Mila

NeurIPS 2024

paper link

Towards Reliable Reinforcement Learning Systems

Job Talk

Arushi Jain

Advisor: Doina Precup
McGill University,
Google DeepMind

Hi, I am Arushi 👋

Bachelors

IIIT-Delhi

Masters & PhD
McGill University, MILA

Internships

Microsoft Research, India

Meta AI Lab

Microsoft Research, Amsterdam

Amazon, Montreal

Motivation for my PhD

*“How can we make reinforcement learning (RL) truly reliable for
real-world application —
ensuring agent behave safely, consistently and learn efficiently from limited data?”*

GVFExplorer:

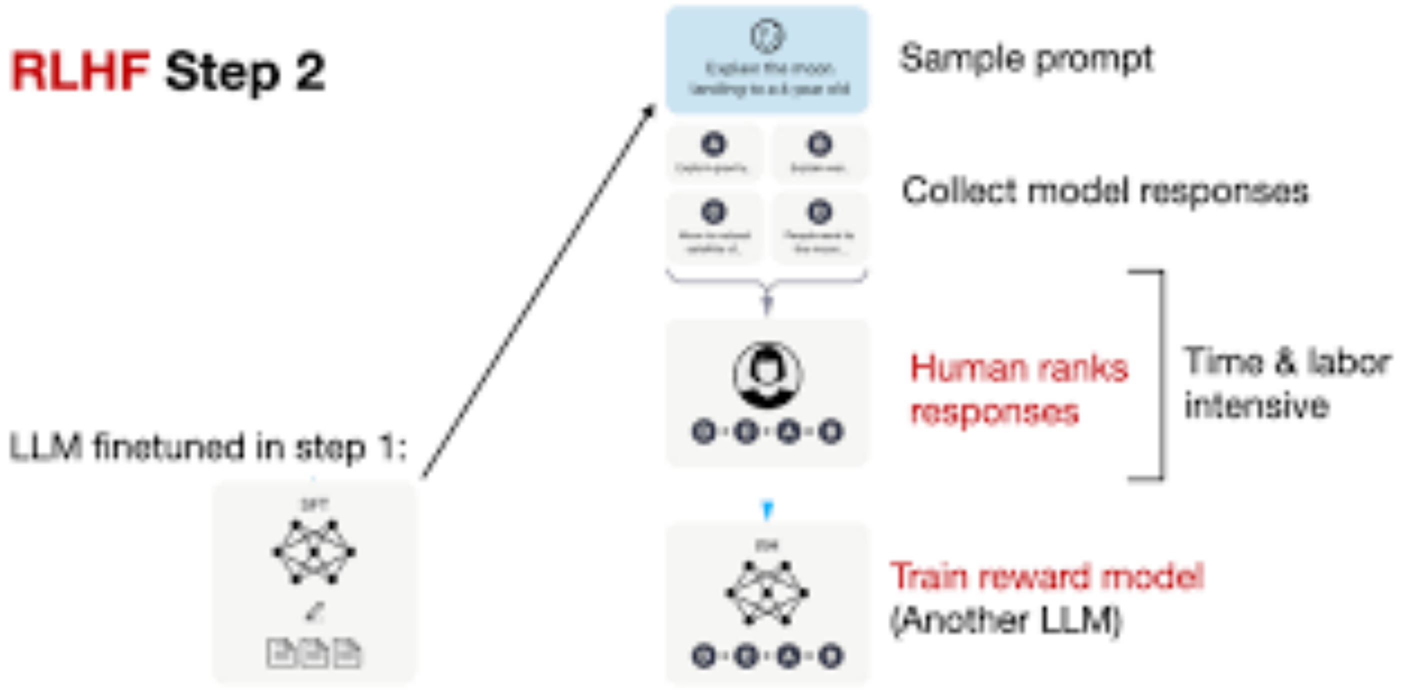
Adaptive Exploration for Data-Efficient General Value Functions

Arushi Jain, Josiah Hanah, Doina Precup

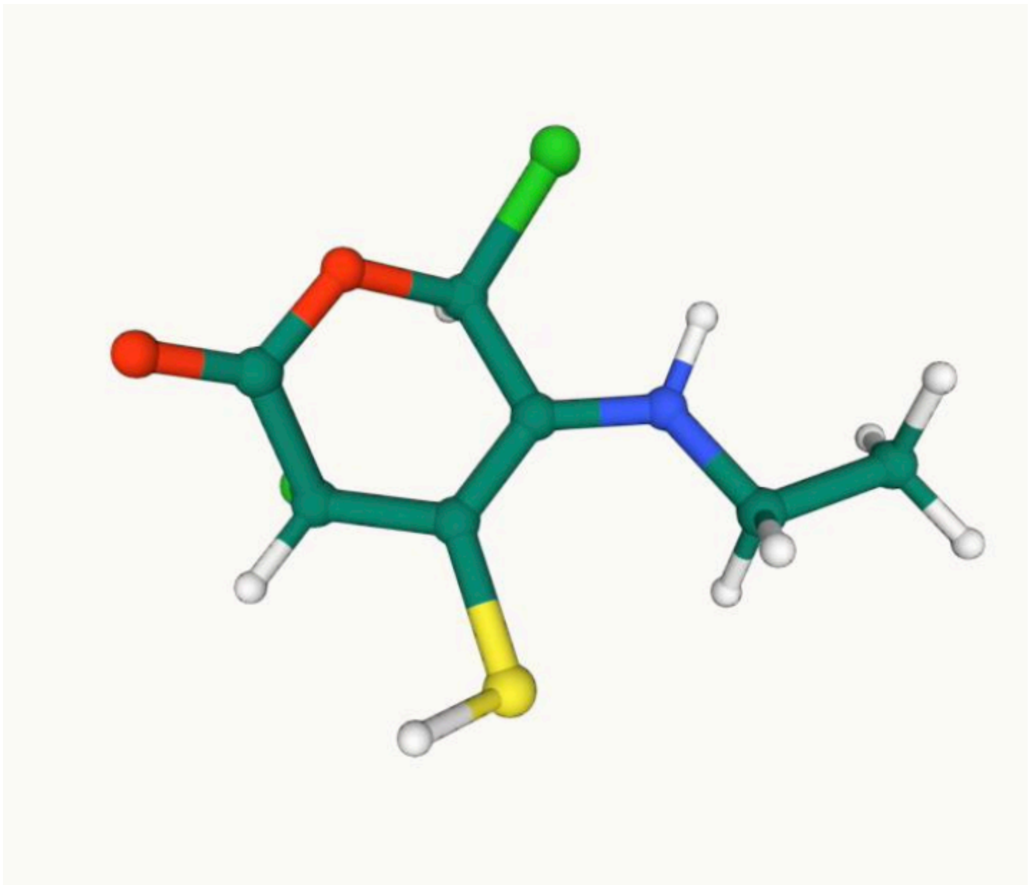
NeurIPS 2024

paper link

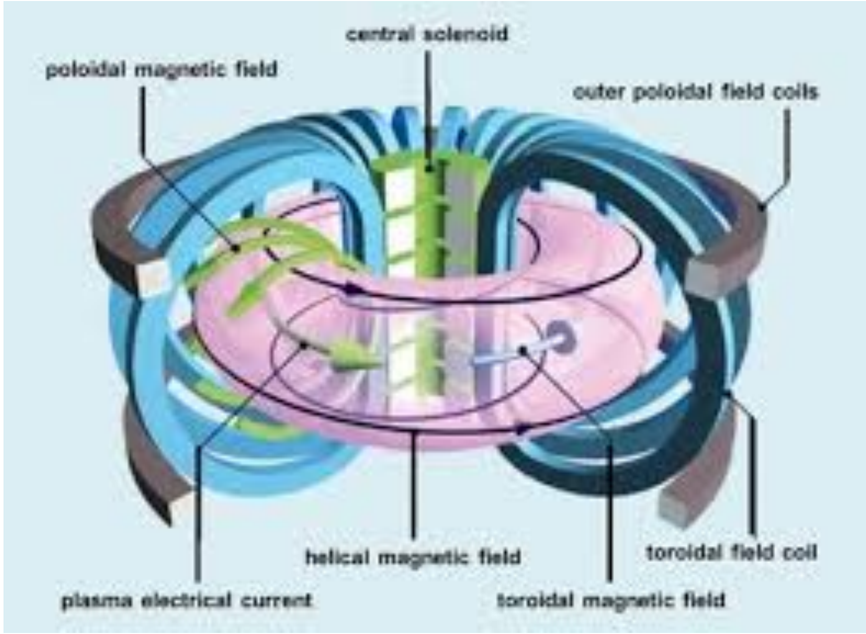
Example of successful RL



RLHF



Molecule RetroSynthesis

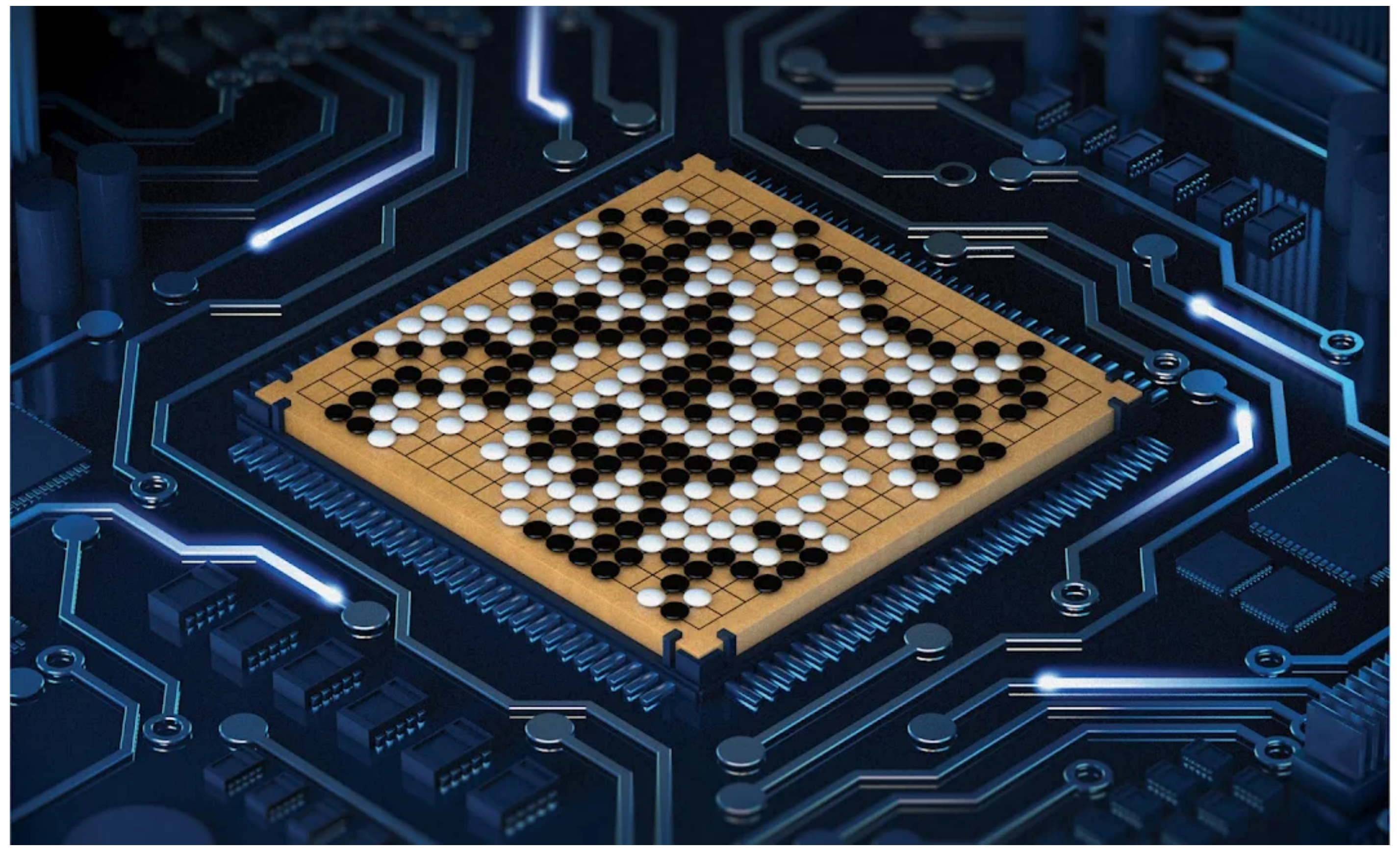


Nuclear Fusion

Robotics

Example of successful RL

RL -> Emergent Behaviours!



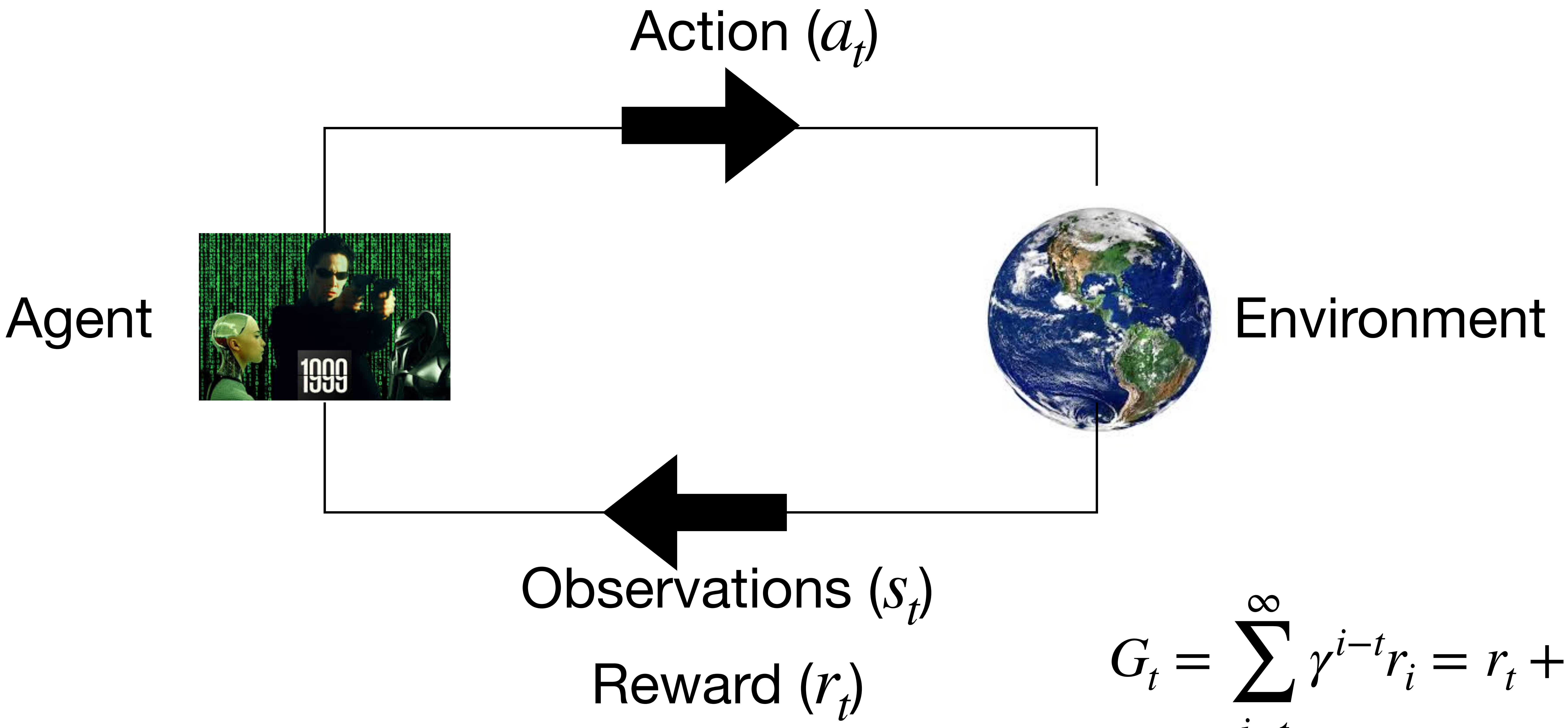
Alpha Go

Discovered novel innovative moves using RL that were not known to human kind

High complex state space: 10^{170}

Reinforcement Learning

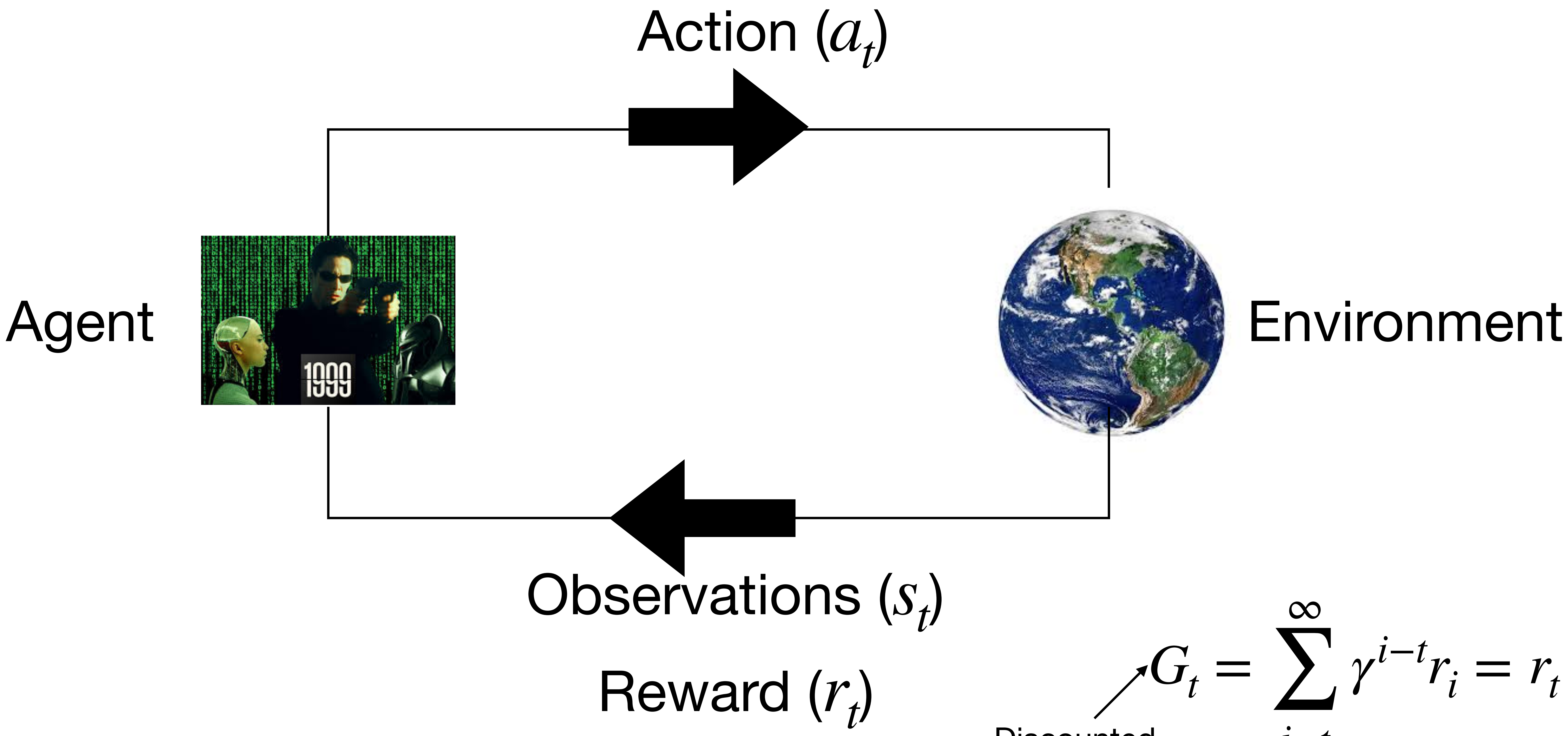
Agent interacts with the environment to maximize the reward r



$$G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i = r_t + \gamma^1 r_{t+1} + \ldots + \gamma^n r_{t+n} + \ldots$$

Reinforcement Learning

Agent interacts with the environment to maximize the reward r



$$G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i = r_t + \gamma^1 r_{t+1} + \ldots + \gamma^n r_{t+n} + \ldots$$

Discounted total reward **(Return)** → G_t

0<discount factor <1 → γ

Defining Q-function

$$G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i = r_t + \gamma^1 r_{t+1} + \ldots + \gamma^n r_{t+n} + \ldots$$

Total reward G_t, is discounted sum of future rewards

Defining Q-function

$$G_t = \sum_{i=t}^{\infty} \gamma^{i-t} r_i = r_t + \gamma^1 r_{t+1} + \ldots + \gamma^n r_{t+n} + \ldots$$

Total reward G_t, is discounted sum of future rewards

$$Q(s_t, a_t) = \mathbb{E}[G_t \,|\, s_t, a_t]$$

Q function captures expected future rewards an agent gets

starting from given state s_t and taking action a_t

What we want an RL agent to do?

$$Q(s_t, a_t) = \mathbb{E}[G_t \mid s_t, a_t]$$

Ultimately, agent needs to find policy $\pi(a \mid s)$ which learns to take best action a at any given state s.

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

Policy π needs to choose action which maximize expected cumulative rewards

Q function and Value function

$$Q(s_t, a_t) = \mathbb{E}[G_t \mid s_t, a_t]$$

state action

Q = expected performance,

given s and a

$$V(s_t) = \mathbb{E}[G_t \mid s_t] = \mathbb{E}[Q(s_t, a_t)]$$

V = expected performance,

given s

Temporal Difference (TD) Learning

Temporal Difference (TD) learning updates value function by bootstrapping the target with 1-step value

$$V(s_t) = \mathbb{E}[G_t \mid s_t]$$

$$= \mathbb{E}[r(s_t) + \gamma r(s_{t+1}) + \ldots \mid s_t]$$

$$V(s_t) = V(s_t) + \alpha \left(r(s_t) + \gamma V(s_{t+1}) - V(s_t) \right)$$

Policy Evaluations and Policy Control

Policy Evaluation

‘How good is this policy?’

Policy Control

Policy Evaluations and Policy Control

Policy Evaluation

'How good is this policy?'

Policy Control

'What should I do to perform better?'

Policy Evaluations and Policy Control

Policy Evaluation

'How good is this policy?'

Given $\pi \rightarrow \mathbb{E}_{a \sim \pi}[G_t \,|\, s = s_t]$

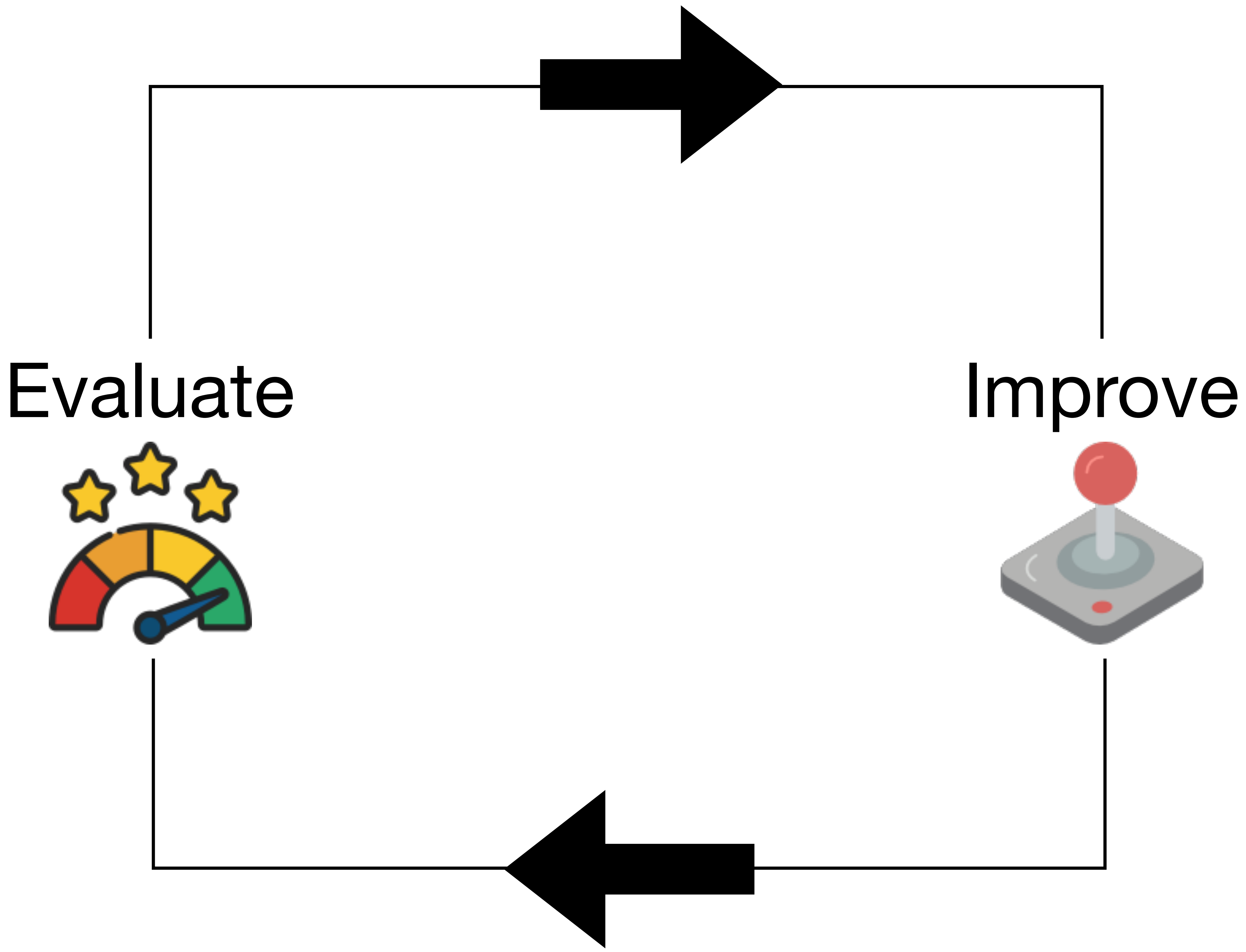
expected total reward following policy π

Policy Control

'What should I do to perform better?'

Given $\pi \rightarrow$ find improved π'

RL Loop



Why Policy Evaluation is important?

Examples

Hypothesis A

outcome?

helpfulness?

user engagement?

‘copilot needs to evaluate effectiveness of policy’

Why Policy Evaluation is important?

Examples

run Policy π_A → evaluate performance

→

Hypothesis A

→

outcome?

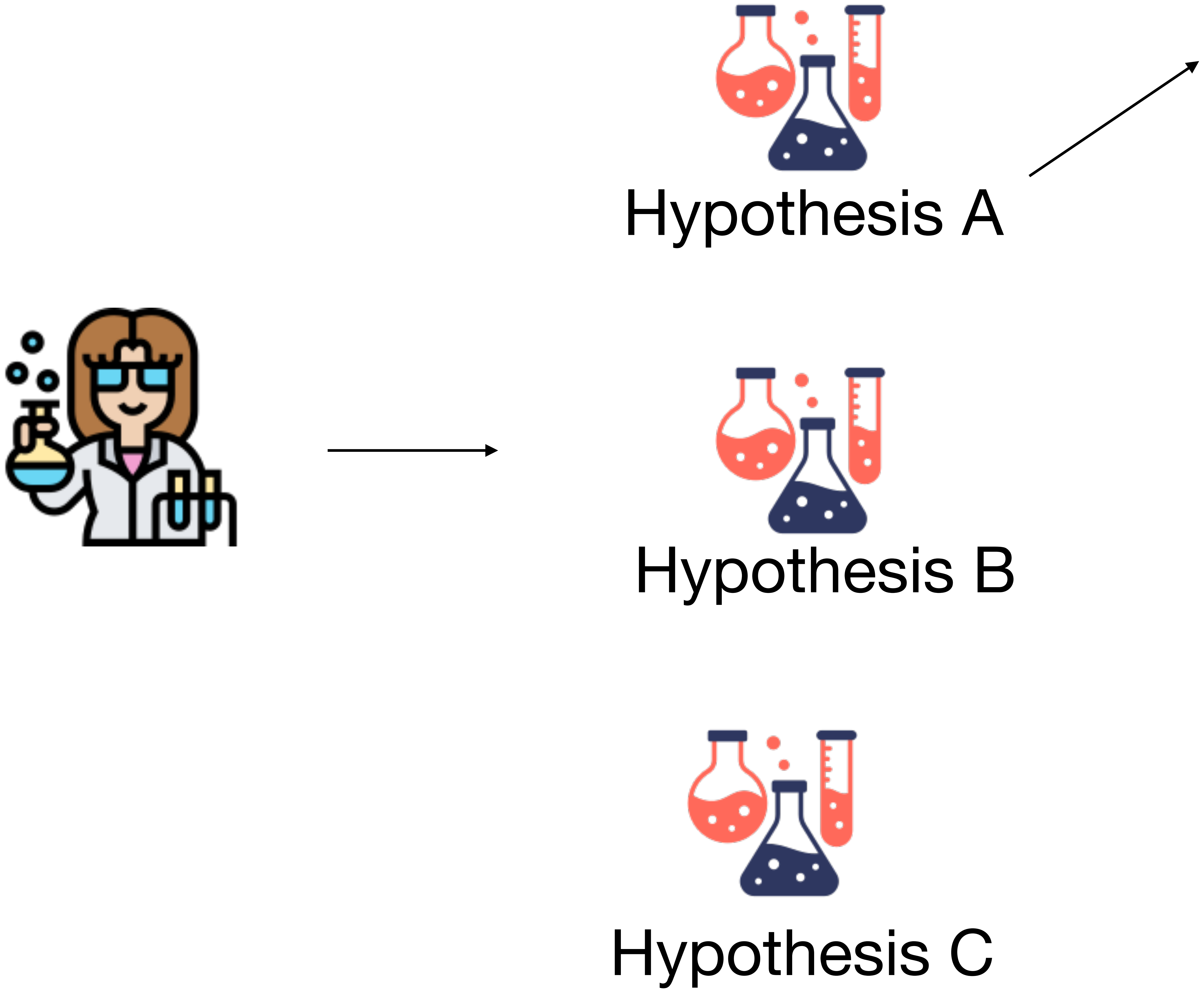
helpfulness?

user engagement?

'copilot needs to evaluate effectiveness of policy'

Think of multiple Policy Evaluation

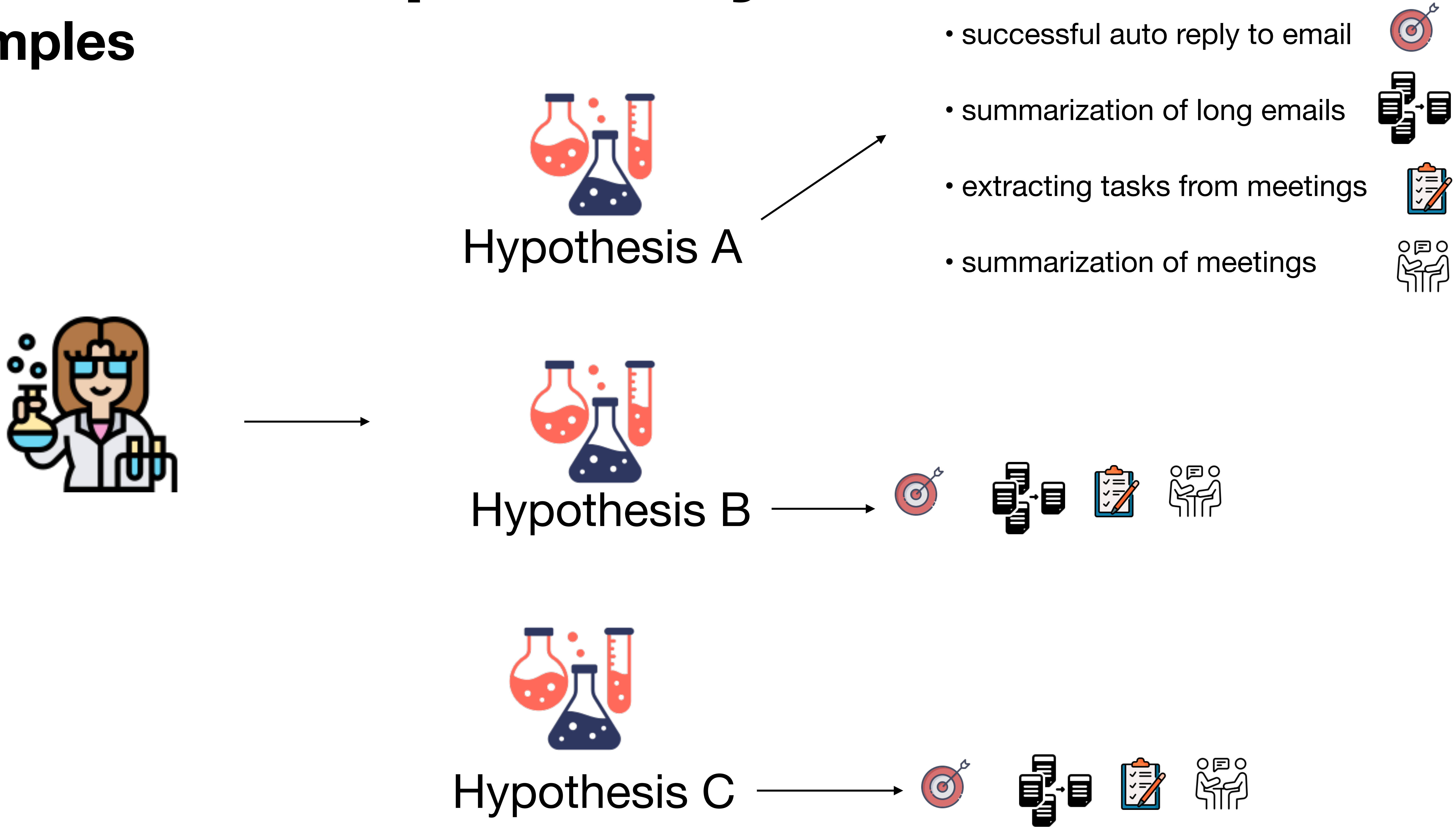
Examples



- successful auto reply to email
- summarization of long emails
- extracting tasks from meetings
- summarization of meetings

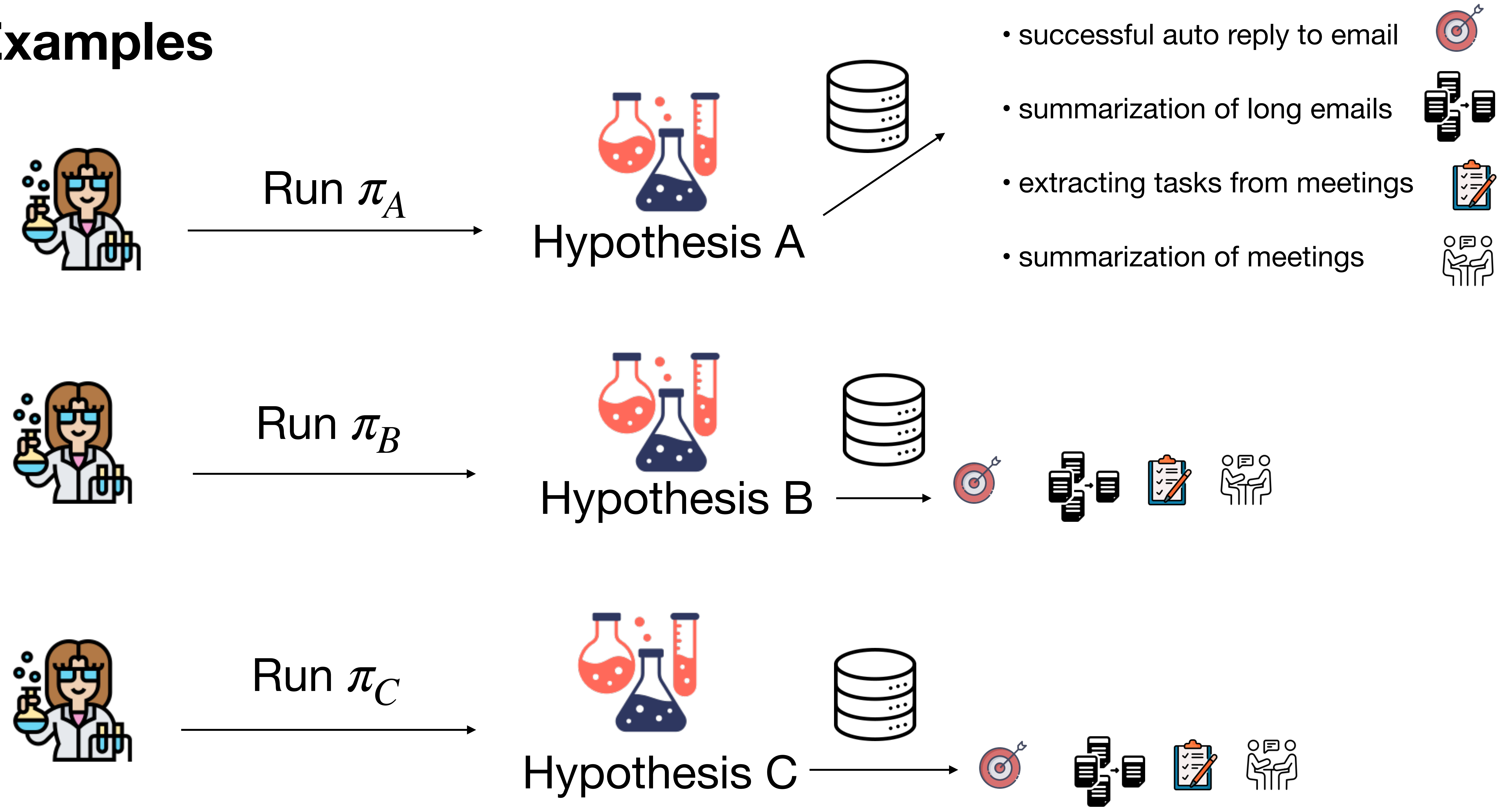
Think of multiple Policy Evaluation

Examples



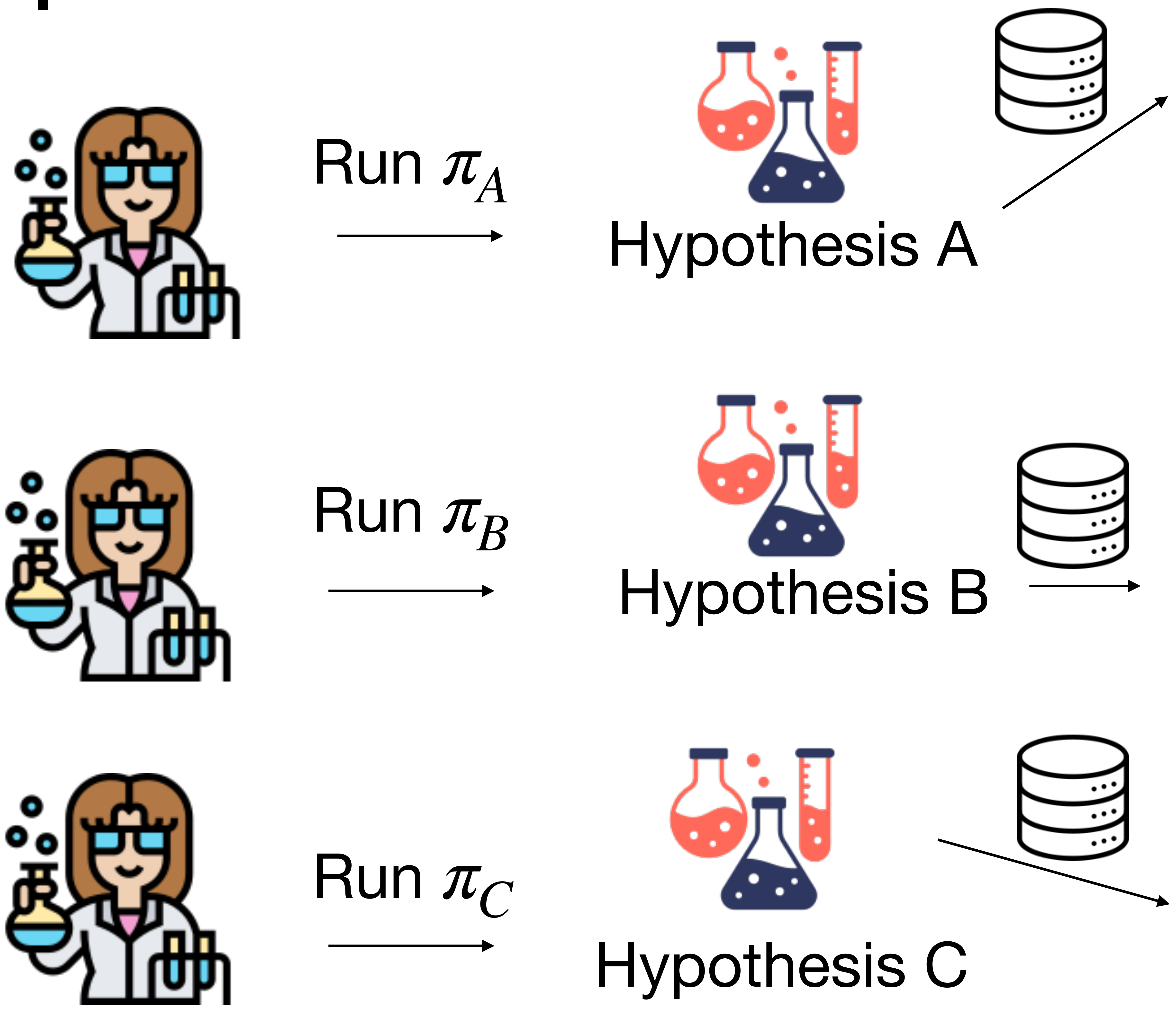
How to do multiple Policy Evaluation?

Examples



Problem: How to do multiple Policy Evaluation?

Examples



- expensive data collection
- safe/unsafe to run A/B testing
- data collection can be slow
- time consuming

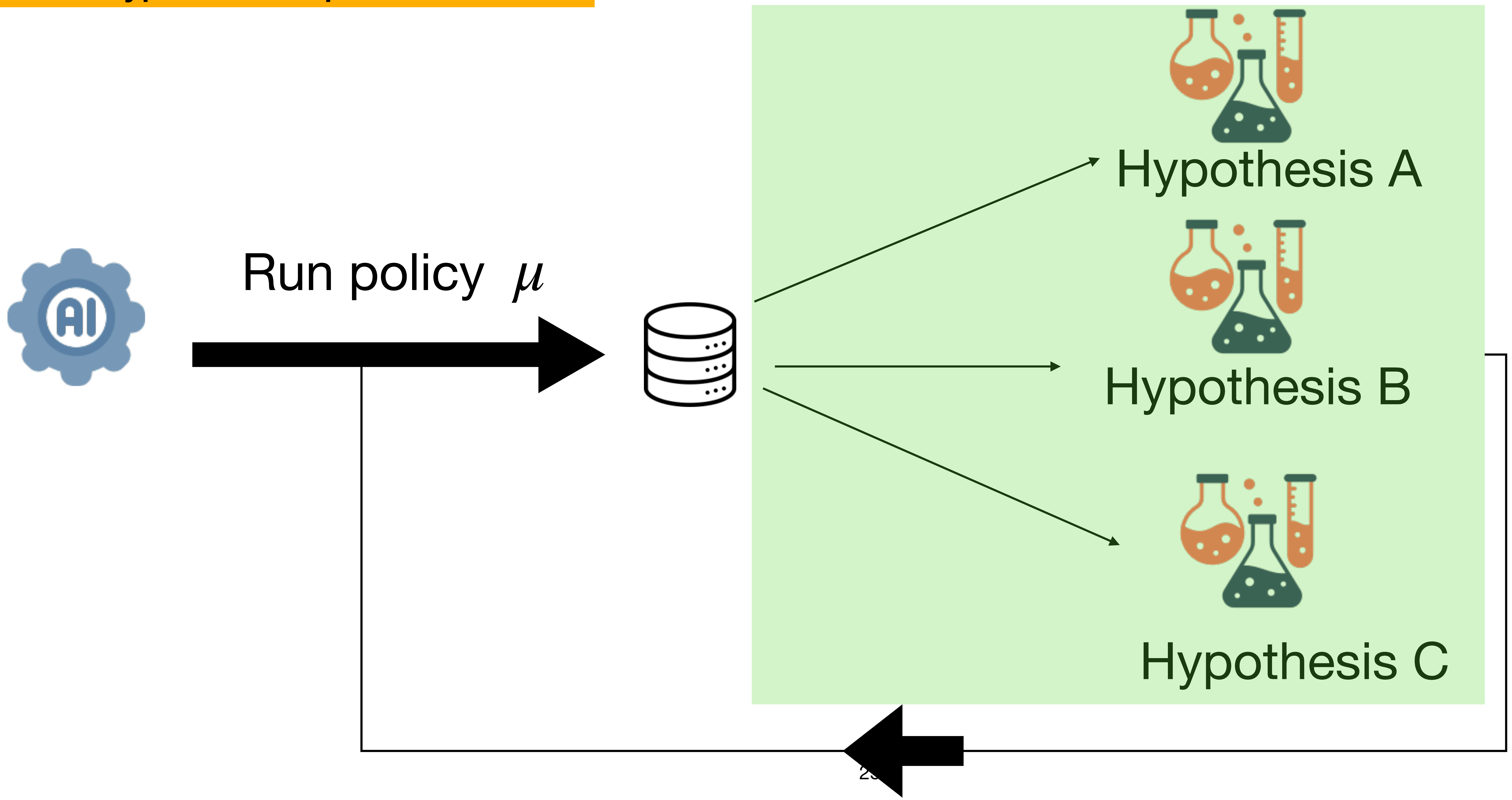
How to evaluate many strategies efficiently and know where we need more data?

Our Solution

Adaptively learn single sampling policy, which collects data to evaluate multiple hypothesis in parallel.

Off Policy Evaluation

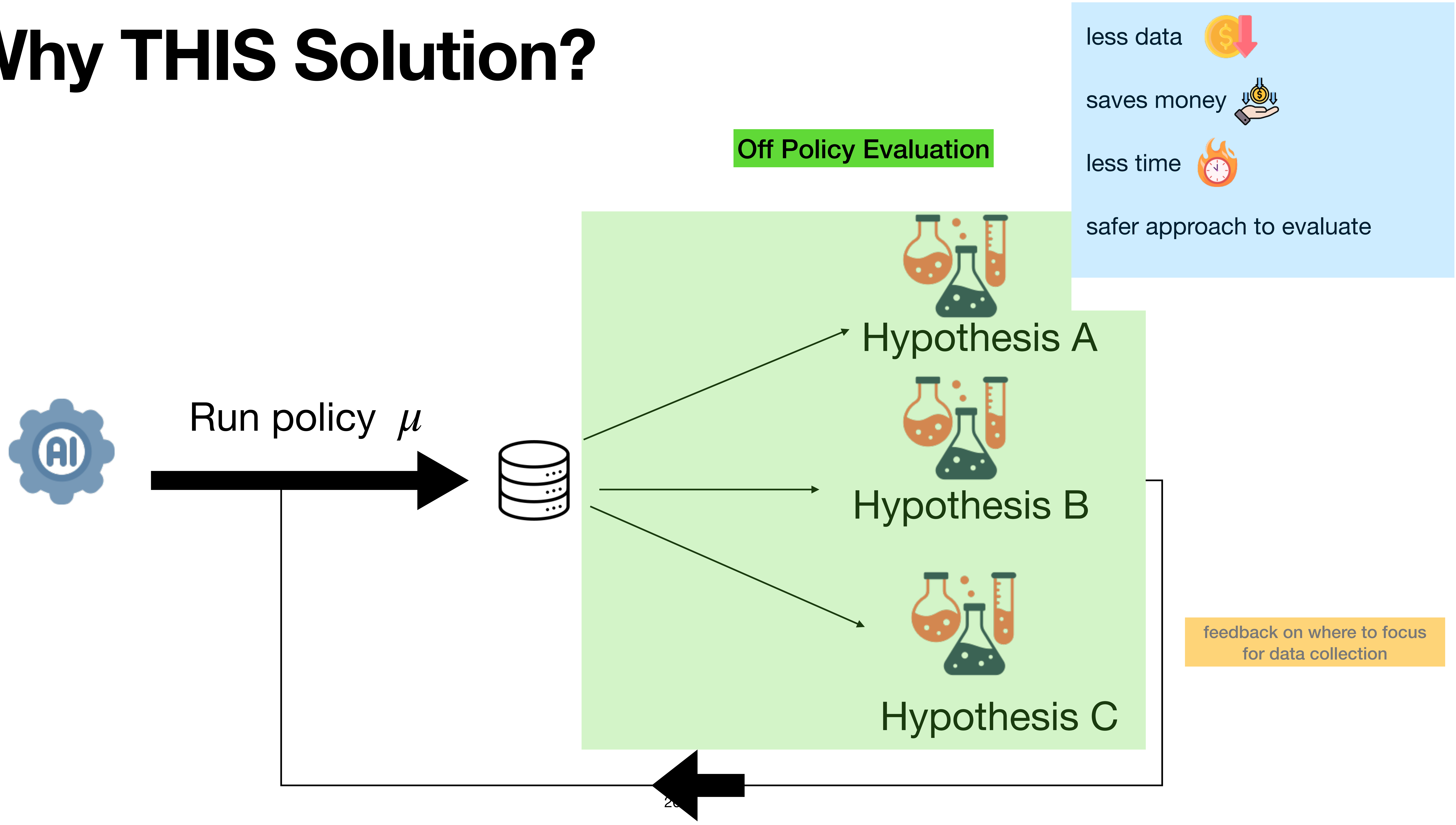
estimate how good policy A is by running another policy B



?

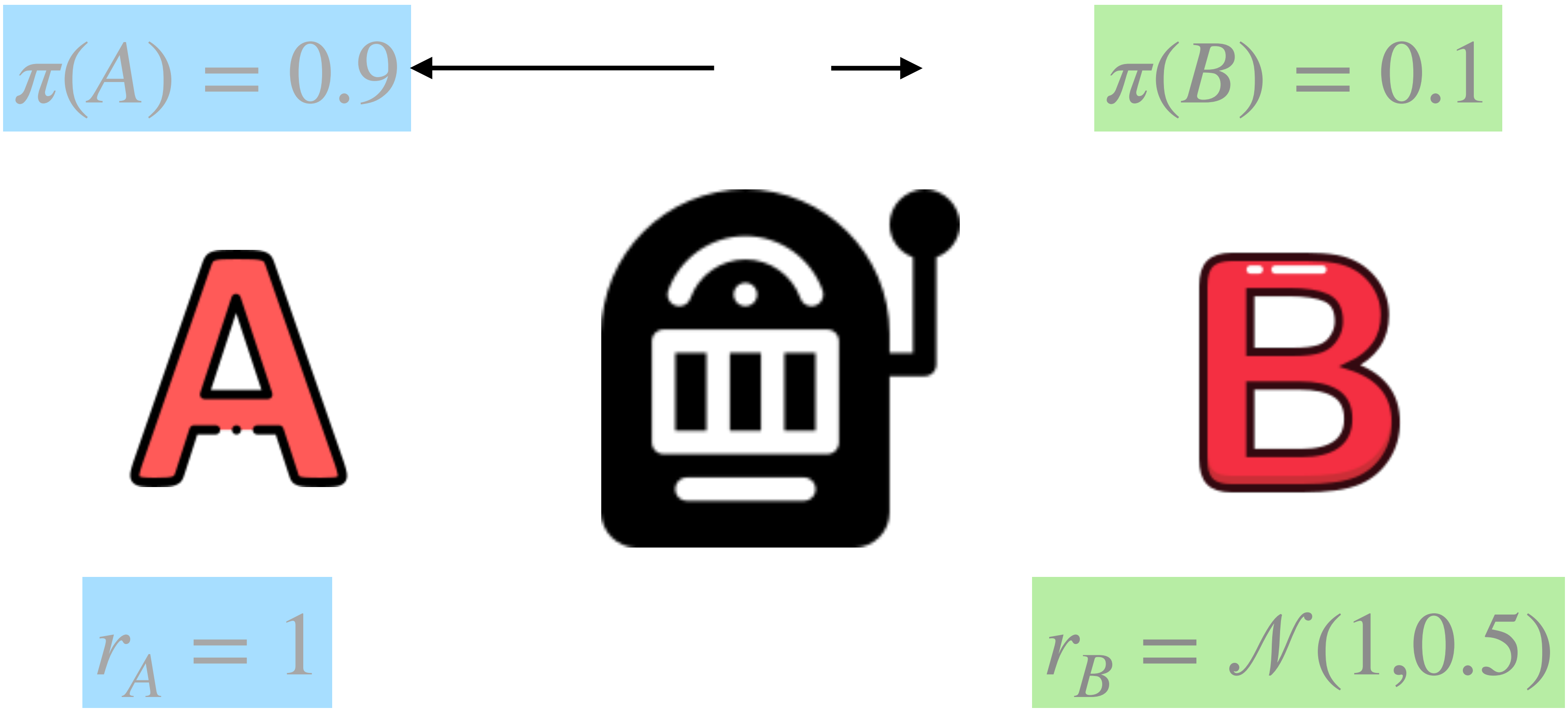
feedback on where to focus for data collection

Why THIS Solution?



What is the Feedback for self-loop?

Feedback = Variance

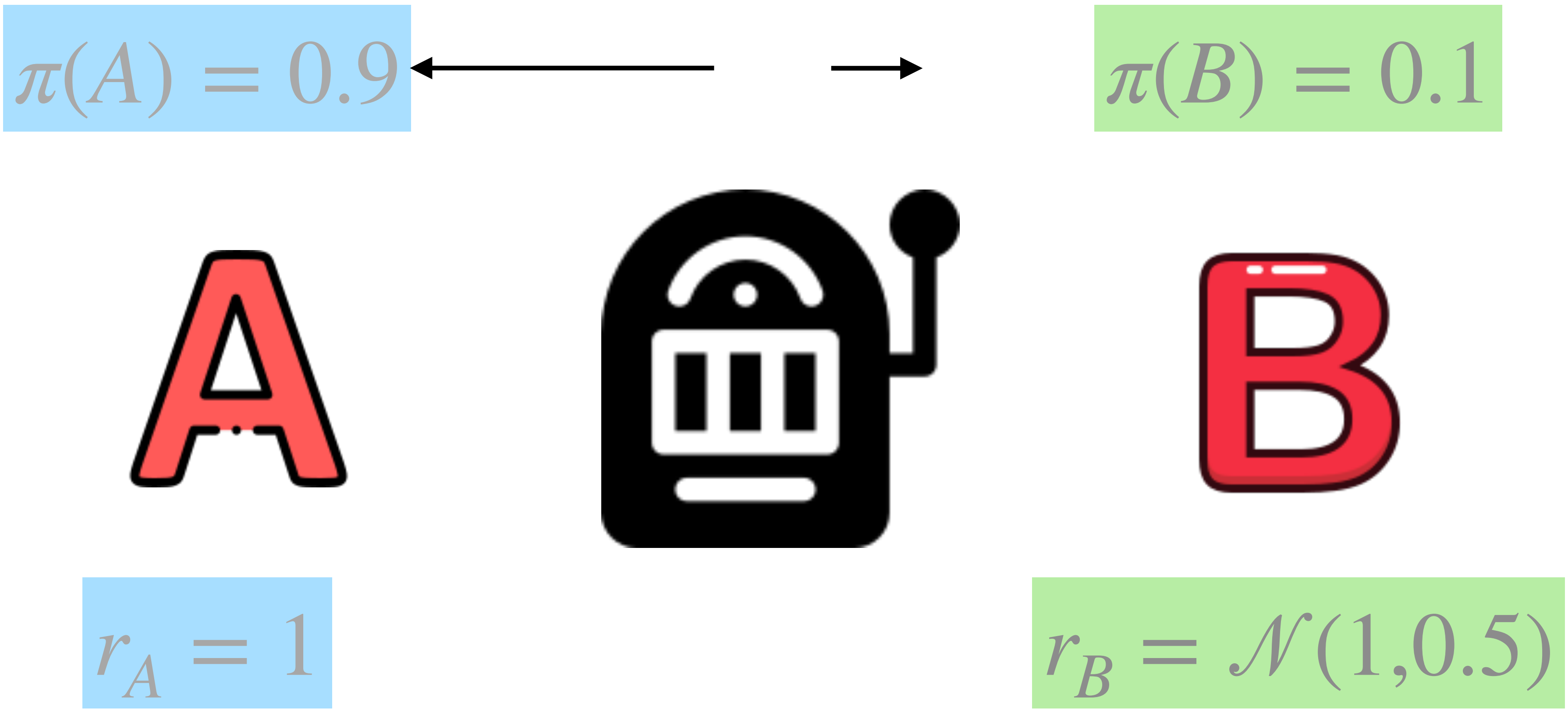


What is the **Feedback** for self-loop?

Feedback = Variance

Goal: how good is this policy?

$\mathbb{E}_{a \sim \pi}[r]$



Monte-Carlo estimation would need more data for correct estimation of expected reward

What is the Feedback for self-loop?

Feedback = Variance

Goal: how good is this policy?

$\mathbb{E}_{a\sim\pi}[r]$

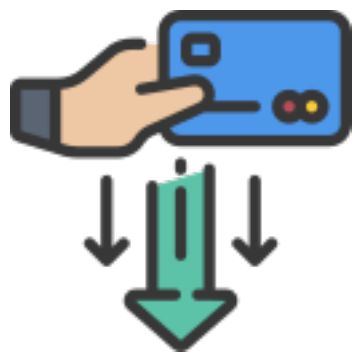
$\pi(A) = 0.9$ $\pi(B) = 0.1$

A  B

$r_A = 1$ $r_B = \mathcal{N}(1,0.5)$

Monte-Carlo estimation would need more data for correct estimation of expected reward

Arm A is stable Arm B is noisy



Sampling data

What is the **Feedback** for self-loop?

Feedback = Variance

Goal: how good is this policy?

$\mathbb{E}_{a \sim \pi}[r]$

$\pi(A) = 0.9$

$\pi(B) = 0.1$

A

B

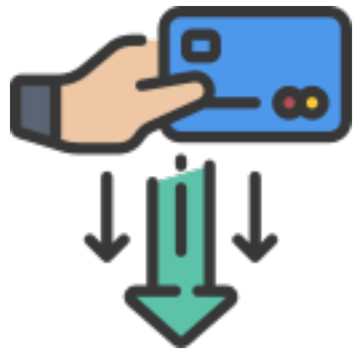
$r_A = 1$

$r_B = \mathcal{N}(1,0.5)$

Monte-Carlo estimation would need more data for correct estimation of expected reward

Low Variance

Arm A is stable



Sampling data

Arm B is noisy

High Variance

Mathematical Formulation

Policy Evaluation of General Value Functions (GVFs)

Policy	π_1	π_2	π_3
Rewards	r_1	r_2	r_3
Evaluate expected total reward under policy	$\mathbb{E}_s[r_1 \mid a \sim \pi_1]$	$\mathbb{E}_s[r_2 \mid a \sim \pi_2]$	$\mathbb{E}_s[r_3 \mid a \sim \pi_3]$

take action a following policy π_i

state

Mathematical Formulation

Policy Evaluation of General Value Functions (GVFs)

Policy	π_1	π_2	π_3
Rewards	r_1	r_2	r_3
Evaluate expected total reward under policy	$\mathbb{E}_s[r_1 \mid a \sim \pi_1]$	$\mathbb{E}_s[r_2 \mid a \sim \pi_2]$	$\mathbb{E}_s[r_3 \mid a \sim \pi_3]$

any observable signal

GVF: Pair (π_i, c_i)

multi-step prediction : $\mathbb{E}[G_i \mid a \sim \pi_i]$

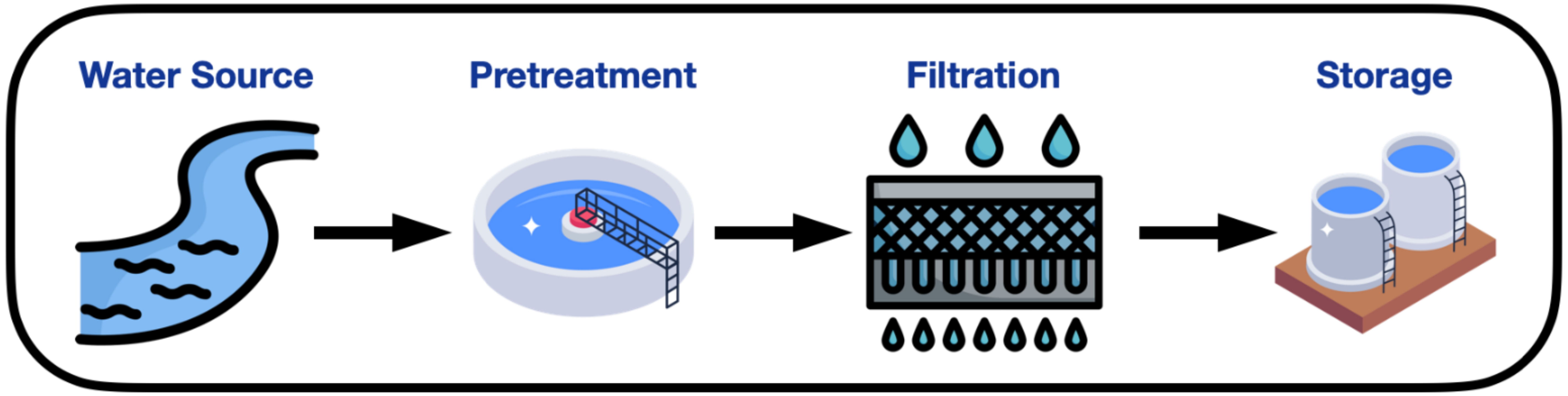
$$G_i = \sum_{t=0}^{\infty} \gamma^t c_t^{[i]}$$

take action a following policy π_i

state

Real World GVF Example

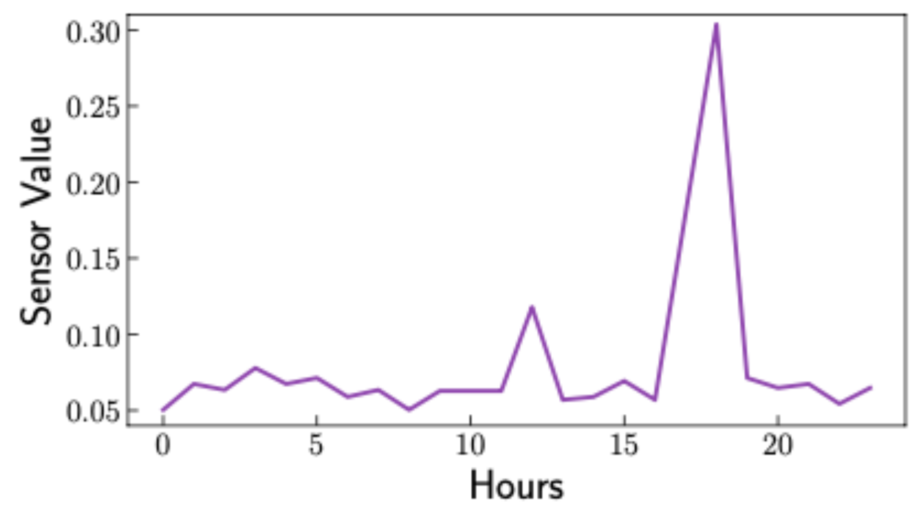
Water treatment Plant (Janjua et al. 2024)



c_1

c_2

Sensor Type
Pressure
Flowmeter
pH
Temperature
Turbidity
Total Organic Carbon (TOC)
Conductivity



Predict the future values of signal c_i

Mathematical Formulation

Find sampling policy μ which minimizes sum of variance in rewards under different policy π_i

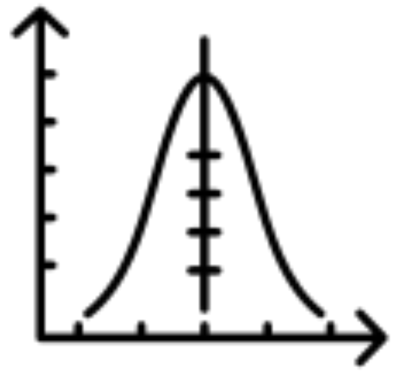
feedback in self-improving loop

$$\mu^* = \arg\min_{\mu} \sum_{i=1}^{n} \text{variance}_{a \sim \mu}(G_i, \pi_i)$$

sampling policy

Mathematical Formulation

sum of future rewards: $G_i = \sum_{t=0}^{\infty} \gamma^t r_t^{[i]}$



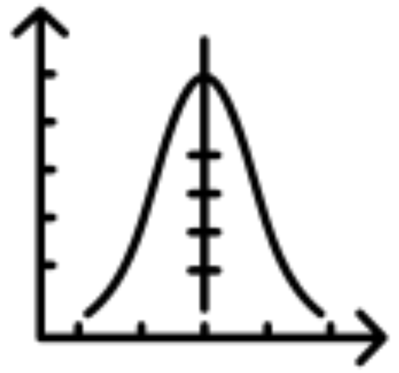
$$\mu^* = \arg\min_{\mu} \sum_{i=1}^{n} \text{variance}_{a \sim \mu}(G_i, \pi_i)$$

sampling policy

policy π_i

Mathematical Formulation

sum of future rewards: $G_i = \sum_{t=0}^{\infty} \gamma^t r_t^{[i]}$



$$\mu^* = \arg\min_{\mu} \sum_{i=1}^{n} \text{variance}_{a\sim\mu}(G_i, \pi_i)$$

policy π_i

sampling policy

Off-Policy Evaluation: run policy μ, but estimate evaluate performance of policy π_i

Why minimize variance?

Minimize MSE(true perf., estimate perf)

= Bias^2 + Variance

Why minimize variance?

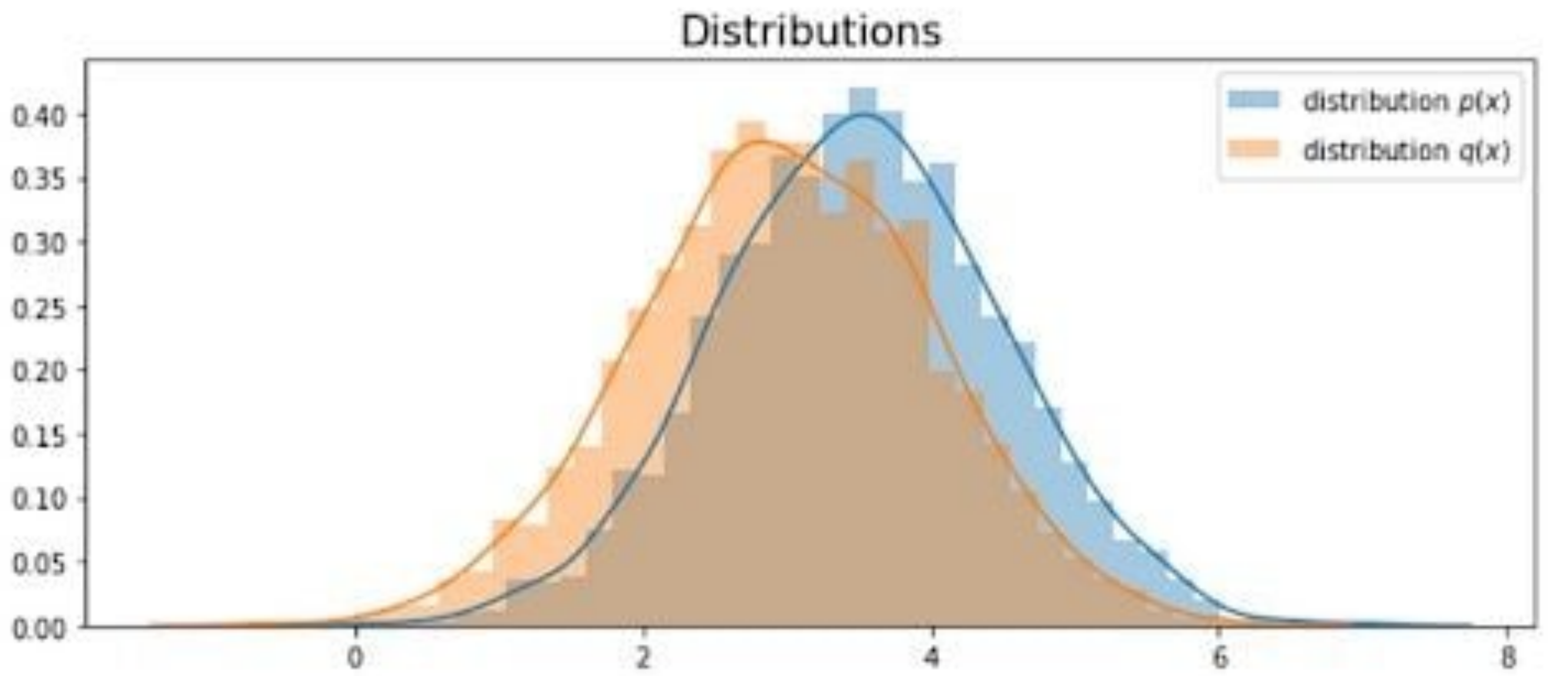
Minimize MSE(true perf., estimate perf)

$= \cancel{\text{Bias}^2} + \text{Variance}$ = minimize Variance

We will use importance sampling for off-policy correction = unbiased estimates

Importance Sampling

IS = unbiased estimator



$$\mathbb{E}[x \mid x \sim p] = \frac{1}{n}\sum_i p(x_i)x_i$$

$$= \frac{1}{n}\sum_i q(x_i) \times \frac{p(x_i)}{q(x_i)} x_i$$

$$= \mathbb{E}\left[\frac{p(x)}{q(x)} x \mid x \sim q\right]$$

importance sampling (IS) correction

Sampling from q but estimating expectation under p

Variance Estimator

Jain et al., AAAI 2021

variance(G) = $\mathbb{E}[(G -$ exptected perf G$)^2]$

$$\mu^* = \arg\min_{\mu} \sum_{i=1}^{n} \text{variance}_{a\sim\mu}(G_i, \pi_i)$$

sum of future reward

We use TD estimator of variance
to build approximation of this measure

Variance Estimator

Jain et al., AAAI 2021

variance(G) = $\mathbb{E}[(G -$ exptected perf G$)^2]$

$$\mu^* = \arg\min_{\mu} \sum_{i=1}^{n} \text{variance}_{a \sim \mu}(G_i, \pi_i)$$

cumulative discounted rewards

We use TD estimator of variance to build approximation of this measure

Why TD estimator?

- Faster updates
- easy to scale in complex environments

Contributions

iterately update

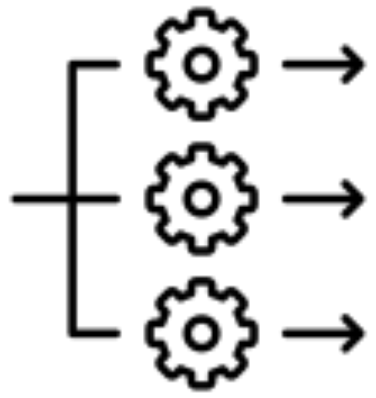
sampling policy

decrease in

prediction error

of performance

sample efficient



parallel task evaluations

Adaptive Sampling Policy

target policy: "*how good I am?*"

$$\mu^* = \arg\min_{\mu} \sum_{i=1}^{n} \text{variance}_{a \sim \mu}(G_i, \pi_i)$$

→

$$\mu_{k+1} = \frac{\sqrt{\sum_{i=0}^{N} \pi_i(a \mid s)^2 var_i^{\mu_k}(s, a)}}{\sum_{a'} \sqrt{\sum_{i=0}^{N} \pi_i(a' \mid s)^2 var_i^{\mu_k}(s, a')}}$$

iterate

sampling policy

normalization factor

Adaptive Sampling Policy

target policy: "*how good I am?*"

sampling policy

normalization factor

$$\mu^* = \arg\min_{\mu} \sum_{i=1}^{n} \text{variance}_{a\sim\mu}(G_i, \pi_i) \longrightarrow \mu_{k+1} = \frac{\sqrt{\sum_{i=0}^{N} \pi_i(a\,|\,s)^2 var_i^{\mu_k}(s,a)}}{\sum_{a'} \sqrt{\sum_{i=0}^{N} \pi_i(a'\,|\,s)^2 var_i^{\mu_k}(s,a')}}$$

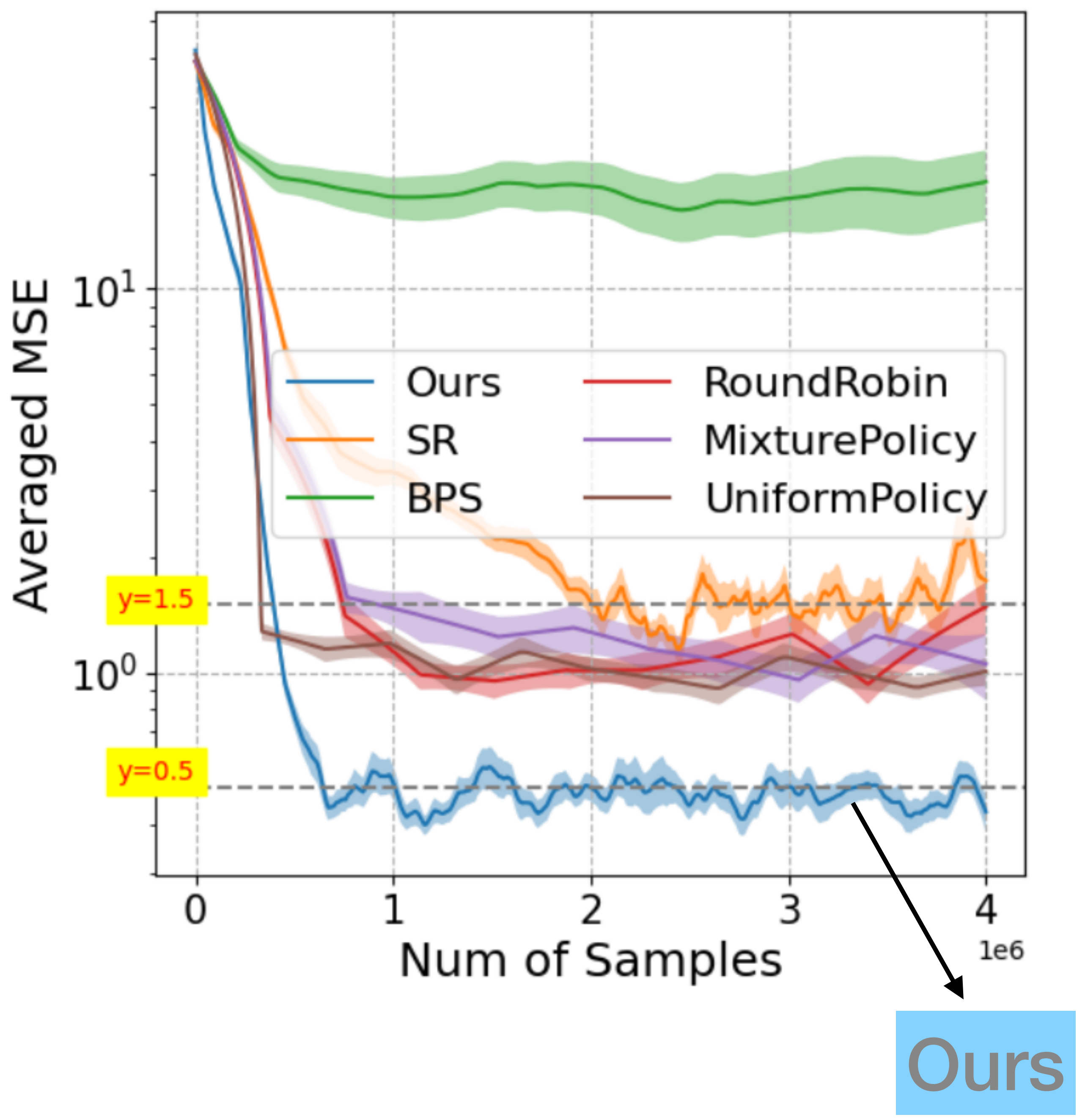
iterate

$$\text{sampling policy}_{k+1} \propto \text{target policy} \times \sqrt{\text{var(cumulative rewards)}}$$

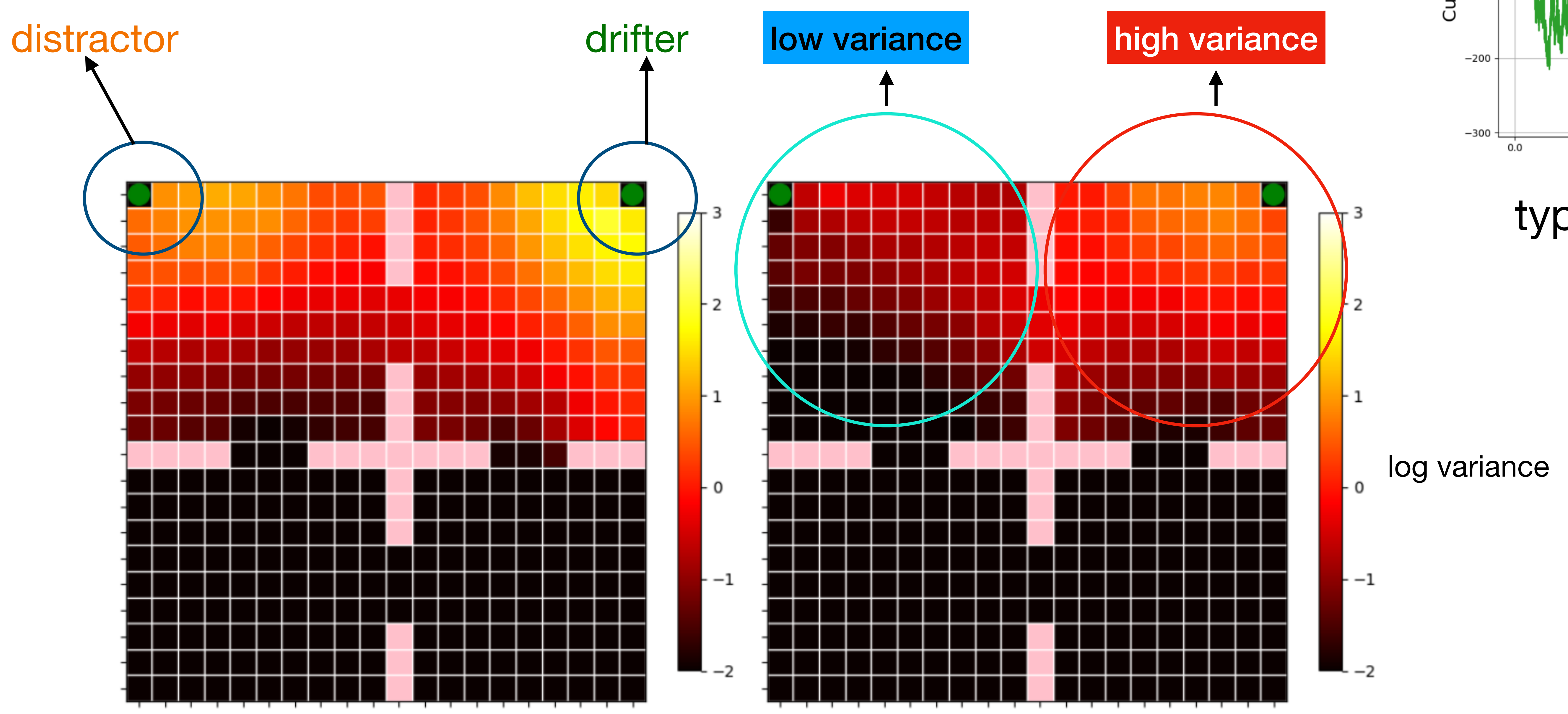
more data | high uncertainty

less data | low uncertainty

Empirical Results



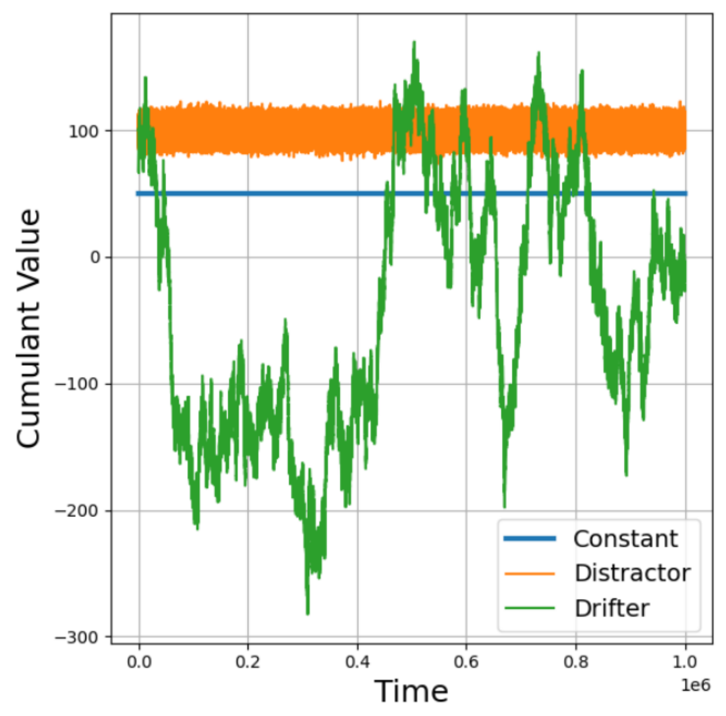
Lower MSE of performance is better



variance in early learning

variance in later learning

Our approach effectiveness in tracking non-stationary reward signal

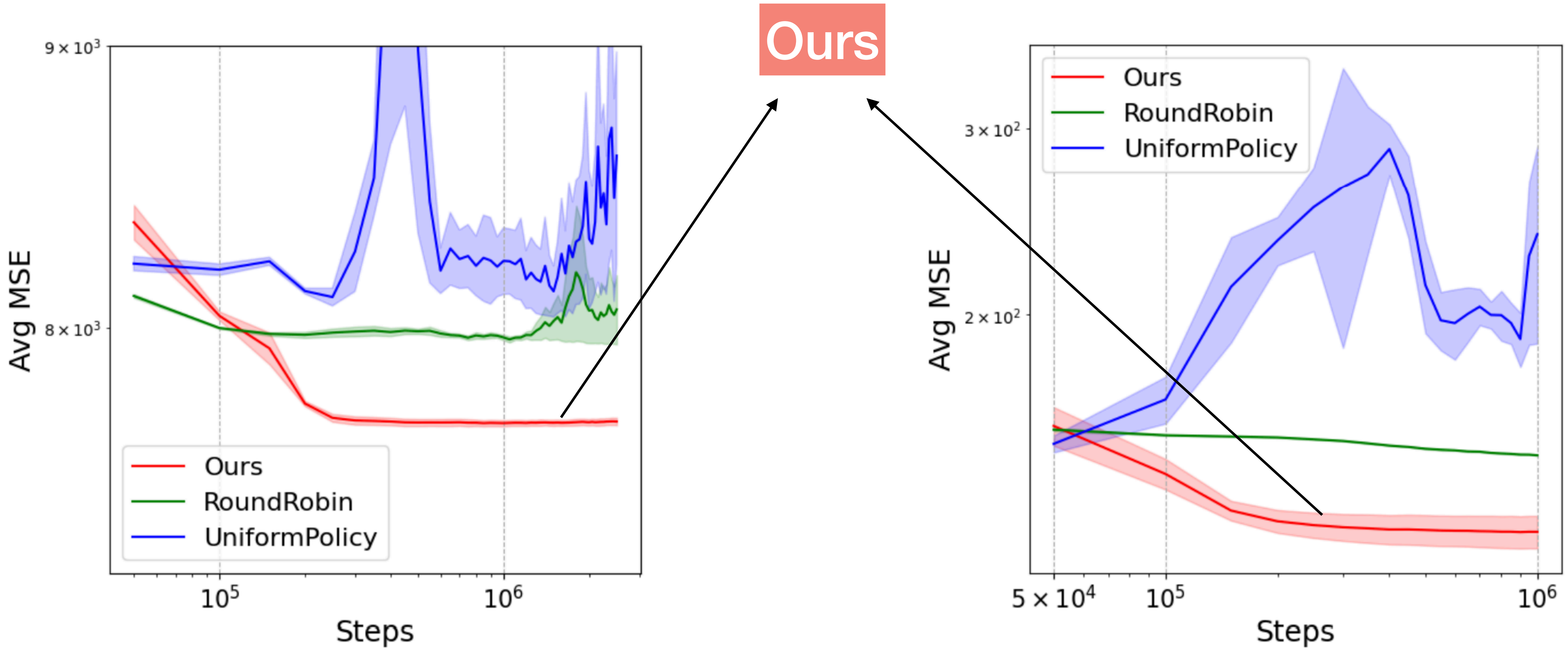


type of rewards

Mujoco Environments

Continuous states and actions

lower MSE
is better



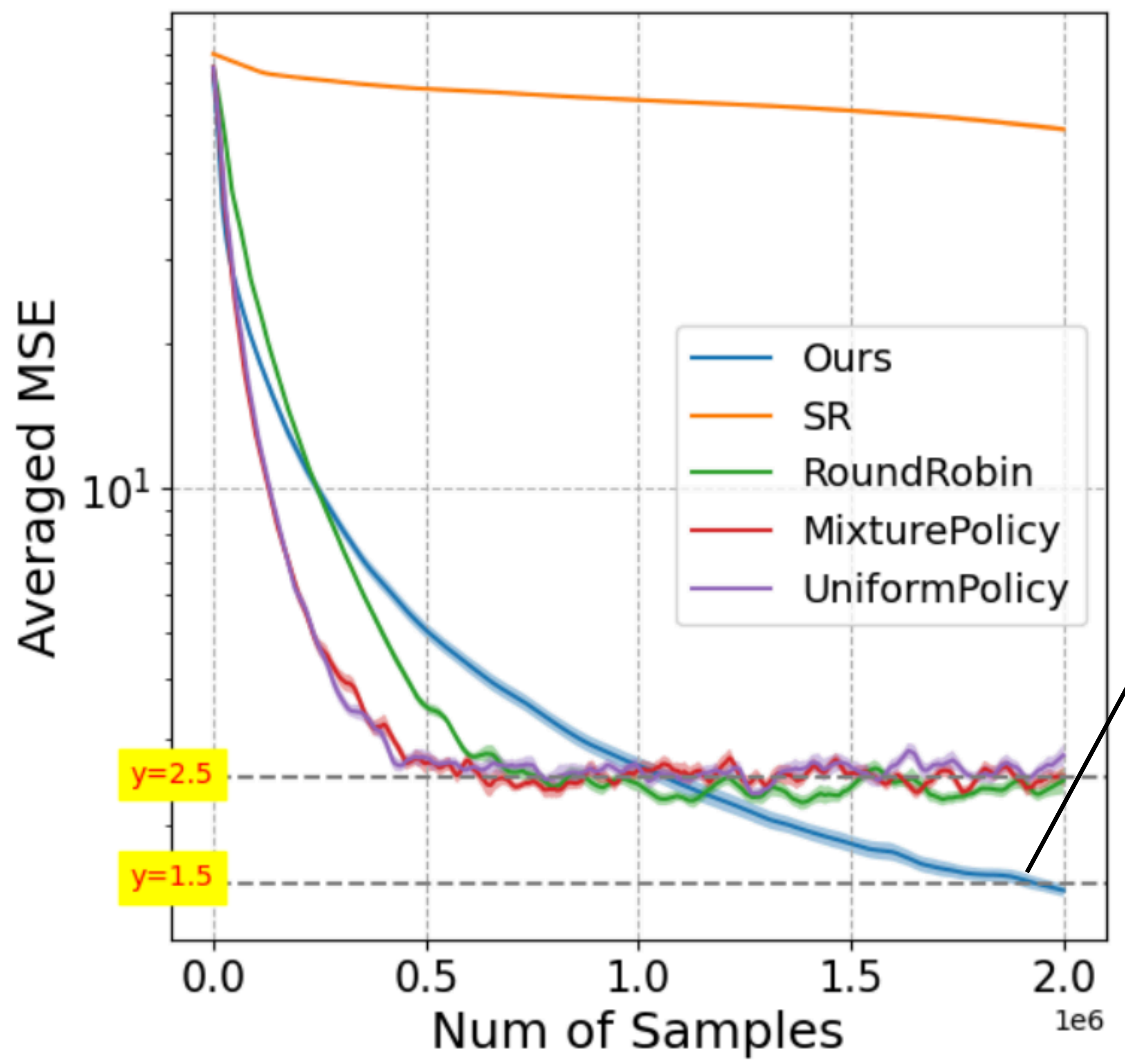
(a) Averaged MSE in Walker

(b) Averaged MSE in Cheetah

Two tasks: Walk, flip

Two tasks: Walk, run

Higher number of parallel tasks



Ours

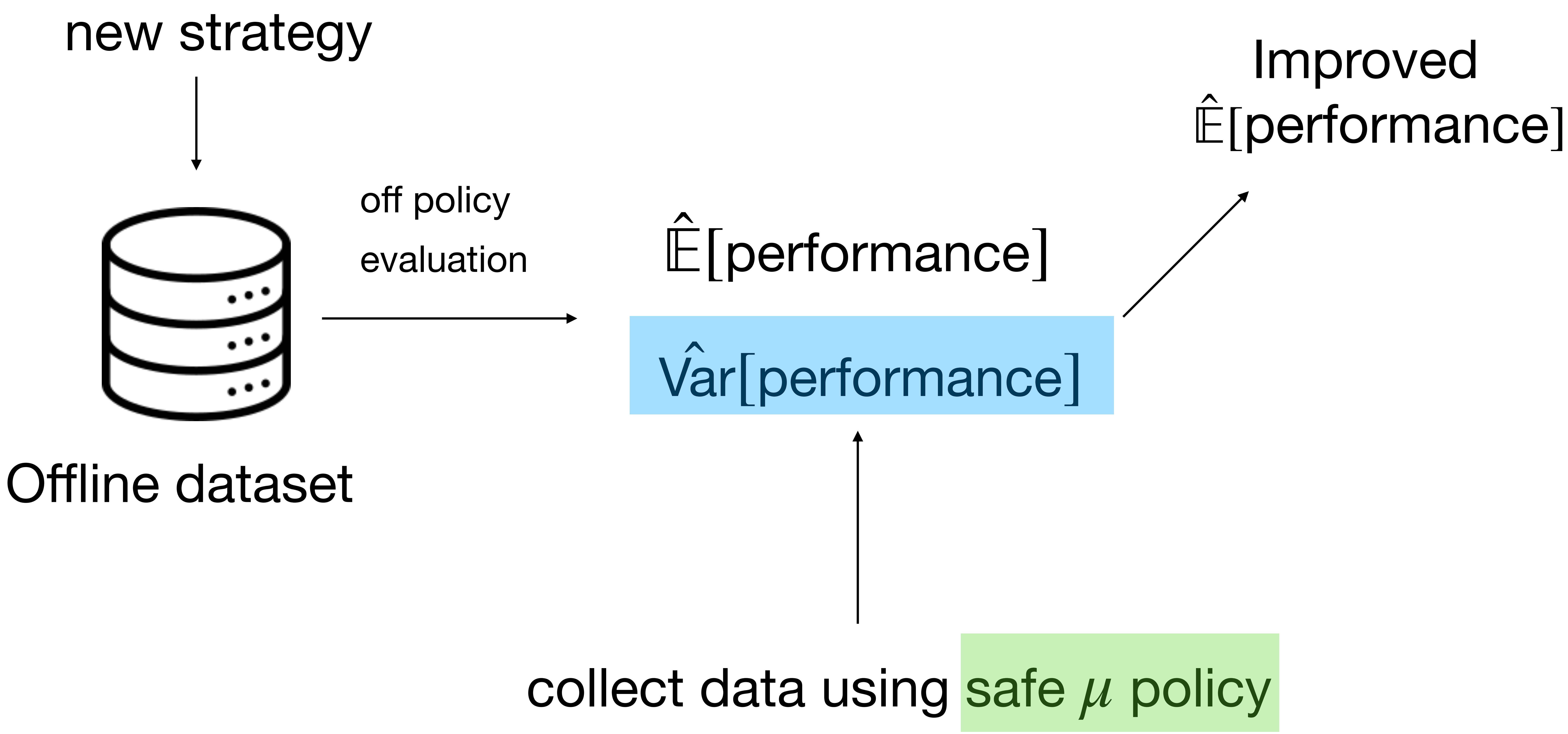
Faster MSE reduction

with less number of samples

Average MSE of 40 number of tasks

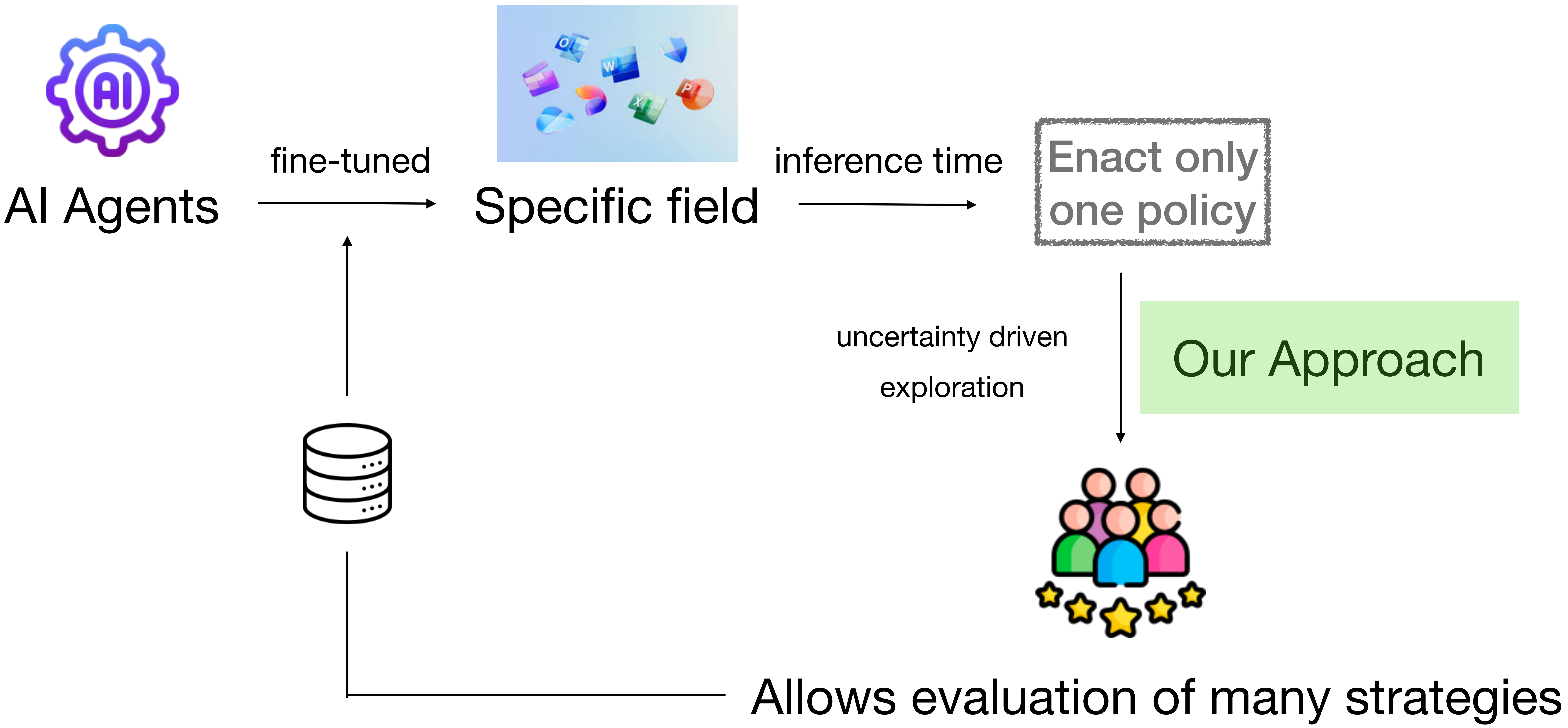
Other useful scenarios

Improving Offline Evaluations



Future Motivation

Evaluating many strategies from one acting policy



Thanks You!

arushi.jain@mail.mcgill.ca

Additional Slide

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

expected performance

$$Var(s) = \mathbb{E}[\delta_t^2 + \gamma^2 Var(s_{t+1}) \,|\, s = s_t]$$

variance in performance

$$\delta_t = r_t + \gamma V(s_{t+1}) - V(s_t)$$