Exploration vs. Exploitation

Q-learning

Алгоритм

Q-learning

Алгоритм

1. Инициализируем $Q(s, a) = 0$ для всех s, a

Q-learning

Алгоритм

1. Инициализируем $Q(s, a) = 0$ для всех s, a

2. Взаимодействуем со средой:

   - $a^* = \arg\max_a Q(s, a)$ выбираем действие

Q-learning

Алгоритм

1. Инициализируем $Q(s, a) = 0$ для всех s, a

2. Взаимодействуем со средой:
   - $a^* = \arg\max_a Q(s, a)$ выбираем действие
   - $s', r = \text{СРЕДА}(s, a^*)$ применяем его в среде

Q-learning

Алгоритм

1. Инициализируем $Q(s,a) = 0$ для всех s, a

2. Взаимодействуем со средой:
   - $a^* = \arg\max_a Q(s,a)$ выбираем действие
   - $s', r = \text{СРЕДА}(s, a^*)$ применяем его в среде
   - $Q(s,a^*) := r + \gamma \max_{a'} Q(s', a')$ обновляем **Q**-функцию

Q-learning

Алгоритм

1. Инициализируем $Q(s,a) = 0$ для всех s, a

2. Взаимодействуем со средой:

- $a^* = \arg\max_a Q(s,a)$ выбираем действие

- $s', r = \text{СРЕДА}(s, a^*)$ применяем его в среде

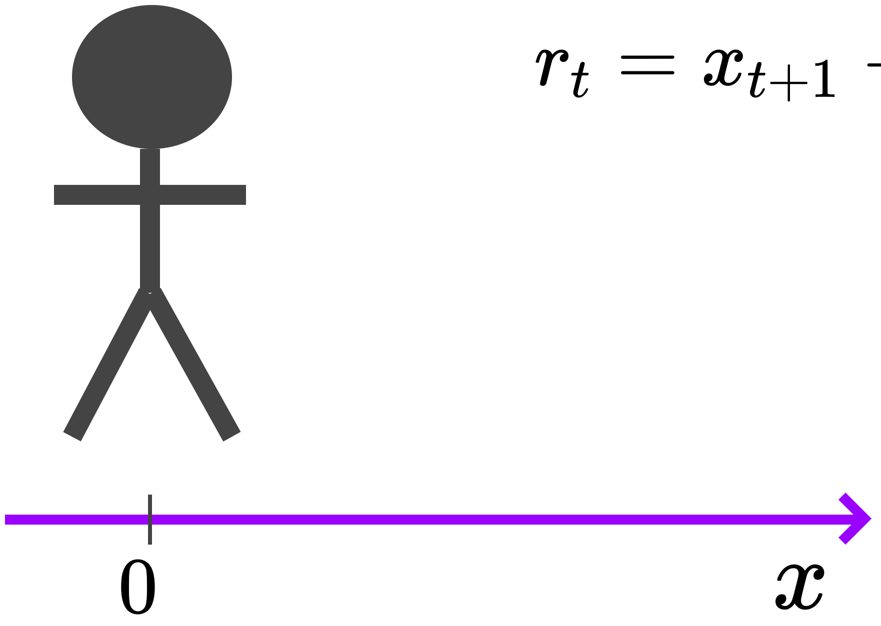
- $Q(s,a^*) := r + \gamma \max_{a'} Q(s',a')$ обновляем **Q**-функцию

- $s \leftarrow s'$

Exploration vs. Exploitation dilemma

ϵ-жадная стратегия

Давайте научим робота идти вперед



$$r_t = x_{t+1} - x_t$$

Оценка Q-функции:

$$Q(s_0, a = \text{УПАСТЬ}) = 0$$

$$Q(s_0, a = \text{ШАГНУТЬ}) = 0$$

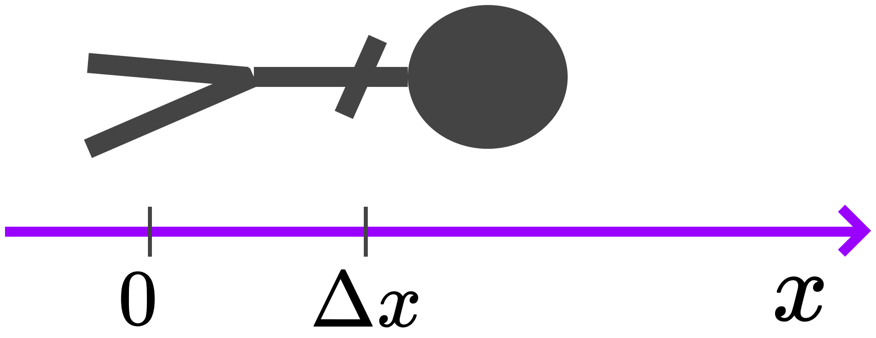
$$\arg\max_a Q(s_0, a) = \text{УПАСТЬ}$$

Exploration vs. Exploitation dilemma

ϵ-жадная стратегия

Давайте научим робота идти вперед

$$r_t = x_{t+1} - x_t$$



Оценка Q-функции:

$$Q(s_0, a = \text{УПАСТЬ}) = 0$$

$$Q(s_0, a = \text{ШАГНУТЬ}) = 0$$

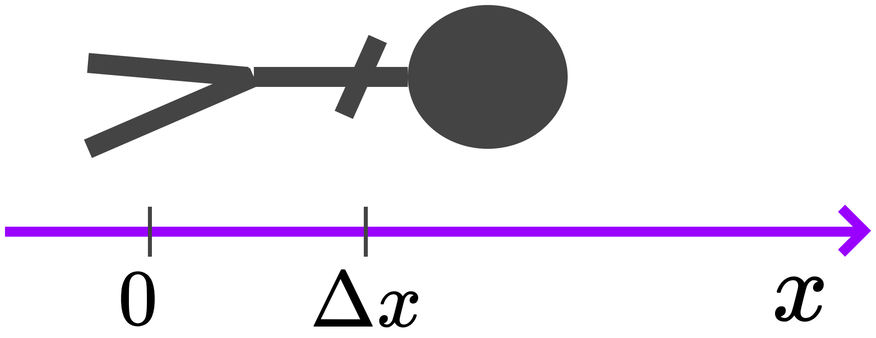
$$\arg\max_a Q(s_0, a) = \text{УПАСТЬ}$$

Exploration vs. Exploitation dilemma

ϵ-жадная стратегия

Давайте научим робота идти вперед

$$r_t = x_{t+1} - x_t$$



Оценка Q-функции:

$$Q(s_0, a = \text{УПАСТЬ}) = \Delta x$$

$$Q(s_0, a = \text{ШАГНУТЬ}) = 0$$

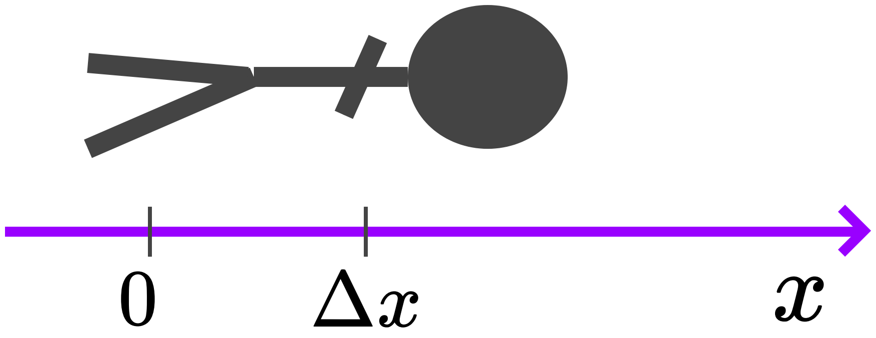
$$\arg\max_a Q(s_0, a) = \text{УПАСТЬ}$$

Exploration vs. Exploitation dilemma

ϵ-жадная стратегия

Давайте научим робота идти вперед

$$r_t = x_{t+1} - x_t$$



Решение: с вероятностью ϵ делаем случайное действие
Иначе, жадное

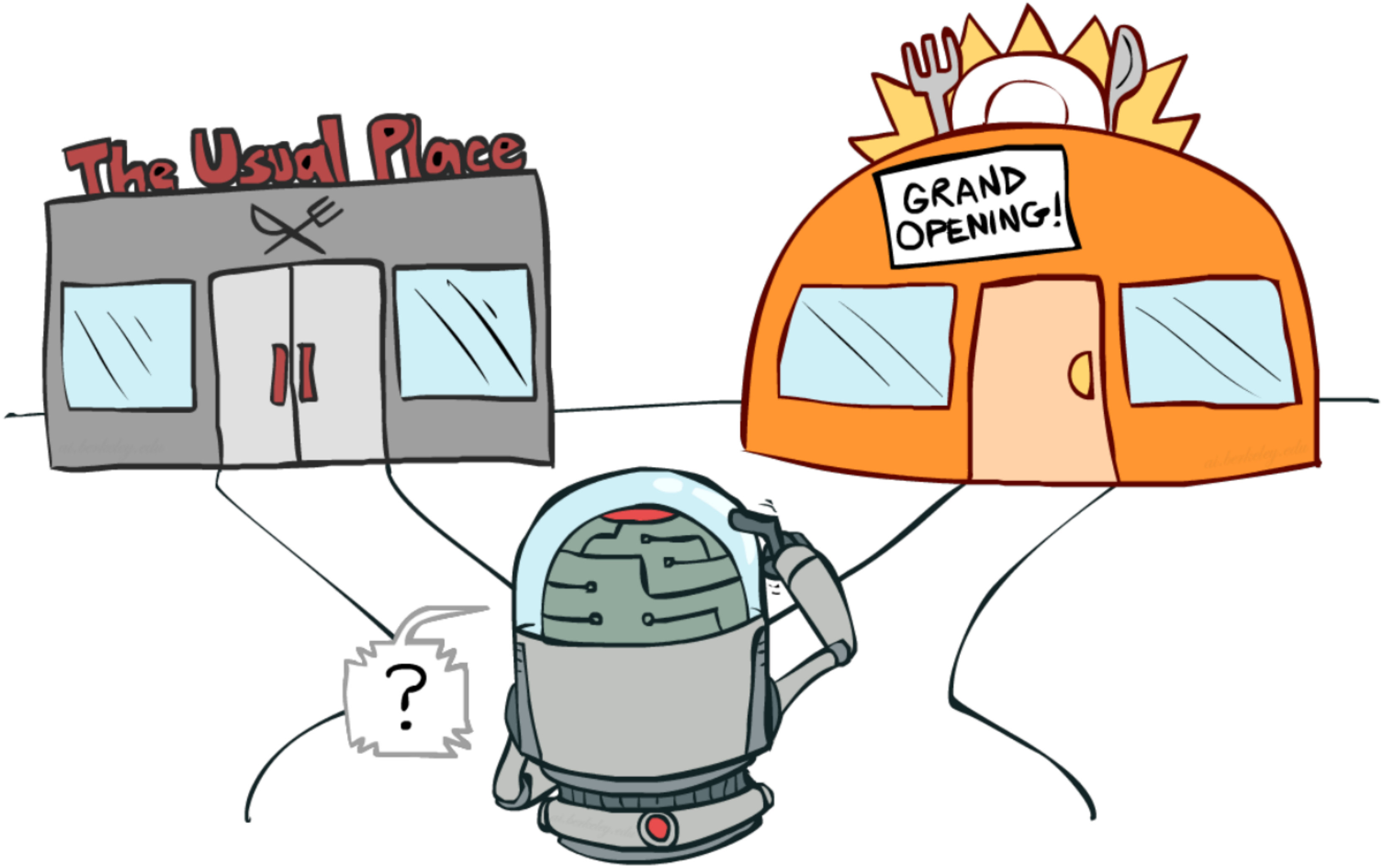
Оценка Q-функции:

$$Q(s_0, a = \text{УПАСТЬ}) = \Delta x$$

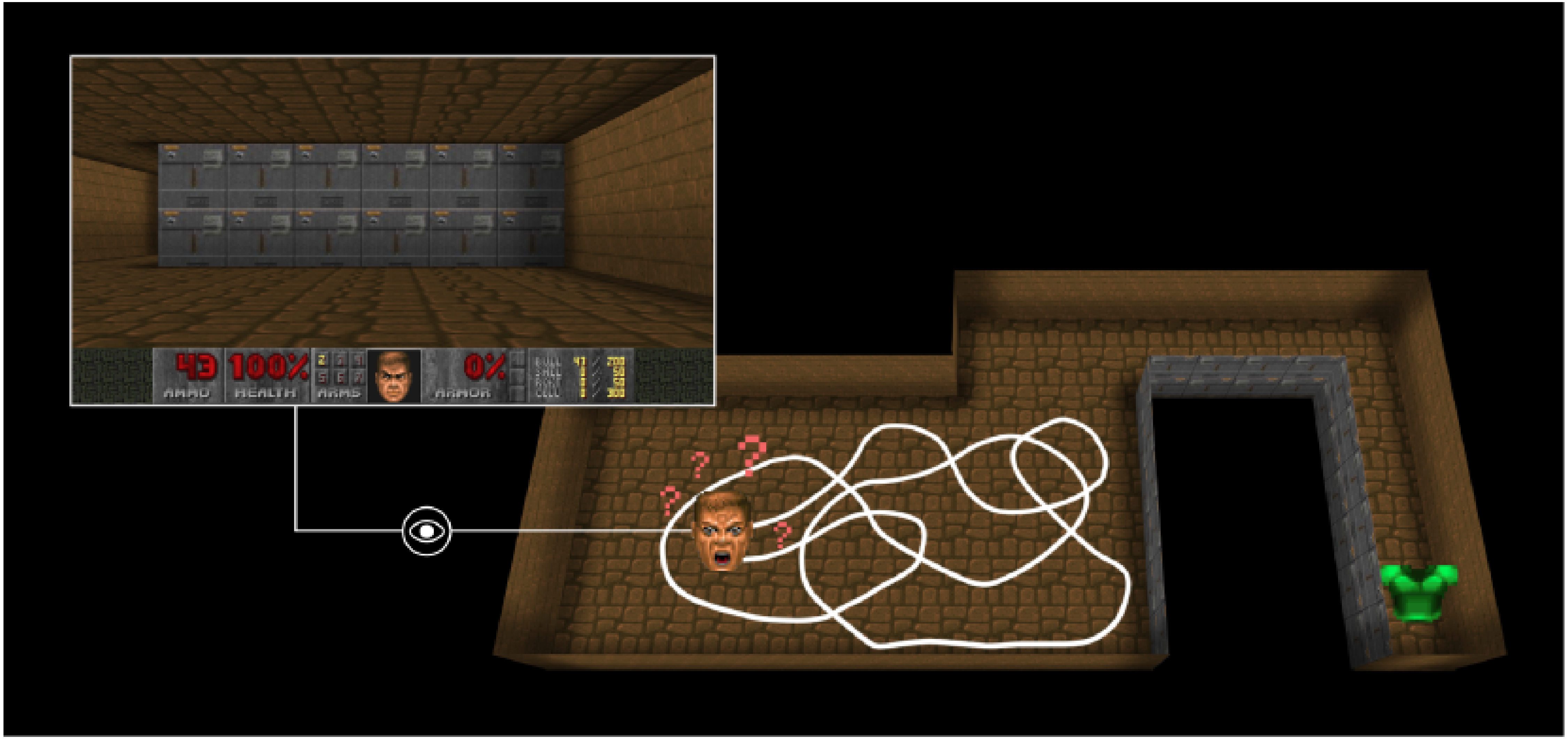
$$Q(s_0, a = \text{ШАГНУТЬ}) = 0$$

$$\arg\max_a Q(s_0, a) = \text{УПАСТЬ}$$

Exploration vs. Exploitation dilemma



Exploration is hard



[1]picture source

Self-Supervision in Reinforcement Learning

Markov Decision Process

- $\mathcal{S}$ — set of states
- $\mathcal{A}$ — set of actions
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ — transitions
- $\mathcal{R}$ — reward function

Self-Supervision in Reinforcement Learning

Markov Decision Process

- $\mathcal{S}$ — set of states
- $\mathcal{A}$ — set of actions
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ — transitions
- $\mathcal{R}$ — **reward function**

Self-Supervision in Reinforcement Learning

Markov Decision Process

- $\mathcal{S}$ — set of states
- $\mathcal{A}$ — set of actions
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ — transitions
- $\mathcal{R}^{\text{extr}}$ — **extrinsic reward function**

Extrinsic Motivation
(the task we want to solve)

Auxiliary task

- $\mathcal{S}$ — set of states
- $\mathcal{A}$ — set of actions
- $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{P}(\mathcal{S})$ — transitions
- $\mathcal{R}^{\text{intr}}$ — **intrinsic reward function**

Intrinsic Motivation
("self-supervised")

Difference between motivations

Extrinsic motivation:

- cakes, pain, life goals

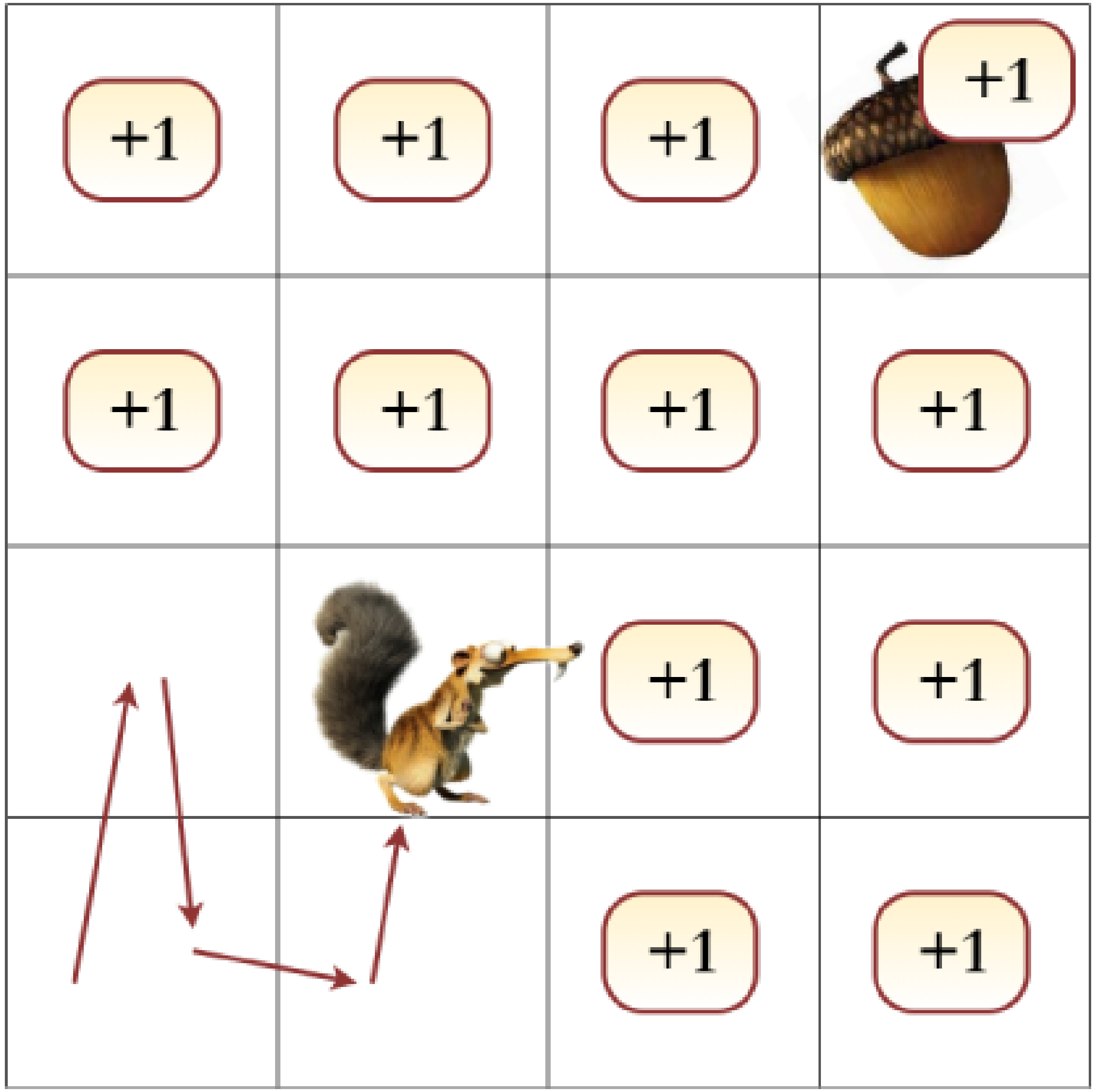
«(provided by environment)»

Intrinsic motivation:

- joy, fun, curiosity

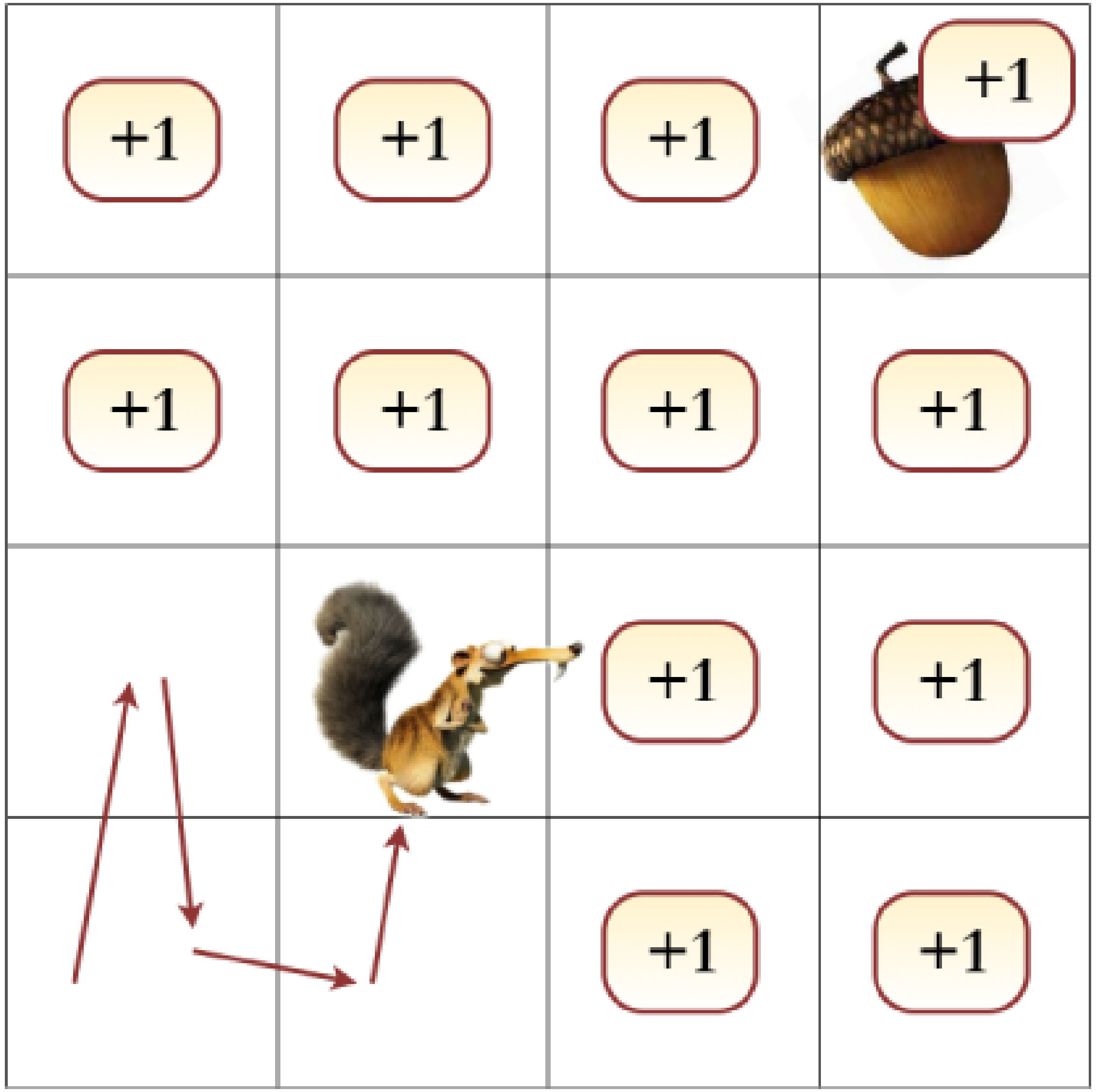
«(helps developing broad set of skills)»

Exploration Bonuses



Basic idea: $+1$ for visiting new state

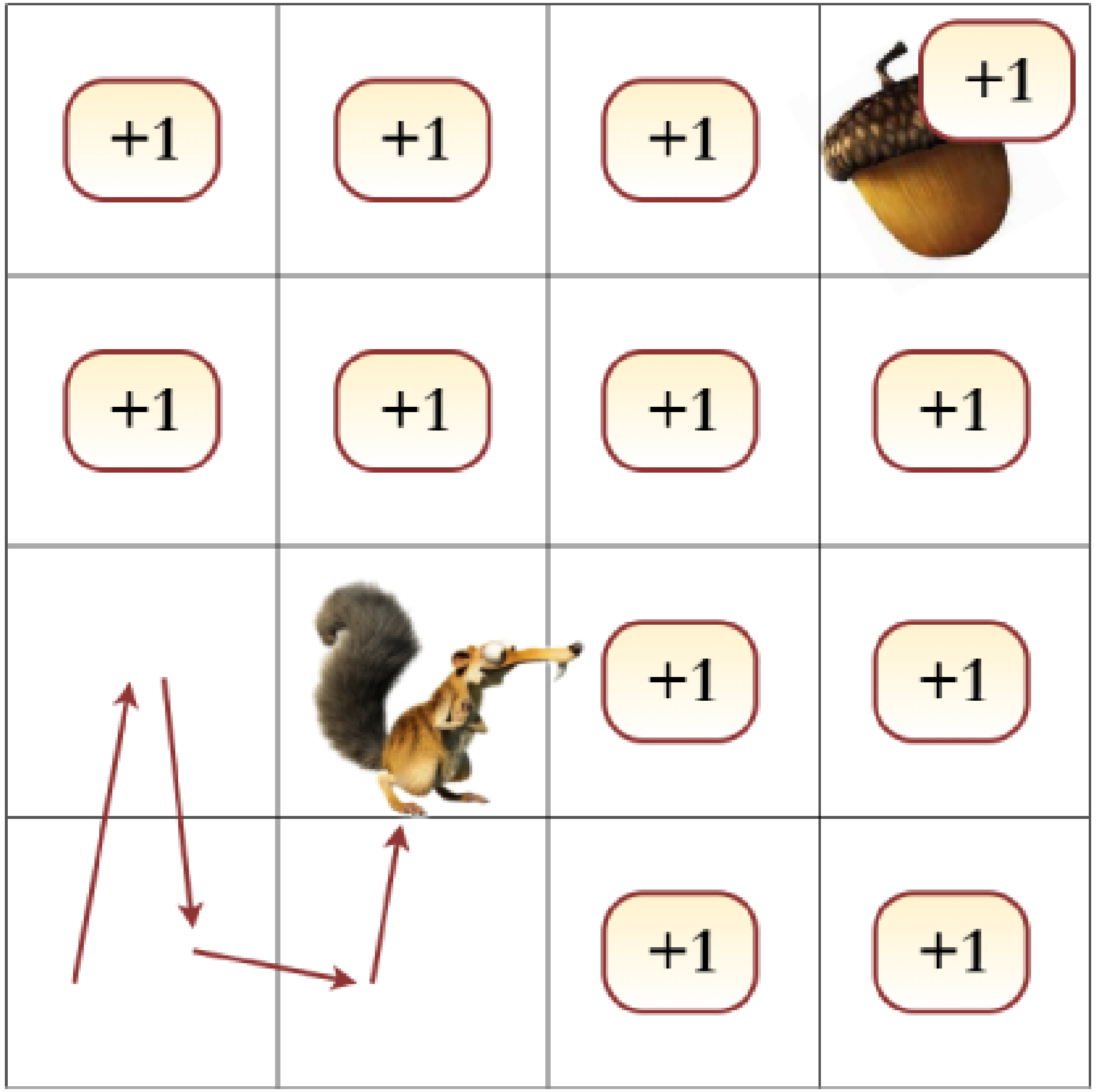
Exploration Bonuses



Basic idea: $+1$ for visiting new state

a) episodic (respawns each episode)

Exploration Bonuses

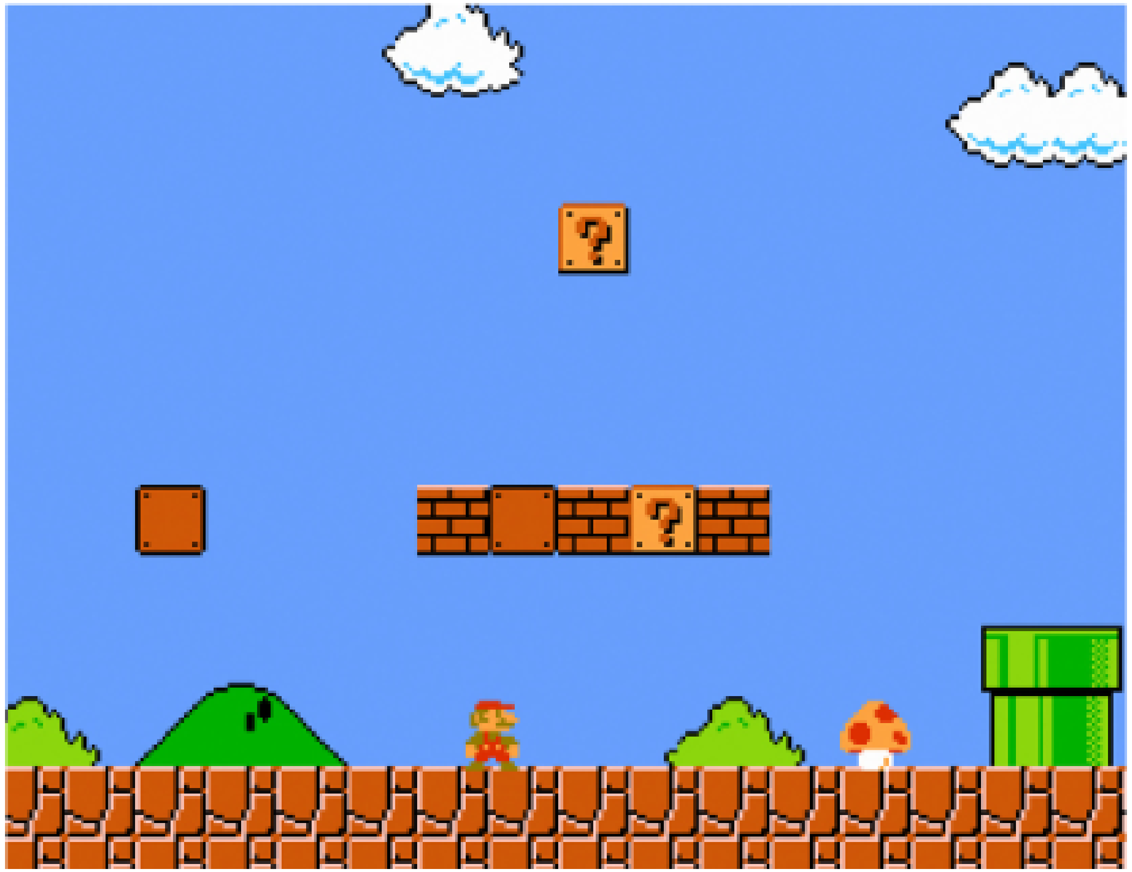


Basic idea: $+1$ for visiting new state

a) episodic (respawns each episode)

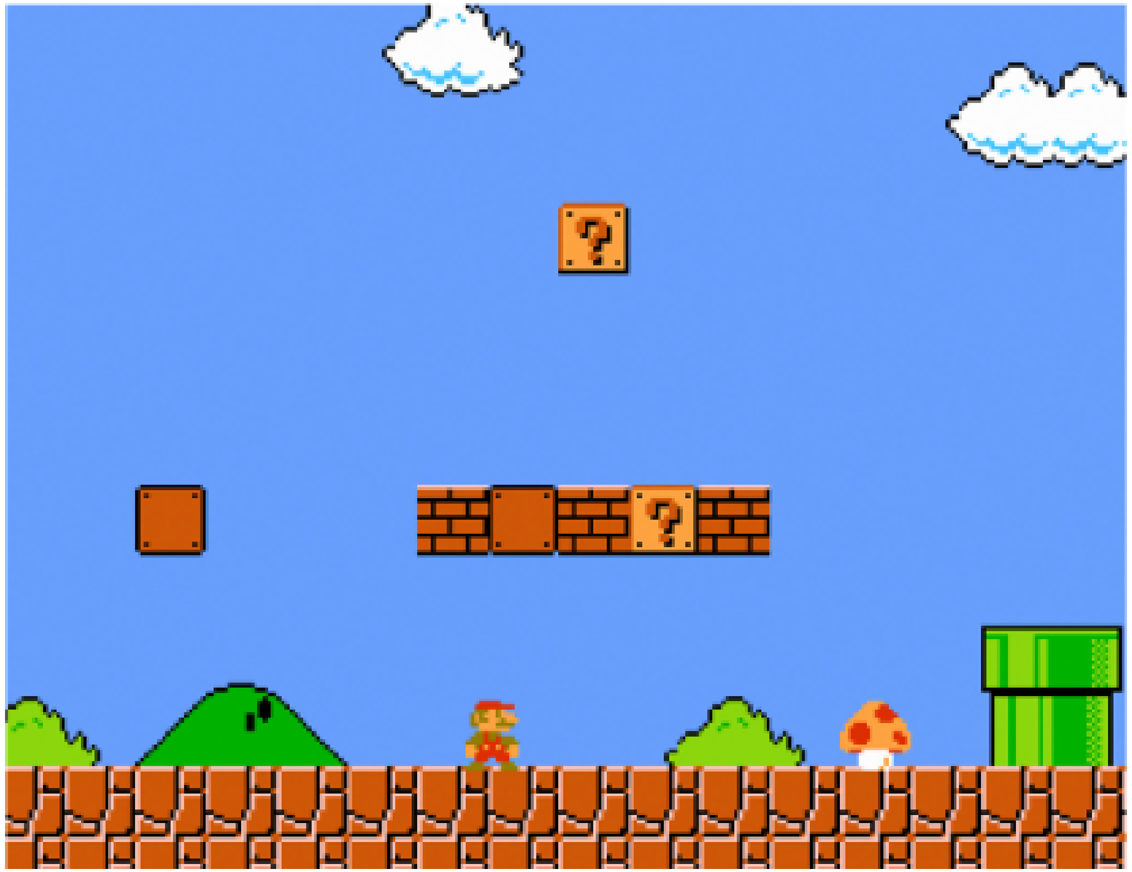
b) non-stationary: $+\frac{1}{n(s)}$, where $n(s)$ is a number of state visitations (during training!)

Mario Oracle



Huge bonus for your x-coordinate!
(positive reward when going right,
negative reward when going left)

Mario Oracle



Huge bonus for your x-coordinate!
(positive reward when going right,
negative reward when going left)

Please, go right, Mario!

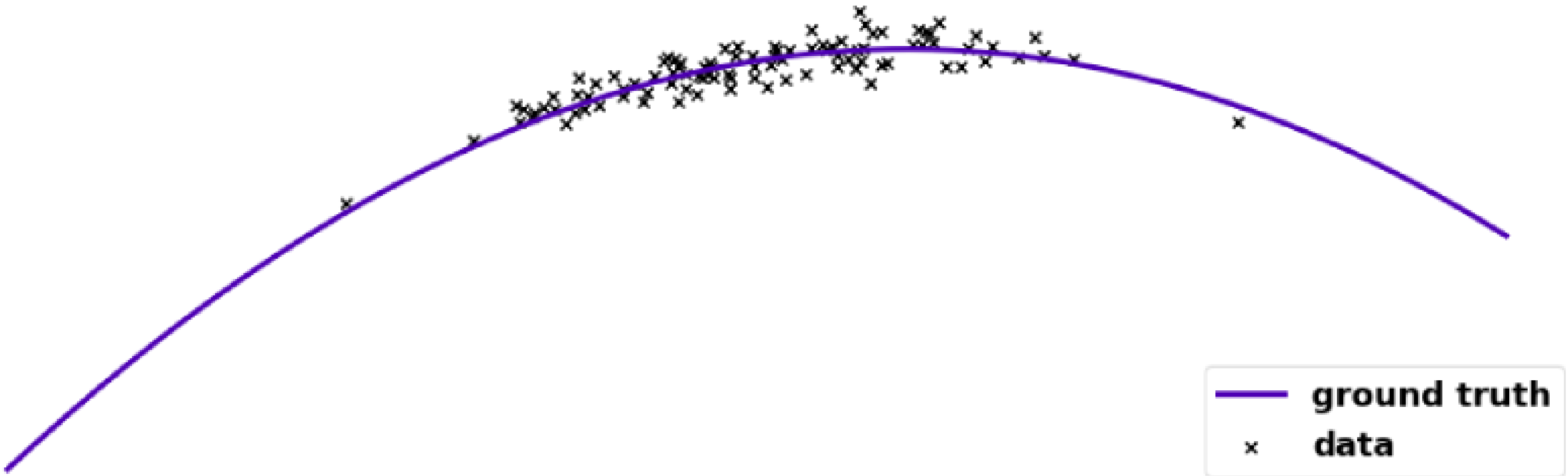
Novelty Detection

How to estimate novelty of states?

Novelty Detection

How to estimate novelty of states?

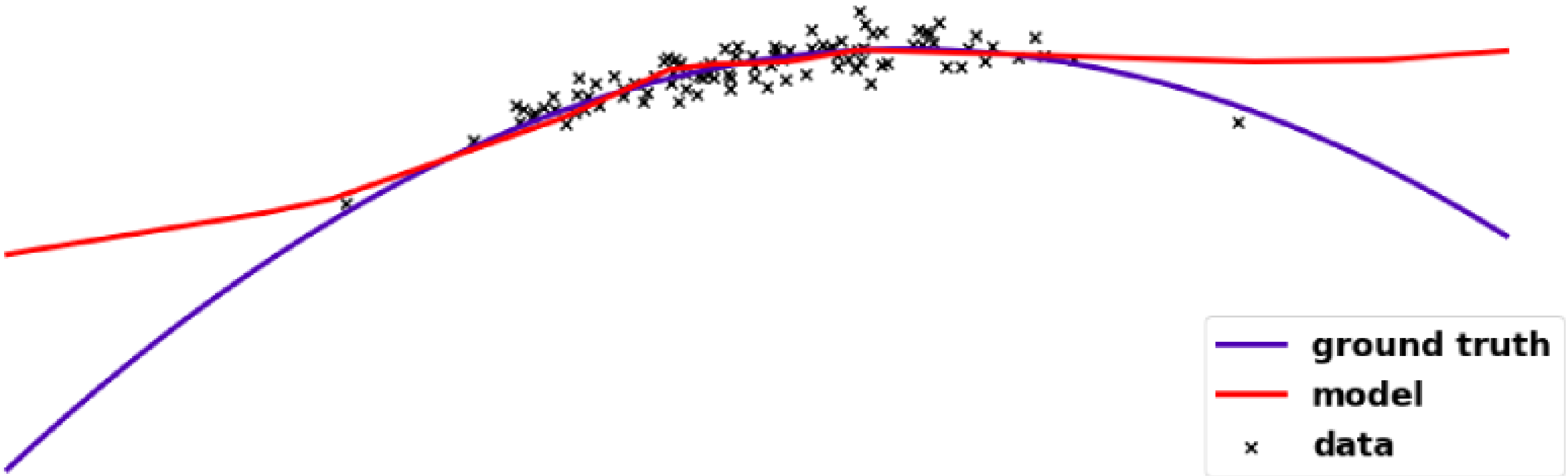
- solve some regression task with states as input



Novelty Detection

How to estimate novelty of states?

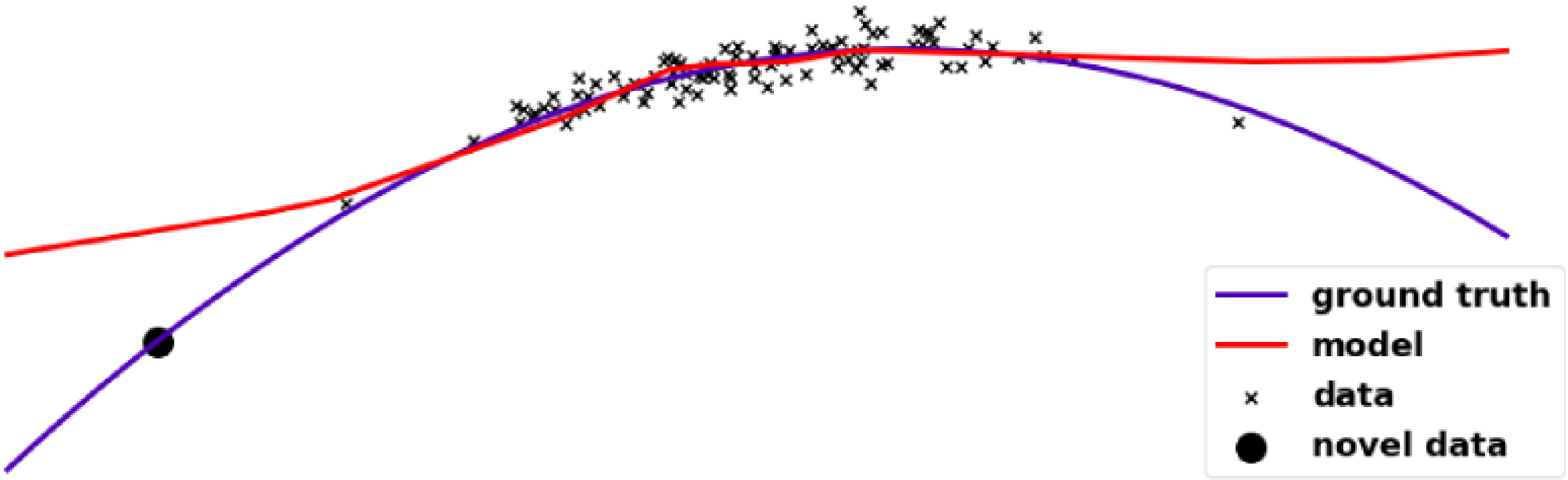
- solve some regression task with states as input



Novelty Detection

How to estimate novelty of states?

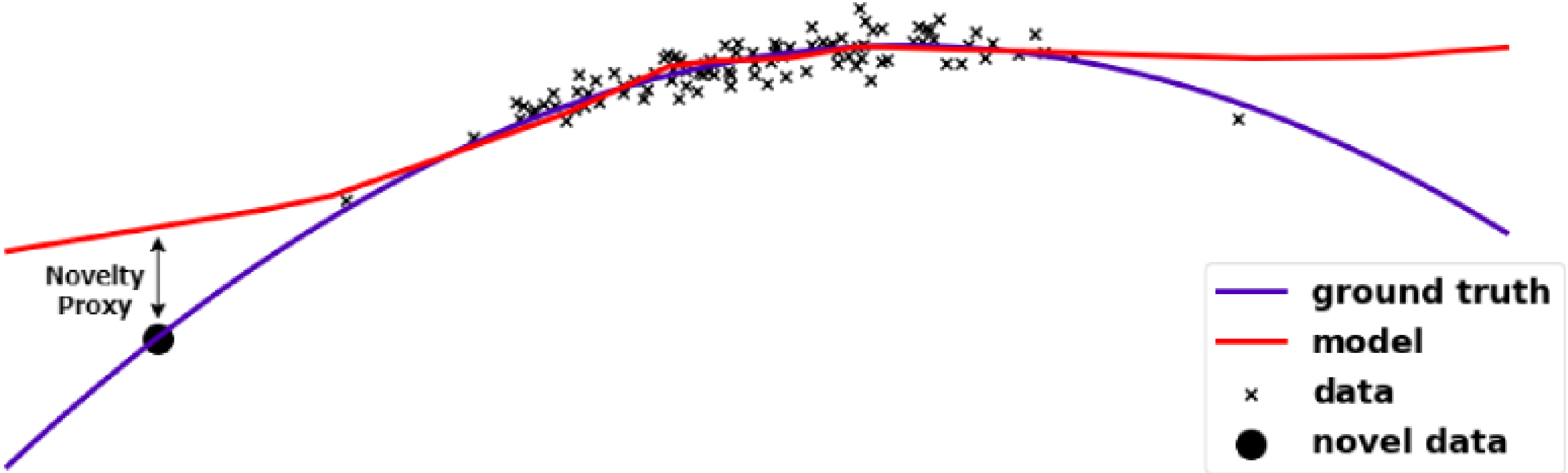
- solve some regression task with states as input
- the error on some state $s \in \mathcal{S}$ is a proxy of novelty!



Novelty Detection

How to estimate novelty of states?

- solve some regression task with states as input
- the error on some state $s \in \mathcal{S}$ is a proxy of novelty!



Why prediction error can be high?

✓ <u>**data novelty**</u>

Why prediction error can be high?

✓ **data novelty**

× **stochastic targets**: if target is stochastic function of input, error of deterministic model will not be zero.

Why prediction error can be high?

✓ **data novelty**

× **stochastic targets**: if target is stochastic function of input, error of deterministic model will not be zero.

× **non-stationary targets**: if target changes with time, prediction error will spike proportional to changes.

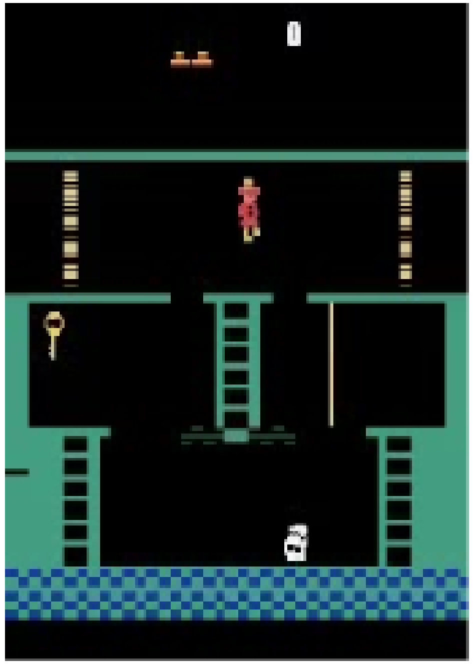
Why prediction error can be high?

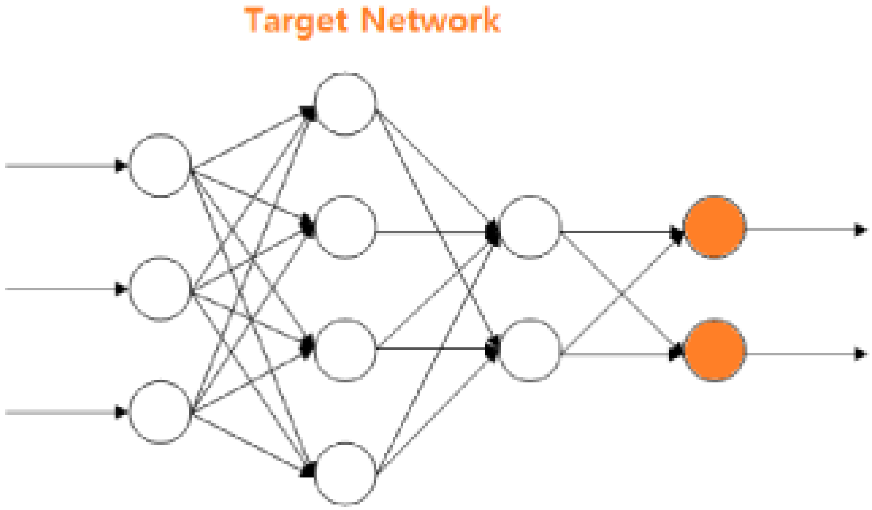
- ✓ **data novelty**
- × **stochastic targets**: if target is stochastic function of input, error of deterministic model will not be zero.
- × **non-stationary targets**: if target changes with time, prediction error will spike proportional to changes.
- × **task is too complex**: if your model is not rich enough or lacks some necessary input, error will not be zero.

Why prediction error can be high?

- ✓ **data novelty**
- × **stochastic targets**: if target is stochastic function of input, error of deterministic model will not be zero.
- × **non-stationary targets**: if target changes with time, prediction error will spike proportional to changes.
- × **task is too complex**: if your model is not rich enough or lacks some necessary input, error will not be zero.
- × **imperfect optimizer**

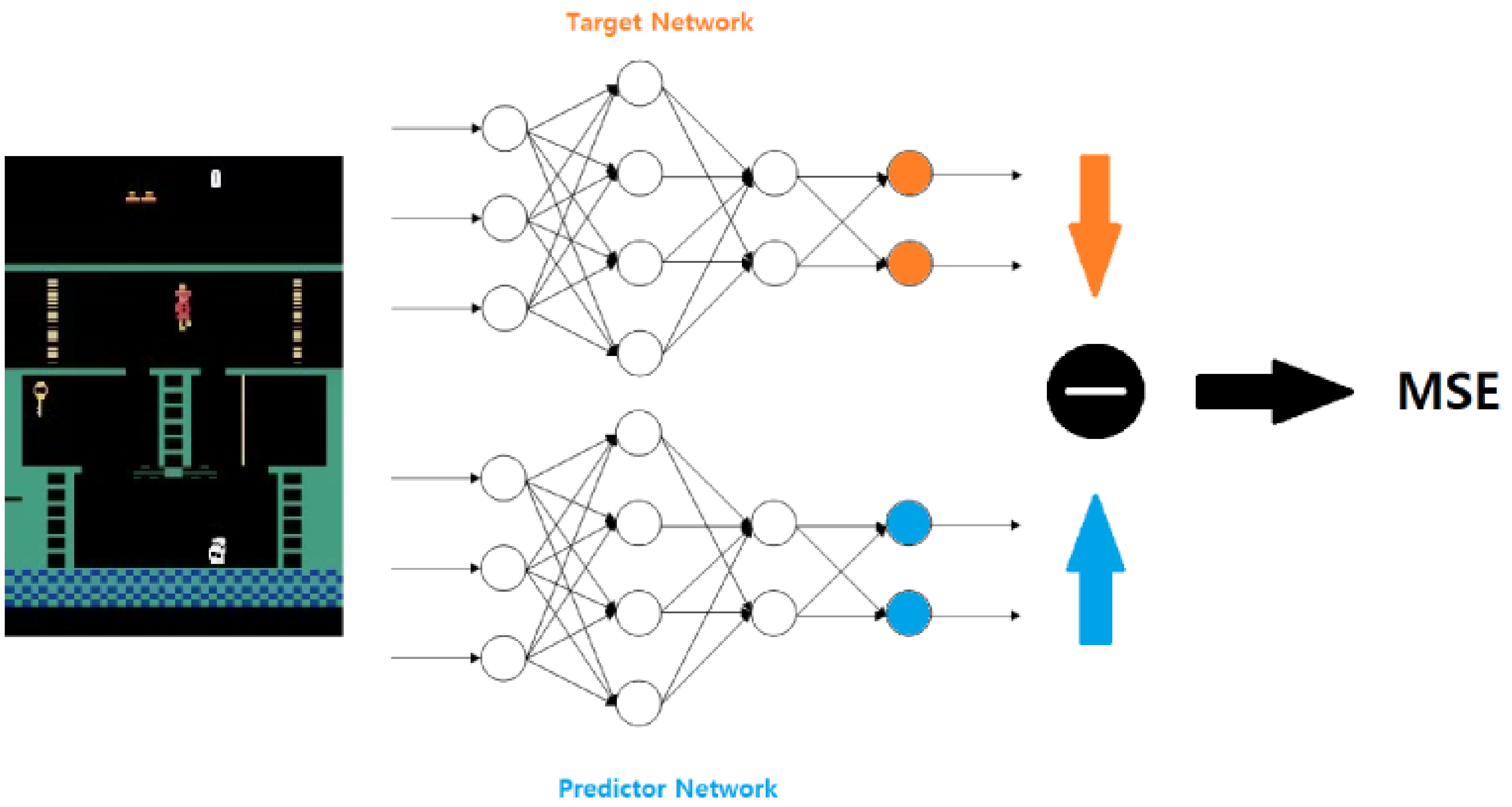
Random Network Distillation (RND)[4]





[4] Exploration by Random Network Distillation (2018)

Random Network Distillation (RND)[4]



[4] Exploration by Random Network Distillation (2018)

RND: details

$$r^{\text{intr}}(s) := \frac{1}{2}\|\phi^{\text{predictor}}(s) - \phi^{\text{target}}(s)\|_2^2$$

[5] How to Break Your Random Network Distillation

RND: details

$$r^{\text{intr}}(s) := \frac{1}{2}\|\phi^{\text{predictor}}(s) - \phi^{\text{target}}(s)\|_2^2$$

- **carefull initialization**: random network should produce *various* outputs for different states.[5]

[5] How to Break Your Random Network Distillation

RND: details

$$r^{\text{intr}}(s) := \frac{1}{2}\|\phi^{\text{predictor}}(s) - \phi^{\text{target}}(s)\|_2^2$$

- **carefull initialization**: random network should produce *various* outputs for different states.[5]
- **carefull normalization**: normalize states using some mean and std (precomputed using several random agent games with no training)

[5] How to Break Your Random Network Distillation

RND: details

$$r^{\text{intr}}(s) := \frac{1}{2}\|\phi^{\text{predictor}}(s) - \phi^{\text{target}}(s)\|_2^2$$

- **carefull initialization**: random network should produce *various* outputs for different states.[5]
- **carefull normalization**: normalize states using some mean and std (precomputed using several random agent games with no training)
  - do not change these statistics during training!

[5] How to Break Your Random Network Distillation

RND: details

$$r^{\text{intr}}(s) := \frac{1}{2}\|\phi^{\text{predictor}}(s) - \phi^{\text{target}}(s)\|_2^2$$

- **carefull initialization**: random network should produce *various* outputs for different states.[5]
- **carefull normalization**: normalize states using some mean and std (precomputed using several random agent games with no training)
  - do not change these statistics during training!
- **start RL algorithm a bit later**: when you are just born, everything is very novel!

[5] How to Break Your Random Network Distillation

RND: details

$$r^{\text{intr}}(s) := \frac{1}{2}\|\phi^{\text{predictor}}(s) - \phi^{\text{target}}(s)\|_2^2$$

- **carefull initialization**: random network should produce *various* outputs for different states.[5]
- **carefull normalization**: normalize states using some mean and std (precomputed using several random agent games with no training)
  - do not change these statistics during training!
- **start RL algorithm a bit later**: when you are just born, everything is very novel!
- **carefull scaling**: look after scale of your intrinsic reward!

[5] How to Break Your Random Network Distillation

RND: details

$$r^{\text{intr}}(s) := \frac{1}{2}\|\phi^{\text{predictor}}(s) - \phi^{\text{target}}(s)\|_2^2$$

- **carefull initialization**: random network should produce *various* outputs for different states.[5]
- **carefull normalization**: normalize states using some mean and std (precomputed using several random agent games with no training)
  - × do not change these statistics during training!
- **start RL algorithm a bit later**: when you are just born, everything is very novel!
- **carefull scaling**: look after scale of your intrinsic reward!
  - RND: divide by running std of intrinsic rewards!

[5] How to Break Your Random Network Distillation

Combining motivations

Straightforward idea: RL algorithm works with

$$r = r^{\text{intr}} + r^{\text{extr}}$$

Combining motivations

Straightforward idea: RL algorithm works with

$$r = r^{\text{intr}} + r^{\text{extr}}$$

Scaling is important though good exploration bonus expires with time!

Combining motivations

Straightforward idea: RL algorithm works with

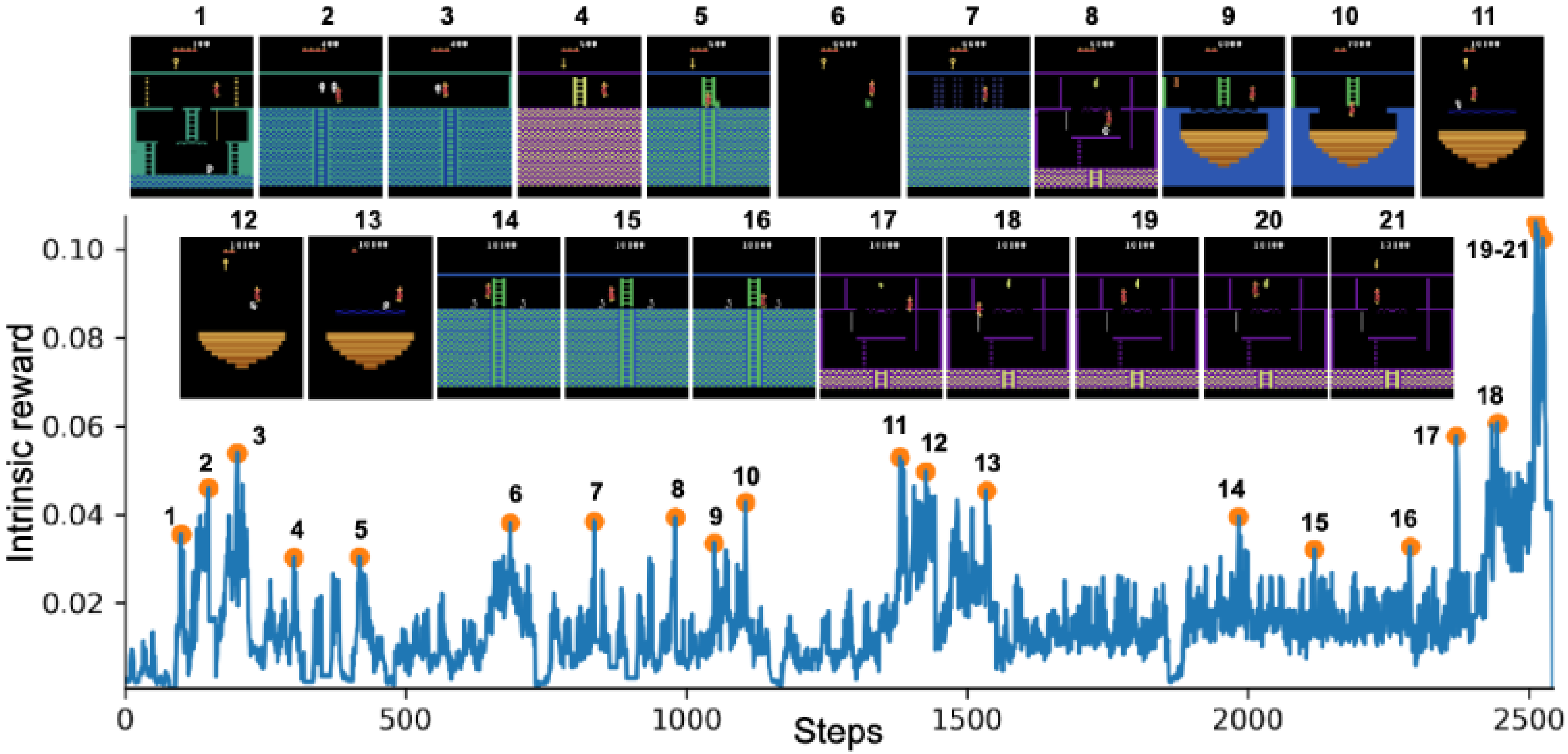
$$r = r^{\text{intr}} + r^{\text{extr}}$$

Scaling is important though good exploration bonus expires with time!

Possible option:

$$Q^{\pi}(s, a) = Q^{\pi}_{\text{intr}}(s, a) + Q^{\pi}_{\text{extr}}(s, a)$$

RND: intrinsic motivation signal



Curiosity

Curiosity is the error of your world model.[1]

[1] Curious model-building control systems (1991)

Curiosity

Curiosity is the error of your world model.[1]

World model $f(s, a)$ tries to predict the outcome s':

$$\frac{1}{2}\|f(s, a) - s'\|_2^2 \to \min_f$$

[1] Curious model-building control systems (1991)

Curiosity

Curiosity is the error of your world model.[1]

World model $f(s, a)$ tries to predict the outcome s':

$$\frac{1}{2}\|f(s, a) - s'\|_2^2 \to \min_f$$

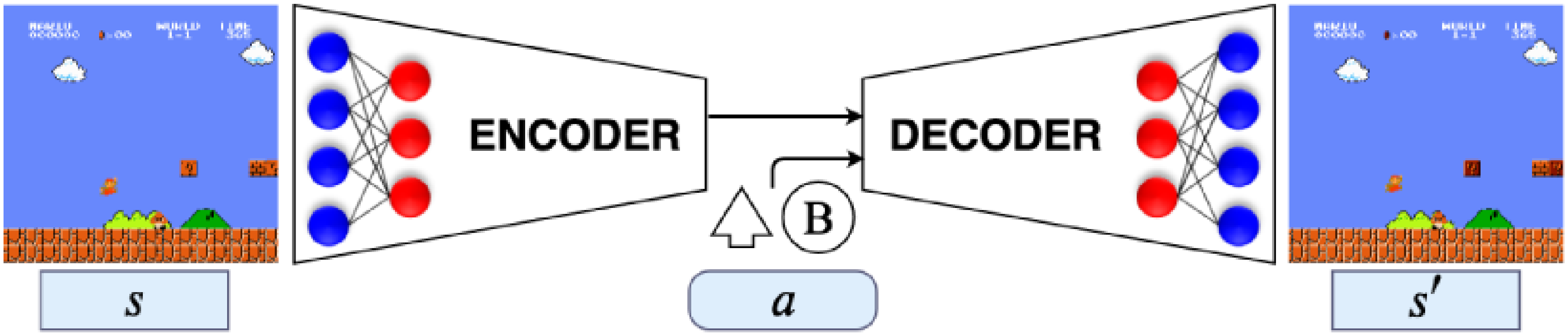
Its error is intrinsic reward:

$$r^{\text{intr}}(s, a, s') := \frac{1}{2}\|f(s, a) - s'\|_2^2$$

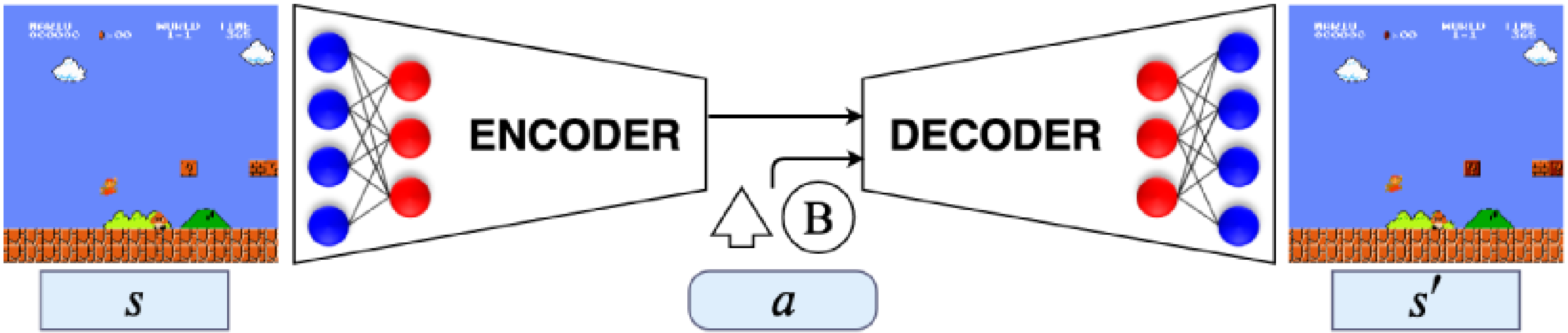
[1] Curious model-building control systems (1991)

Surprise Maximization vs State Novelty

Curiosity: naive approach



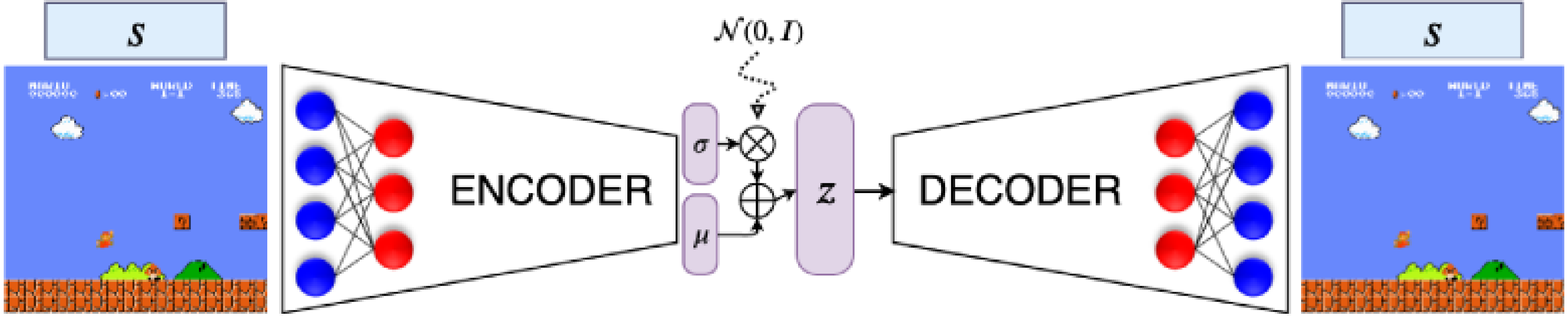
Curiosity: naive approach

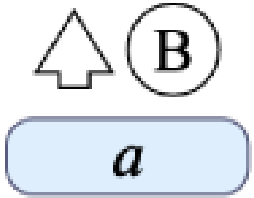


Many problems here...

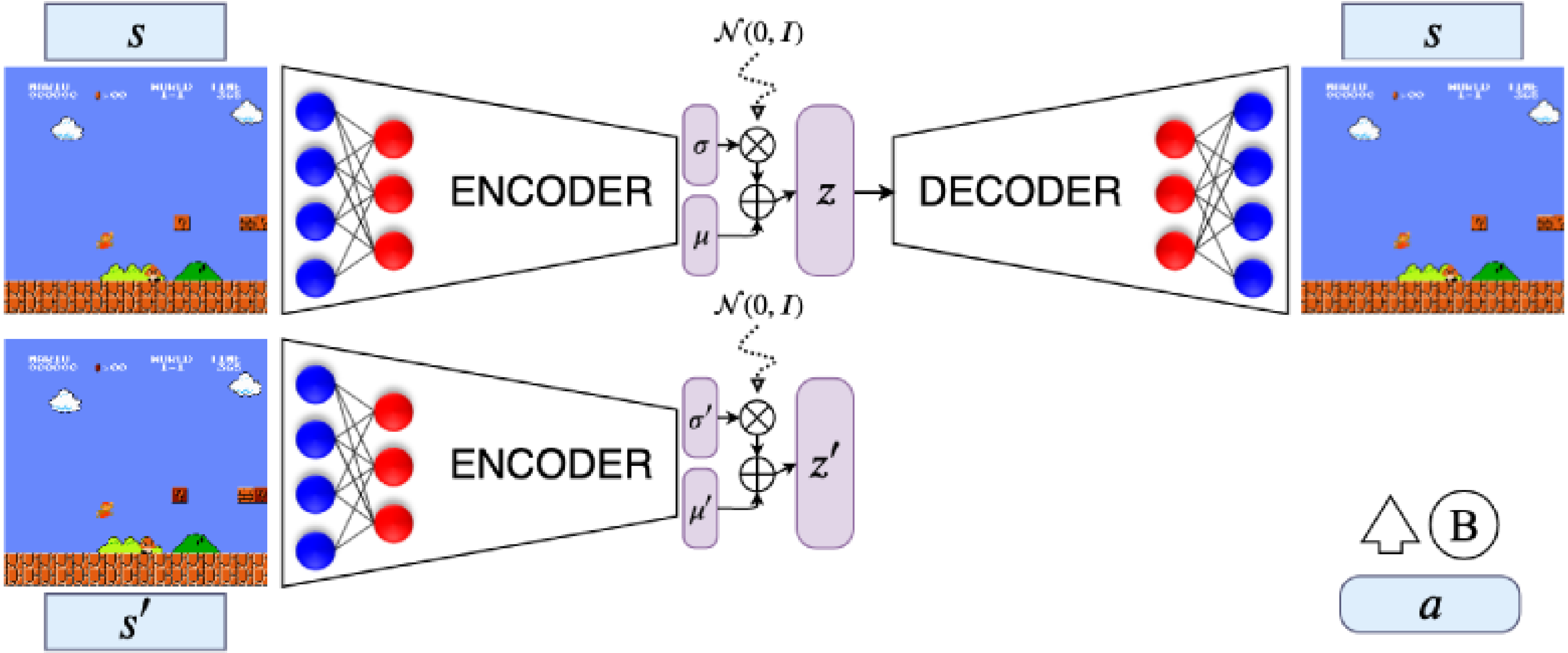
- × generating pictures is pretty expensive
- × learning a lot of unnecessary information
- × comparing pictures in raw pixels space?
- × what if environment is not deterministic?

Curiosity on VAE features

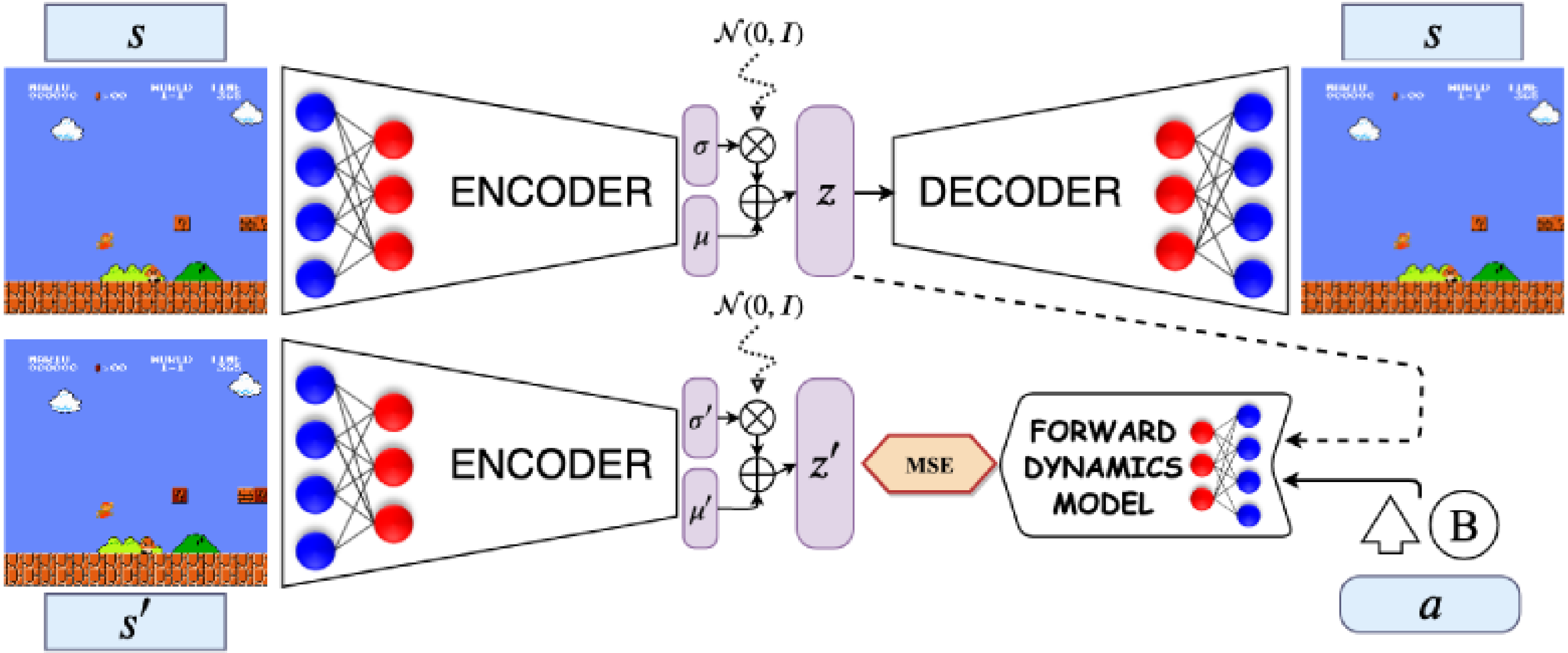




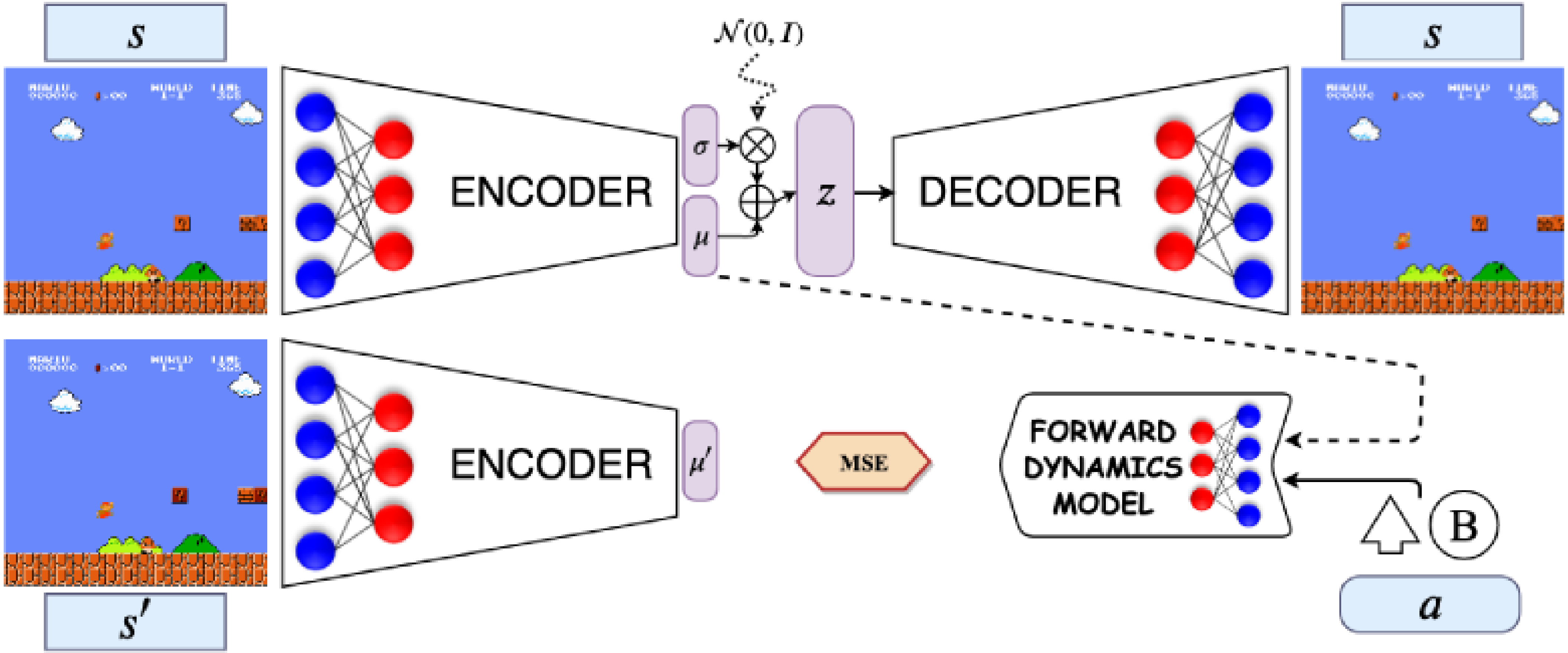
Curiosity on VAE features



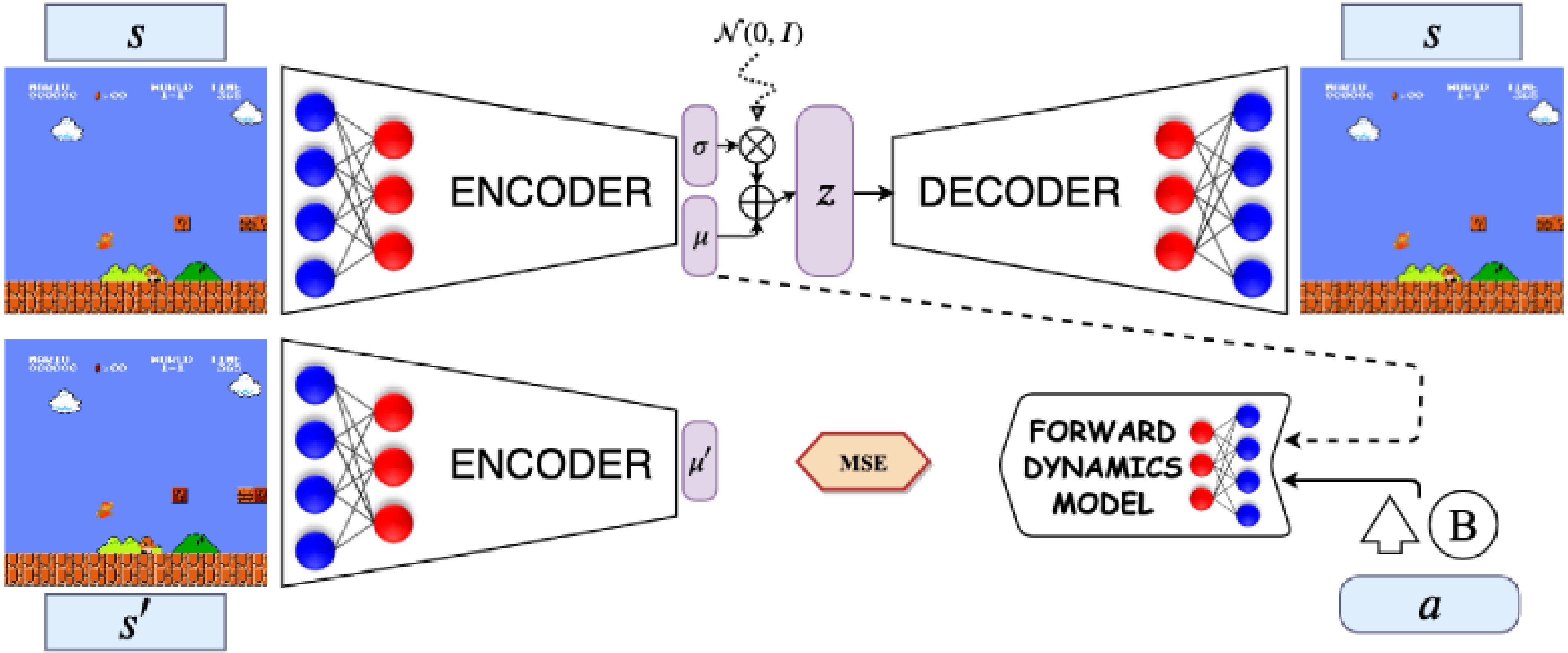
Curiosity on VAE features



Curiosity on VAE features



Curiosity on VAE features



More general idea actually!

Encode complex state spaces in embeddings and solve RL tasks in embedding space!

Noisy TV Problem

Try to predict next frame! In raw pixels, please.

All you need is good embedding

Let $\phi(s)\colon \mathcal{S} \to \mathbb{R}^d$ construct embeddings of states.

What do we want from this embedding:

All you need is good embedding

Let $\phi(s)\colon \mathcal{S} \to \mathbb{R}^d$ construct embeddings of states.

What do we want from this embedding:

- **Filtered**: no irrelevant noise should be present.

All you need is good embedding

Let $\phi(s)\colon \mathcal{S} \to \mathbb{R}^d$ construct embeddings of states.

What do we want from this embedding:

- **Filtered**: no irrelevant noise should be present.
- **Sufficient**: potentially task-relevant information must be saved.

All you need is good embedding

Let $\phi(s)\colon \mathcal{S} \to \mathbb{R}^d$ construct embeddings of states.

What do we want from this embedding:

- **Filtered**: no irrelevant noise should be present.
- **Sufficient**: potentially task-relevant information must be saved.
- **Compact**: we do not want to train image generators.

All you need is good embedding

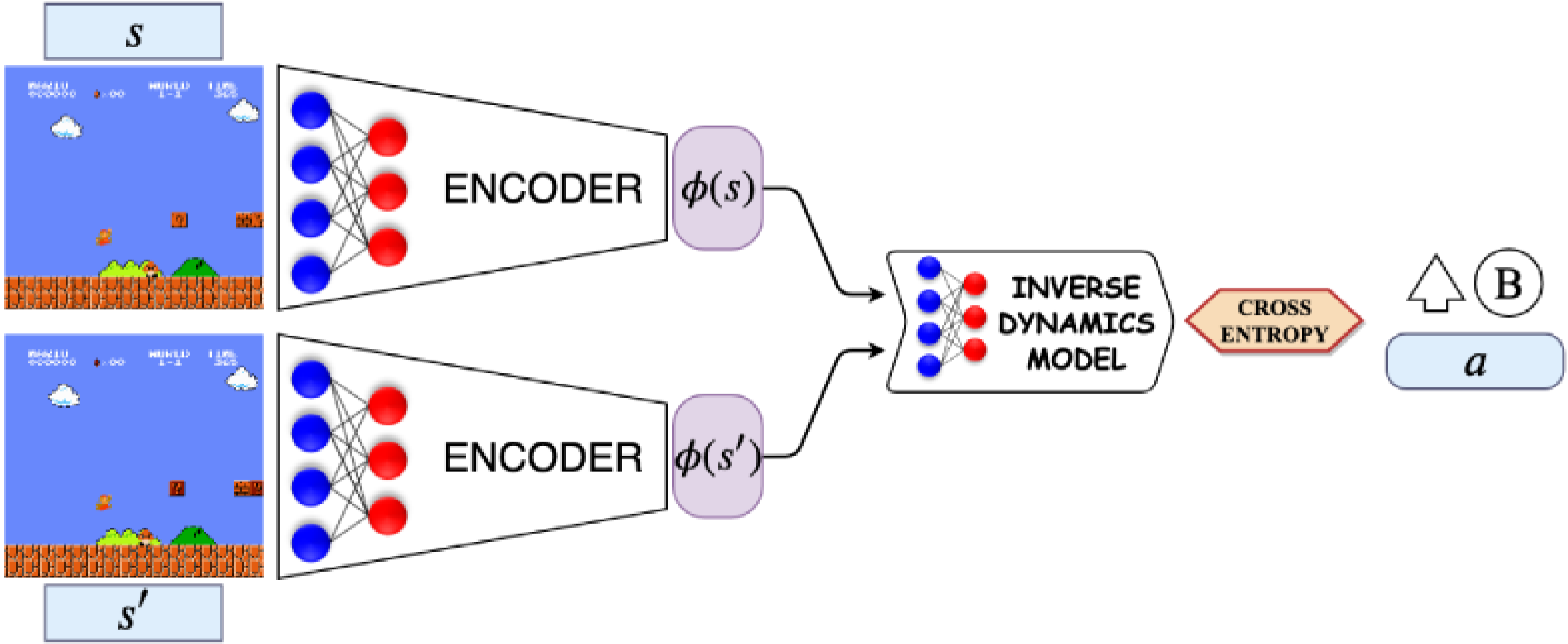
Let $\phi(s)\colon \mathcal{S} \to \mathbb{R}^d$ construct embeddings of states.

What do we want from this embedding:

- **Filtered**: no irrelevant noise should be present.
- **Sufficient**: potentially task-relevant information must be saved.
- **Compact**: we do not want to train image generators.
- **Stable**: ϕ should not change with time or at least change as slow as possible.

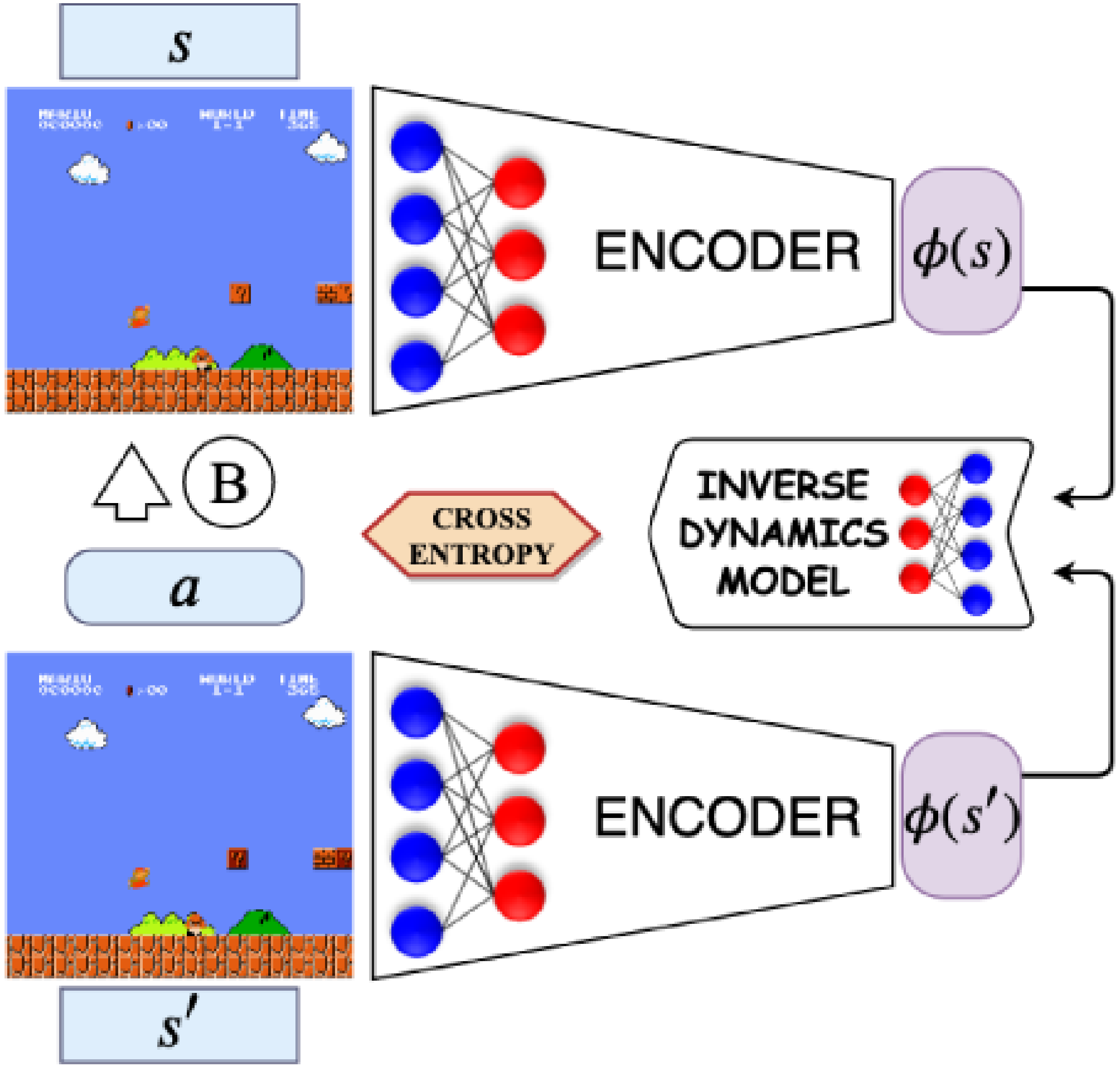
Inverse Dynamics Model

Question: can loss of inverse model be used as curiosity?

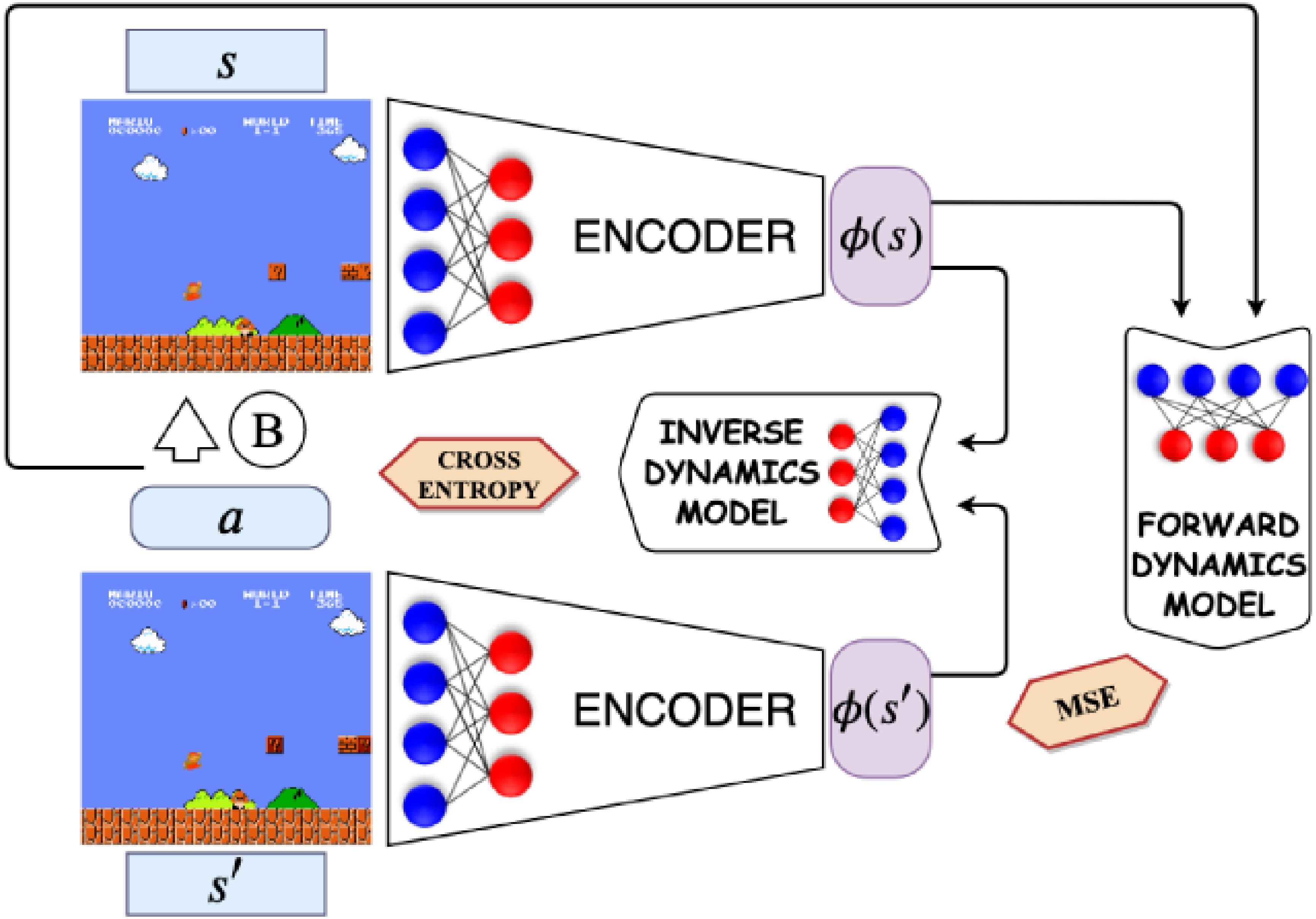


[1]Learning to Play with Intrinsically-Motivated Self-Aware Agents (2018) showed that policy selecting actions that maximize loss of inverse model (you need one more network predicting the losses then) leads to «child playing» behaviors.

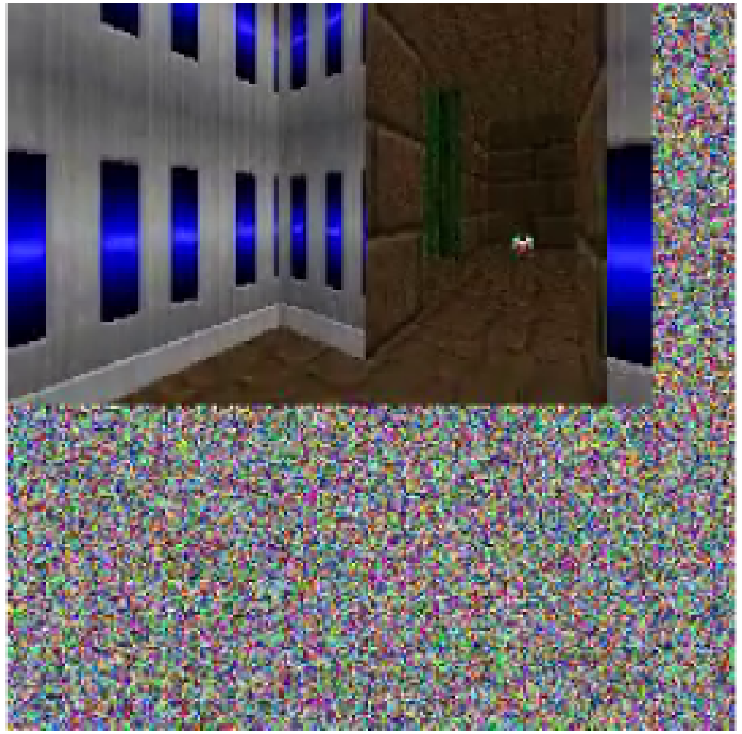
Intrinsic Curiosity Module (ICM)

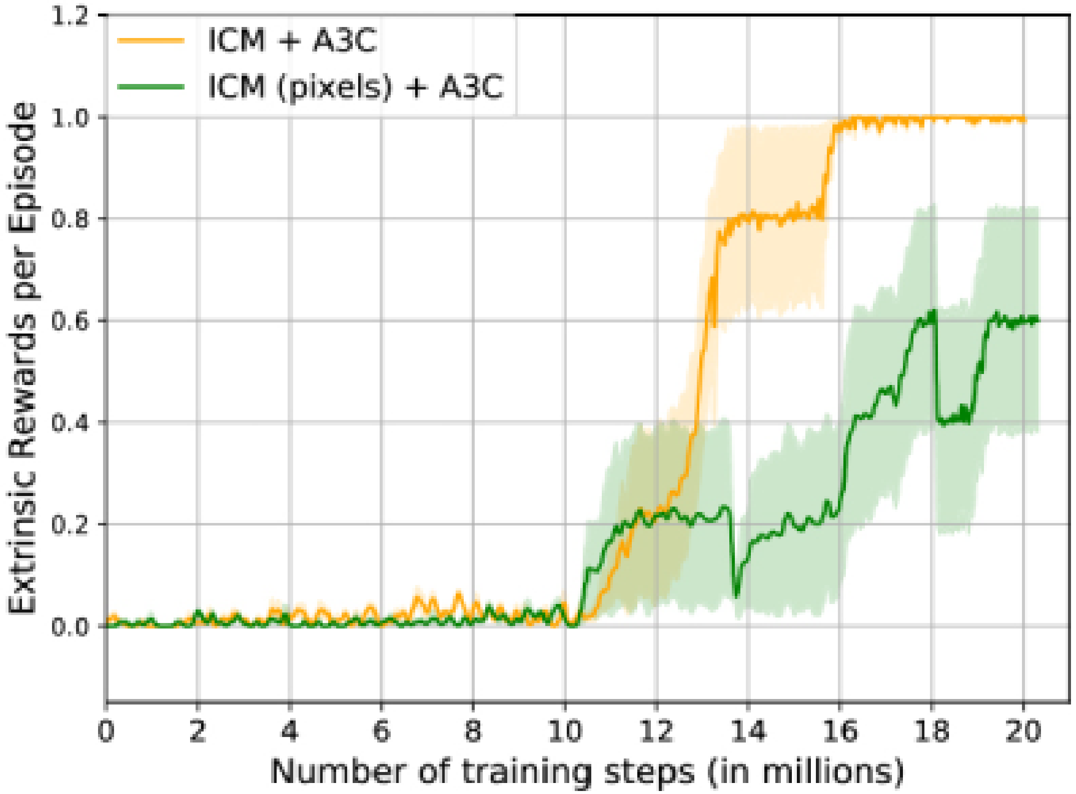


Intrinsic Curiosity Module (ICM)



ICM: Results[6]



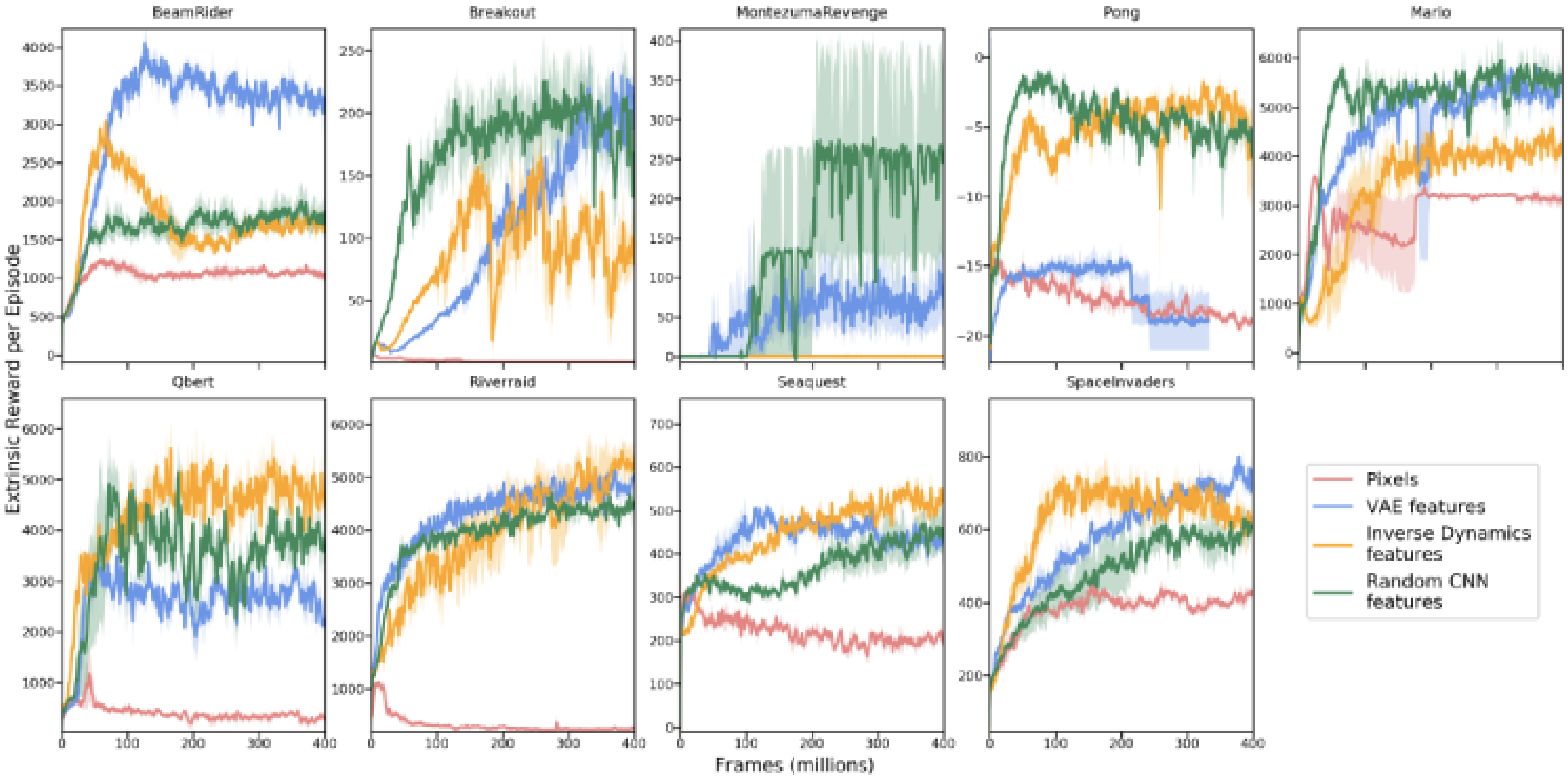


40% of observation image is augmented with noise (VizDoom environment).

ICM still performs well!

[6] Curiosity-driven Exploration by Self-supervised Prediction (2017)

Comparing embeddings[7]



[7] Large-Scale Study of Curiosity-Driven Learning (2018)

Unity ML Agents: Pyramids environment[8]

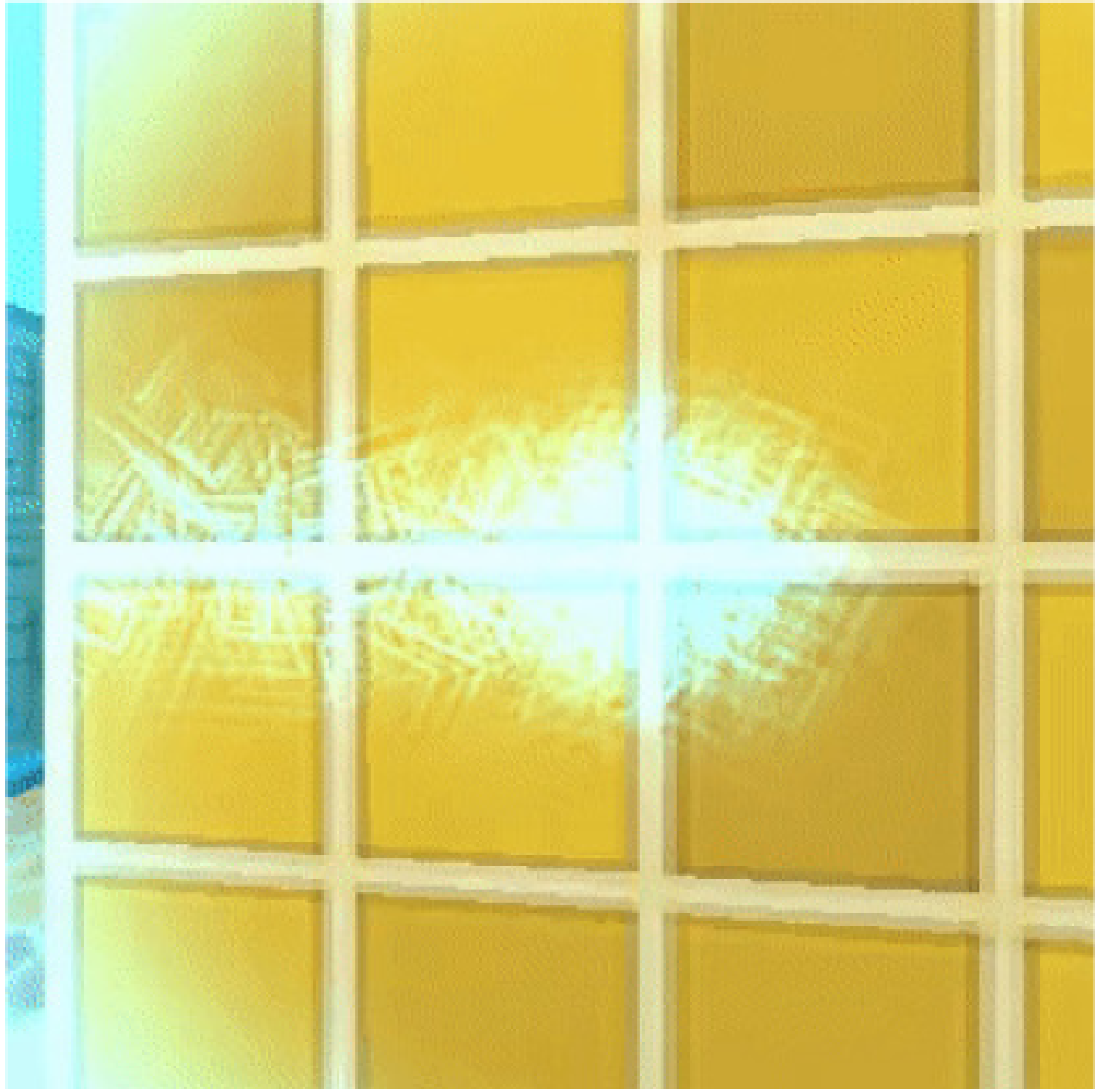
[8] Solving Sparse Reward Tasks with Curiosity

Pyramids: Curiosity only!

Pyramids: Extrinsic motivation + Curiosity

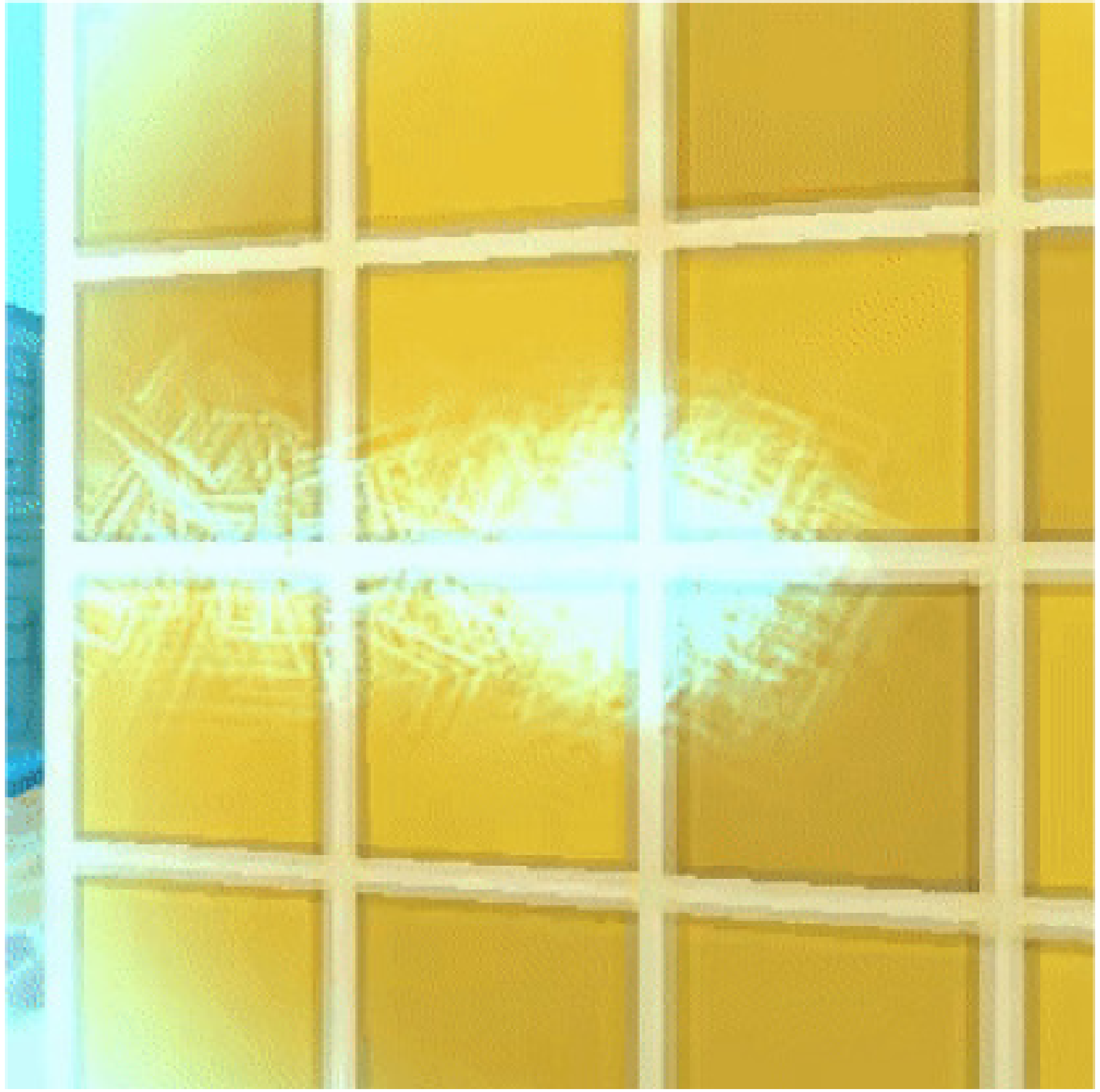
Curiosity: results

ICM: issues



Procrastination issue: Noisy TV that agent can *interact* with!

ICM: issues



<u>Procrastination issue:</u> Noisy TV that agent can *interact* with!

<u>«Short-termed» issue:</u> ICM considers only one-step transitions s, a, s'.