Reward Poisoning Attacks on Offline Multi-Agent Reinforcement Learning

Young Wu, Jeremy McMahan, Xiaojin Zhu, Qiaomin Xie

***University of Wisconsin-Madison**

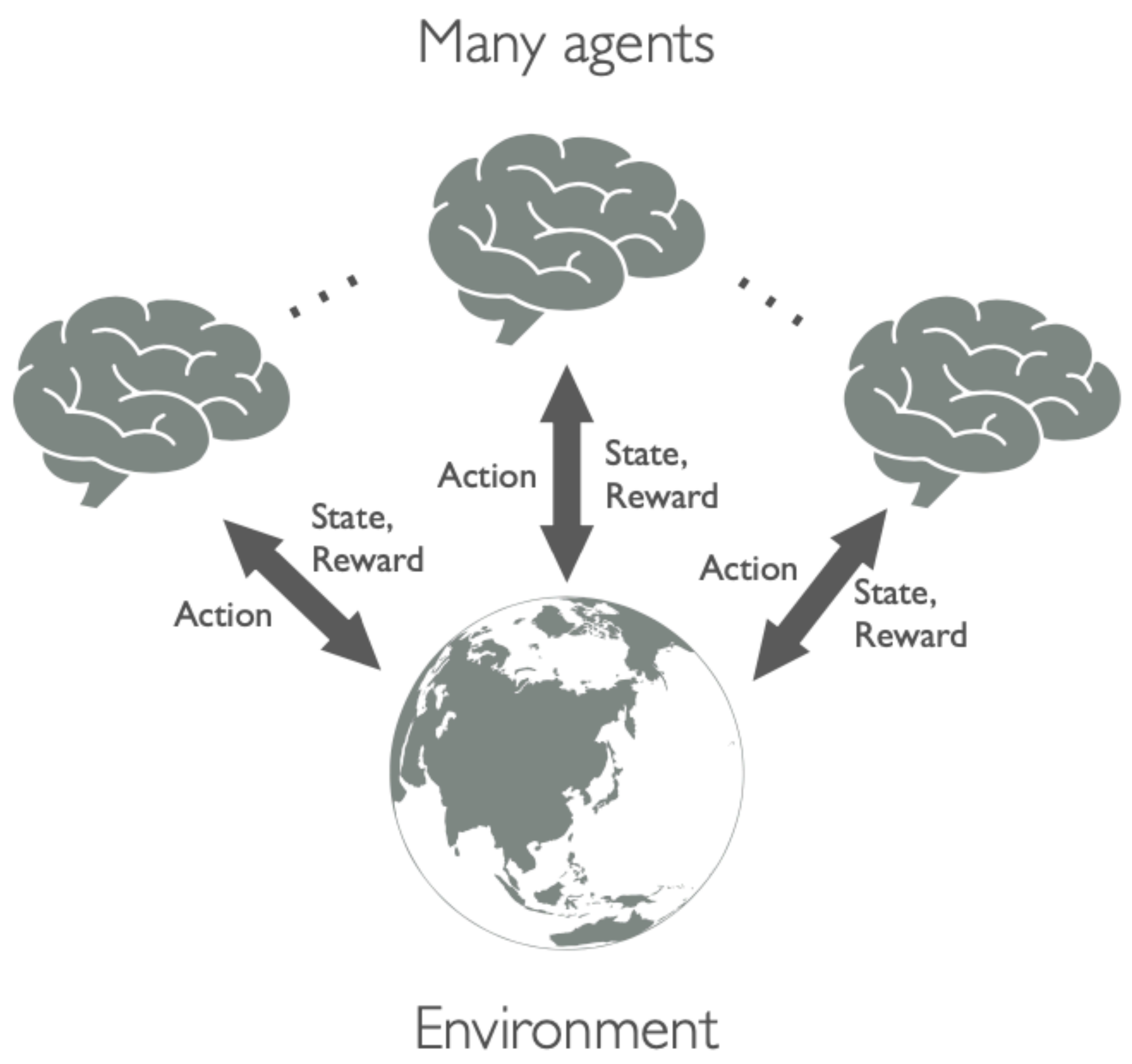
How to Manipulate Competitive Agents

Young Wu, Jeremy McMahan, Xiaojin Zhu, Qiaomin Xie

***University of Wisconsin-Madison**

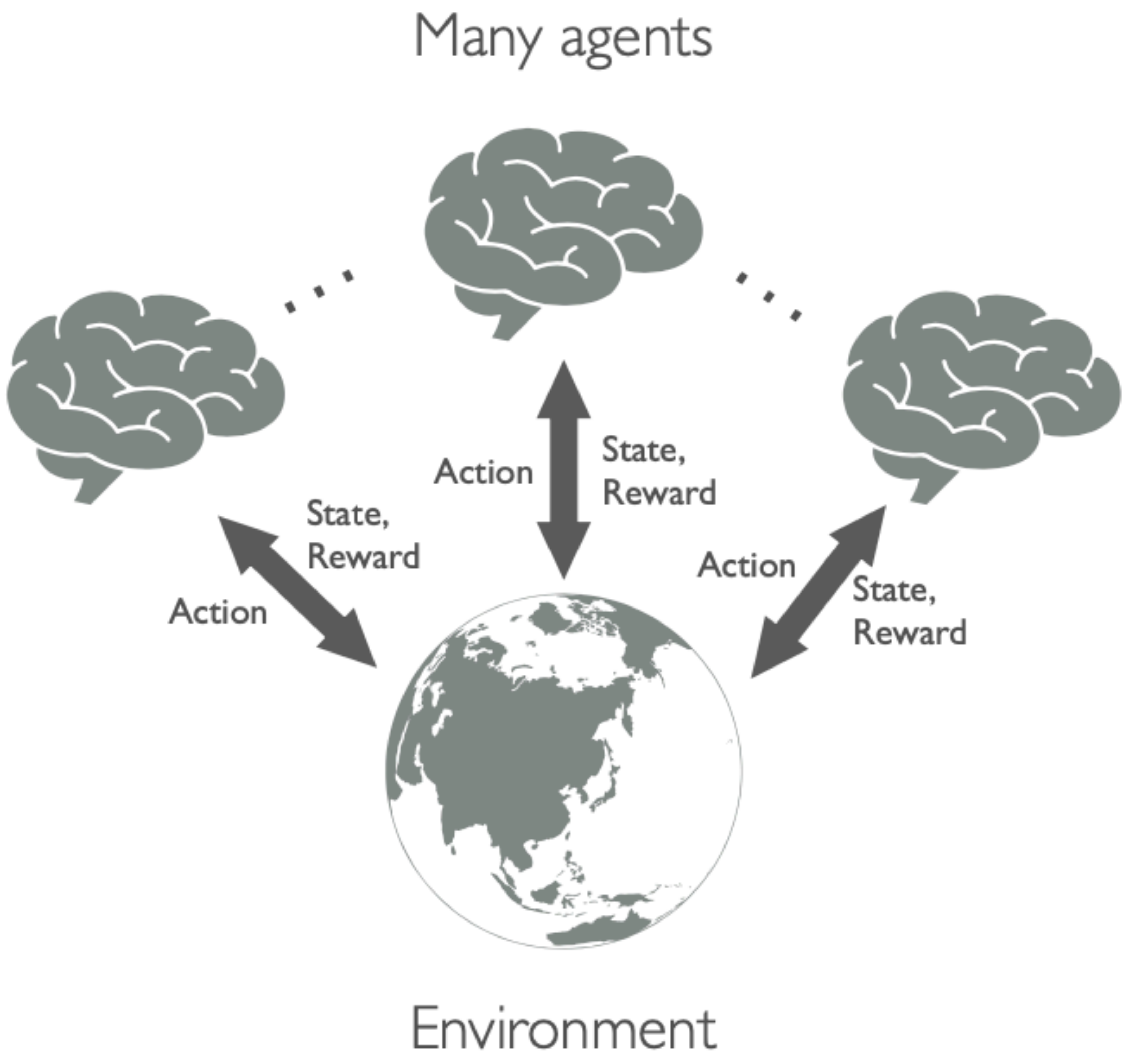
MARL

Learning Goals



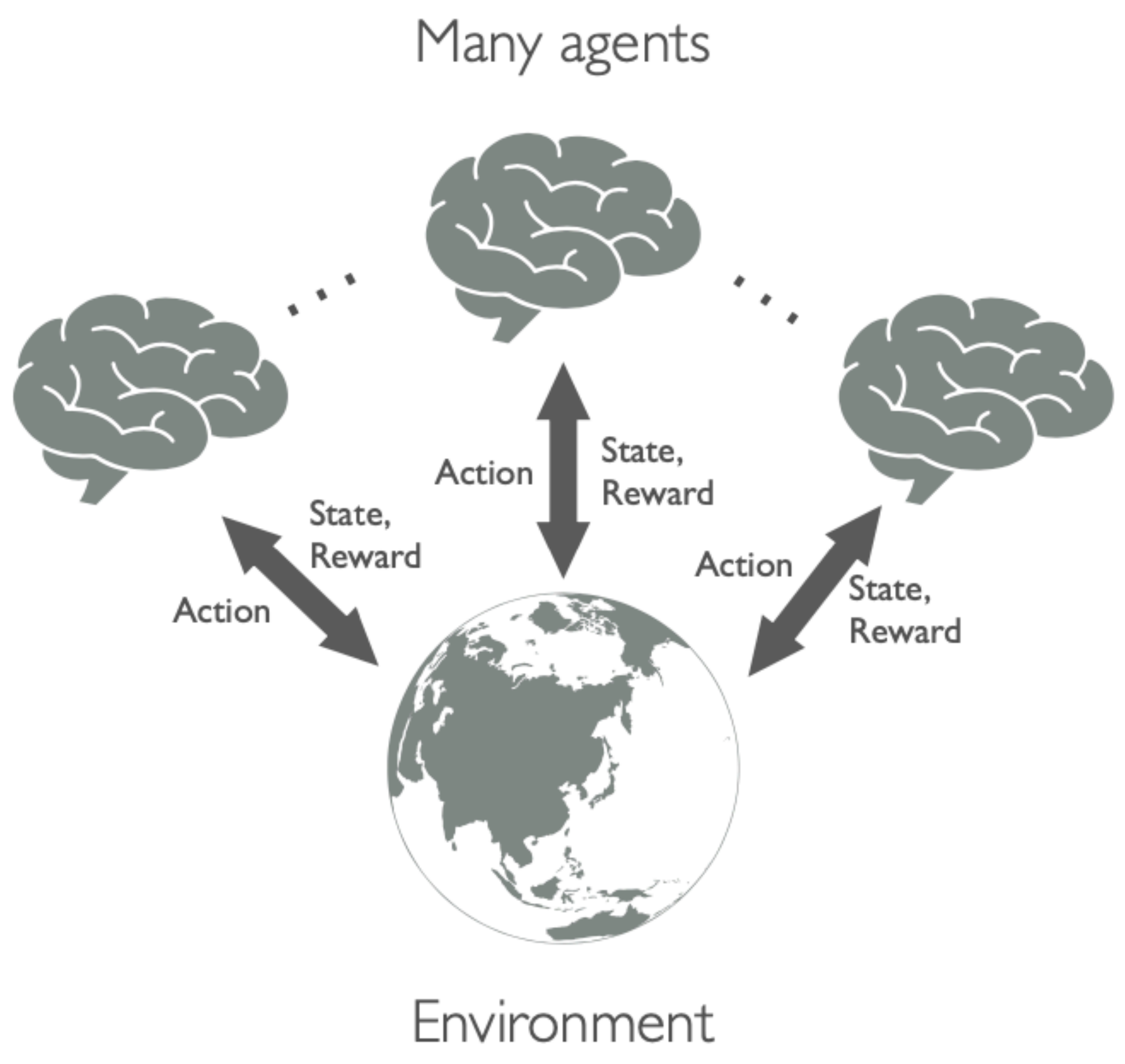
Learning Goals

- Agents learn a joint policy $\pi : \mathcal{S} \to \Delta(\mathcal{A})$.



Learning Goals

- Agents learn a joint policy $\pi : \mathcal{S} \rightarrow \Delta(\mathcal{A})$.
- π is an "optimal" strategy.



Offline Learning

Offline Learning

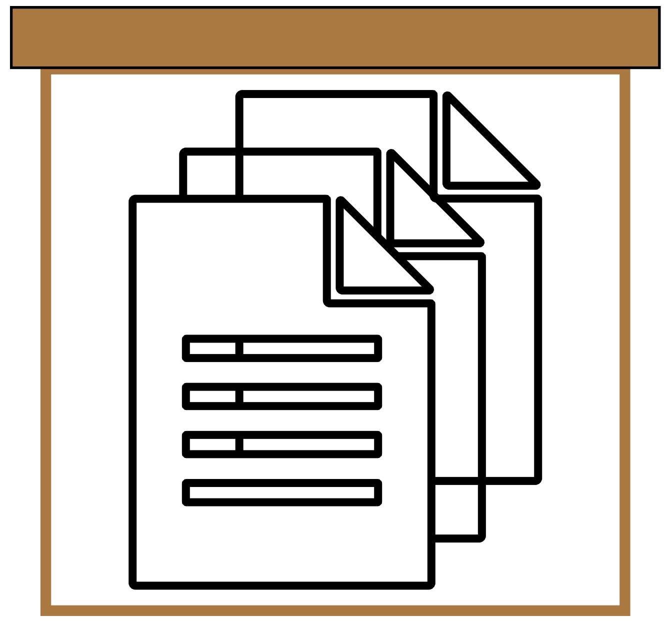
- Offline dataset records the episodes of the interaction.

Offline Learning

- Offline dataset records the episodes of the interaction.
- Agents use the shared data to compute a joint policy π.

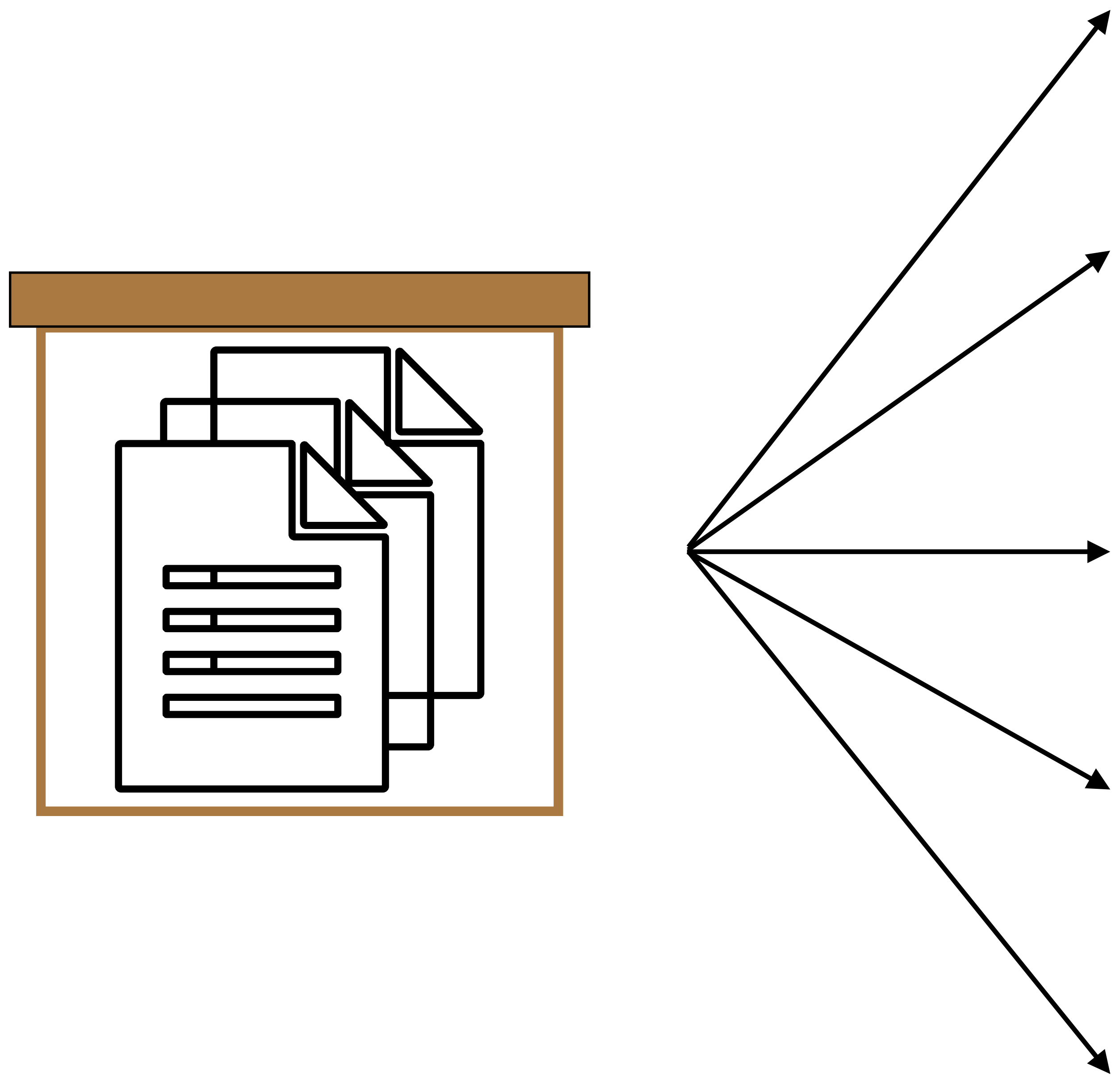
Offline Learning

- Offline dataset records the episodes of the interaction.
- Agents use the shared data to compute a joint policy π.



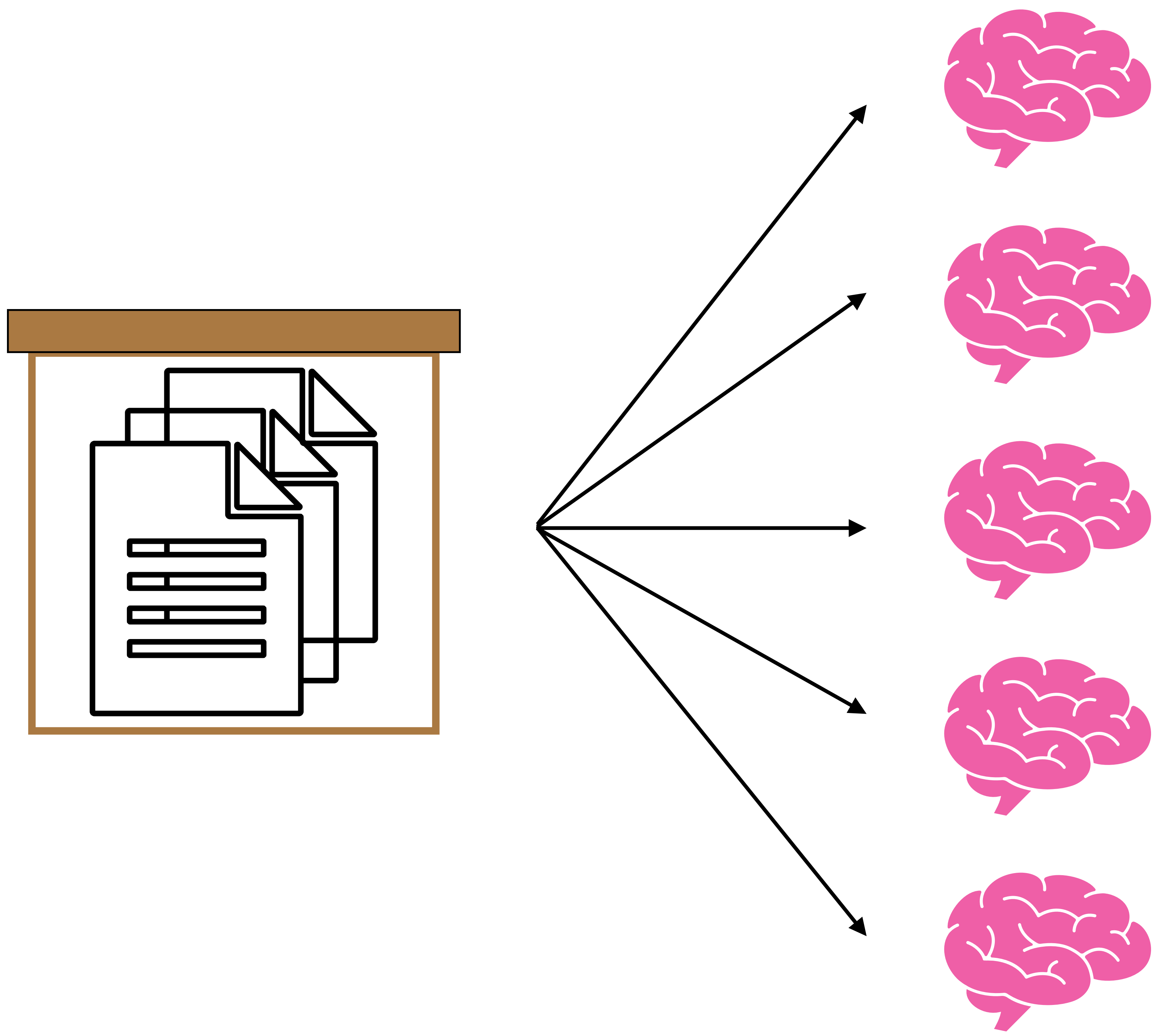
Offline Learning

- Offline dataset records the episodes of the interaction.
- Agents use the shared data to compute a joint policy π.



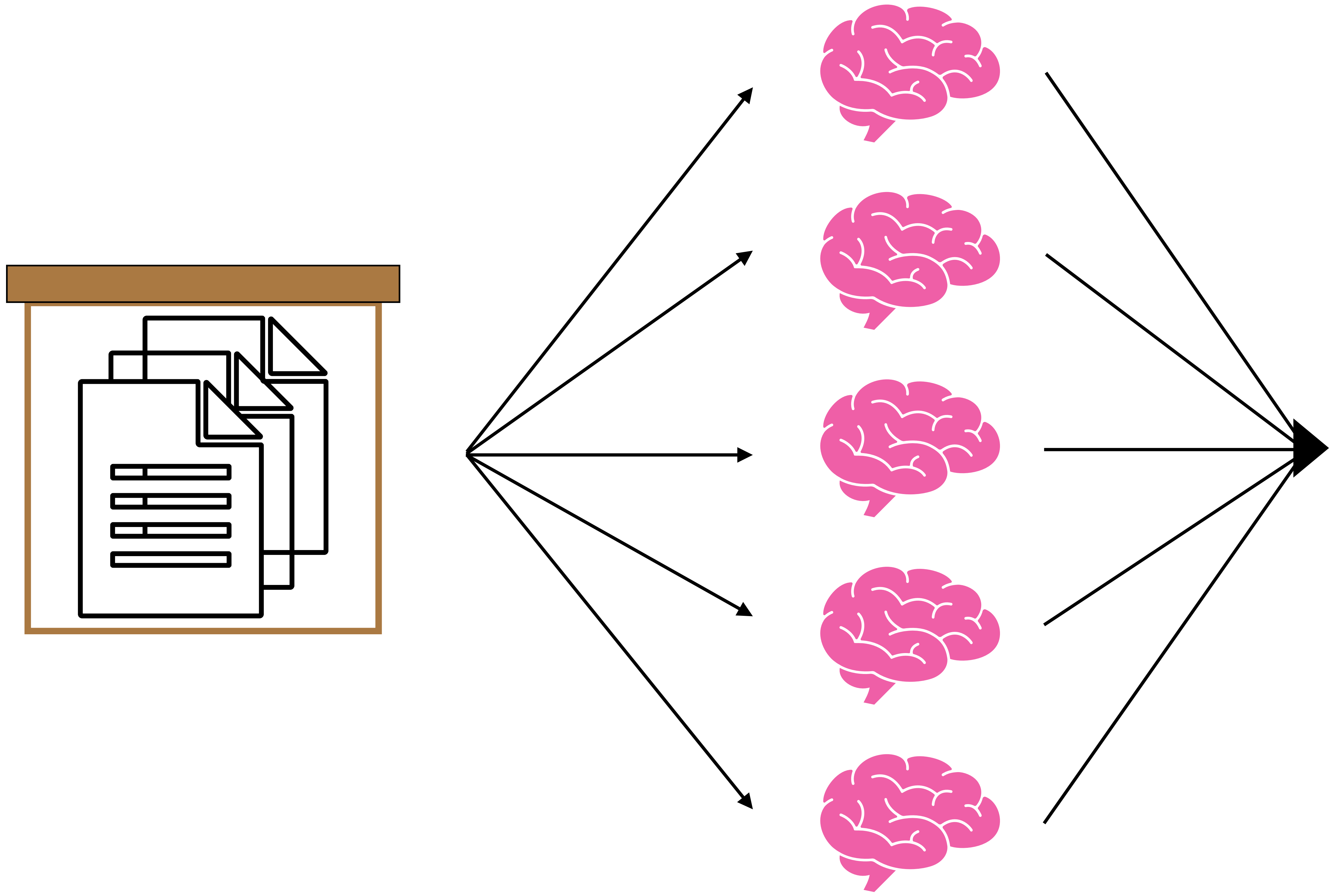
Offline Learning

- Offline dataset records the episodes of the interaction.
- Agents use the shared data to compute a joint policy π.



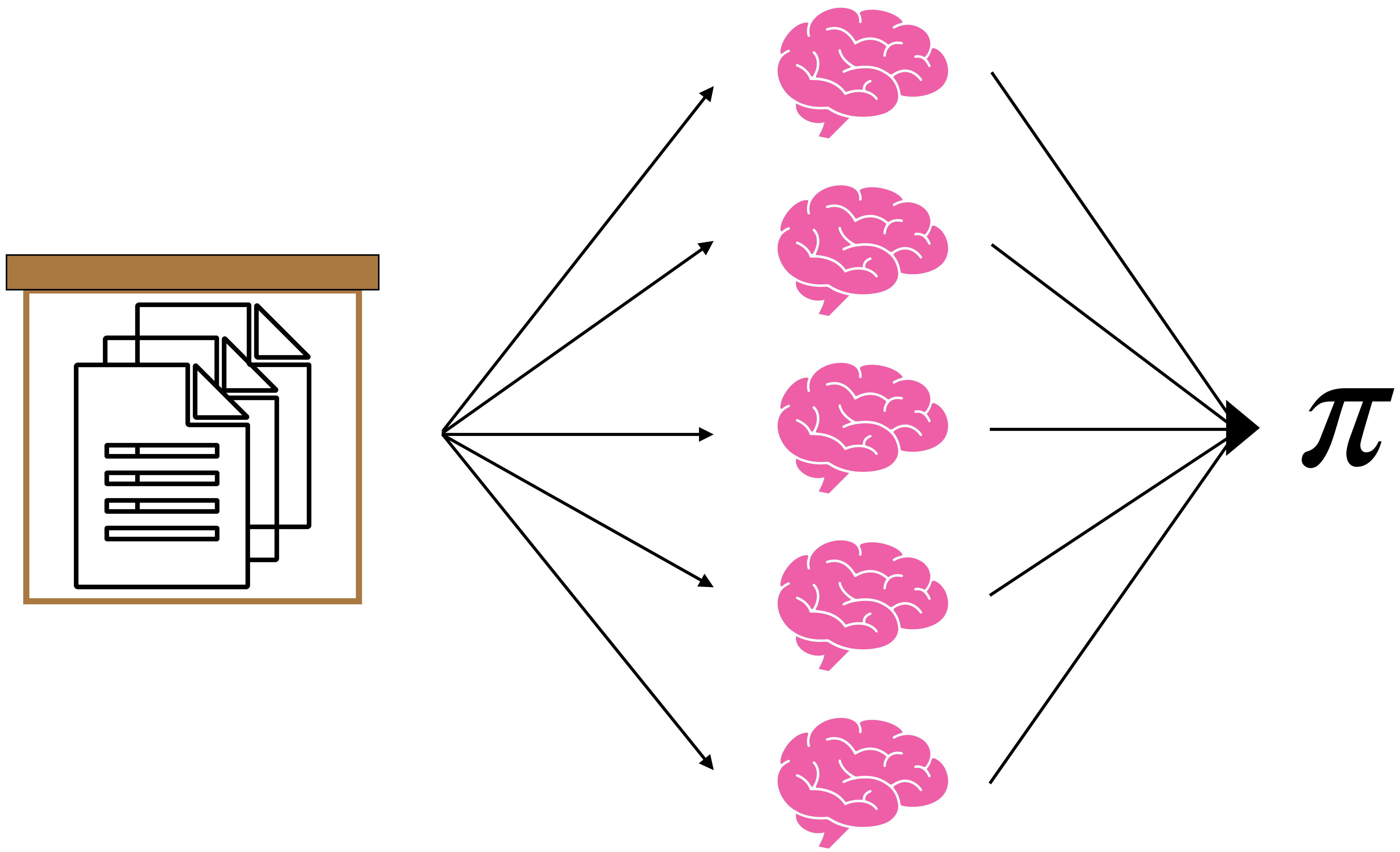
Offline Learning

- Offline dataset records the episodes of the interaction.
- Agents use the shared data to compute a joint policy π.



Offline Learning

- Offline dataset records the episodes of the interaction.
- Agents use the shared data to compute a joint policy π.



Markov Games

Markov Games

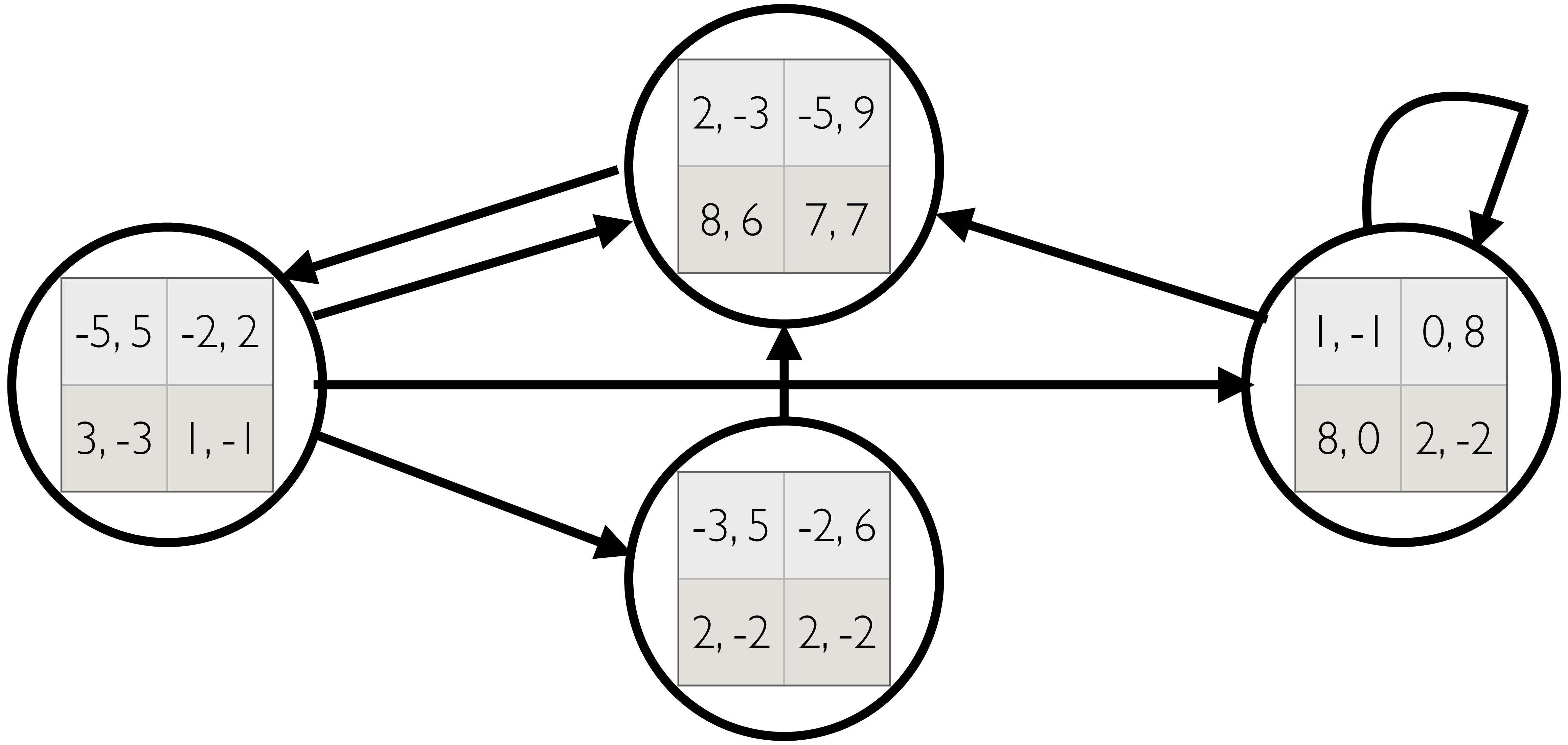
- Extension of MDPs to the multi-agent setting.

Markov Games

- Extension of MDPs to the multi-agent setting.
- Think of: MDP with a game-reward matrix at each state.

Markov Games

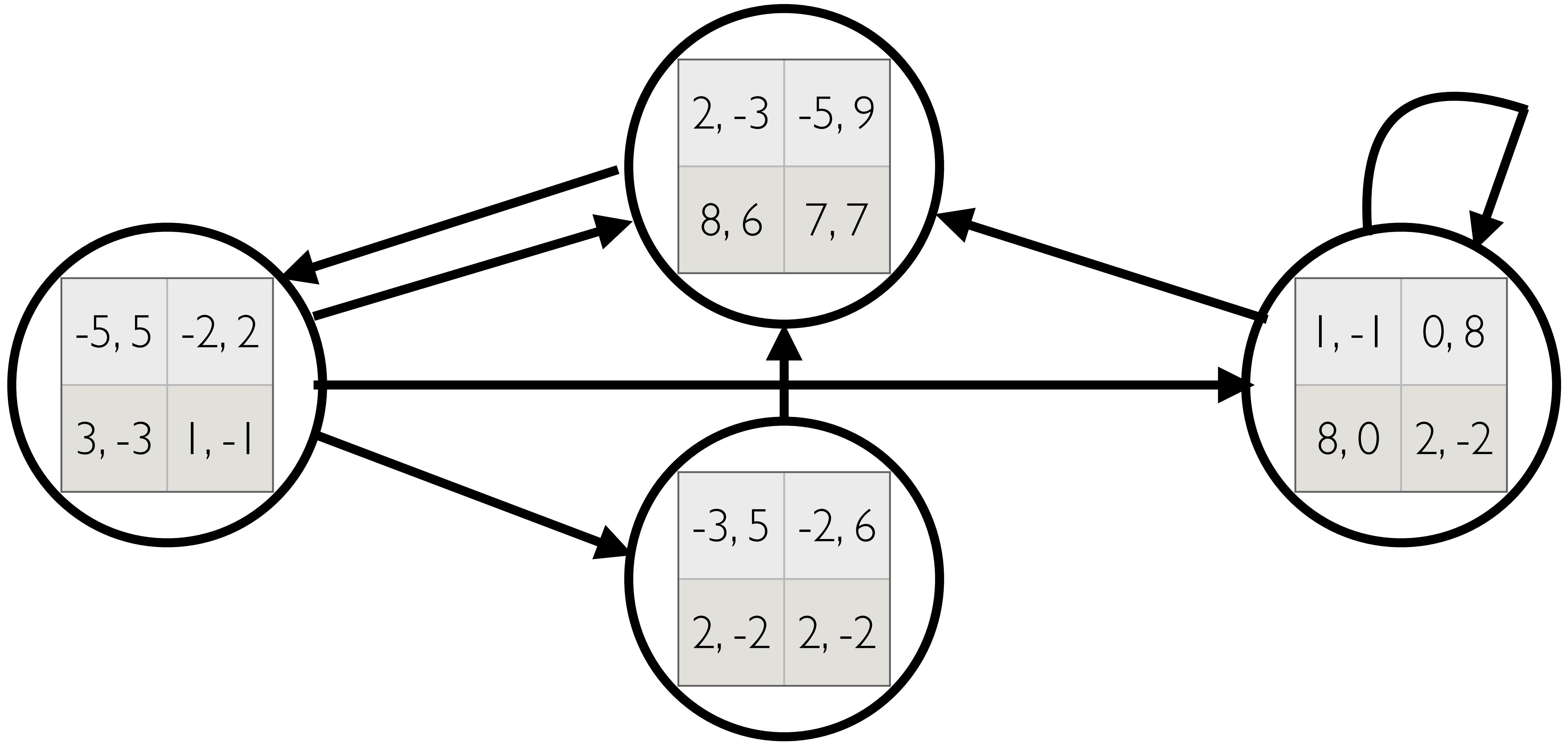
- Extension of MDPs to the multi-agent setting.
- Think of: MDP with a game-reward matrix at each state.



Markov Games

- Extension of MDPs to the multi-agent setting.
- Think of: MDP with a game-reward matrix at each state.

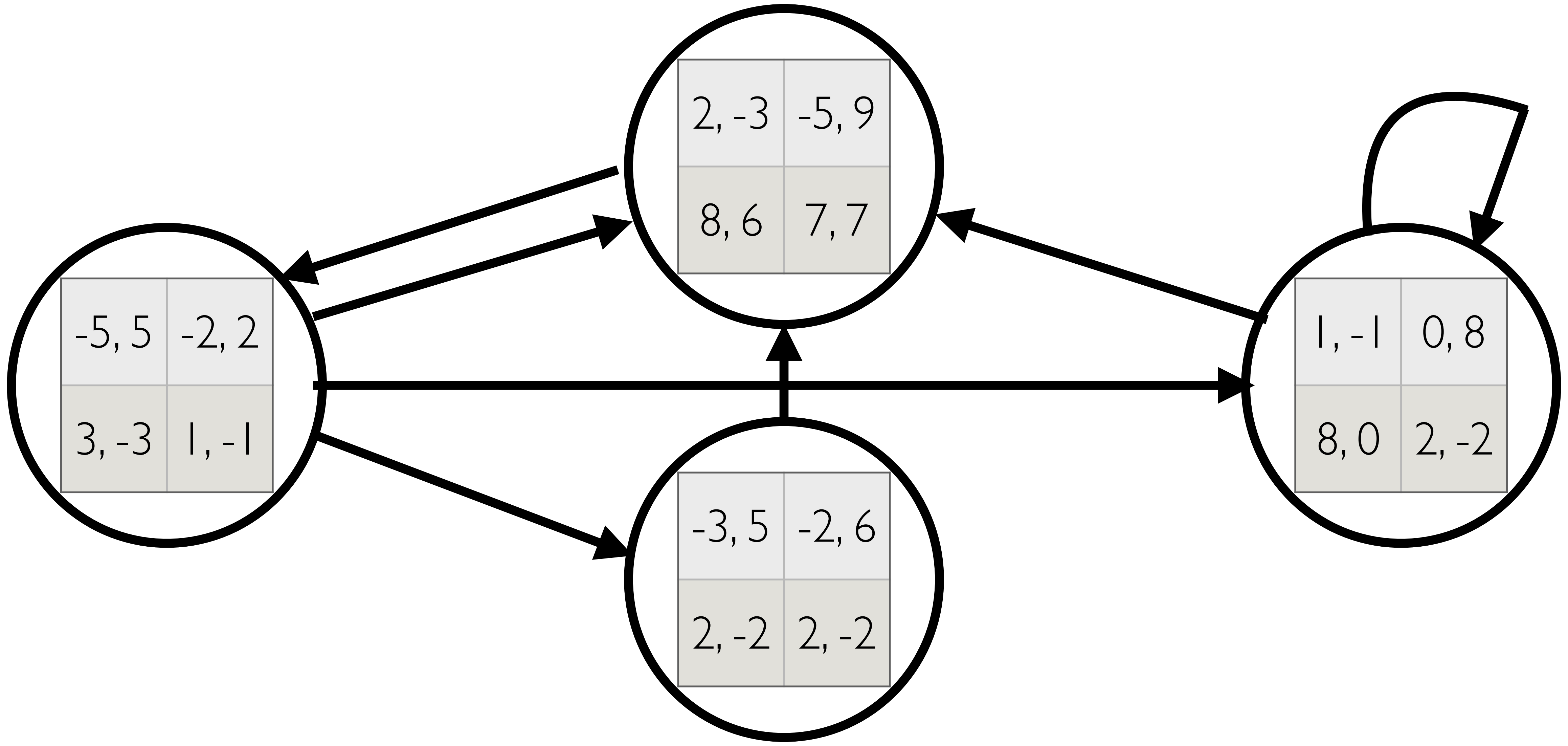
Reward depends on actions of all players.



Markov Games

- Extension of MDPs to the multi-agent setting.
- Think of: MDP with a game-reward matrix at each state.

Reward depends on actions of all players.



Transition depends on actions of all players.

Solution Concepts

Solution Concepts

- Solution to a game takes form of an Equilibrium.

Solution Concepts

- Solution to a game takes form of an Equilibrium.
- Examples: NE, DSE, CCE

Rationality

Rationality

- Simplest assumption on rationality: *no agent takes a strictly dominated action*, $Q_i(s, (a_i, a_{-i})) < Q_i(s, (a_i', a_{-i}))$.

Rationality

- Simplest assumption on rationality: *no agent takes a strictly dominated action*, $Q_i(s,(a_i,a_{-i})) < Q_i(s,(a'_i,a_{-i}))$.

- Strict Markov Perfect Dominant Strategy Equilibrium (MPDSE) is the corresponding equilibrium concept.

Rationality

- Simplest assumption on rationality: *no agent takes a strictly dominated action*, $Q_i(s, (a_i, a_{-i})) < Q_i(s, (a_i', a_{-i}))$.

- Strict Markov Perfect Dominant Strategy Equilibrium (MPDSE) is the corresponding equilibrium concept.

Key Fact: Rational agents always play the MPDSE if it exists.

Robust Learners

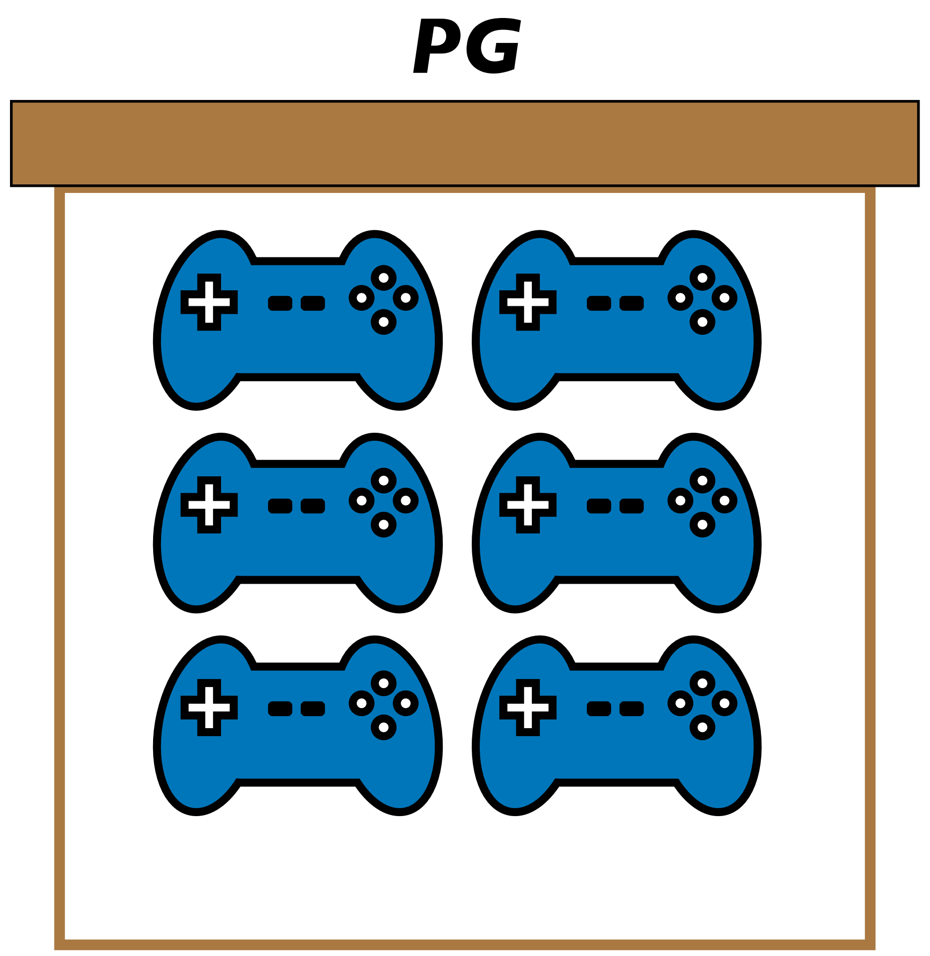
Plausible Games

Plausible Games

- To deal with dataset uncertainty, robust learners create a set of plausible games, *PG*.

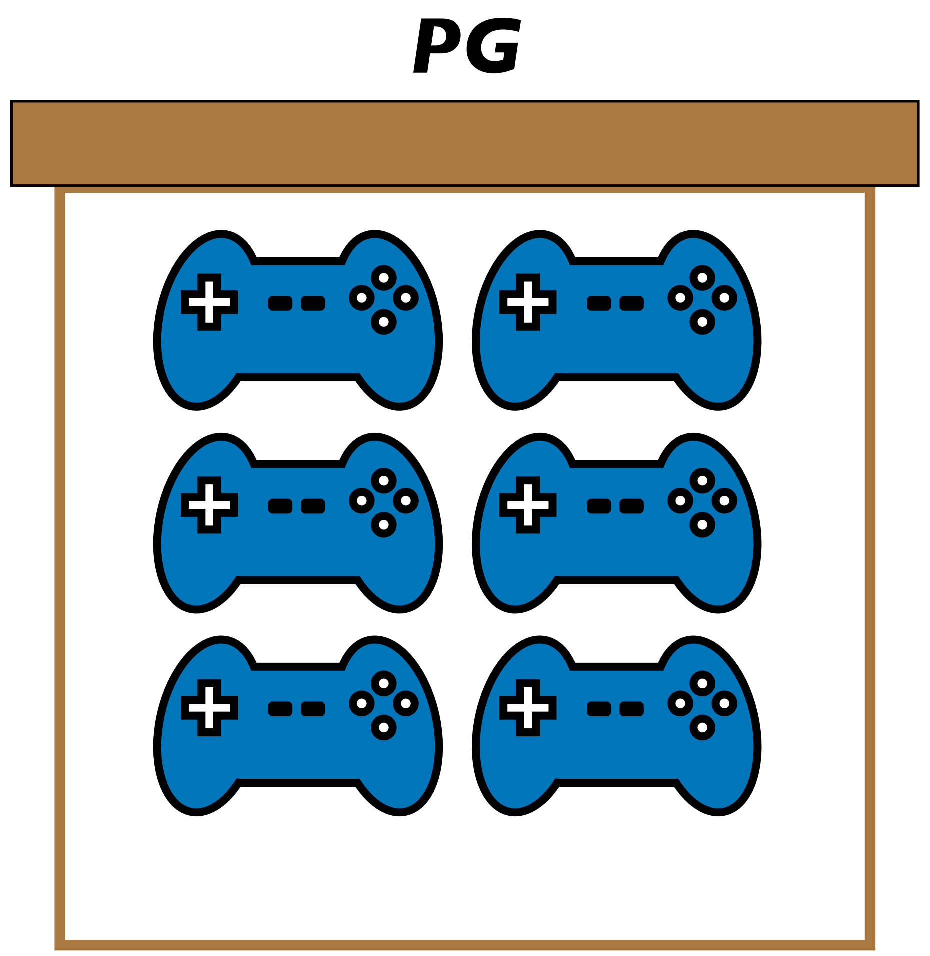
Plausible Games

- To deal with dataset uncertainty, robust learners create a set of plausible games, *PG*.



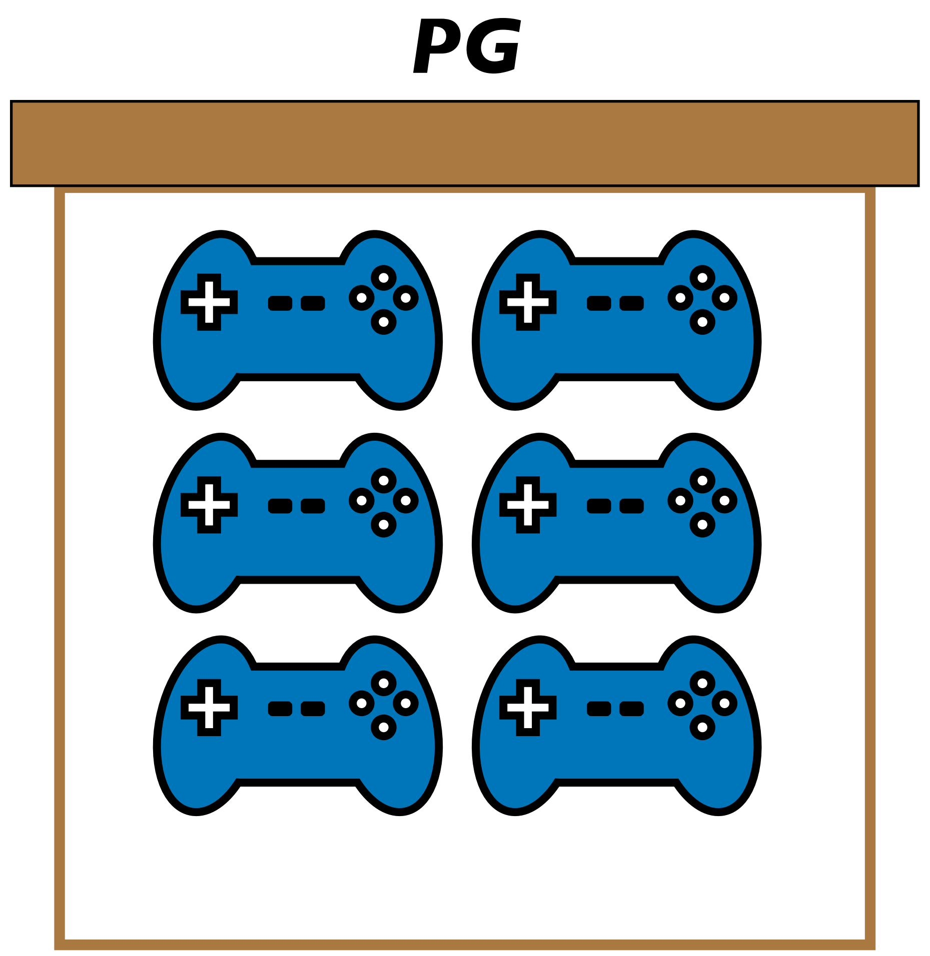
Plausible Games

- To deal with dataset uncertainty, robust learners create a set of plausible games, *PG*.
- Agents believe the true Markov Game lies within PG w.h.p.



Plausible Games

- To deal with dataset uncertainty, robust learners create a set of plausible games, *PG*.
- Agents believe the true Markov Game lies within PG w.h.p.
- Example: Confidence Bounded Learners (CBL) assume that $CI_i^R(s,a) = \left\{ R_i(s,a) \in [-b,b] \mid |R_i(s,a) - \hat{R}_i(s,a)| \le \rho^R(s,a) \right\}$.

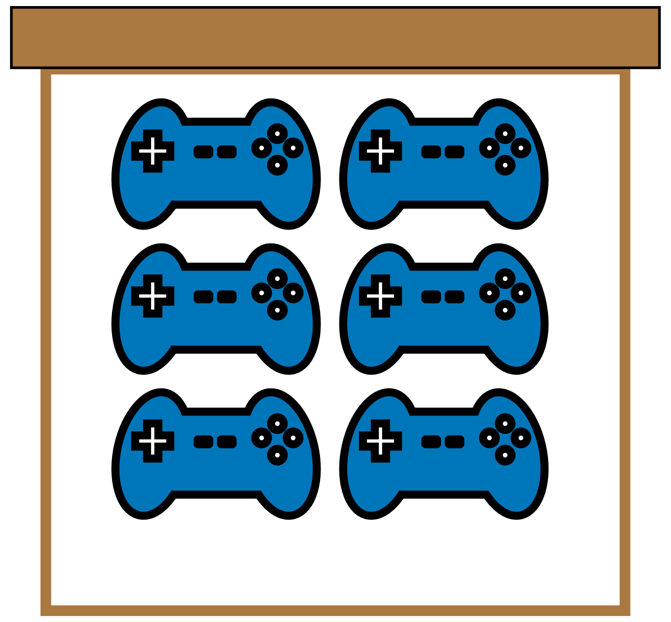


Robust Policies

Assumption: the policy $\boldsymbol{\pi}$ the agents learn is a solution to one of the games in PG.

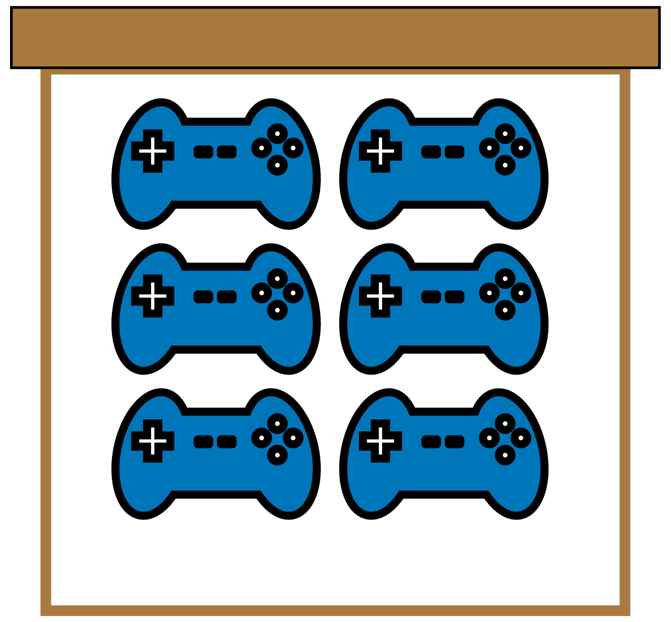
Robust Policies

Assumption: the policy $\boldsymbol{\pi}$ the agents learn is a solution to one of the games in PG.

Sol () $= \boldsymbol{\pi}$

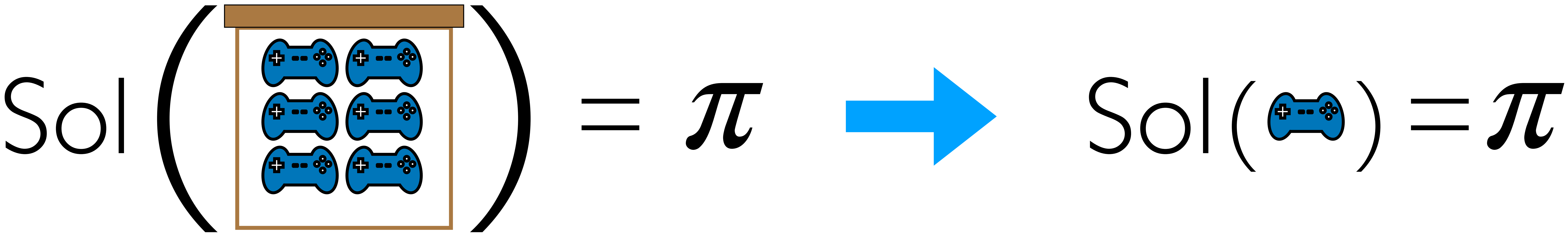
Robust Policies

Assumption: the policy π the agents learn is a solution to one of the games in PG.

Sol () $= \pi$

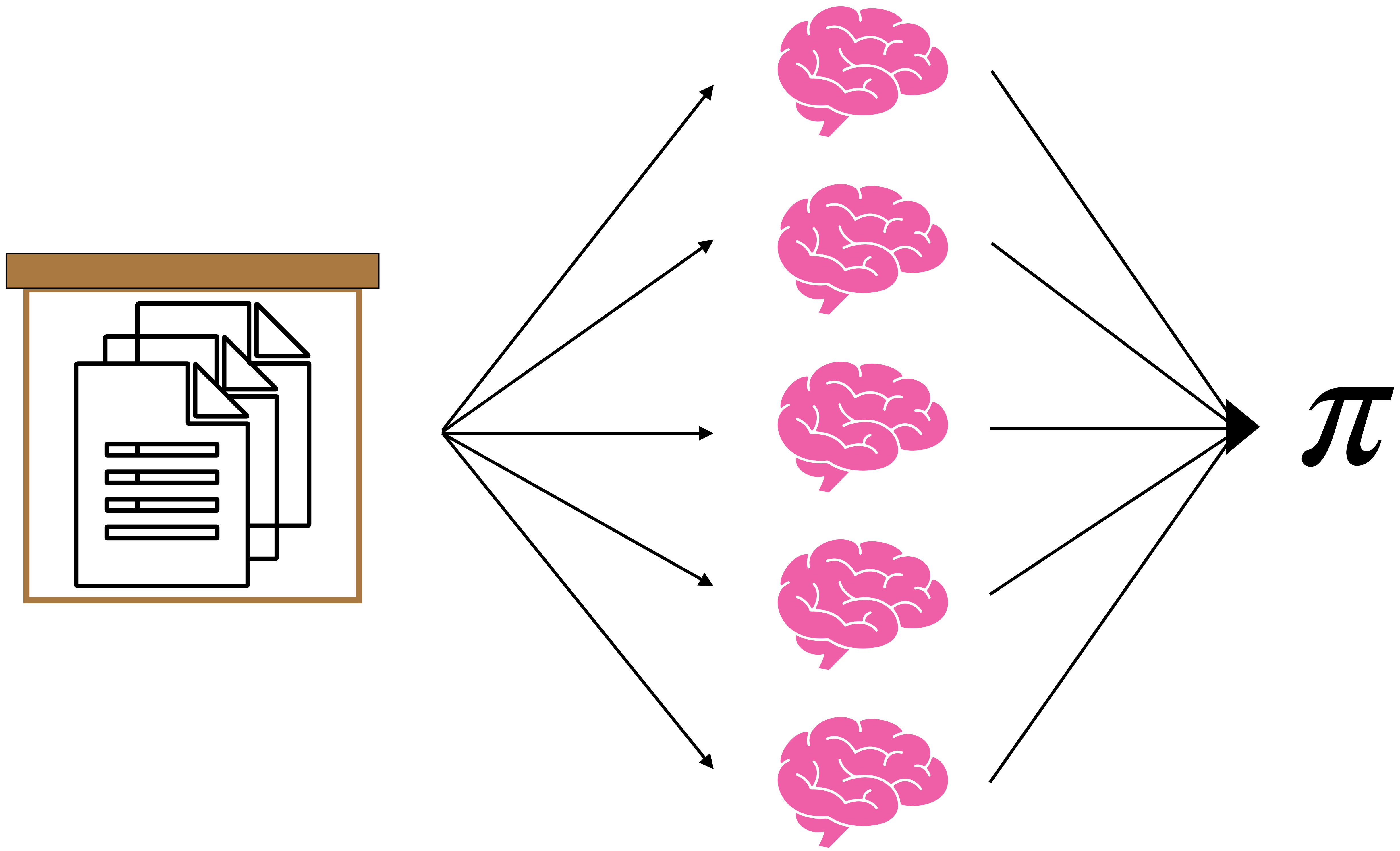
Robust Policies

Assumption: the policy π the agents learn is a solution to one of the games in PG.

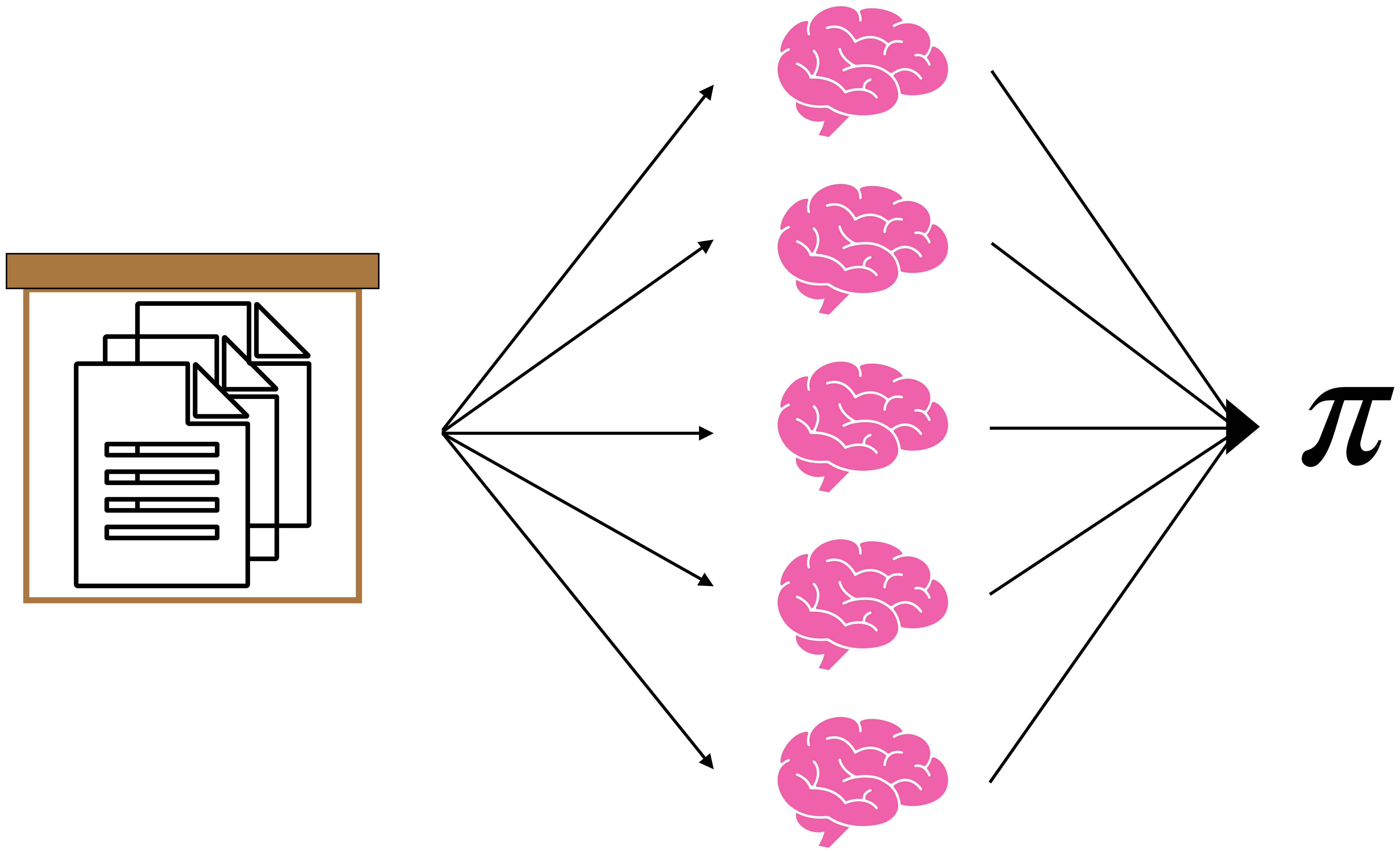


Poisoning

Offline Poisoning

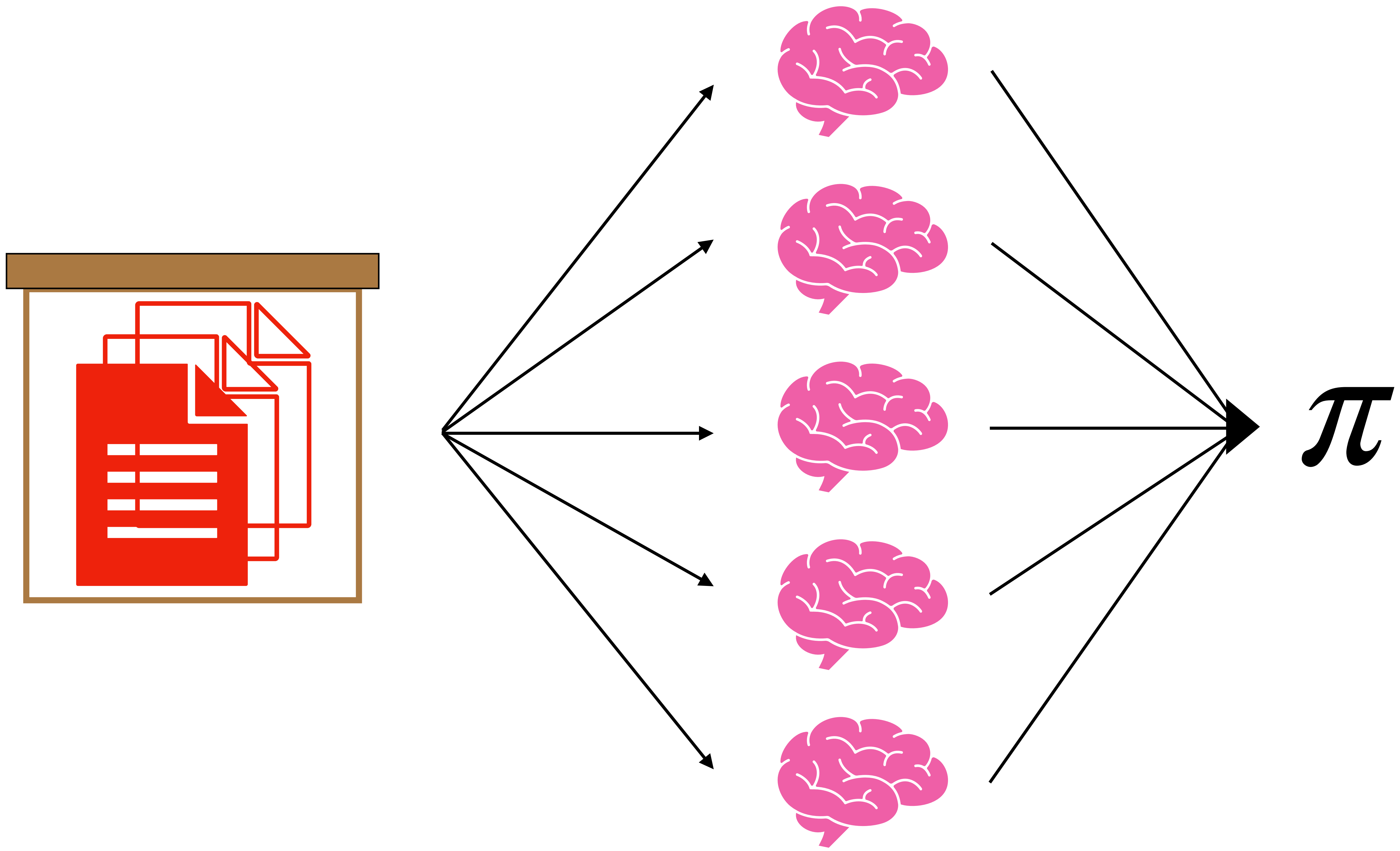


Offline Poisoning

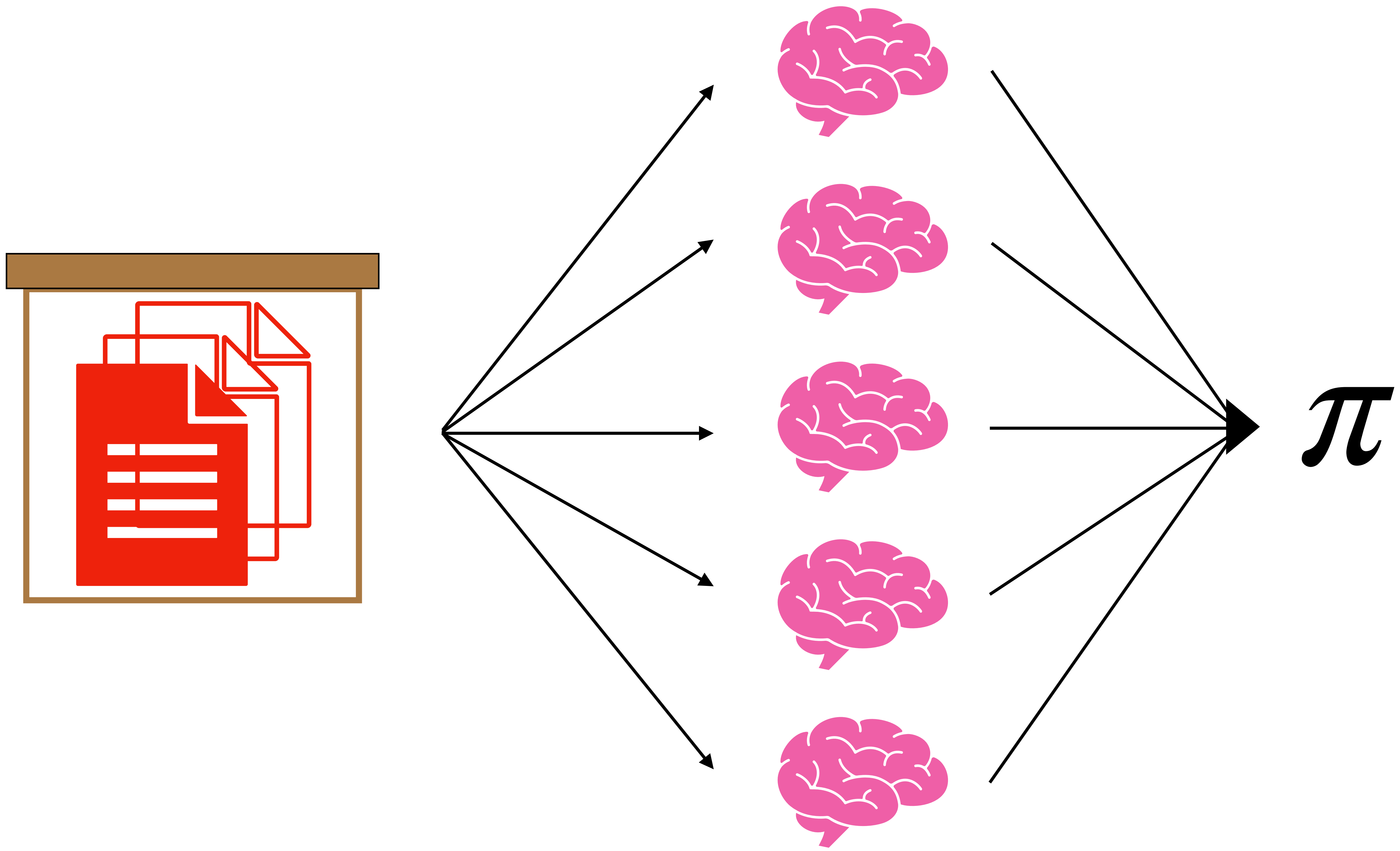


What the agent sees.

Offline Poisoning

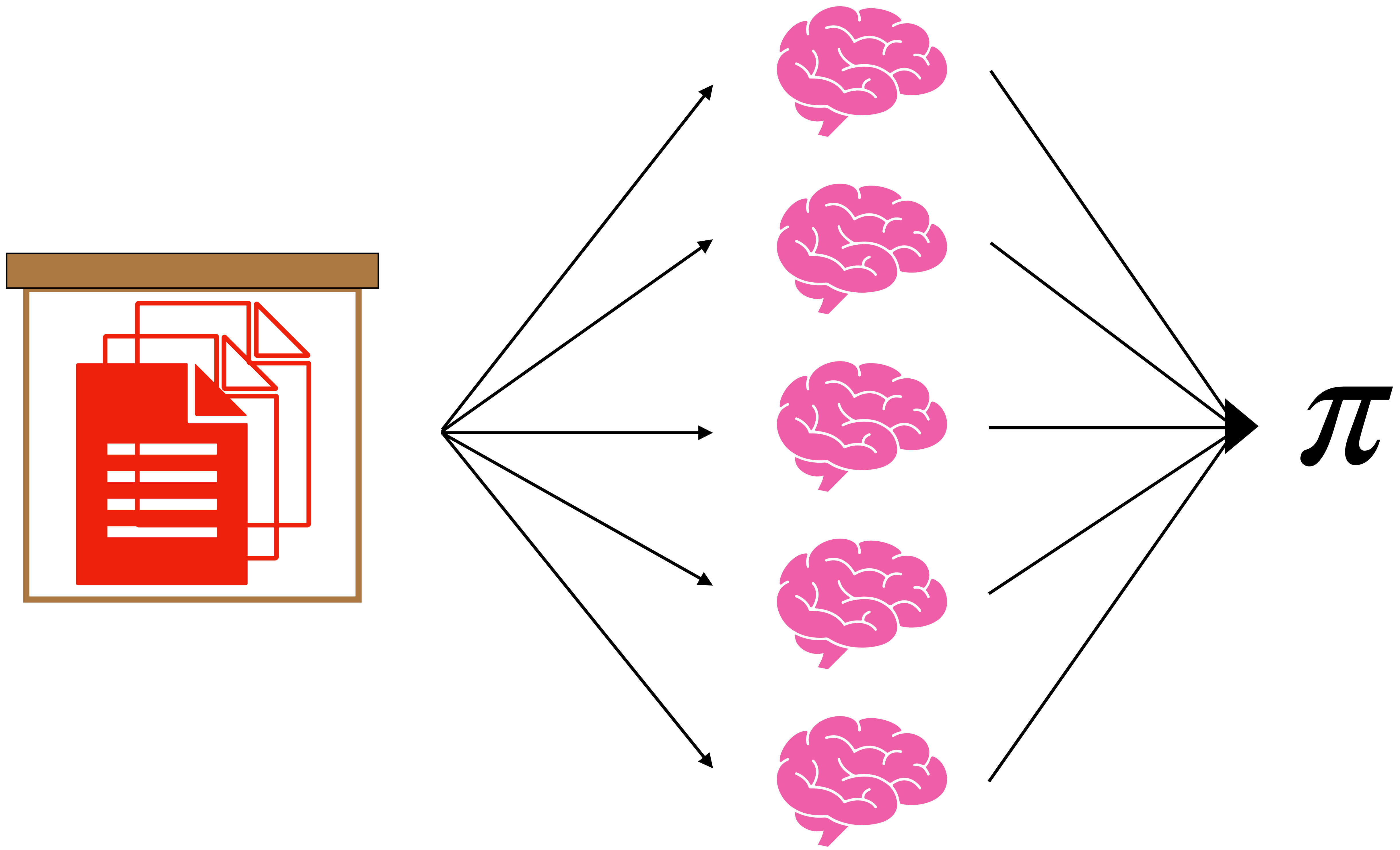


Offline Poisoning

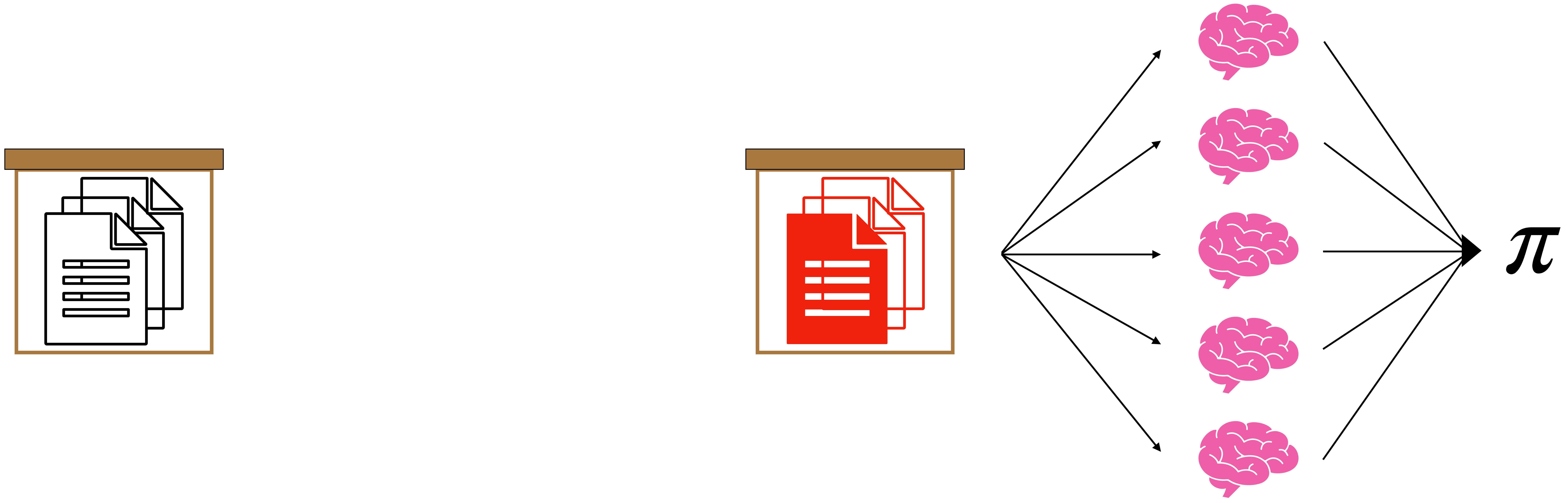


The Data is Corrupted!

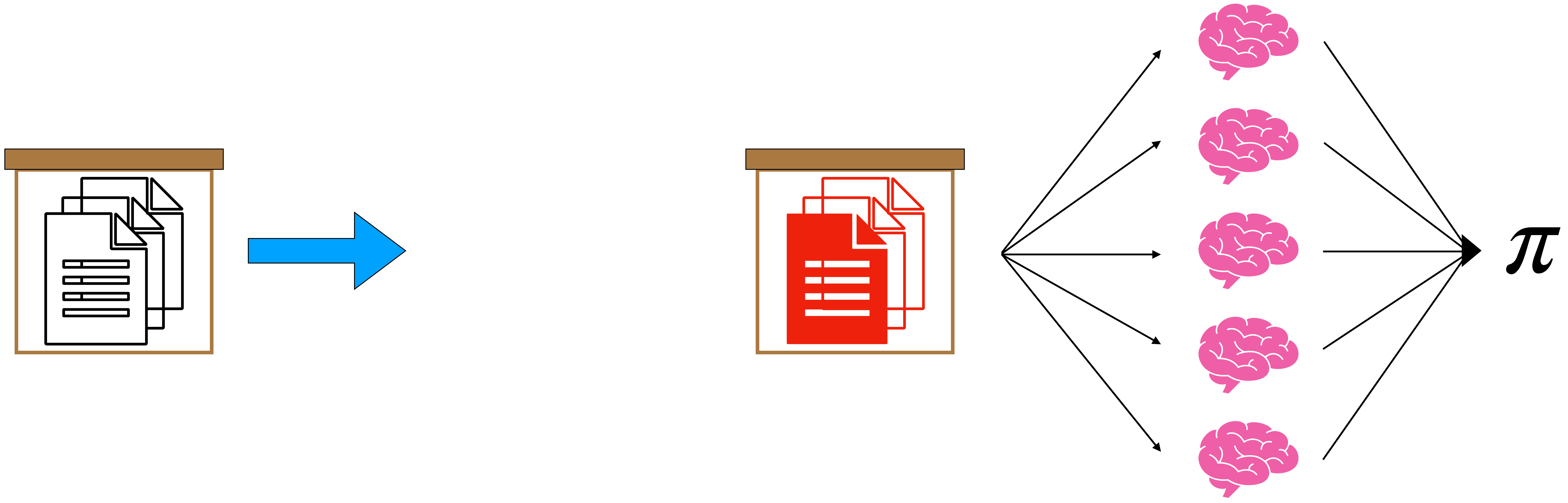
Offline Poisoning



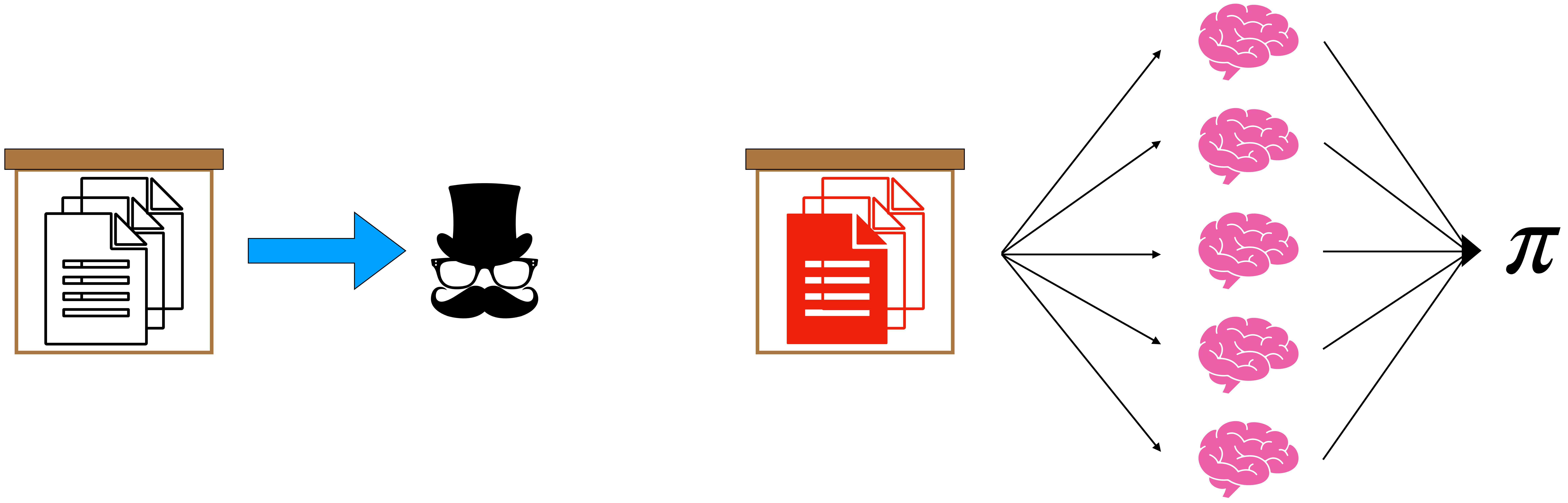
Offline Poisoning



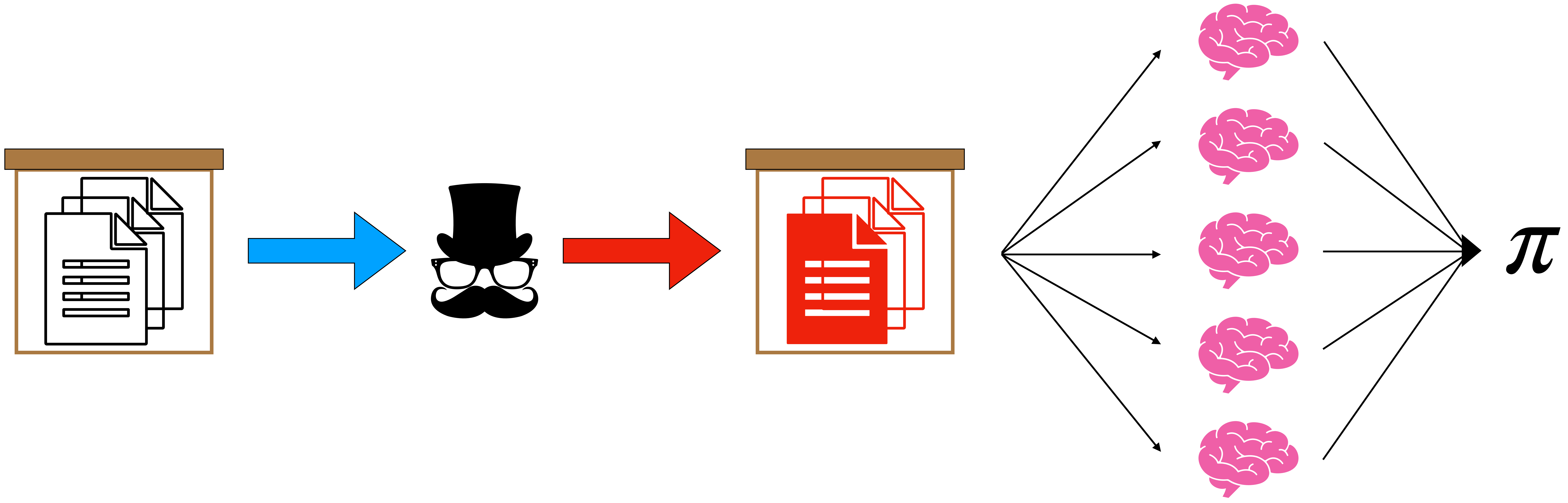
Offline Poisoning



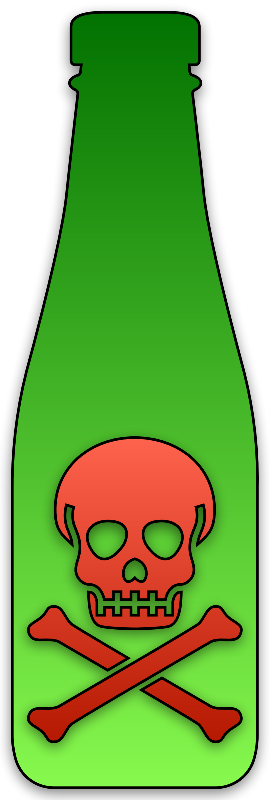
Offline Poisoning



Offline Poisoning

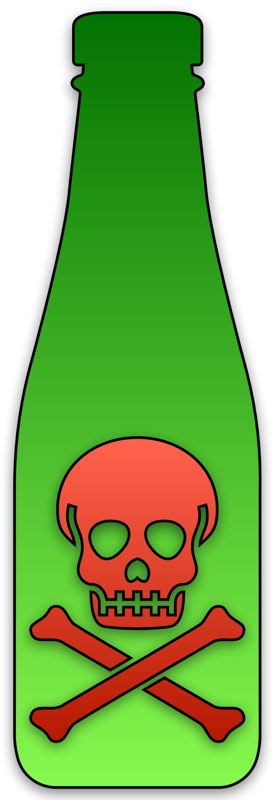


Attack Goal



Attack Goal

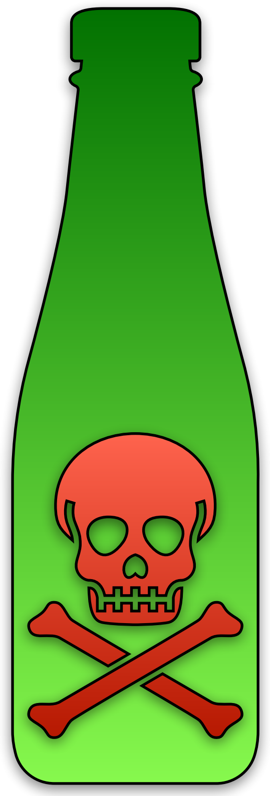
Attacker wants $\pi = \pi^\dagger$.



Attack Goal

Attacker wants $\pi = \pi^{\dagger}$.

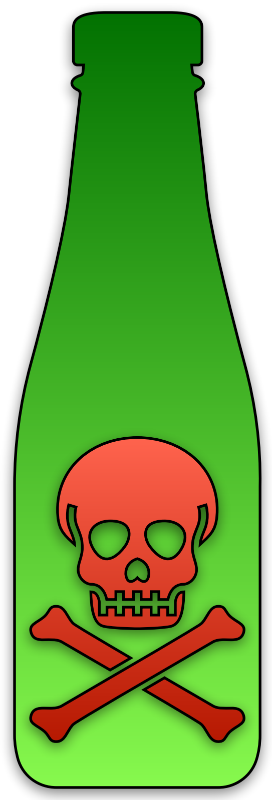
- Attacker can change the *rewards* appearing in the dataset at some *cost*.



Attack Goal

Attacker wants $\pi = \pi^{\dagger}$.

- Attacker can change the *rewards* appearing in the dataset at some *cost*.
- Attacker wants to minimize its cost, usually the $L1$ norm: $||r^0 - r^{\dagger}||_1$.



Attack Goal

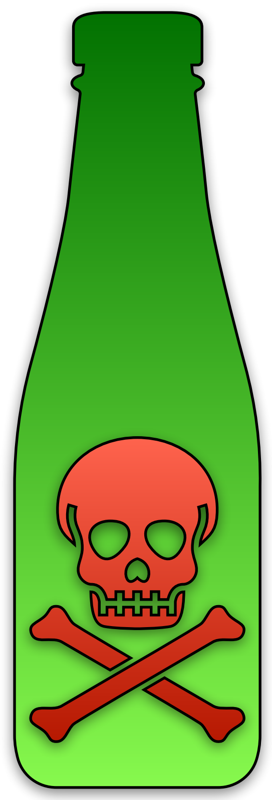
Attacker wants $\pi = \pi^\dagger$.

- Attacker can change the *rewards* appearing in the dataset at some *cost*.
- Attacker wants to minimize its cost, usually the $L1$ norm: $||r^0 - r^\dagger||_1$.

The Attack Problem:

$$\min_{r^\dagger} ||r^0 - r^\dagger||_1$$

s.t. $\pi^\dagger$ is learned from $r^\dagger$



Bottlenecks

Bottlenecks

- Rewards must lie in the natural range $[-b, b]$.

Bottlenecks

- Rewards must lie in the natural range $[-b, b]$.
- Data may be scarce (Low Data Coverage).

Bottlenecks

- Rewards must lie in the natural range $[-b, b]$.
- Data may be scarce (Low Data Coverage).

-b, -b **4x**	-b, -b **10x**
-b, -b **2x**	**0x**

Bottlenecks

- Rewards must lie in the natural range $[-b, b]$.
- Data may be scarce (Low Data Coverage).

-b, -b **4x**	-b, -b **10x**
-b, -b **2x**	**0x**

$$\pi^{\dagger} = (2,2)$$

Can *never* be learned for certain learners!

Bottlenecks

- Rewards must lie in the natural range $[-b, b]$.
- Data may be scarce (Low Data Coverage).

-b, -b **4x**	-b, -b **10x**
-b, -b **2x**	**0x**

$\pi^\dagger = (2,2)$

Can *never* be learned for certain learners!

What can the Attacker do?

Algorithms

Bandit Games

A bandit game is a single normal form game ($S = H = 1$).

Bandit Games

A bandit game is a single normal form game ($S = H = 1$).

-1, -1	1, 1
1, 1	1, 1

Bandit Games

A bandit game is a single normal form game ($S = H = 1$).

-1, -1	1, 1
1, 1	1, 1

To ensure $\pi^\dagger = (1,1)$ is learned, suffices to make it a strict DSE.

Bandit Games

A bandit game is a single normal form game ($S = H = 1$).

-1, -1	1, 1
1, 1	1, 1

To ensure $\pi^\dagger = (1,1)$ is learned, suffices to make it a strict DSE.

-1, -1	1, 1
1, 1	1, 1

Bandit Games

A bandit game is a single normal form game ($S = H = 1$).

-1, -1	1, 1
1, 1	1, 1

To ensure $\pi^\dagger = (1,1)$ is learned, suffices to make it a strict DSE.

-1, -1	1, 1
1, 1	1, 1

Bandit Games

A bandit game is a single normal form game ($S = H = 1$).

-1, -1	1, 1
1, 1	1, 1

To ensure $\pi^{\dagger} = (1,1)$ is learned, suffices to make it a strict DSE.

-1, -1	1, 1
1, 1	1, 1

1, 1	1, 0
0, 1	0, 0

Strict DSE

Strict DSE

$R_1(1,1) > R_1(2,1)$

1, 1	1, 0
0, 1	0, 0

Strict DSE

$R_1(1,1) > R_1(2,1)$

<u>1</u>, 1	1, 0
<u>0</u>, 1	0, 0

$R_1(1,2) > R_1(2,2)$

1, 1	<u>1</u>, 0
0, 1	<u>0</u>, 0

Strict DSE

$R_1(1,1) > R_1(2,1)$

1, 1	1, 0
0, 1	0, 0

$R_1(1,2) > R_1(2,2)$

1, 1	1, 0
0, 1	0, 0

$R_2(1,1) > R_2(1,2)$

1, 1	1, 0
0, 1	0, 0

Strict DSE

$R_1(1,1) > R_1(2,1)$

1, 1	1, 0
0, 1	0, 0

$R_1(1,2) > R_1(2,2)$

1, 1	1, 0
0, 1	0, 0

$R_2(1,1) > R_2(1,2)$

1, 1	1, 0
0, 1	0, 0

$R_2(2,1) > R_2(2,2)$

1, 1	1, 0
0, 1	0, 0

Optimal Poisoning

Optimal Poisoning

Can formulate an LP to compute optimal cost attacks:

Optimal Poisoning

Can formulate an LP to compute optimal cost attacks:

$$R_i(\pi_i^\dagger, a_{-i}) \geq R_i(a_i, a_{-i}) + \epsilon \qquad \forall i, a_i \neq \pi_i^\dagger, a_{-i}$$

Optimal Poisoning

Can formulate an LP to compute optimal cost attacks:

$$R_i(\pi_i^\dagger, a_{-i}) \geq R_i(a_i, a_{-i}) + \epsilon \qquad \forall i, a_i \neq \pi_i^\dagger, a_{-i}$$

1, 1	1, 1-ε
1-ε, 1	1-ε, 1-ε

Dominance

The *dominance equation* ensures $\boldsymbol{\pi}$ is a strict MPDSE for any game with Q-function $\boldsymbol{Q}$:

$$Q_i^{\pi^\dagger}(s,(\pi_i^\dagger(s),a_{-i})) > Q_i^{\pi^\dagger}(s,(a_i',a_{-i})) \quad \forall s,i,a_{-i},a_i'$$

Dominance

The *dominance equation* ensures $\boldsymbol{\pi}$ is a strict MPDSE for any game with Q-function $\boldsymbol{Q}$:

$$Q_i^{\pi^\dagger}(s, (\pi_i^\dagger(s), a_{-i})) > Q_i^{\pi^\dagger}(s, (a_i', a_{-i})) \quad \forall s, i, a_{-i}, a_i'$$

- MPDSE is equivalent to forcing a DSE in each stage game.

Dominance

The *dominance equation* ensures $\boldsymbol{\pi}$ is a strict MPDSE for any game with Q-function $\boldsymbol{Q}$:

$$Q_i^{\pi^\dagger}(s, (\pi_i^\dagger(s), a_{-i})) > Q_i^{\pi^\dagger}(s, (a_i', a_{-i})) \quad \forall s, i, a_{-i}, a_i'$$

- MPDSE is equivalent to forcing a DSE in each stage game.
  - Boils down to *Optimal Game Design*.

Attacker’s Strategy

Attacker's Strategy

- Force $\pi^\dagger$ to be a MPDSE in *every* plausible game.

Attacker's Strategy

- Force $\pi^\dagger$ to be a MPDSE in *every* plausible game.
    - Ensures robust rational agents learn $\pi^\dagger$ by assumption.

Attacker's Strategy

- Force $\pi^\dagger$ to be a MPDSE in *every* plausible game.
  - Ensures robust rational agents learn $\pi^\dagger$ by assumption.
- Let $PQ = \{Q \mid Q = Q_G^{\pi^\dagger}, G \in PG\}$ be the set of plausible Qs.

Attacker's Strategy

- Force $\pi^\dagger$ to be a MPDSE in *every* plausible game.
  - Ensures robust rational agents learn $\pi^\dagger$ by assumption.
- Let $PQ = \{Q \mid Q = Q_G^{\pi^\dagger}, G \in PG\}$ be the set of plausible Qs.
  - Attacker needs dominance to hold for all $Q \in PQ$.

Complications

Complications

PQ alone could be difficult to characterize or compute!

Complications

PQ alone could be difficult to characterize or compute!

Complications

PQ alone could be difficult to characterize or compute!

↓

Instead, focus on nice supersets of PQ.

Extreme Dominance

Sufficient condition: ensure domination between the extreme Q-functions,

Extreme Dominance

Sufficient condition: ensure domination between the extreme Q-functions,

$$\underline{Q}_i^{\pi^\dagger}(s, (\pi_i^\dagger(s), a_{-i})) > \overline{Q}_i^{\pi^\dagger}(s, (a_i', a_{-i})) \quad \forall s, i, a_{-i}, a_i'$$

Extreme Dominance

Sufficient condition: ensure domination between the extreme Q-functions,

$$\underline{Q}_i^{\pi^\dagger}(s, (\pi_i^\dagger(s), a_{-i})) > \overline{Q}_i^{\pi^\dagger}(s, (a_i', a_{-i})) \quad \forall s, i, a_{-i}, a_i'$$

Where, the Q's are the point-wise extremes:

$$\underline{Q}_i^{\pi^\dagger}(s, a) = \min_{G \in PG} Q_{G,i}^{\pi^\dagger}(s, a)$$

$$\overline{Q}_i^{\pi^\dagger}(s, a) = \max_{G \in PG} Q_{G,i}^{\pi^\dagger}(s, a)$$

Linear Programming

Linear Programming

- The Extreme Dominance Constraint is linear.

Linear Programming

- The Extreme Dominance Constraint is linear.
- For CBL, the extreme Q-functions are defined by linear inequalities.

Linear Programming

- The Extreme Dominance Constraint is linear.
- For CBL, the extreme Q-functions are defined by linear inequalities.
    - This extends the previous ideas about games to datasets.

Linear Programming

- The Extreme Dominance Constraint is linear.
- For CBL, the extreme Q-functions are defined by linear inequalities.
    - This extends the previous ideas about games to datasets.

The attacker can efficiently compute minimum cost attacks using a Linear Program

Solutions

Feasibility

Feasibility

Can the attacker make any $\boldsymbol{\pi}^\dagger$ a MPDSE?

Feasibility

Can the attacker make any $\boldsymbol{\pi}^{\dagger}$ a MPDSE?

Theorem: Poisoning CBL is feasible if the following condition holds:

$$\iota \leq 2b - (H+1)\,\rho_h^R(s,a), \;\; \forall\, h \in [H], s \in S, a \in A$$

Feasibility

Can the attacker make any $\pi^\dagger$ a MPDSE?

Theorem: Poisoning CBL is feasible if the following condition holds:

$$\iota \leq 2b - (H+1)\,\rho_h^R(s,a), \;\; \forall\, h \in [H], s \in S, a \in A$$

What does this mean?

Coverage Requirements

Feasibility through data coverage.

Coverage Requirements

Feasibility through data coverage.

Corollary: Poisoning CBL is feasible if the following condition holds:

$$N_h(s,a) \geq \frac{4b^2 (H+1)^2 \log \left(\left(H |S| \, |A| \right) / \delta \right)}{\left(2b - \iota\right)^2} = \tilde{\Omega}(H^2) \, .$$

Coverage Requirements

Feasibility through data coverage.

Corollary: Poisoning CBL is feasible if the following condition holds:

$$N_h(s,a) \geq \frac{4b^2(H+1)^2 \log\left(\left(H|S||A|\right)/\delta\right)}{\left(2b-\iota\right)^2} = \tilde{\Omega}(H^2)\,.$$

Yes, if given enough data!

Coverage Requirements

Feasibility through data coverage.

Corollary: Poisoning CBL is feasible if the following condition holds:

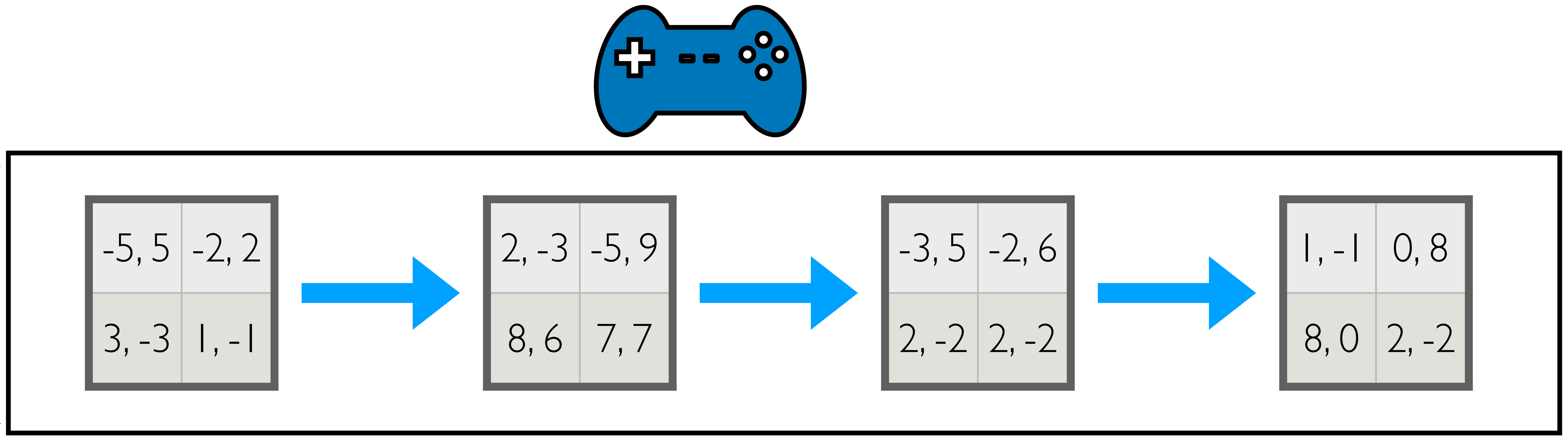
$$N_h(s,a) \geq \frac{4b^2(H+1)^2 \log\left(\left(H|S||A|\right)/\delta\right)}{\left(2b-\iota\right)^2} = \tilde{\Omega}(H^2)\,.$$

Yes, if given enough data!

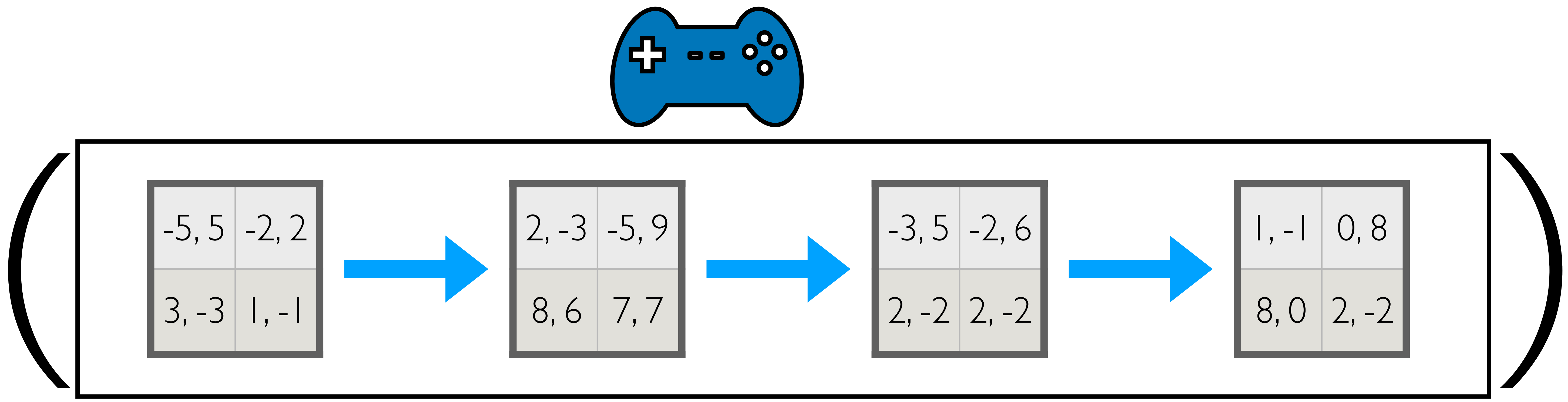
This implies: $K \geq H^3SA$.

Cost Analysis

Cost Analysis

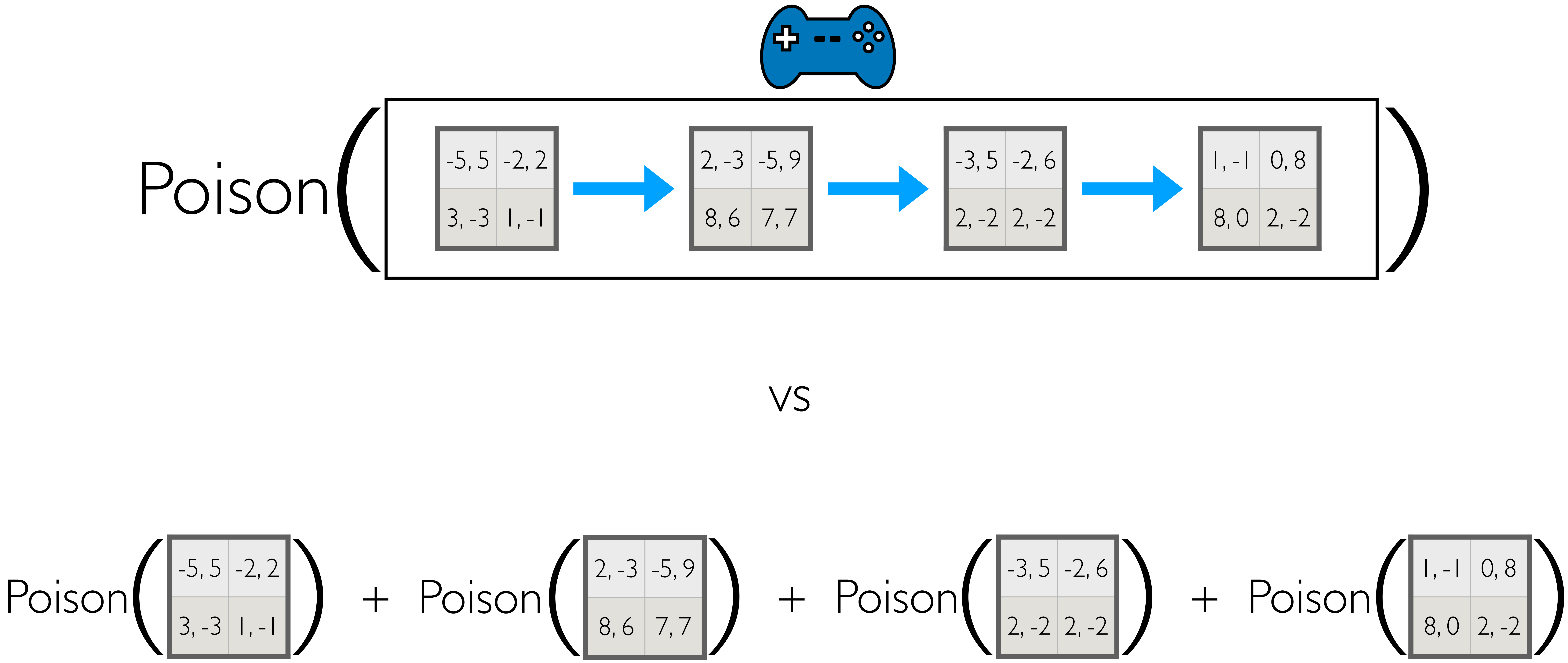
Poison()

Cost Analysis

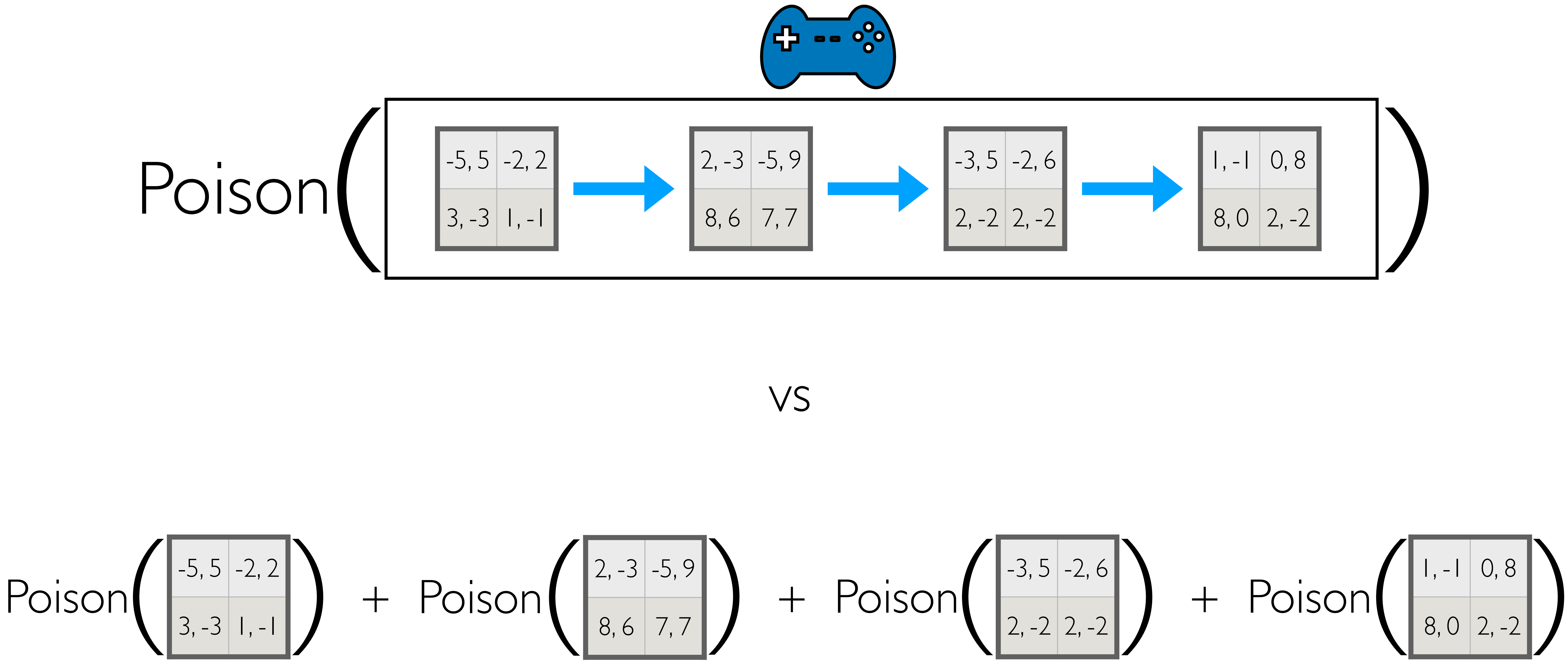
Poison


vs

Cost Analysis

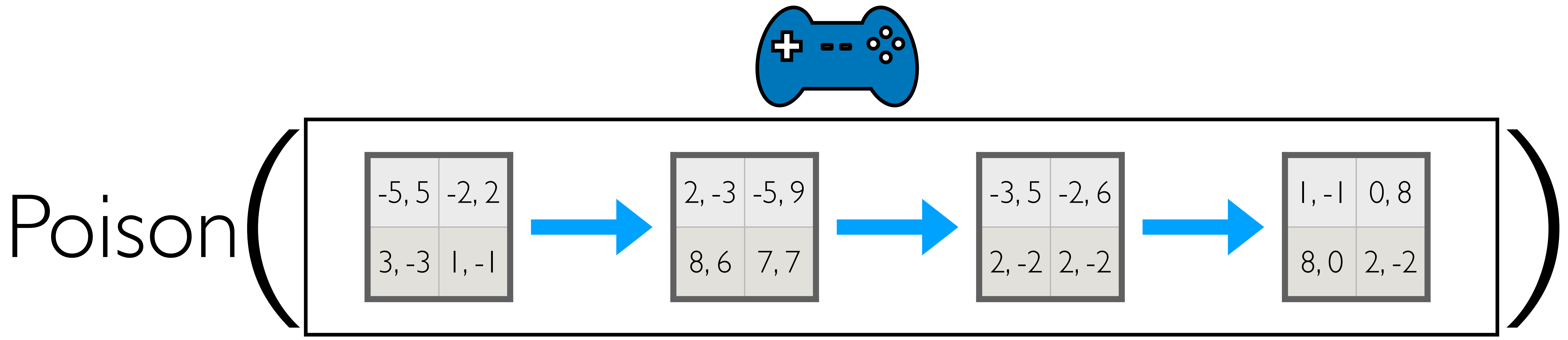


Cost Analysis

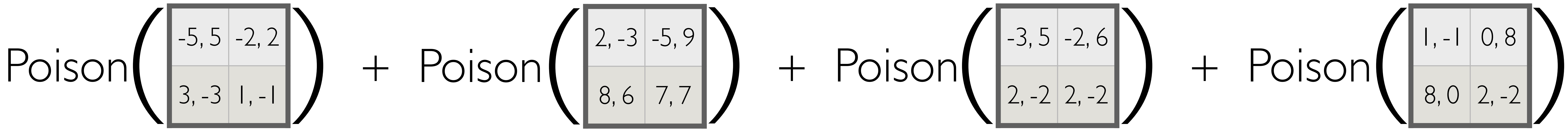


***Poisoning is not separable over stage games.**

Cost Analysis



vs



Can exactly characterize!

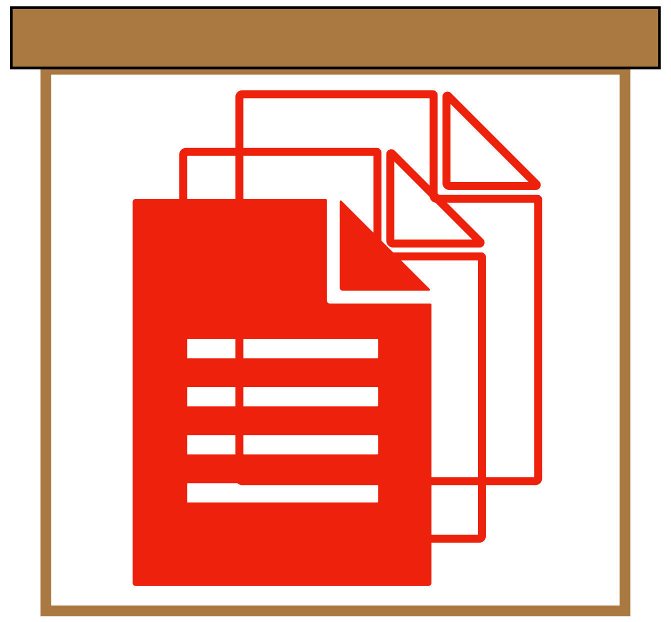
***Poisoning is not separable over stage games.**

Bound Reduction

Cost Bounds on Optimal Data Poisoning are derived through Bandit Data Poisoning.

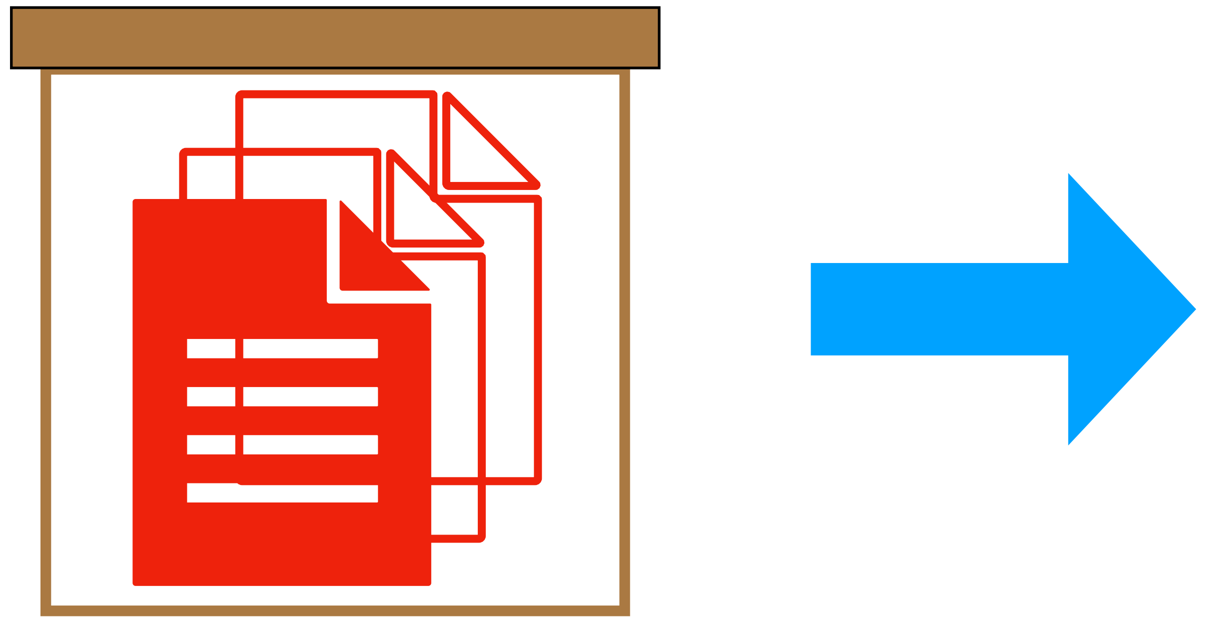
Bound Reduction

Cost Bounds on Optimal Data Poisoning are derived through Bandit Data Poisoning.



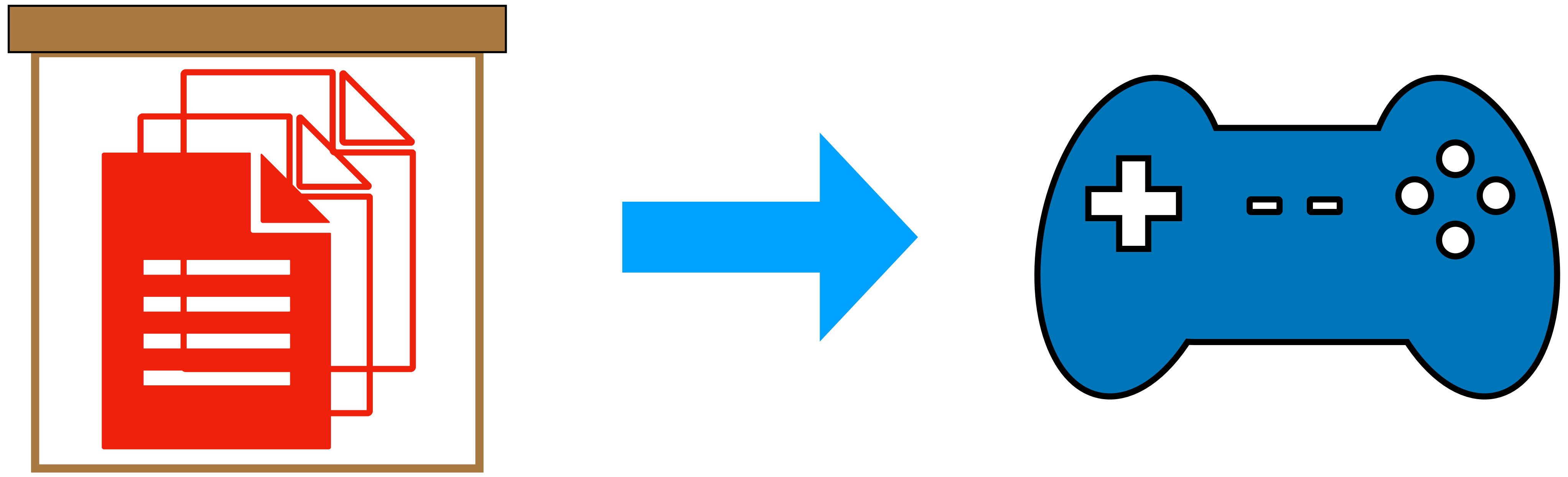
Bound Reduction

Cost Bounds on Optimal Data Poisoning are derived through Bandit Data Poisoning.



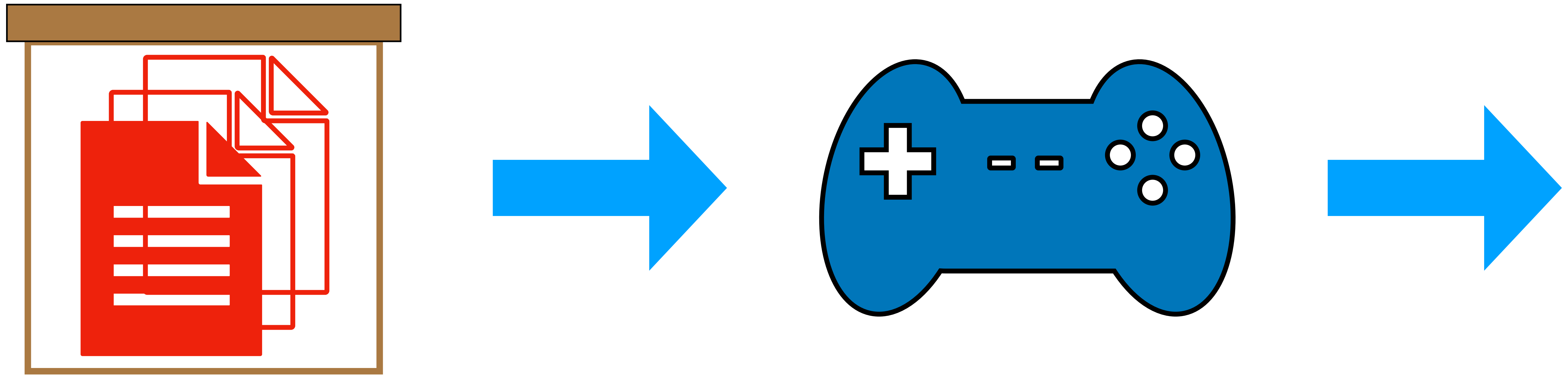
Bound Reduction

Cost Bounds on Optimal Data Poisoning are derived through Bandit Data Poisoning.



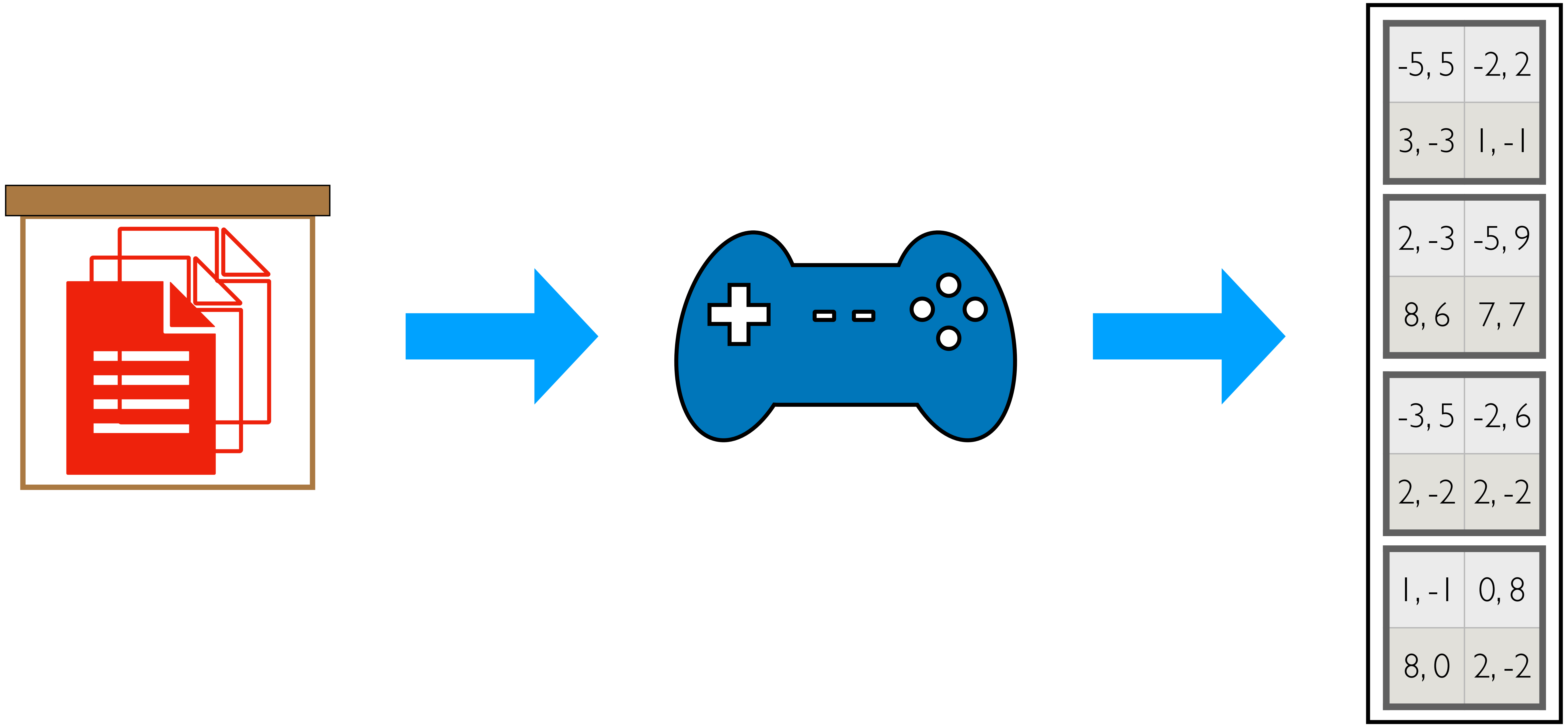
Bound Reduction

Cost Bounds on Optimal Data Poisoning are derived through Bandit Data Poisoning.



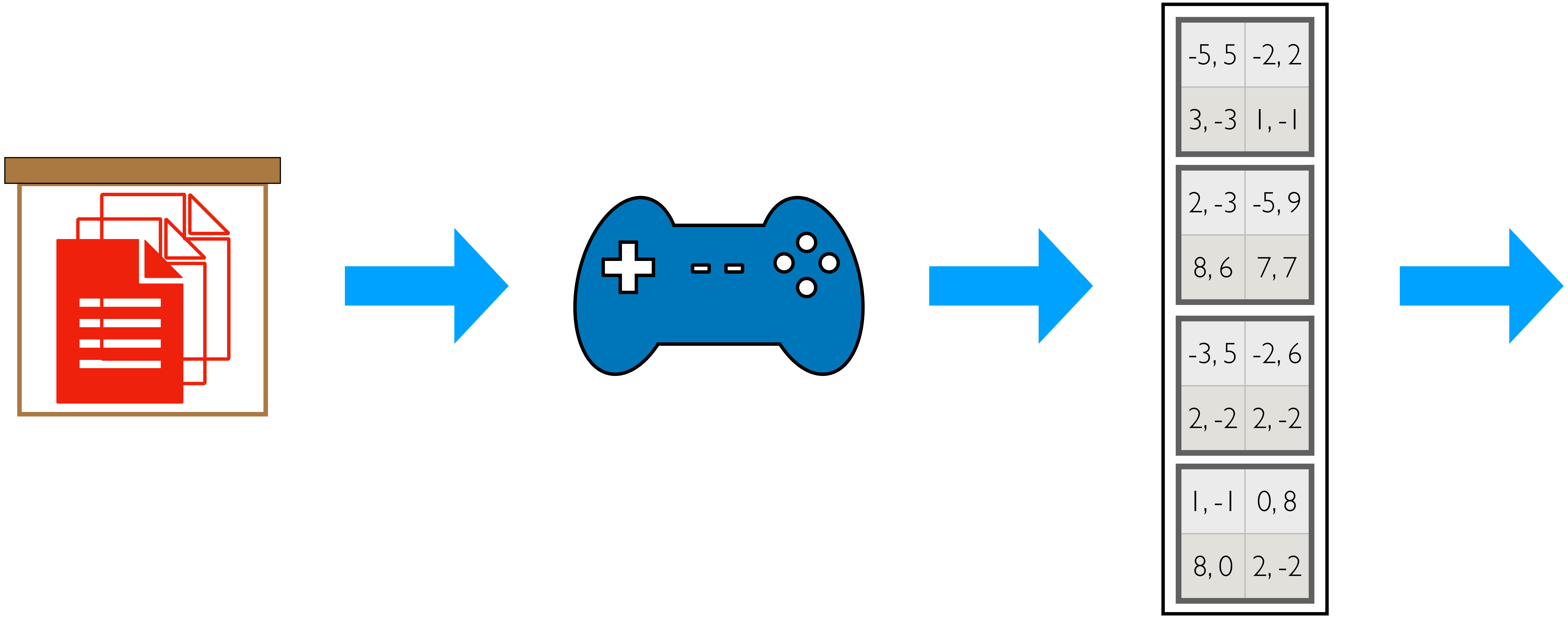
Bound Reduction

Cost Bounds on Optimal Data Poisoning are derived through Bandit Data Poisoning.



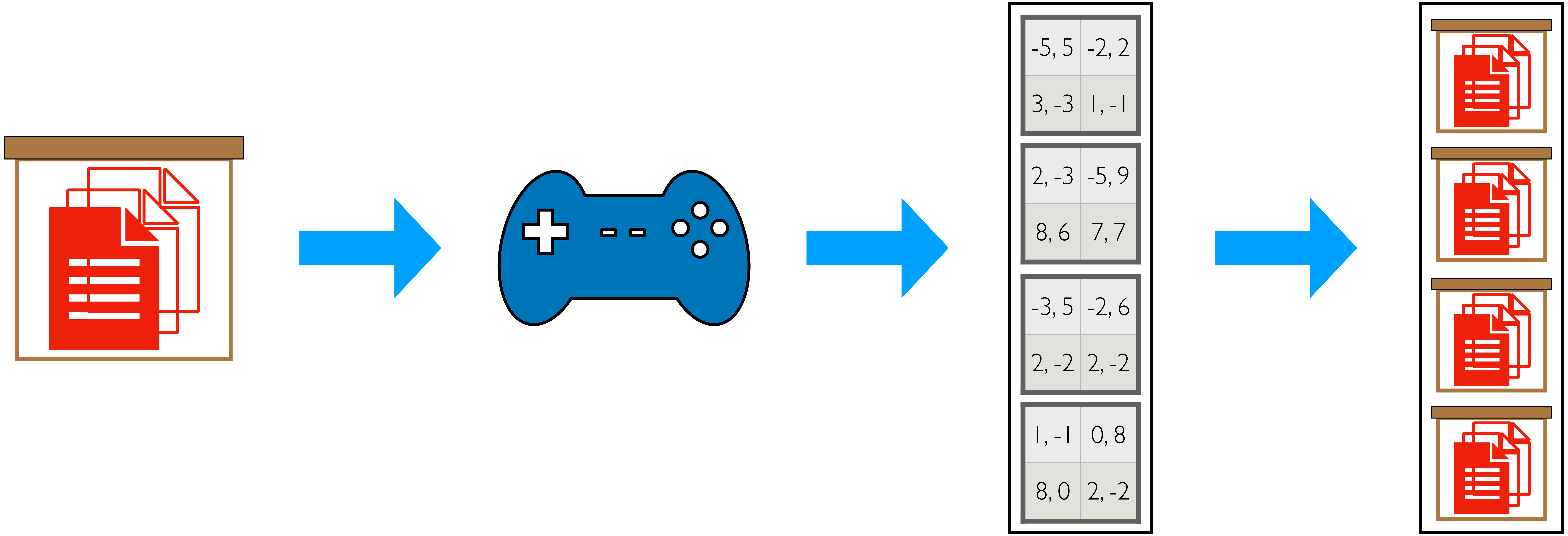
Bound Reduction

Cost Bounds on Optimal Data Poisoning are derived through Bandit Data Poisoning.



Bound Reduction

Cost Bounds on Optimal Data Poisoning are derived through Bandit Data Poisoning.



Cost Lower-bounds

Cost Lower-bounds

$\mathcal{A}_1/\mathcal{A}_2$	1	2	...	$\|\mathcal{A}_2\|$
1	$-b, -b$	$-b, b$	...	$-b, b$
2	$b, -b$	b, b	...	b, b
...	...	...	...	...
$\|\mathcal{A}_1\|$	$b, -b$	b, b	...	b, b

Before Attack

Cost Lower-bounds

$\mathcal{A}_1/\mathcal{A}_2$	1	2	...	$\lvert\mathcal{A}_2\rvert$
1	$-b, -b$	$-b, b$	...	$-b, b$
2	$b, -b$	b, b	...	b, b
...	...	...	...	...
$\lvert\mathcal{A}_1\rvert$	$b, -b$	b, b	...	b, b

Before Attack

Cost Lower-bounds

$\mathcal{A}_1/\mathcal{A}_2$	1	2	...	$\lvert\mathcal{A}_2\rvert$
1	$-b, -b$	$-b, b$	...	$-b, b$
2	$b, -b$	b, b	...	b, b
...	...	...	...	...
$\lvert\mathcal{A}_1\rvert$	$b, -b$	b, b	...	b, b

Before Attack

$\mathcal{A}_1/\mathcal{A}_2$	1	2
1	b, b	$b, b-2\rho-\iota$
2	$b-2\rho-\iota, b$	$b-2\rho-\iota, b-2\rho-\iota$
...	...	...
$\lvert\mathcal{A}_1\rvert$	$b-2\rho-\iota, b$	$b-2\rho-\iota, b-2\rho-\iota$

After Attack

Cost Lower-bounds

$\mathcal{A}_1/\mathcal{A}_2$	1	2	...	$\lvert\mathcal{A}_2\rvert$
1	$-b, -b$	$-b, b$	...	$-b, b$
2	$b, -b$	b, b	...	b, b
...	...	...	...	...
$\lvert\mathcal{A}_1\rvert$	$b, -b$	b, b	...	b, b

Before Attack

$\mathcal{A}_1/\mathcal{A}_2$	1	2
1	b, b	$b, b-2\rho-\iota$
2	$b-2\rho-\iota, b$	$b-2\rho-\iota, b-2\rho-\iota$
...	...	...
$\lvert\mathcal{A}_1\rvert$	$b-2\rho-\iota, b$	$b-2\rho-\iota, b-2\rho-\iota$

After Attack

Optimal Attack Cost:

$$H\,|S|\min_{h,s,a} N_h(s,a)\,|A|^{n-1}(2b+2\rho+\iota)$$

Cost Lower-bounds

$\mathcal{A}_1/\mathcal{A}_2$	1	2	...	$\lvert\mathcal{A}_2\rvert$
1	$-b,-b$	$-b,b$	...	$-b,b$
2	$b,-b$	b,b	...	b,b
...	...	...	...	...
$\lvert\mathcal{A}_1\rvert$	$b,-b$	b,b	...	b,b

Before Attack

$\mathcal{A}_1/\mathcal{A}_2$	1	2
1	b,b	$b,b-2\rho-\iota$
2	$b-2\rho-\iota,b$	$b-2\rho-\iota,b-2\rho-\iota$
...	...	...
$\lvert\mathcal{A}_1\rvert$	$b-2\rho-\iota,b$	$b-2\rho-\iota,b-2\rho-\iota$

After Attack

Optimal Attack Cost:

$$H\,|S|\min_{h,s,a} N_h(s,a)\,|A|^{n-1}(2b+2\rho+\iota)$$

Exponential dependency on n!

The Roles of ρ

The Roles of ρ

ρ^P If the uncertainty in transition is high,

$$C(D) \geq \sum_{i=1}^{H} C(D_h)$$

The Roles of ρ

ρ^P If the uncertainty in transition is high,

$$C(D) \geq \sum_{i=1}^{H} C(D_h)$$

ρ^R If the uncertainty in reward is low,

$$C(D) \leq \sum_{i=1}^{H} C(D_h)$$

The Roles of ρ

ρ^P If the uncertainty in transition is high,

$$C(D) \geq \sum_{i=1}^{H} C(D_h)$$

ρ^R If the uncertainty in reward is low,

$$C(D) \leq \sum_{i=1}^{H} C(D_h)$$

The optimal cost could potentially be greater than optimally poisoning each subdataset!

Conclusion

Summary

- In large datasets, poisoning is always feasible, though costly.
- Thus, we illustrate the need for provable defenses against offline reward poisoning.