Exploration

CMPT 729 G100

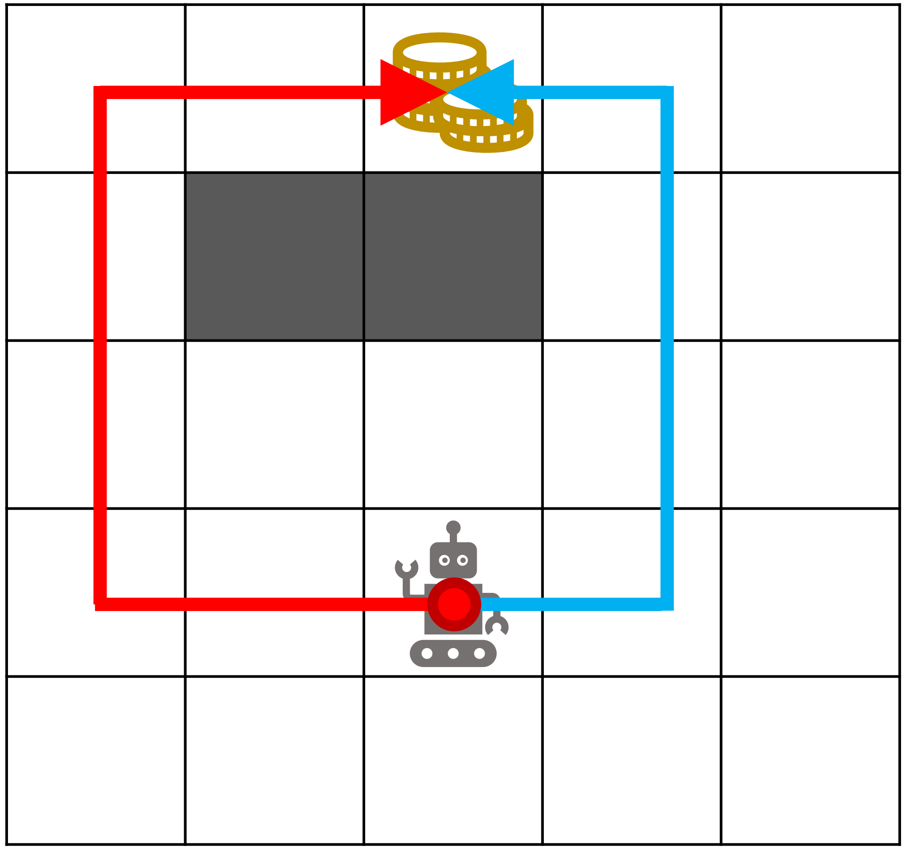
Jason Peng

Overview

- Exploration Exploitation Tradeoff
- Dense vs Sparse Rewards
- Intrinsic Motivation
- Count-Based Exploration
- Surprise Maximization

Exploration-Exploitation

Need to try new actions in case they are better



Exploration-Exploitation

Keep going to the same restaurant

Try new restaurant

Oil Drilling

Drill at best known location

Drill at a new location

Ad Recommendation

Show most successful ad

Show a different ad

Simple Exploration Strategies

ϵ-greedy exploration:

$$\pi(\mathbf{a}|\mathbf{s}) = \begin{cases} 1-\epsilon & \text{if } \mathbf{a} = \arg\max_{\mathbf{a}'} \; Q^k(\mathbf{s}, \mathbf{a}') \\ \epsilon & \text{otherwise} \end{cases}$$

Boltzmann exploration:

$$\pi(\mathbf{a}|\mathbf{s}) = \frac{1}{Z}\exp\left(\frac{1}{\beta} Q^k(\mathbf{s}, \mathbf{a})\right)$$

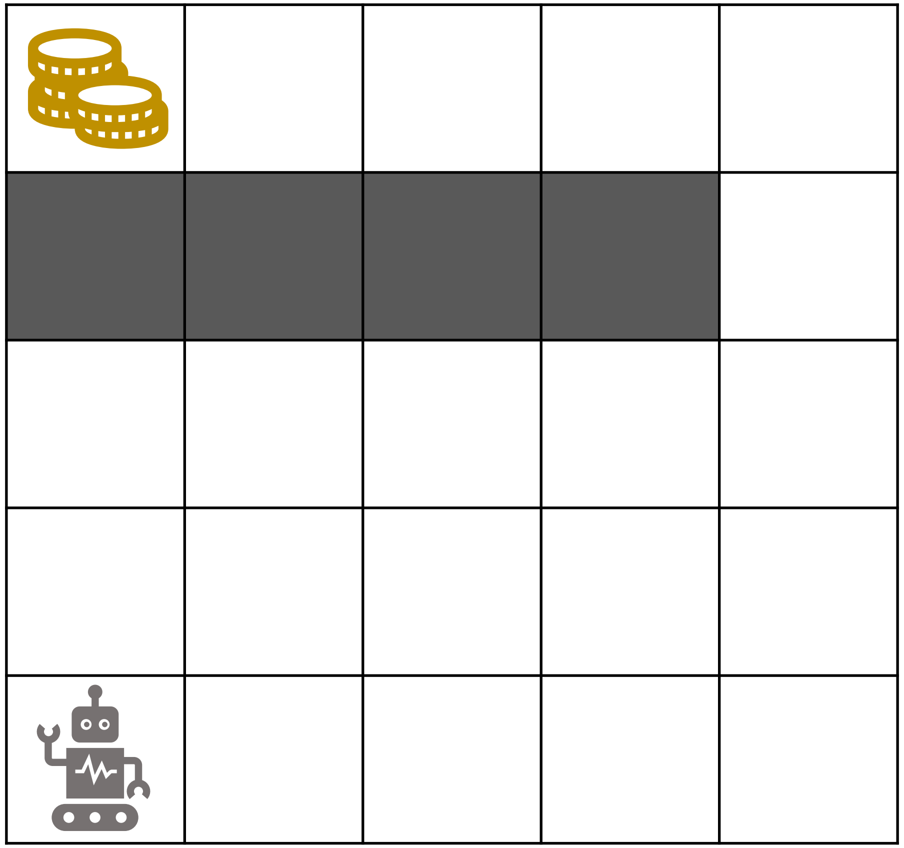
✓ Good action coverage

✗ Bad state coverage

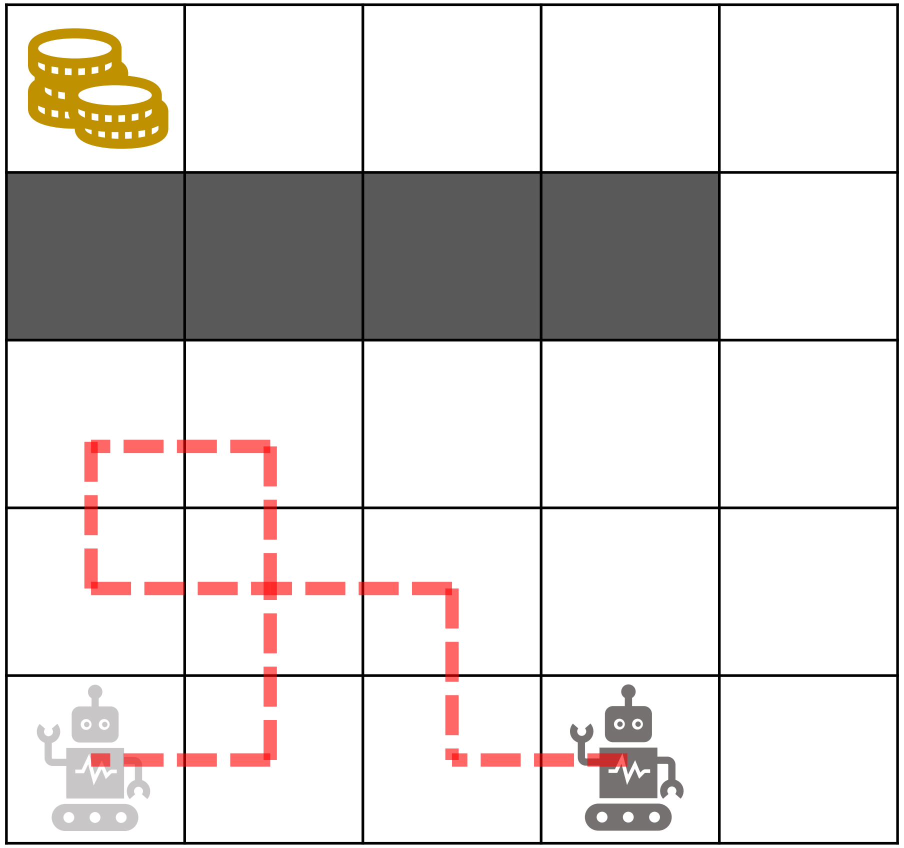
Gaussian policy:

$$\pi(\mathbf{a}|\mathbf{s}) = \frac{1}{Z}\exp\left(-\frac{1}{2}(\mathbf{a} - \mu_\pi(\mathbf{s}))^\intercal \, \Sigma(\mathbf{s})^{-1} (\mathbf{a} - \mu_\pi(\mathbf{s}))\right)$$

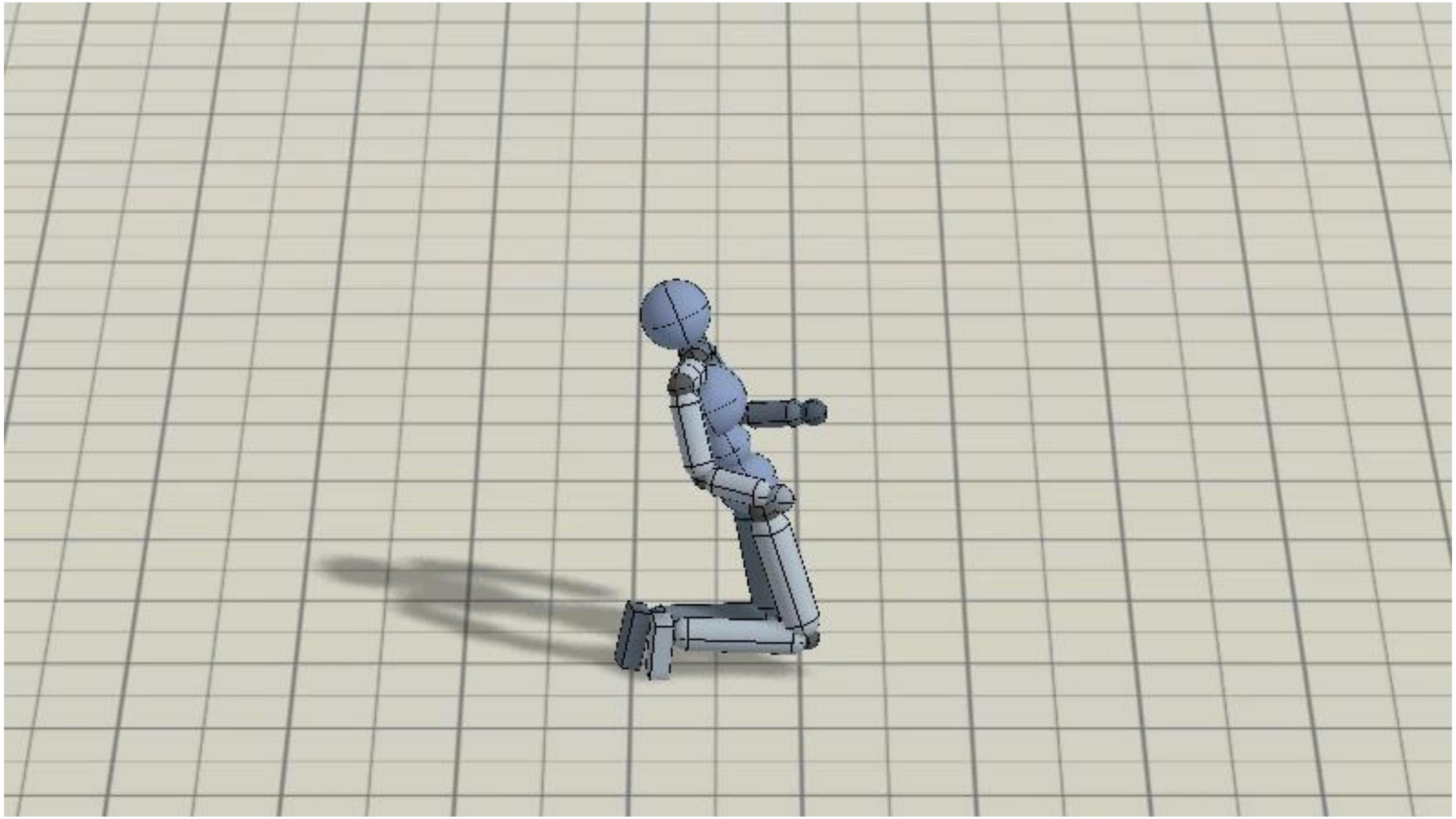
Simple Exploration Strategies



Simple Exploration Strategies



Simple Exploration Strategies



Reward Functions

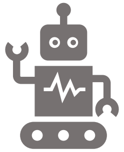
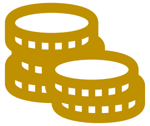
- Reward function guides policy towards better actions
- Structure of reward function can have dramatic impact on exploration and performance
  - Well-shaped reward function: hard tasks can be made easy
  - Poorly shaped reward function: easy tasks can be made almost impossible

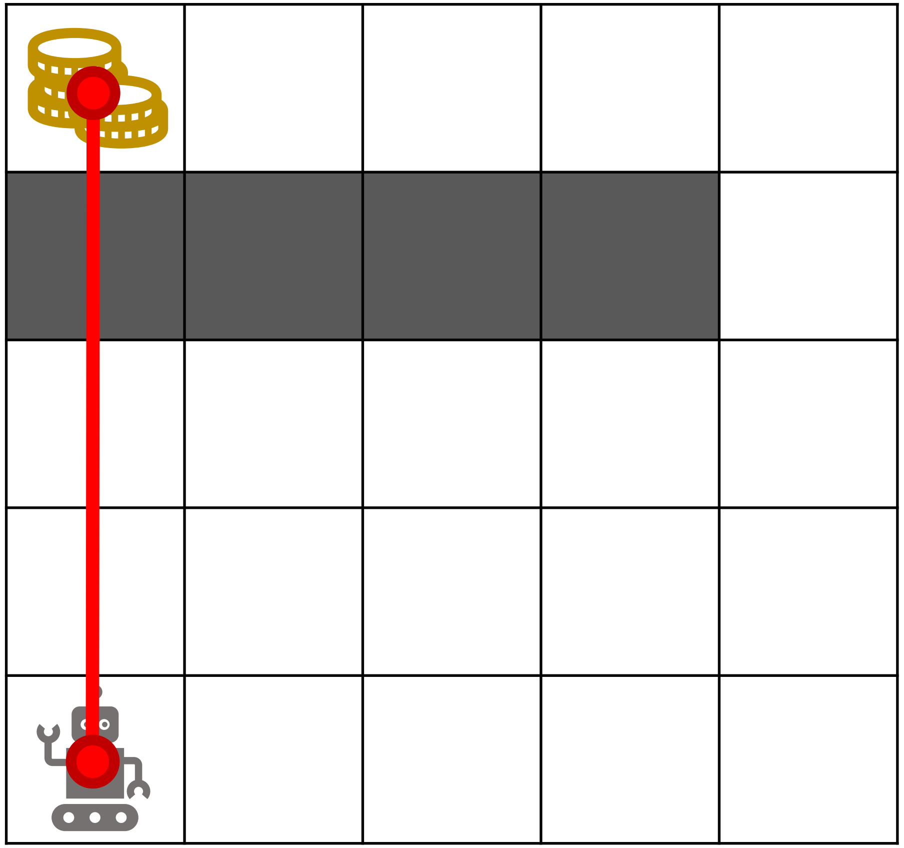
Reward Functions

- Dense reward
    - Non-zero reward at every timestep reflecting progress towards goal
- Sparse reward
    - Agent receives nonzero reward only when goal is completed

Dense Reward

Non-zero reward at every timestep reflecting progress towards goal

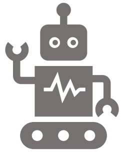
$r_t = - ||$  $-$  $||^2$

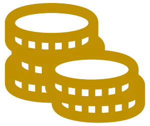


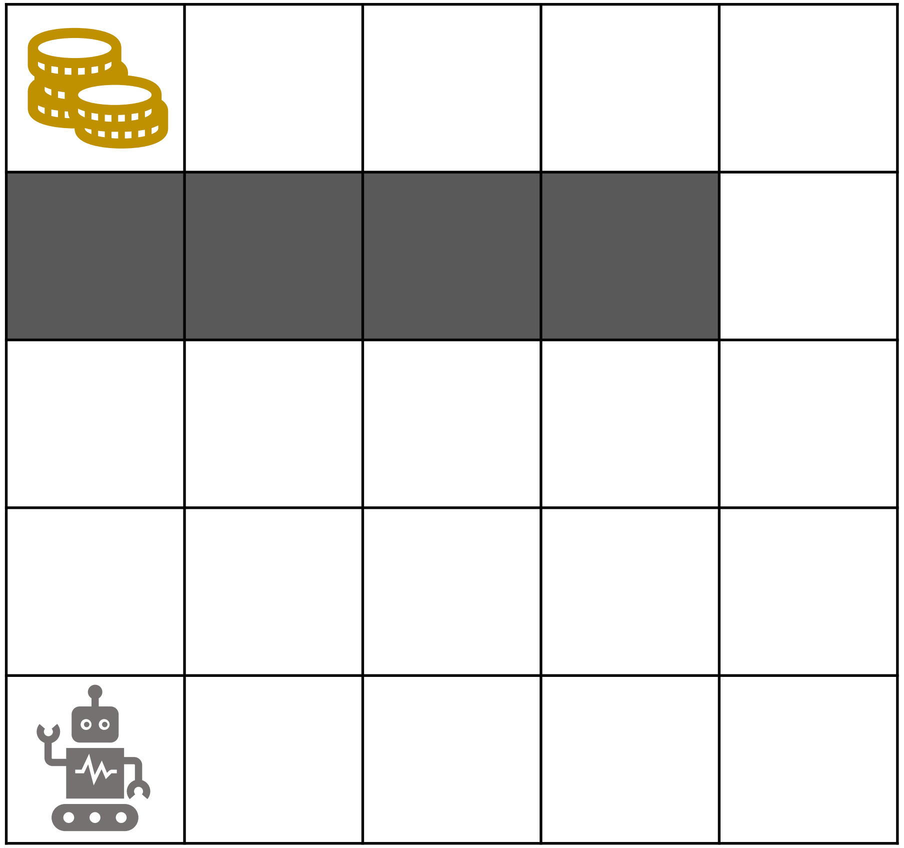
Sparse Reward

Agent receives nonzero reward only when goal is completed

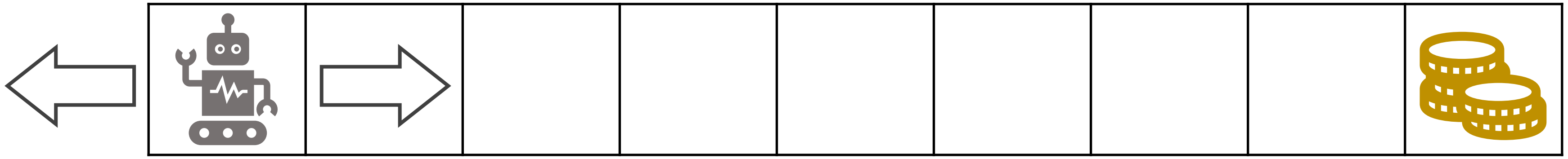
$$r_t = \begin{cases} 1 & \text{if } \quad \text{ at } \\ 0 & \text{otherwise} \end{cases}$$



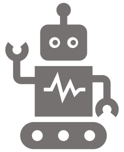
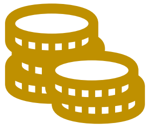


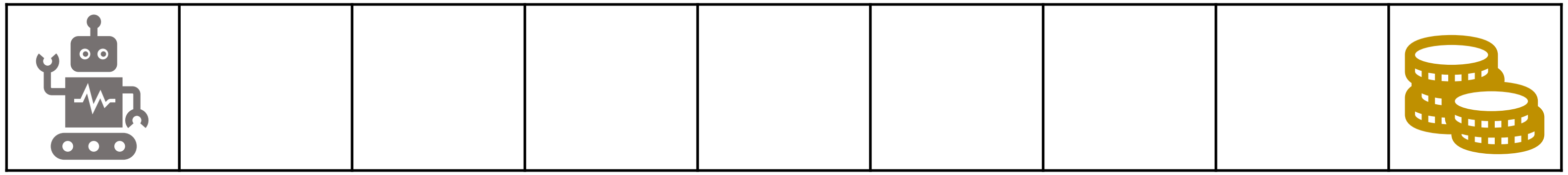


Reward Functions

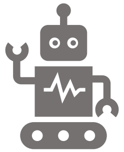
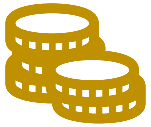


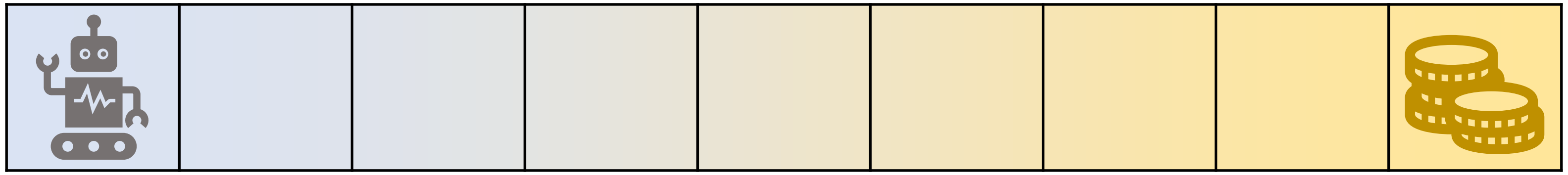
Dense Rewards

$r_t = -\,||$  $-$  $||^2$

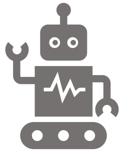
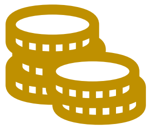


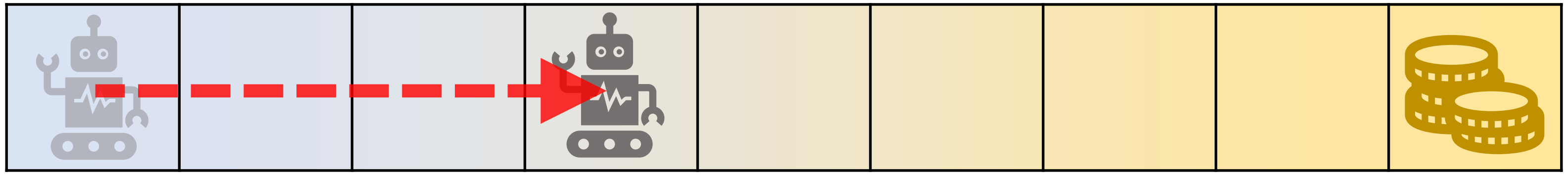
Dense Rewards

$r_t = -\,||$  $-$  $||^2$



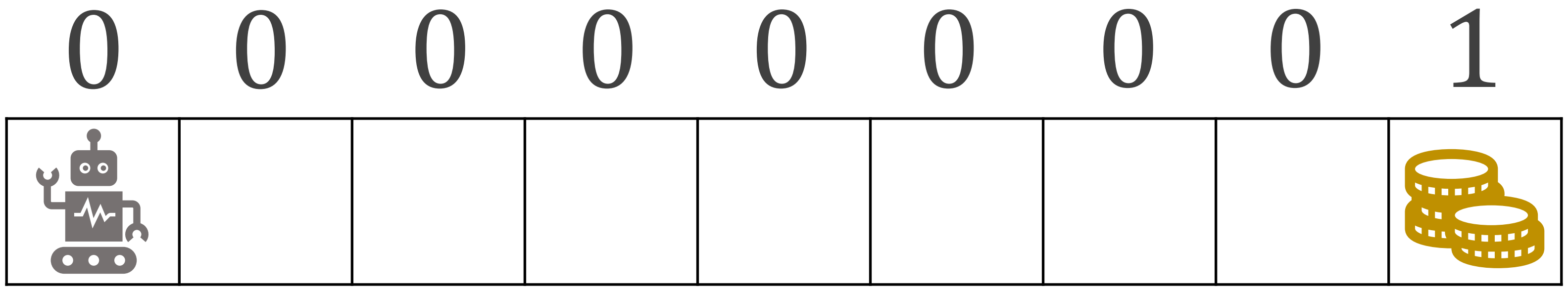
Dense Rewards

$r_t = -\,||$  $-$  $||^2$

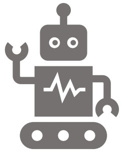
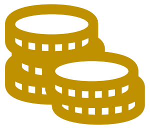


Sparse Rewards

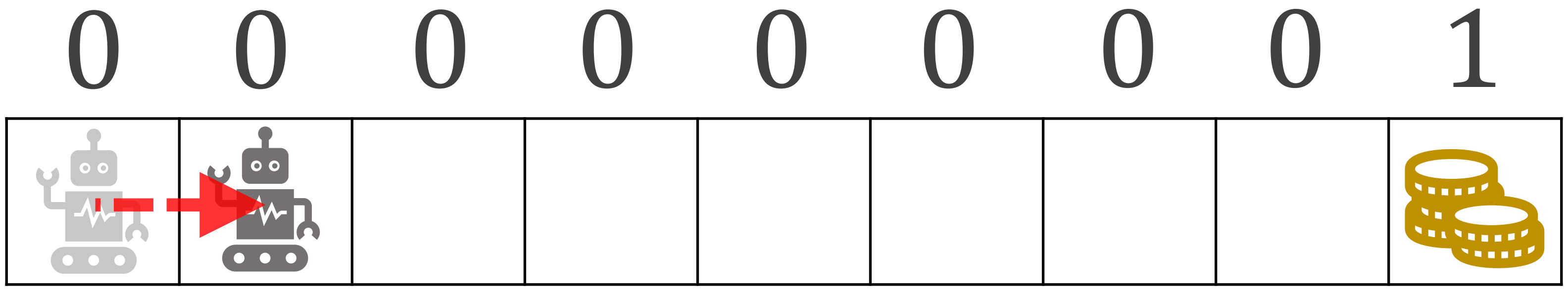
$$r_t = \begin{cases} 1 & \text{if } \quad \text{ at } \\ 0 & \text{otherwise} \end{cases}$$



Sparse Rewards

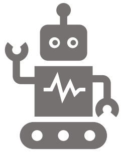
$r_t =$ 1 if  at 

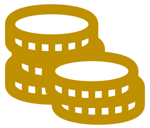
0 otherwise



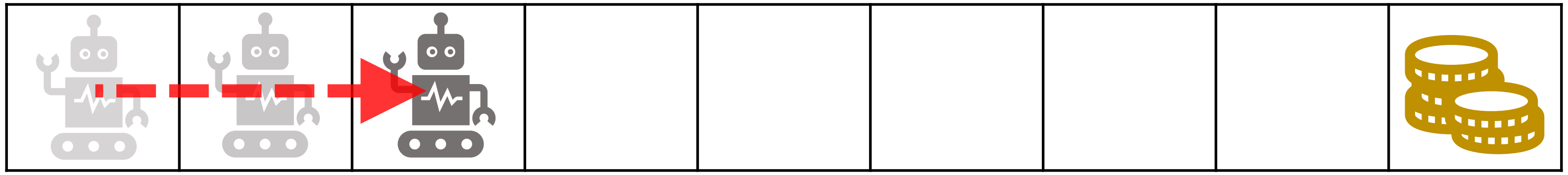
Sparse Rewards

$$r_t = \begin{cases} 1 & \text{if } \quad \text{at} \\ 0 & \text{otherwise} \end{cases}$$



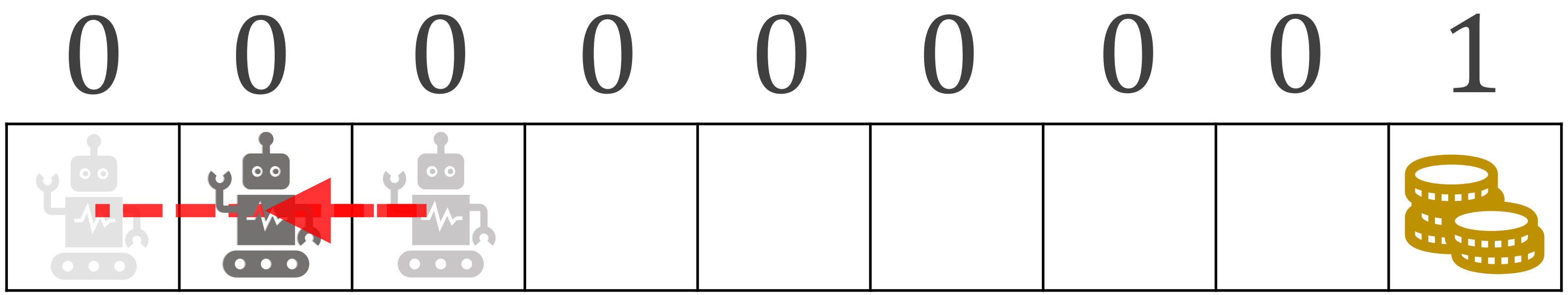


0 0 0 0 0 0 0 0 1

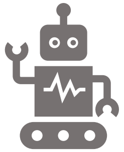
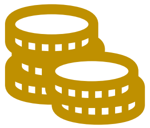


Sparse Rewards

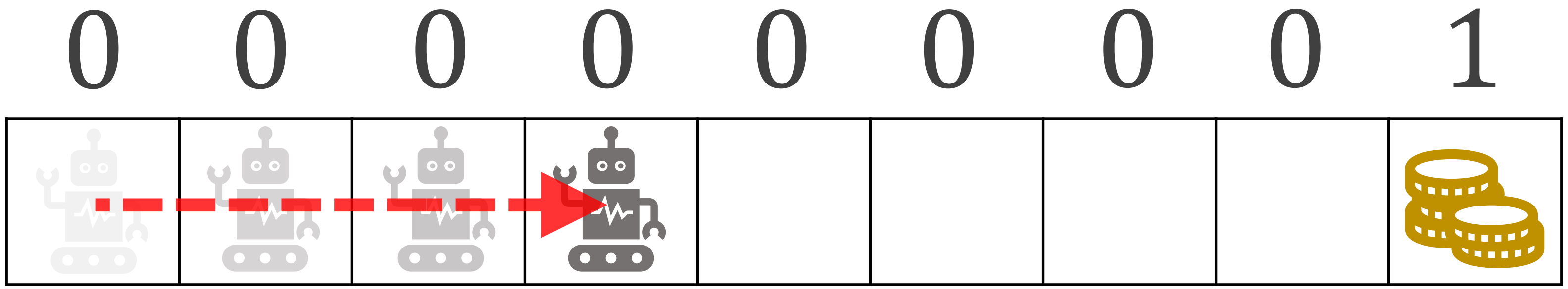
$$r_t = \begin{cases} 1 & \text{if } \quad \text{at } \\ 0 & \text{otherwise} \end{cases}$$



Sparse Rewards

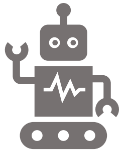
$r_t = \begin{cases} 1 & \text{if} \end{cases}$  at 

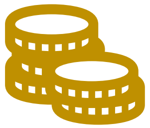
$\phantom{r_t = \begin{cases}} 0 \quad \text{otherwise}$

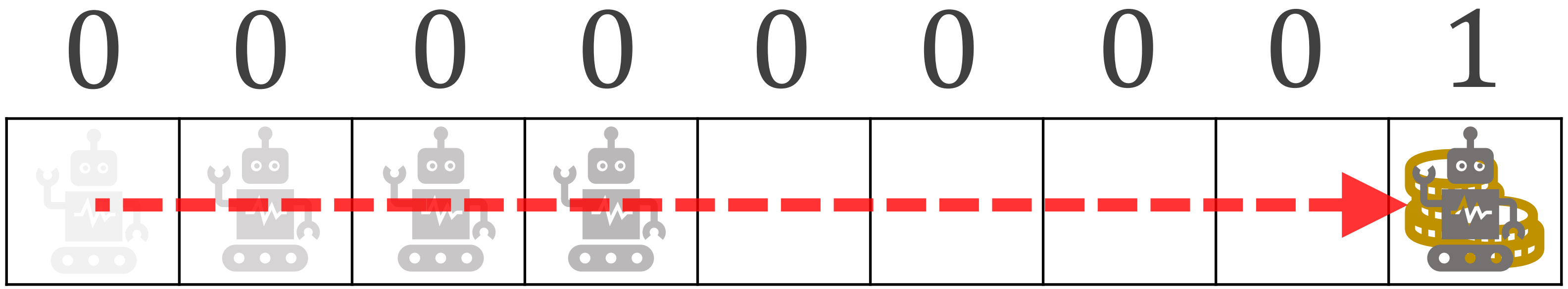


Sparse Rewards

$$r_t = \begin{cases} 1 & \text{if} \quad \text{at} \\ 0 & \text{otherwise} \end{cases}$$

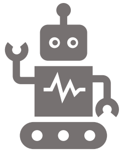


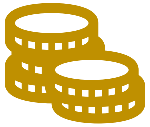




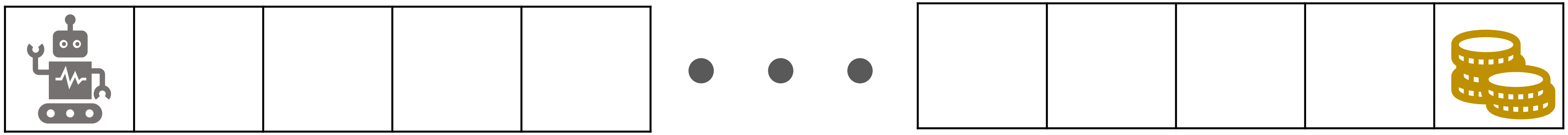
Sparse Rewards

$$r_t = \begin{cases} 1 & \text{if } \square \text{ at } \square \\ 0 & \text{otherwise} \end{cases}$$

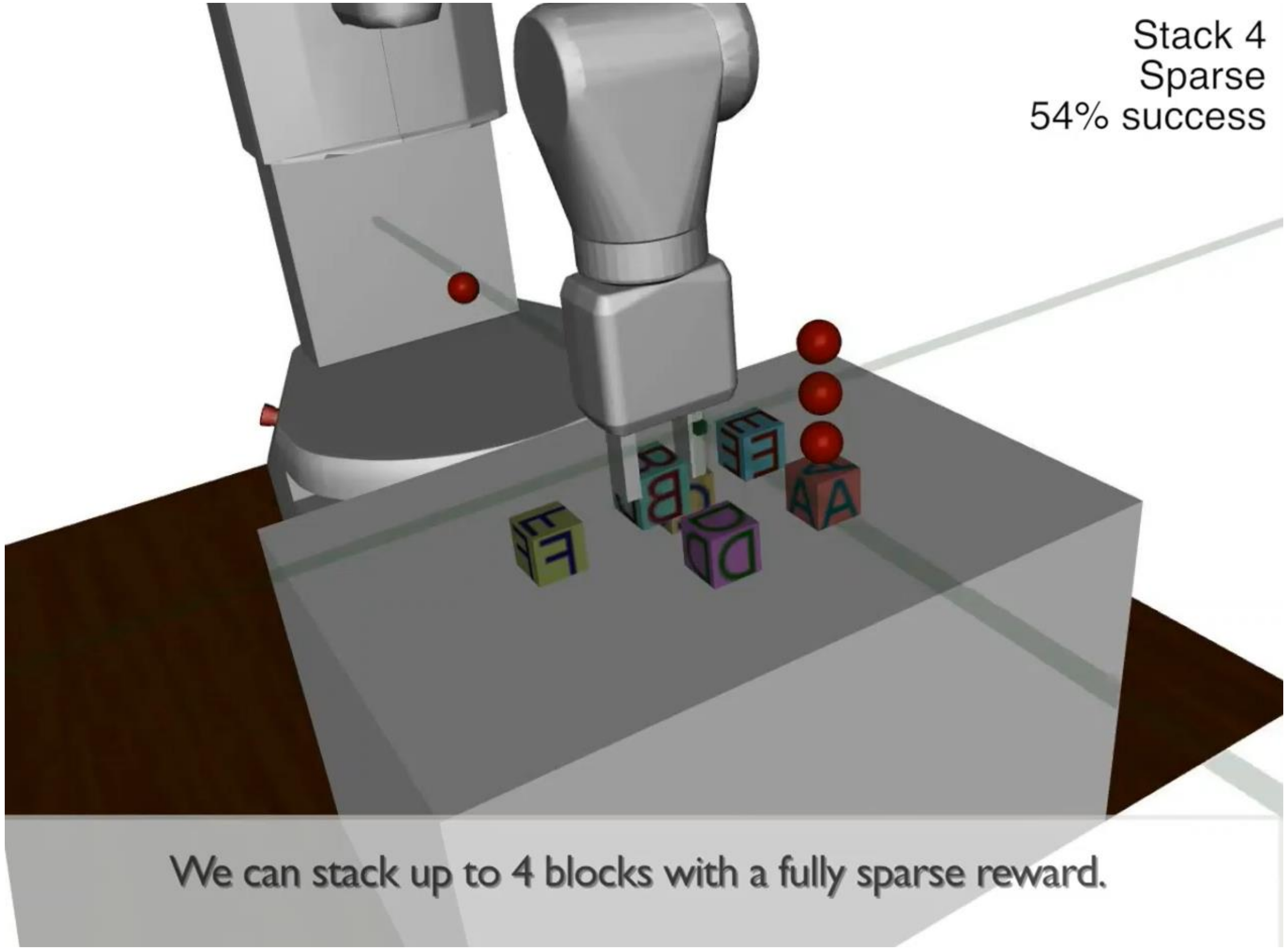




Random exploration can be very inefficient for long horizon tasks



Long Horizon Tasks



Overcoming Exploration in Reinforcement Learning with Demonstrations
[Nair et al. 2018]

Dense vs Sparse Rewards

Dense reward

✓ Frequent feedback (faster learning)

✓ Easier exploration

✘ Shaping bias

✘ Harder to design

Sparse reward

✘ Infrequent feedback (slower learning)

✘ Harder exploration

✓ Less shaping bias

✓ Easier to design

Dense vs Sparse Rewards

Dense reward

✓ Frequent feedback (faster learning)

✓ Better exploration

✘ Shaping bias

✘ Harder to design

Sparse reward

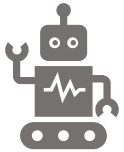
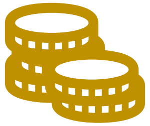
✘ Infrequent feedback (slower learning)

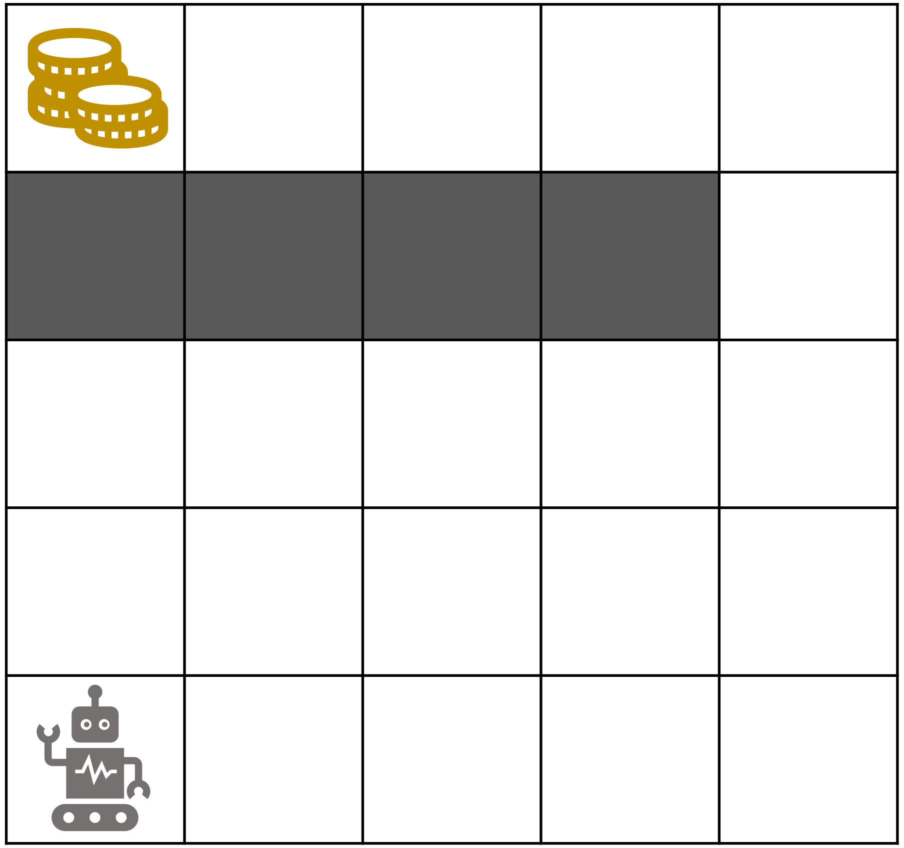
✘ Harder exploration

✓ Less shaping bias

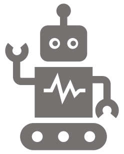
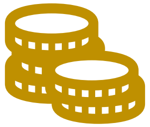
✓ Easier to design

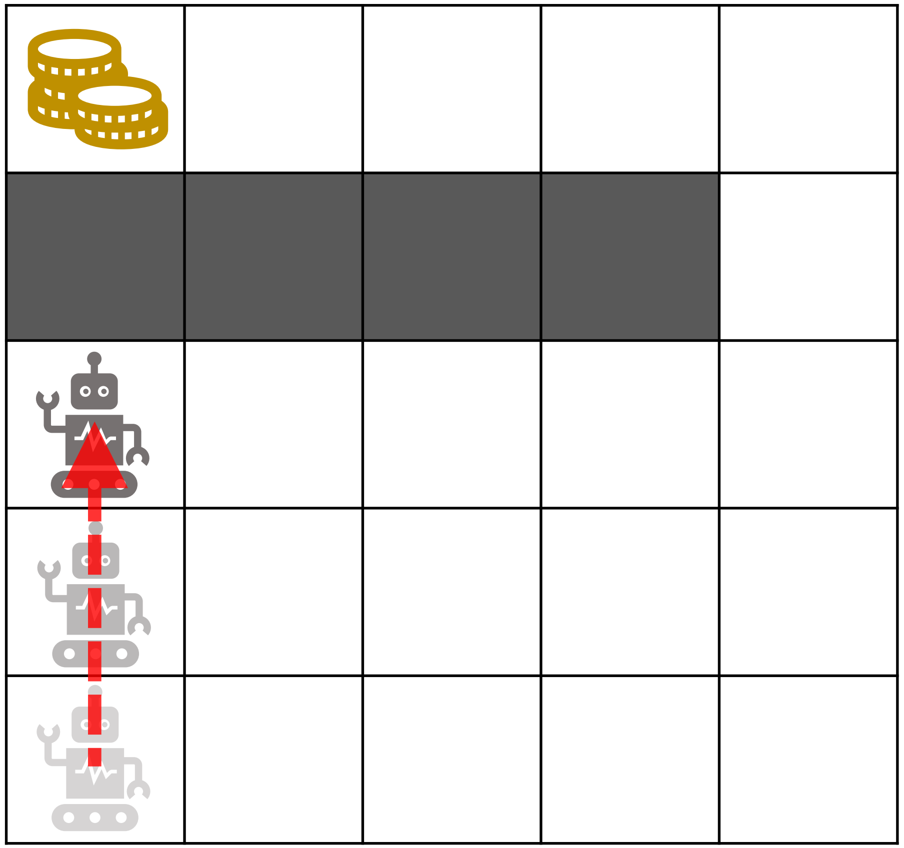
Shaping Bias

$r_t = -||$  $-$  $||^2$

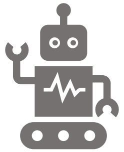
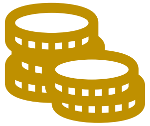


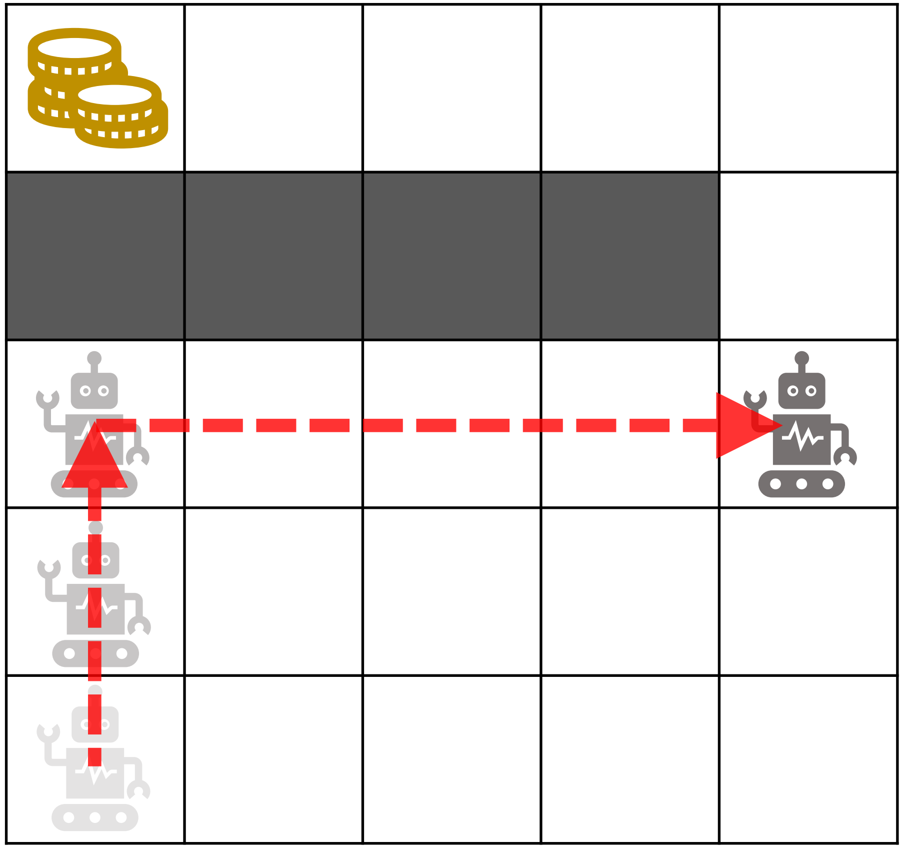
Shaping Bias

$r_t = -||$  $-$  $||^2$

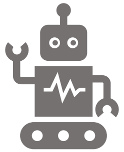
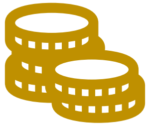


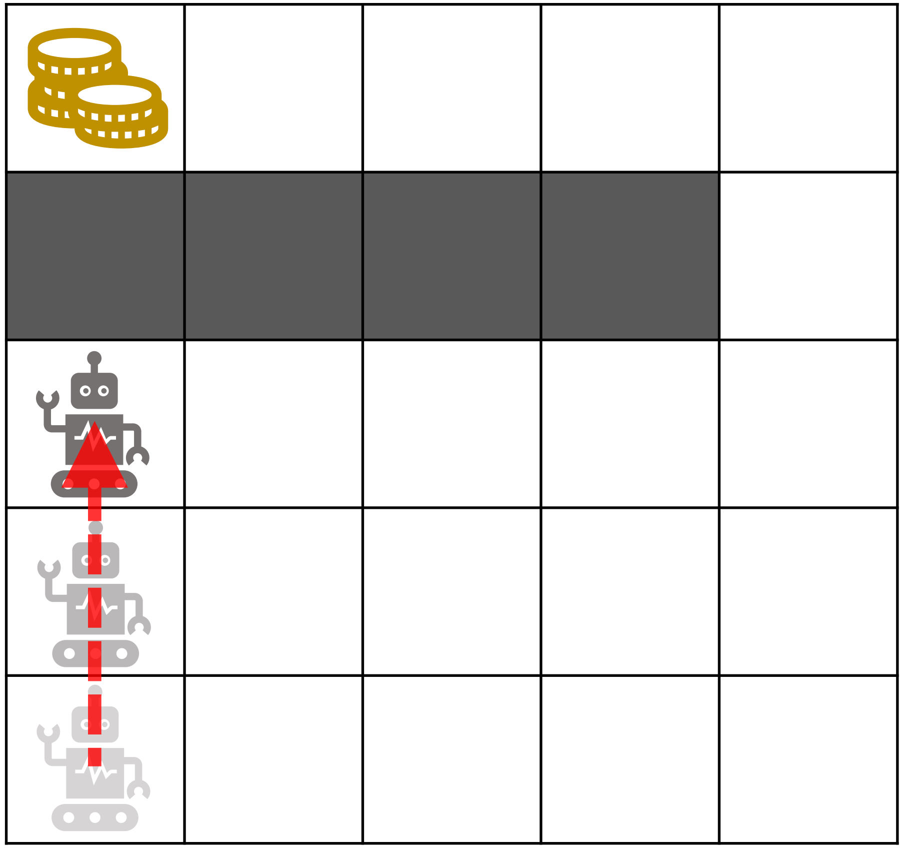
Shaping Bias

$r_t = - ||$  $-$  $||^2$



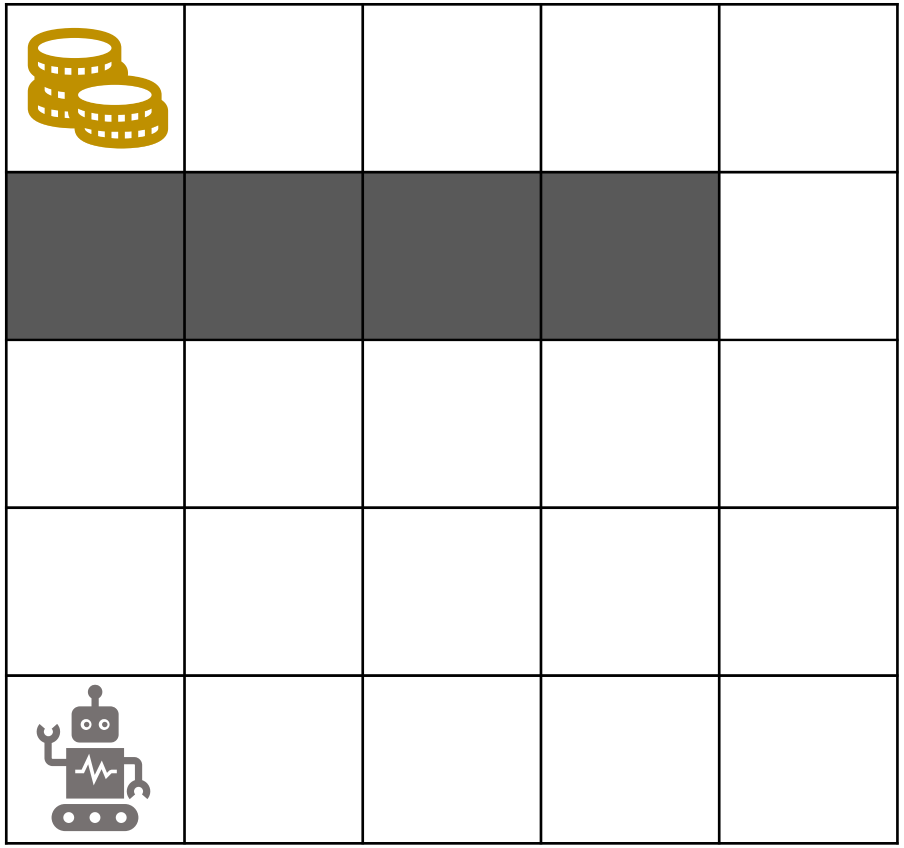
Shaping Bias

$r_t = -||$  $-$  $||^2$



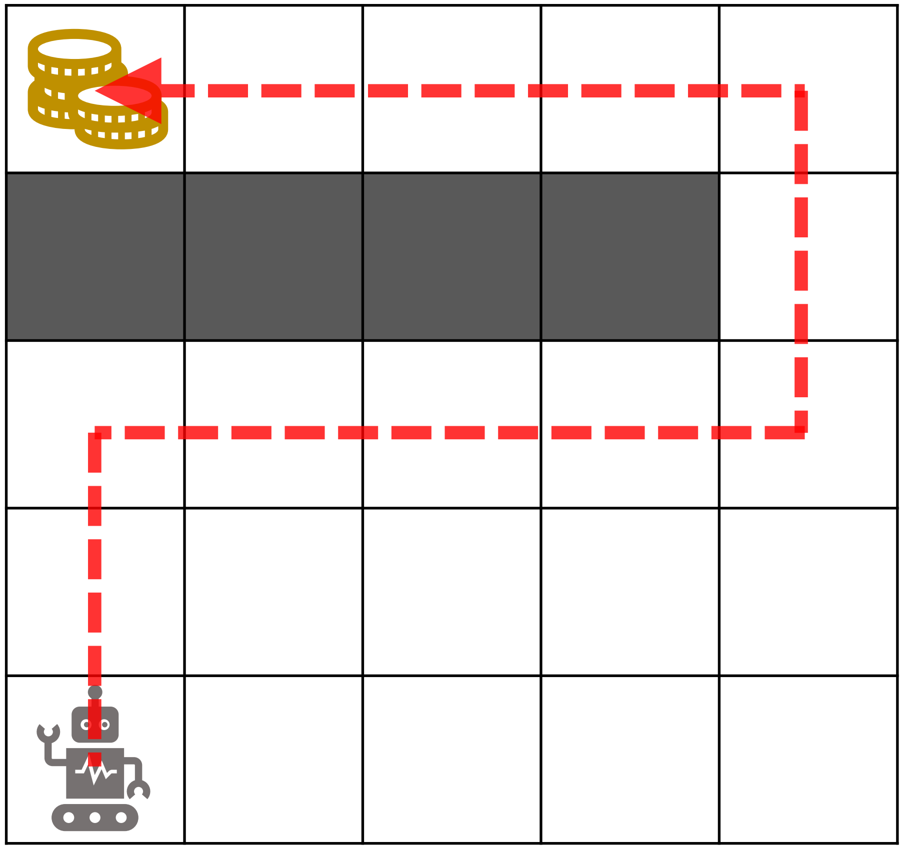
Shaping Bias

$$r_t = \begin{cases} 1 & \text{if } \quad \text{ at } \\ 0 & \text{otherwise} \end{cases}$$



Shaping Bias

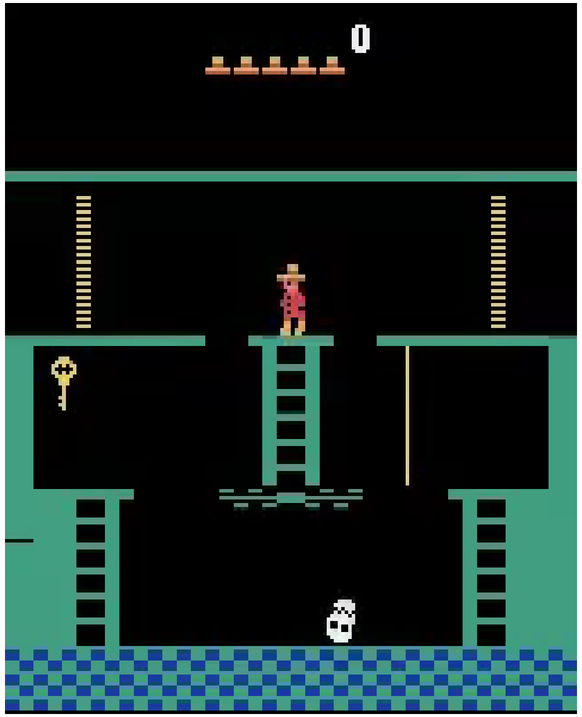
$$r_t = \begin{cases} 1 & \text{if} \quad \text{at} \\ 0 & \text{otherwise} \end{cases}$$



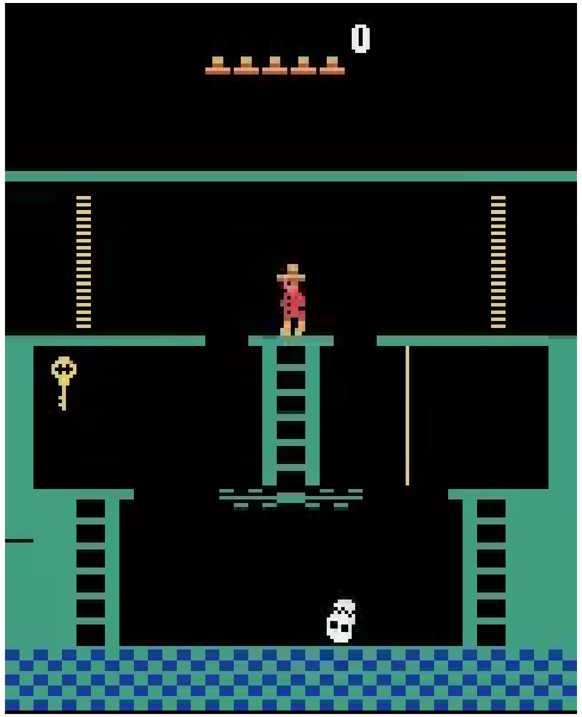
Atari

Fairly Easy

Almost Impossible

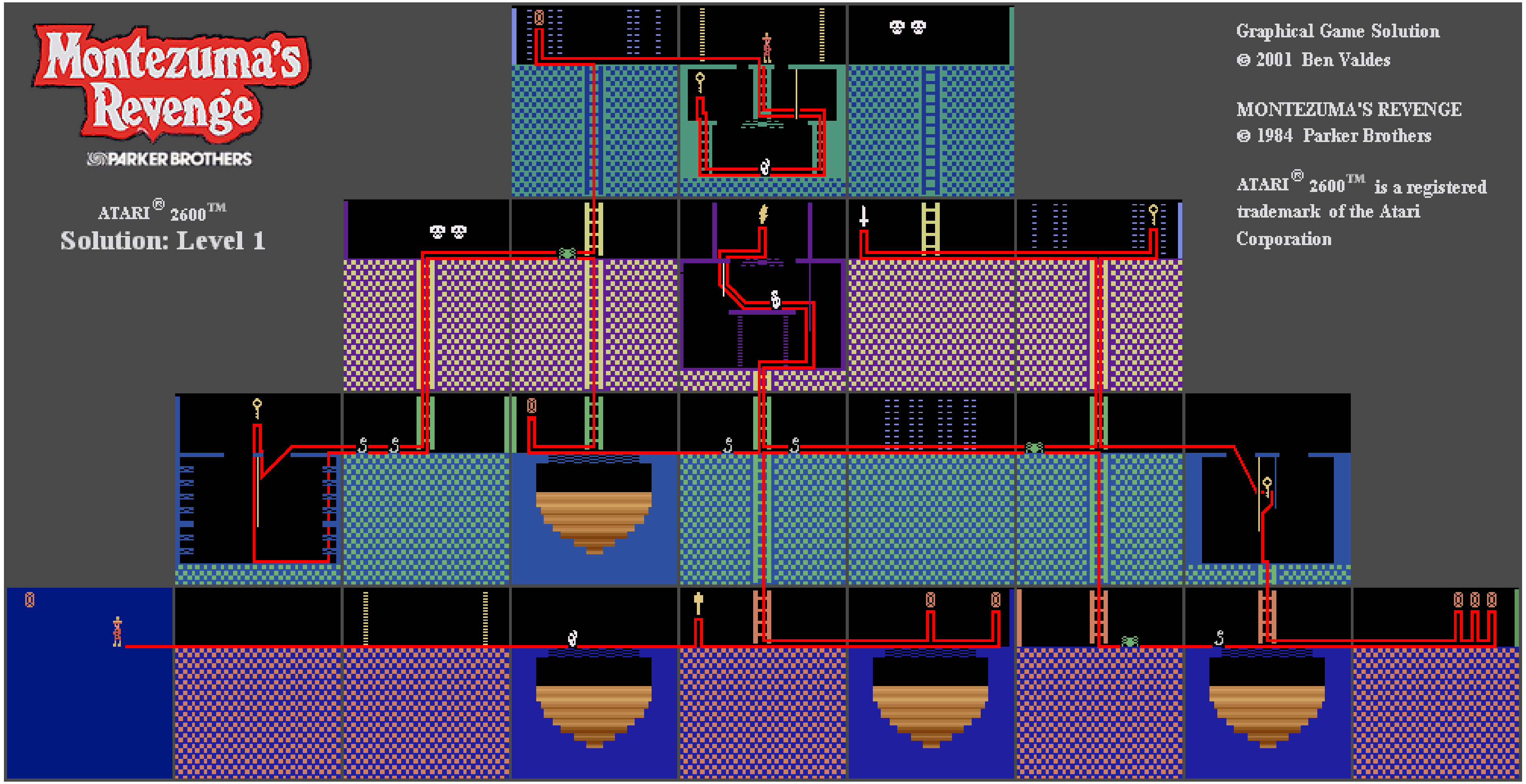


Montezuma's Revenge



- Very sparse reward
- +1: getting key
- +1: opening door
- RL algorithm has no idea what keys and doors are

Montezuma’s Revenge

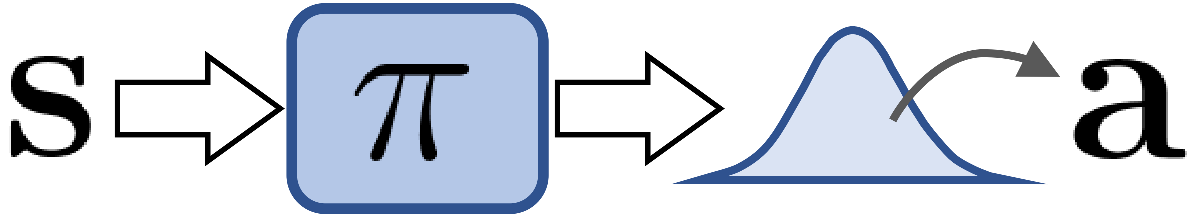


[Graphic Created by Ben Valdes]

Better Exploration Strategies

- Agent needs to visit new states and try new actions to find optimal strategies
- Encourage coverage of *both* states and actions

relatively easy

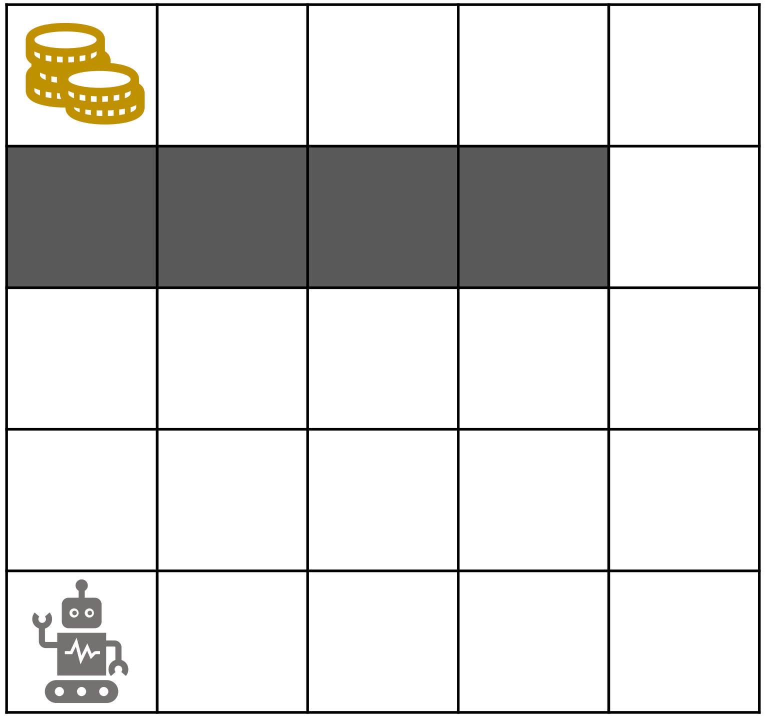


Better Exploration Strategies

- Agent needs to visit new states and try new actions to find optimal strategies
- Encourage coverage of *both* states and actions

much harder

encourage agent to
visit new states



Intrinsic Motivation

$$r_t$$

Intrinsic Motivation

$$\hat{r}_t = \underbrace{r_t^{\mathrm{ext}}}_{\text{Extrinsic Reward}} + \beta \underbrace{r_t^{\mathrm{int}}}_{\text{Intrinsic Reward}}$$

Extrinsic reward

- from environment
- encourage agent to perform a given task

Intrinsic reward

- from the agent itself
- encourage agent to explore new states

Intrinsic Reward

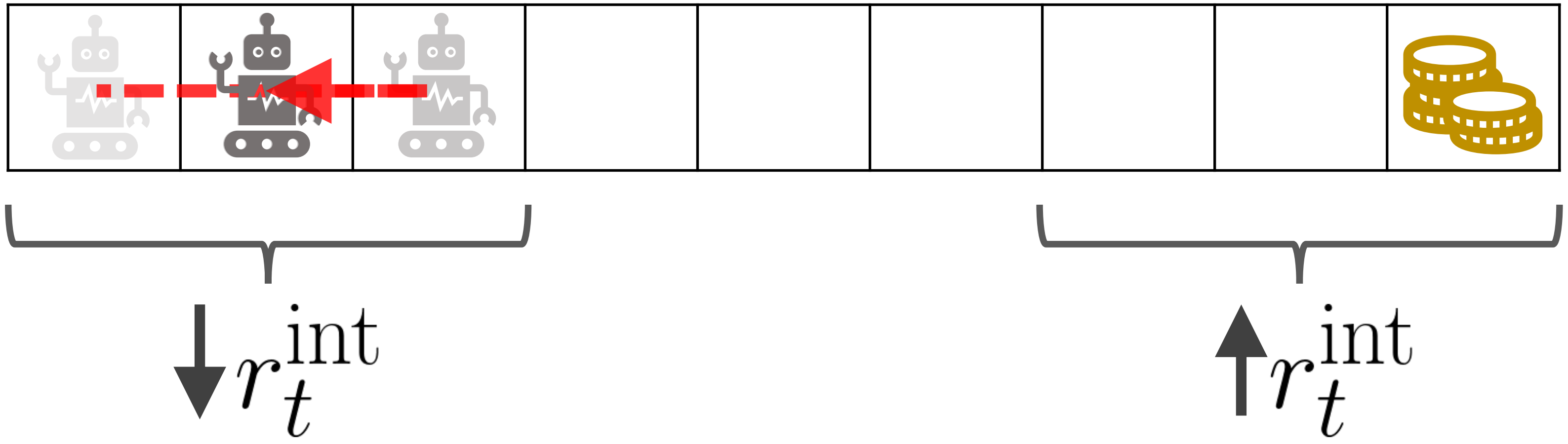
- Nonstationary reward
- Low for frequently visited states $\downarrow r_t^{\text{int}}$
- High for rarely visited states $\uparrow r_t^{\text{int}}$

Count-Based Exploration

Keep count $N(\mathbf{s})$ on how many times agent visited a particular state

dense reward

$$\boxed{r_t^{\text{int}}} = \frac{1}{1 + N(\mathbf{s}_t)}$$



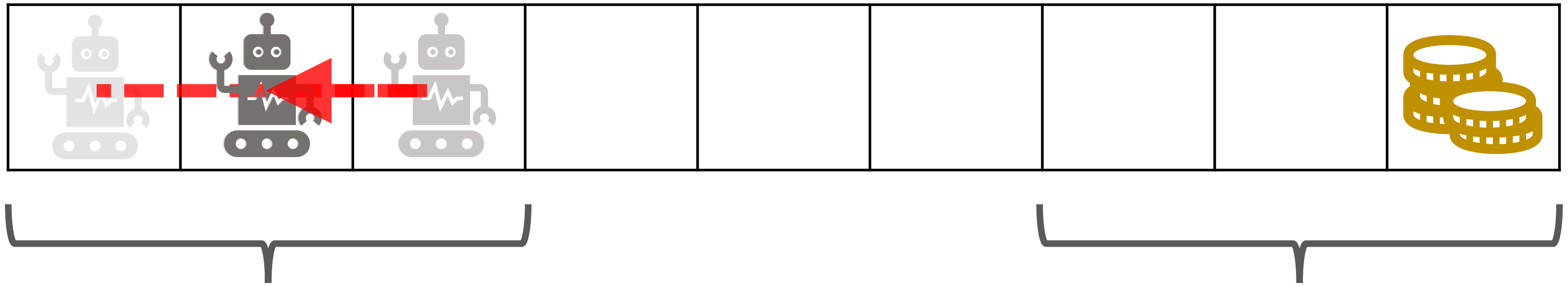
Near-Bayesian Exploration in Polynomial Time
[Kolter and Ng 2009]

Count-Based Exploration

Keep count $N(\mathbf{s})$ on how many times agent visited a particular state

$$r_t^{\text{int}} = \frac{1}{1 + \boxed{N(\mathbf{s}_t)}}$$

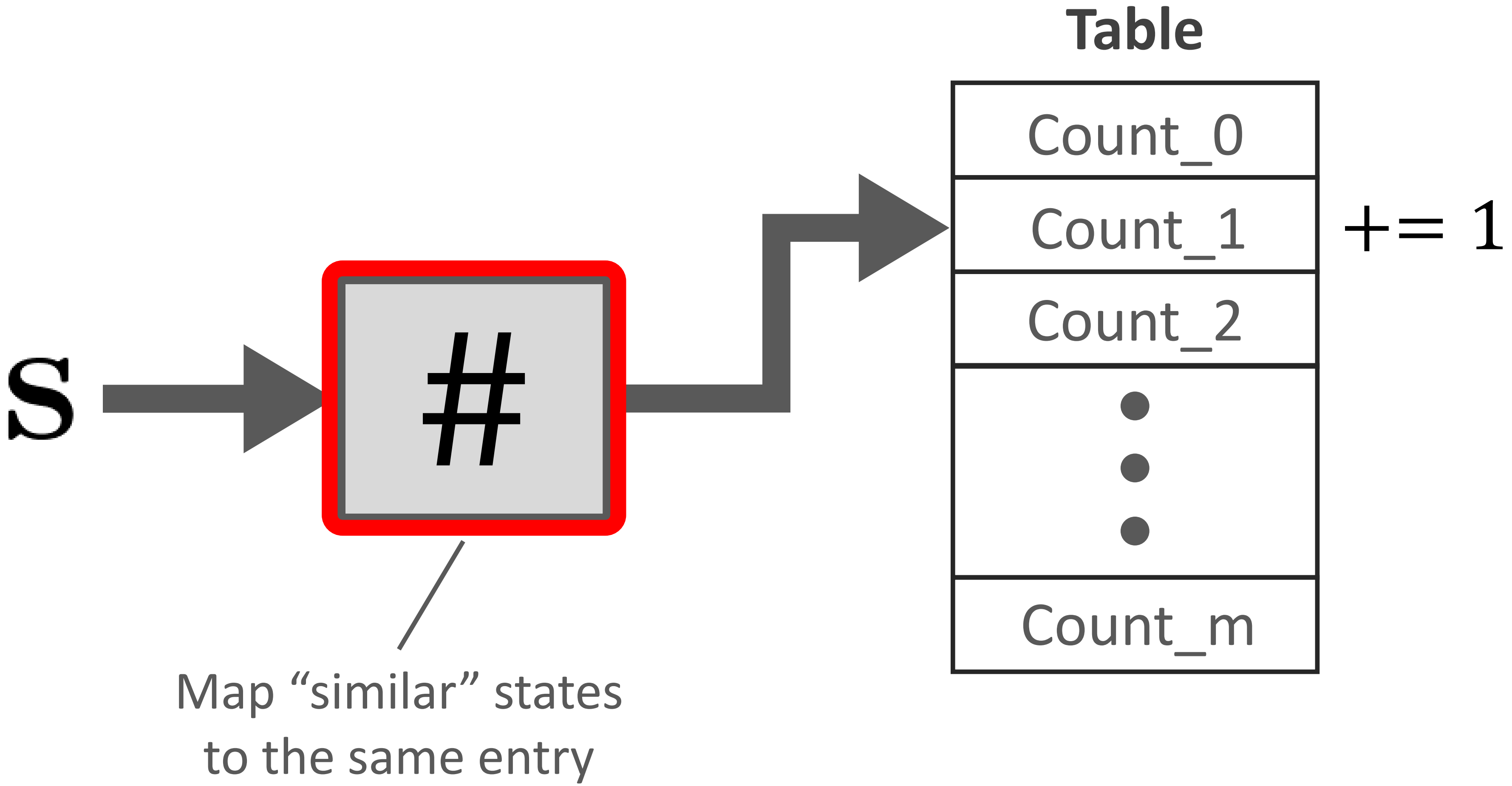
What about large/ continuous states?



$\downarrow r_t^{\text{int}}$ $\uparrow r_t^{\text{int}}$

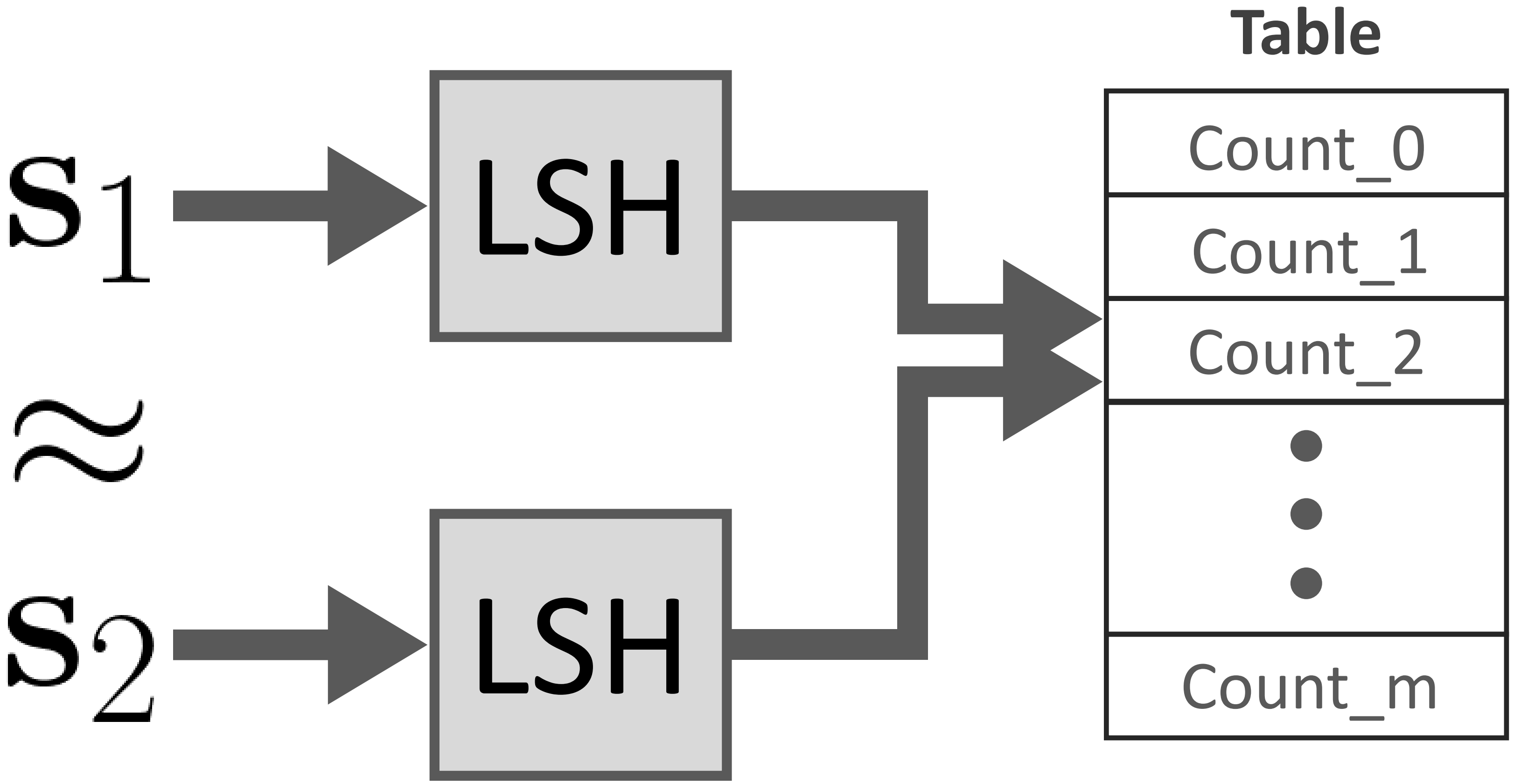
Near-Bayesian Exploration in Polynomial Time
[Kolter and Ng 2009]

State Hashing



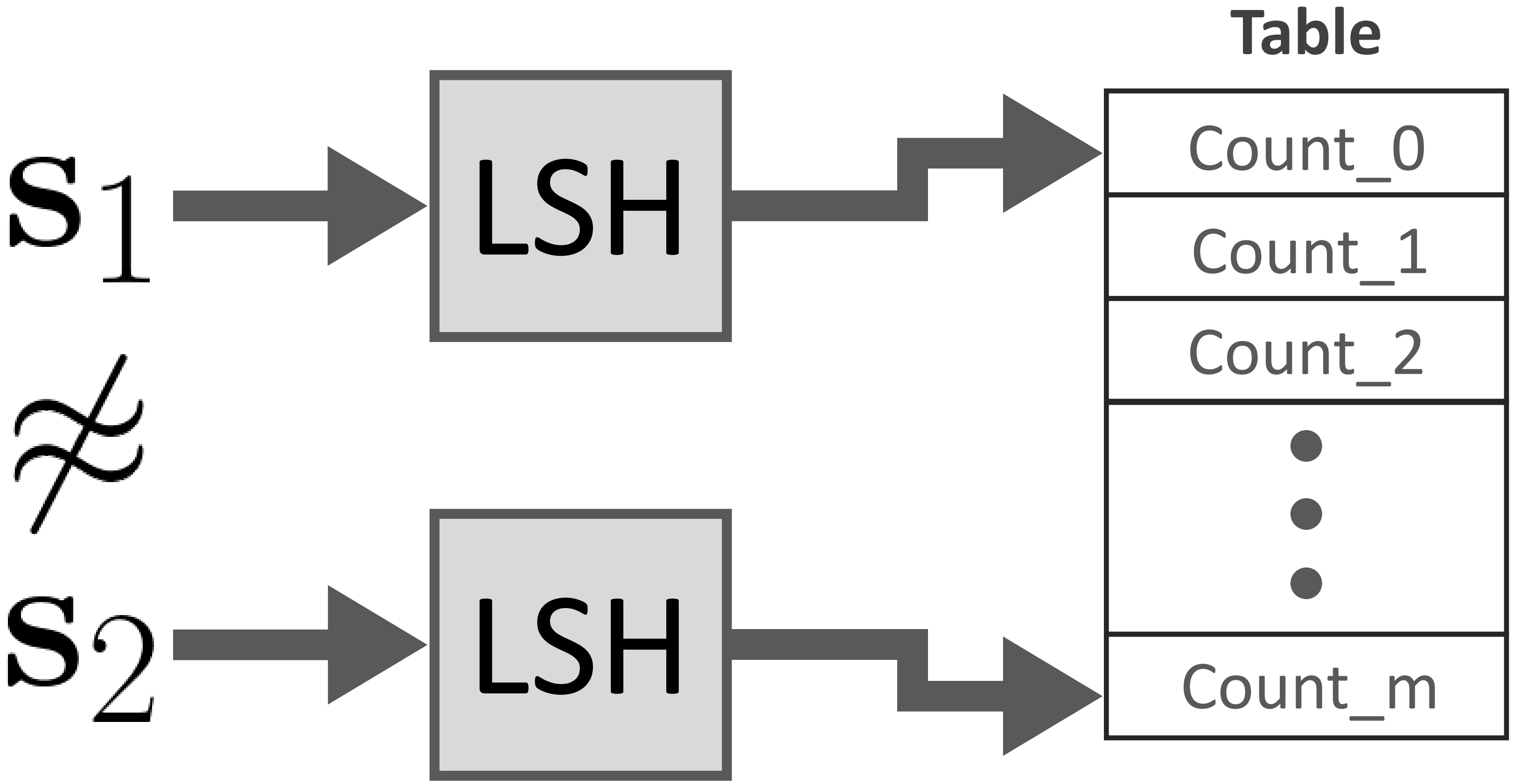
#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning
[Tang et al. 2017]

Locality-Sensitive Hashing



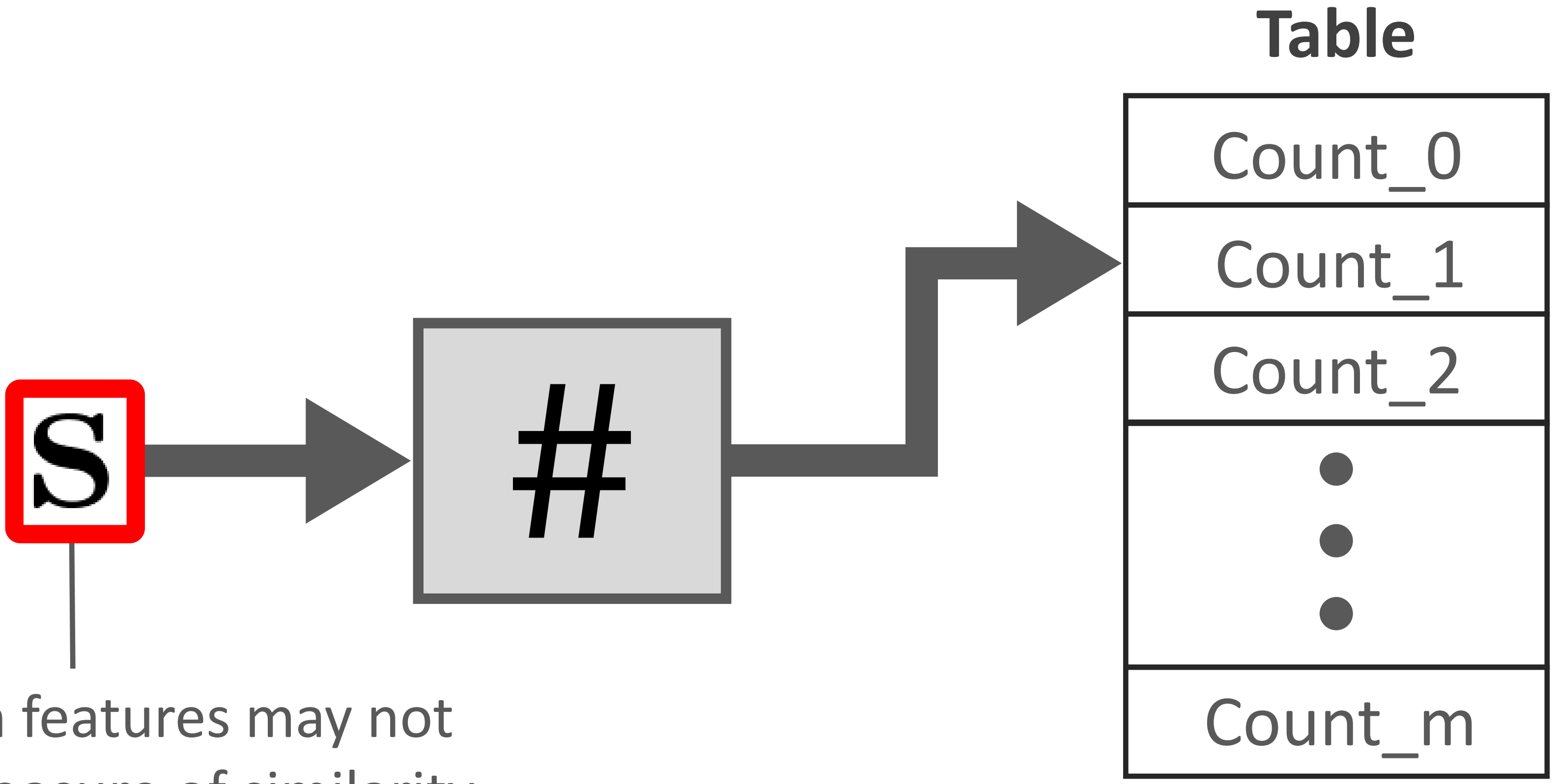
Similarity Estimation Techniques From Rounding Algorithms
[Charikar 2002]

Locality-Sensitive Hashing



Similarity Estimation Techniques From Rounding Algorithms
[Charikar 2002]

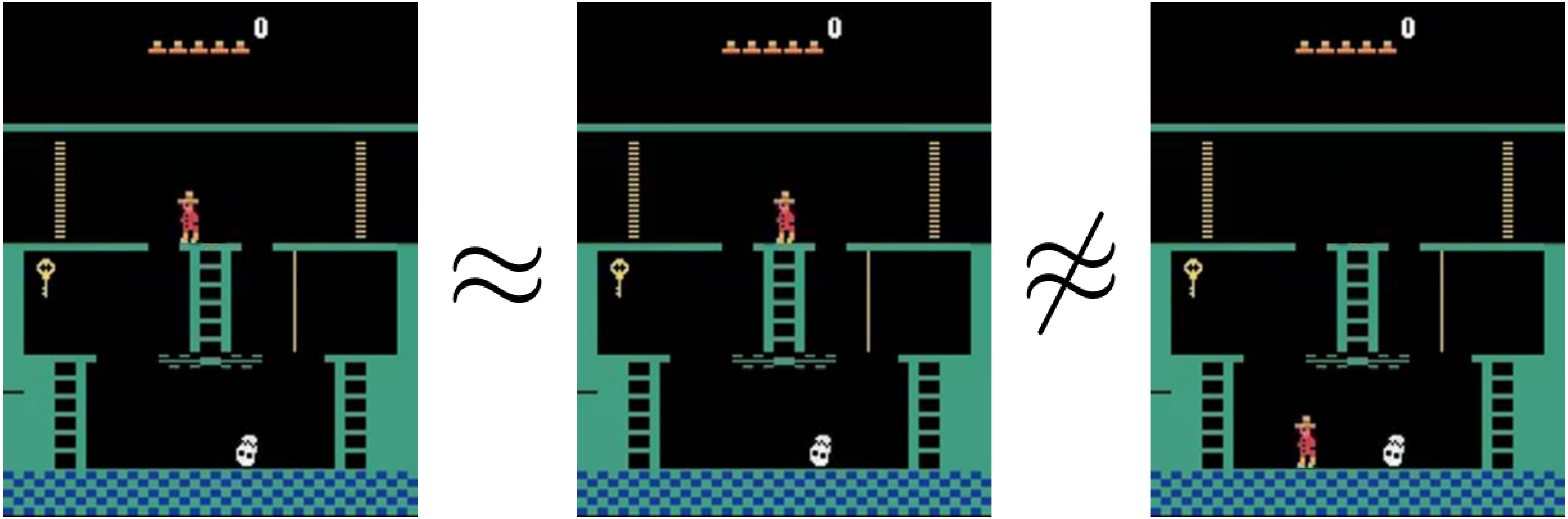
Locality-Sensitive Hashing



Observation features may not be a good measure of similarity

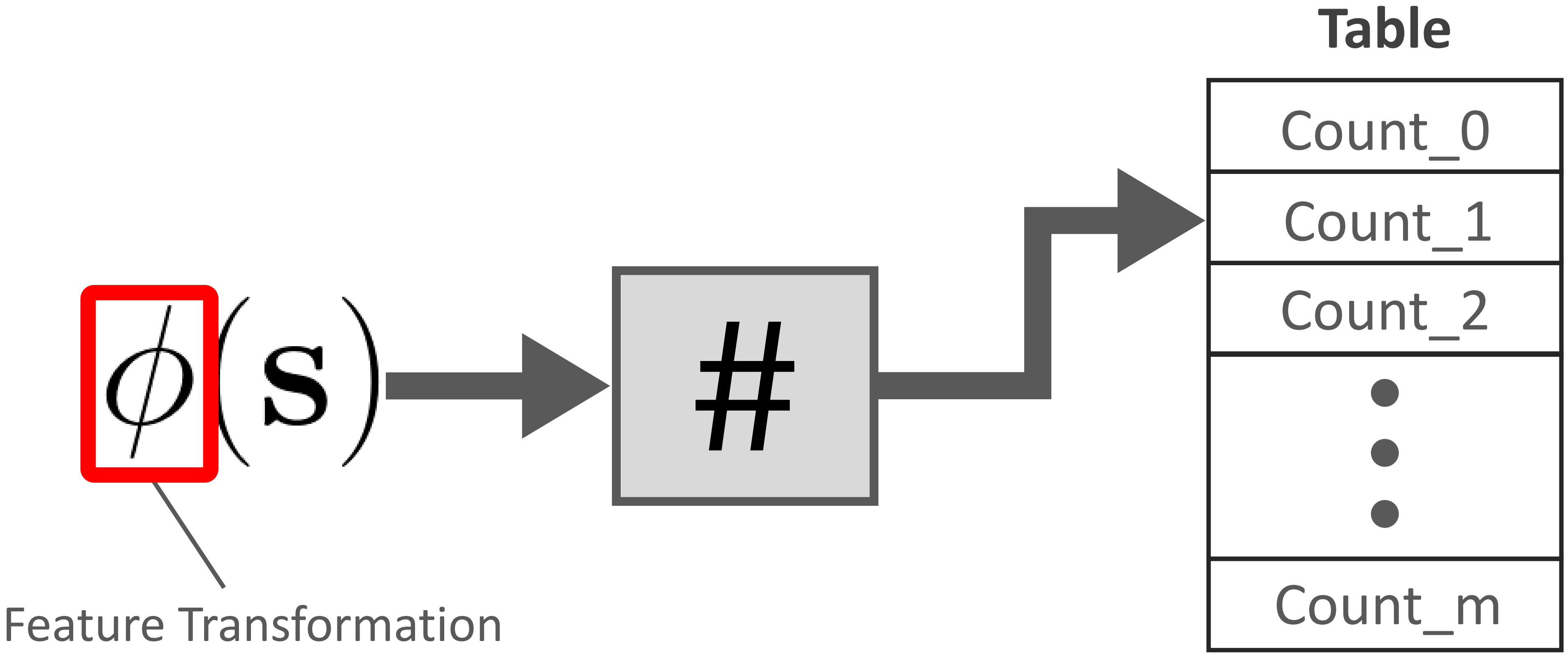
#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning [Tang et al. 2017]

Locality-Sensitive Hashing



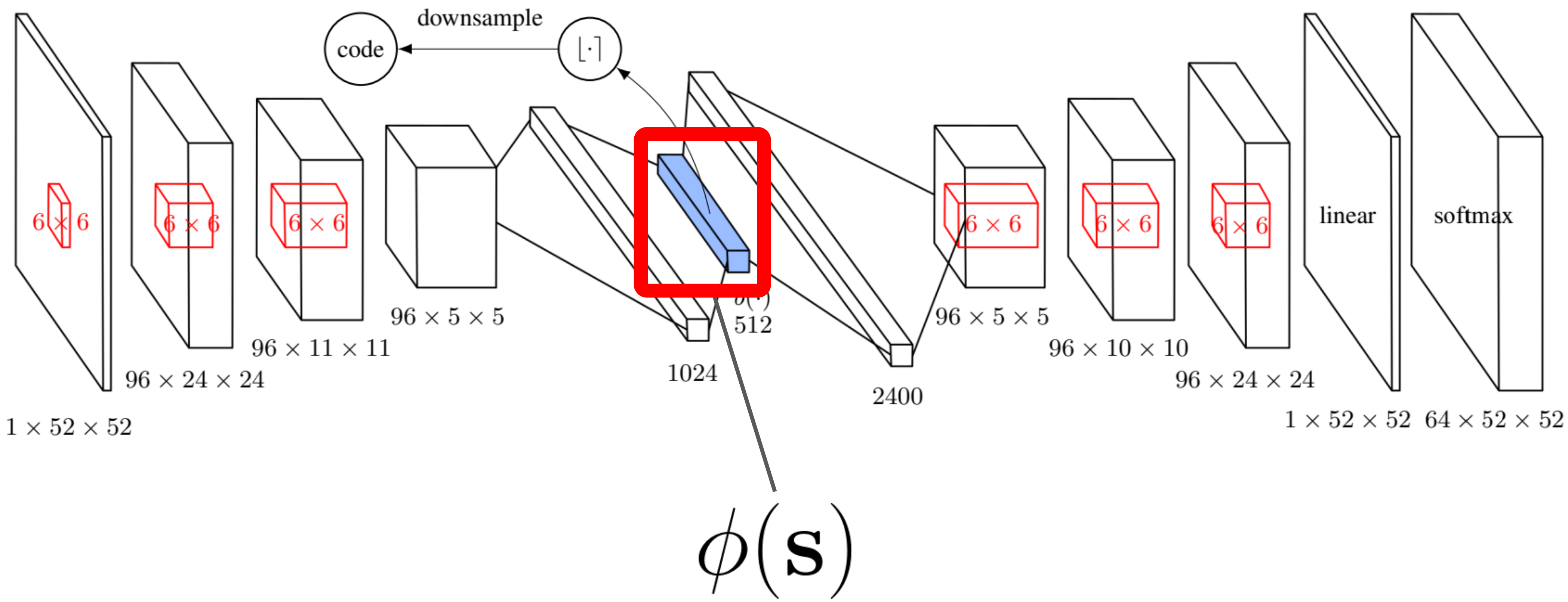
#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning
[Tang et al. 2017]

Locality-Sensitive Hashing



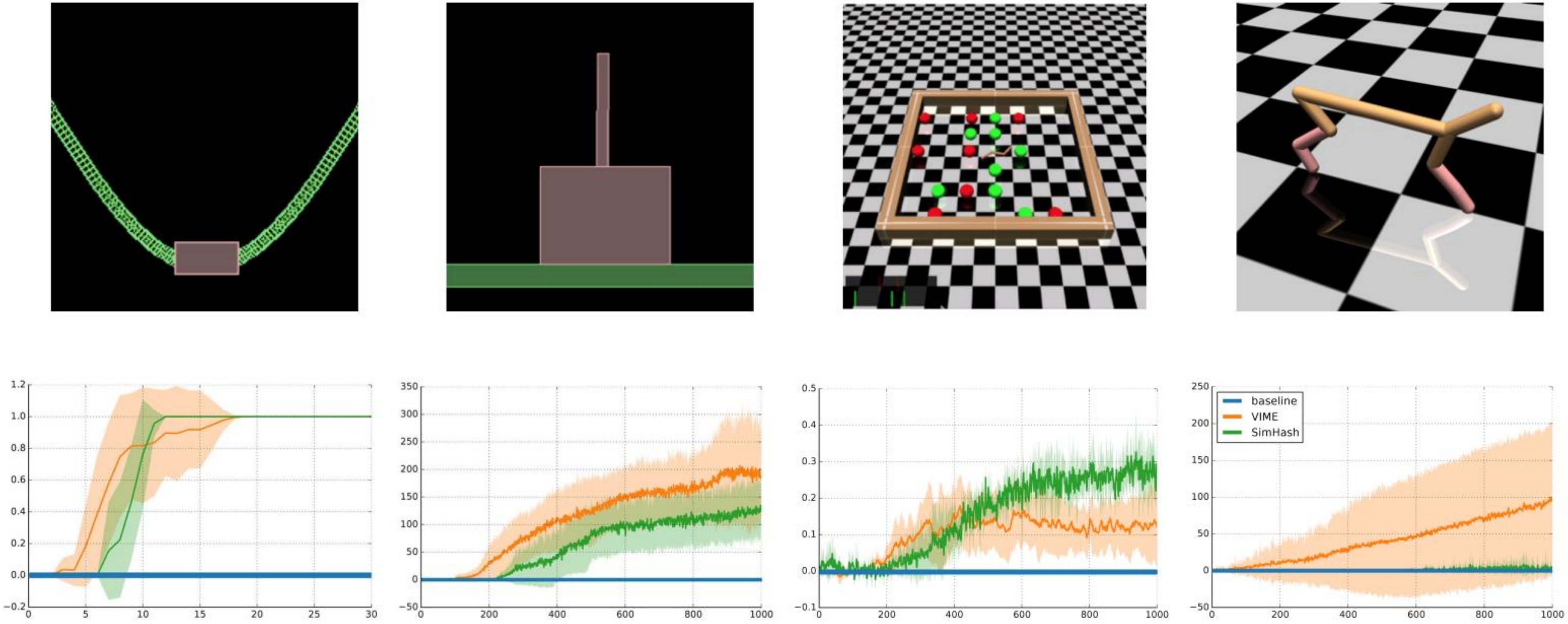
#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning
[Tang et al. 2017]

Feature Embedding



#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning [Tang et al. 2017]

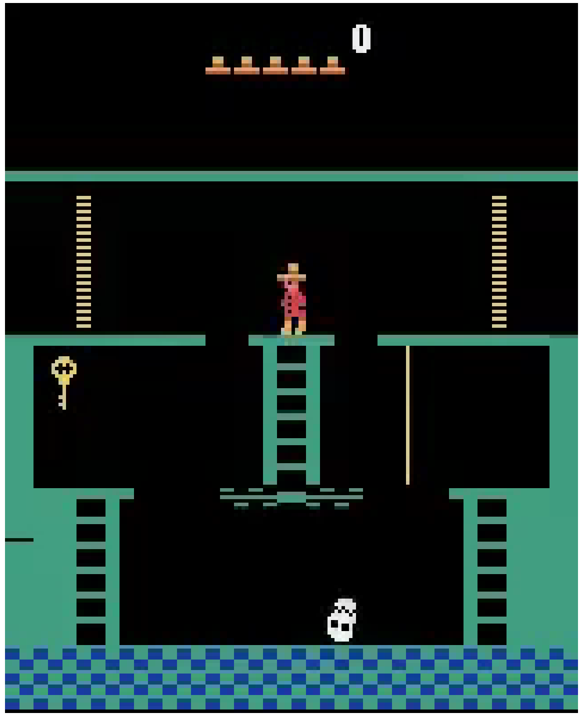
State Hashing

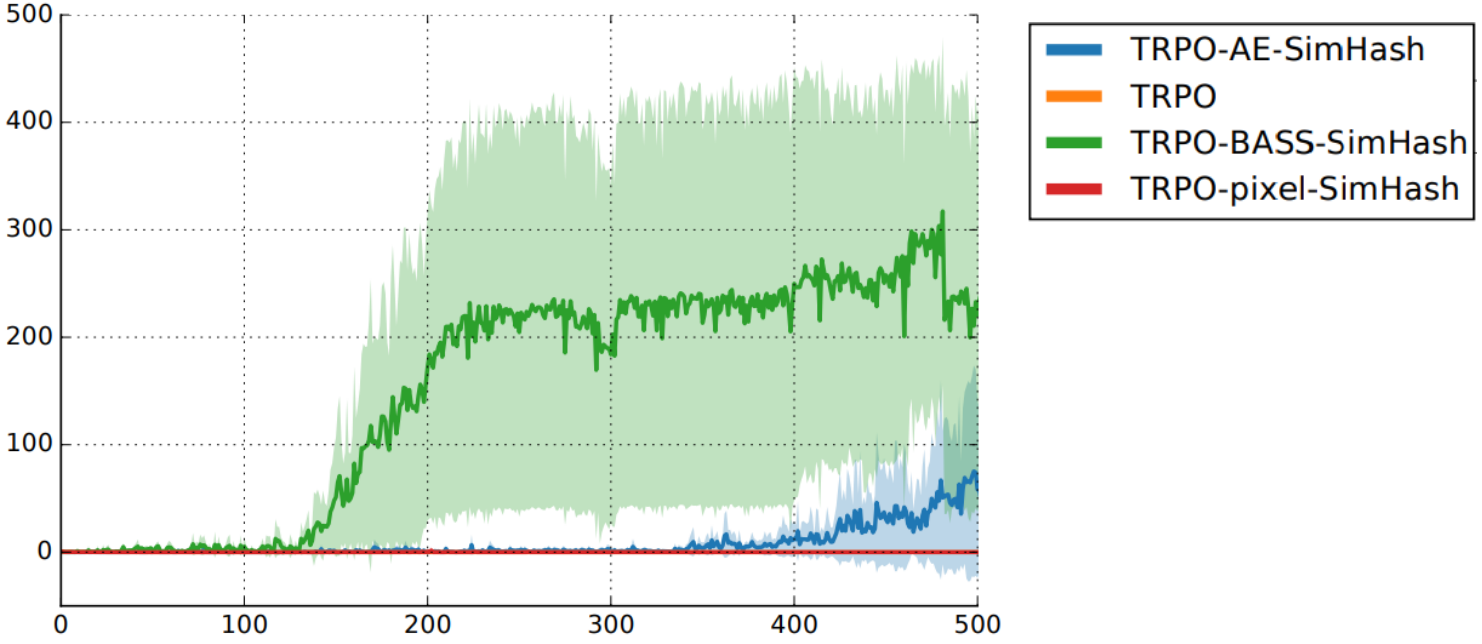


(a) MountainCar (b) CartPoleSwingup (c) SwimmerGather (d) HalfCheetah

#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning
[Tang et al. 2017]

State Hashing





(d) Montezuma's Revenge

#Exploration: A Study of Count-Based Exploration for Deep Reinforcement Learning [Tang et al. 2017]

State Hashing (Drawbacks)

- Learning an effective representation for hashing can be difficult
- Prone to aliasing
- Distribution of states changes during training (feature transform needs to be updated)
- Hard to pick hash table size a-priori

Density Estimation

count

$$p(\mathbf{s}) = \frac{N(\mathbf{s})}{n}$$

probability / density

total timesteps

Density Estimation

$$p(\mathbf{s}) = \frac{N(\mathbf{s})}{n}$$

Idea: use density to estimate count

Density Estimation

Agent visits a state $\mathbf{s}$

Before

$$p(\mathbf{s}) = \frac{N(\mathbf{s})}{n}$$

After

$$p'(\mathbf{s}) = \frac{N(\mathbf{s}) + 1}{n + 1}$$

2 equations and 2 unknowns ($N(\mathbf{s})$, n)

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Density Estimation

Solve for $N(\mathbf{s})$

$$\boxed{p(\mathbf{s})} = \frac{N(\mathbf{s})}{n} \qquad \boxed{p'(\mathbf{s})} = \frac{N(\mathbf{s})+1}{n+1}$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Density Estimation

Solve for $N(\mathbf{s})$

$$p(\mathbf{s}) = \frac{N(\mathbf{s})}{n} \qquad\qquad p'(\mathbf{s}) = \frac{N(\mathbf{s})+1}{n+1}$$

$$N(\mathbf{s}) = np(\mathbf{s})$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Density Estimation

Solve for $N(\mathbf{s})$

$$p(\mathbf{s}) = \frac{N(\mathbf{s})}{n} \qquad p'(\mathbf{s}) = \frac{N(\mathbf{s})+1}{n+1}$$

$$N(\mathbf{s}) = np(\mathbf{s})$$

$$np'(\mathbf{s}) + p'(\mathbf{s}) = N(\mathbf{s}) + 1$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Density Estimation

Solve for $N(\mathbf{s})$

$$p(\mathbf{s}) = \frac{N(\mathbf{s})}{n} \qquad\qquad p'(\mathbf{s}) = \frac{N(\mathbf{s})+1}{n+1}$$

$$N(\mathbf{s}) = np(\mathbf{s})$$

$$\begin{aligned} np'(\mathbf{s}) + p'(\mathbf{s}) &= \underline{N(\mathbf{s})} + 1 \\ &= np(\mathbf{s}) \end{aligned}$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Density Estimation

Solve for $N(\mathbf{s})$

$$p(\mathbf{s}) = \frac{N(\mathbf{s})}{n} \qquad p'(\mathbf{s}) = \frac{N(\mathbf{s})+1}{n+1}$$

$$N(\mathbf{s}) = np(\mathbf{s})$$

$$np'(\mathbf{s}) + p'(\mathbf{s}) = N(\mathbf{s}) + 1$$

$$np'(\mathbf{s}) + p'(\mathbf{s}) = np(\mathbf{s}) + 1$$

$$n(p'(\mathbf{s}) - p(\mathbf{s})) = 1 - p'(\mathbf{s})$$

$$n = \frac{1 - p'(\mathbf{s})}{p'(\mathbf{s}) - p(\mathbf{s})}$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Density Estimation

Solve for $N(\mathbf{s})$

$$p(\mathbf{s}) = \frac{N(\mathbf{s})}{n} \qquad p'(\mathbf{s}) = \frac{N(\mathbf{s})+1}{n+1}$$

$$N(\mathbf{s}) = np(\mathbf{s})$$

$$np'(\mathbf{s}) + p'(\mathbf{s}) = N(\mathbf{s}) + 1$$

$$np'(\mathbf{s}) + p'(\mathbf{s}) = np(\mathbf{s}) + 1$$

$$n(p'(\mathbf{s}) - p(\mathbf{s})) = 1 - p'(\mathbf{s})$$

$$n = \frac{1 - p'(\mathbf{s})}{p'(\mathbf{s}) - p(\mathbf{s})}$$

$$\boxed{N(\mathbf{s}) = \frac{1 - p'(\mathbf{s})}{p'(\mathbf{s}) - p(\mathbf{s})} p(\mathbf{s})}$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Density Estimation

$$N(\mathbf{s}) = \frac{1 - p'(\mathbf{s})}{p'(\mathbf{s}) - p(\mathbf{s})} p(\mathbf{s})$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Pseudo-Count

$$\hat{N}(\mathbf{s}) = \frac{1 - p'(\mathbf{s})}{p'(\mathbf{s}) - p(\mathbf{s})} p(\mathbf{s})$$

"pseudo-count"

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Pseudo-Count

$$\hat{N}(\mathbf{s}) = \frac{1 - \boxed{p'(\mathbf{s})}}{\boxed{p'(\mathbf{s})} - \boxed{p(\mathbf{s})}} \boxed{p(\mathbf{s})}$$

How?

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Pseudo-Count

$$\hat{N}(\mathbf{s}) = \frac{1 - p'(\mathbf{s})}{p'(\mathbf{s}) - p(\mathbf{s})} p(\mathbf{s})$$

Fit density model

- E.g. flow models, autoregressive models, CTS (Bellemare et al. 2016], etc.

$$\rho(\mathbf{s}) \approx p(\mathbf{s}) \qquad \rho'(\mathbf{s}) \approx p'(\mathbf{s})$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Exploration Pseudo-Count

ALGORITHM: Exploration with Pseudo Counts

$\mathcal{D} \leftarrow$ initialize dataset
Fit density model $\rho(\mathbf{s})$ to $\mathcal{D}$

for every timestep t **do**
  Observe state $\mathbf{s}_t$
  Sample action from policy $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)$
  Apply $\mathbf{a}_t$ and observe new state $\mathbf{s}_{t+1}$ and extrinsic reward r_t^{ext}
  Store $\mathbf{s}_{t+1}$ in $\mathcal{D}$

  Fit density model $\rho'(\mathbf{s})$ to $\mathcal{D}$
  Use $\rho(\mathbf{s}_{t+1})$ and $\rho'(\mathbf{s}_{t+1})$ to estimate pseudo-count $\hat{N}(\mathbf{s}_{t+1})$
  Calculate intrinsic reward r_t^{int} with $N(\mathbf{s}_{t+1})$
  Calculate reward $r_t = r_t^{\text{ext}} + \beta r_t^{\text{int}}$

  $\rho \leftarrow \rho'$
end for

$$r_t^{\text{int}} = \frac{1}{1 + N(\mathbf{s}_{t+1})}$$

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Exploration Pseudo-Count

ALGORITHM: Exploration with Pseudo Counts

$\mathcal{D} \leftarrow$ initialize dataset
Fit density model $\rho(\mathbf{s})$ to $\mathcal{D}$

for every timestep t **do**
  Observe state $\mathbf{s}_t$
  Sample action from policy $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)$
  Apply $\mathbf{a}_t$ and observe new state $\mathbf{s}_{t+1}$ and extrinsic reward r_t^{ext}
  Store $\mathbf{s}_{t+1}$ in $\mathcal{D}$

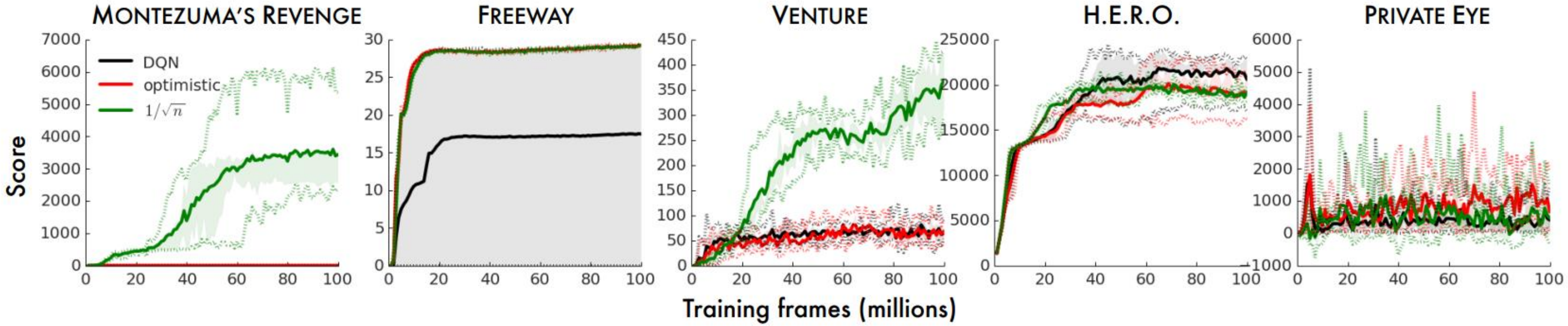
  Fit density model $\rho'(\mathbf{s})$ to $\mathcal{D}$
  Use $\rho(\mathbf{s}_{t+1})$ and $\rho'(\mathbf{s}_{t+1})$ to estimate pseudo-count $\hat{N}(\mathbf{s}_{t+1})$
  $\boxed{\text{Calculate intrinsic reward } r_t^{\text{int}} \text{ with } N(\mathbf{s}_{t+1})}$
  Calculate reward $r_t = r_t^{\text{ext}} + \beta r_t^{\text{int}}$

  $\rho \leftarrow \rho'$
end for

$$r_t^{\text{int}} = \frac{1}{\sqrt{0.01 + N(\mathbf{s}_{t+1})}}$$

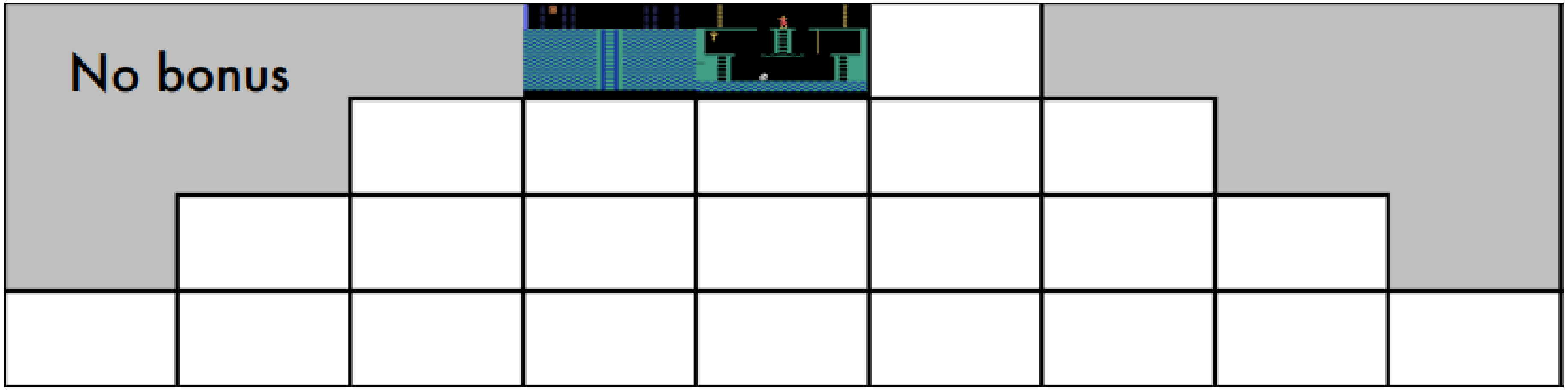
Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

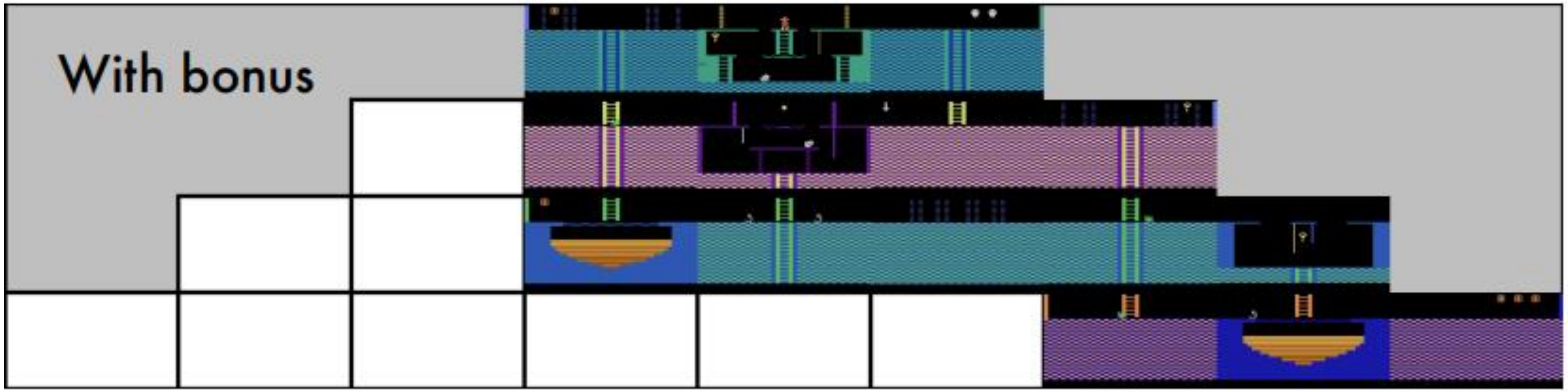
Exploration Pseudo-Count



Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Exploration Pseudo-Count





Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Count-Based Reward

$$r_t^{\text{int}} = \frac{1}{1 + N(\mathbf{s}_{t+1})}$$

$\rho(\mathbf{s})$

Entropy-Based Reward

$$r_t^{\text{int}} = -\log\left(\rho(\mathbf{s}_{t+1})\right)$$

maximize state entropy

$$\mathcal{H}(\mathbf{s})$$

No need to estimate likelihood before and after state visitation

$$\rho(\mathbf{s}) \approx p(\mathbf{s}) \qquad \rho'(\mathbf{s}) \approx p'(\mathbf{s})$$

Exploration Pseudo-Count

ALGORITHM: Exploration with Pseudo Counts

```
$\mathcal{D} \leftarrow$ initialize dataset
Fit density model $\rho(\mathbf{s})$ to $\mathcal{D}$

for every timestep $t$ do
    Observe state $\mathbf{s}_t$
    Sample action from policy $\mathbf{a}_t \sim \pi(\mathbf{a}_t|\mathbf{s}_t)$
    Apply $\mathbf{a}_t$ and observe new state $\mathbf{s}_{t+1}$ and extrinsic reward $r_t^{\text{ext}}$
    Store $\mathbf{s}_{t+1}$ in $\mathcal{D}$

    Fit density model $\rho'(\mathbf{s})$ to $\mathcal{D}$
    Use $\rho(\mathbf{s}_{t+1})$ and $\rho'(\mathbf{s}_{t+1})$ to estimate pseudo-count $\hat{N}(\mathbf{s}_{t+1})$
    Calculate intrinsic reward $r_t^{\text{int}}$ with $\hat{N}(\mathbf{s}_{t+1})$
    Calculate reward $r_t = r_t^{\text{ext}} + \beta r_t^{\text{int}}$

    $\rho \leftarrow \rho'$
end for
```

Density estimation
is hard!

Unifying Count-Based Exploration and Intrinsic Motivation
[Bellemare et al. 2016]

Surprise Maximization

[Sikana]

[Science Channel]

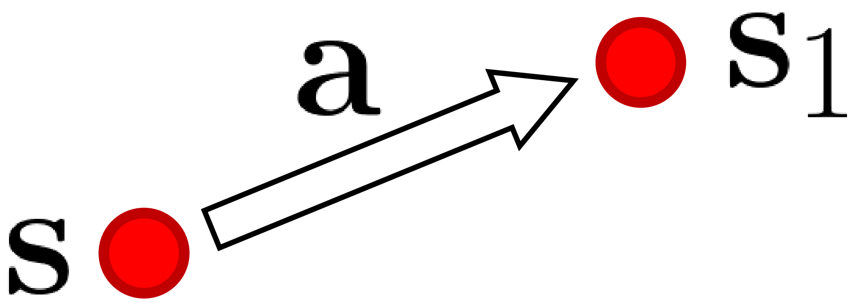
Surprise Maximization

- Use surprise as a proxy for novelty

How do we measure surprise?

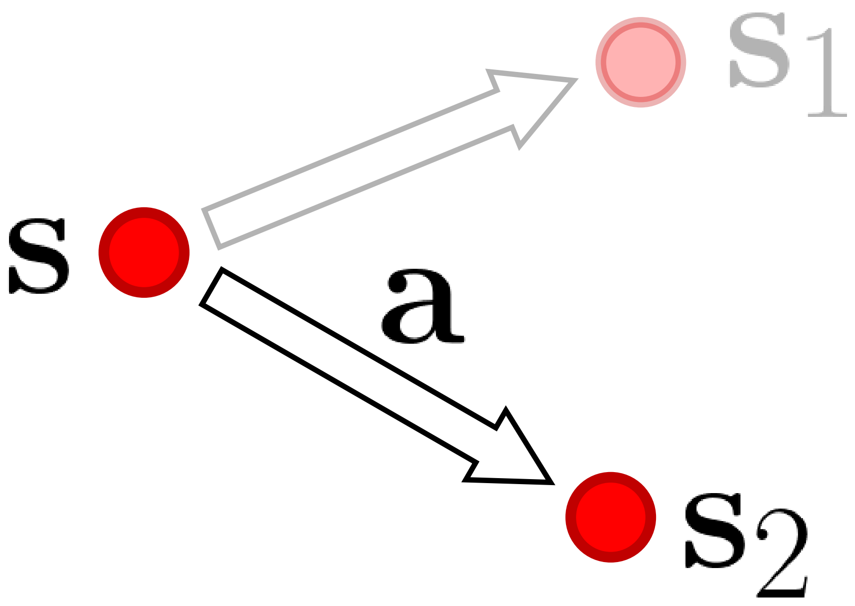
Surprise Maximization

- Use surprise as a proxy for novelty
- Detect surprise via prediction error



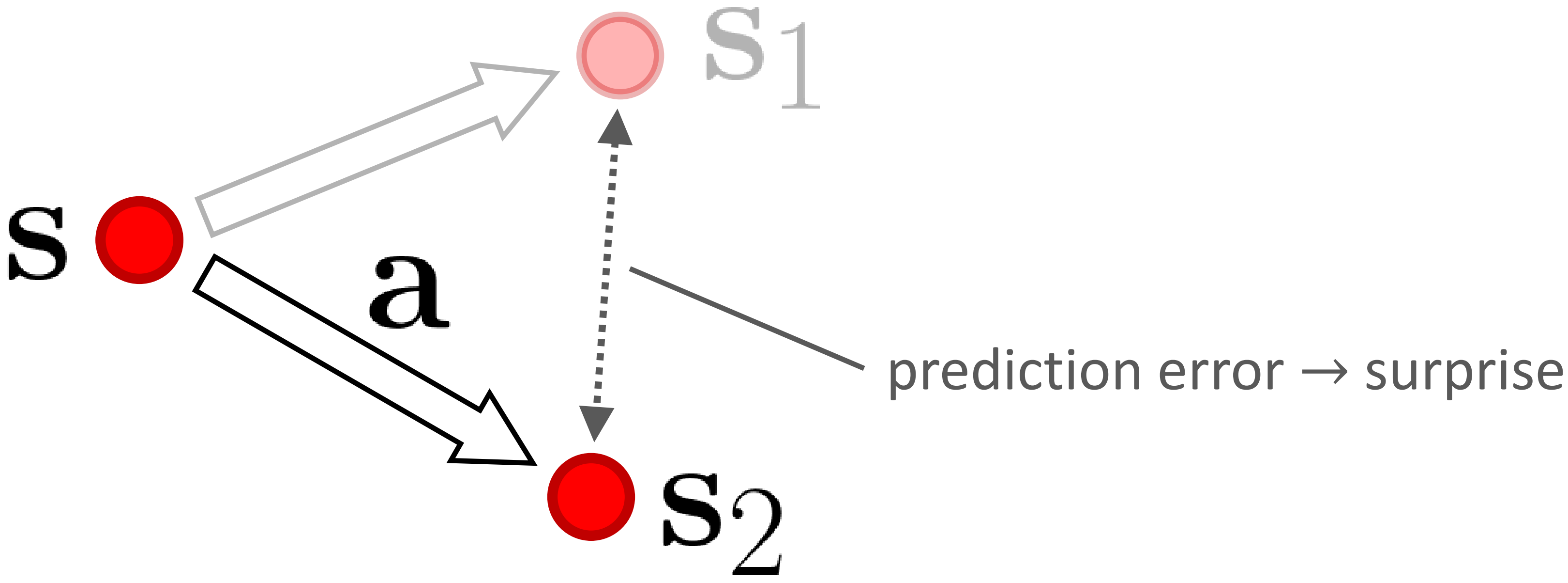
Surprise Maximization

- Use surprise as a proxy for novelty
- Detect surprise via prediction error



Surprise Maximization

- Use surprise as a proxy for novelty
- Detect surprise via prediction error



Dynamics Model

Detect surprise using a dynamics model

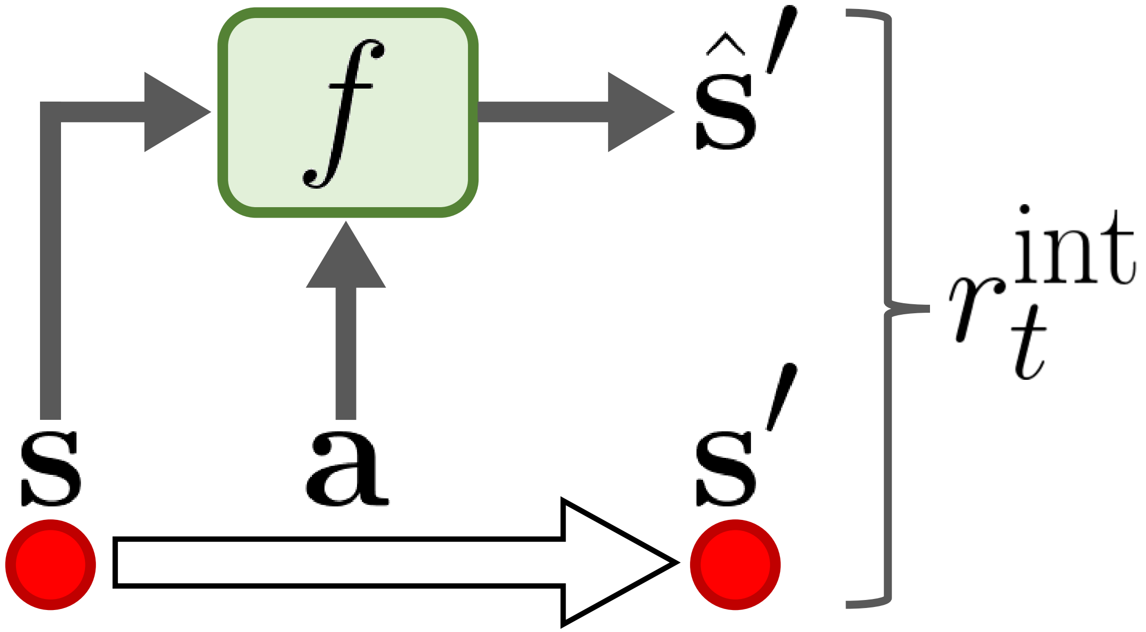
$$f(\mathbf{s}'|\mathbf{s}, \mathbf{a})$$

Intrinsic reward maximizes prediction error

$$r_t^{\text{int}} = -\log f(\mathbf{s}_{t+1}|\mathbf{s}_t, \mathbf{a}_t)$$

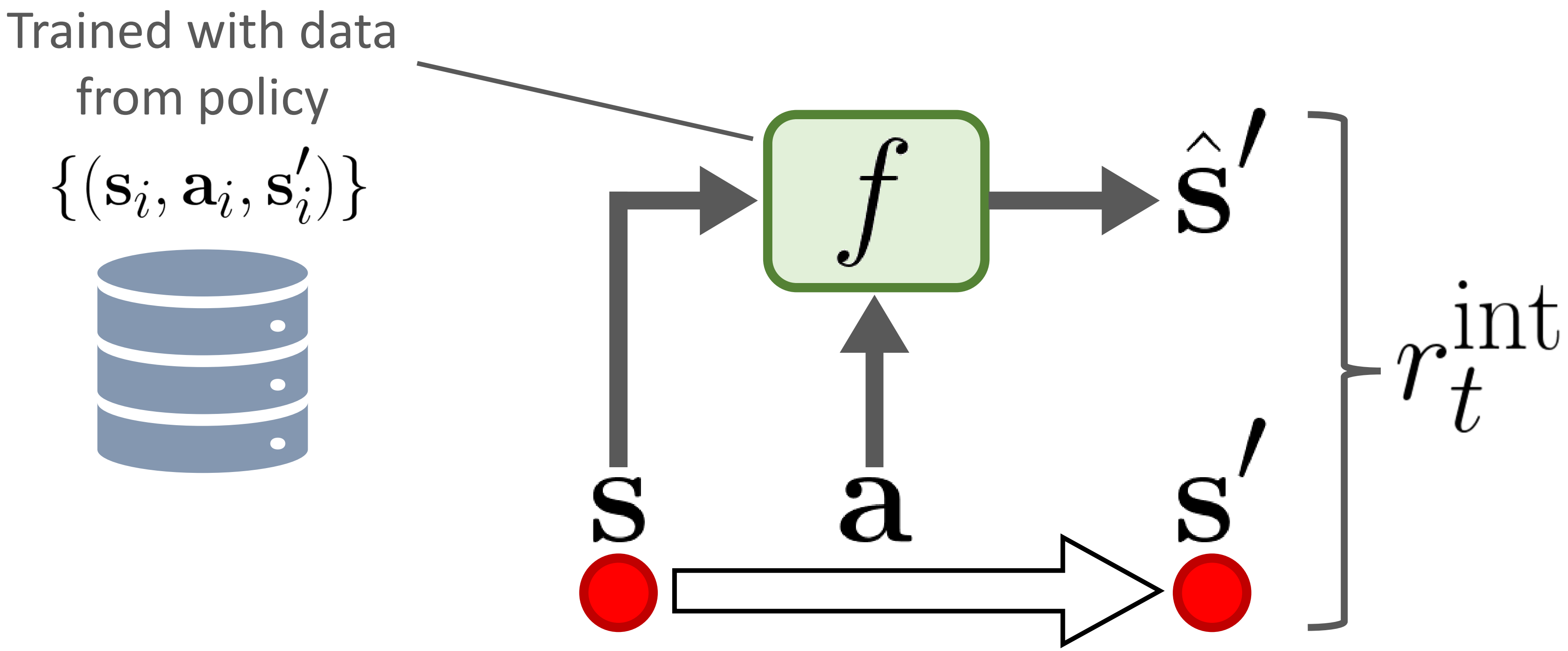
Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning
[Achiam and Sastry 2017]

Dynamics Model



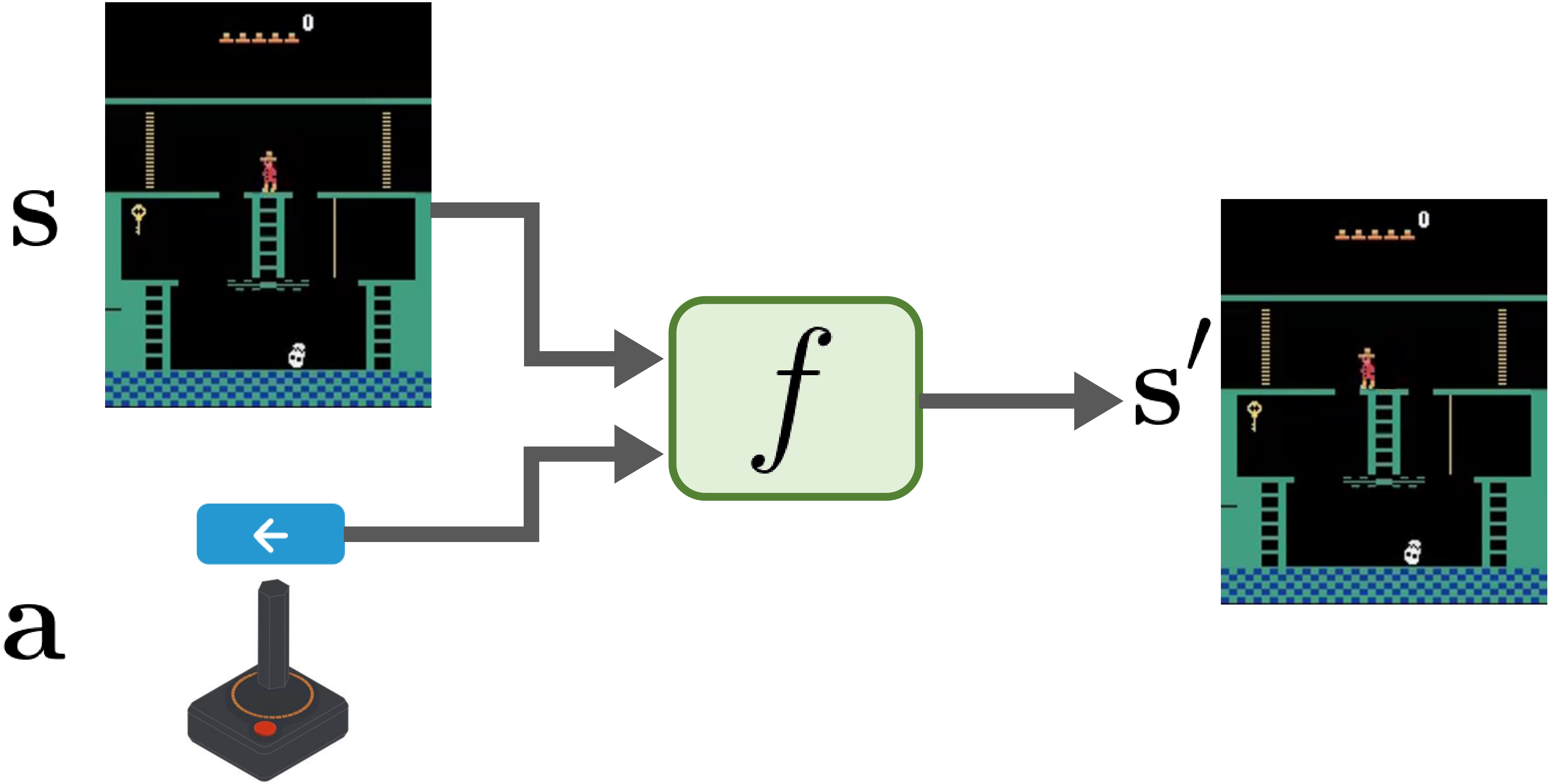
Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning
[Achiam and Sastry 2017]

Dynamics Model



Surprise-Based Intrinsic Motivation for Deep Reinforcement Learning
[Achiam and Sastry 2017]

Dynamics Model



Dynamics Models

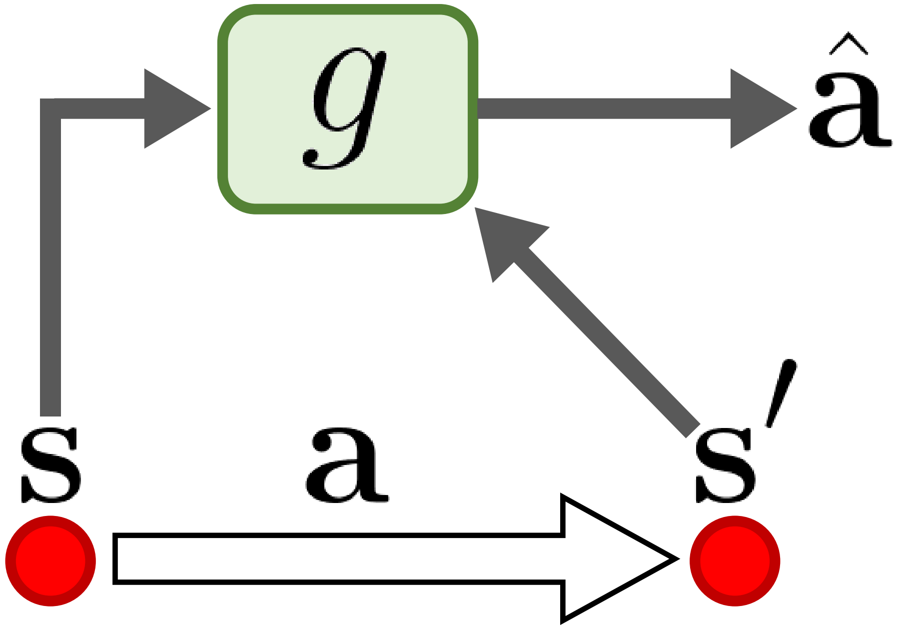
Forward dynamics model:

$$f(\mathbf{s}'|\mathbf{s}, \mathbf{a})$$

Inverse dynamics model:

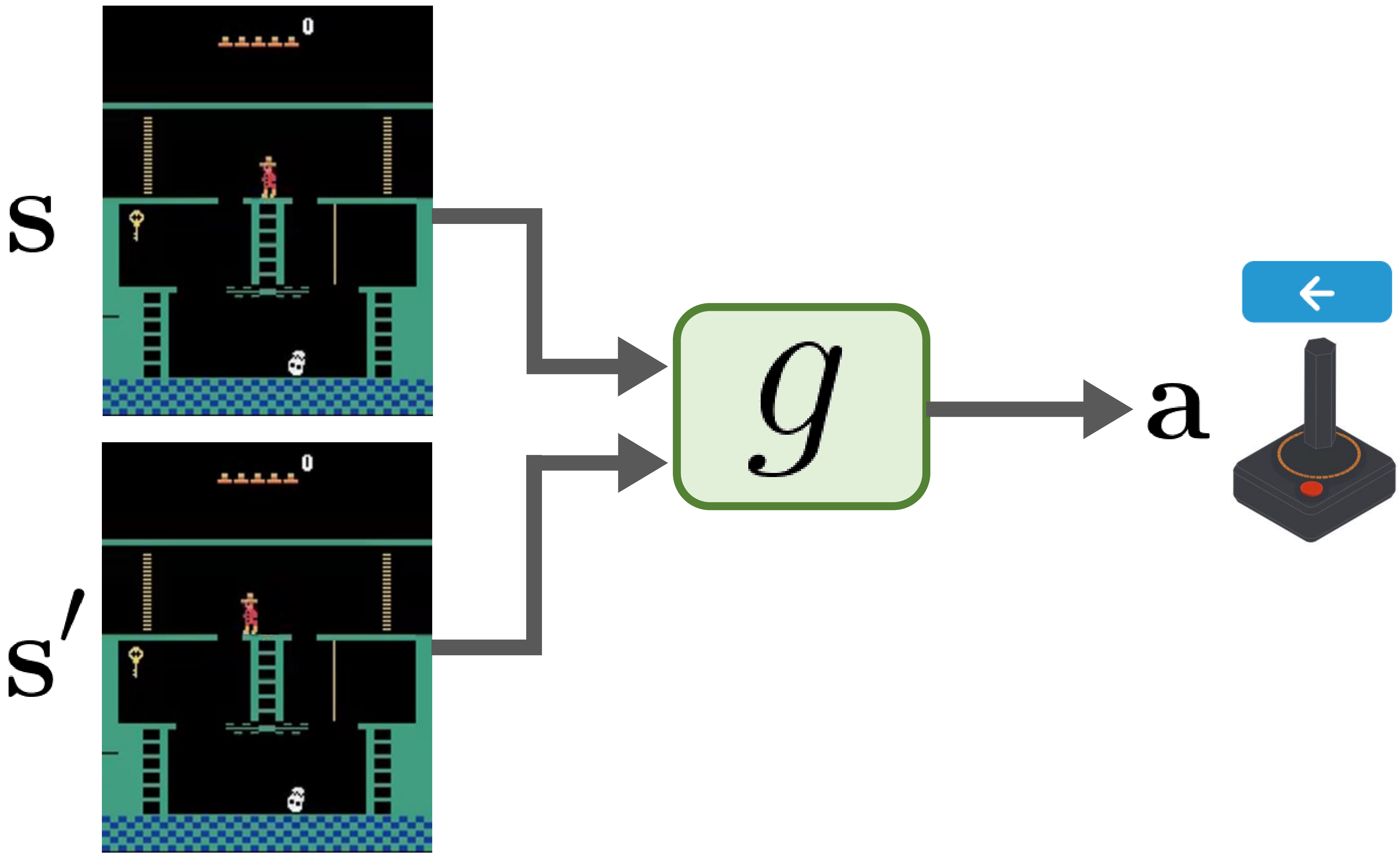
$$g(\mathbf{a}|\mathbf{s}, \mathbf{s}')$$

Inverse Dynamics Model



$$r_t^{\text{int}} = -\log g(\mathbf{a}_t | \mathbf{s}_t, \mathbf{s}_{t+1})$$

Inverse Dynamics Model

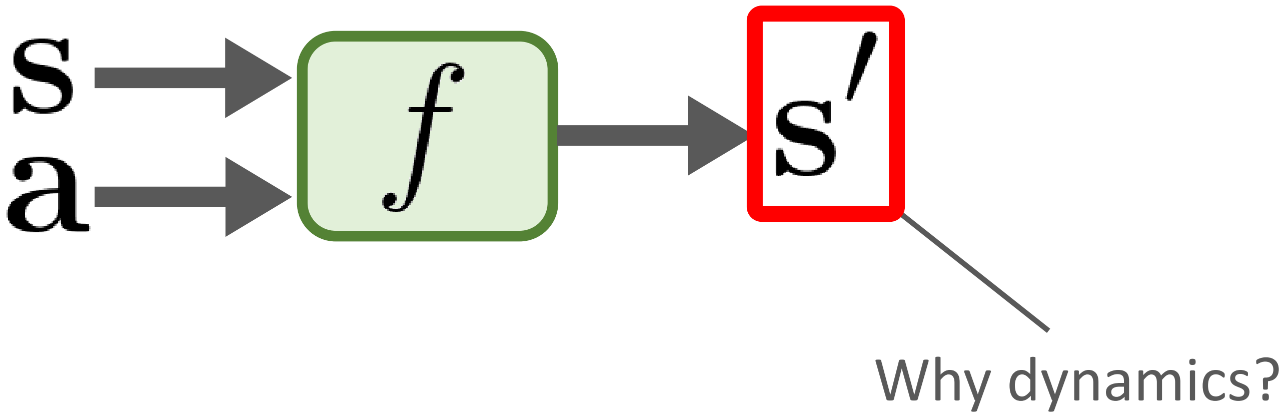


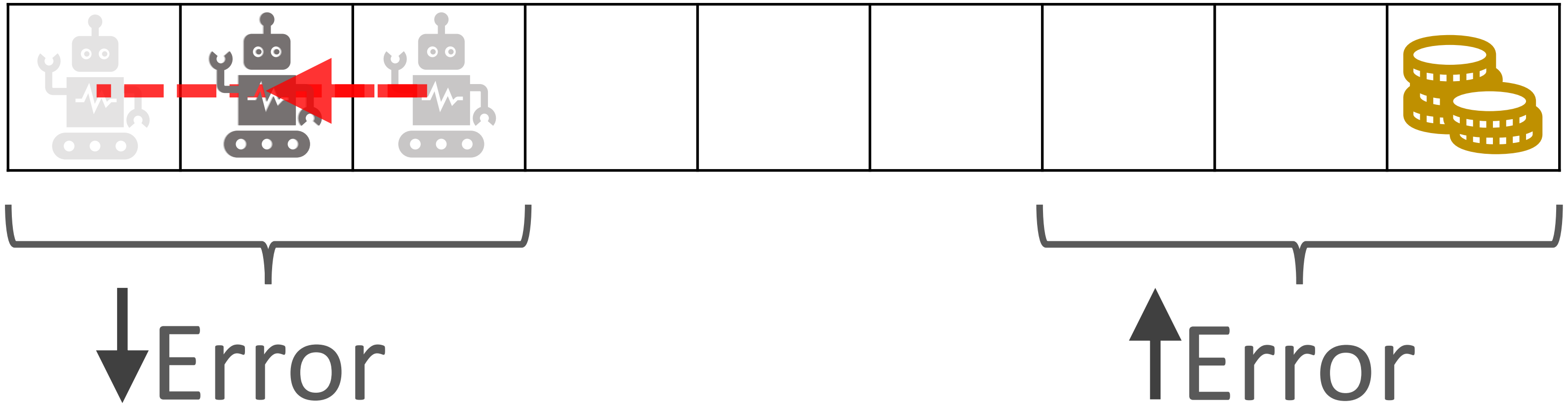
Intrinsic Curiosity Module (ICM)

No extrinsic reward!

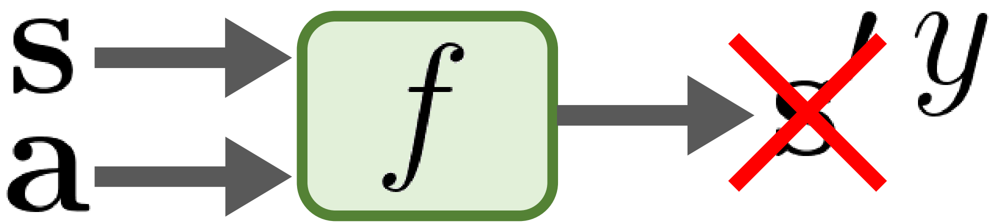
Curiosity-driven Exploration by Self-supervised Prediction
[Pathak et al. 2017]

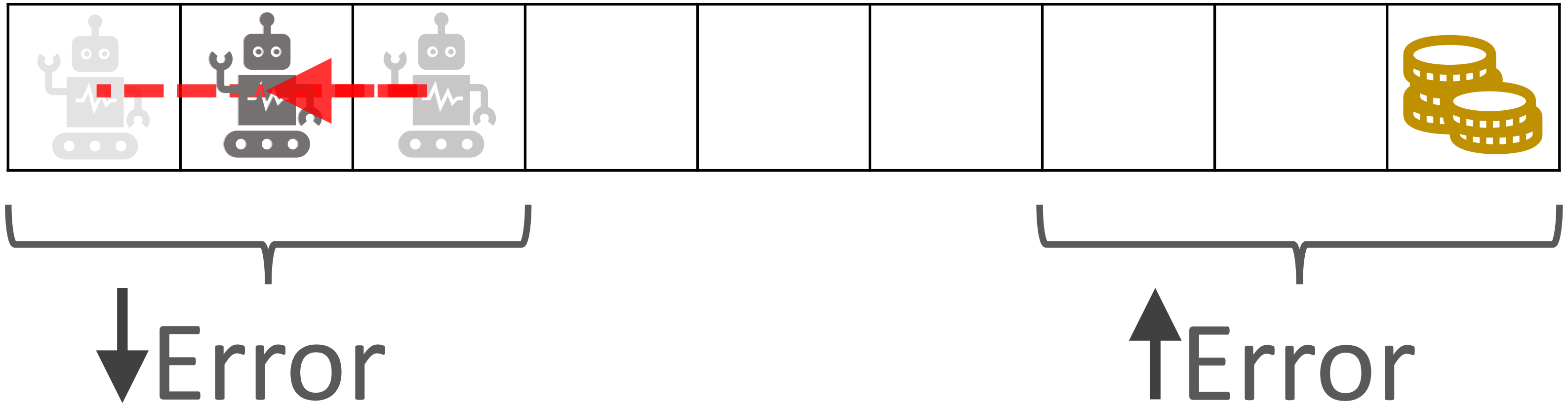
Prediction Error





Prediction Error





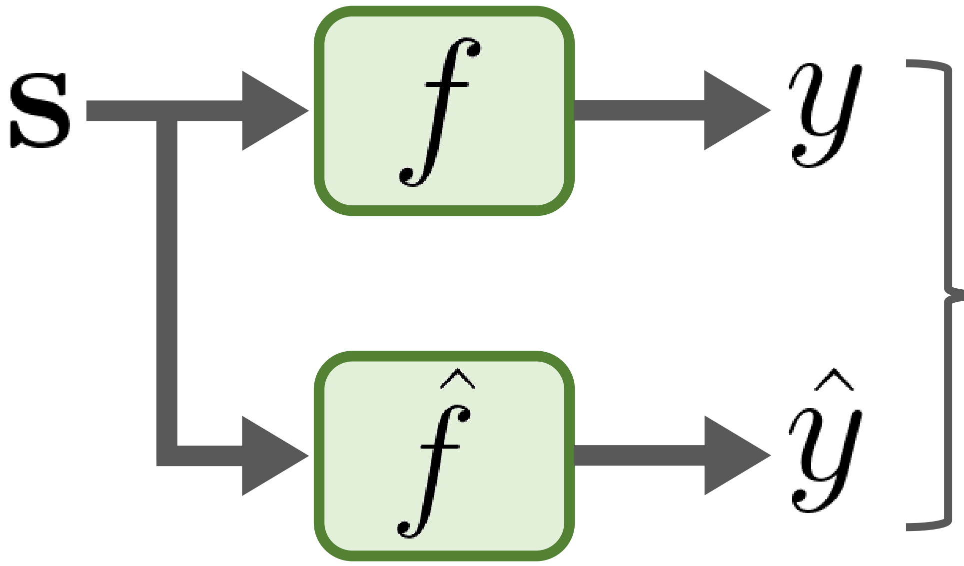
Random Network Distillation

Predictor Network f

Target Network $\hat{f}$

Exploration by Random Network Distillation
[Burda et al. 2019]

Random Network Distillation

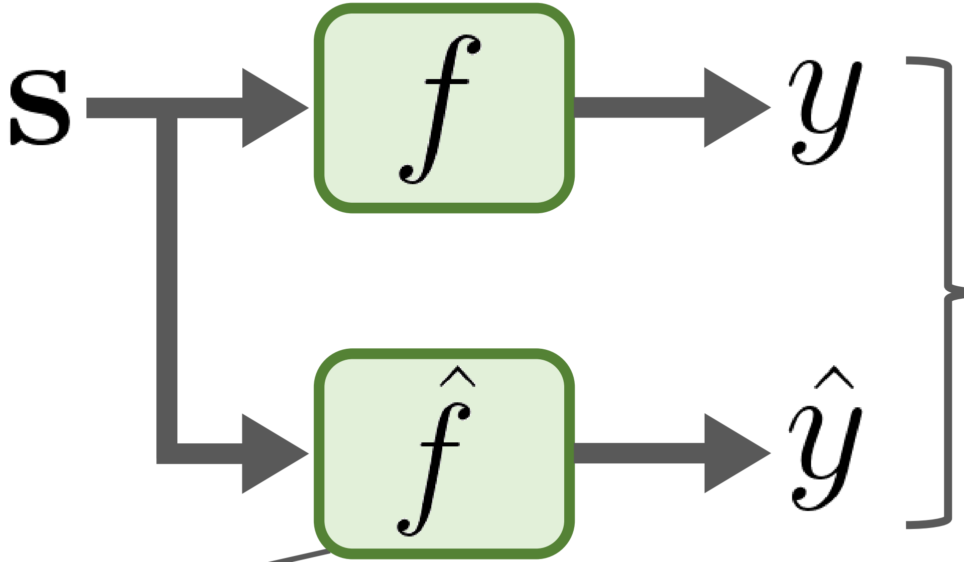


$$r_t^{\text{int}} = ||\hat{y} - y||^2$$

$$\arg\max_{f} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}} \left[||\hat{f}(\mathbf{s}) - f(\mathbf{s})||^2 \right]$$

Exploration by Random Network Distillation
[Burda et al. 2019]

Random Network Distillation



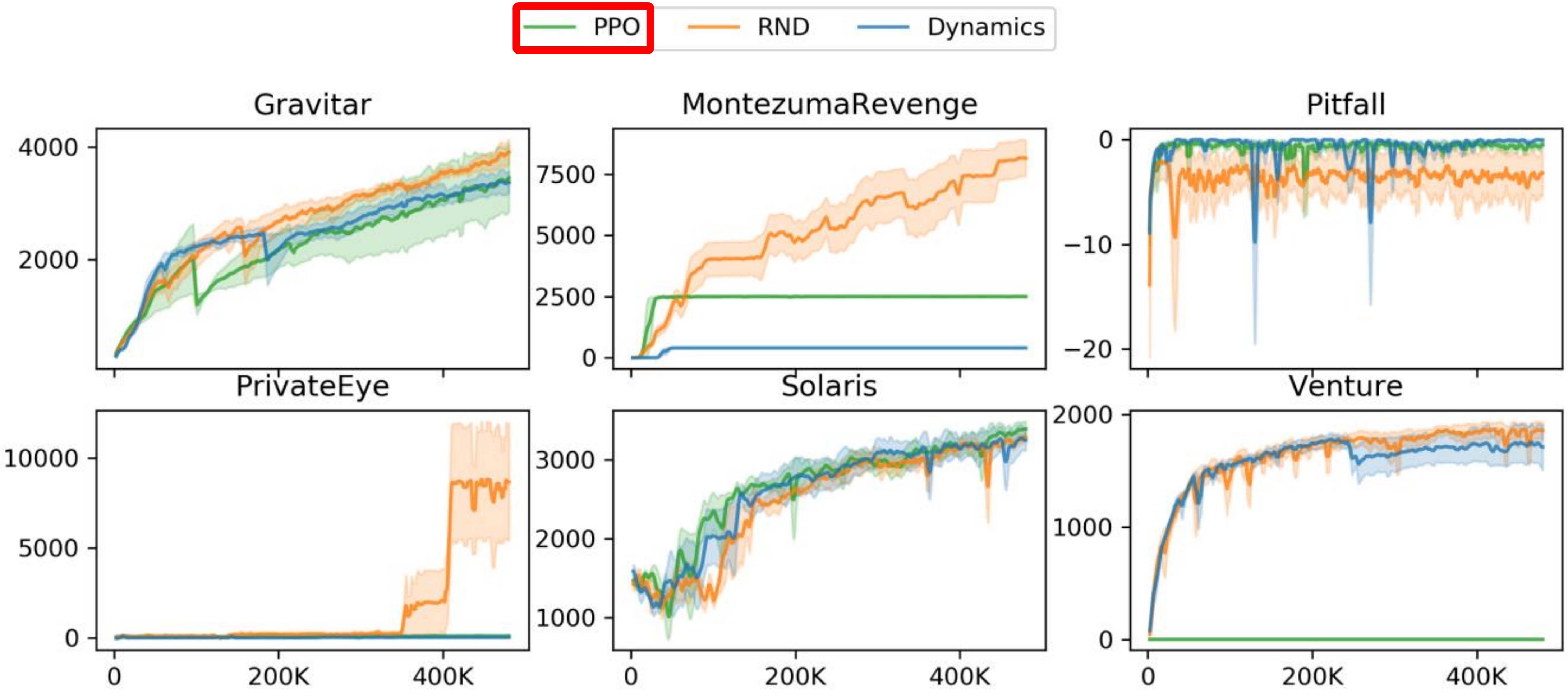
$$r_t^{\text{int}} = ||\hat{y} - y||^2$$

randomly initialized

$$\arg\max_{f} \mathbb{E}_{\mathbf{s}\sim\mathcal{D}}\left[||\hat{f}(\mathbf{s}) - f(\mathbf{s})||^2\right]$$

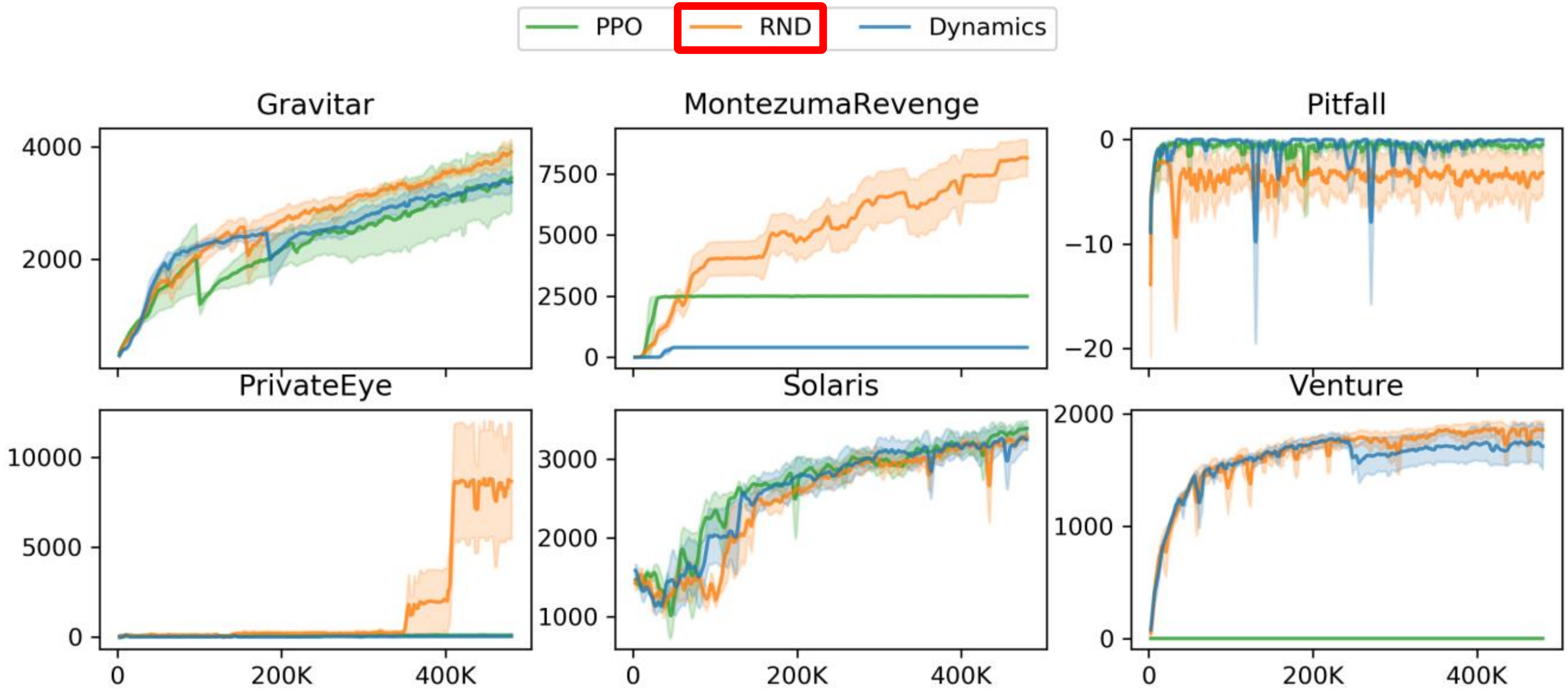
Exploration by Random Network Distillation
[Burda et al. 2019]

Random Network Distillation



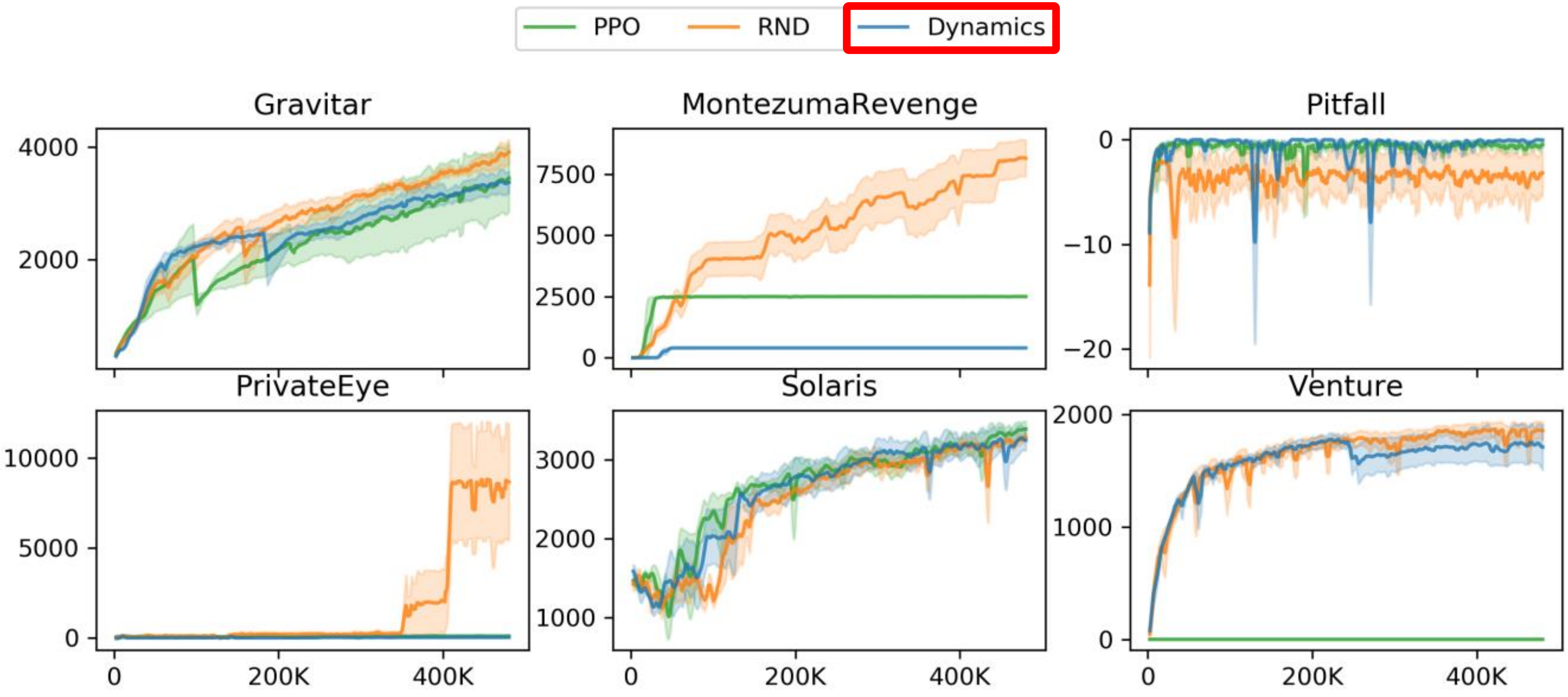
Exploration by Random Network Distillation
[Burda et al. 2019]

Random Network Distillation



Exploration by Random Network Distillation
[Burda et al. 2019]

Random Network Distillation



Exploration by Random Network Distillation
[Burda et al. 2019]

Novelty

Prediction Error $\longrightarrow$ Novelty

Is this always true?

Novelty

No!

Prediction Error → Novelty

Is this always true?

Novelty

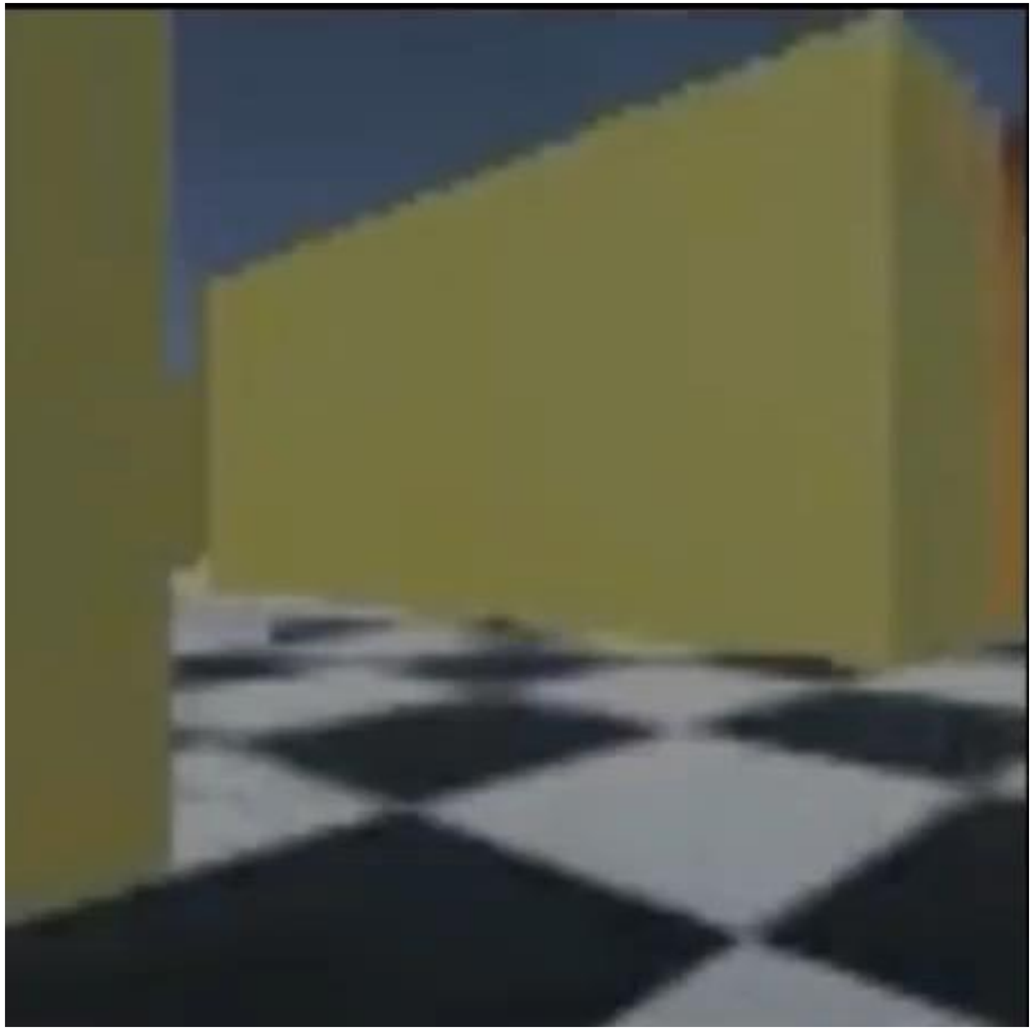
No!

Prediction Error $\not\rightarrow$ Novelty

Is this always true?

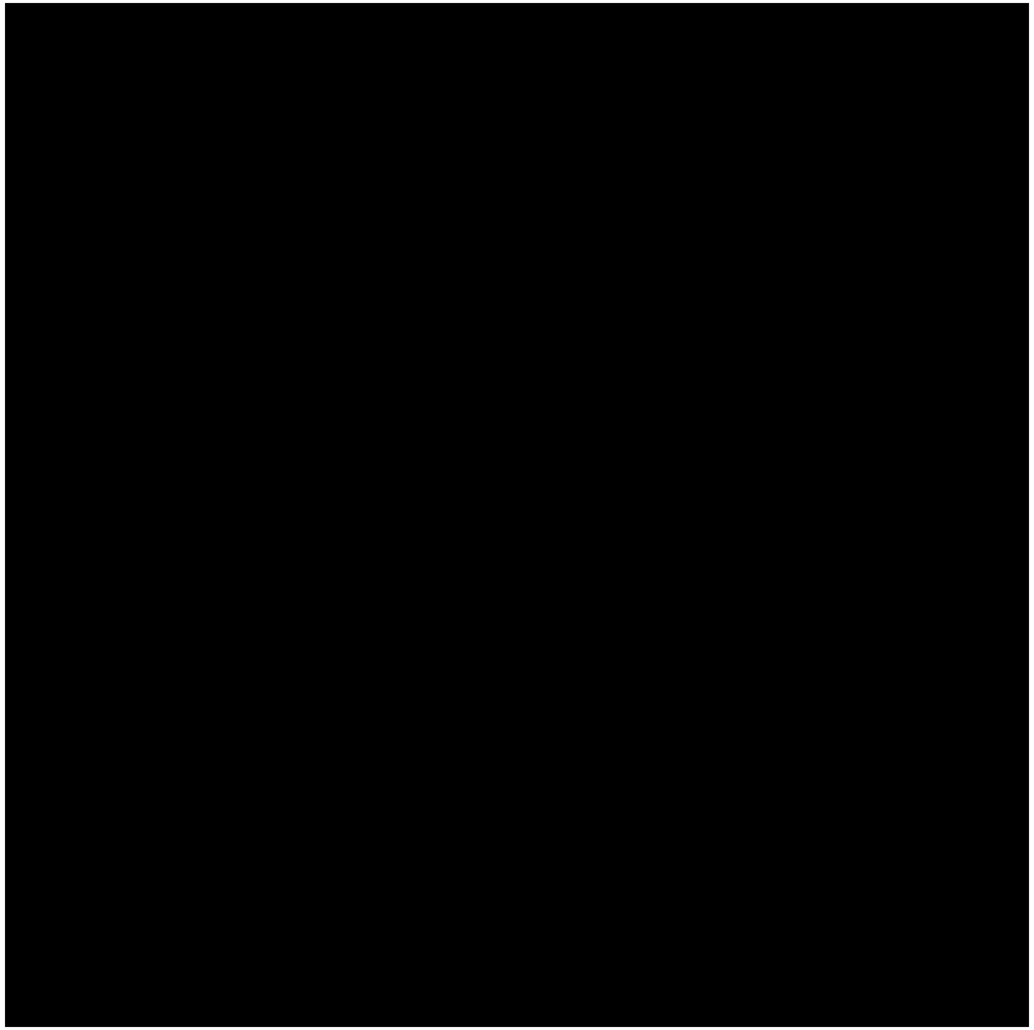
Unpredictable $\neq$ Novel

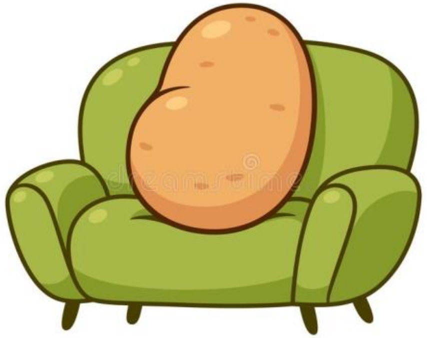
Noisy-TV Problem



Large-Scale Study of Curiosity-Driven Learning
[Burda et al. 2018]

Noisy-TV Problem





- Prediction error works well in <u>*static*</u> envs
- But can get distracted by variability in <u>*dynamic*</u> envs

Large-Scale Study of Curiosity-Driven Learning
[Burda et al. 2018]

Surprise Minimization

Maximize Surprise

$$r_t^{\text{int}} = -\log\left(\rho(\mathbf{s}_{t+1})\right)$$

$$\Downarrow$$

Minimize Surprise

$$r_t^{\text{int}} = \log\left(\rho(\mathbf{s}_{t+1})\right)$$

SMiRL: Surprise Minimizing Reinforcement Learning in Dynamic Environments
[Berseth et al. 2020]

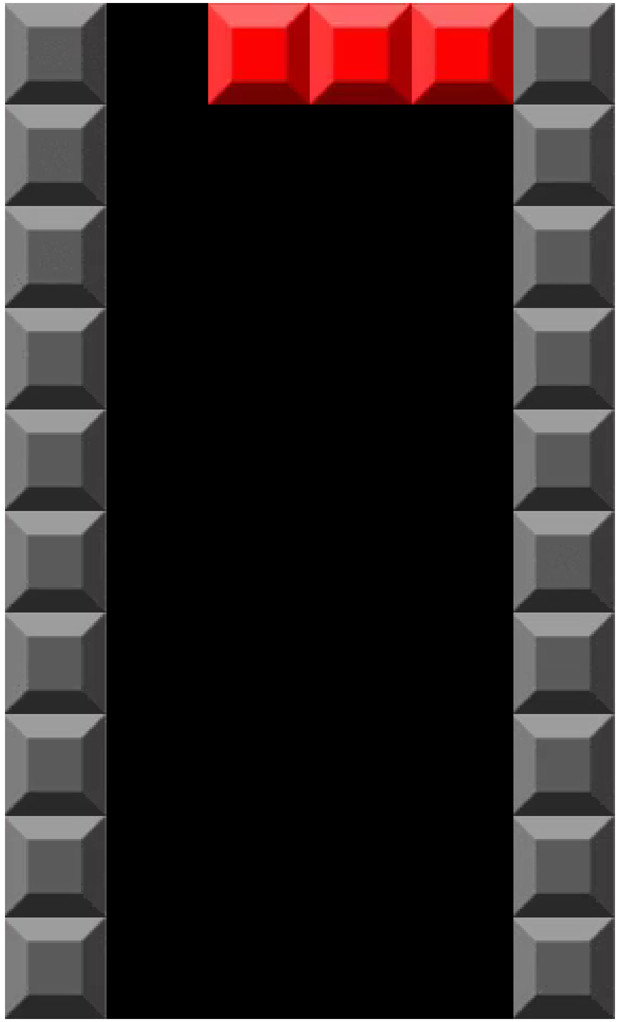
Surprise Minimization

Maximize Surprise

$$r_t^{\text{int}} = -\log\left(\rho(\mathbf{s}_{t+1})\right)$$

Minimize Surprise

$$r_t^{\text{int}} = \log\left(\rho(\mathbf{s}_{t+1})\right)$$



SMiRL: Surprise Minimizing Reinforcement Learning in Dynamic Environments [Berseth et al. 2020]

Summary

- Exploration Exploitation Tradeoff
- Dense vs Sparse Rewards
- Intrinsic Motivation
- Count-Based Exploration
- Surprise Maximization