Gradient Boosting for RL in Complex Domains

David Abel[2], Alekh Agarwal[1], Fernando Diaz[1], Akshay Krishnamurthy[1], Robert Schapire[1]

[1]Microsoft Research
[2]Brown University

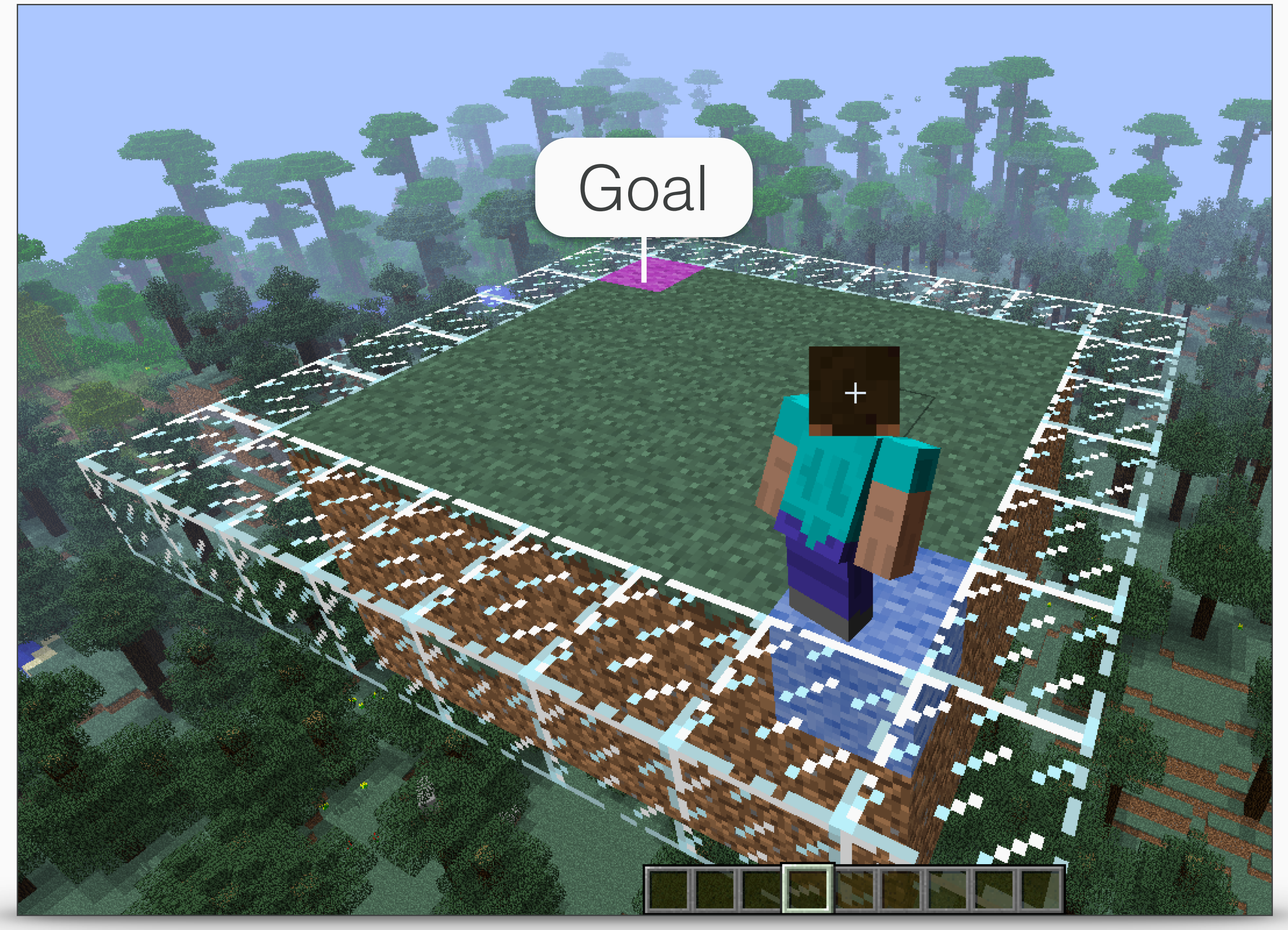
ICML RL and Abstraction Workshop 2016

Goal

Develop simple and scalable Reinforcement Learning (RL) techniques that can solve high dimensional problems.

Minecraft

MALMO: Minecraft AI Testbed



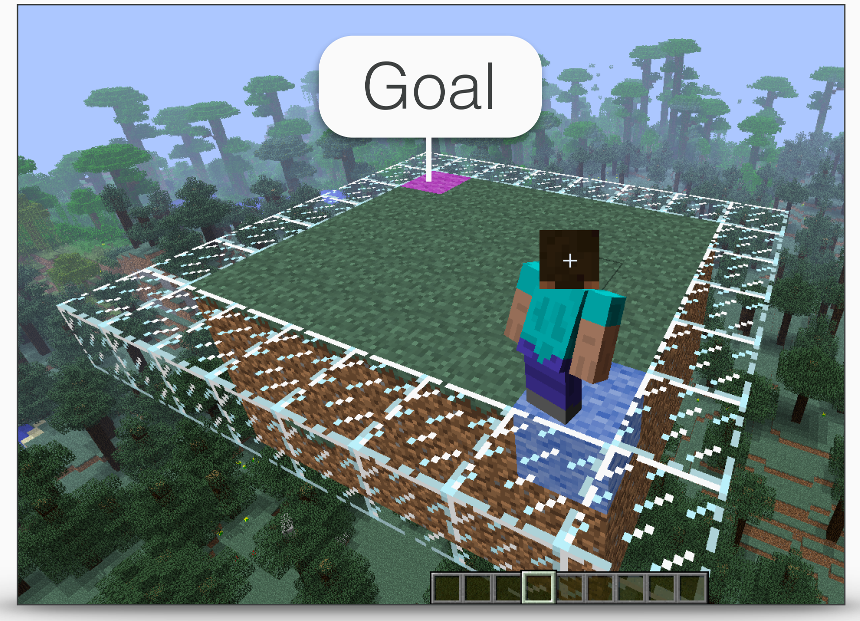
Gridworld

MALMO: an API for developing agents in Minecraft

http://research.microsoft.com/en-us/projects/project-malmo/

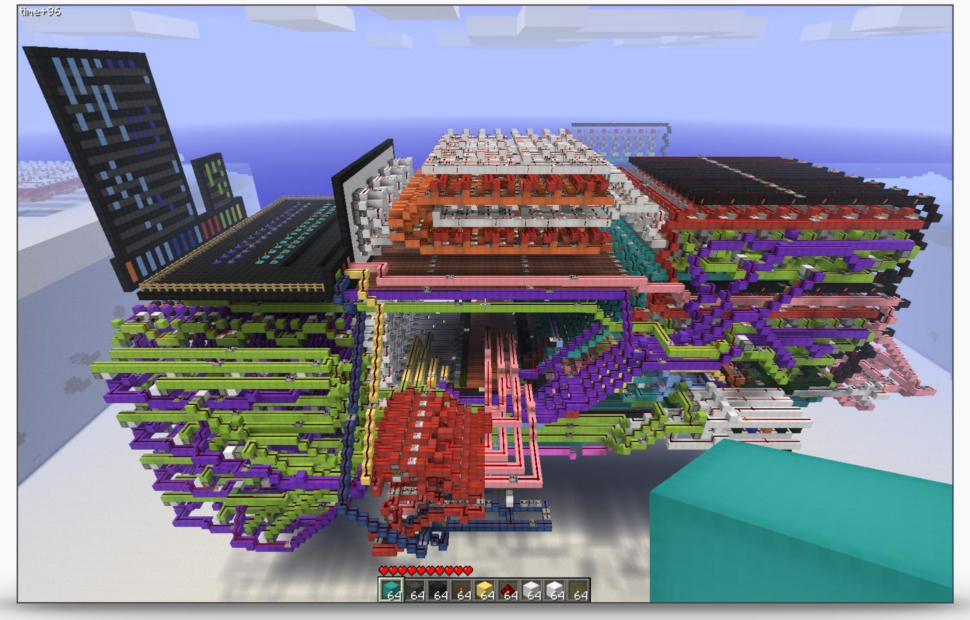
MALMO: Minecraft AI Testbed

Gridworld



difficulty

Build 32 bit ALU



Key Components

Developed an RL agent for Minecraft-scale problems:

Key Components

Developed an RL agent for Minecraft-scale problems:

1) A *vision system* capable of real-time RL in Minecraft.

Key Components

Developed an RL agent for Minecraft-scale problems:

1) A *vision system* capable of real-time RL in Minecraft.

2) **A *new lightweight function approximator* for RL.**

   → Gradient Boosting *[Friedman 2001, Mason 1999]*

Key Components

Developed an RL agent for Minecraft-scale problems:

1) A *vision system* capable of real-time RL in Minecraft.

2) **A *new lightweight function approximator* for RL.**

3) An *exploration* technique for model-free RL
(but: preliminary experiments are inconclusive).

Gradient Boosting for RL

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

1) Fix an ε-greedy policy with respect to $\hat{Q}$

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

1) Fix an ε-greedy policy with respect to $\hat{Q}$

2) Run an episode —> receive a dataset:

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

1) Fix an ε-greedy policy with respect to $\hat{Q}$

2) Run an episode —> receive a dataset:

$$\boxed{\mathcal{D} = \langle (\underbrace{s_1}_{\text{state}}, \underbrace{a_1}_{\text{action}}, \underbrace{r_1}_{\text{reward}}) \, , \; \ldots \; , (s_N, a_N, r_N) \rangle}$$

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

1) Fix an ε-greedy policy with respect to $\hat{Q}$

2) Run an episode —> receive a dataset:

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

1) Fix an ε-greedy policy with respect to $\hat{Q}$

2) Run an episode —> receive a dataset:

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_h \sum_{i=1}^{N} \left[\boxed{h(s_i, a_i)} + \boxed{\hat{Q}(s_i, a_i)} - \boxed{(r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a'))} \right]^2$$

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_h \sum_{i=1}^{N} \Big[\underbrace{\boxed{h(s_i, a_i)}}_{\text{new weak learner}} + \underbrace{\boxed{\hat{Q}(s_i, a_i)}}_{\text{previous estimate}} - \underbrace{\boxed{\big(r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')\big)}}_{\text{Bellman residual}} \Big]^2$$

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_h \sum_{i=1}^{N} \left[\boxed{h(s_i, a_i)} + \boxed{\hat{Q}(s_i, a_i)} - \boxed{\left(r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')\right)} \right]^2$$

Where:

$$\hat{Q}(s, a) = \sum_{e=1}^{E} h_e(s, a)$$

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_{h} \sum_{i=1}^{N} \left[\boxed{h(s_i, a_i)} + \boxed{\hat{Q}(s_i, a_i)} - \boxed{\left(r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')\right)} \right]^2$$

 # episodes

Where:

one weak learner per episode

$$\hat{Q}(s, a) = \sum_{e=1}^{\boxed{E}} \boxed{h_e(s, a)}$$

Treat RL as a Regression problem for the Q-function

Gradient Boosting for RL

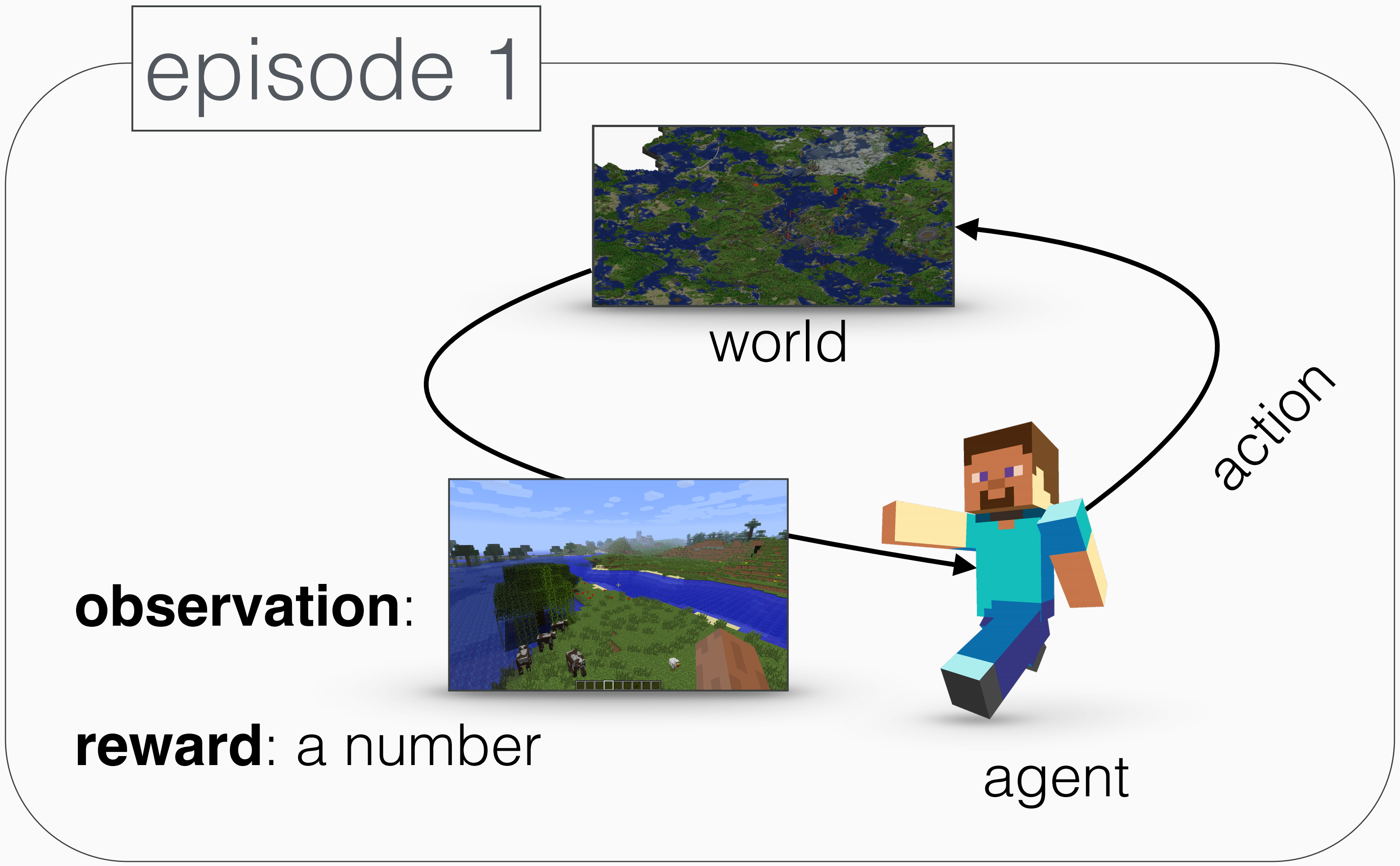
3) Fit a new estimate of $\hat{Q}$ by minimizing the Bellman Residual on the data set, $\mathcal{D}$:

$$\min_{h} \sum_{i=1}^{N} \left[h(s_i, a_i) + \hat{Q}(s_i, a_i) - (r_i + \gamma \max_{a'} \hat{Q}(s_{i+1}, a')) \right]^2$$

We solve this using **regression trees** as the weak learner

Treat RL as a Regression problem for the Q-function

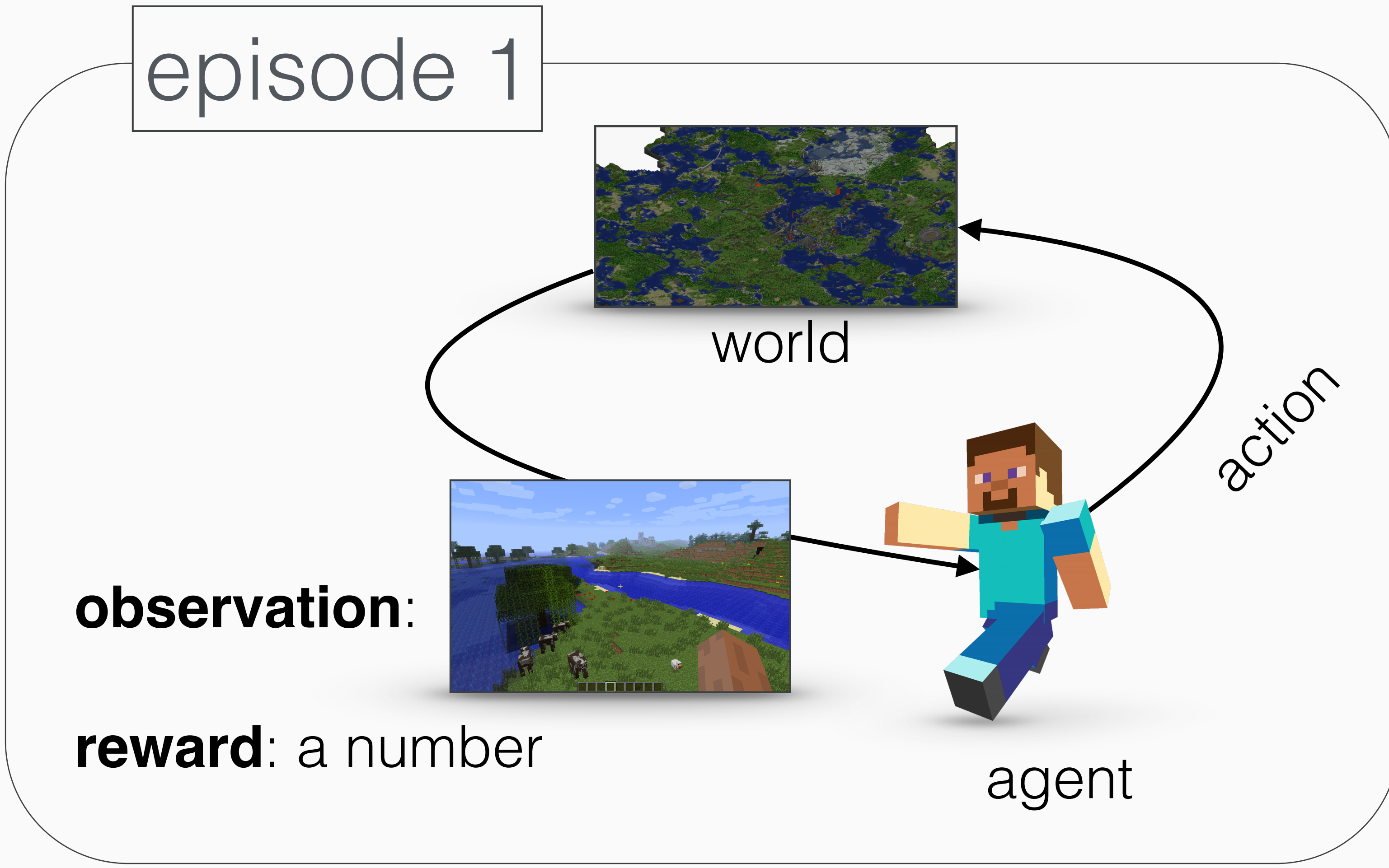
High Level



High Level

$$\hat{Q} = \sum(\text{[redacted]})$$

High Level

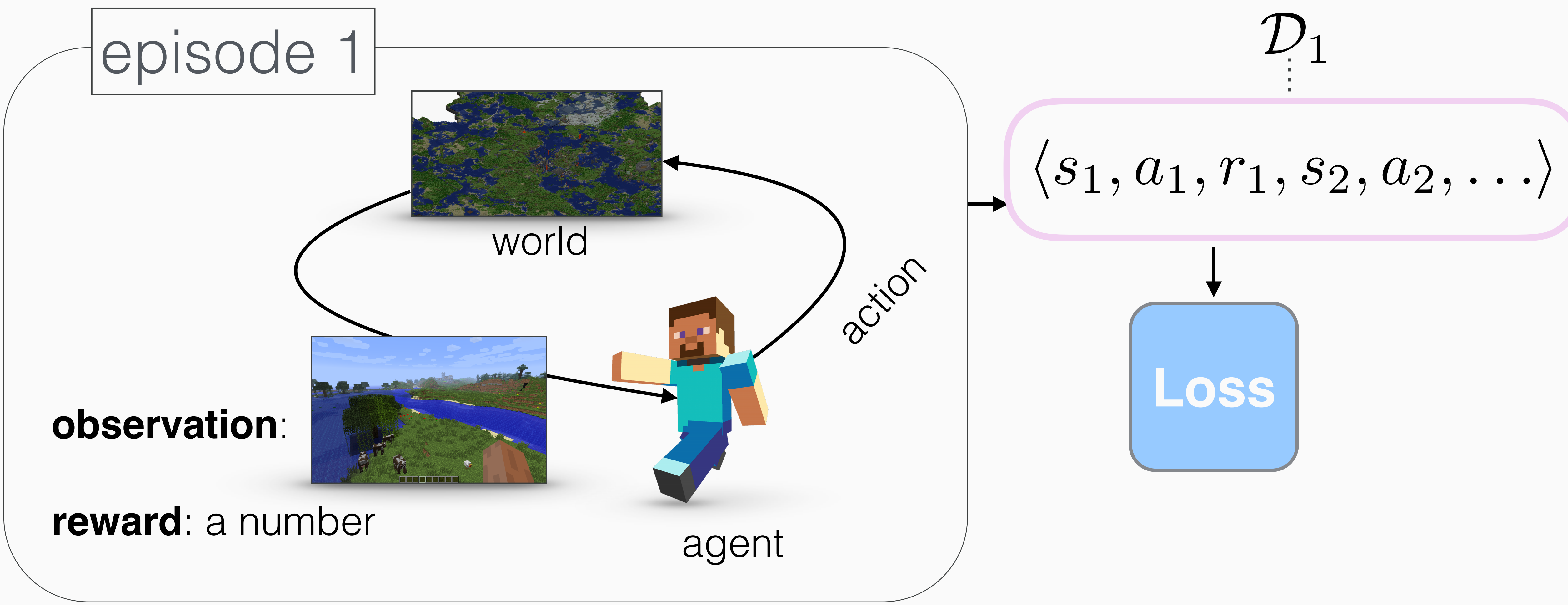


$\mathcal{D}_1$

$\langle s_1, a_1, r_1, s_2, a_2, \ldots \rangle$

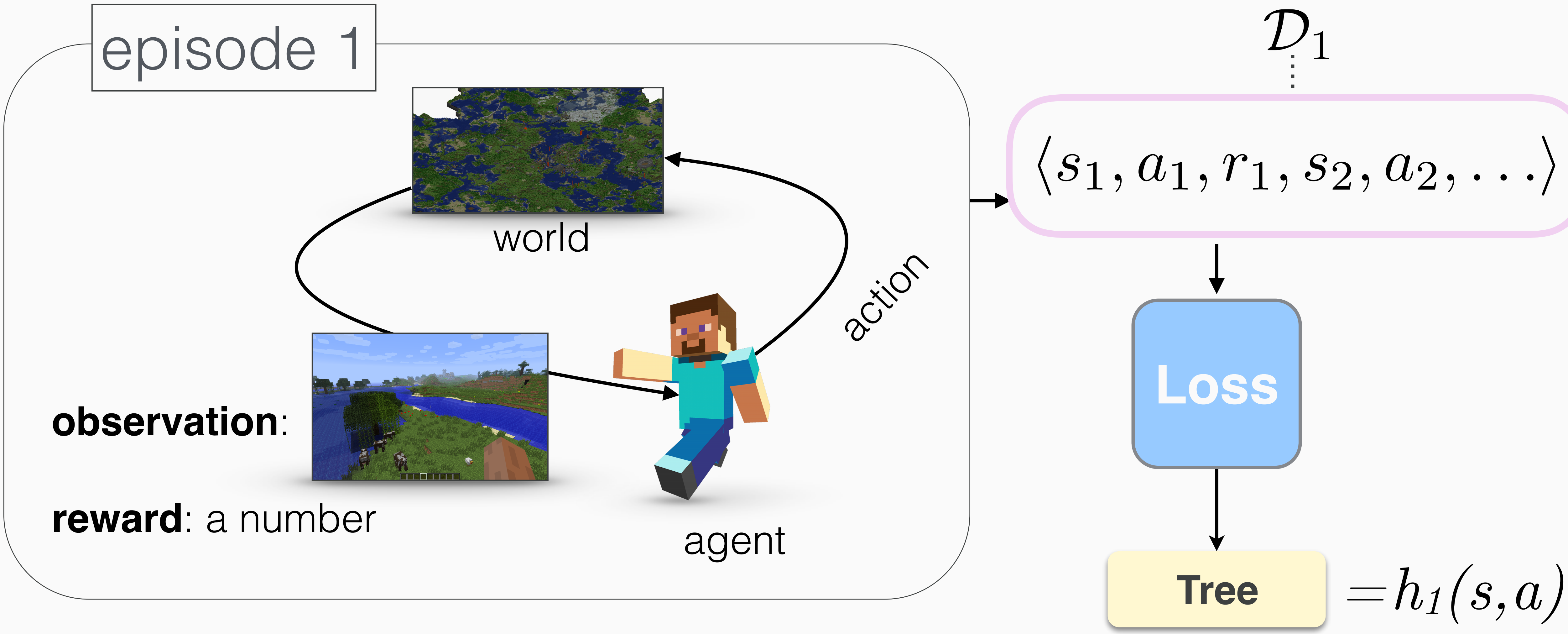
$\hat{Q} = \sum($ [redacted] $)$

High Level



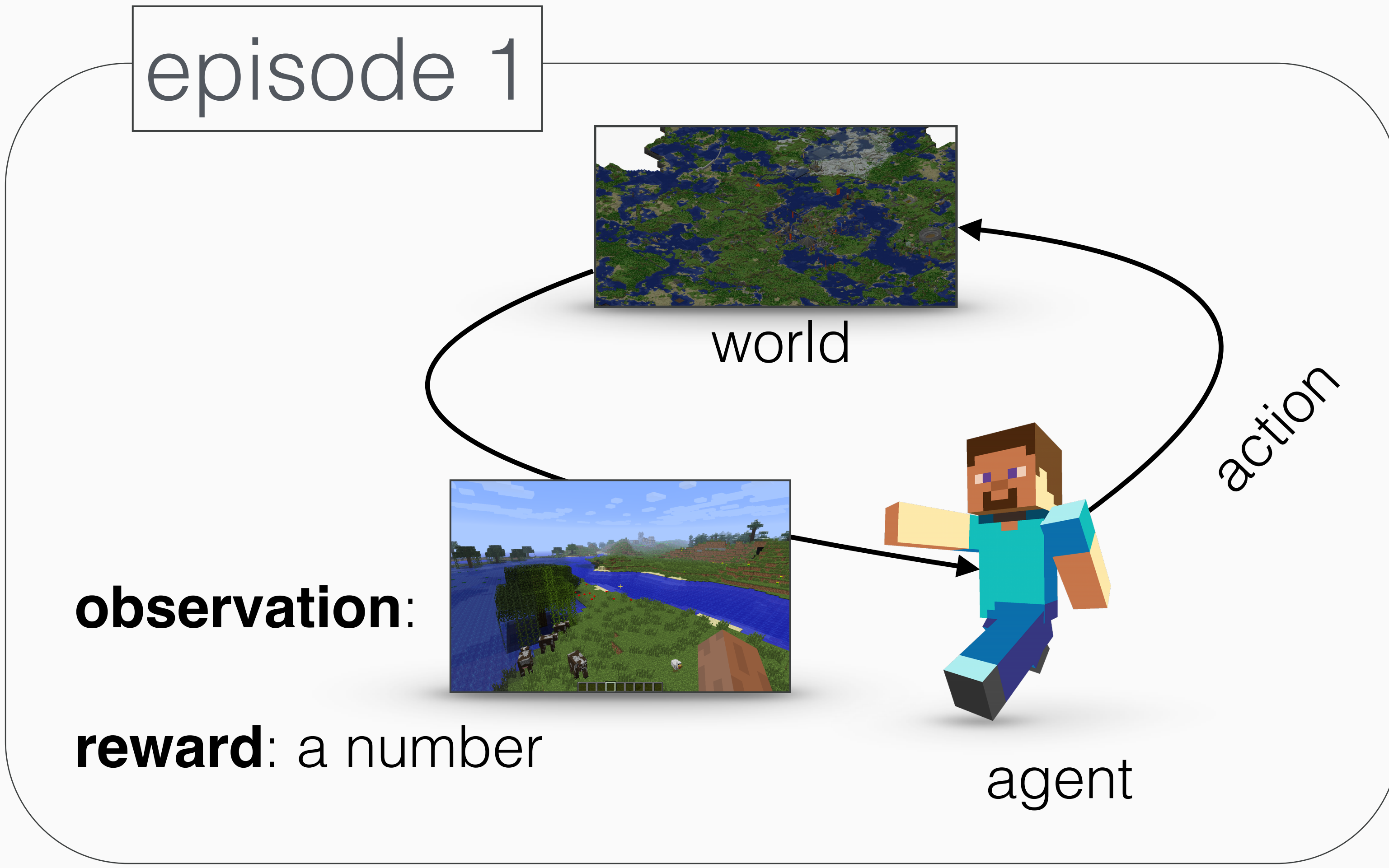
$\hat{Q} = \sum($ [redacted] $)$

High Level



$$\hat{Q} = \sum (\text{[redacted]})$$

High Level



$\mathcal{D}_1$

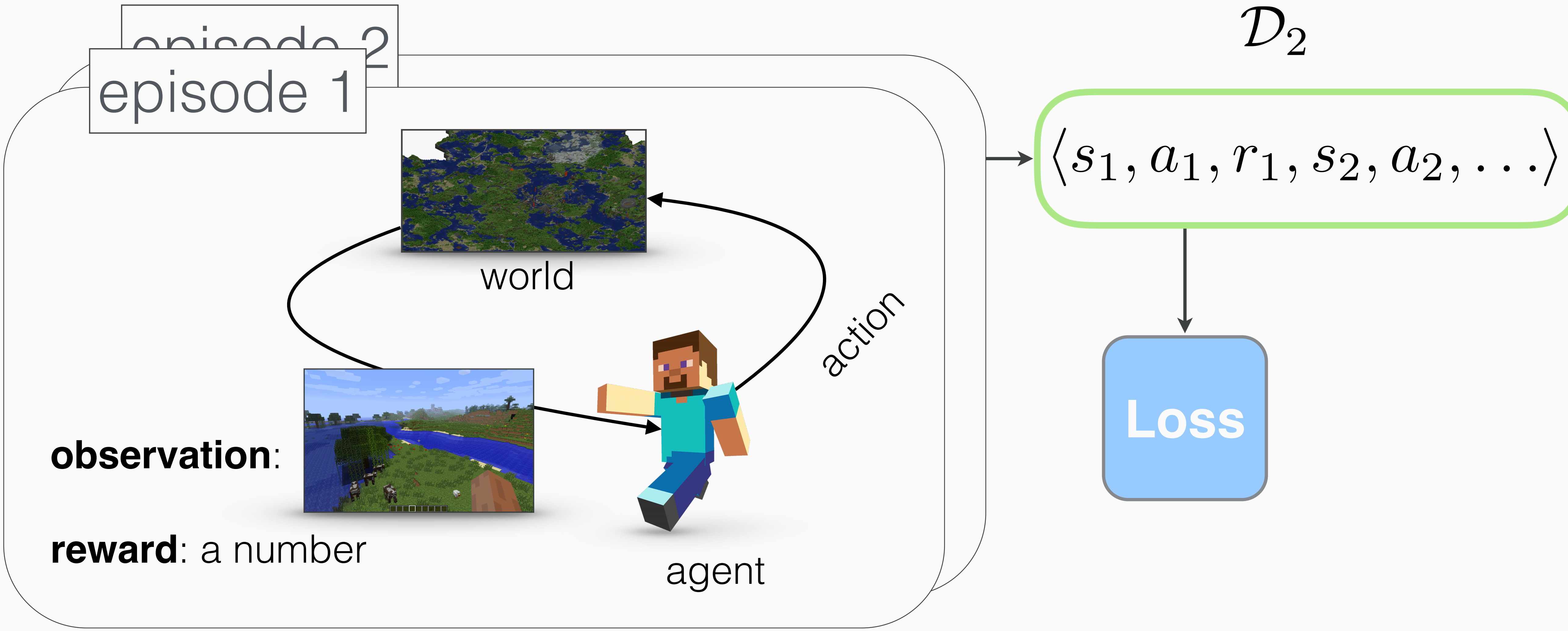
$$\langle s_1, a_1, r_1, s_2, a_2, \ldots \rangle$$

Loss

$$\hat{Q} = \sum \left(h_1 \right)$$

$\mathcal{D}_1$

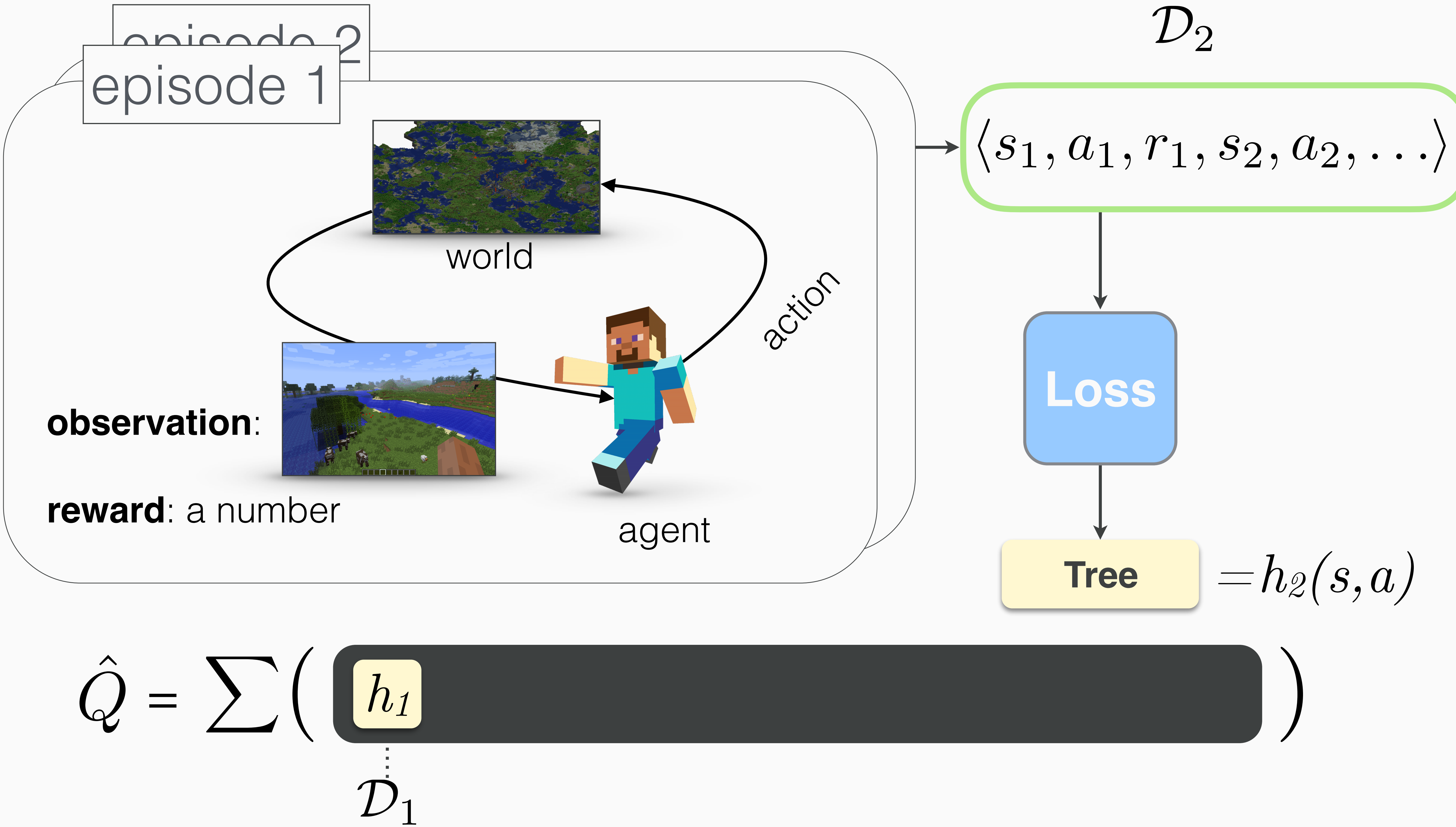
High Level



$$\hat{Q} = \sum\left(h_1 \right)$$

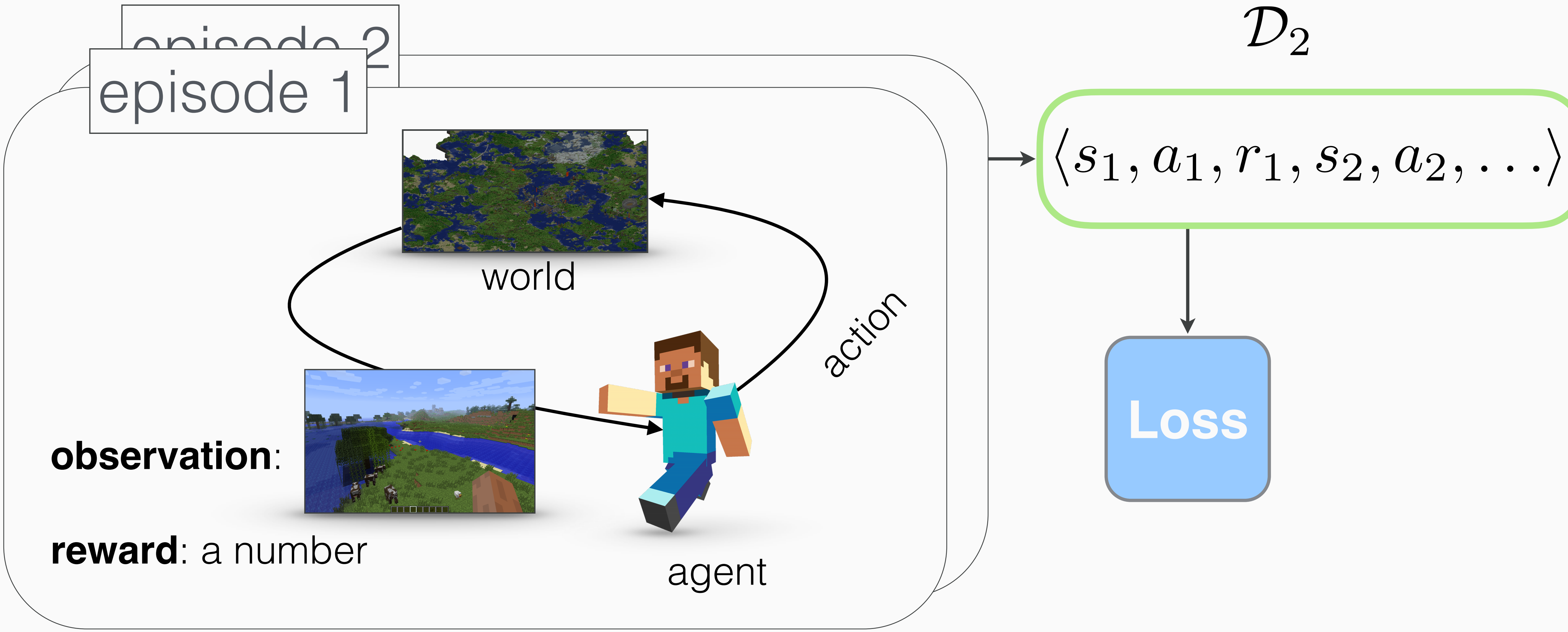
$\mathcal{D}_1$

High Level



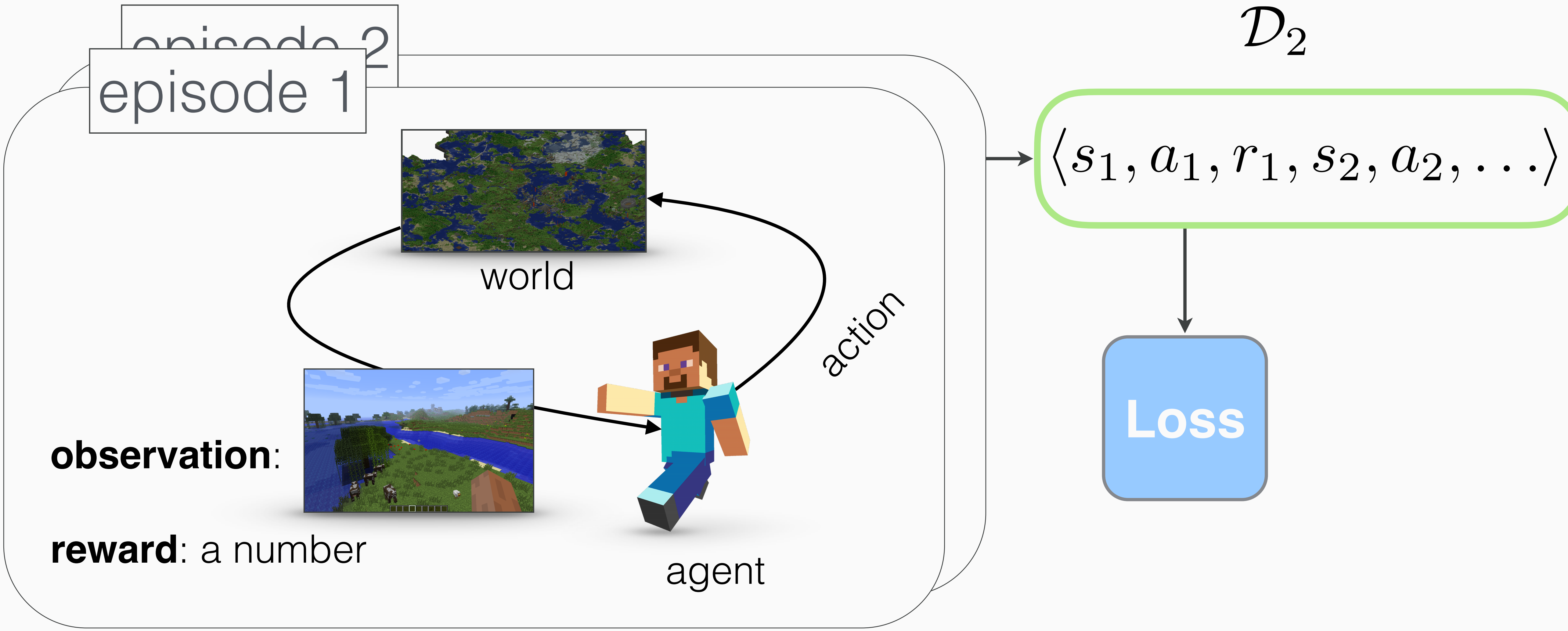
episode 2
episode 1
world
action
observation:
reward: a number
agent
$\mathcal{D}_2$
$\langle s_1, a_1, r_1, s_2, a_2, \ldots \rangle$
Loss
Tree
$= h_2(s, a)$
$\hat{Q} = \sum \left(h_1 \right)$
$\mathcal{D}_1$

High Level



$$\hat{Q} = \sum \Big(\underbrace{h_1}_{\mathcal{D}_1}, \underbrace{h_2}_{\mathcal{D}_2} \Big)$$

High Level



$$\hat{Q} = \sum\left(\underset{\mathcal{D}_1}{h_1}, \underset{\mathcal{D}_2}{h_2}, \underset{\dots}{h_3}, h_4, \underset{\dots}{h_5}, h_6, \underset{\dots}{h_7}, \underset{\mathcal{D}_8}{h_8} \right)$$

Intuitively Nice Properties

- Non-parametric
- Simple, easy to implement, minimal hand-engineering
- Interleaved data collection
- Rich theoretical literature, room for analysis.
- Only need to store one episode’s worth of data.

Experiments: Baselines

- Baseline 1 .... *(Linear Approximator)*
- Baseline 2 .... *(Random Forest Approximator)*
- Baseline 3 .... *(Batch Boost Approximator)*

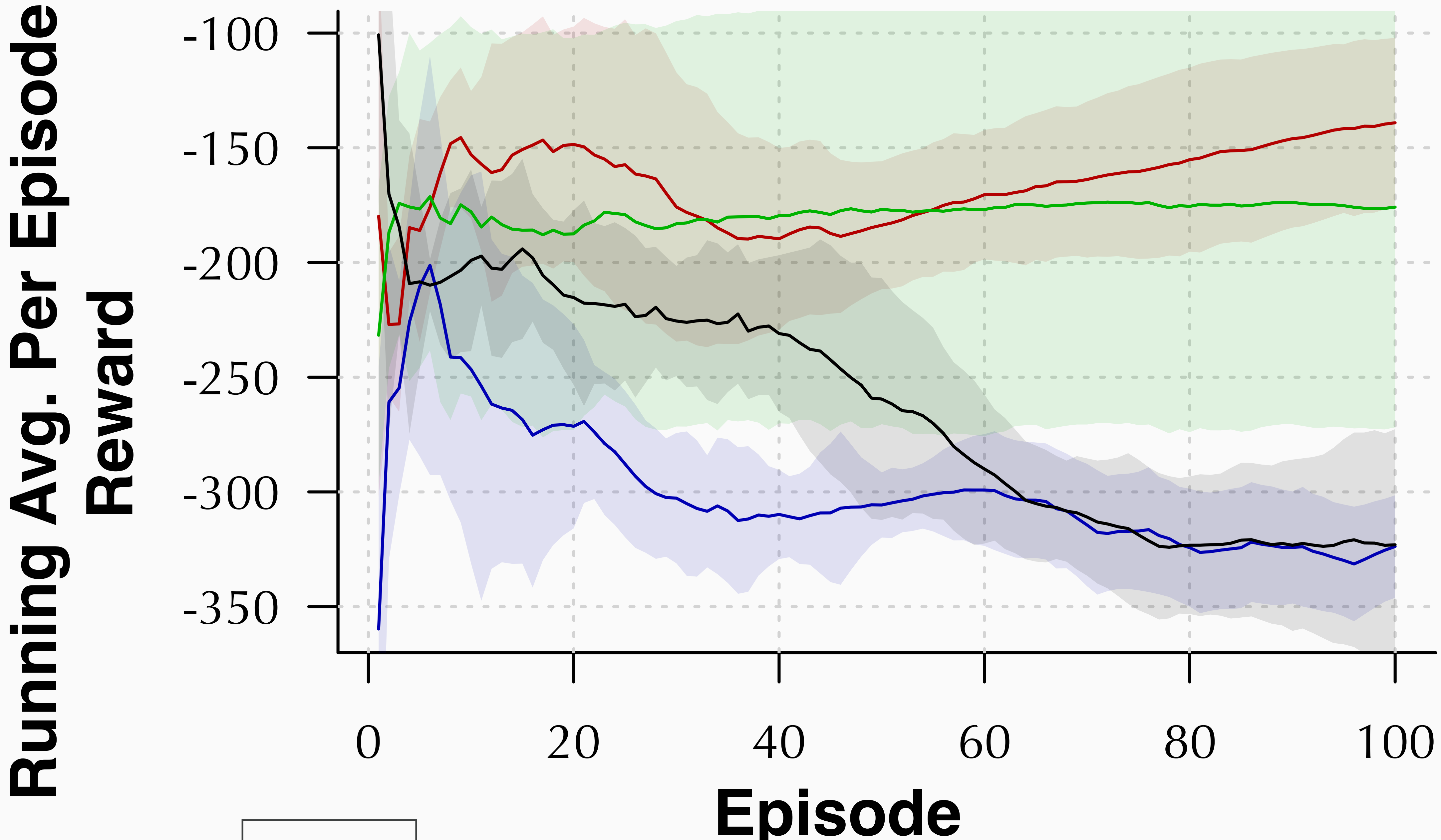
Experiments: Baselines

- Baseline 1 .... *(Linear Approximator)*
- Baseline 2 .... *(Random Forest Approximator)*
- Baseline 3 .... *(Batch Boost Approximator)*

Similar to Fitted *Q*-iteration *[Ernst et al. 2005]*

Experiments: Visual Grid

Visual Grid: Results

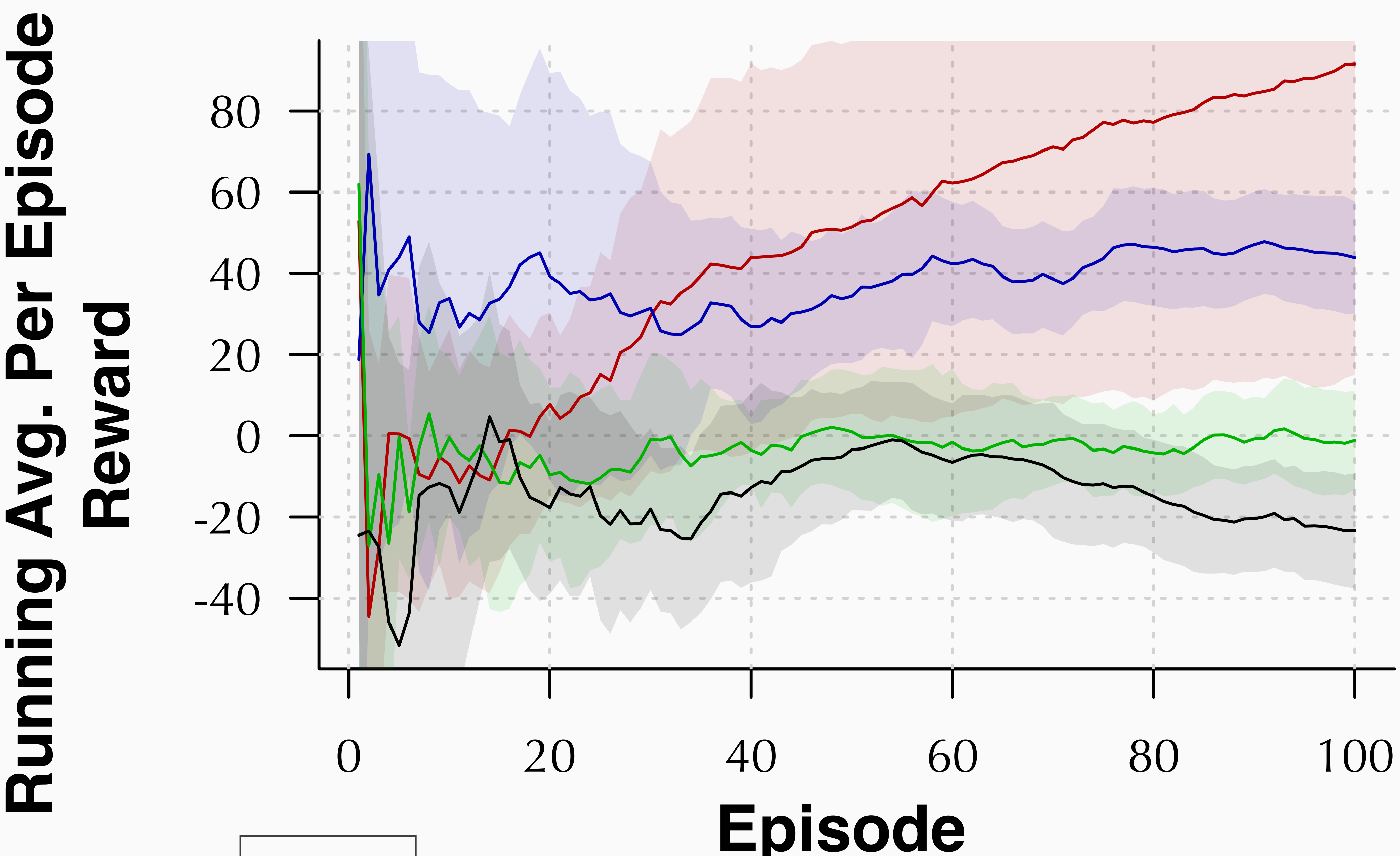


Key

Gradient Booster *Batch Booster* *Linear* *Forest*

Experiments: Hillclimbing

Visual Hill Climb: Results



Key

Gradient Booster *Batch Booster* *Linear* *Forest*

Next Steps

- Investigate relevant exploration techniques inspired by Gradient Boosting.
- Use rich foundation of theory on gradient boosting to inspire analysis of this approach.
- Further experimentation.

Acknowledgments

A big thank you to The *MALMO* team!

David Bignell, Katja Hofmann, Tim Hutton, Matthew Johnson, Pushmeet Kohli, Nate Kushman, Ewa Luger, Bhaskar Mitra, Jamie Shotton, Evelyne Viegas.

http://research.microsoft.com/en-us/projects/project-malmo/