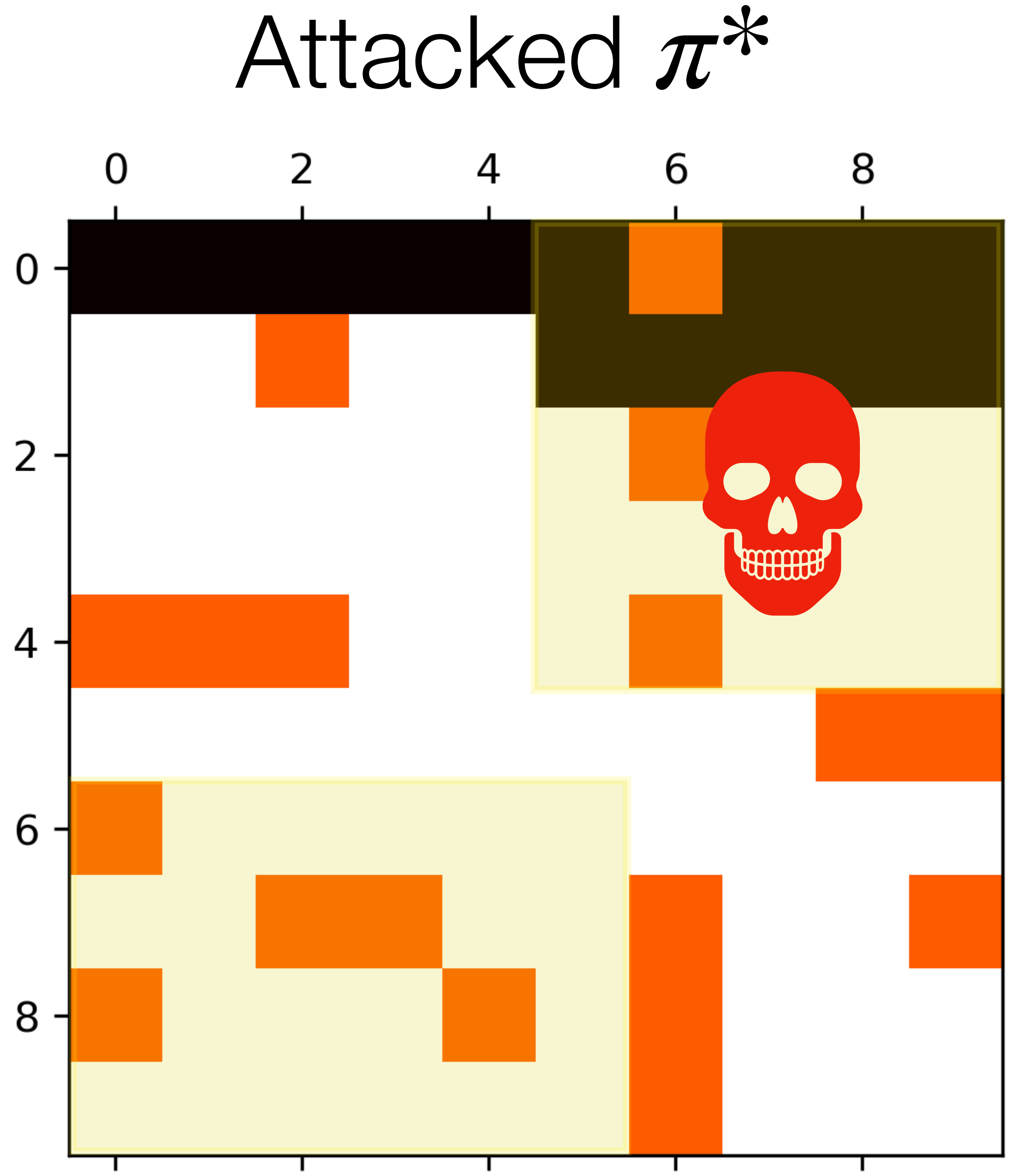
Motivation

Motivation

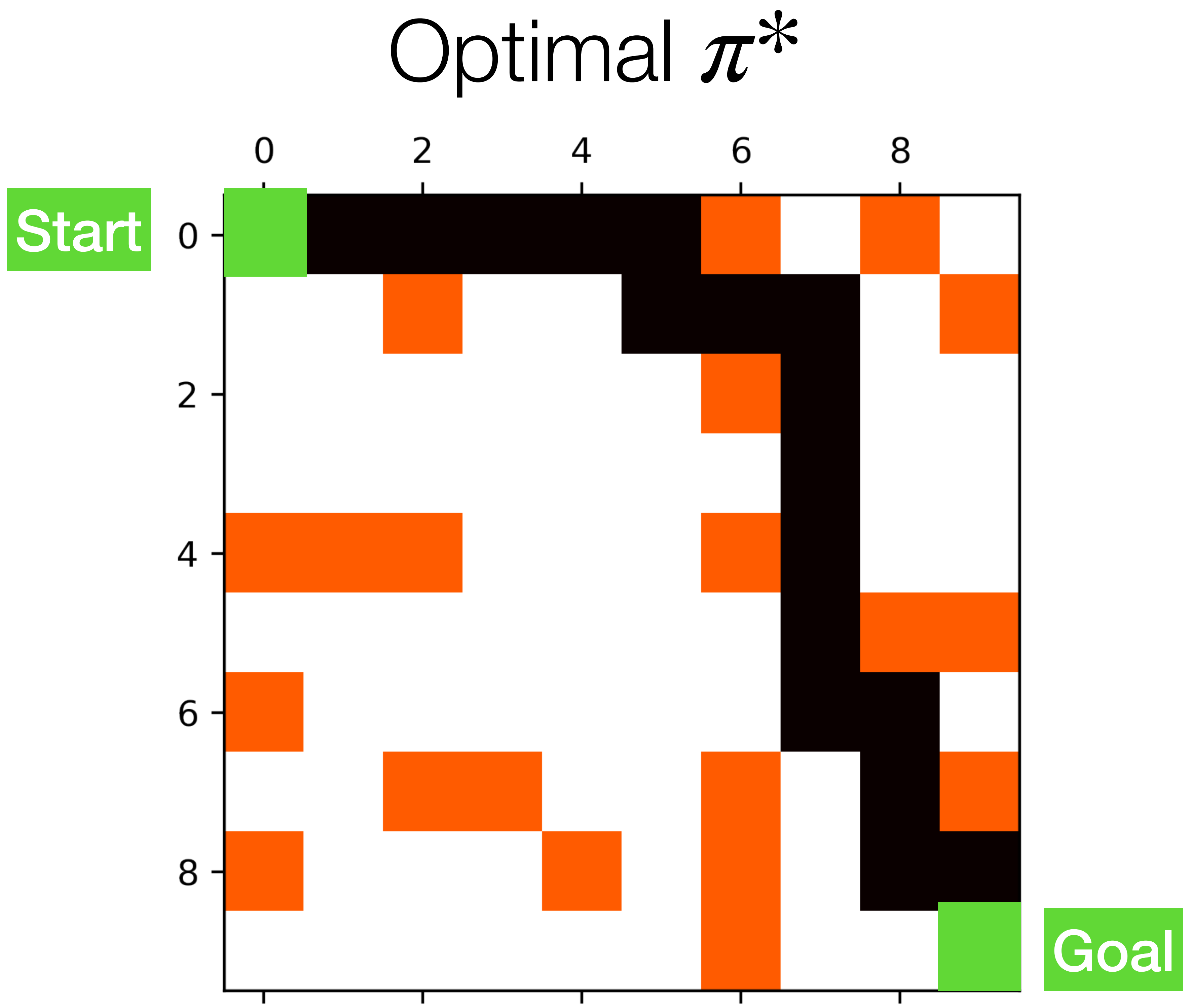


Motivation



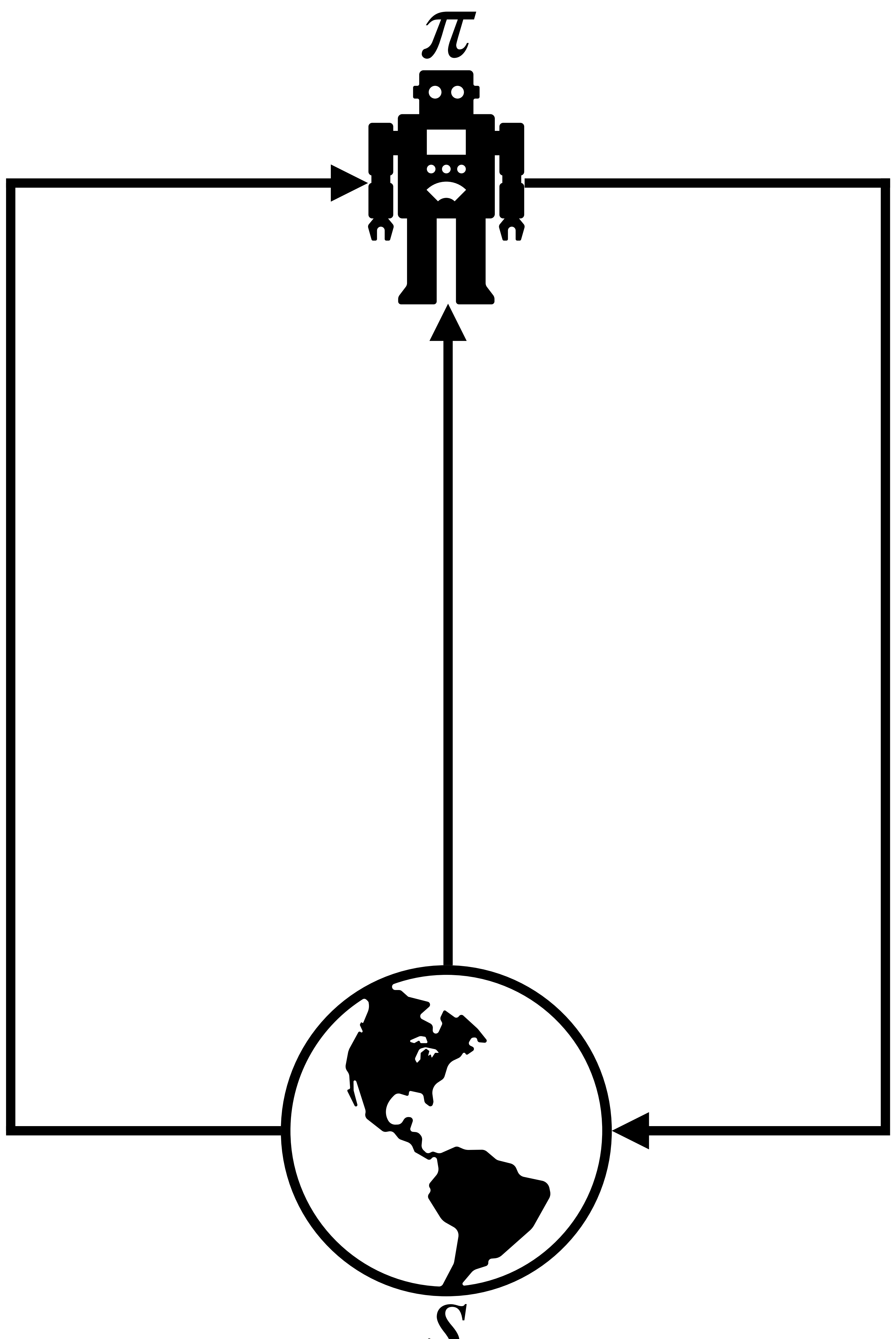


Motivation

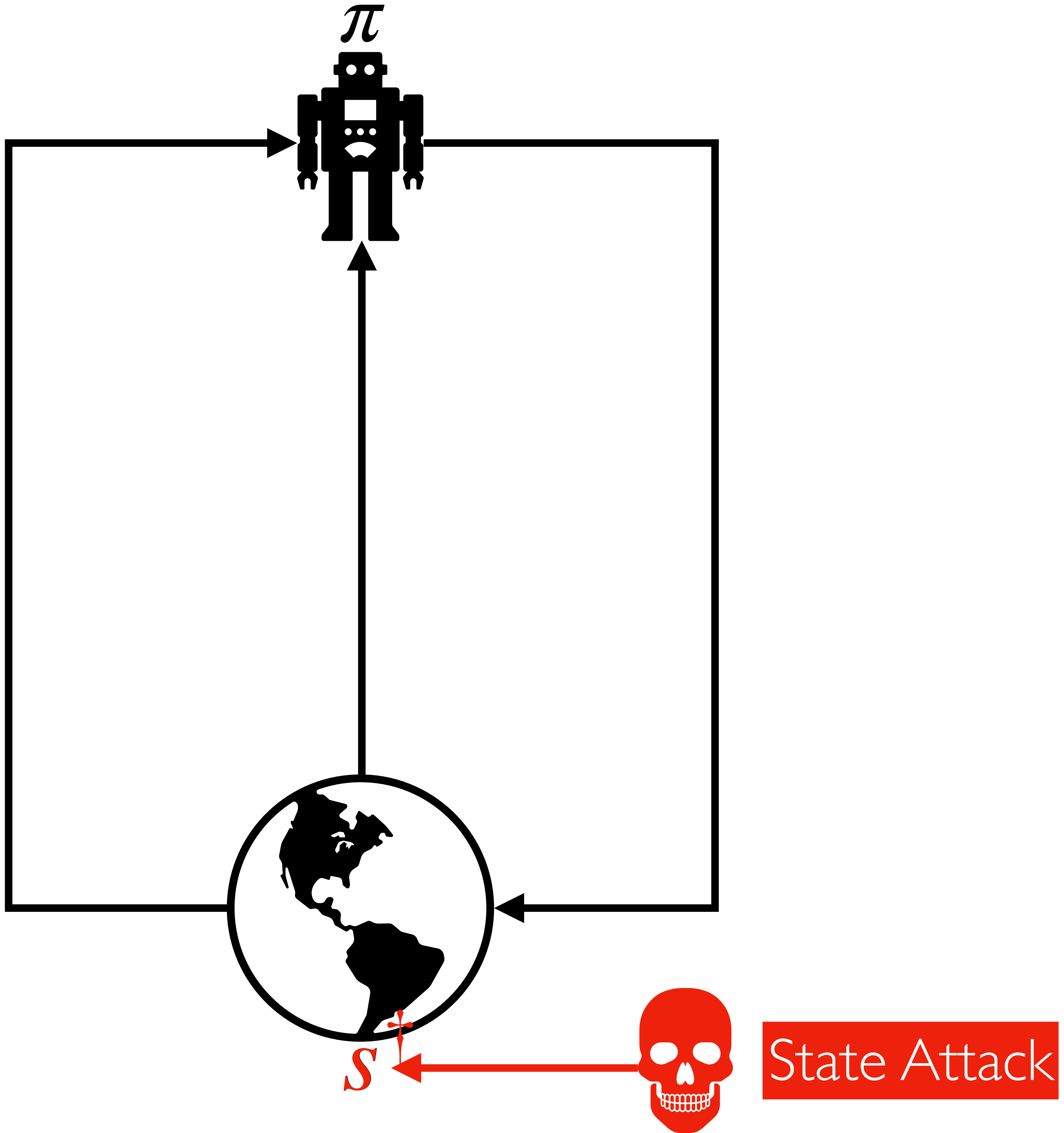


Attack Surfaces

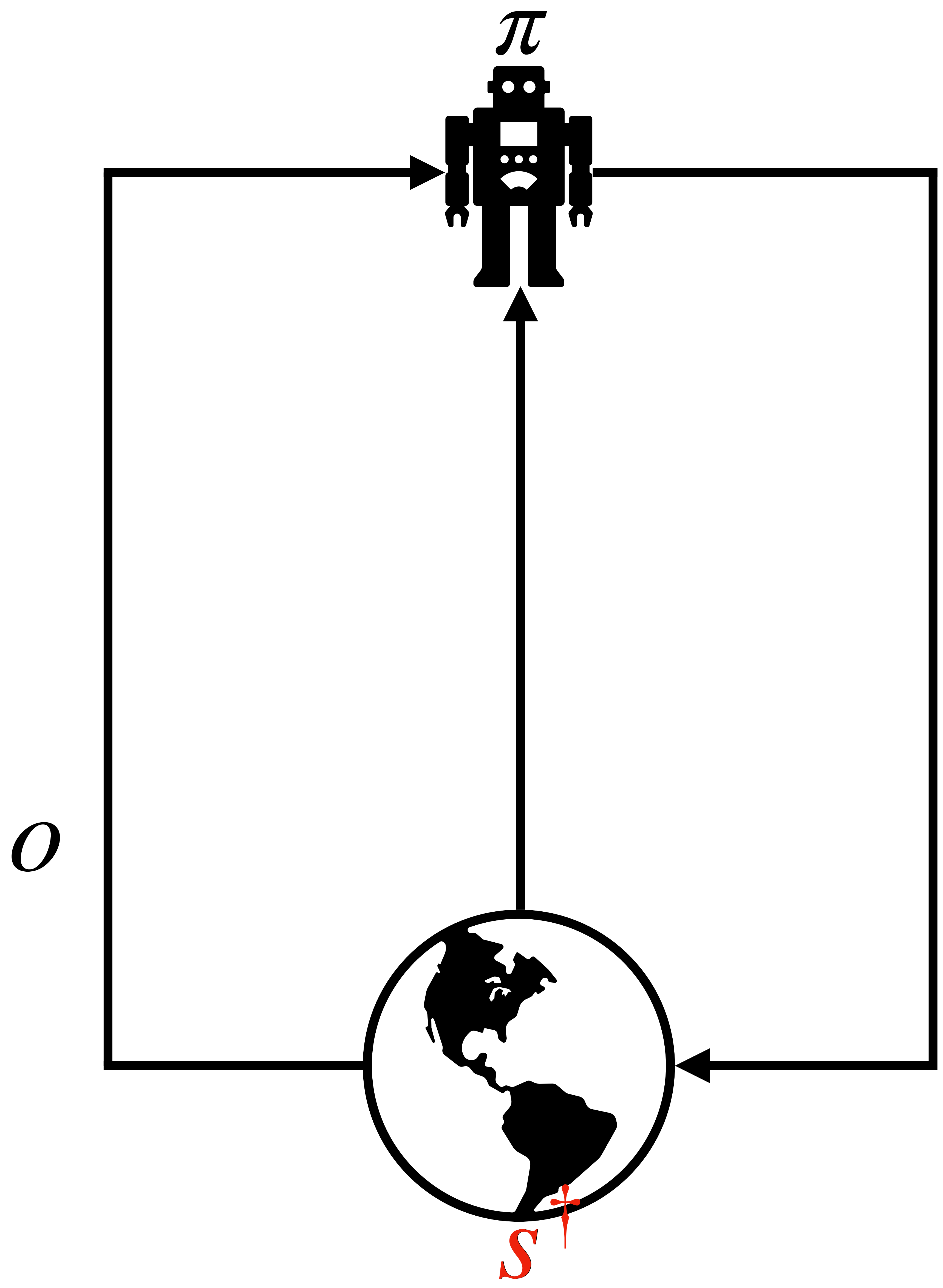
Attack Surfaces



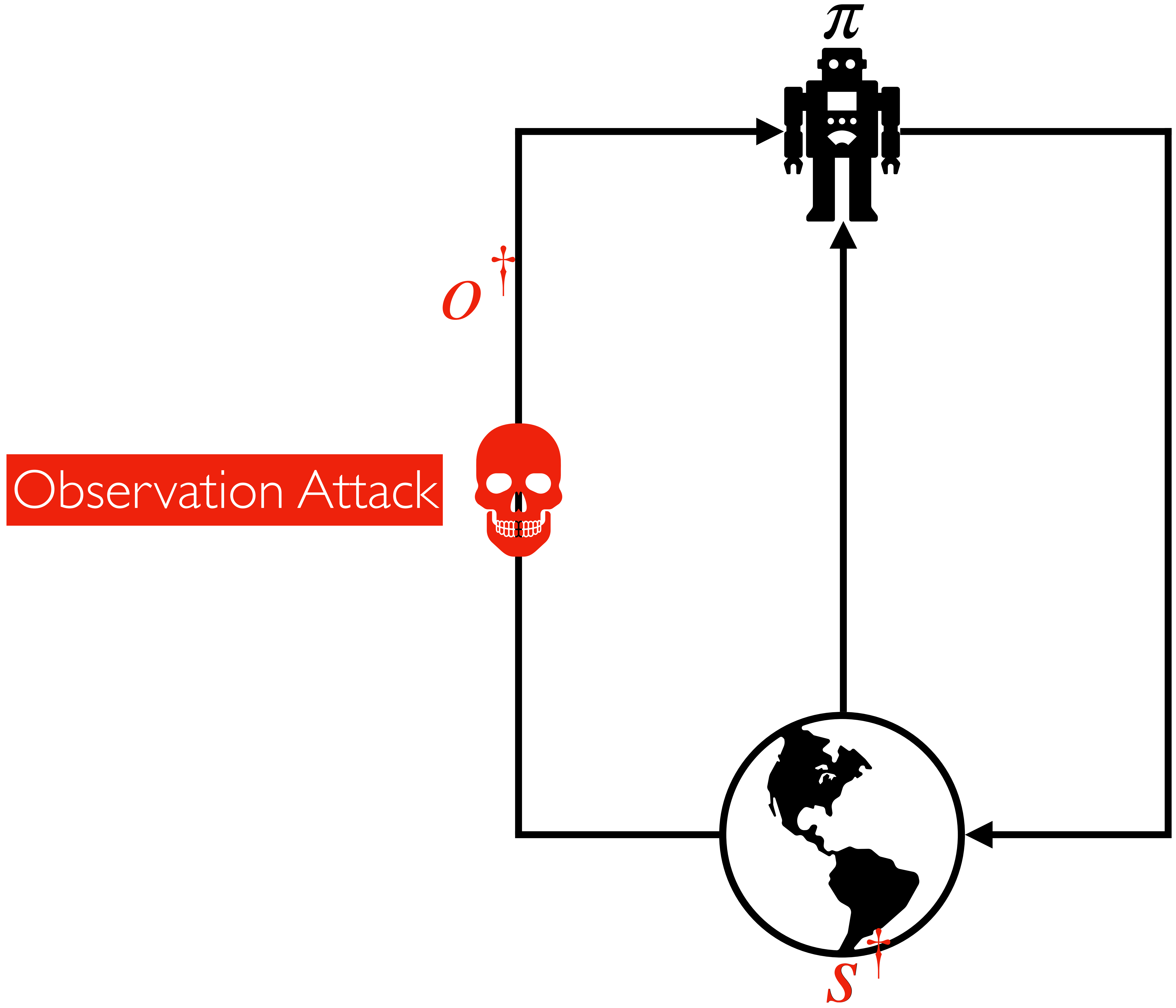
Attack Surfaces



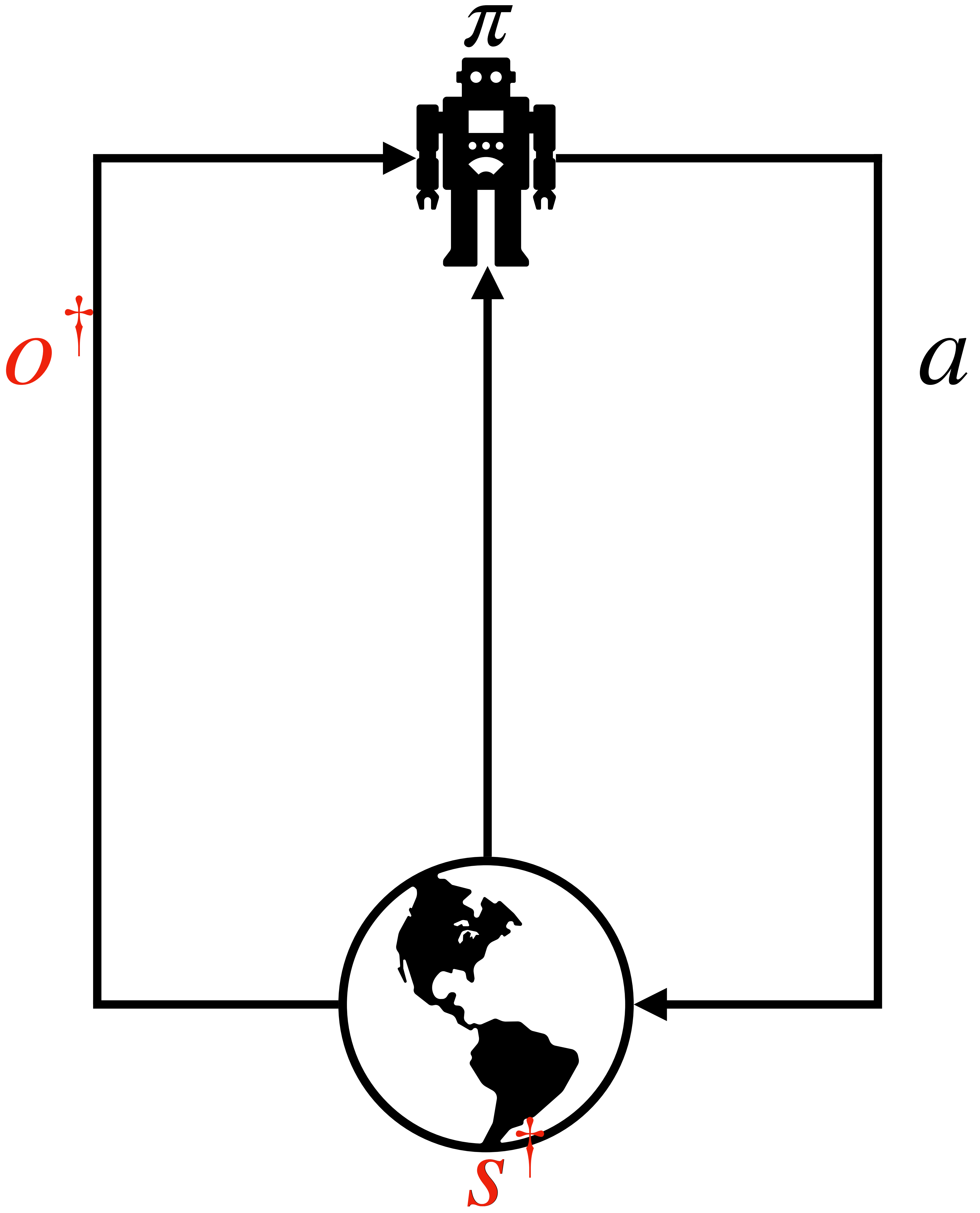
Attack Surfaces



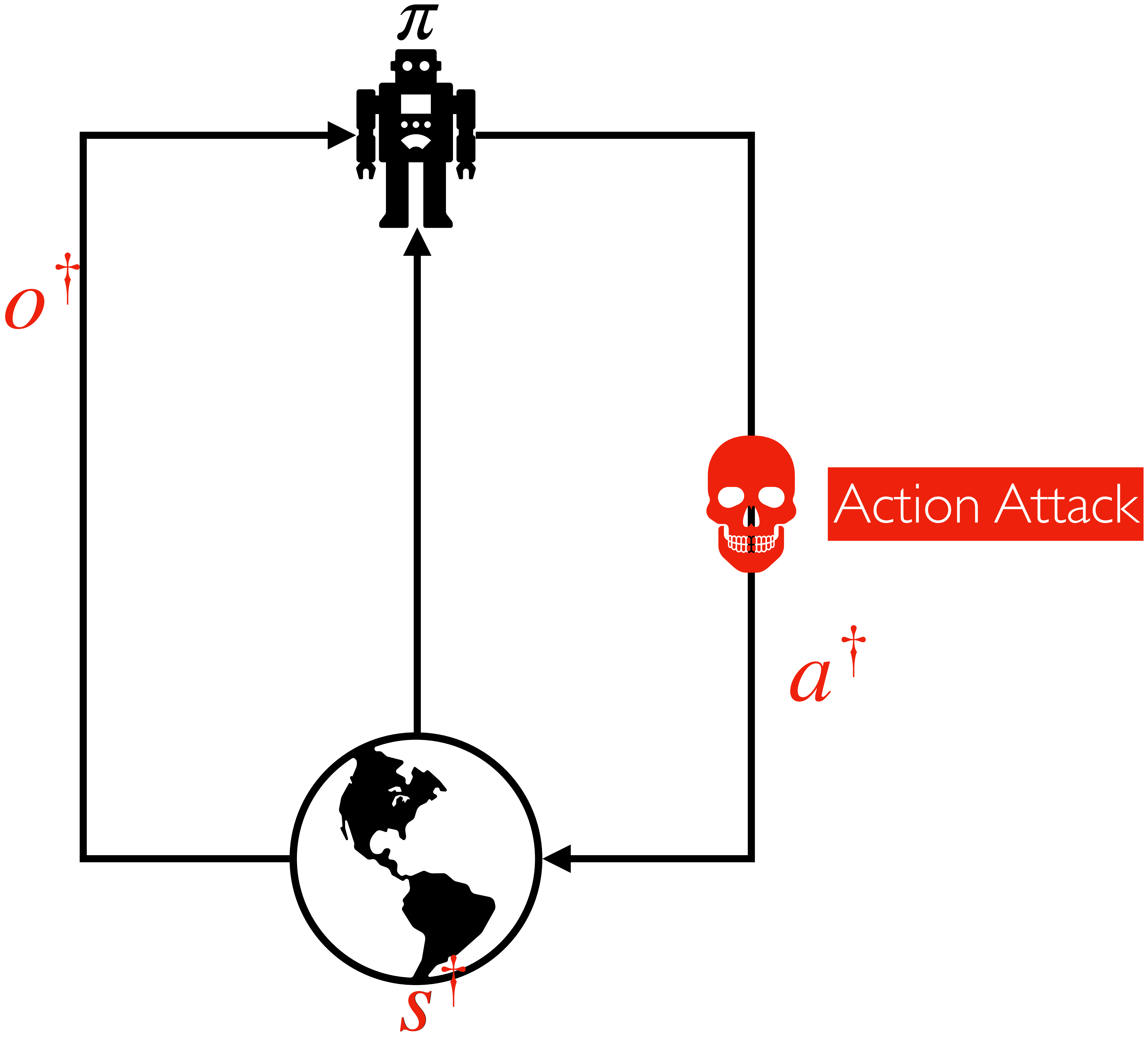
Attack Surfaces



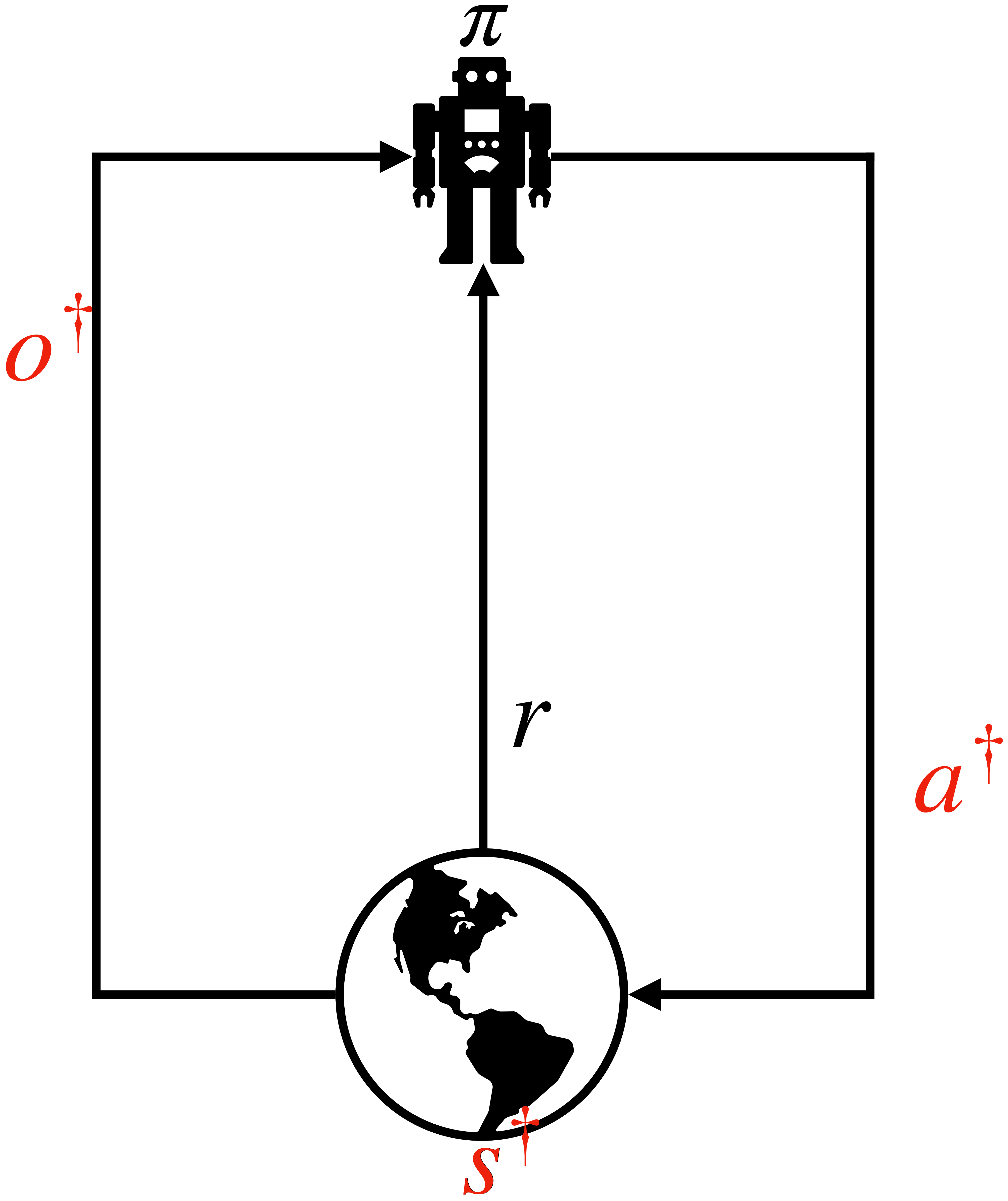
Attack Surfaces



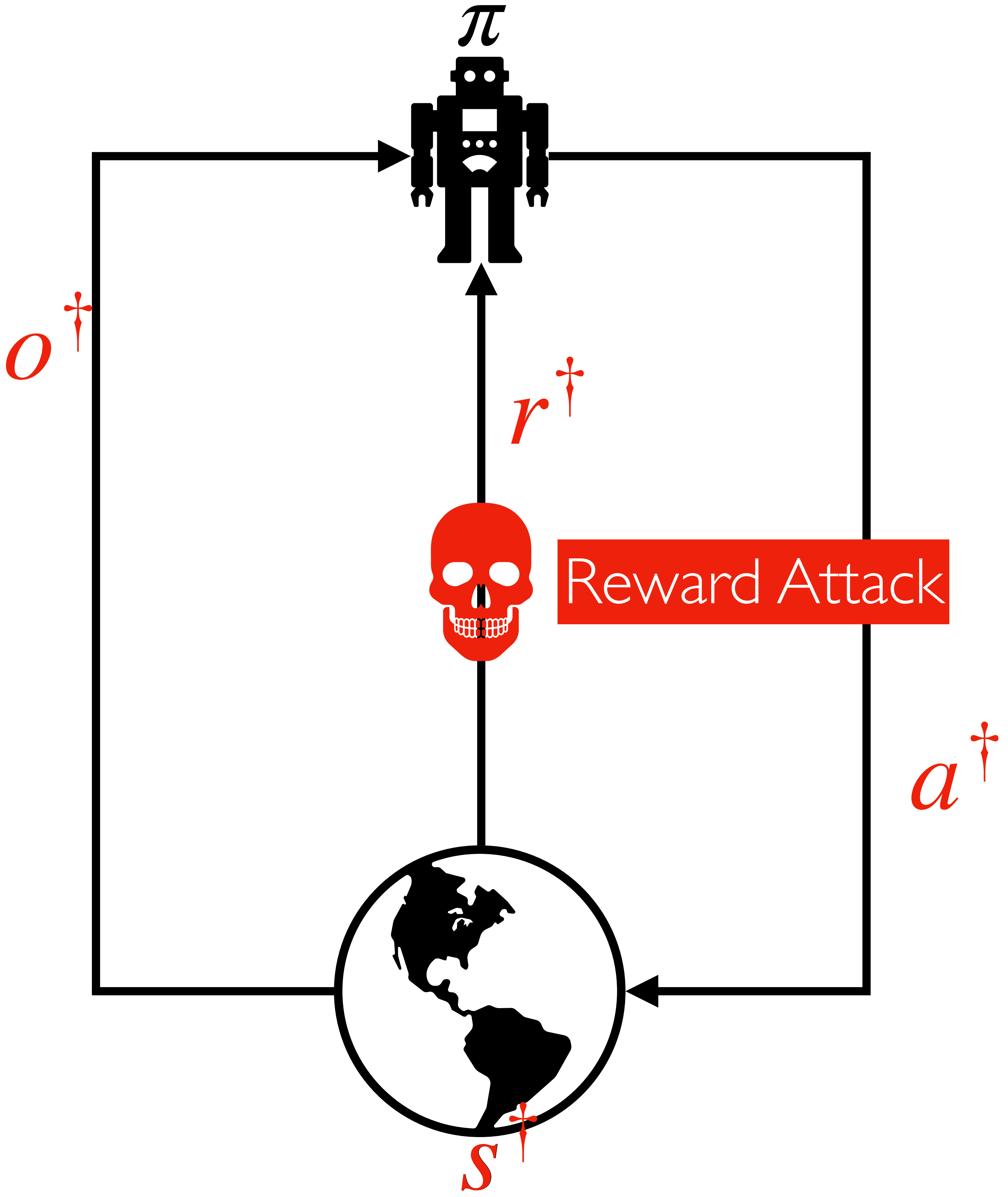
Attack Surfaces



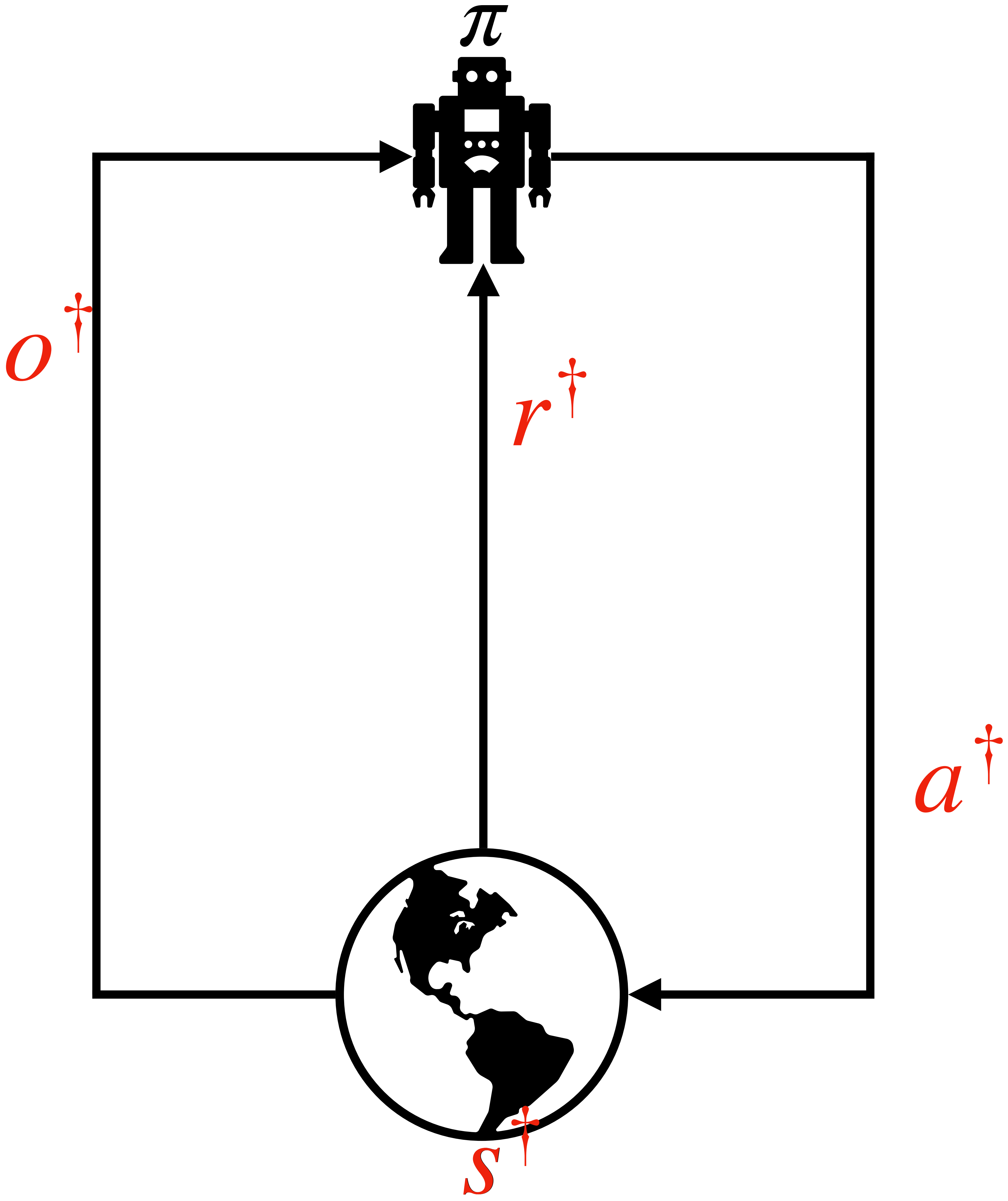
Attack Surfaces



Attack Surfaces



Attack Surfaces



Attacker's Perspective

Attacker's Perspective

Attacker has its own reward $g(s_t, a_t, r_t)$ that depends on the victim's.

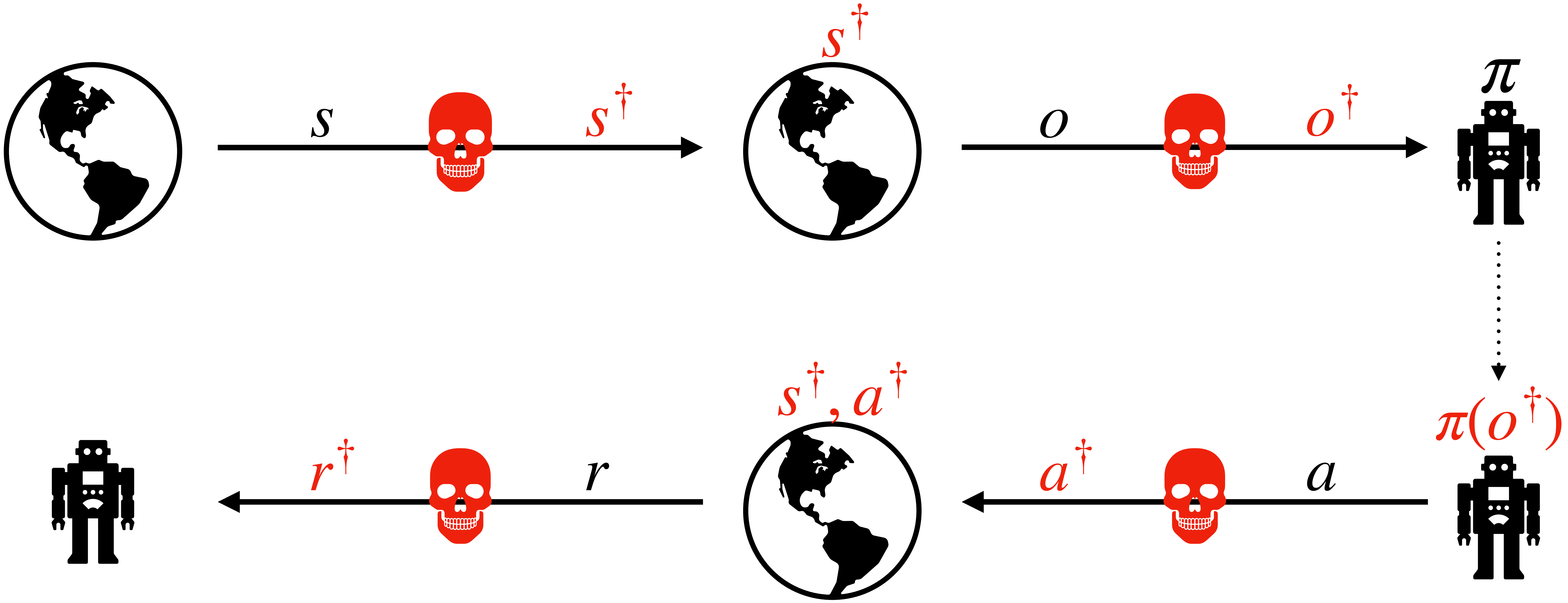
Attacker's Perspective

Attacker has its own reward $g(s_t, a_t, r_t)$ that depends on the victim's.

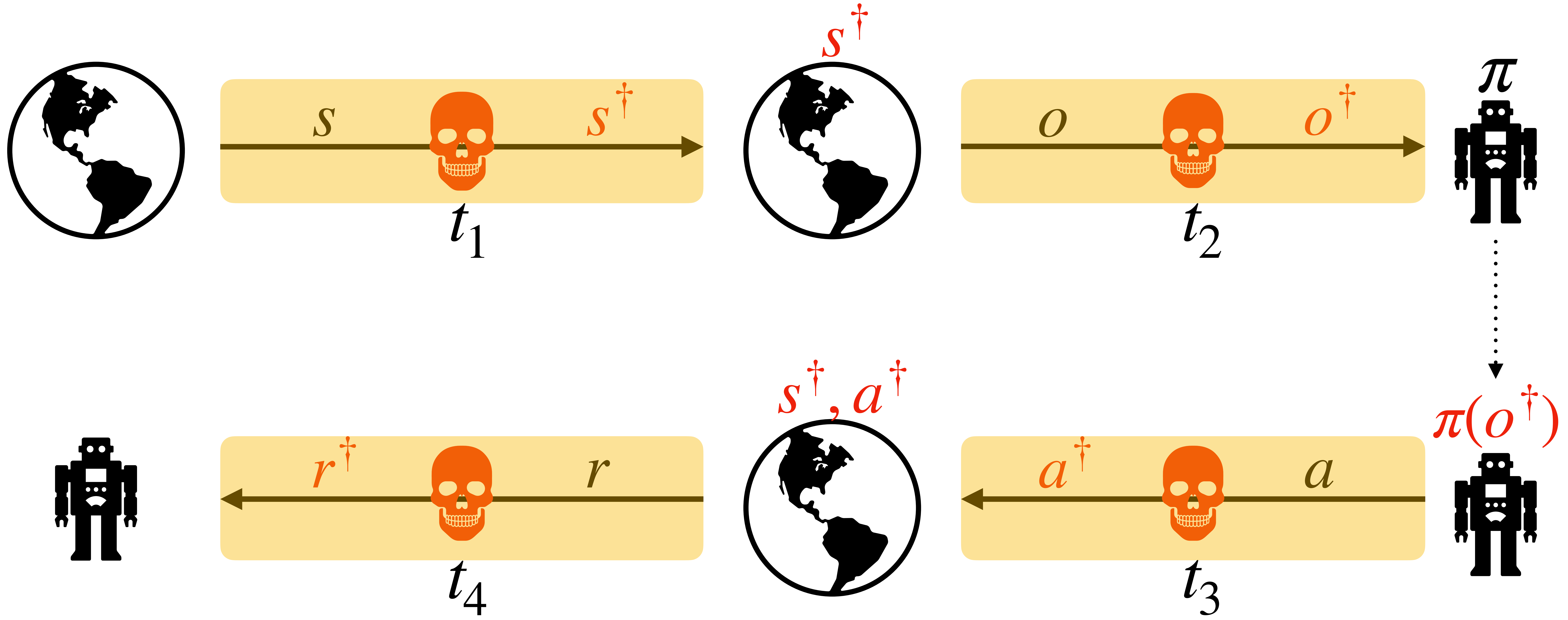
Definition 1 (Attack Problem)**.** For any π, the attacker's seeks a policy $\nu^* \in N$ that maximizes its expected reward from the victim-attacker-M interaction:

$$\nu^* \in \arg\max_{\nu \in N} \mathbb{E}_M^{\pi,\nu} \left[\sum_{t=0}^{\infty} \gamma^t g(s_t, a_t, r_t) \right].$$

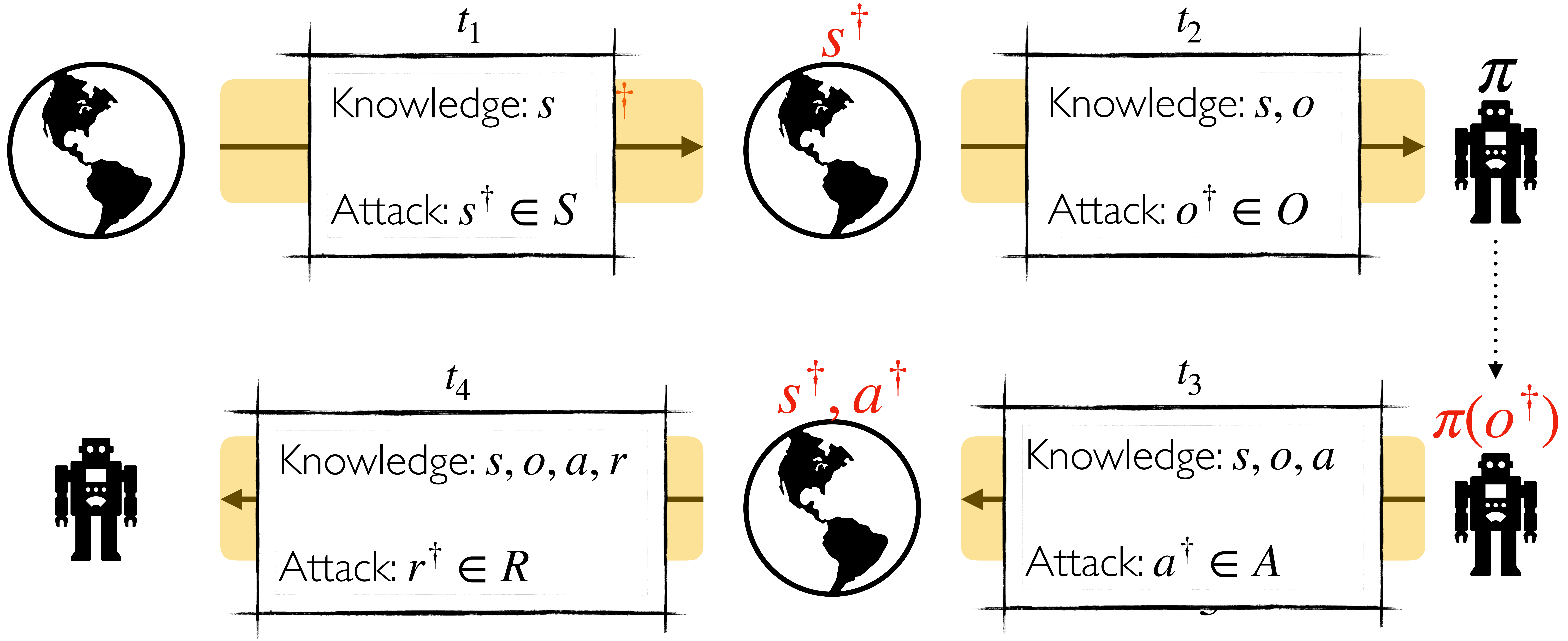
Adversarial Decomposition



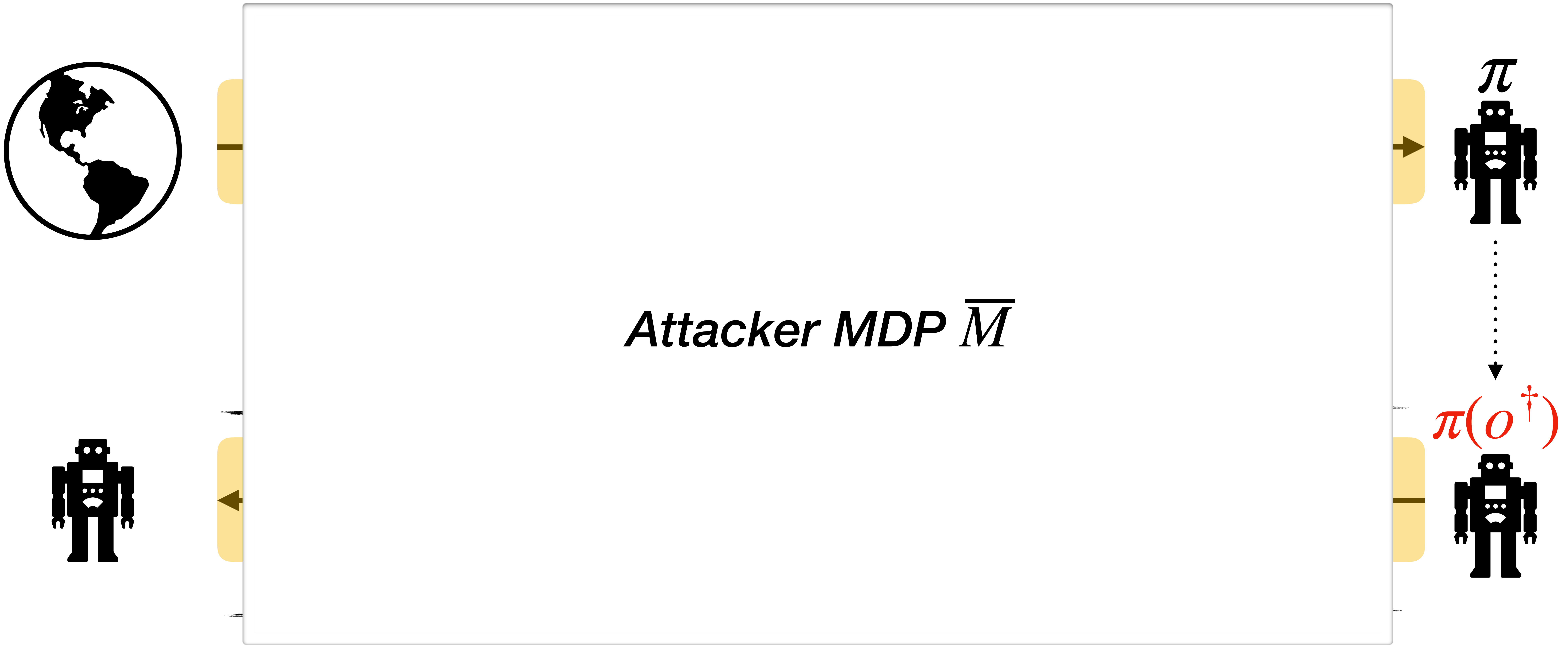
Adversarial Decomposition



Adversarial Decomposition



Adversarial Decomposition



Attack Results

Attack Results

Theorem: *An optimal attack involving any combination of attack surfaces can be computed in time poly*$(|M|, |\pi|)$.

Attack Results

Theorem: *An optimal attack involving any combination of attack surfaces can be computed in time poly*$(|M|, |\pi|)$.

First results beyond observation attacks!

The Defense Problem

The Defense Problem

Let $(V_1^{\pi,\nu}, V_2^{\pi,\nu})$ denote the victim's and attacker's value, respectively.

The Defense Problem

Let $(V_1^{\pi,\nu}, V_2^{\pi,\nu})$ denote the victim's and attacker's value, respectively.

Definition 2 (Defense Problem)**.** The victim seeks a policy π^* that maximizes its expected reward from the victim-attacker-M interaction under the worst-case attack:

$$\pi^* \in \arg\max_{\pi \in \Pi} \min_{\nu \in BR(\pi)} V_1^{\pi,\nu}.$$

The Defense Problem

Let $(V_1^{\pi,\nu}, V_2^{\pi,\nu})$ denote the victim's and attacker's value, respectively.

Definition 2 (Defense Problem)**.** The victim seeks a policy π^* that maximizes its expected reward from the victim-attacker-M interaction under the worst-case attack:

$$\pi^* \in \arg\max_{\pi \in \Pi} \min_{\nu \in BR(\pi)} V_1^{\pi,\nu}.$$

$$BR(\pi) := \arg\max_{\nu \in N} V_2^{\pi,\nu}$$

The Defense Problem

Let $(V_1^{\pi,\nu}, V_2^{\pi,\nu})$ denote the victim's and attacker's value, respectively.

Definition 2 (Defense Problem). The victim seeks a policy π^* that maximizes its expected reward from the victim-attacker-M interaction under the worst-case attack:

$$\pi^* \in \arg\max_{\pi \in \Pi} \min_{\nu \in BR(\pi)} V_1^{\pi,\nu}.$$

$$BR(\pi) := \arg\max_{\nu \in N} V_2^{\pi,\nu}$$

Defense = WSE in a meta game.

Bottlenecks

Bottlenecks

- WSE need not exist.

Bottlenecks

- WSE need not exist.
- WSE are generally non-Markovian.

Bottlenecks

- WSE need not exist.
- WSE are generally non-Markovian.

Proposition: *The defense problem is as hard as solving POMDPs. Thus, is NP-hard to even approximate.*

Approach

Approach

***Solution**: ban observation attacks.*

Approach

***Solution**: ban observation attacks.*

$\overline{G}$

Approach

***Solution**: ban observation attacks.*

$\overline{G}$

Zero-sum:

Approach

***Solution**: ban observation attacks.*

$\overline{G}$

Zero-sum:

WSE → MPNE

Approach

***Solution**: ban observation attacks.*

$\overline{G}$

Zero-sum:

WSE → MPNE

General-sum:

Approach

***Solution**: ban observation attacks.*

$\overline{G}$

Zero-sum:

WSE → MPNE

General-sum:

WSE → Mutual Recursion

Rollback Algorithm

Special Case: Action Attacks

Rollback Algorithm

Special Case: Action Attacks

1. Victim determines Attacker's best response to any action a:

$$BR_h(s,a) = \arg\max_{a^\dagger \in \overline{\mathcal{A}}(s,a)} \left[g_h(s,a,r_h(s,a)) + \mathbb{E}_{s' \sim P_h(s,a^\dagger)} \left[V^*_{h+1,2}(s', \pi^*_{h+1}(s')) \right] \right]$$

Rollback Algorithm

Special Case: Action Attacks

1. Victim determines Attacker's best response to any action a:

$$BR_h(s,a) = \arg\max_{a^\dagger \in \overline{\mathcal{A}}(s,a)} \left[g_h(s,a,r_h(s,a)) + \mathbb{E}_{s' \sim P_h(s,a^\dagger)}\left[V^*_{h+1,2}(s', \pi^*_{h+1}(s'))\right]\right]$$

2. Victim picks a based on the worst-case best-response:

$$V^*_{h,1}(s) = \max_{a \in \mathcal{A}} \min_{a^\dagger \in BR_h(s,a)} \left[r_h(s,a^\dagger) + \mathbb{E}_{s' \sim P_h(s,a^\dagger)}\left[V^*_{h+1,1}(s')\right]\right]$$

Defense Results

Defense Results

Theorem: *An optimal defense can be* ***computed*** *as the WSE of a meta game (POTBMG).*

Defense Results

Theorem: *An optimal defense can be* ***computed*** *as the WSE of a meta game (POTBMG).*

Moreover, the defense is computable in ***polynomial time*** *if observation attacks are banned.*

Defense Results

Theorem: *An optimal defense can be* ***computed*** *as the WSE of a meta game (POTBMG).*

Moreover, the defense is computable in ***polynomial time*** *if observation attacks are banned.*

First results for the general defense problem!

Misinformation Attacks

**RLC 2024*

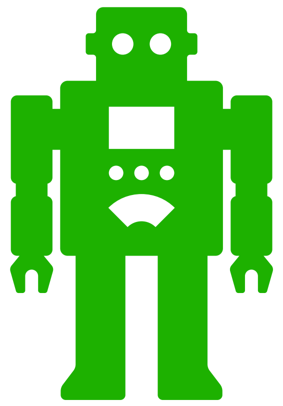
Motivation

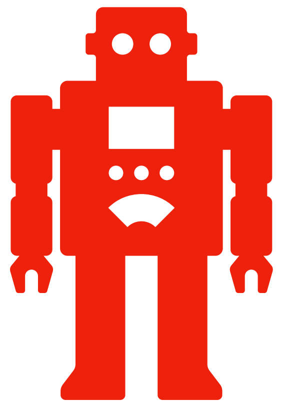
Motivation

More **realistic** attacker: information advantage instead of environment control

Motivation

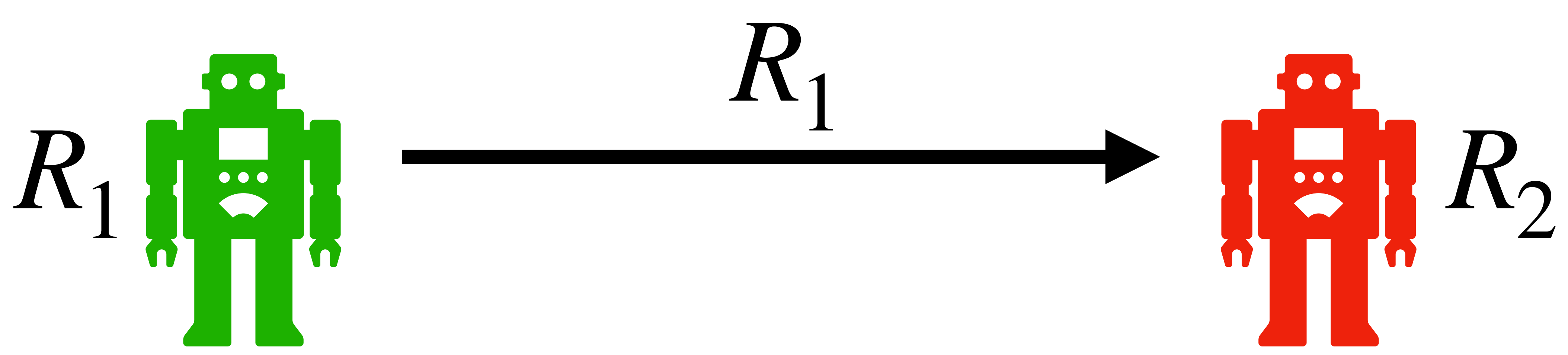
More **realistic** attacker: information advantage instead of environment control

R_1 

 R_2

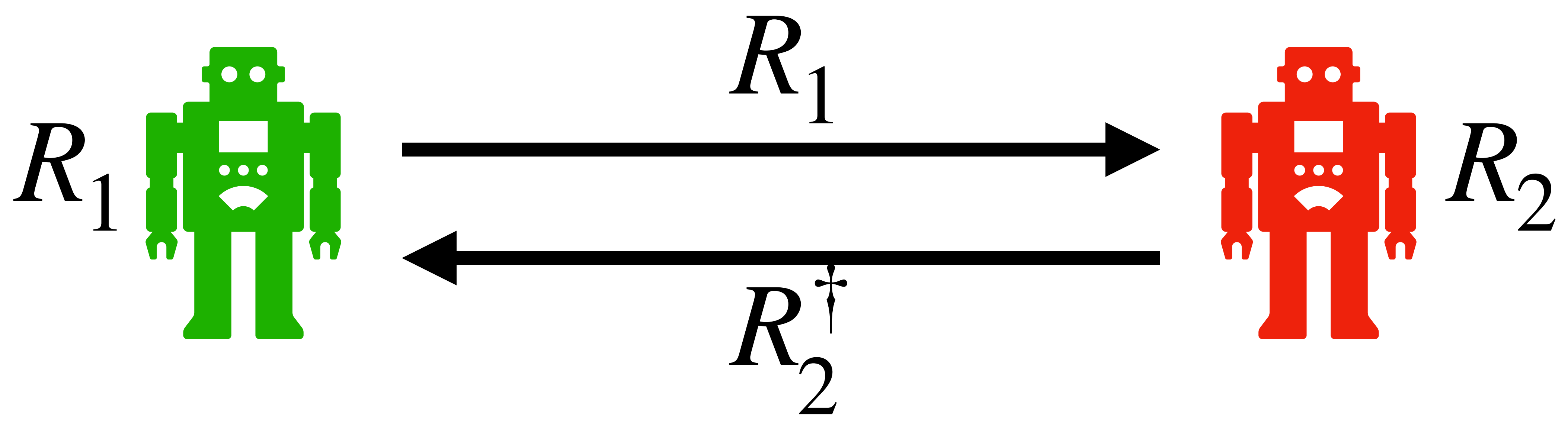
Motivation

More **realistic** attacker: information advantage instead of environment control



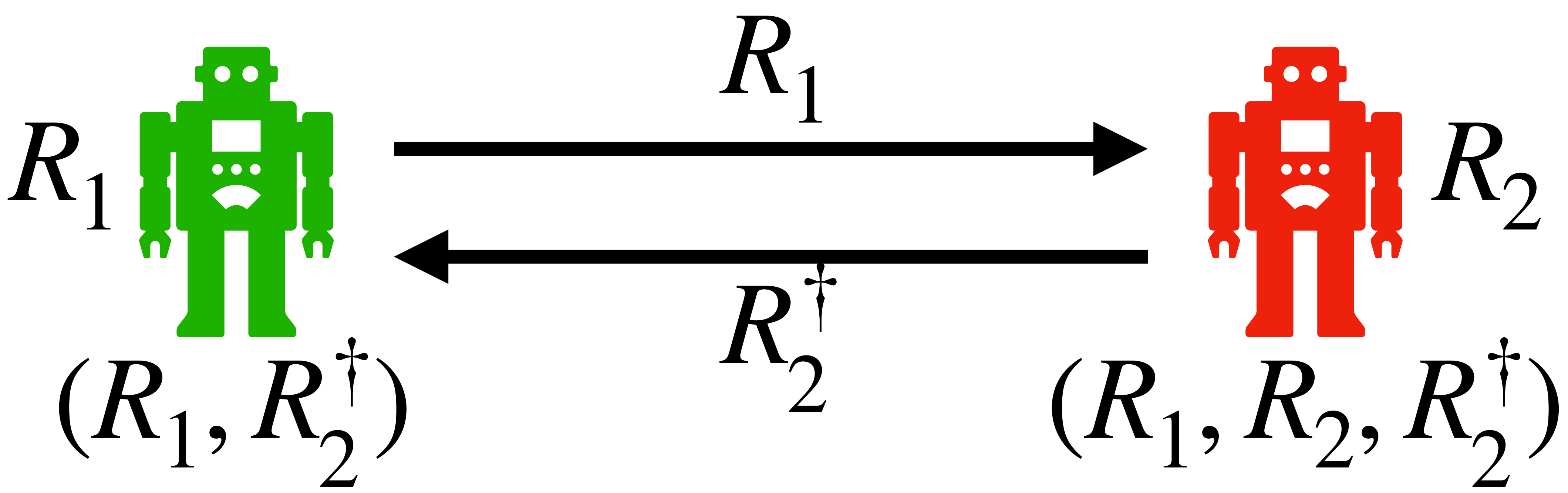
Motivation

More **realistic** attacker: information advantage instead of environment control



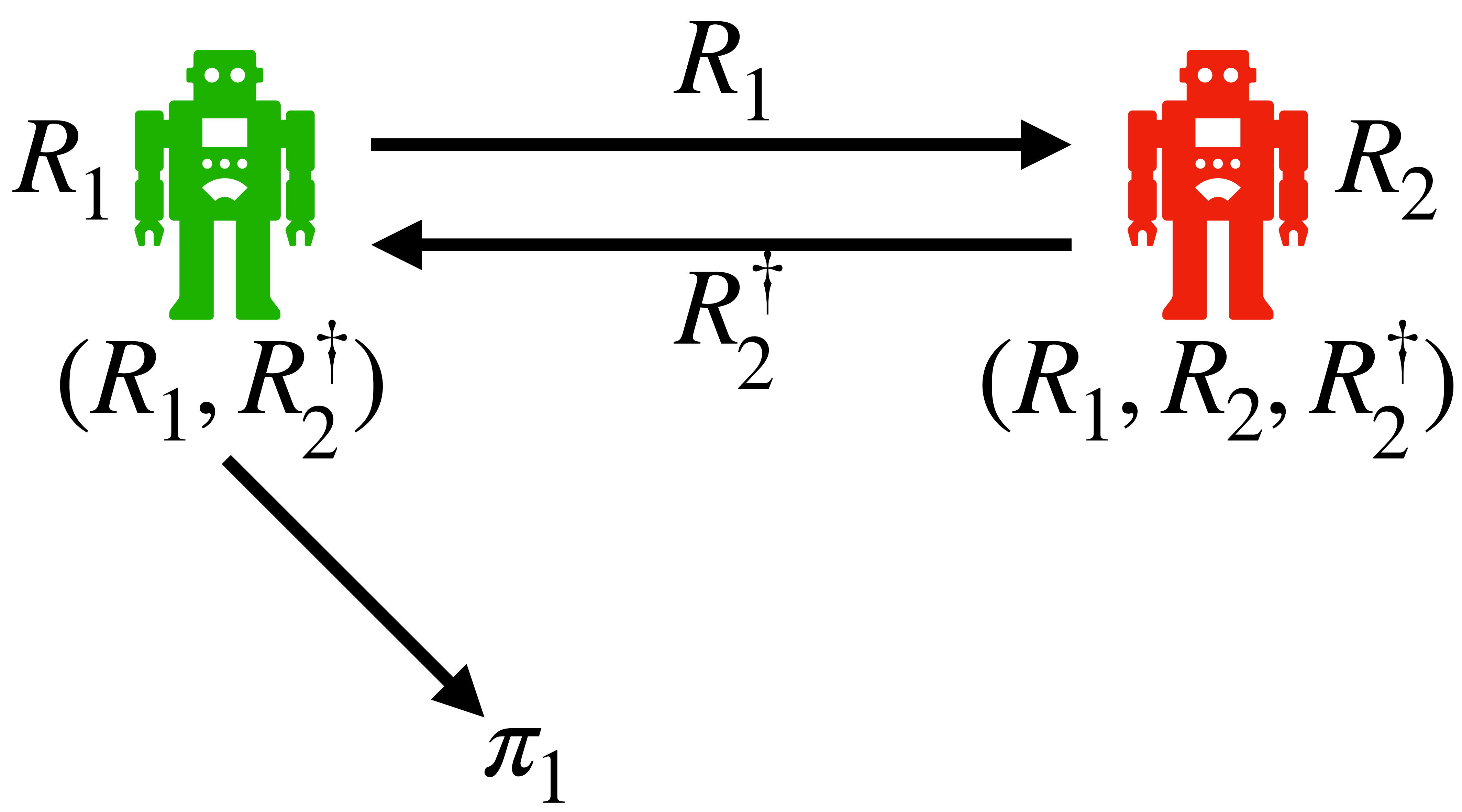
Motivation

More **realistic** attacker: information advantage instead of environment control



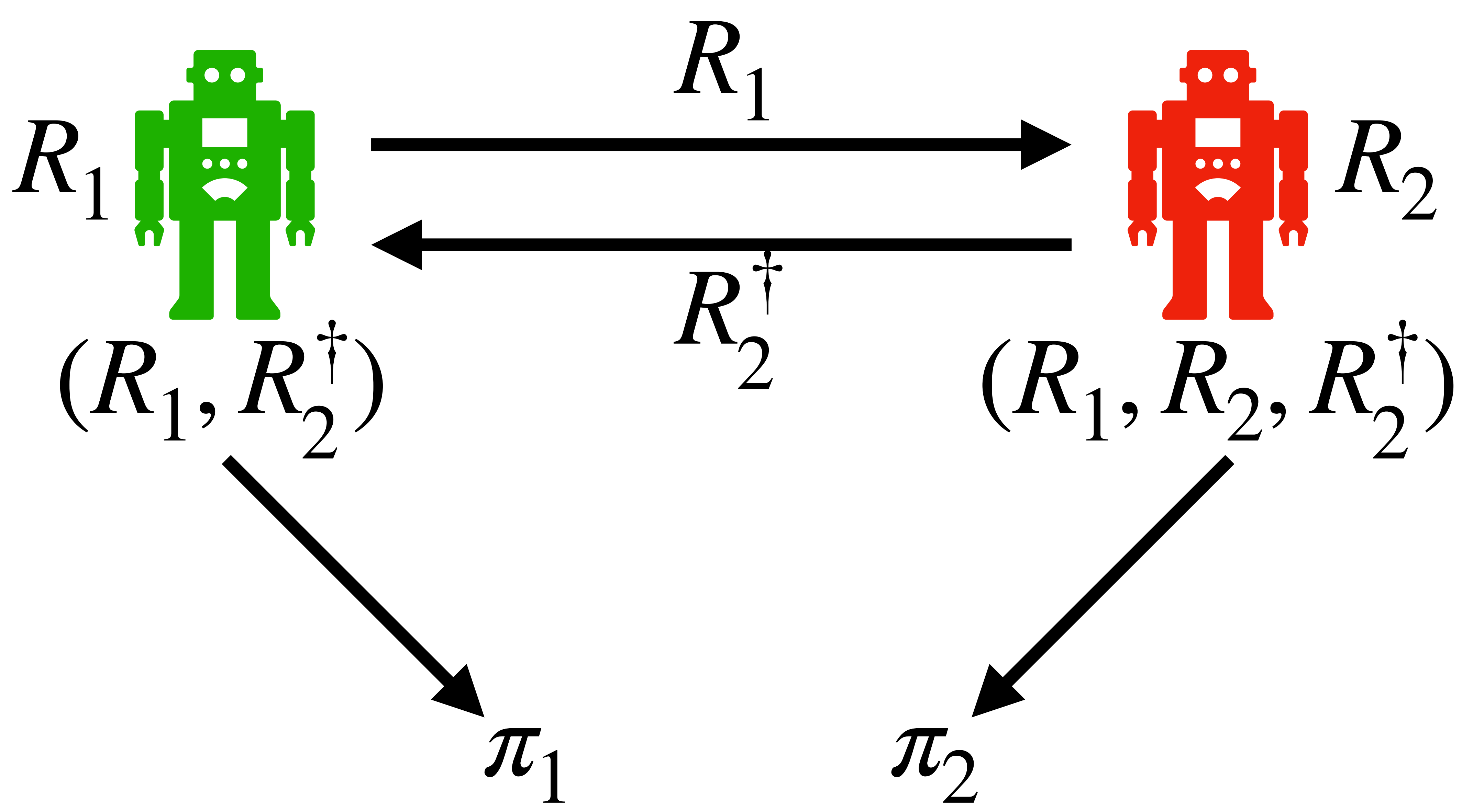
Motivation

More **realistic** attacker: information advantage instead of environment control



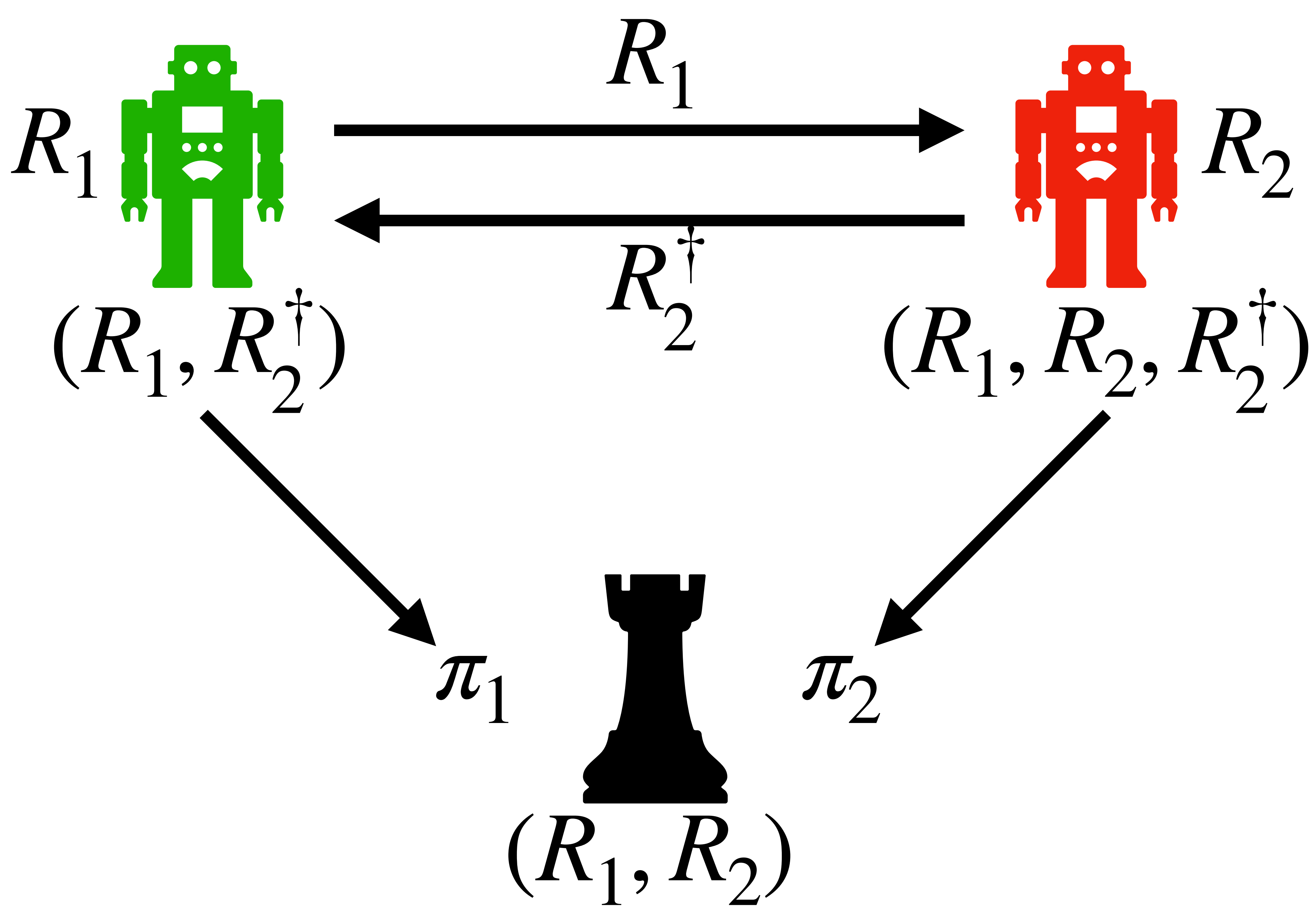
Motivation

More **realistic** attacker: information advantage instead of environment control



Motivation

More **realistic** attacker: information advantage instead of environment control



Inception

Inception

Inception Problem

$$\max_{R_2^\dagger}$$ P2's best worst-case value given P1's beliefs about $R_2^\dagger$

Inception

Inception Problem

$$\max_{R_2^\dagger}$$ P2's best worst-case value given P1's beliefs about $R_2^\dagger$

Belief set: $\Pi_2^b(R_2^\dagger)$

Inception

Inception Problem

$$\max_{R_2^\dagger}$$ P2's best worst-case value given P1's beliefs about $R_2^\dagger$

Belief set: $\Pi_2^b(R_2^\dagger)$ —— P2 is "rational"

Inception

Inception Problem

$$\max_{R_2^\dagger}$$ P2's best worst-case value given P1's beliefs about $R_2^\dagger$

Belief set: $\Pi_2^b(R_2^\dagger)$ —— P2 is "rational"

$$\Pi_2^b(R_2^\dagger) = \left\{ \pi_2 \mid \exists R_2' \in \mathbb{B}_\epsilon(R_2^\dagger), (\,\cdot\,, \pi_2) \in SOL(R_1, R_2') \right\}$$

Inception

Inception Problem

$$\max_{R_2^\dagger} \max_{\pi_2^* \in \Pi_2} \min_{\pi_1^* \in \Pi_1^*} V_2^{\pi_1^*, \pi_2^*}$$

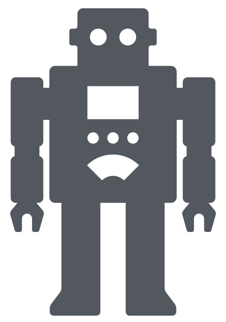
$$\text{s.t. } \Pi_1^* = \arg\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2^b(R_2^\dagger)} V_1^{\pi_1, \pi_2}$$

Belief set: $\Pi_2^b(R_2^\dagger)$ —— P2 is "rational"

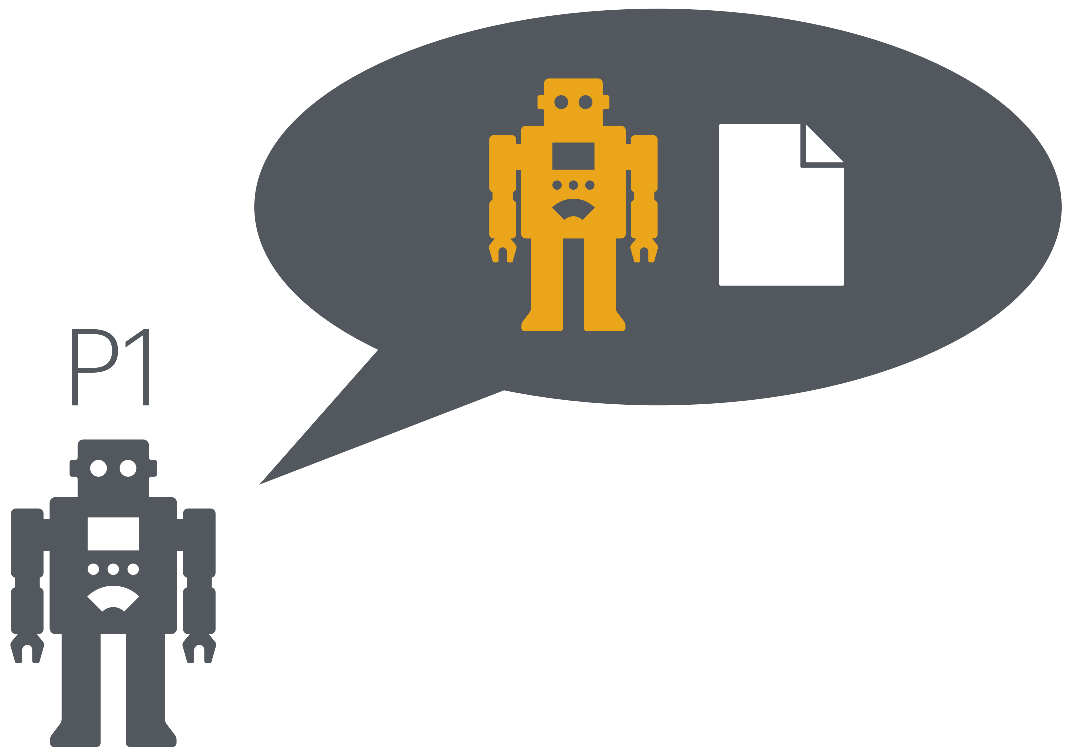
$$\Pi_2^b(R_2^\dagger) = \left\{ \pi_2 \mid \exists R_2' \in \mathbb{B}_\epsilon(R_2^\dagger), (\,\cdot\,, \pi_2) \in SOL(R_1, R_2') \right\}$$

Inception Approach

Inception Approach



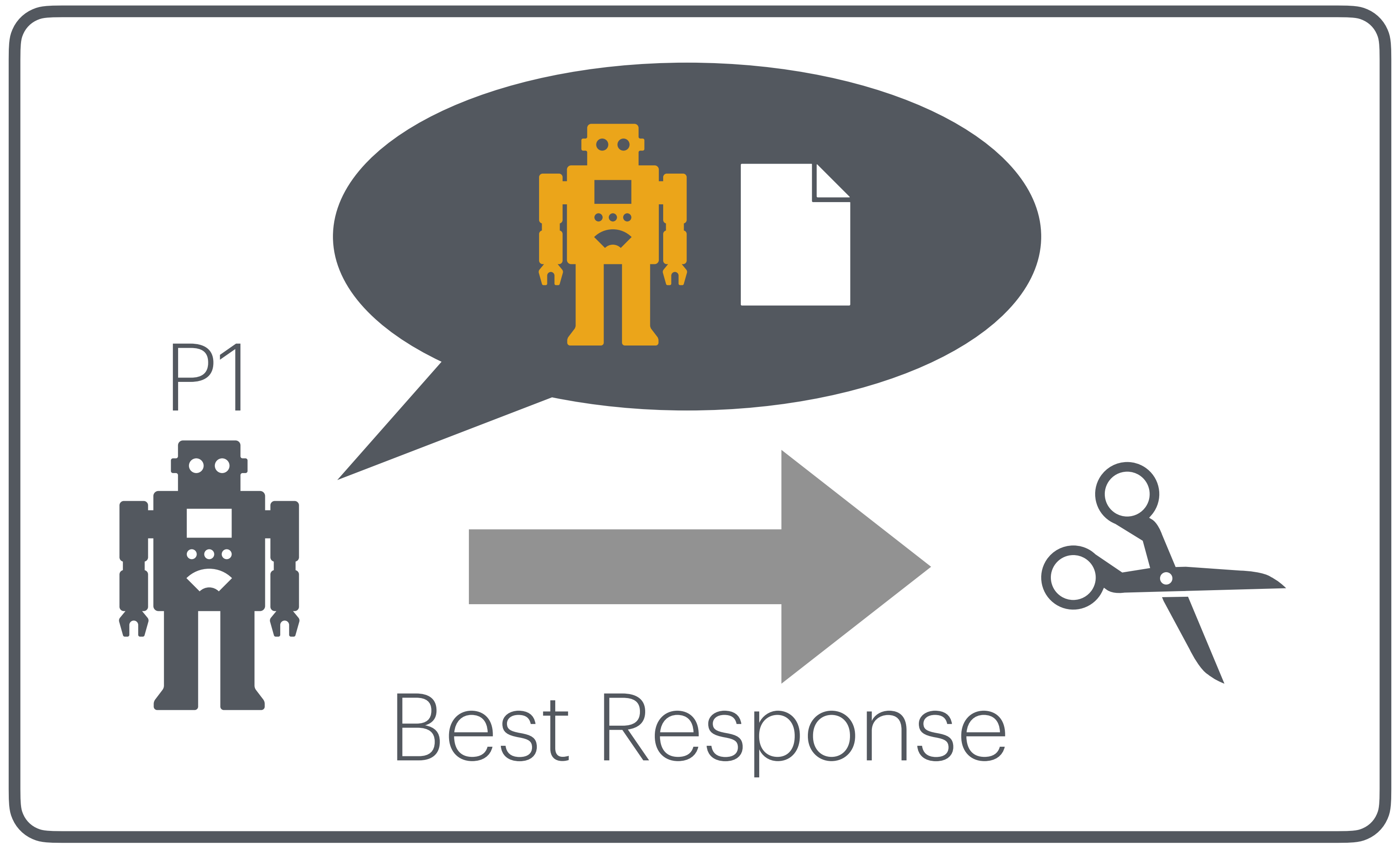
Inception Approach



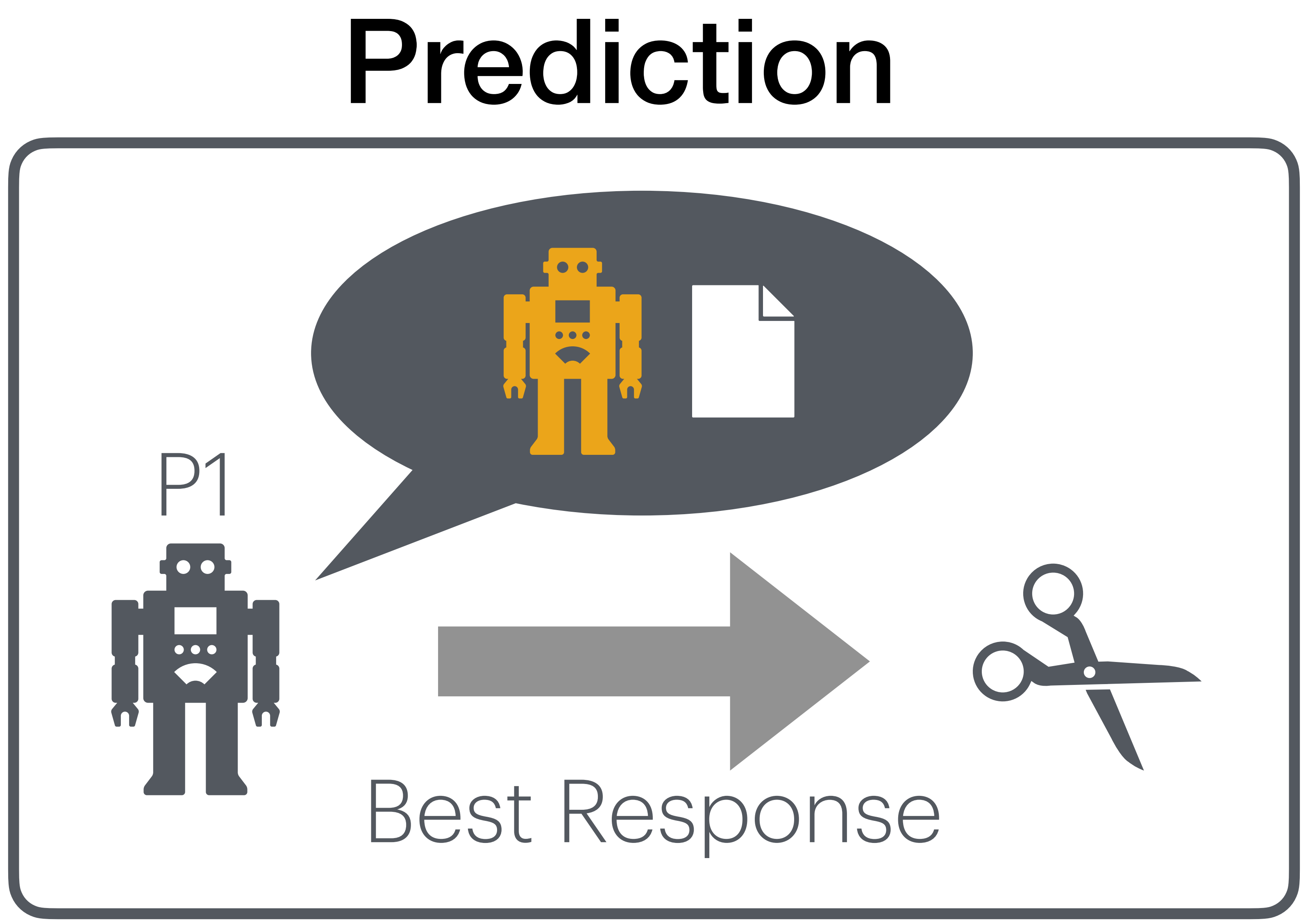
Inception Approach

Inception Approach

Prediction

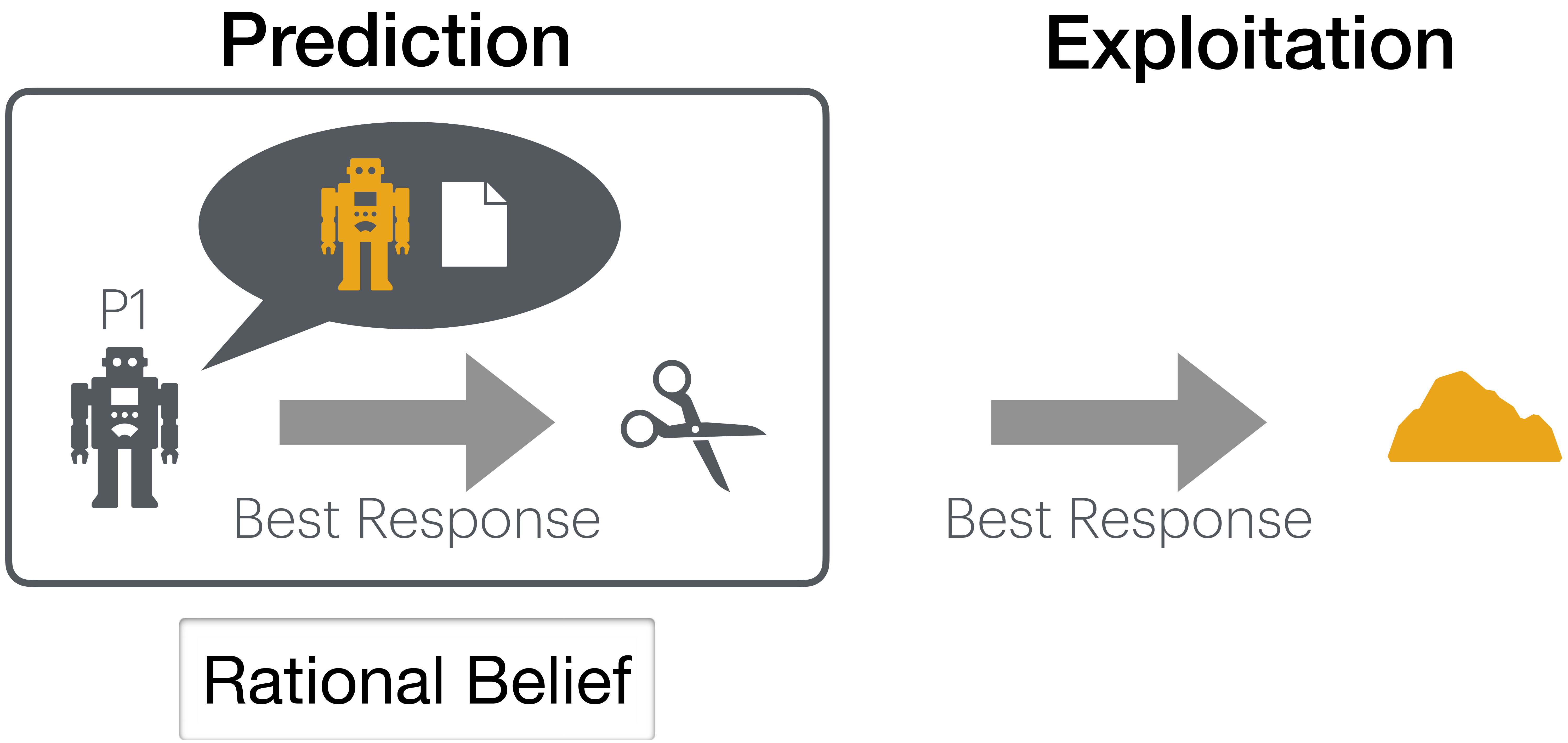


Inception Approach

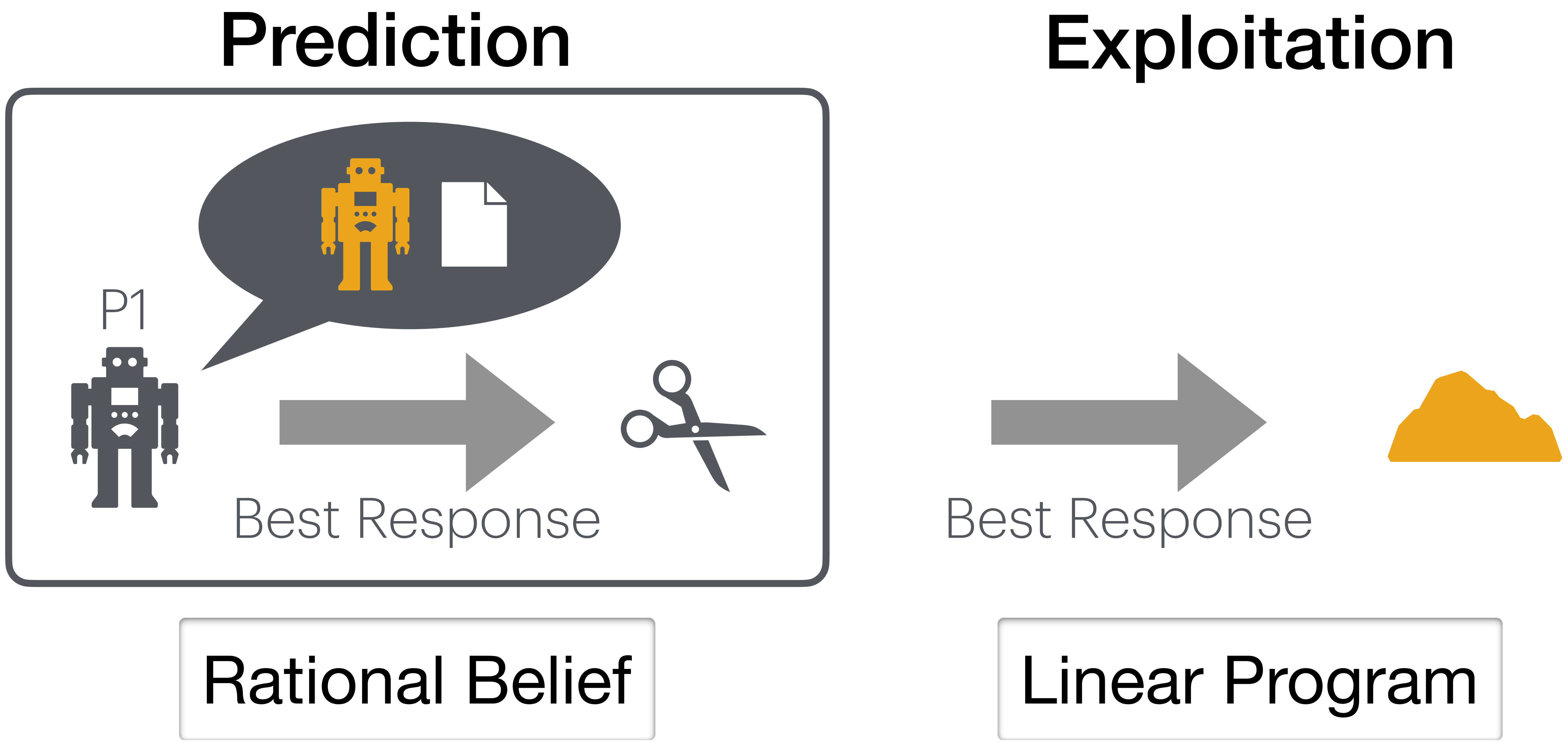


Rational Belief

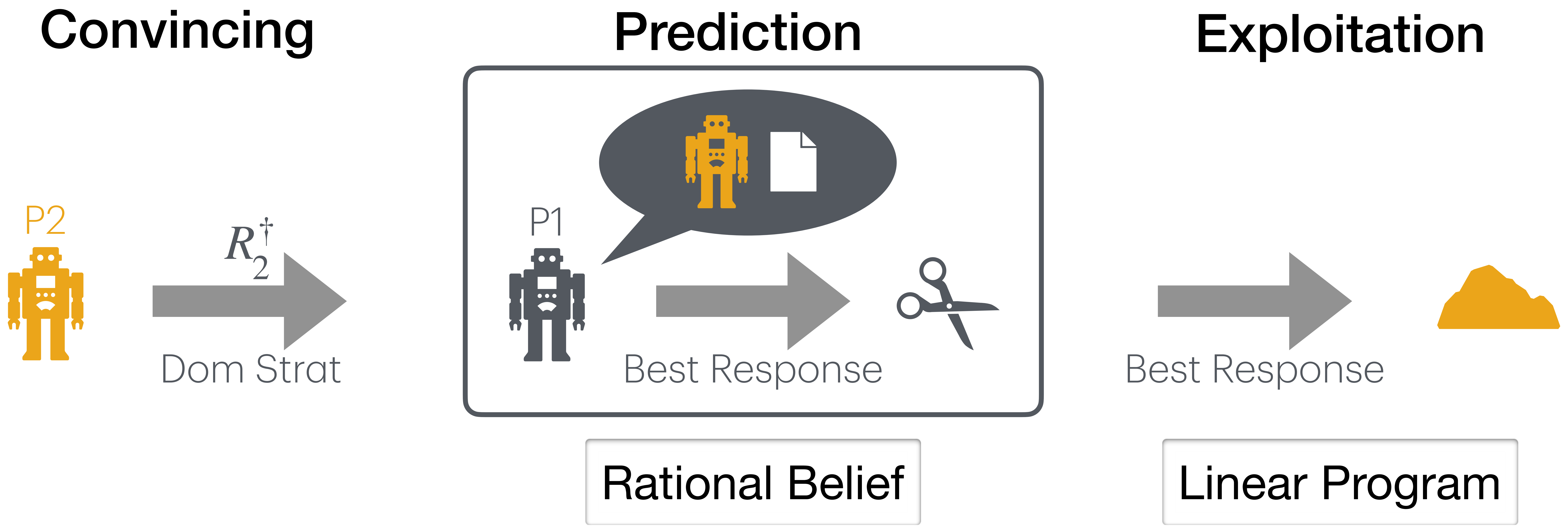
Inception Approach



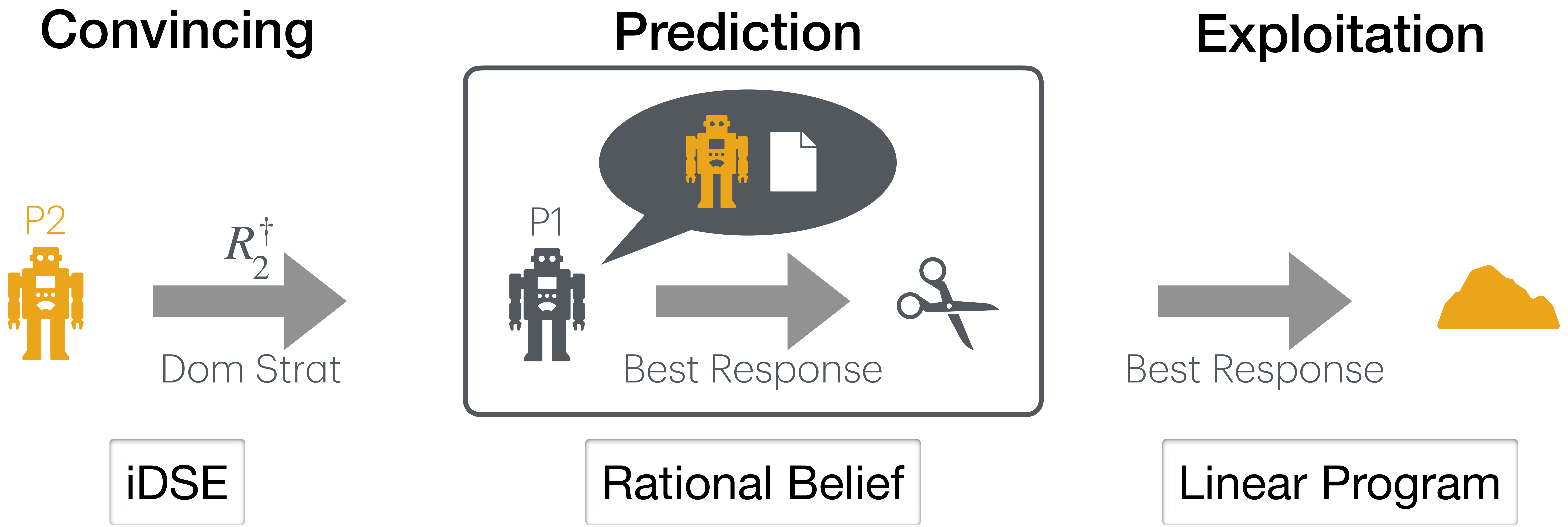
Inception Approach



Inception Approach

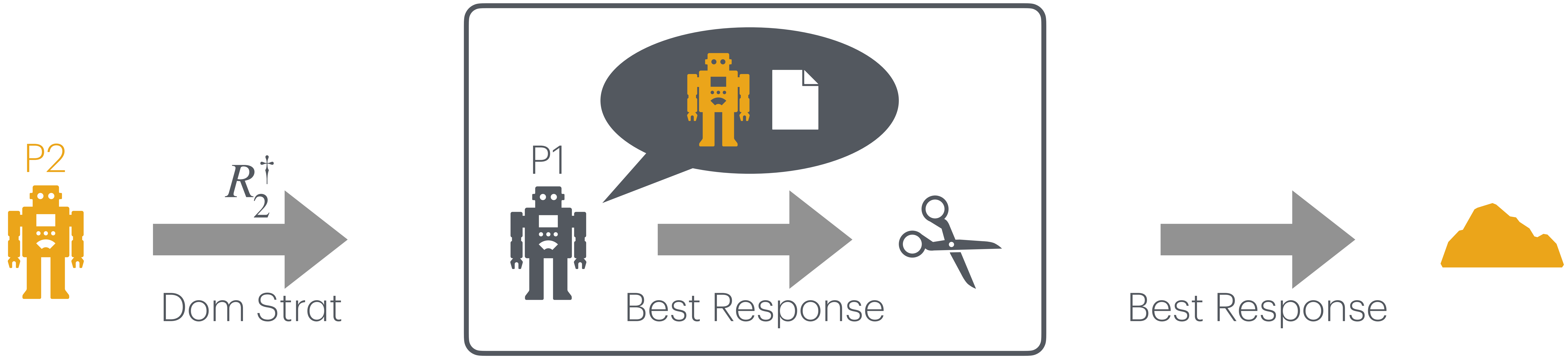


Inception Approach



Inception Approach

Convincing | **Prediction** | **Exploitation**



iDSE | Rational Belief | Linear Program

Repeat to find the best **pure** strategy inception!

Example: True Game

Example: True Game

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 0	0, ϵ

Example: True Game

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 0	0, ϵ

Unique NE

Example: True Game

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 0	0, ϵ

Unique NE

If P1 is rational, P2 gets 0!

Example: True Game

P2 wants → (U, L)

Unique NE → (D, L)

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 0	0, ϵ

If P1 is rational, P2 gets 0!

P2 fakes L

P2 fakes L

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 2ϵ	0, ϵ

Increased

P2 fakes L

P2 → L

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 2ϵ	0, ϵ

P2 fakes L

P2

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 2ϵ	0, ϵ

P1

P2 fakes L

P2

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 2ϵ	0, ϵ

P1

Tie!

P2 fakes L

P2

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 0	0, ϵ

P1

Worst-Case Best Response

P2 fakes L

P2

P1

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 0	0, ϵ

Worst-Case Best Response

P2 gets $\epsilon/2$ from (1/2,1/2) mix

P2 fakes L

P2

	L	R
U	0, 5	1, 0
D	1, ϵ	0, 0
S	1, 0	0, ϵ

P1

Worst-Case Best Response

P2 gets $\epsilon/2$ from (1/2,1/2) mix

Solved by Nash LP!

P2 fakes R

P2 fakes R

	L	R
U	0, 5	1, $5+\epsilon$
D	1, ϵ	0, 2ϵ
S	1, 0	0, ϵ

Increased

P2 fakes R

	L	R
U	0, 5	1, 5+ϵ
D	1, ϵ	0, 2ϵ
S	1, 0	0, ϵ

Unique NE

P2 fakes R

	L	R
U	0, 5	1, 5+ϵ
D	1, ϵ	0, 2ϵ
S	1, 0	0, ϵ

Unique NE

P1 must play U!

P2 fakes R

P2 wins!

	L	R
U	0, 5	1, 5+ϵ
D	1, ϵ	0, 2ϵ
S	1, 0	0, ϵ

Unique NE

P1 must play U!

P2 fakes R

“Inception Attack”

Exploitation

Exploitation

Assuming finite belief: $\Pi_2^b(R_2^\dagger) = \{\pi_2^1, \ldots, \pi_2^K\}$

Exploitation

Assuming finite belief: $\Pi_2^b(R_2^\dagger) = \{\pi_2^1, \ldots, \pi_2^K\}$

Complex

$$\max_{\pi_2^* \in \Pi_2} \min_{\pi_1^* \in \Pi_1^*} V_2^{\pi_1^*, \pi_2^*}$$

$$\text{s.t. } \Pi_1^* = \arg\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2^b(R_2^\dagger)} V_1^{\pi_1, \pi_2}$$

Exploitation

Assuming finite belief: $\Pi_2^b(R_2^\dagger) = \{\pi_2^1, \ldots, \pi_2^K\}$

Complex

$$\max_{\pi_2^* \in \Pi_2} \min_{\pi_1^* \in \Pi_1^*} V_2^{\pi_1^*, \pi_2^*}$$
$$\text{s.t. } \Pi_1^* = \arg\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2^b(R_2^\dagger)} V_1^{\pi_1, \pi_2}$$

Duality

Exploitation

Assuming finite belief: $\Pi_2^b(R_2^\dagger) = \{\pi_2^1, \ldots, \pi_2^K\}$

Complex

$$\max_{\pi_2^* \in \Pi_2} \min_{\pi_1^* \in \Pi_1^*} V_2^{\pi_1^*, \pi_2^*}$$
$$\text{s.t. } \Pi_1^* = \arg\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2^b(R_2^\dagger)} V_1^{\pi_1, \pi_2}$$

Duality

Linear

$$\max_{y \in \mathbb{R}^m, w \in \mathbb{R}^K, \alpha \in \mathbb{R}} \quad z^* 1^\top w - \alpha$$
$$\text{s.t.} \quad \alpha + e_i^\top B y - e_i^\top A' w \geq 0 \quad \forall i \in [n]$$
$$1^\top y = 1, \quad y \geq 0 \quad w \geq 0.$$

Exploitation

Assuming finite belief: $\Pi_2^b(R_2^\dagger) = \{\pi_2^1, \ldots, \pi_2^K\}$

Complex

$$\max_{\pi_2^* \in \Pi_2} \min_{\pi_1^* \in \Pi_1^*} V_2^{\pi_1^*, \pi_2^*}$$
$$\text{s.t. } \Pi_1^* = \arg\max_{\pi_1 \in \Pi_1} \min_{\pi_2 \in \Pi_2^b(R_2^\dagger)} V_1^{\pi_1, \pi_2}$$

Duality →

Linear

$$\max_{y \in \mathbb{R}^m, w \in \mathbb{R}^K, \alpha \in \mathbb{R}} \quad z^* 1^\top w - \alpha$$
$$\text{s.t.} \quad \alpha + e_i^\top B y - e_i^\top A' w \geq 0 \quad \forall i \in [n]$$
$$1^\top y = 1, \quad y \geq 0 \quad w \geq 0.$$

Solve a sequence of LPs for MG case!

Results

Results

Theorem: *rationality enables the* ***polynomial-time*** *computation of* ***misinformation attacks*** *that are optimal amongst the set of dominant-mixture reward functions.*

Results

Theorem: *rationality enables the* ***polynomial-time*** *computation of* ***misinformation attacks*** *that are optimal amongst the set of dominant-mixture reward functions.*

First efficient misinformation attacks on Markov games!

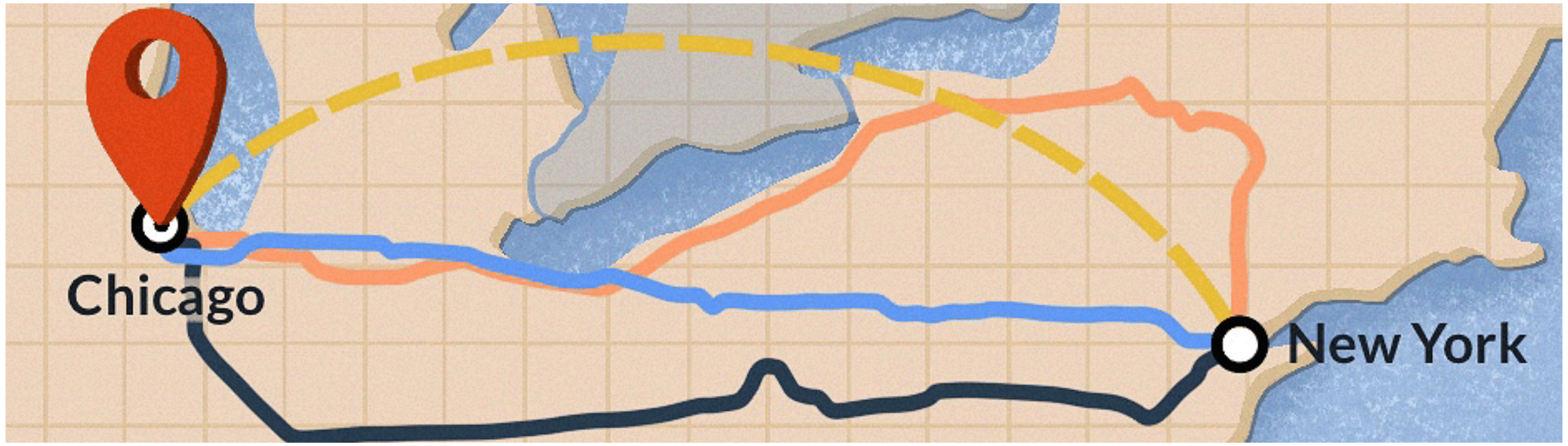
Constrained MARL

Anytime Constraints

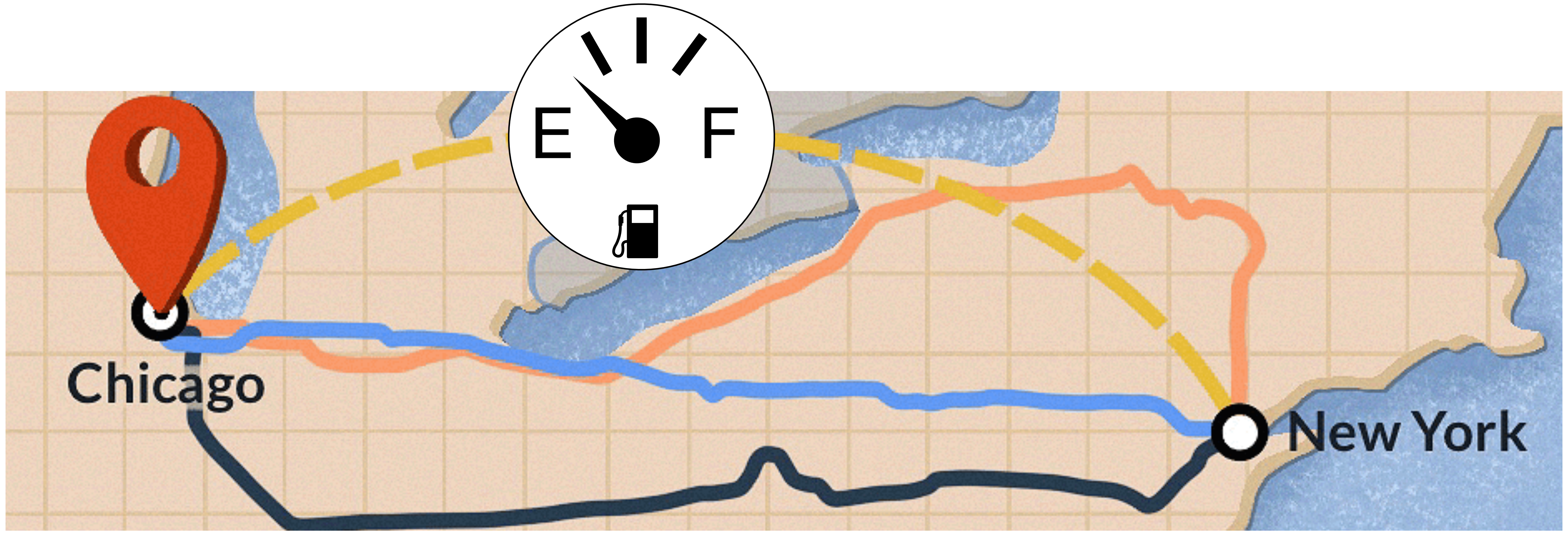
***AISTATS 2024**

Motivation

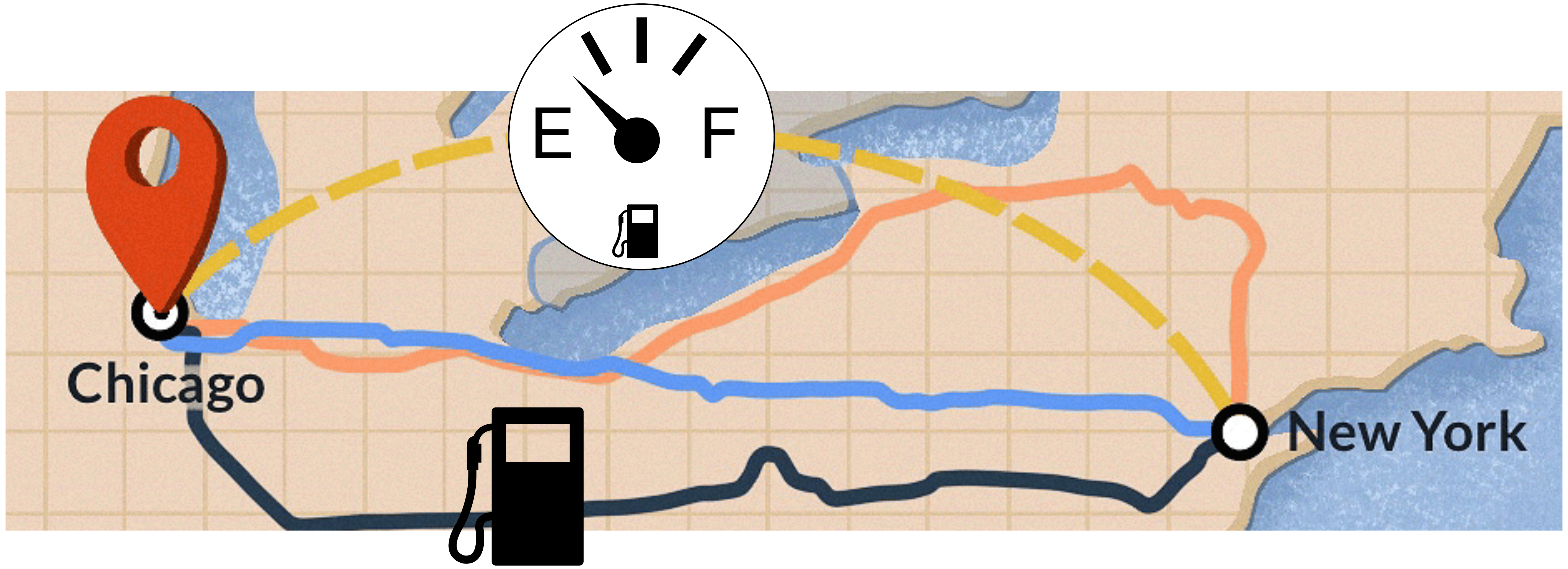
Motivation



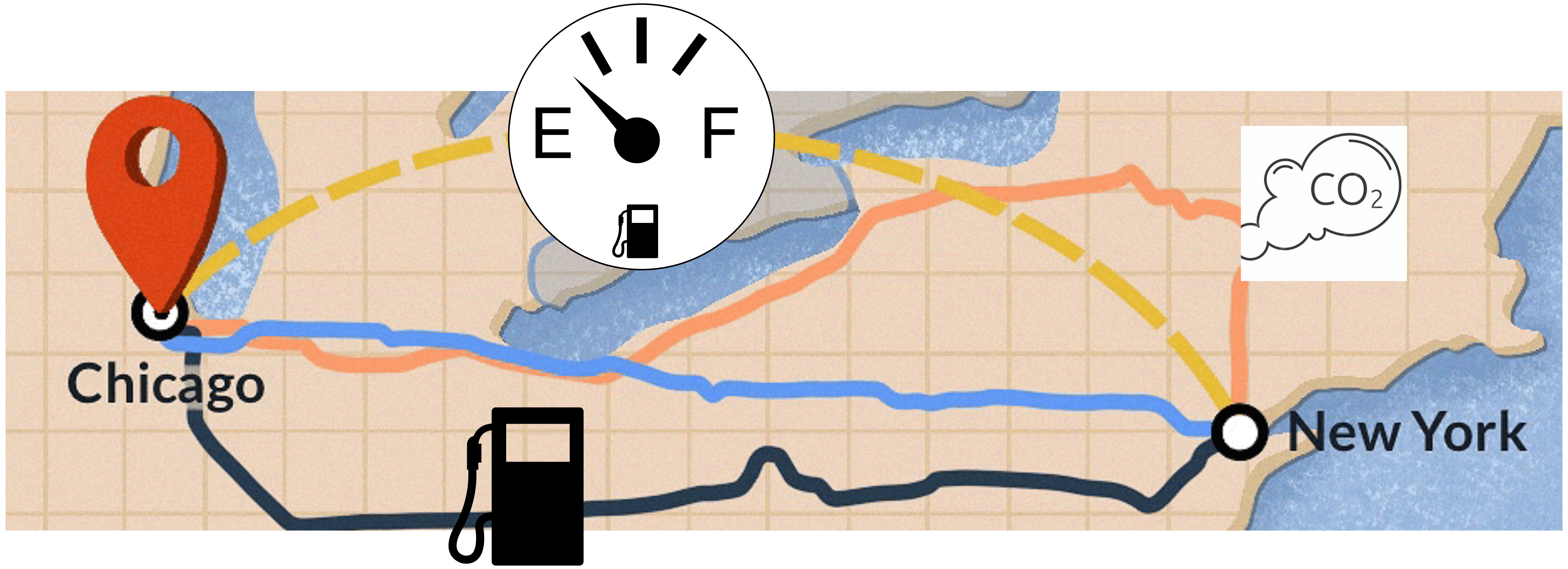
Motivation



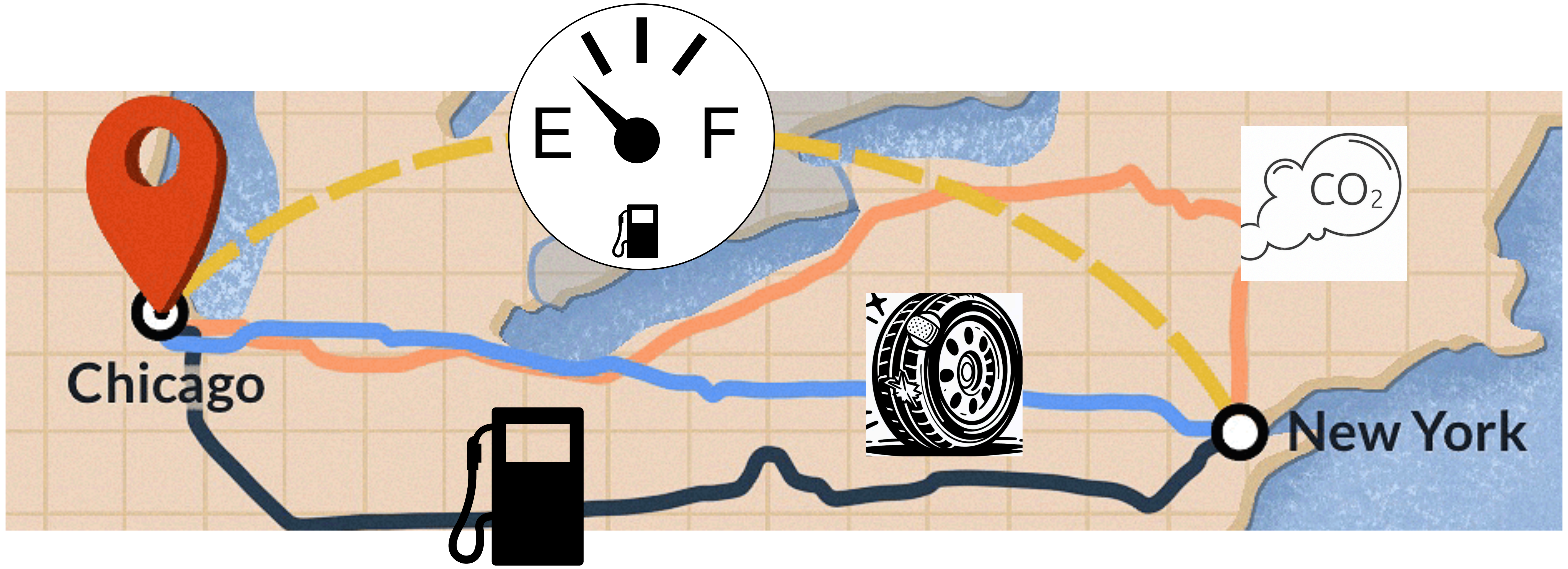
Motivation



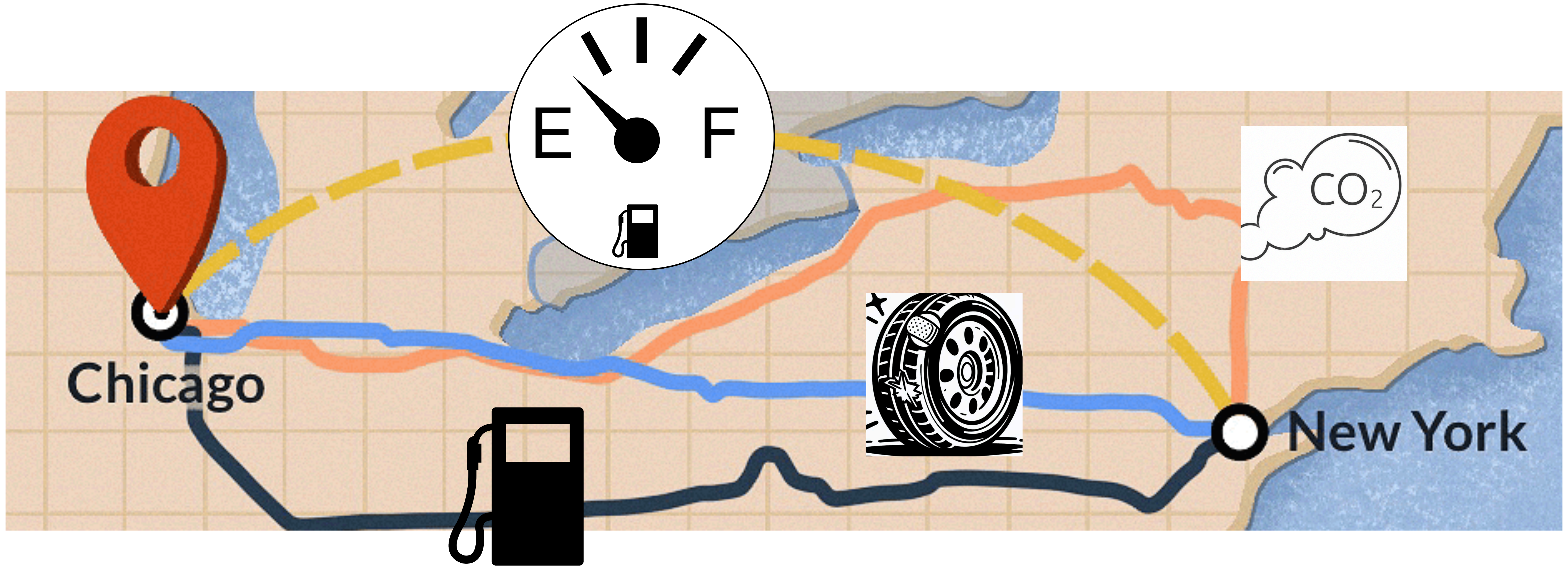
Motivation



Motivation

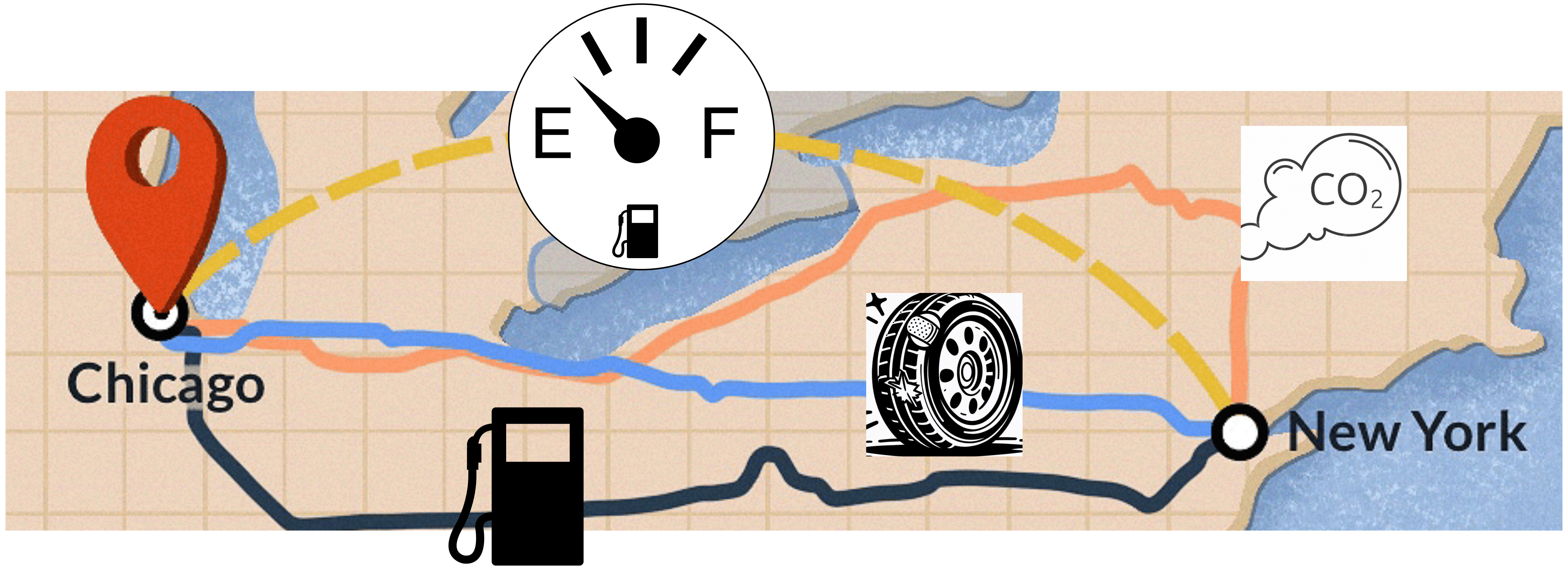


Motivation



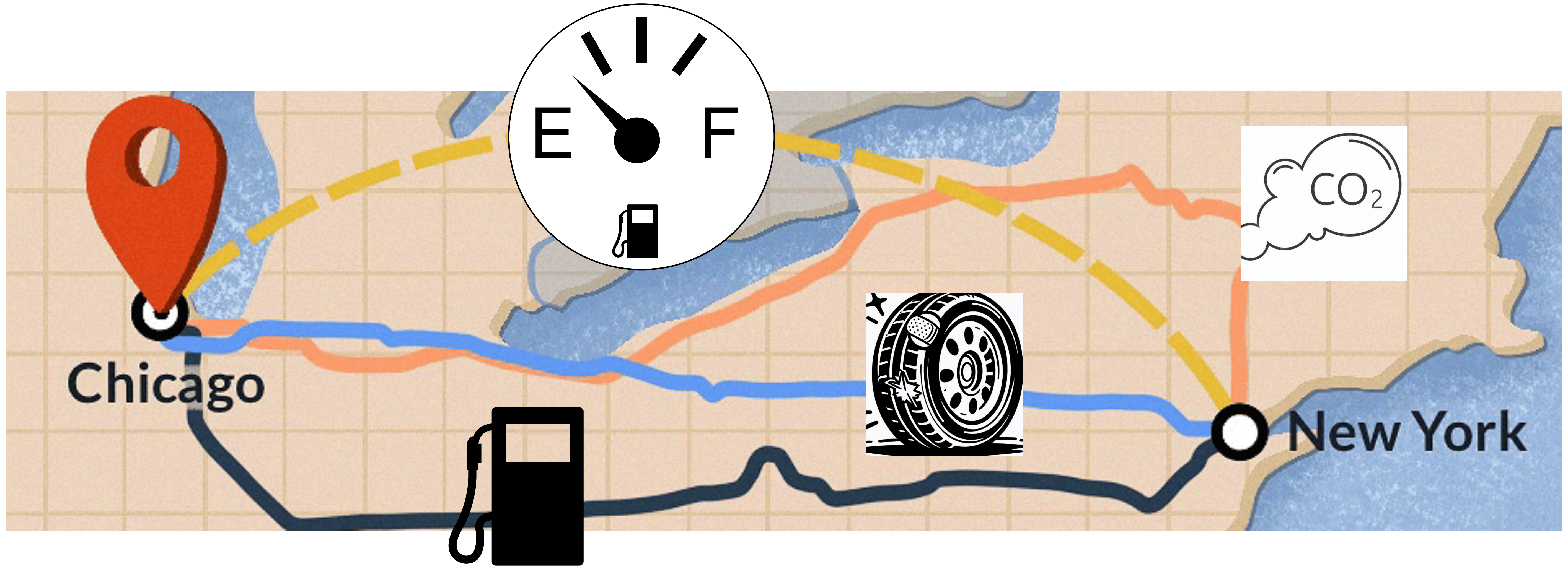
$$\mathbb{P}_M^{\pi}\left[\sum_{h=1}^{H} c_h \leq B\right] = 1$$

Motivation



$$\mathbb{P}^{\pi}_{M}\left[\sum_{h=1}^{H} c_h \leq B\right] = 1$$

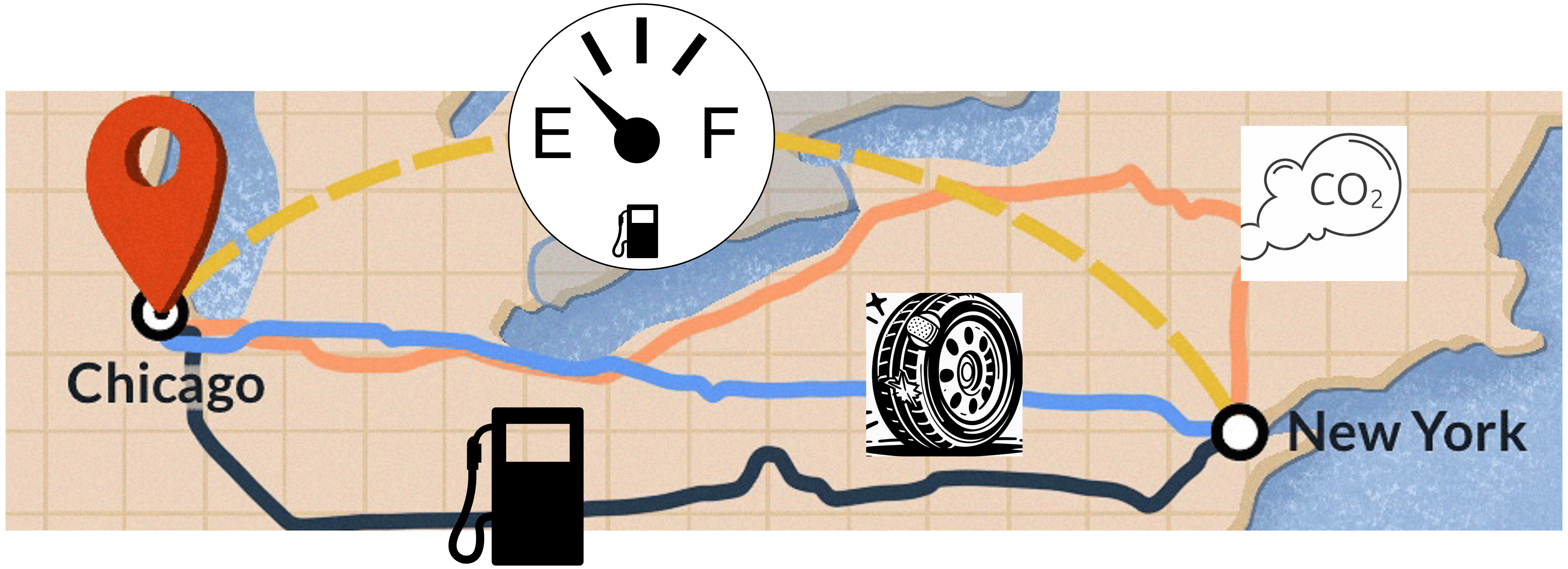
Motivation



✗ $$\mathbb{P}_M^{\pi}\left[\sum_{h=1}^{H} c_h \le B\right] = 1$$

Cannot IOU a gas tank!

Motivation

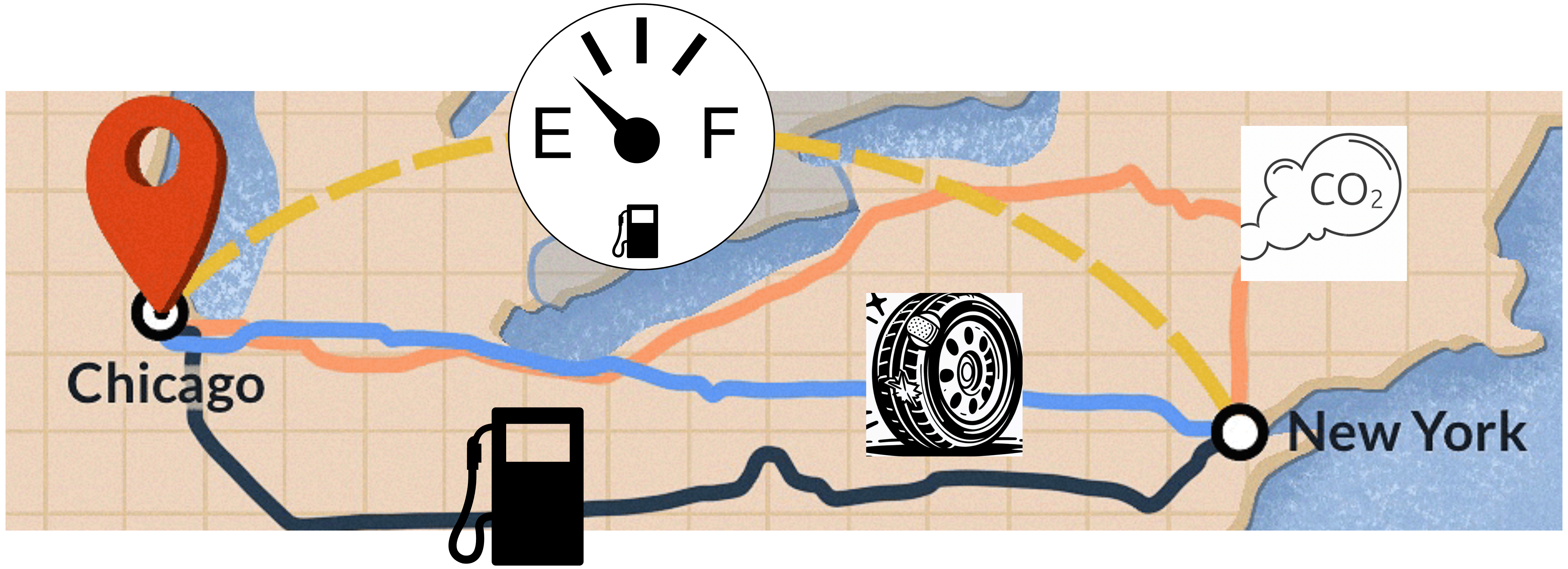


✗ $$\mathbb{P}_M^\pi\left[\sum_{h=1}^{H} c_h \leq B\right] = 1$$

Cannot IOU a gas tank!

$$\mathbb{P}_M^\pi\left[\forall t \in [H], \sum_{h=1}^{t} c_h \leq B\right] = 1$$

Motivation



✗ $$\mathbb{P}^{\pi}_{M}\left[\sum_{h=1}^{H} c_h \leq B\right] = 1$$

Cannot IOU a gas tank!

✓ $$\mathbb{P}^{\pi}_{M}\left[\forall t \in [H], \sum_{h=1}^{t} c_h \leq B\right] = 1$$

Constrained Problem

Constrained Problem

Agent's **goal** is to solve:

Constrained Problem

Agent's **goal** is to solve:

$$\max_{\pi} \mathbb{E}_M^{\pi}\left[\sum_{h=1}^{H} r_h(s_h, a_h)\right] \quad \text{s.t.} \quad \mathbb{P}_M^{\pi}\left[\forall t \in [H],\ \sum_{h=1}^{t} c_h \leq B\right] = 1.$$

Challenges

Challenges

1. Feasible policies non-Markovian

Challenges

1. Feasible policies non-Markovian

2. Optimization is NP-hard

Challenges

1. Feasible policies **non-Markovian**

2. Optimization is **NP-hard**

3. Determining feasibility of ≥ 2 constraints is NP-hard
   $\implies$ **Hardness of (value) Approximation**

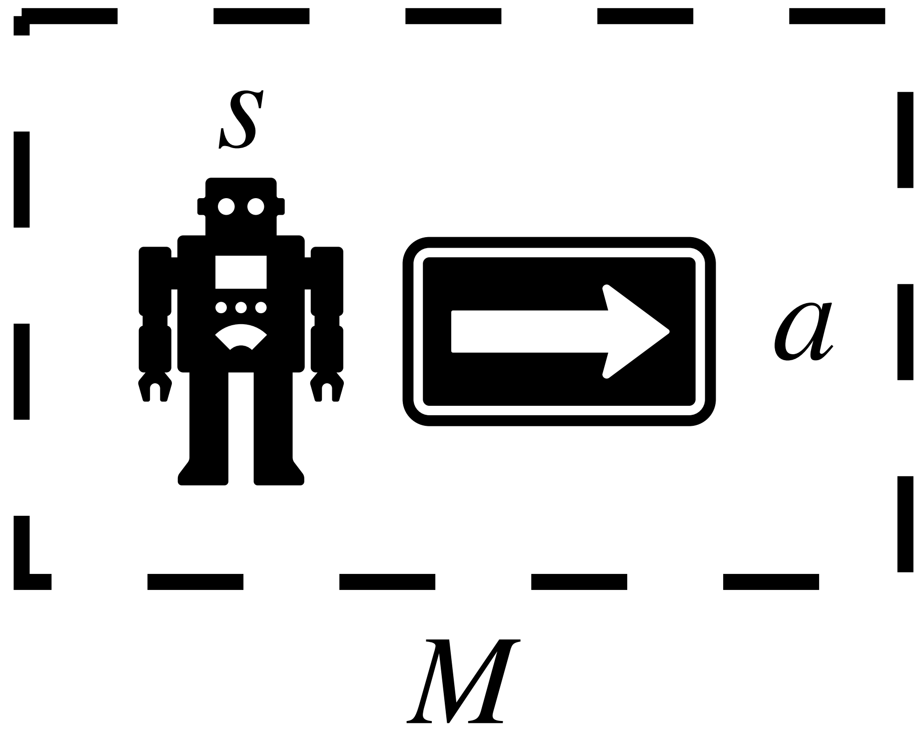
Reduction

Reduction

1. State-Cost Augmentation

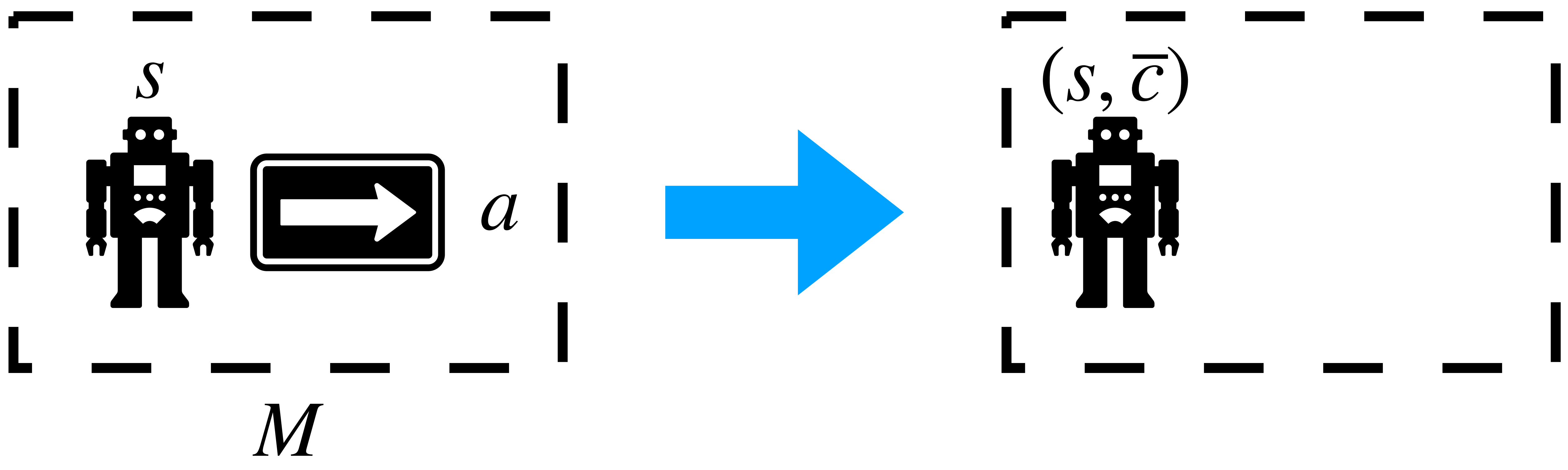
Reduction

1. State-Cost Augmentation



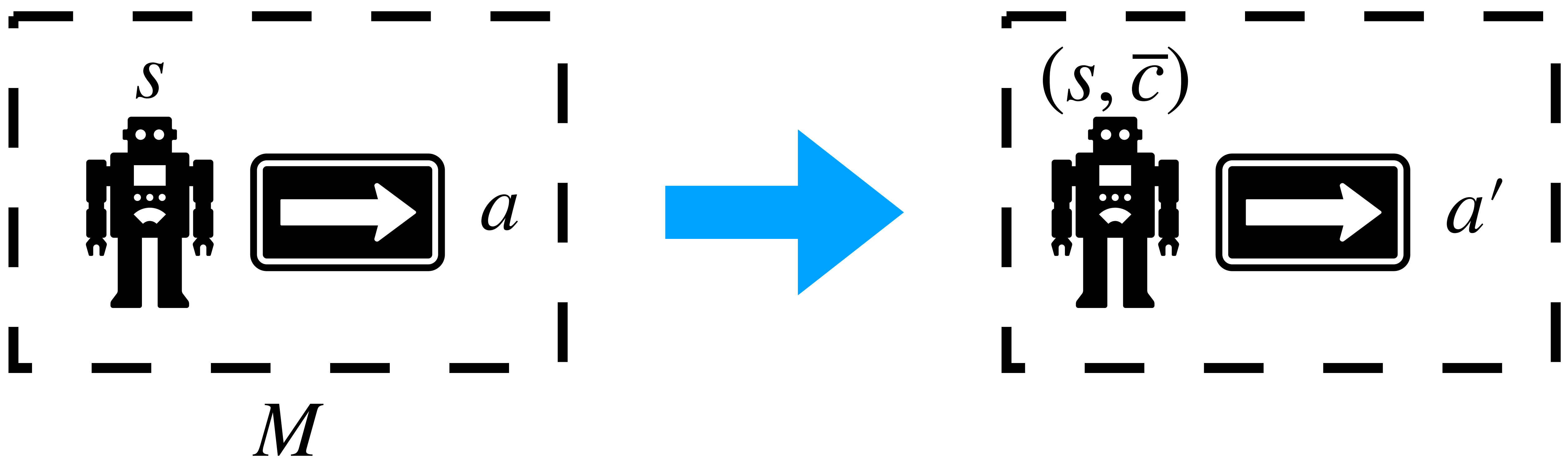
Reduction

1. State-Cost Augmentation



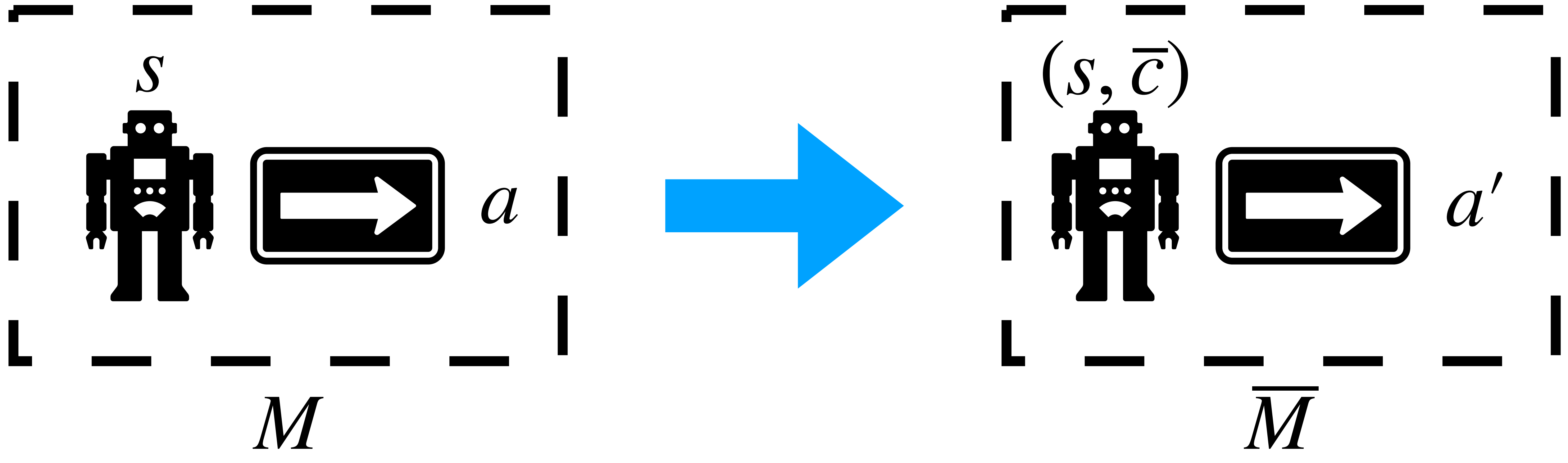
Reduction

1. State-Cost Augmentation



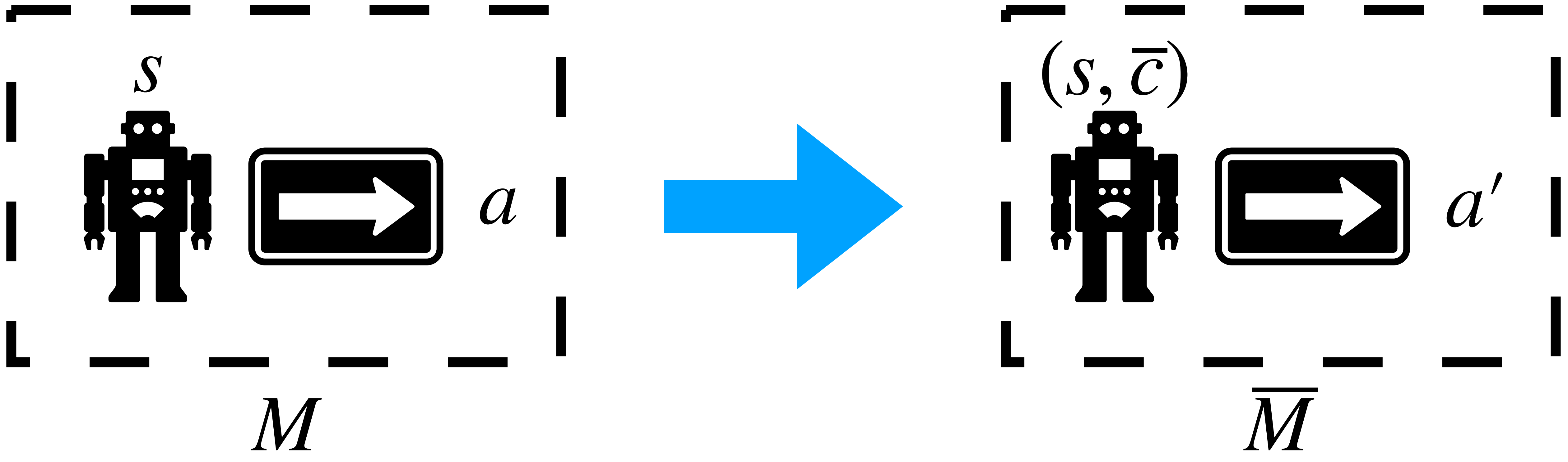
Reduction

1. State-Cost Augmentation



Reduction

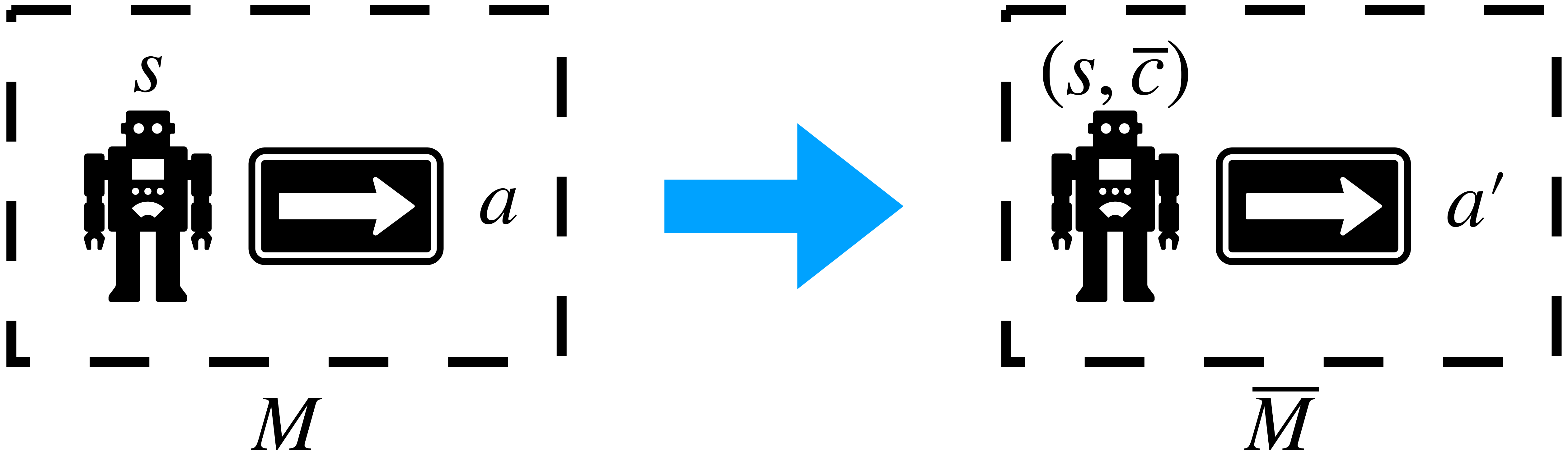
1. State-Cost Augmentation



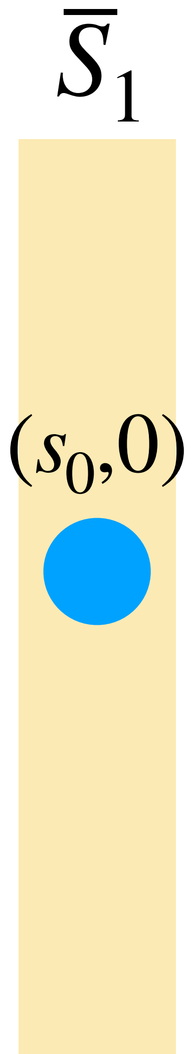
2. BFS Generate Feasible Costs

Reduction

1. State-Cost Augmentation

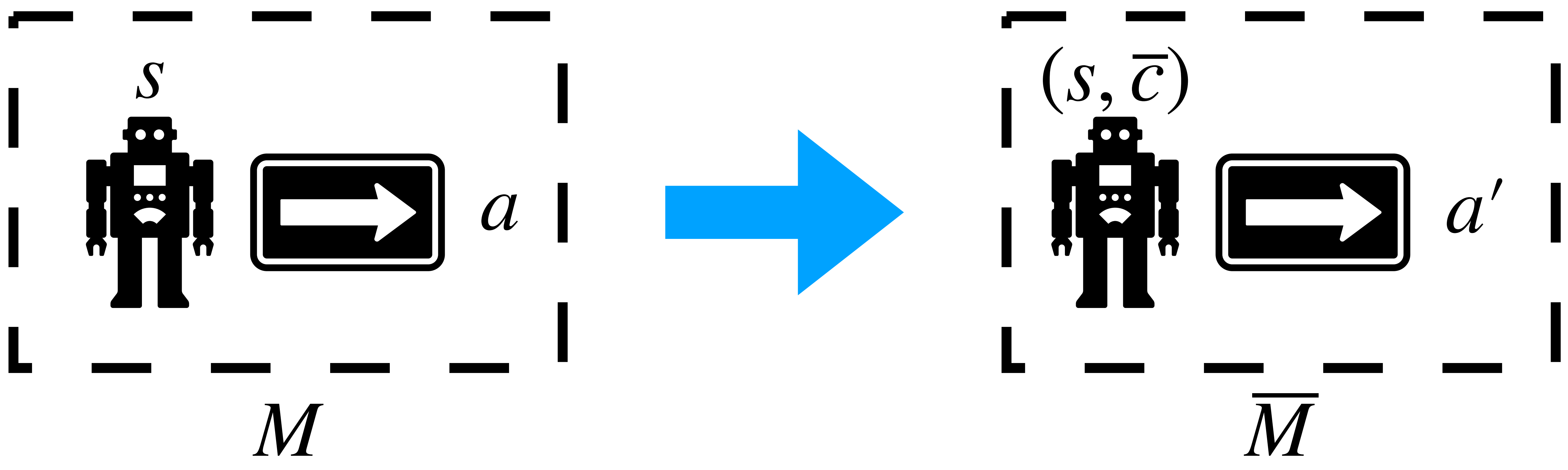


2. BFS Generate Feasible Costs

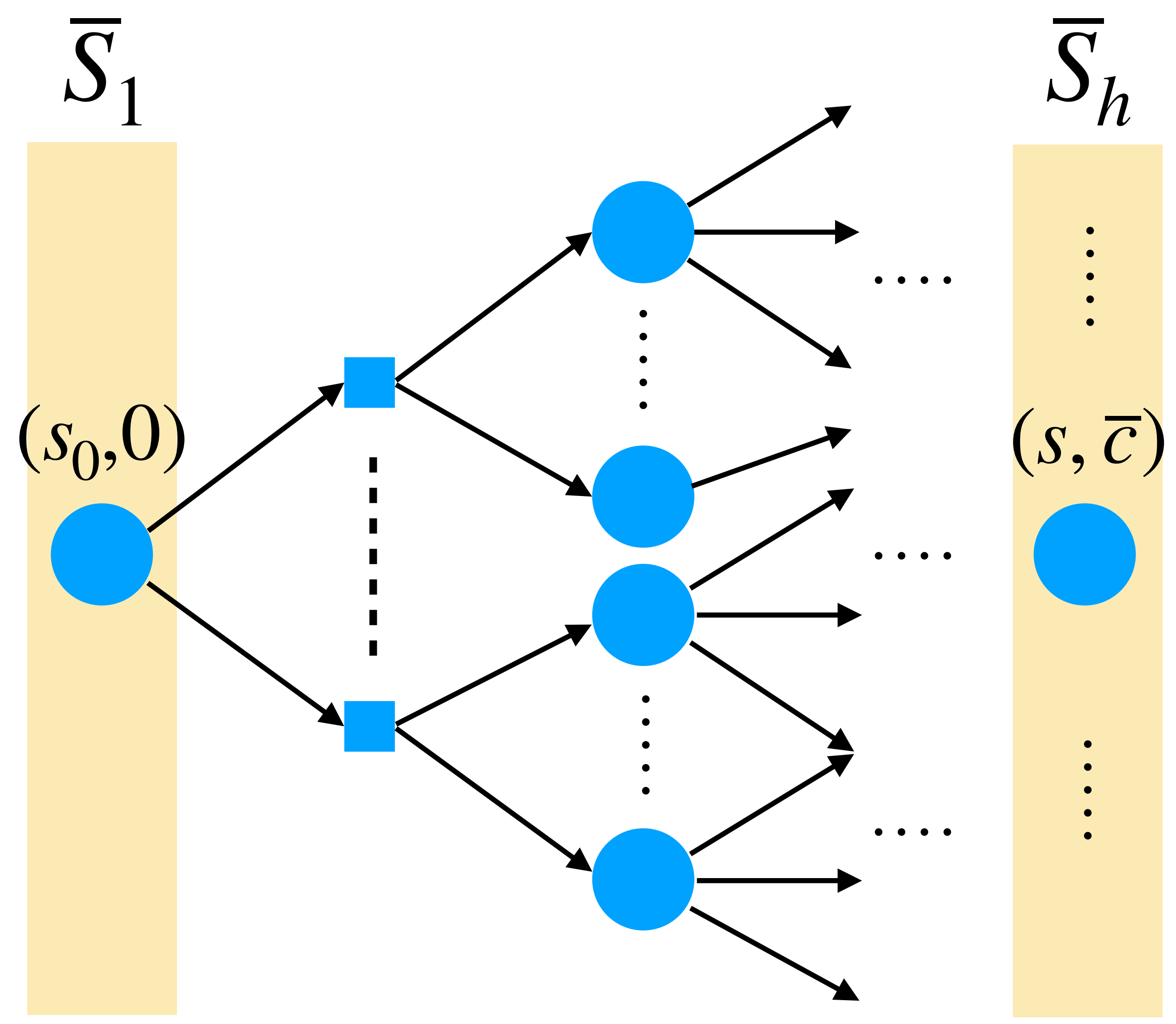


Reduction

1. State-Cost Augmentation

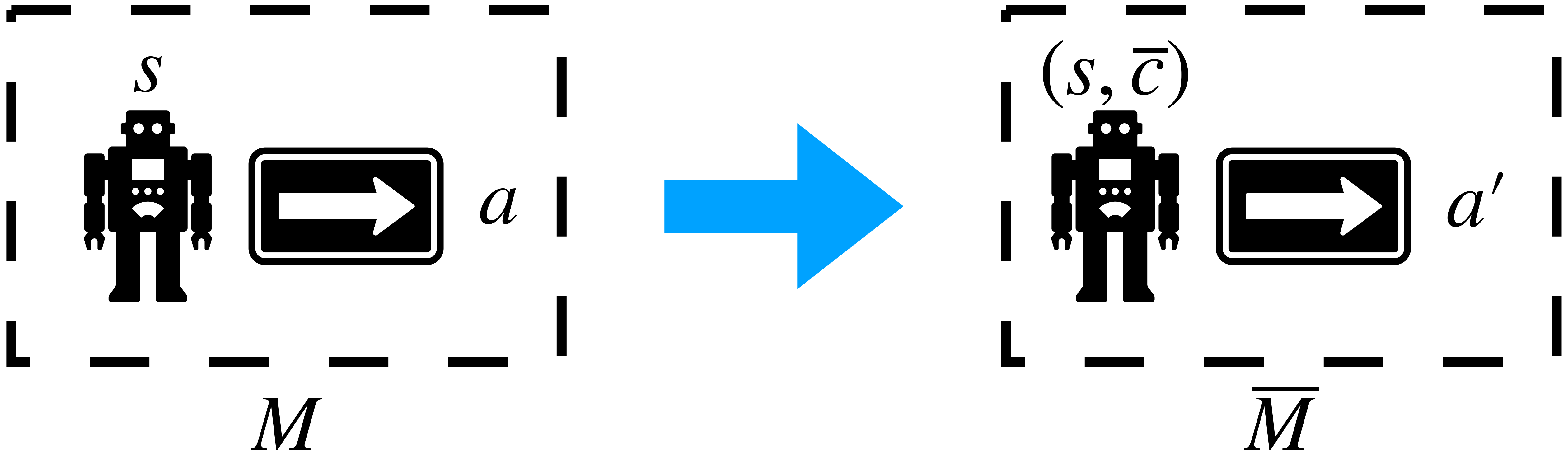


2. BFS Generate Feasible Costs

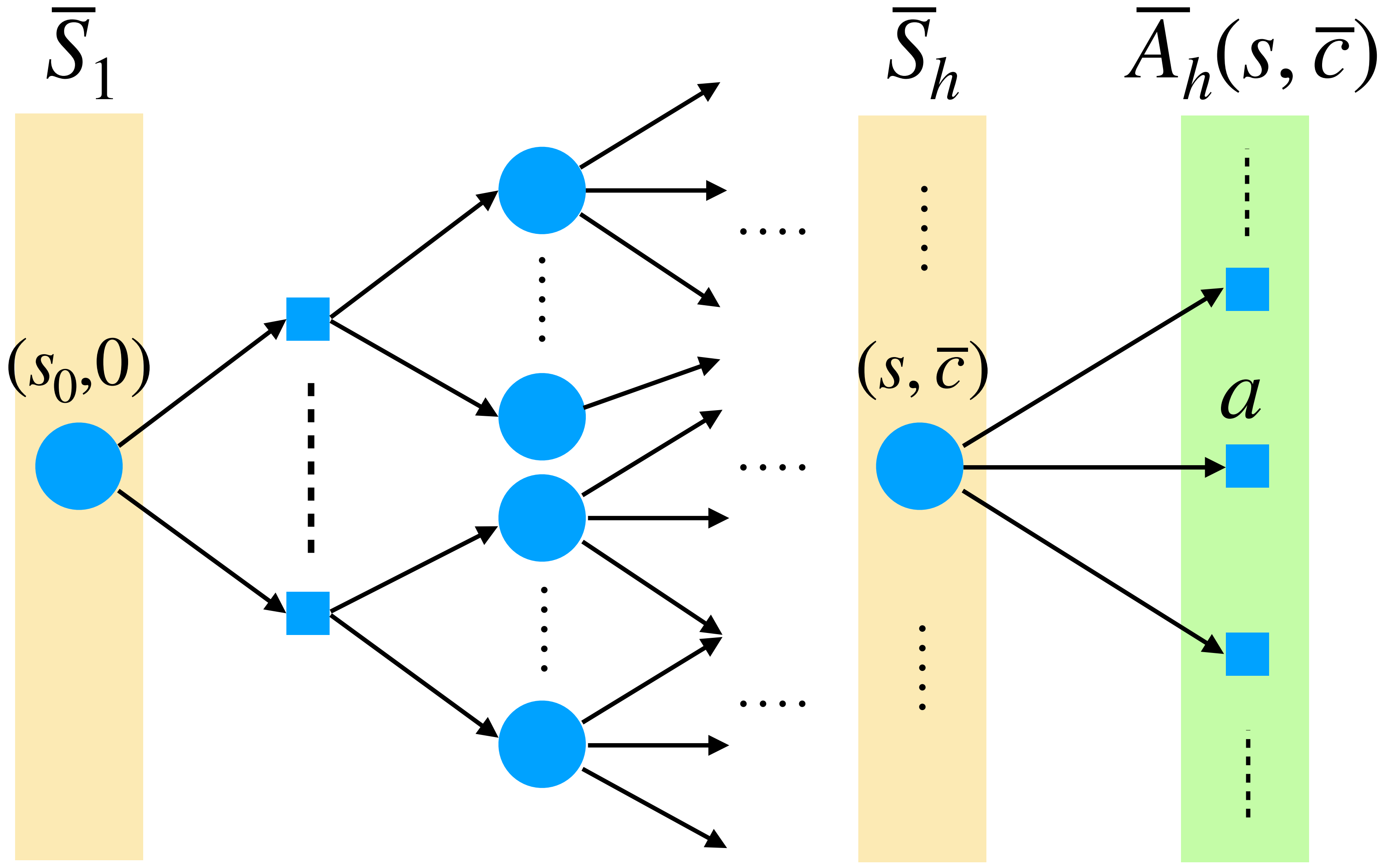


Reduction

1. State-Cost Augmentation

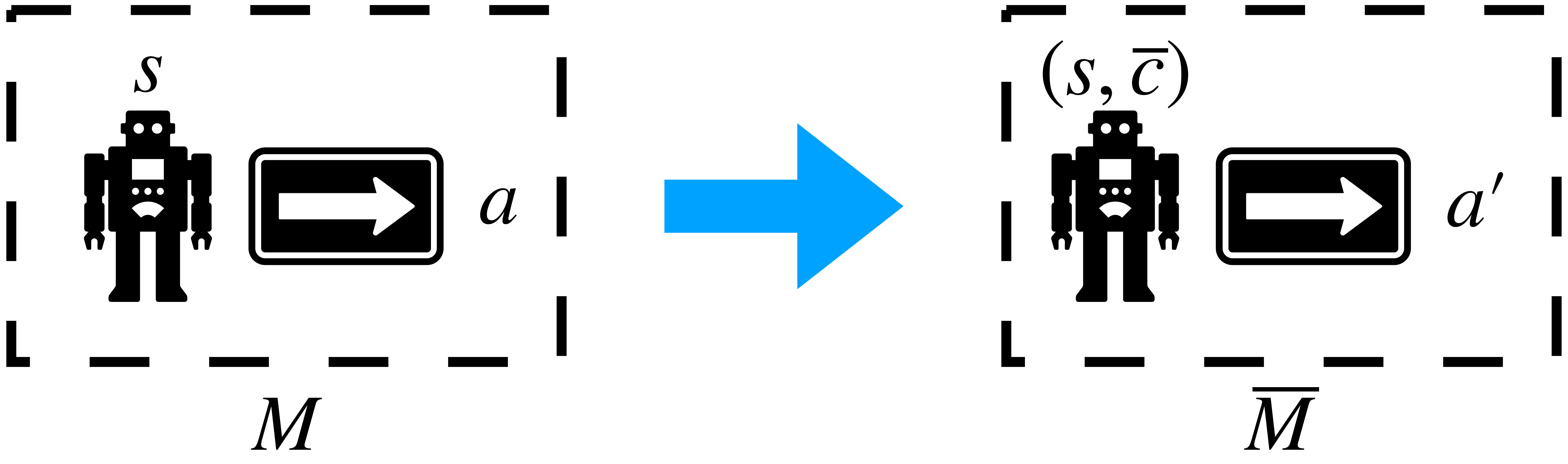


2. BFS Generate Feasible Costs

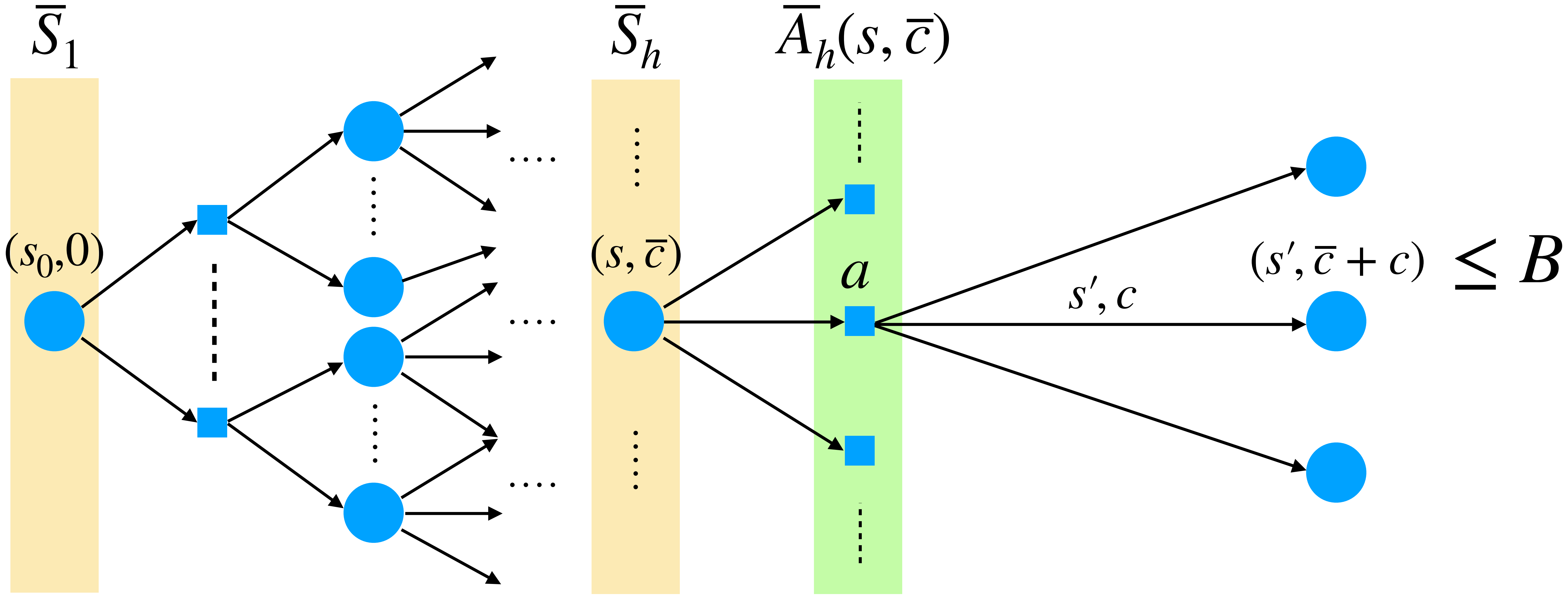


Reduction

1. State-Cost Augmentation

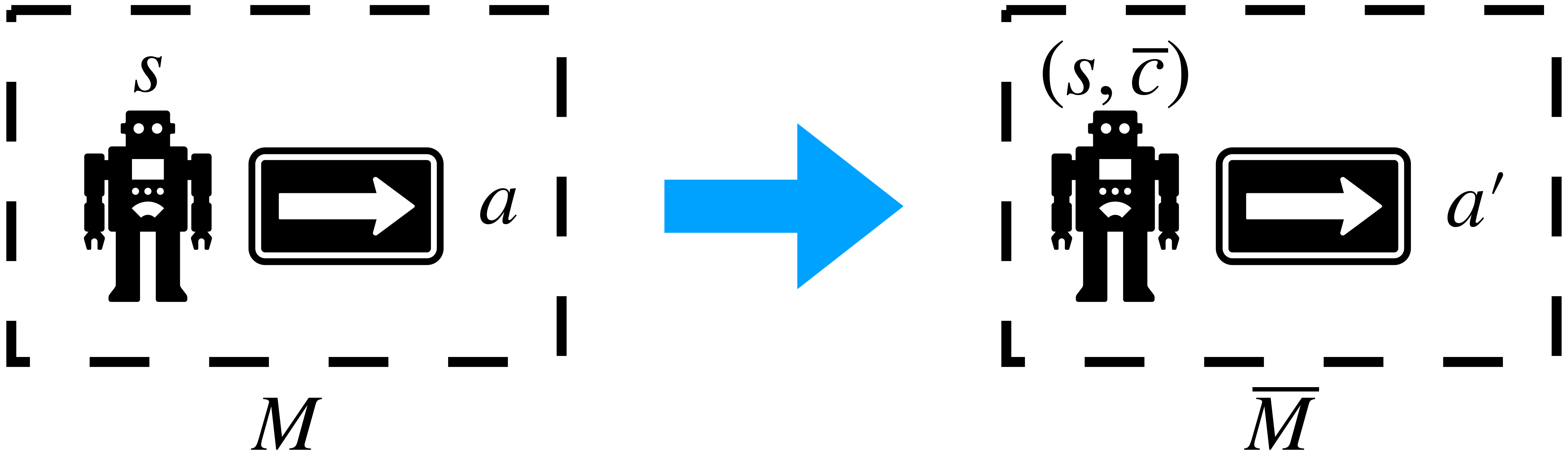


2. BFS Generate Feasible Costs

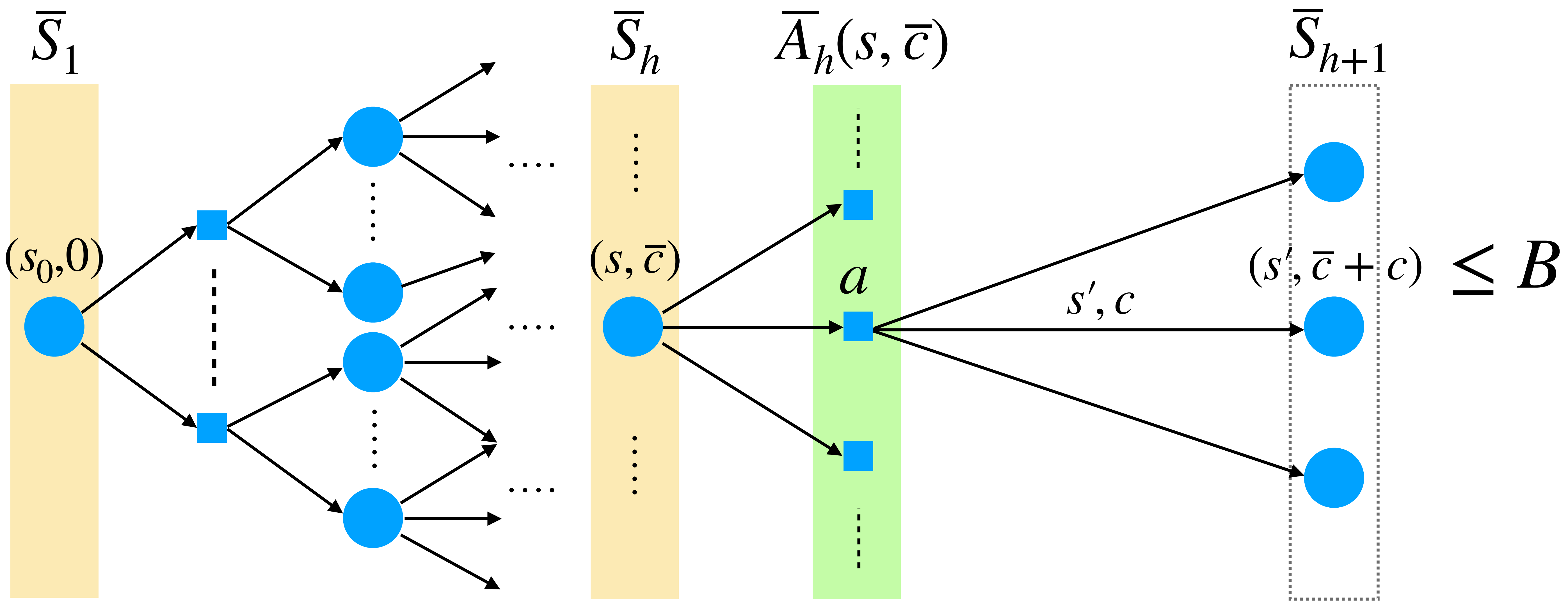


Reduction

1. State-Cost Augmentation

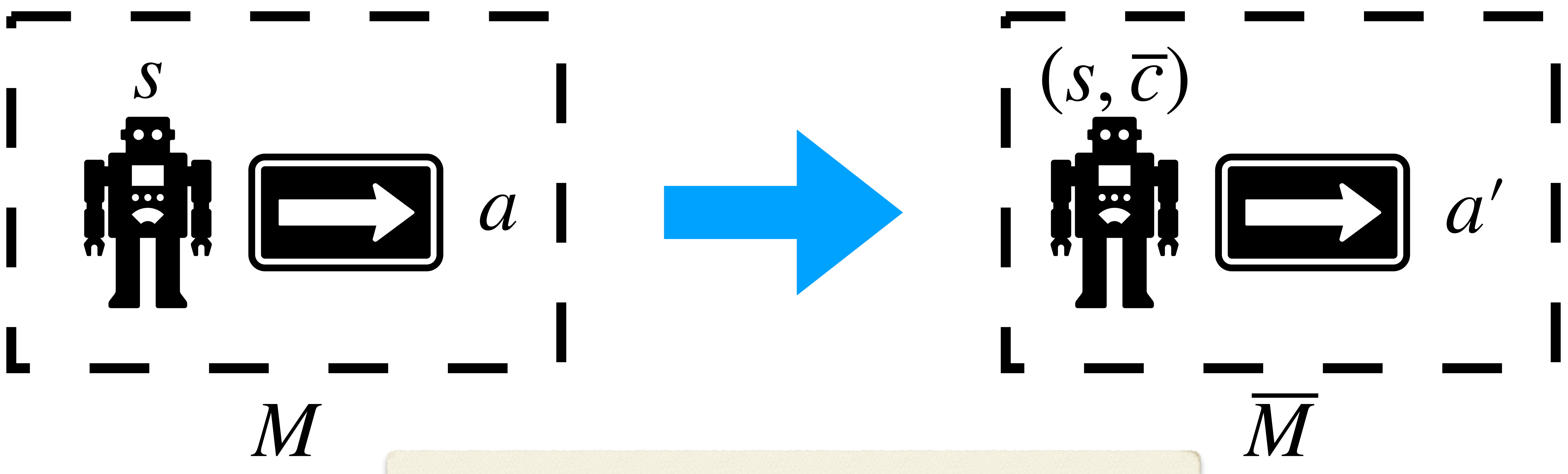


2. BFS Generate Feasible Costs



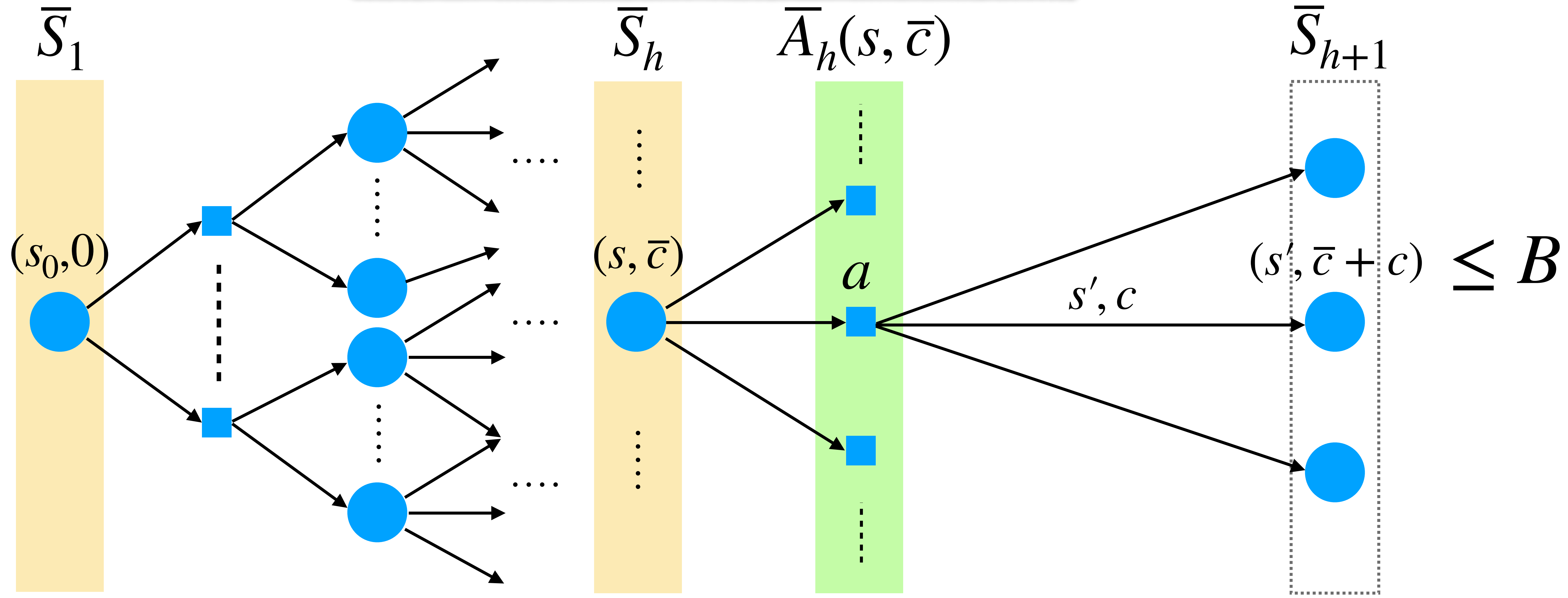
Reduction

1. State-Cost Augmentation



Solve $\overline{M}$ using RL!

2. BFS Generate Feasible Costs



Exact Results

Exact Results

$$\text{cost precision} \leq k \implies |\overline{S}| \leq SH2^{k+1}$$

Exact Results

$$\text{cost precision} \leq k \implies |\overline{S}| \leq SH2^{k+1}$$

Theorem (Fixed-Parameter Tractability): *If the cost precision $k = O(\log(SAH))$, our algorithm outputs an optimal, anytime-constrained policy in polynomial time.*

Approximate Feasibility

Approximate Feasibility

Definition 1 (Approximate Feasibility)**.** For any $\epsilon > 0$, a policy π is ϵ-additive feasible if,

$$\mathbb{P}^{\pi}_{M}\left[\forall k \in [H],\ \sum_{t=1}^{k} c_t \leq B + \epsilon\right] = 1,$$

and ϵ-relative feasible if,

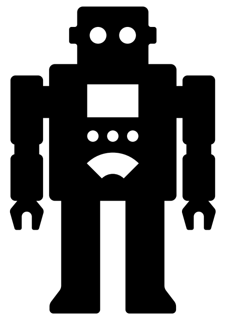
$$\mathbb{P}^{\pi}_{M}\left[\forall k \in [H],\ \sum_{t=1}^{k} c_t \leq B(1 + \epsilon)\right] = 1.$$

Approximation

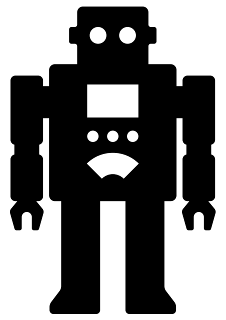
Approximation

1. *Truncate*

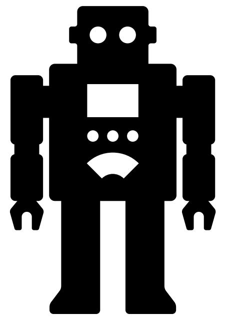
Approximation

1. Truncate $(\overline{c}, c)$ 

Approximation

1. *Truncate* $(\overline{c}, c)$  $[B - Hc^{max}, B + 1]$

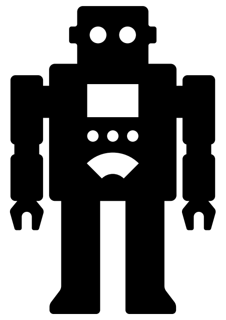
Approximation

1. Truncate $(\bar{c}, c)$  $[B - Hc^{max}, B + 1]$

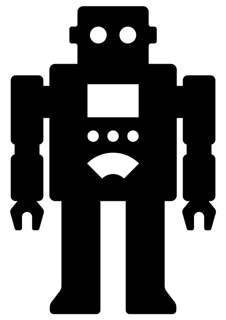
---

2. ℓ*-Discretize*

Approximation

1. *Truncate* $(\bar{c}, c)$  → $[B - Hc^{max}, B + 1]$

---

2. *ℓ-Discretize* $(\bar{c}, c)$ 

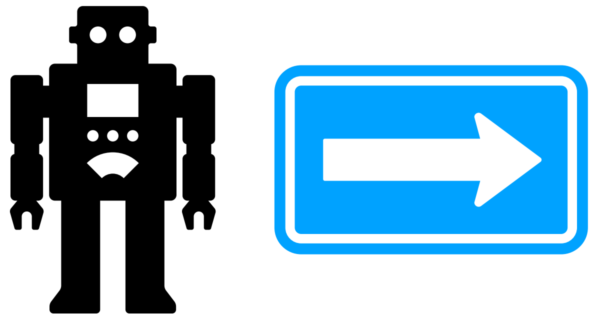
Approximation

1. *Truncate* $(\bar{c}, c) \Rightarrow [B - Hc^{max}, B + 1]$

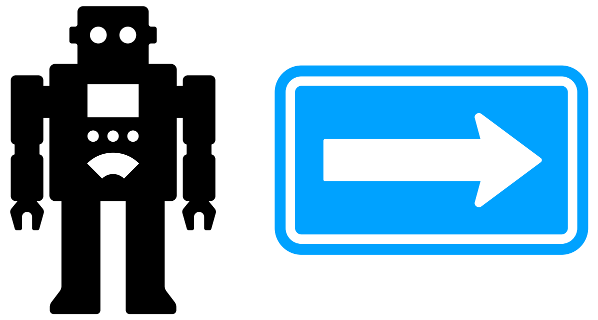
---

2. *ℓ-Discretize* $(\bar{c}, c) \Rightarrow \left\lfloor \frac{\bar{c} + c}{\ell} \right\rfloor \ell$

Approximation

1. *Truncate* $(\overline{c}, c)$  $[B - Hc^{max}, B + 1]$

---

2. *ℓ-Discretize* $(\overline{c}, c)$  $\left\lfloor \frac{\overline{c} + c}{\ell} \right\rfloor \ell = \overline{c} + \left\lfloor \frac{c}{\ell} \right\rfloor \ell$

Approximation Results

Approximation Results

$$\ell = \frac{\epsilon}{H} \implies c \leq \hat{c} + \frac{\epsilon}{H} \implies \sum_h c_h \leq B + \epsilon$$

Approximation Results

$$\ell = \frac{\epsilon}{H} \implies c \leq \hat{c} + \frac{\epsilon}{H} \implies \sum_h c_h \leq B + \epsilon$$

Theorem (Approx): *If d is constant and $c^{max} \leq poly(|M|)$, our algorithm outputs an **optimal**-value, ϵ-**feasible** policy in time $poly(|M|, \frac{1}{\epsilon})$*

**Guarantees are best-possible given hardness results.*

Approximation Results

$$\ell = \frac{\epsilon}{H} \implies c \leq \hat{c} + \frac{\epsilon}{H} \implies \sum_h c_h \leq B + \epsilon$$

Theorem (Approx): *If d is constant and $c^{max} \leq poly(|M|)$, our algorithm outputs an* ***optimal****-value, ϵ-**feasible** policy in time* $poly(|M|, \frac{1}{\epsilon})$

First poly-time algorithm for anytime and almost sure constraints!

**Guarantees are best-possible given hardness results.*

Single-Constraint FPTAS

**NeurIPS 2024*

Motivation

Motivation

1. Previous approach cannot guarantee feasibility

Motivation

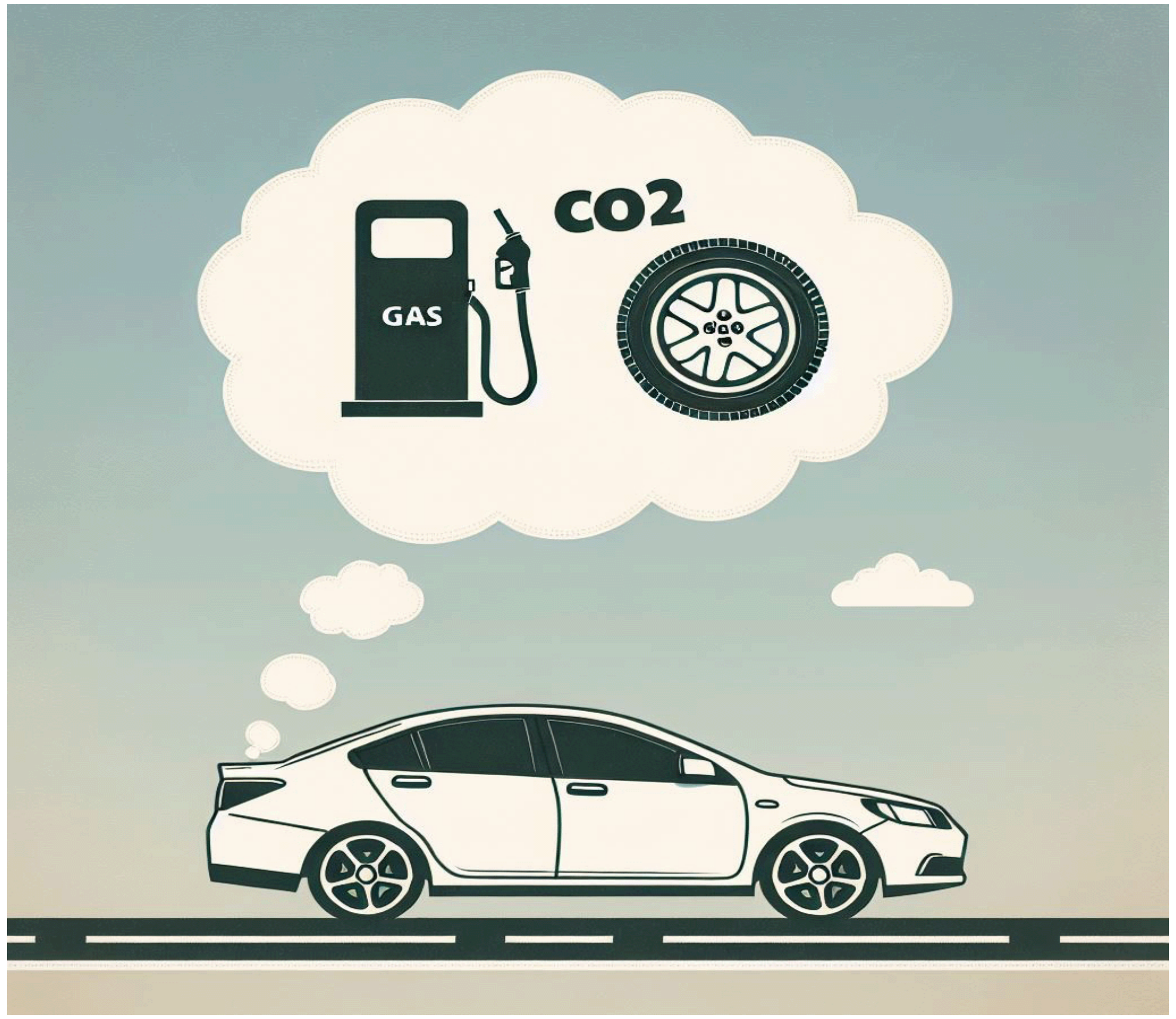
1. Previous approach cannot guarantee feasibility

2. Only works for anytime constraints

Motivation

1. Previous approach cannot guarantee feasibility

2. Only works for anytime constraints



Packing Form

Packing Form

$$\max_{\pi \in \Pi} \mathbb{E}_M^\pi \left[\sum_{h=1}^{H} r_h(s_h, a_h) \right] \quad \text{s.t.} \quad \begin{cases} C_M^\pi \leq B \\ \end{cases}$$

Packing Form

$$\max_{\pi \in \Pi} \mathbb{E}_M^{\pi} \left[\sum_{h=1}^{H} r_h(s_h, a_h) \right] \quad \text{s.t.} \quad \begin{cases} C_M^{\pi} \leq B \\ \pi \text{ deterministic} \end{cases}$$

Packing Form

$$\max_{\pi \in \Pi} \mathbb{E}_M^\pi \left[\sum_{h=1}^{H} r_h(s_h, a_h) \right] \quad \text{s.t.} \quad \begin{cases} C_M^\pi \leq B \\ \pi \text{ deterministic} \end{cases}$$

Expectation: $C_M^\pi := \mathbb{E}_M^\pi \left[\sum_{h=1}^{H} c_h \right]$

Packing Form

$$\max_{\pi \in \Pi} \mathbb{E}_M^\pi \left[\sum_{h=1}^{H} r_h(s_h, a_h) \right] \quad \text{s.t.} \quad \begin{cases} C_M^\pi \leq B \\ \pi \text{ deterministic} \end{cases}$$

Expectation: $C_M^\pi := \mathbb{E}_M^\pi \left[\sum_{h=1}^{H} c_h \right]$

Anytime: $C_M^\pi := \max_t \max_{\tau : \mathbb{P}^\pi[\tau] > 0} \sum_{h=1}^{t} c_h$

Duality

Duality

Packing (Primal)

$$\begin{aligned} \max_{\pi \in \Pi^D} \quad & V_M^\pi \\ \text{s.t.} \quad & C_M^\pi \le B \end{aligned}$$

Duality

Packing (Primal)

$$\max_{\pi \in \Pi^D} \quad V_M^\pi \qquad \text{s.t.} \quad C_M^\pi \le B$$

↓

Optimum value, but approximate cost

Duality

Packing (Primal)

$$V^* — \max_{\pi \in \Pi^D} \quad V_M^\pi$$
$$\text{s.t.} \quad C_M^\pi \leq B$$

↓

Optimum value, but approximate cost

Duality

Packing (Primal)

$$V^* = \max_{\pi \in \Pi^D} \; V_M^\pi \quad \text{s.t.} \quad C_M^\pi \leq B$$

Optimum value, but approximate cost

Covering (Dual)

$$\min_{\pi \in \Pi^D} \; C_M^\pi \quad \text{s.t.} \quad V_M^\pi \geq V^*$$

Duality

Packing (Primal)

$V^* —$
$$\max_{\pi \in \Pi^D} \quad V_M^\pi \qquad \text{s.t.} \quad C_M^\pi \leq B$$

Optimum value, but approximate cost

Covering (Dual)

$$\min_{\pi \in \Pi^D} \quad C_M^\pi \qquad \text{s.t.} \quad V_M^\pi \geq V^*$$

Optimum cost, but approximate value

Duality

Packing (Primal)

$$V^* \text{ — } \max_{\pi \in \Pi^D} \quad V_M^\pi$$
$$\text{s.t.} \quad C_M^\pi \le B$$

Optimum value, but approximate cost

Covering (Dual)

$$\min_{\pi \in \Pi^D} \quad C_M^\pi$$
$$\text{s.t.} \quad V_M^\pi \ge V^*$$

Optimum cost, but approximate value

Feasible!

Value-Demand Augmentation

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad & C_h^\pi(\tau_h) \\ \text{s.t.} \quad & V_h^\pi(\tau_h) \geq v \end{aligned}$$

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}_h^*(s,v) = \min_{\pi \in \Pi^D} \quad C_h^\pi(\tau_h) \qquad \text{s.t.} \quad V_h^\pi(\tau_h) \geq v$$

Dual = $\overline{C}_1^*(s_0, V^*)$

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}^*_h(s, v) = \min_{\pi \in \Pi^D} \quad C^\pi_h(\tau_h) \quad \text{s.t.} \quad V^\pi_h(\tau_h) \geq v$$

➡ Dual = $\overline{C}^*_1(s_0, V^*)$

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}^*_h(s,v) = \min_{\pi \in \Pi^D} \quad C^\pi_h(\tau_h) \quad \text{s.t.} \quad V^\pi_h(\tau_h) \geq v$$

Dual = $\overline{C}^*_1(s_0, V^*)$

$\overline{S} = S \times \mathcal{V}$ — *all possible values*

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad C_h^\pi(\tau_h) \qquad \text{s.t.} \quad V_h^\pi(\tau_h) \geq v$$

Dual = $\overline{C}_1^*(s_0, V^*)$

$\overline{S} = S \times \mathcal{V}$ — *all possible values*

Invariant: $v \leq \overline{V}_h^\pi(s, v)$

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}^*_h(s,v) = \min_{\pi \in \Pi^D} \quad C^\pi_h(\tau_h) \quad \text{s.t.} \quad V^\pi_h(\tau_h) \geq v$$

Dual = $\overline{C}^*_1(s_0, V^*)$

$\overline{S} = S \times \mathcal{V}$ — *all possible values*

Invariant: $v \leq \overline{V}^\pi_h(s,v) = r_h(s,a) + \sum_{s'} P_h(s' \mid s,a)\overline{V}^\pi_{h+1}(s', v_{s'})$ **PE**

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad C_h^\pi(\tau_h) \qquad \text{s.t.} \quad V_h^\pi(\tau_h) \geq v$$

Dual = $\overline{C}_1^*(s_0, V^*)$

$\overline{S} = S \times \mathcal{V}$ — *all possible values*

Invariant: $v \leq \overline{V}_h^\pi(s, v) = r_h(s, a) + \sum_{s'} P_h(s' \mid s, a) \overline{V}_{h+1}^\pi(s', v_{s'})$ **PE**

$$\overline{\mathcal{A}}_h(s, v) := \left\{ (a, \mathbf{v}) \in \mathcal{A} \times \mathcal{V}^S \mid r_h(s, a) + \sum_{s'} P_h(s' \mid s, a) v_{s'} \geq v \right\}$$

Outer Algorithm

Outer Algorithm

1. *Solve:*

Outer Algorithm

Anytime Constraints

1. Solve: $$\overline{C}_h^*(s,v) = \min_{a,\mathbf{v} \in \mathcal{A}_h(s,v)} c_h(s,a) + \underbrace{\max_{s'} \overline{C}_h^*(s', v_{s'})}_{\text{Anytime Constraints}}$$

Outer Algorithm

1. Solve:

$$\overline{C}_h^*(s, v) = \min_{a, \mathbf{v} \in \mathcal{A}_h(s,v)} c_h(s, a) + \sum_{s'} \underbrace{P_h(s' \mid s, a)\overline{C}_{h+1}^*(s', v_{s'})}_{\textit{Expectation Constraints}}$$

Outer Algorithm

1. Solve:

$$\overline{C}_h^*(s, v) = \min_{a, \mathbf{v} \in \mathcal{A}_h(s,v)} c_h(s, a) + \sum_{s'} P_h(s' \mid s, a)\overline{C}_{h+1}^*(s', v_{s'})$$

---

2. Output:

Outer Algorithm

1. Solve: $$\overline{C}_h^*(s, v) = \min_{a, \mathbf{v} \in \mathcal{A}_h(s,v)} c_h(s, a) + \sum_{s'} P_h(s' \mid s, a)\overline{C}_{h+1}^*(s', v_{s'})$$

2. Output: $$V_M^* = \max\{v \in \mathcal{V} \mid \overline{C}_1^*(s_0, v) \leq B\}$$

Outer Algorithm

1. Solve:

$$\overline{C}_h^*(s, v) = \min_{a, \mathbf{v} \in \mathcal{A}_h(s,v)} c_h(s, a) + \sum_{s'} P_h(s' \mid s, a)\overline{C}_{h+1}^*(s', v_{s'})$$

---

2. Output:

$$V_M^* = \max \{v \in \mathcal{V} \mid \overline{C}_1^*(s_0, v) \leq B\}$$

Feasible!

Outer Algorithm

1. Solve:

$$\overline{C}_h^*(s,v) = \min_{a,\mathbf{v} \in \mathcal{A}_h(s,v)} c_h(s,a) + \sum_{s'} P_h(s' \mid s,a)\overline{C}_{h+1}^*(s', v_{s'})$$

Exponential!

---

2. Output:

$$V_M^* = \max \{ v \in \mathcal{V} \mid \overline{C}_1^*(s_0, v) \le B \}$$

Feasible!

Solving $\overline{M}$ Fast

Solving $\overline{M}$ Fast

Optimality Equations

$$\overline{C}_h^*(s, v) = \min_{(a,\mathbf{v})} c_h(s,a) + \sum_{s'} P_h(s' \mid s,a)\overline{C}_h^*(s, v_{s'})$$

$$\text{s.t. } r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) v_{s'} \geq v$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v} \in \mathcal{V}^S} \sum_{s'} P_h(s' \mid s, a) \overline{C}^*_h(s, v_{s'})$$

$$\sum_{s'} P_h(s' \mid s, a) v_{s'} \geq v$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v} \in \mathcal{V}^S} \sum_{s'} P_h(s' \mid s, a) \overline{C}^*_h(s, v_{s'})$$

$$\sum_{s'} \quad p_{s'} \quad v_{s'} \quad \geq v$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v} \in \mathcal{V}^S} \sum_{s'} w_{s'} \overline{C}_h^*(s, v_{s'})$$

$$\sum_{s'} p_{s'} v_{s'} \geq v$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v}\in\mathcal{V}^S} \sum_{s'} w_{s'} \, f(v_{s'})$$

$$\sum_{s'} p_{s'} \, v_{s'} \geq v$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v}\in\mathcal{V}^S} \sum_{s'} w_{s'} \quad f(v_{s'})$$

$$\sum_{s'} p_{s'} \quad v_{s'} \quad \geq v$$

↔

Knapsack Problem

$$\min_{x\in X^n} \sum_i w_i x_i$$

$$\text{s.t.} \sum_i p_i x_i \geq P$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v}\in\mathcal{V}^S} \sum_{s'} w_{s'} \quad f(v_{s'})$$

$$\sum_{s'} p_{s'} \quad v_{s'} \quad \geq v$$

Knapsack Problem

$$\min_{x\in X^n} \sum_i w_i x_i$$

$$\text{s.t.} \sum_i p_i x_i \geq P$$

Knapsack Approx!

$$MC(s', p)$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v}\in\mathcal{V}^S} \sum_{s'} w_{s'} \; f(v_{s'})$$

$$\sum_{s'} p_{s'} \; v_{s'} \geq v$$

Knapsack Problem

$$\min_{x\in X^n} \sum_i w_i x_i$$

$$\text{s.t.} \sum_i p_i x_i \geq P$$

Knapsack Approx!

$$\sum_{i=1}^{s'-1} p_i v_i$$

$$MC(s', p)$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v}\in\mathcal{V}^S} \sum_{s'} w_{s'} \quad f(v_{s'})$$

$$\sum_{s'} p_{s'} \quad v_{s'} \quad \geq v$$

Knapsack Problem

$$\min_{x\in X^n} \sum_i w_i x_i$$

$$\text{s.t.} \sum_i p_i x_i \geq P$$

Knapsack Approx!

$$\sum_{i=1}^{s'-1} p_i v_i$$

$$MC(s', p) = \min_{v_{s'}\in\mathcal{V}} \quad w_{s'} \quad f(v_{s'})$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v}\in\mathcal{V}^S} \sum_{s'} w_{s'} \quad f(v_{s'})$$

$$\sum_{s'} p_{s'} \quad v_{s'} \quad \geq v$$

Knapsack Problem

$$\min_{x\in X^n} \sum_i w_i x_i$$

$$\text{s.t.} \sum_i p_i x_i \geq P$$

Knapsack Approx!

$$\sum_{i=1}^{s'-1} p_i v_i$$

$$MC(s', p) = \min_{v_{s'}\in\mathcal{V}} w_{s'} \quad f(v_{s'}) \quad + MC(s'+1, p + \quad p_{s'} \quad v_{s'})$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v}\in\mathcal{V}^S} \sum_{s'} w_{s'} \, f(v_{s'})$$
$$\sum_{s'} p_{s'} \, v_{s'} \geq v$$

Knapsack Problem

$$\min_{x\in X^n} \sum_i w_i x_i$$
$$\text{s.t.} \sum_i p_i x_i \geq P$$

Knapsack Approx!

$$\sum_{i=1}^{s'-1} p_i v_i \qquad\qquad \sum_{i=1}^{s'} p_i v_i$$

$$MC(s', p) = \min_{v_{s'}\in\mathcal{V}} w_{s'} \, f(v_{s'}) + MC(s'+1, p + p_{s'} \, v_{s'})$$

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v}\in\mathcal{V}^S} \sum_{s'} w_{s'} \quad f(v_{s'})$$
$$\sum_{s'} p_{s'} \quad v_{s'} \quad \geq v$$

Knapsack Problem

$$\min_{x\in X^n} \sum_i w_i x_i$$
$$\text{s.t.} \sum_i p_i x_i \geq P$$

Knapsack Approx!

$$MC(s', p) = \min_{v_{s'}\in\mathcal{V}} w_{s'} \quad f(v_{s'}) \quad + MC(s'+1, p + \quad p_{s'} \quad v_{s'})$$

Round for approx

Solving $\overline{M}$ Fast

Optimality Equations

$$\min_{\mathbf{v} \in \mathcal{V}^S} \sum_{s'} w_{s'} \quad f(v_{s'})$$

$$\sum_{s'} p_{s'} \quad v_{s'} \quad \geq v$$

Knapsack Problem

$$\min_{x \in X^n} \sum_i w_i x_i$$

$$\text{s.t.} \sum_i p_i x_i \geq P$$

Knapsack Approx!

$$MC(s', p) = \min_{v_{s'} \in \mathcal{V}} P_h(s' \mid s, a)\overline{C}^*_{h+1}(s', v_{s'}) + MC(s'+1, p + P_h(s' \mid s, a)v_{s'})$$

Round for approx

Time-Space Rounding

Time-Space Rounding

Round $v's$ down $\Longrightarrow$ cost goes down!

Time-Space Rounding

Round $v's$ down $\implies$ cost goes down!

Feasible!

Time-Space Rounding

Round $v's$ down $\Longrightarrow$ cost goes down!

Feasible!

Rounding $v's$ causes error over time

Time-Space Rounding

Round $v's$ down $\implies$ cost goes down!

Feasible!

Rounding $v's$ causes error over time → $\hat{V}^{\pi} \geq V^{*} - \ell H$

Time-Space Rounding

Round $v's$ down $\Longrightarrow$ cost goes down!

Feasible!

Rounding $v's$ causes error over **time** → $\hat{V}^{\pi} \geq V^{*} - \ell H$

Rounding $p's$ causes error over **space**

Time-Space Rounding

Round $v's$ down $\Longrightarrow$ cost goes down!

Feasible!

Rounding $v's$ causes error over **time** → $\hat{V}^{\pi} \geq V^{*} - \ell H$

Rounding $p's$ causes error over **space** → $\hat{V}^{\pi} \geq V^{*} - \ell SH$

Time-Space Rounding

Round $v's$ down $\implies$ cost goes down! **Feasible!**

Rounding $v's$ causes error over time ➡ $\hat{V}^{\pi} \geq V^{*} - \ell H$

Rounding $p's$ causes error over **space** ➡ $\hat{V}^{\pi} \geq V^{*} - \ell SH$

$$\ell = \frac{\epsilon}{SH} \implies \hat{V}^{\pi} \geq V^{*} - \epsilon$$

Results

Results

Theorem (FPTAS): *If the rewards are poly-bounded, our algorithm outputs a* ***feasible*** *policy with value* $V^* - \epsilon$ *in time* $poly(|M|, \frac{1}{\epsilon})$

**Guarantees are best-possible given hardness results.*

Results

Theorem (FPTAS): *If the rewards are poly-bounded, our algorithm outputs a* ***feasible*** *policy with value* $V^* - \epsilon$ *in time* $poly(|M|, \frac{1}{\epsilon})$

First ever poly-time algorithm for ***deterministic****, expectation-constrained policies!*

**Guarantees are best-possible given hardness results.*

Multi-Constraint Bicriteria

**ICML 2025*

Motivation

Motivation

Full Problem

$$\begin{aligned} \max_{\pi \in \Pi^D} \quad & V^\pi \\ \text{s.t.} \quad & C_1^\pi \leq B_1 \\ & C_2^\pi \leq B_2 \\ & \vdots \\ & C_m^\pi \leq B_m \end{aligned}$$

Motivation

Full Problem

$$\begin{aligned} \max_{\pi \in \Pi^D} \quad & V^\pi \\ \text{s.t.} \quad & C_1^\pi \leq B_1 \\ & C_2^\pi \leq B_2 \\ & \vdots \\ & C_m^\pi \leq B_m \end{aligned}$$

Expectation: $$\mathbb{E}_M^\pi \left[\sum_{h=1}^{H} c_h \right] \leq B$$

Chance: $$\mathbb{P}_M^\pi \left[\sum_{h=1}^{H} c_h > B \right] \leq \delta$$

Almost Sure: $$\mathbb{P}_M^\pi \left[\sum_{h=1}^{H} c_h \leq B \right] = 1$$

Anytime: $$\mathbb{P}_M^\pi \left[\forall t,\ \sum_{h=1}^{t} c_h \leq B \right] = 1$$

Motivation

Full Problem

$$\begin{aligned} \max_{\pi \in \Pi^D} \quad & V^\pi \\ \text{s.t.} \quad & C_1^\pi \leq B_1 \\ & C_2^\pi \leq B_2 \\ & \vdots \\ & C_m^\pi \leq B_m \end{aligned}$$

Expectation: $\mathbb{E}_M^\pi \left[\sum_{h=1}^{H} c_h \right] \leq B$

Chance: $\mathbb{P}_M^\pi \left[\sum_{h=1}^{H} c_h > B \right] \leq \delta$

Almost Sure: $\mathbb{P}_M^\pi \left[\sum_{h=1}^{H} c_h \leq B \right] = 1$

Anytime: $\mathbb{P}_M^\pi \left[\forall t, \ \sum_{h=1}^{t} c_h \leq B \right] = 1$

Can we create a framework that works for ***any combination*** *of constraints?*

Budget Augmentation

Budget Augmentation

Full Form

$$
\begin{aligned}
\max_{\pi \in \Pi^D} \quad & V^\pi \\
\text{s.t.} \quad & C_1^\pi \leq B_1 \\
& C_2^\pi \leq B_2 \\
& \vdots \\
& C_m^\pi \leq B_m
\end{aligned}
$$

Budget Augmentation

Full Form

$$\begin{aligned} \max_{\pi \in \Pi^D} \quad & V^\pi \\ \text{s.t.} \quad & C_1^\pi \leq B_1 \\ & C_2^\pi \leq B_2 \\ & \vdots \\ & C_m^\pi \leq B_m \end{aligned}$$

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{V}_h^*(s,b) = \max_{\pi \in \Pi^D} \quad & V_h^\pi(\tau_h) \\ \text{s.t.} \quad & C_h^\pi(\tau_h) \leq b \end{aligned}$$

Budget Augmentation

Full Form

$$\begin{aligned} \max_{\pi \in \Pi^D} \quad & V^\pi \\ \text{s.t.} \quad & C_1^\pi \le B_1 \\ & C_2^\pi \le B_2 \\ & \vdots \\ & C_m^\pi \le B_m \end{aligned}$$

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{V}_h^*(s,b) = \max_{\pi \in \Pi^D} \quad & V_h^\pi(\tau_h) \\ \text{s.t.} \quad & C_h^\pi(\tau_h) \le b \end{aligned}$$

Primal = $\overline{V}_1^*(s_0, B)$

Budget Augmentation

Full Form

$$\begin{aligned} \max_{\pi \in \Pi^D} \quad & V^\pi \\ \text{s.t.} \quad & C_1^\pi \leq B_1 \\ & C_2^\pi \leq B_2 \\ & \vdots \\ & C_m^\pi \leq B_m \end{aligned}$$

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{V}_h^*(s,b) = \max_{\pi \in \Pi^D} \quad & V_h^\pi(\tau_h) \\ \text{s.t.} \quad & C_h^\pi(\tau_h) \leq b \end{aligned}$$

Primal = $\overline{V}_1^*(s_0, B)$

Use previous approach but with rounding up!

Constraint Assumptions

Constraint Assumptions

1. *Recursion:*

Constraint Assumptions

1. Recursion: $C_h^\pi(\tau_h) = c_h(s,a) + f_{s'} g(P_h(s' \mid s,a)) C_{h+1}^\pi(s')$

Constraint Assumptions

1. Recursion: $$C_h^\pi(\tau_h) = c_h(s,a) + f_{s'} g(P_h(s' \mid s,a)) C_{h+1}^\pi(s')$$

Required for inner DP

Constraint Assumptions

	Exp	AS
f	$\sum_{s'}$	$\max_{s'}$
g	id	$[x > 0]$

1. Recursion:

$$C_h^\pi(\tau_h) = c_h(s,a) + f_{s'} g(P_h(s' \mid s,a)) C_{h+1}^\pi(s')$$

Required for inner DP

Constraint Assumptions

1. Recursion:

$$C_h^\pi(\tau_h) = c_h(s,a) + f_{s'} g(P_h(s' \mid s,a)) C_{h+1}^\pi(s')$$

	Exp	AS
f	$\sum_{s'}$	$\max_{s'}$
g	id	$[x > 0]$

Required for inner DP

Constraint Assumptions

1. *Recursion:*

$$C_h^\pi(\tau_h) = c_h(s,a) + f_{s'} g(P_h(s' \mid s,a)) C_{h+1}^\pi(s')$$

	Exp	AS
f	$\sum_{s'}$	$\max_{s'}$
g	id	$[x > 0]$

Required for inner DP

2. *1-Lipschitz:*

Constraint Assumptions

1. Recursion: $C_h^\pi(\tau_h) = c_h(s,a) + f_{s'} g(P_h(s' \mid s,a)) C_{h+1}^\pi(s')$

	Exp	AS
f	$\sum_{s'}$	$\max_{s'}$
g	id	$[x > 0]$

Required for inner DP

---

2. 1-Lipschitz: $f(x, \mathrm{round}(y)) \leq f(x, y + \ell) \leq f(x, y) + \ell$

Constraint Assumptions

1. Recursion:

$$C_h^\pi(\tau_h) = c_h(s,a) + f_{s'} g(P_h(s' \mid s,a)) C_{h+1}^\pi(s')$$

	Exp	AS
f	$\sum_{s'}$	$\max_{s'}$
g	id	$[x > 0]$

Required for inner DP

2. 1-Lipschitz:

$$f(x, \text{round}(y)) \leq f(x, y + \ell) \leq f(x, y) + \ell$$

Required for rounding error analysis

Results

Results

Theorem (Bicriteria): *Our algorithm computes an **optimal**-value, ϵ-**feasible** policy in **polynomial time**, so long as the costs are poly-bounded and satisfy the SR condition.*

**Guarantees are best-possible given hardness results.*

Results

Theorem (Bicriteria): *Our algorithm computes an **optimal**-value, ϵ-**feasible** policy in **polynomial time**, so long as the costs are poly-bounded and satisfy the SR condition.*

*Includes **all** classical constraints!*

**Guarantees are best-possible given hardness results.*

Results

Theorem (Bicriteria): *Our algorithm computes an **optimal**-value, ϵ-**feasible** policy in **polynomial time**, so long as the costs are poly-bounded and satisfy the SR condition.*

*Includes **all** classical constraints!*

*First ever poly-time algorithm for **chance** constraints and **non-homogenous** constraints!*

*Guarantees are best-possible given hardness results.

Future Directions

1. Beyond Worst-case Analysis for all works
   (especially POMDPs for defense and anytime constraints)

2. Submodular Constrained Reinforcement Learning

3. Optimal learning under constraints.

Thank you!

Backup

Motivating Example

Motivating Example

Motivating Example

1. Robust to visual noise (ash)

Motivating Example

1. Robust to visual noise (ash)

2. Robust to other rescue vehicles

Motivating Example

1. Robust to visual noise (ash)

2. Robust to other rescue vehicles

3. Coordinate well with teammates

Motivating Example

1. Robust to visual noise (ash)

2. Robust to other rescue vehicles

3. Coordinate well with teammates

1. Effective fuel management

Motivating Example

1. Robust to visual noise (ash)
2. Robust to other rescue vehicles
3. Coordinate well with teammates

1. Effective fuel management
2. Avoids dangerous terrain (lava)

Motivating Example

1. Robust to visual noise (ash)
2. Robust to other rescue vehicles
3. Coordinate well with teammates

1. Effective fuel management
2. Avoids dangerous terrain (lava)
3. Balances risks of difficult terrain

Framework Extensions

Framework Extensions

1. Multiple agents

Framework Extensions

1. Multiple agents
2. Infinite discounting

Framework Extensions

1. Multiple agents
2. Infinite discounting
3. Stochastic costs

Framework Extensions

1. Multiple agents
2. Infinite discounting
3. Stochastic costs
    1. Discrete

Framework Extensions

1. Multiple agents
2. Infinite discounting
3. Stochastic costs
    1. Discrete
    2. Bounded Continuous

Framework Extensions

1. Multiple agents
2. Infinite discounting
3. Stochastic costs
    1. Discrete
    2. Bounded Continuous
4. Continuous States

Chance Constraints

Chance Constraints

1. Use Discretized $\hat{M}$ from anytime constraints section

Chance Constraints

1. Use Discretized $\hat{M}$ from anytime constraints section
2. Define $C_h^\pi(s, \overline{c}) = \mathbb{P}^\pi \left[\exists k, \overline{c} + \sum_{t=h}^{k} c_t > B \right]$

Chance Constraints

1. Use Discretized $\hat{M}$ from anytime constraints section
2. Define $C_h^\pi(s,\overline{c}) = \mathbb{P}^\pi\left[\exists k, \overline{c} + \sum_{t=h}^{k} c_t > B\right]$ satisfies,

$$C_h^\pi(s,\overline{c}) = [\overline{c} + c_h(s,a) > B] + \sum_{s'} P_h(s' \mid s,a) C_{h+1}^\pi(s, \overline{c} + c_h(s,a))$$

Chance Constraints

1. Use Discretized $\hat{M}$ from anytime constraints section
2. Define $C_h^\pi(s, \overline{c}) = \mathbb{P}^\pi \left[\exists k, \overline{c} + \sum_{t=h}^{k} c_t > B \right]$ satisfies,

$$C_h^\pi(s, \overline{c}) = \underbrace{[\overline{c} + c_h(s, a) > B]}_{\text{New } c'_h((s, \overline{c}), a)} + \sum_{s'} P_h(s' \mid s, a) C_{h+1}^\pi(s, \overline{c} + c_h(s, a))$$

Action Space

Action Space

Policy Evaluation Equation: $C_h^\pi(s) = c_h(s, a) + \sum_{s'} P_h(s' \mid s, a) C_{h+1}^\pi(s')$

Action Space

Policy Evaluation Equation: $C_h^{\pi}(s) = c_h(s,a) + \sum_{s'} P_h(s' \mid s,a) C_{h+1}^{\pi}(s')$

Same form as before!

$$\overline{\mathcal{A}}_h(s,b) := \left\{ (a, \mathbf{b}) \in \mathcal{A} \times \mathbb{R}^S \;\middle|\; c_h(s,a) + \sum_{s'} P_h(s' \mid s,a) b_{s'} \leq b \right\}$$

Budget Augmentation

Budget Augmentation

Intuition: Build $\overline{M}$ satisfying,

Budget Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{V}^*_h(s, b) = \max_{\pi \in \Pi^D} \quad & V^\pi_h(\tau_h) \\ \text{s.t.} \quad & C^\pi_h(\tau_h) \leq b \end{aligned}$$

Budget Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{V}^*_h(s,b) = \max_{\pi \in \Pi^D} \quad V^\pi_h(\tau_h)$$
$$\text{s.t.} \quad C^\pi_h(\tau_h) \leq b$$

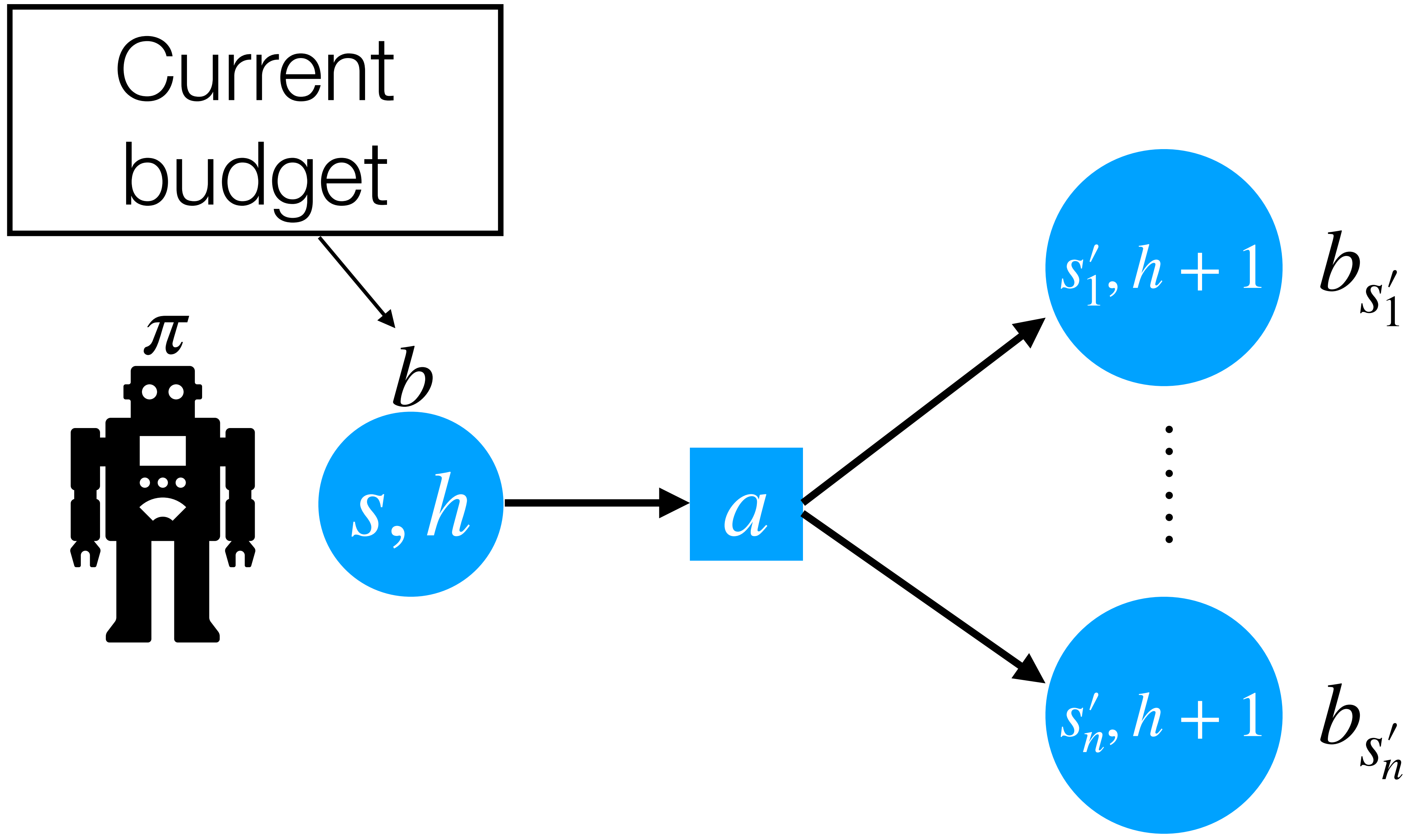
$\Rightarrow$ Primal = $\overline{V}^*_1(s_0, B)$

Budget Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{V}^*_h(s,b) = \max_{\pi \in \Pi^D} \quad V^\pi_h(\tau_h) \quad \text{s.t.} \quad C^\pi_h(\tau_h) \le b$$

Primal = $\overline{V}^*_1(s_0, B)$



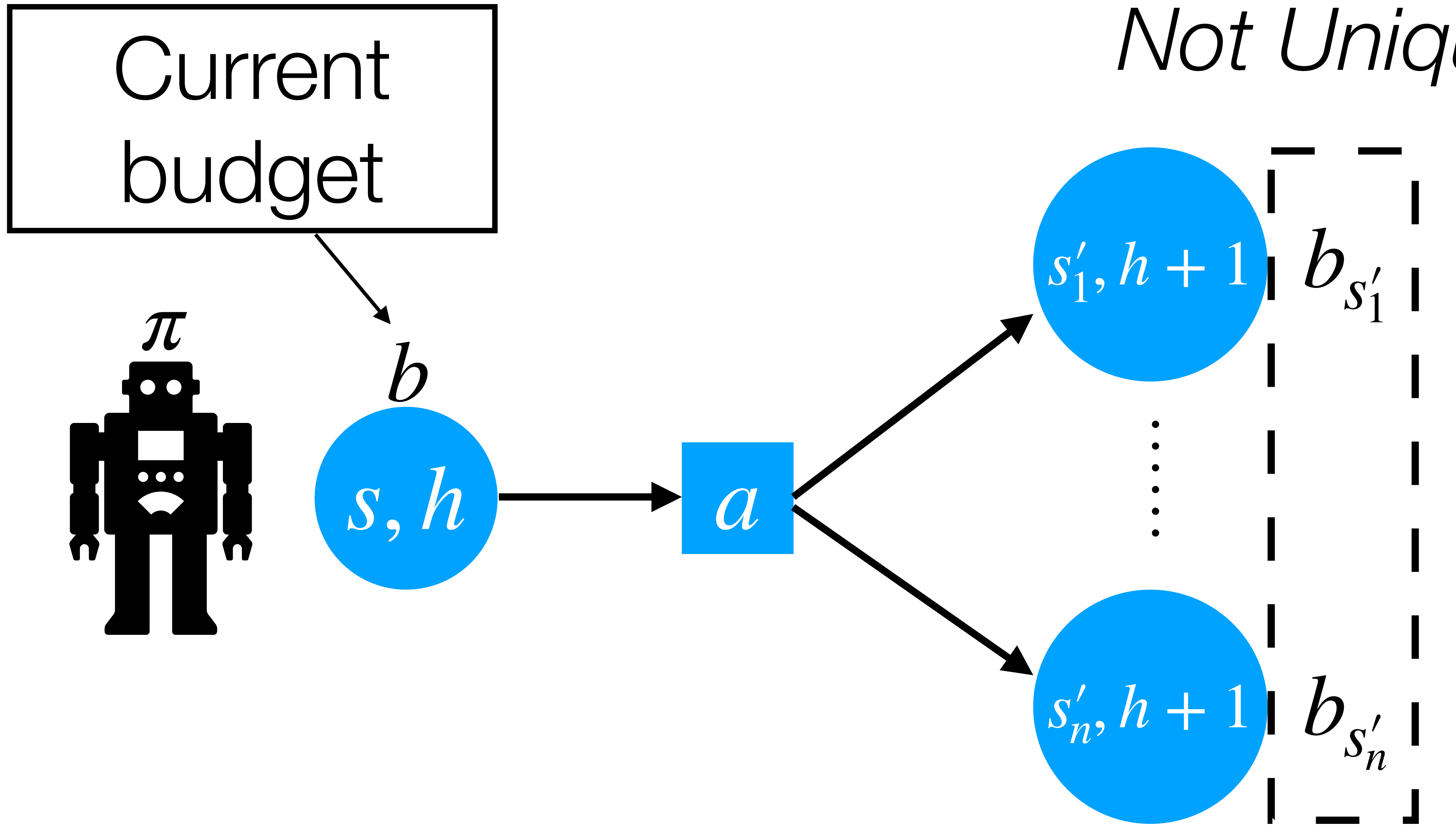
Budget Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{V}^*_h(s,b) = \max_{\pi \in \Pi^D} \quad & V^\pi_h(\tau_h) \\ \text{s.t.} \quad & C^\pi_h(\tau_h) \le b \end{aligned}$$

Primal = $\overline{V}^*_1(s_0, B)$

Not Unique!

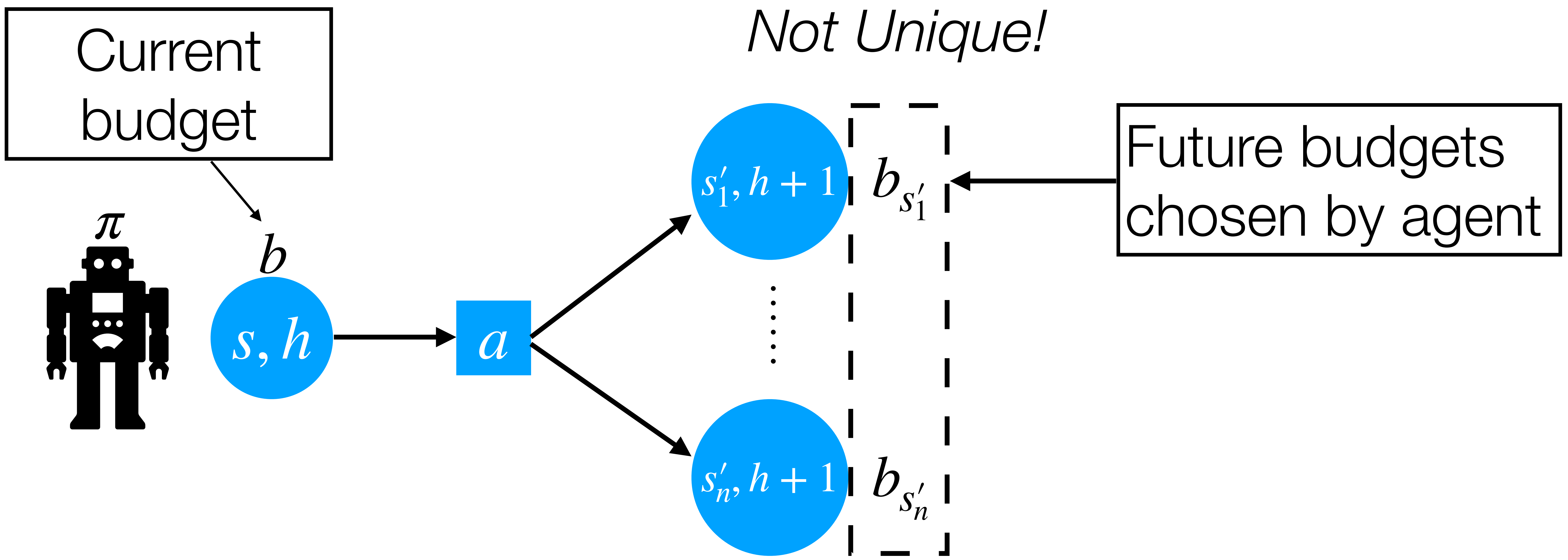


Budget Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{V}^*_h(s,b) = \max_{\pi \in \Pi^D} \; V^\pi_h(\tau_h) \quad \text{s.t.} \quad C^\pi_h(\tau_h) \leq b$$

Primal = $\overline{V}^*_1(s_0, B)$

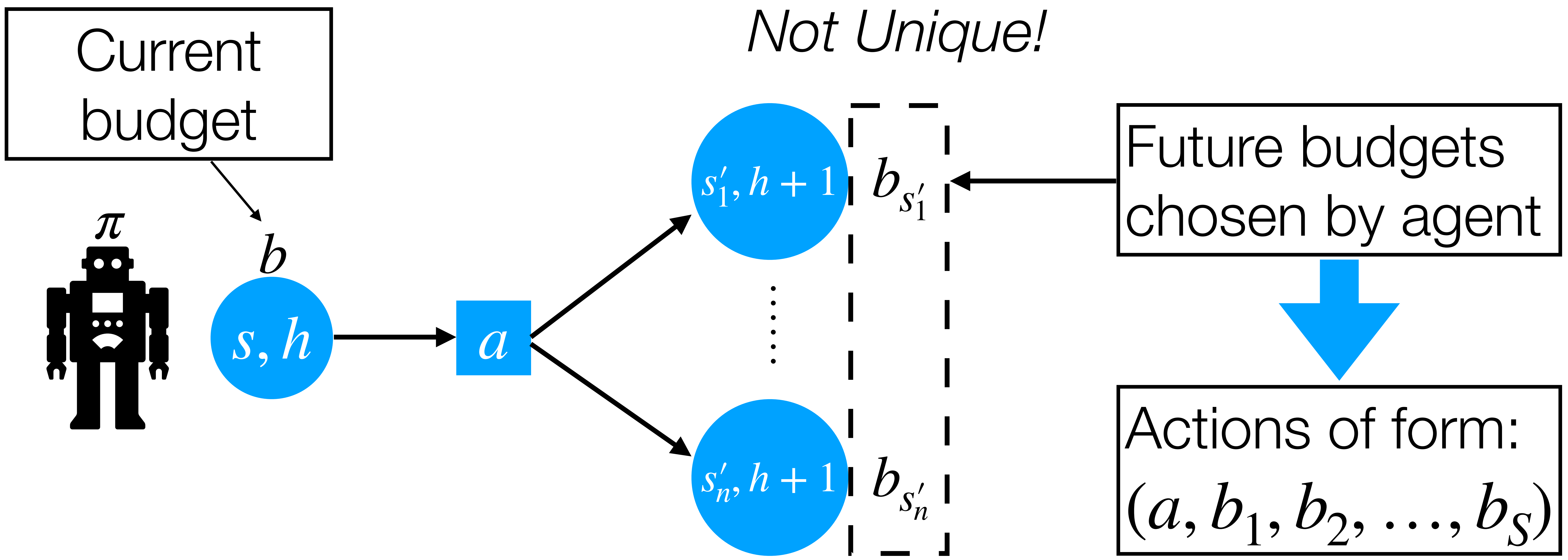


Budget Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{V}^*_h(s,b) = \max_{\pi \in \Pi^D} \quad V^\pi_h(\tau_h) \quad \text{s.t.} \quad C^\pi_h(\tau_h) \leq b$$

Primal = $\overline{V}^*_1(s_0, B)$



Definition 1 (TSR). We call a cost criterion C *time-recursive* (TR) if for any cMDP M and policy $\pi \in \Pi^D$, π's cost decomposes recursively into $C_M^\pi = C_1^\pi(s_0)$. Here, $C_{H+1}^\pi(\cdot) = \mathbf{0}$ and for any $h \in [H]$ and $\tau_h \in \mathcal{H}_h$,

$$C_h^\pi(\tau_h) = c_h(s,a) + f\left(\left(P_h(s' \mid s,a), C_{h+1}^\pi\left(\tau_h, a, s'\right)\right)_{s' \in P_h(s,a)} \right), \tag{TR}$$

where $s = s_h(\tau_h)$, $a = \pi_h(\tau_h)$, and f is a non-decreasing function[1] computable in $O(S)$ time. For technical reasons, we also require that $f(x) = \infty$ whenever $\infty \in x$.

We further say C is *time-space-recursive* (TSR) if the f term above is equal to $g_h^{\tau_h,a}(1)$. Here, $g_h^{\tau_h,a}(S+1) = 0$ and for any $t \leq S$,

$$g_h^{\tau_h,a}(t) = \alpha\left(\beta\left(P_h(t \mid s,a), C_{h+1}^\pi\left(\tau_h, a, t\right)\right), g_h^{\tau_h,a}(t+1)\right), \tag{SR}$$

where α is a non-decreasing function, and both α, β are computable in $O(1)$ time. We also assume that $\alpha(\cdot, \infty) = \infty$, and β satisfies $\alpha(\beta(0, \cdot), x) = x$ to match f's condition.

Generalization

Generalization

Recursive cost optimization suffices for our algorithm

Generalization

Recursive cost optimization suffices for our algorithm

Assumption [time-space recursive]: *the optimal cost is computable recursively over both **time** and state **space***

Generalization

Recursive cost optimization suffices for our algorithm

Assumption [time-space recursive]: *the optimal cost is computable recursively over both* ***time*** *and state* ***space***

**holds for expectation, almost sure, and anytime constraints*

Action Space

Action Space

Policy Evaluation Equation: $V_h^\pi(s) = r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) V_{h+1}^\pi(s')$

Action Space

Policy Evaluation Equation: $V_h^\pi(s) = r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) V_{h+1}^\pi(s')$

Guarantee demand by:

Action Space

Policy Evaluation Equation: $V_h^\pi(s) = r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) V_{h+1}^\pi(s')$

Guarantee demand by:

1. $\forall i,\ V_{h+1}^\pi(s_i') \geq v_{s_i'}$

Action Space

Policy Evaluation Equation: $V_h^\pi(s) = r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) V_{h+1}^\pi(s')$

Guarantee demand by:

1. $\forall i,\ V_{h+1}^\pi(s_i') \geq v_{s_i'}$
2. $r_h(s,a) + \sum_i P_h(s_i' \mid s,a) v_{s_i'} \geq v$

Action Space

Policy Evaluation Equation: $V_h^\pi(s) = r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) V_{h+1}^\pi(s')$

Guarantee demand by:

1. $\forall i,\ V_{h+1}^\pi(s_i') \geq v_{s_i'}$

2. $r_h(s,a) + \sum_i P_h(s_i' \mid s,a) v_{s_i'} \geq v$

$$\overline{\mathcal{A}}_h(s,v) := \left\{ (a,\mathbf{v}) \in \mathcal{A} \times \mathcal{V}^S \mid r_h(s,a) + \sum_{s'} P_h(s' \mid s,a) v_{s'} \geq v \right\}$$

Action-Space Dynamic Programming

Action-Space Dynamic Programming

$$\begin{aligned} \min_{\mathbf{v} \in \mathcal{V}^S} \quad & P_h(1 \mid s,a)\overline{C}^*_{h+1}(1, v_1) + \cdots + P_h(S \mid s,a)\overline{C}^*_{h+1}(S, v_S) \\ \text{s.t.} \quad & P_h(1 \mid s,a) v_1 + \cdots + P_h(S \mid s,a) v_S \geq v - r_h(s,a) \end{aligned}$$

Action-Space Dynamic Programming

$$\begin{aligned} \min_{\mathbf{v} \in \mathcal{V}^S} \quad & P_h(1 \mid s,a)\overline{C}^*_{h+1}(1, v_1) + \cdots + P_h(S \mid s,a)\overline{C}^*_{h+1}(S, v_S) \\ \text{s.t.} \quad & P_h(1 \mid s,a) v_1 + \cdots + P_h(S \mid s,a) v_S \geq v - r_h(s,a) \end{aligned}$$

Action-Space Dynamic Programming

$$\min_{\mathbf{v} \in \mathcal{V}^S} \quad P_h(1 \mid s,a)\overline{C}^*_{h+1}(1, v_1) + \cdots + P_h(S \mid s,a)\overline{C}^*_{h+1}(S, v_S)$$

$$\text{s.t.} \quad P_h(1 \mid s,a) v_1 + \cdots + P_h(S \mid s,a) v_S \geq v - r_h(s,a)$$

Can choose each v_i independently if track the partial demand

Action-Space Dynamic Programming

$$\min_{\mathbf{v}\in\mathcal{V}^S} \quad P_h(1 \mid s,a)\overline{C}^*_{h+1}(1, v_1) + \cdots + P_h(S \mid s,a)\overline{C}^*_{h+1}(S, v_S)$$

$$\text{s.t.} \quad P_h(1 \mid s,a) v_1 + \cdots + P_h(S \mid s,a) v_S \geq v - r_h(s,a)$$

Can choose each v_i independently if track the partial demand

Space Recursion!

Action-Space Dynamic Programming

$$\min_{\mathbf{v} \in \mathcal{V}^S} \quad P_h(1 \mid s,a)\overline{C}^*_{h+1}(1, v_1) + \cdots + P_h(S \mid s,a)\overline{C}^*_{h+1}(S, v_S)$$

$$\text{s.t.} \quad P_h(1 \mid s,a)v_1 + \cdots + P_h(S \mid s,a)v_S \geq v - r_h(s,a)$$

Can choose each v_i independently if track the partial demand

Space Recursion!

$$g(t,u) = \min_{v_t \in \mathcal{V}} P_h(t \mid s,a)C^*_{h+1}(t, v_t) + g(t+1, u + P_h(t \mid s,a)v_t)$$

Action-Space Dynamic Programming

$$\min_{\mathbf{v}\in\mathcal{V}^S} \quad P_h(1 \mid s,a)\overline{C}^*_{h+1}(1, v_1) + \cdots + P_h(S \mid s,a)\overline{C}^*_{h+1}(S, v_S)$$

$$\text{s.t.} \quad P_h(1 \mid s,a) v_1 + \cdots + P_h(S \mid s,a) v_S \geq v - r_h(s,a)$$

Can choose each v_i independently if track the partial demand

Space Recursion!

Partial demand

$$g(t,u) = \min_{v_t \in \mathcal{V}} P_h(t \mid s,a) C^*_{h+1}(t, v_t) + g(t+1, u + P_h(t \mid s,a) v_t)$$

Action-Space Dynamic Programming

$$
\begin{aligned}
\min_{\mathbf{v} \in \mathcal{V}^S} \quad & P_h(1 \mid s,a)\overline{C}^*_{h+1}(1, v_1) + \cdots + P_h(S \mid s,a)\overline{C}^*_{h+1}(S, v_S) \\
\text{s.t.} \quad & P_h(1 \mid s,a) v_1 + \cdots + P_h(S \mid s,a) v_S \geq v - r_h(s,a)
\end{aligned}
$$

Can choose each v_i independently if track the partial demand

---

Space Recursion!

Partial demand

$$
g(t, u) = \min_{v_t \in \mathcal{V}} P_h(t \mid s,a) C^*_{h+1}(t, v_t) + g(t+1, u + P_h(t \mid s,a) v_t)
$$

Value check at end: $g(S+1, u) := \chi_{\{u \geq v\}}$

Why Deterministic Policies?

Why Deterministic Policies?

- Cheap [1]

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]
  - Predictable

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]
  - Predictable

Why Deterministic Policies?

- Cheap [1]
- Multi-agent coordination [2]
- Trust-worthy [3]
  - Predictable
- Optimal for modern constraints [4]

Value-Demand Augmentation

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad & C_h^\pi(\tau_h) \\ \text{s.t.} \quad & V_h^\pi(\tau_h) \geq v \end{aligned}$$

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{C}^*_h(s,v) = \min_{\pi \in \Pi^D} \quad & C^\pi_h(\tau_h) \\ \text{s.t.} \quad & V^\pi_h(\tau_h) \geq v \end{aligned}$$

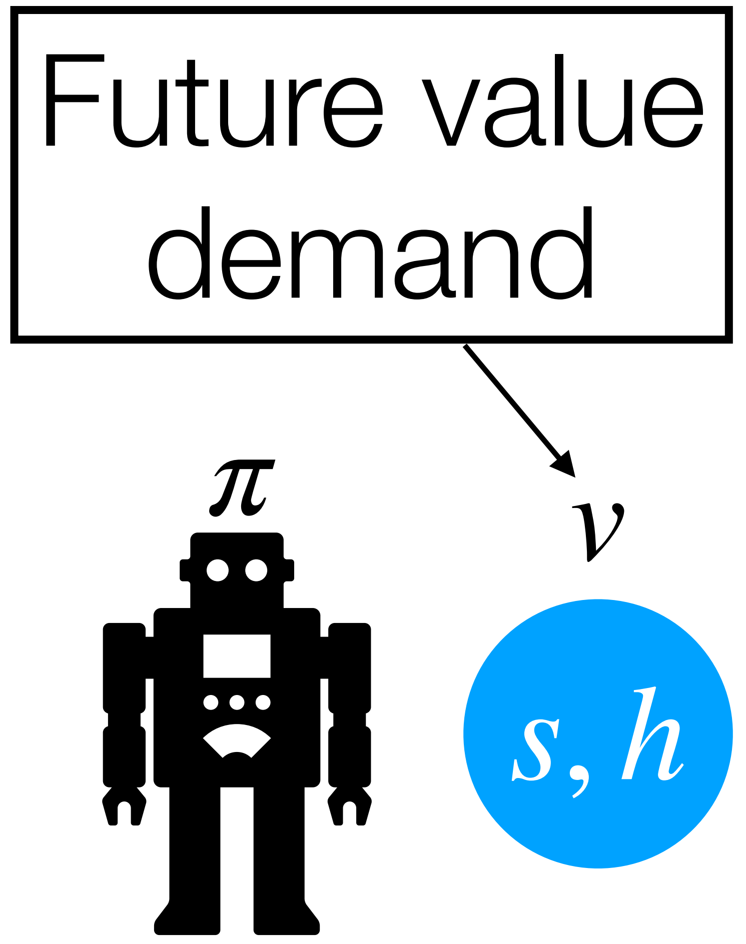
→ Dual = $\overline{C}^*_1(s_0, V^*)$

Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad & C_h^\pi(\tau_h) \\ \text{s.t.} \quad & V_h^\pi(\tau_h) \geq v \end{aligned}$$

Dual = $\overline{C}_1^*(s_0, V^*)$

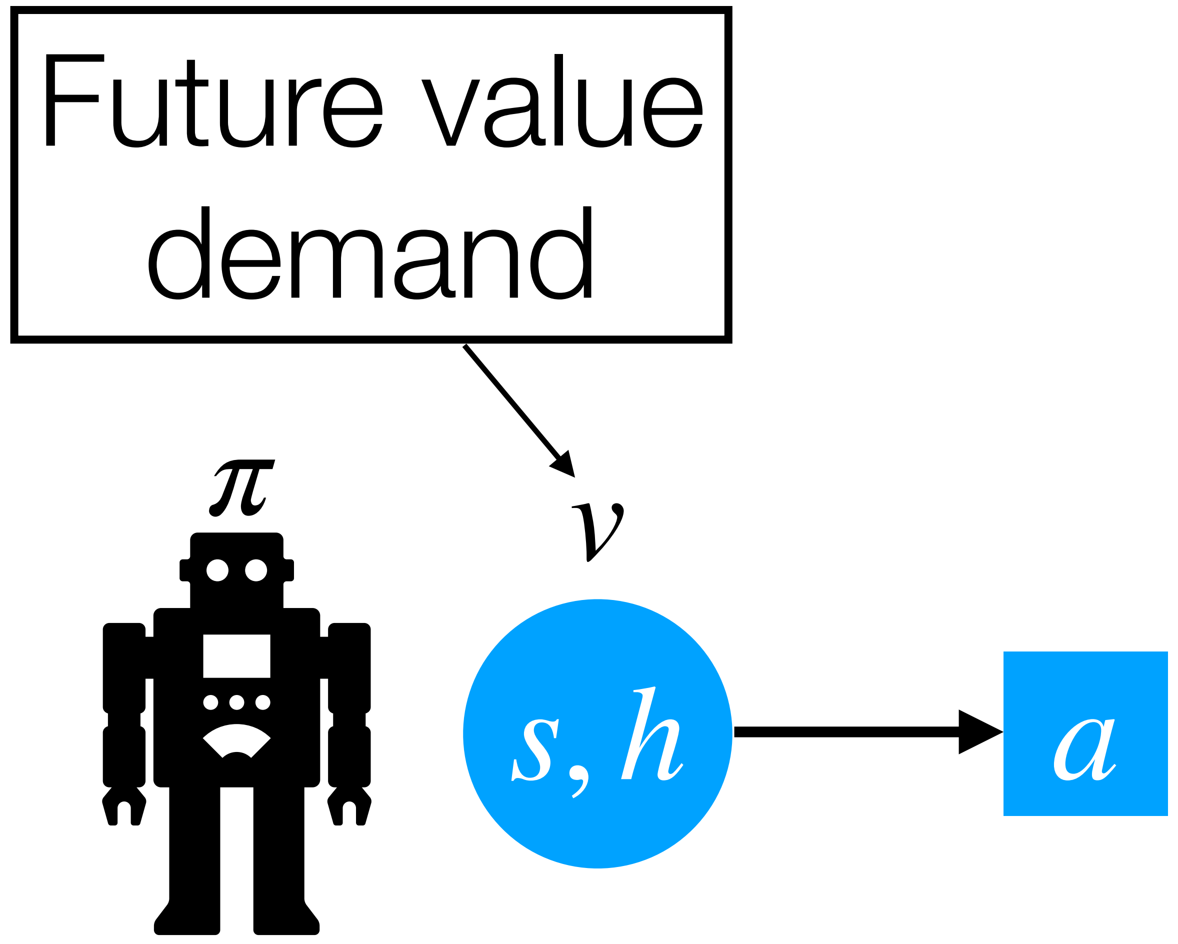


Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\begin{aligned} \overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad & C_h^\pi(\tau_h) \\ \text{s.t.} \quad & V_h^\pi(\tau_h) \geq v \end{aligned}$$

Dual = $\overline{C}_1^*(s_0, V^*)$

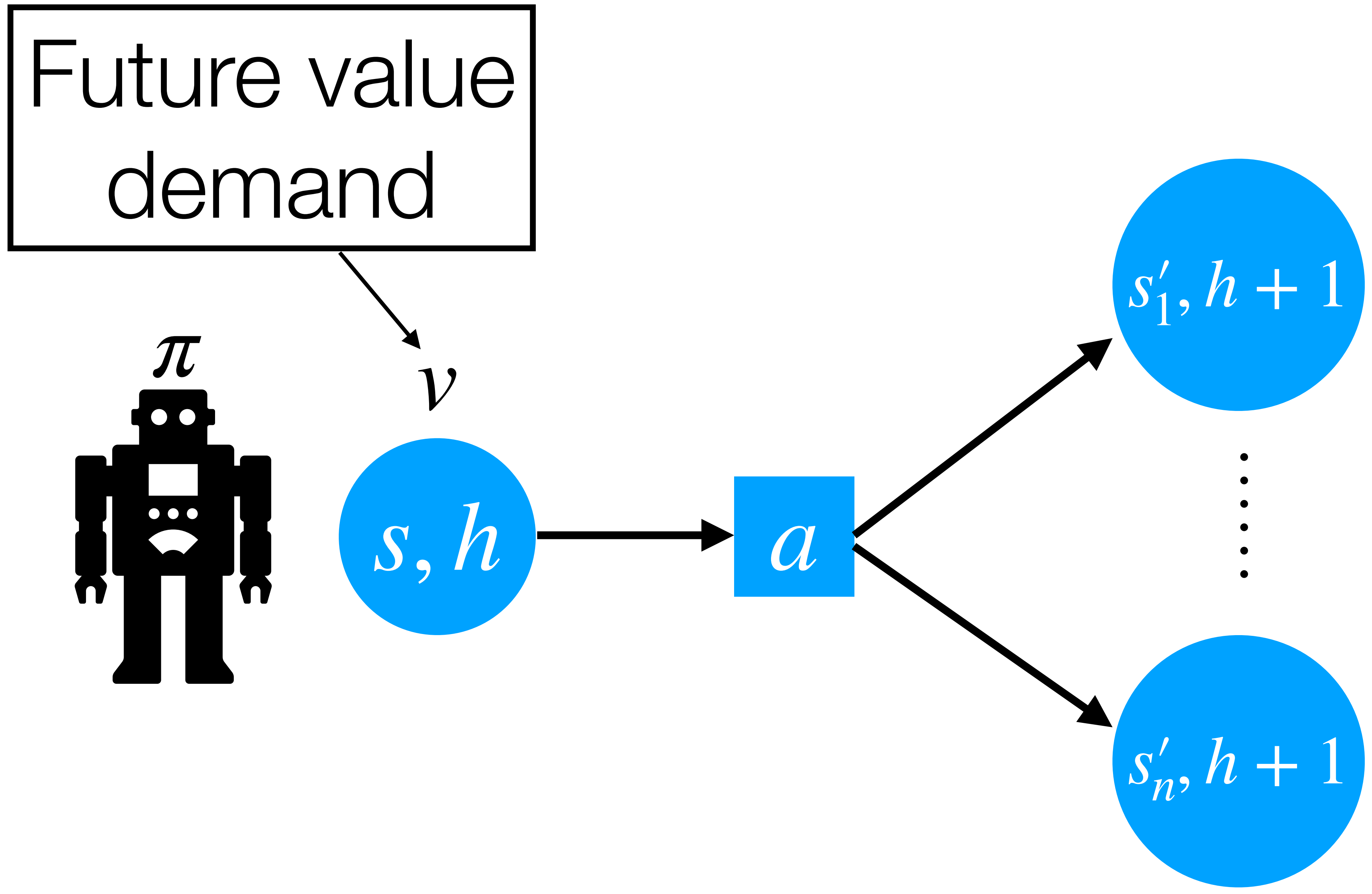


Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad C_h^\pi(\tau_h) \quad \text{s.t.} \quad V_h^\pi(\tau_h) \geq v$$

Dual = $\overline{C}_1^*(s_0, V^*)$

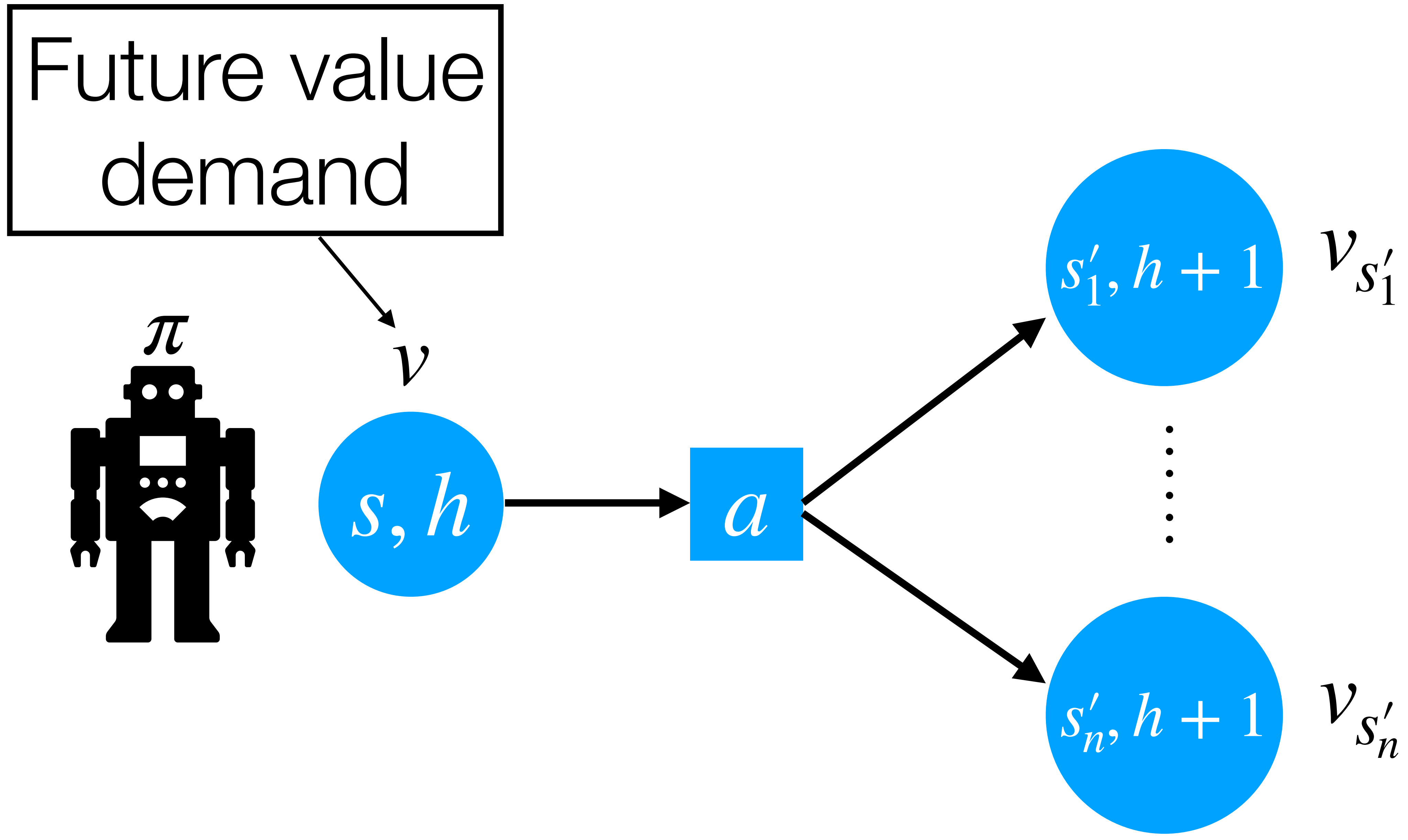


Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}^*_h(s, v) = \min_{\pi \in \Pi^D} \quad C^\pi_h(\tau_h) \quad \text{s.t.} \quad V^\pi_h(\tau_h) \geq v$$

Dual = $\overline{C}^*_1(s_0, V^*)$

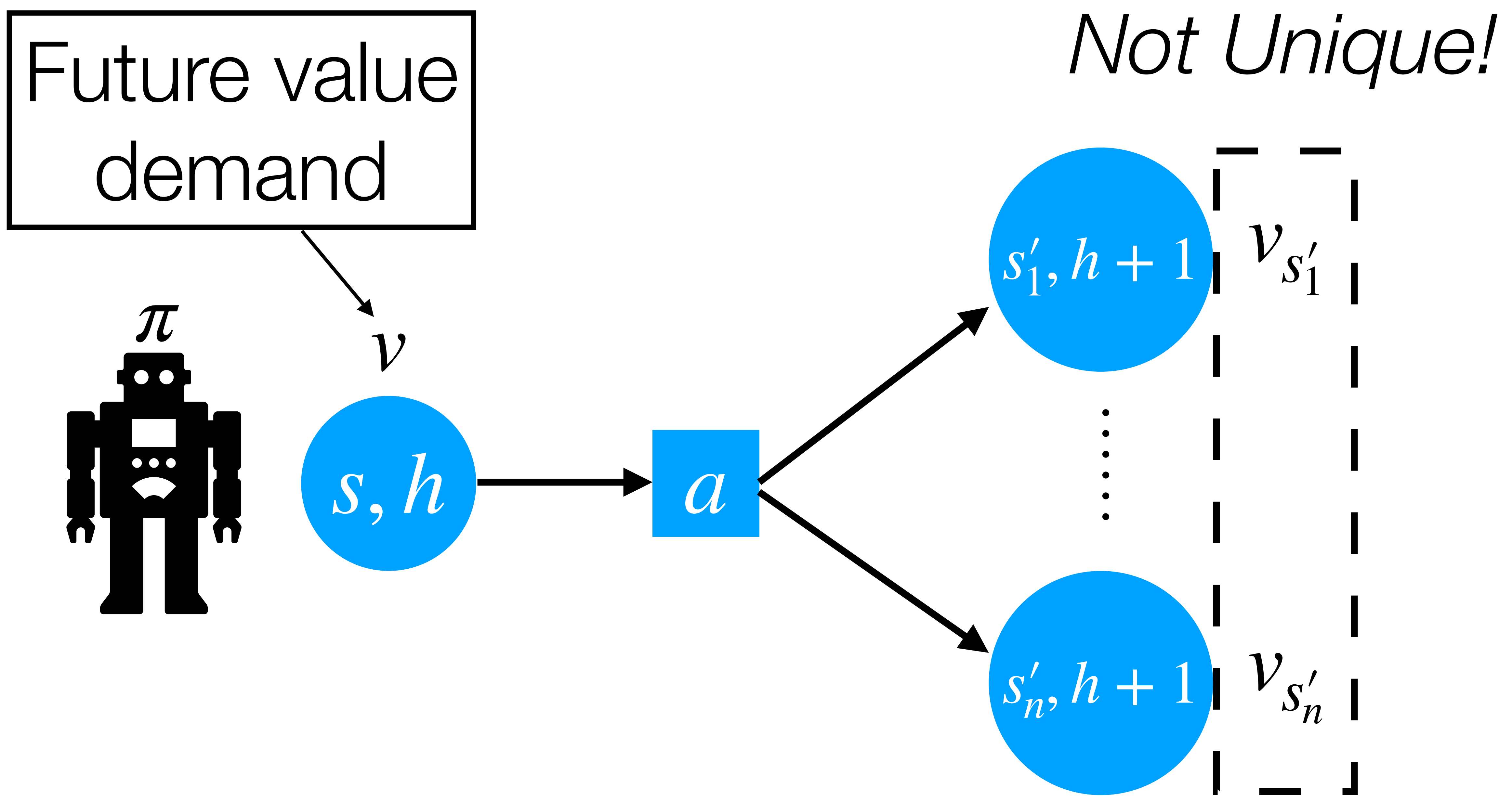


Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad C_h^\pi(\tau_h) \qquad \text{s.t.} \quad V_h^\pi(\tau_h) \geq v$$

Dual = $\overline{C}_1^*(s_0, V^*)$

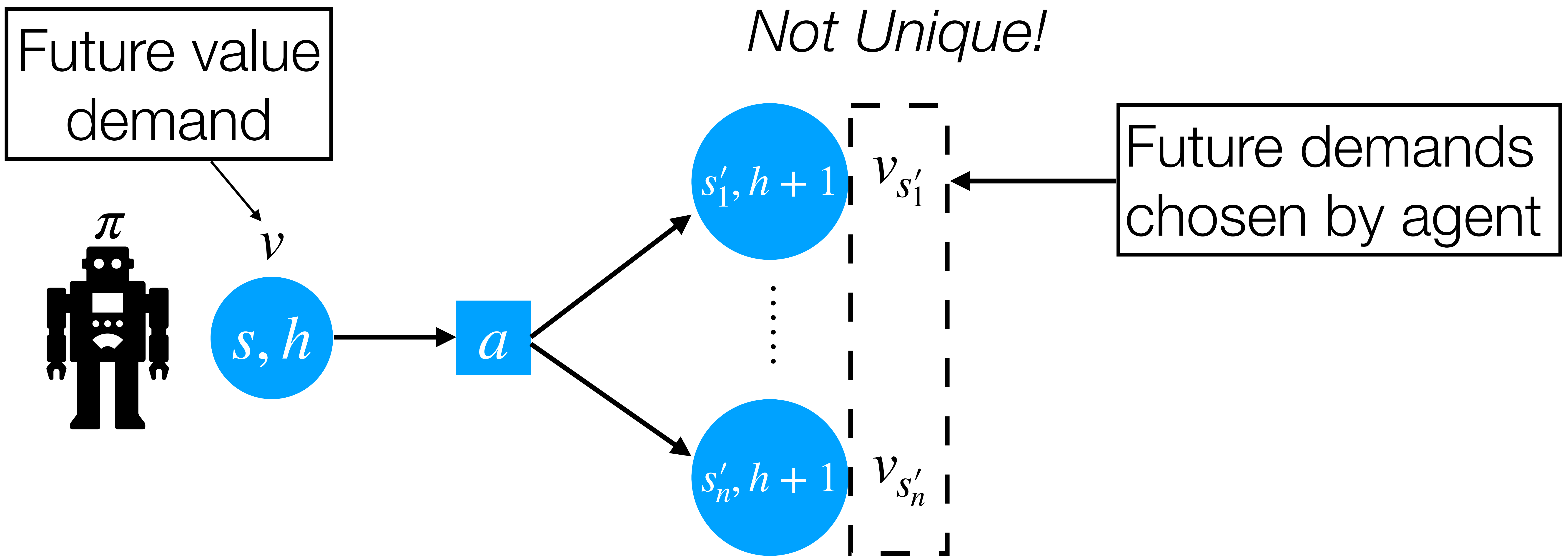


Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$
\begin{aligned}
\overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad & C_h^\pi(\tau_h) \\
\text{s.t.} \quad & V_h^\pi(\tau_h) \geq v
\end{aligned}
$$

Dual = $\overline{C}_1^*(s_0, V^*)$

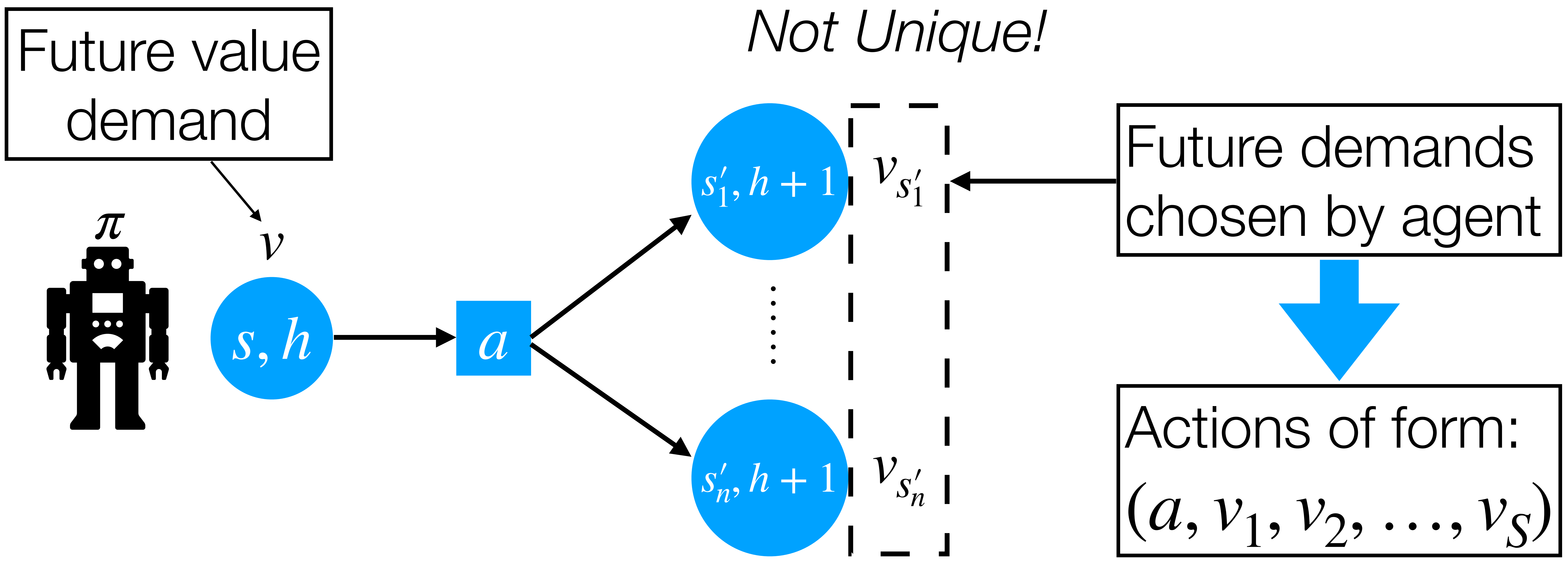


Value-Demand Augmentation

Intuition: Build $\overline{M}$ satisfying,

$$\overline{C}_h^*(s, v) = \min_{\pi \in \Pi^D} \quad C_h^\pi(\tau_h) \quad \text{s.t.} \quad V_h^\pi(\tau_h) \geq v$$

Dual = $\overline{C}_1^*(s_0, V^*)$



Constraint Landscape

Put the formulas in here

Constraint Landscape

Put the formulas in here

Constraints

- Expectation
- Chance
- Almost Sure
- Anytime

Constraint Landscape

Put the formulas in here

Constraints

Flexibility

- Expectation
- Chance
- Almost Sure
- Anytime

Constraint Landscape

Put the formulas in here

Constraints

Flexibility — • Expectation

Precision — • Chance

• Almost Sure

• Anytime

Constraint Landscape

Put the formulas in here

Constraints

- Expectation — Flexibility
- Chance — Precision
- Almost Sure — Safety
- Anytime — Safety

Constraint Landscape

Put the formulas in here

Constraints

Flexibility — Expectation

Precision — Chance

Safety {
- Almost Sure
- Anytime

Policies

Constraint Landscape

Put the formulas in here

Constraints

- Expectation — Flexibility
- Chance — Precision
- Almost Sure — Safety
- Anytime — Safety

Policies

- Stochastic

Constraint Landscape

Put the formulas in here

Constraints

- Expectation — Flexibility
- Chance — Precision
- Almost Sure
- Anytime

Safety: Almost Sure, Anytime

Policies

- Stochastic
- Deterministic

Constraint Landscape

Put the formulas in here

Constraints

Flexibility — Expectation

Precision — Chance

Safety { Almost Sure, Anytime

- Expectation
- Chance
- Almost Sure
- Anytime

Policies

- Stochastic
- Deterministic

{ Predictable

Constraint Landscape

Put the formulas in here

Constraints

- Expectation — Flexibility
- Chance — Precision
- Almost Sure
- Anytime

Safety (Almost Sure, Anytime)

Policies

- Stochastic
- Deterministic

Predictable

Cheap

Constraint Landscape

Put the formulas in here

Constraints

- Expectation — Flexibility
- Chance — Precision
- Almost Sure
- Anytime

Safety: Almost Sure, Anytime

Policies

- Stochastic
- Deterministic

Predictable

Cheap

Coordination

Constraint Landscape

Put the formulas in here

Constraints

- Expectation (Flexibility)
- Chance (Precision)
- Almost Sure (Safety)
- Anytime (Safety)

Policies

- Stochastic
- Deterministic

Predictable
Cheap
Coordination

Constraint Landscape

Put the formulas in here

Constraints

- Expectation — Flexibility
- Chance — Precision
- Almost Sure
- Anytime

Safety (Almost Sure, Anytime)

Policies

- Stochastic
- Deterministic

Predictable
Cheap
Coordination

Are any of the others ever **value-approximable**?

Constraint Landscape

Put the formulas in here

Constraints

- Expectation — Flexibility
- Chance — Precision
- Almost Sure — Safety
- Anytime — Safety

Policies

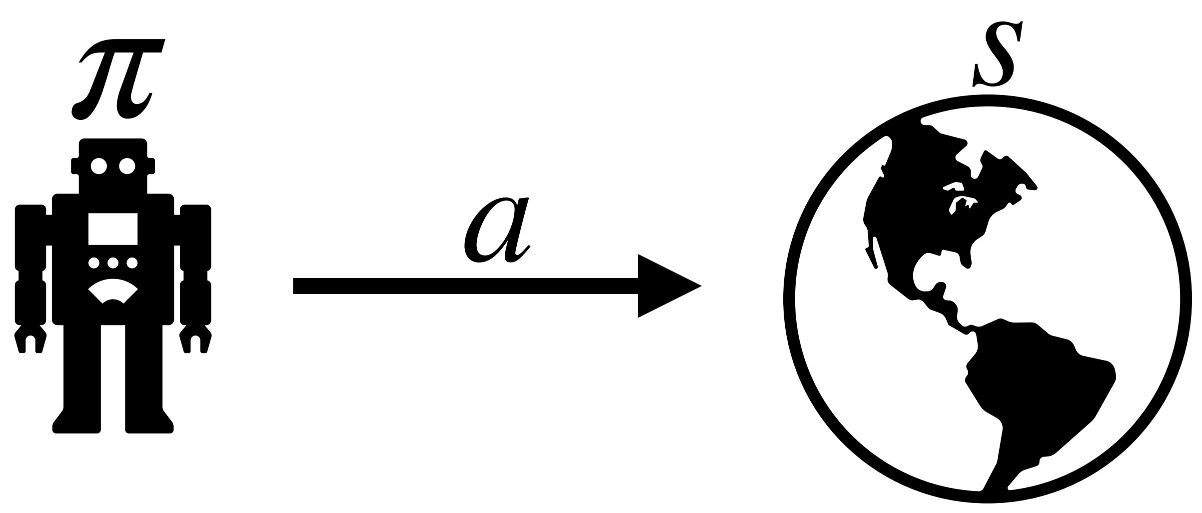
- Stochastic
- Deterministic

Deterministic: Predictable, Cheap, Coordination

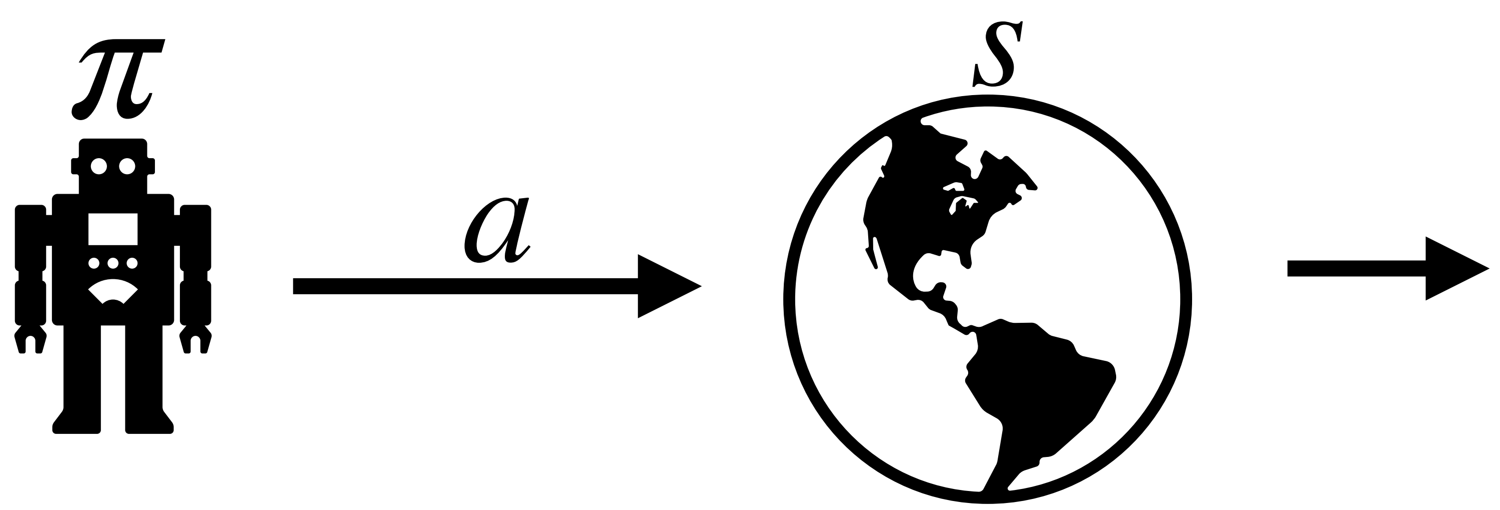
Are any of the others ever **value-approximable**?

General Formulation

General Formulation

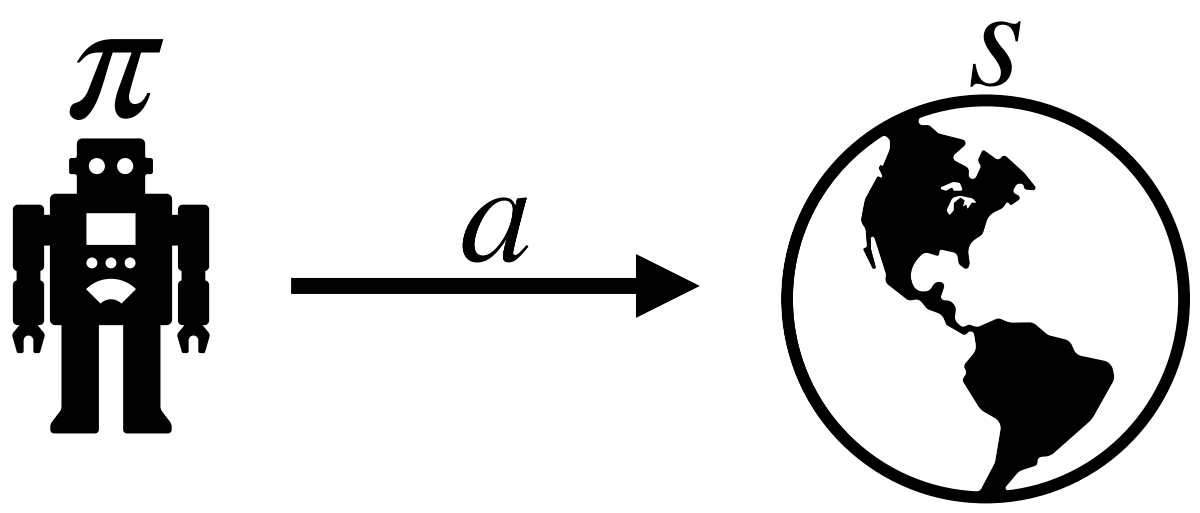


General Formulation



$$\begin{cases} r_h \sim R_h(s, a) \\ c_h \sim C_h(s, a) \end{cases}$$

General Formulation



$$\begin{cases} r_h \sim R_h(s, a) \\ c_h \sim C_h(s, a) \end{cases}$$

Agent's goal:

$$\max_{\pi} V^{\pi}$$

s.t. constraints on $\sum_{h=1}^{H} c_h$