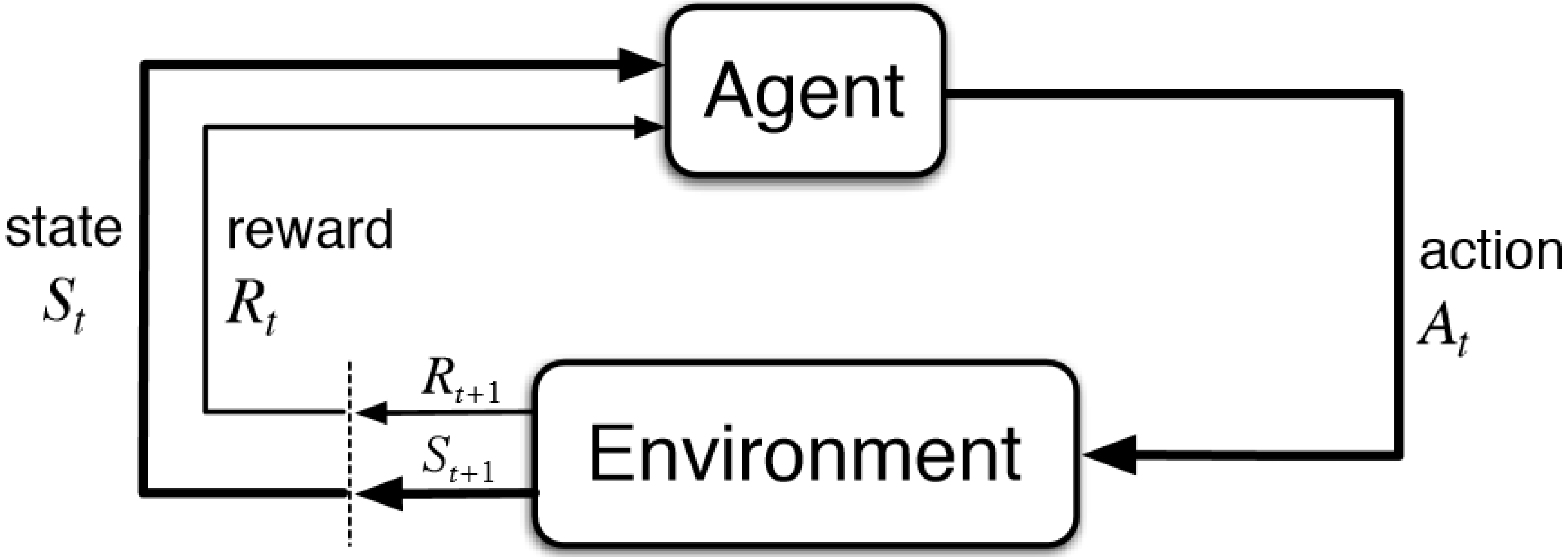
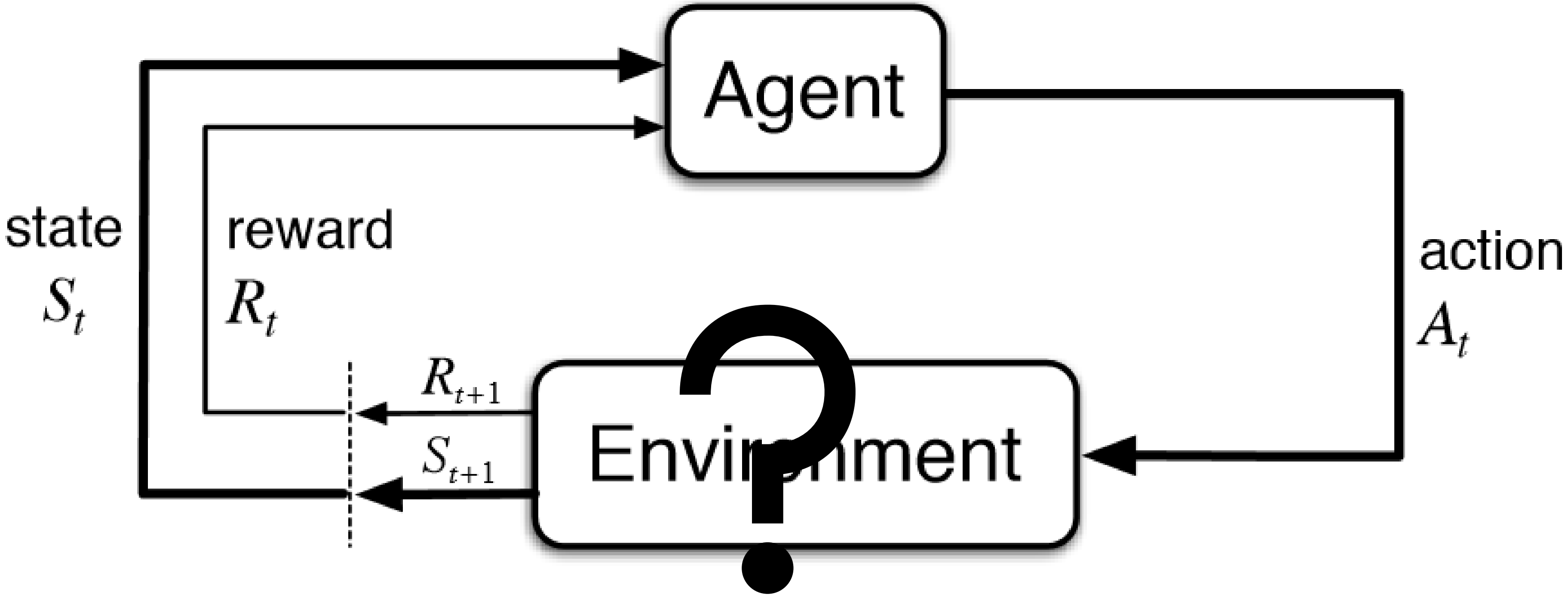
Model Based Reinforcement Learning

Presenter: Adrian Hoffmann

Basic Set-Up



Basic Set-Up



Overview

- Definitions and their problems
- Why I want a model
- Paper “The Effect of Planning Shape on Dyna-style Planning in High-dimensional State Spaces”
- Paper “World Models”

Definition Model-Based

Definition Model-Based

Algorithm 1 Model-based reinforcement learning

Input: state sample procedure d
Input: model m
Input: policy π
Input: predictions v
Input: environment $\mathcal{E}$
Get initial state $s \leftarrow \mathcal{E}$
for iteration $\in \{1, 2, \ldots, K\}$ **do**
    for interaction $\in \{1, 2, \ldots, M\}$ **do**
        Interactions with the real Environment
    end for
    for planning step $\in \{1, 2, \ldots, P\}$ **do**
        Take advantage of the model
    end for
end for

Van Hasselt et al. 2019

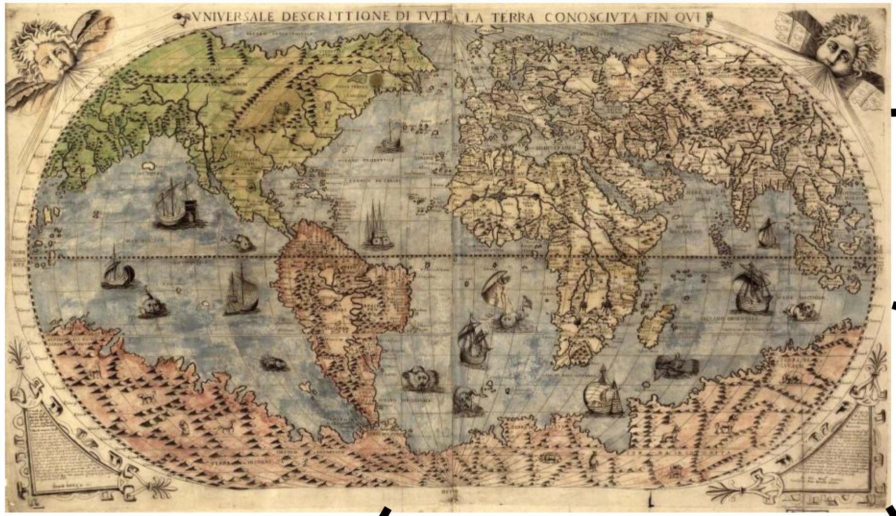
Definition Model-Based

Algorithm 1 Model-based reinforcement learning

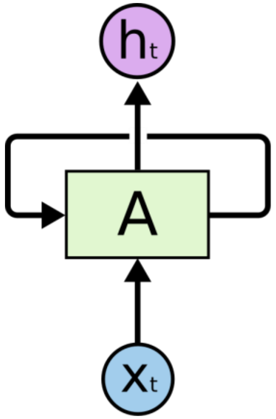
Input: state sample procedure d
Input: model m
Input: policy π
Input: predictions v
Input: environment $\mathcal{E}$
Get initial state $s \leftarrow \mathcal{E}$
for iteration $\in \{1, 2, \ldots, K\}$ **do**
    for interaction $\in \{1, 2, \ldots, M\}$ **do**
        Interactions with the real Environment
    end for
    for planning step $\in \{1, 2, \ldots, P\}$ **do**
        Take advantage of the model
    end for
end for

Van Hasselt et al. 2019

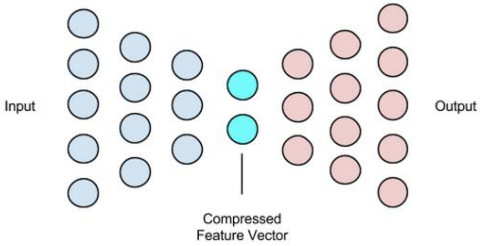
Interlude – Model



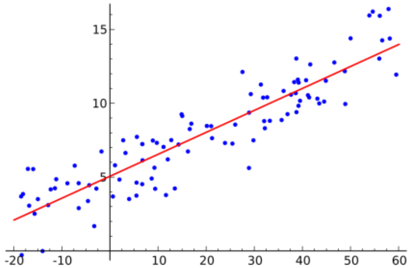
RNN, LSTM



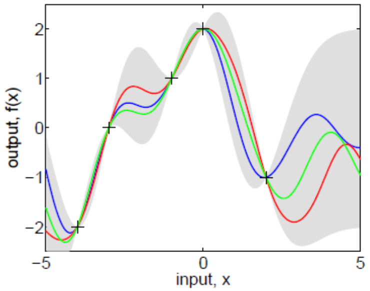
(Variational Auto) Encoders



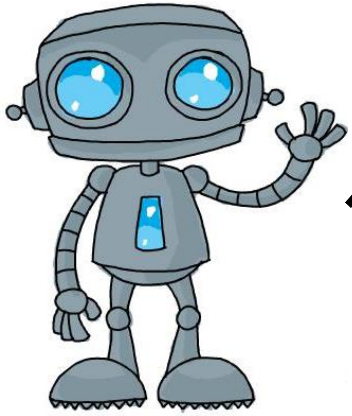
Statistical Models



Gaussian Process



Definition Model-Based



agent

Algorithm 1 Model-based reinforcement learning

Input: state sample procedure d

Input: model m

Input: policy π

Input: predictions v

Input: environment $\mathcal{E}$

Get initial state $s \leftarrow \mathcal{E}$

for iteration $\in \{1, 2, \ldots, K\}$ **do**

    for interaction $\in \{1, 2, \ldots, M\}$ **do**

        Interactions with the real Environment

    end for

    for planning step $\in \{1, 2, \ldots, P\}$ **do**

        Take advantage of the model

    end for

end for

Van Hasselt et al. 2019

Definition Model-Based

Algorithm 1 Model-based reinforcement learning

Input: state sample procedure d
Input: model m
Input: policy π
Input: predictions v
Input: environment $\mathcal{E}$
Get initial state $s \leftarrow \mathcal{E}$
for iteration $\in \{1, 2, \ldots, K\}$ **do**
    for interaction $\in \{1, 2, \ldots, M\}$ **do**
        Generate action: $a \leftarrow \pi(s)$
        Generate reward, next state: $r, s' \leftarrow \mathcal{E}(a)$
        $m, d \leftarrow$ UPDATEMODEL(s, a, r, s')
        $\pi, v \leftarrow$ UPDATEAGENT(s, a, r, s')
        Update current state: $s \leftarrow s'$
    end for
    for planning step $\in \{1, 2, \ldots, P\}$ **do**
        Take advantage of the model
    end for
end for

Van Hasselt et al. 2019

Definition Model-Based

Algorithm 1 Model-based reinforcement learning

Input: state sample procedure d
Input: model m
Input: policy π
Input: predictions v
Input: environment $\mathcal{E}$
Get initial state $s \leftarrow \mathcal{E}$
for iteration $\in \{1, 2, \ldots, K\}$ **do**
    for interaction $\in \{1, 2, \ldots, M\}$ **do**
        Generate action: $a \leftarrow \pi(s)$
        Generate reward, next state: $r, s' \leftarrow \mathcal{E}(a)$
        $m, d \leftarrow \textsc{UpdateModel}(s, a, r, s')$
        $\pi, v \leftarrow \textsc{UpdateAgent}(s, a, r, s')$
        Update current state: $s \leftarrow s'$
    end for
    for planning step $\in \{1, 2, \ldots, P\}$ **do**
        Generate state, action $\tilde{s}, \tilde{a} \leftarrow d$
        Generate reward, next state: $\tilde{r}, \tilde{s}' \leftarrow m(\tilde{s}, \tilde{a})$
        $\pi, v \leftarrow \textsc{UpdateAgent}(\tilde{s}, \tilde{a}, \tilde{r}, \tilde{s}')$
    end for
end for

Van Hasselt et al. 2019

Definition (purely) Model-Based

Algorithm 1 Model-based reinforcement learning

```
Input: state sample procedure d
Input: model m
Input: policy π
Input: predictions v
Input: environment ℰ
Get initial state s ← ℰ
for iteration ∈ {1, 2, . . . , K} do
    for interaction ∈ {1, 2, . . . , M} do
        Generate action: a ← π(s)
        Generate reward, next state: r, s′ ← ℰ(a)
        m, d ← UpdateModel(s, a, r, s′)

        Update current state: s ← s′
    end for
    for planning step ∈ {1, 2, . . . , P} do
        Generate state, action s̃, ã ← d
        Generate reward, next state: r̃, s̃′ ← m(s̃, ã)
        π, v ← UpdateAgent(s̃, ã, r̃, s̃′)
    end for
end for
```

Van Hasselt et al. 2019

Definition Model-Free

Algorithm 1 Model-based reinforcement learning

Input: policy π
Input: predictions v
Input: environment $\mathcal{E}$
Get initial state $s \leftarrow \mathcal{E}$
for iteration $\in \{1, 2, \ldots, K\}$ **do**
  for interaction $\in \{1, 2, \ldots, M\}$ **do**
    Generate action: $a \leftarrow \pi(s)$
    Generate reward, next state: $r, s' \leftarrow \mathcal{E}(a)$
    $\pi, v \leftarrow \textsc{UpdateAgent}(s, a, r, s')$
    Update current state: $s \leftarrow s'$
  end for
end for

Van Hasselt et al. 2019

Definition Model-Based

DQN is model-based due to it's replay buffer

Algorithm 1 Model-based reinforcement learning

Input: state sample procedure d
Input: model m
Input: policy π
Input: predictions v
Input: environment $\mathcal{E}$
Get initial state $s \leftarrow \mathcal{E}$
for iteration $\in \{1, 2, \ldots, K\}$ **do**
    for interaction $\in \{1, 2, \ldots, M\}$ **do**
        Interactions with the real Environment
    end for
    for planning step $\in \{1, 2, \ldots, P\}$ **do**
        Take advantage of the model
    end for
end for

Van Hasselt et al. 2019

Definition Model-Based

The fourth and final element of some reinforcement learning systems is a *model* of the environment. This is something that mimics the behavior of the environment, or more generally, that allows inferences to be made about how the environment will behave.

Sutton & Barto 2015

Definition Model-Based

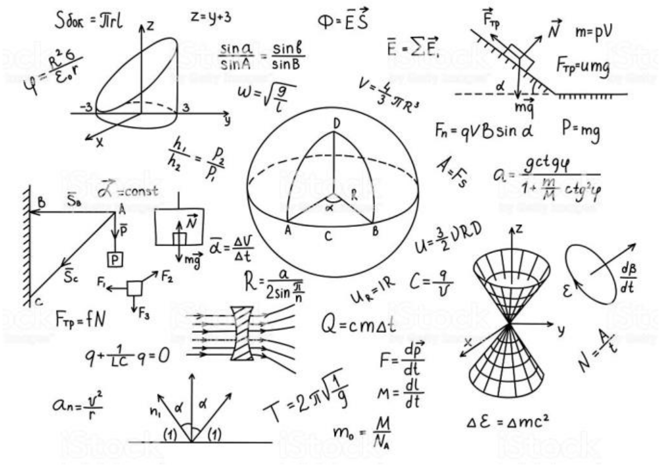
The fourth and final element of some reinforcement learning systems is a *model* of the environment. This is something that mimics the behavior of the environment, or more generally, that allows inferences to be made about how the environment will behave.

Again, DQN's replay buffer fits this description

Sutton & Barto 2015

Distinction to normal Control Problems

Distinction to normal Control Problems



vs

Control Problems

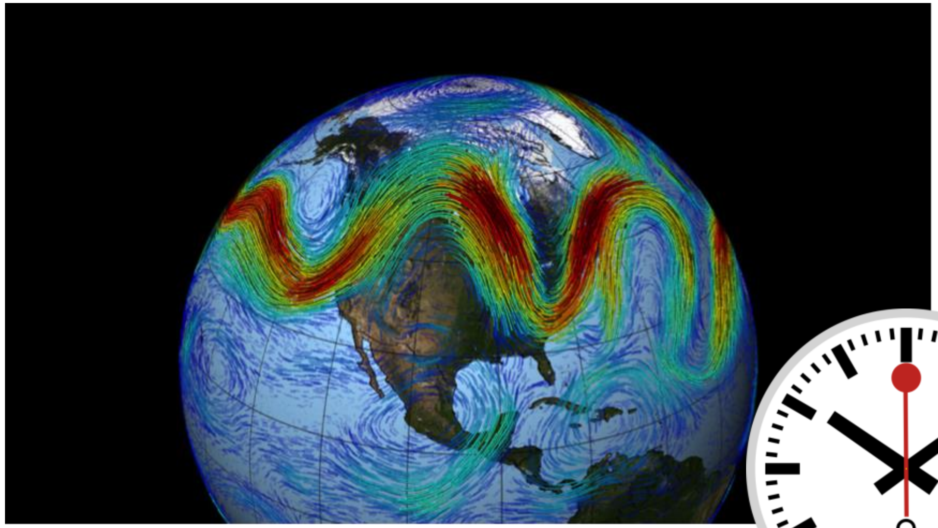
Model Based RL

Why I want a model

Why I want a model – Sample efficiency

$

Save exploration



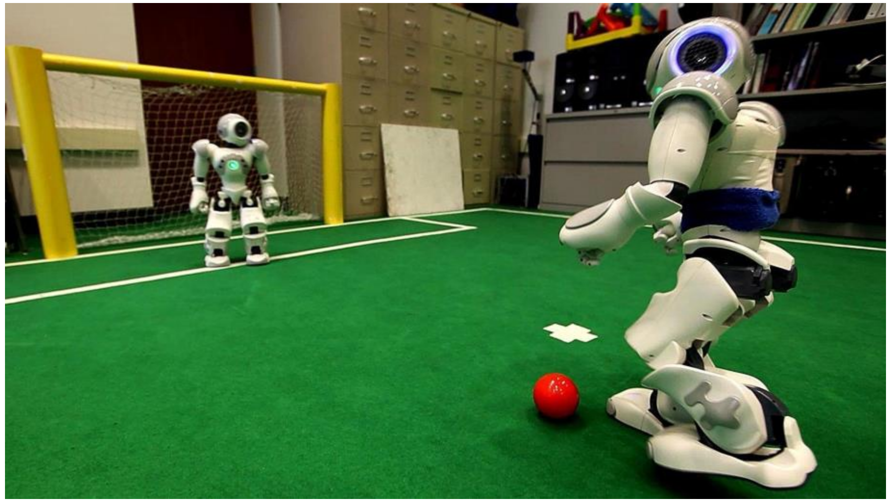
High computational cost

Create Samples with Model

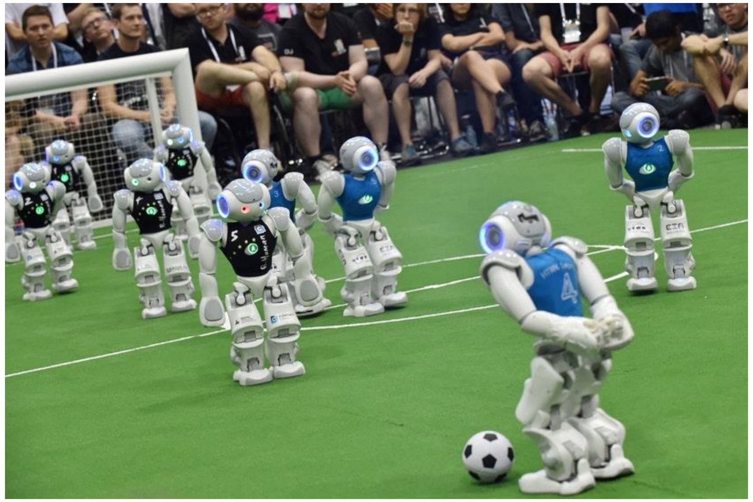
Why I want a model – Planning

Learn a model and then use a planning algorithm

Why I want a model – Transfer of Knowledge



Learn how to kick the ball



Concentrate on teamplay

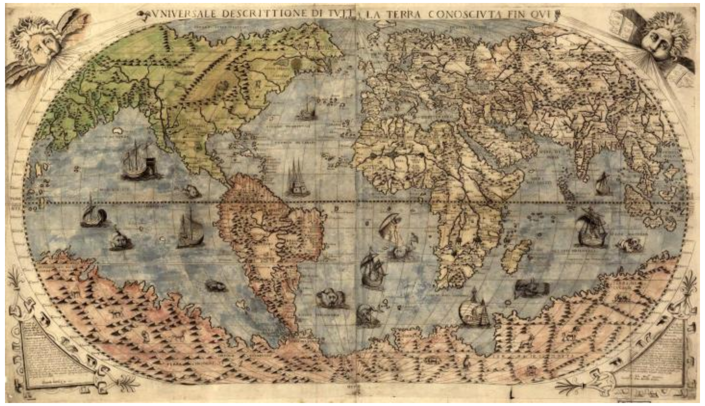
The effect of Planning Shape On Dyna-style planning in High-dimensional State Spaces

(Zacharias Holland et al. 2018, arXiv:1806.01825)

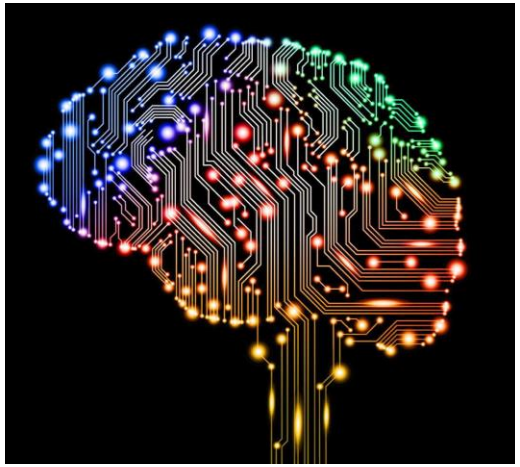
How does Dyna perform at Arcade games in a Deep Learning Setting?

Zacharias Holland et al. 2018

Important parts



Model

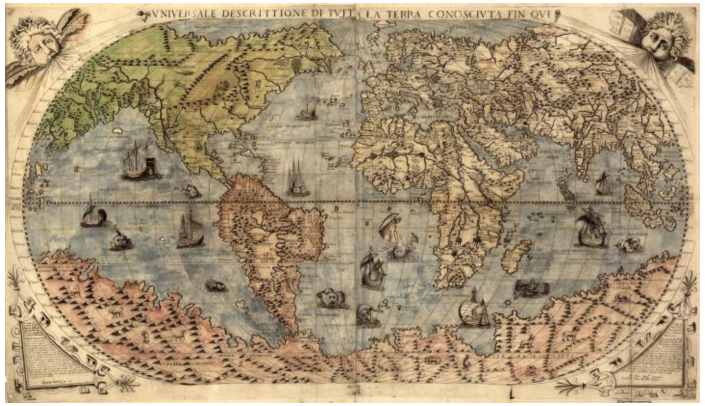


Learner

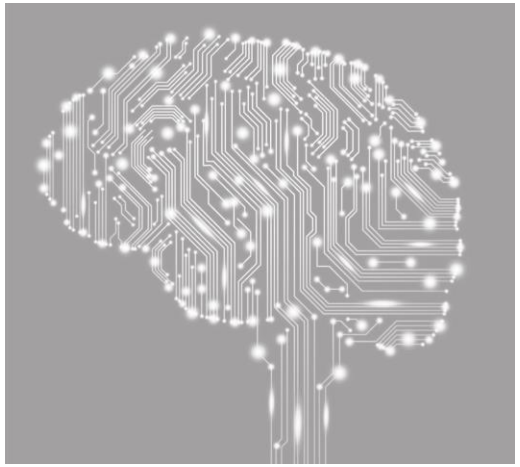
Roll-out Shapes

Zacharias Holland et al. 2018

Important parts



Model

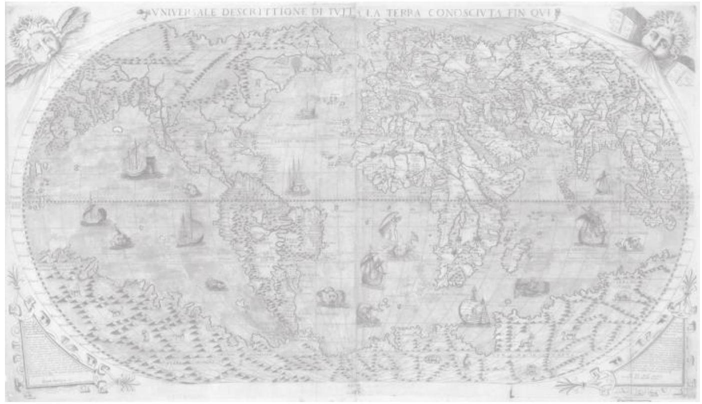


Learner

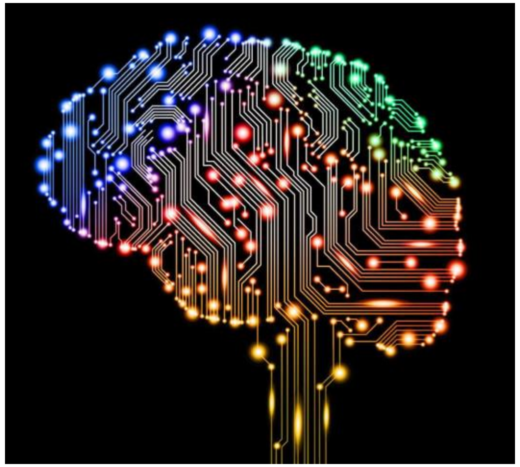
Roll-out Shapes

Zacharias Holland et al. 2018

Important parts



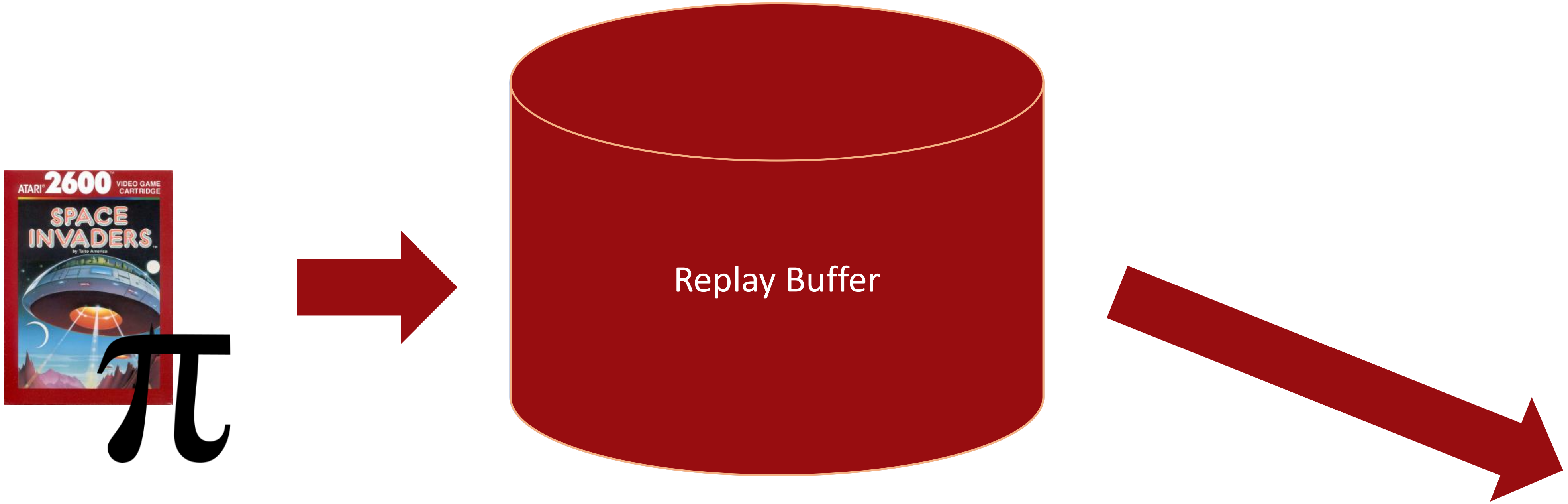
Model



Learner

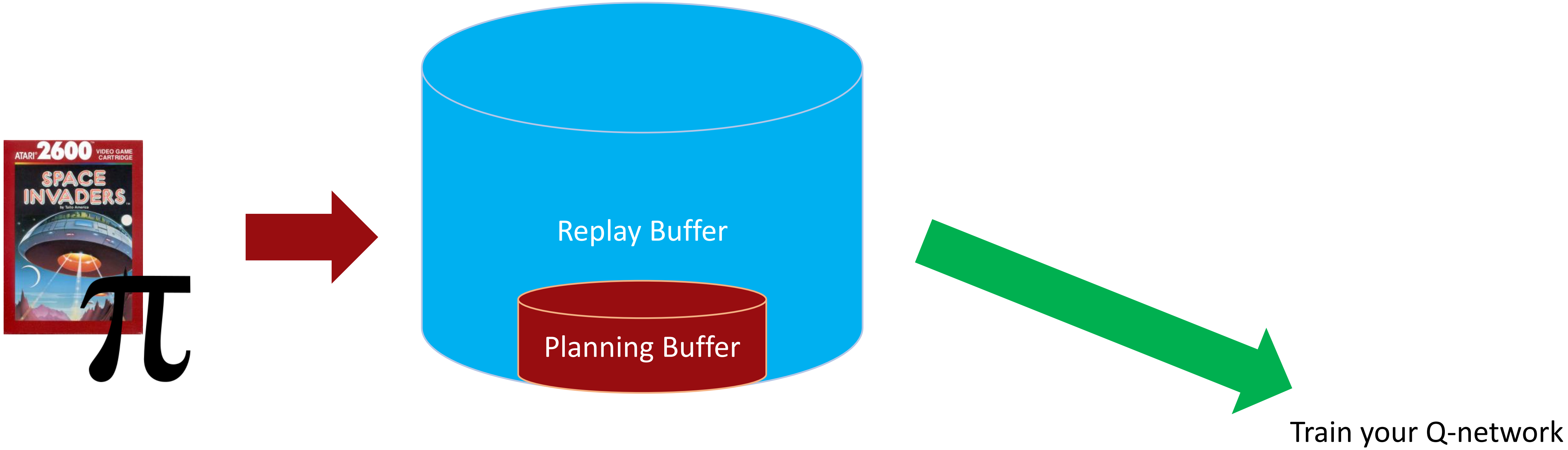
Roll-out Shapes

Zacharias Holland et al. 2018

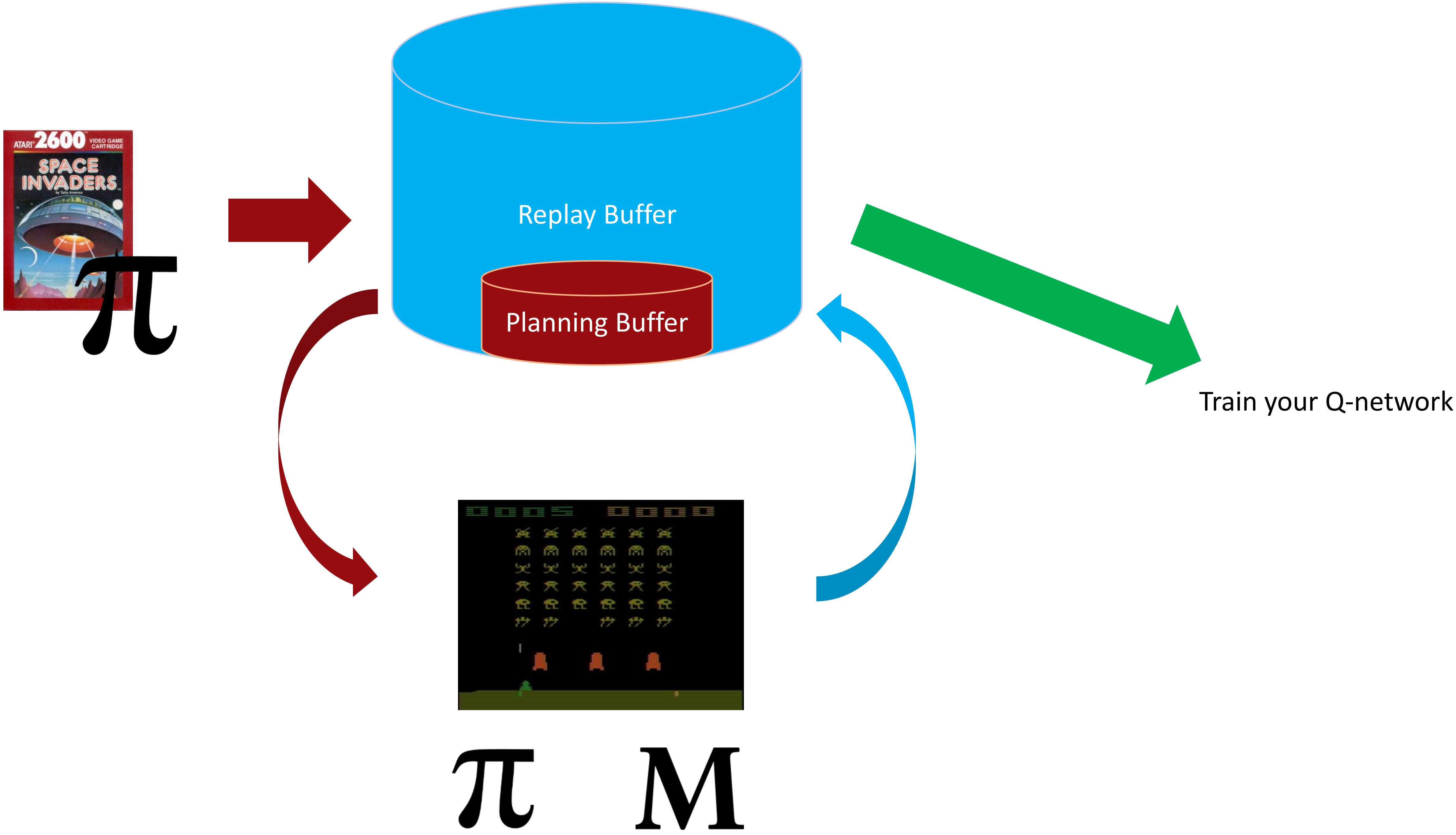


Train your Q-network

Zacharias Holland et al. 2018

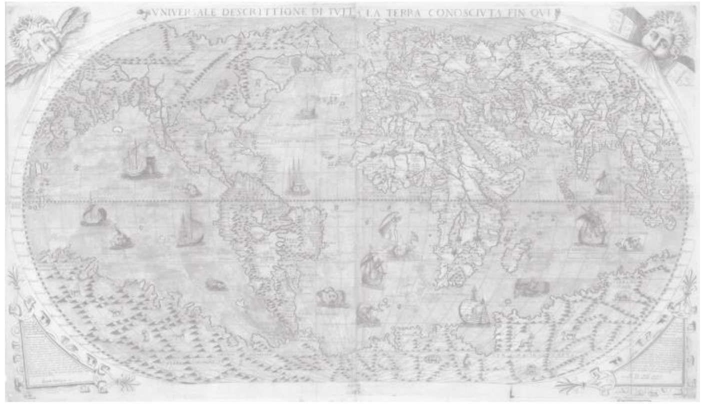


Zacharias Holland et al. 2018

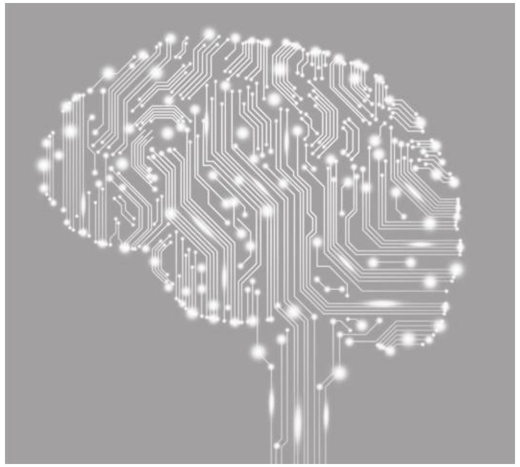


Zacharias Holland et al. 2018

Important parts



Model

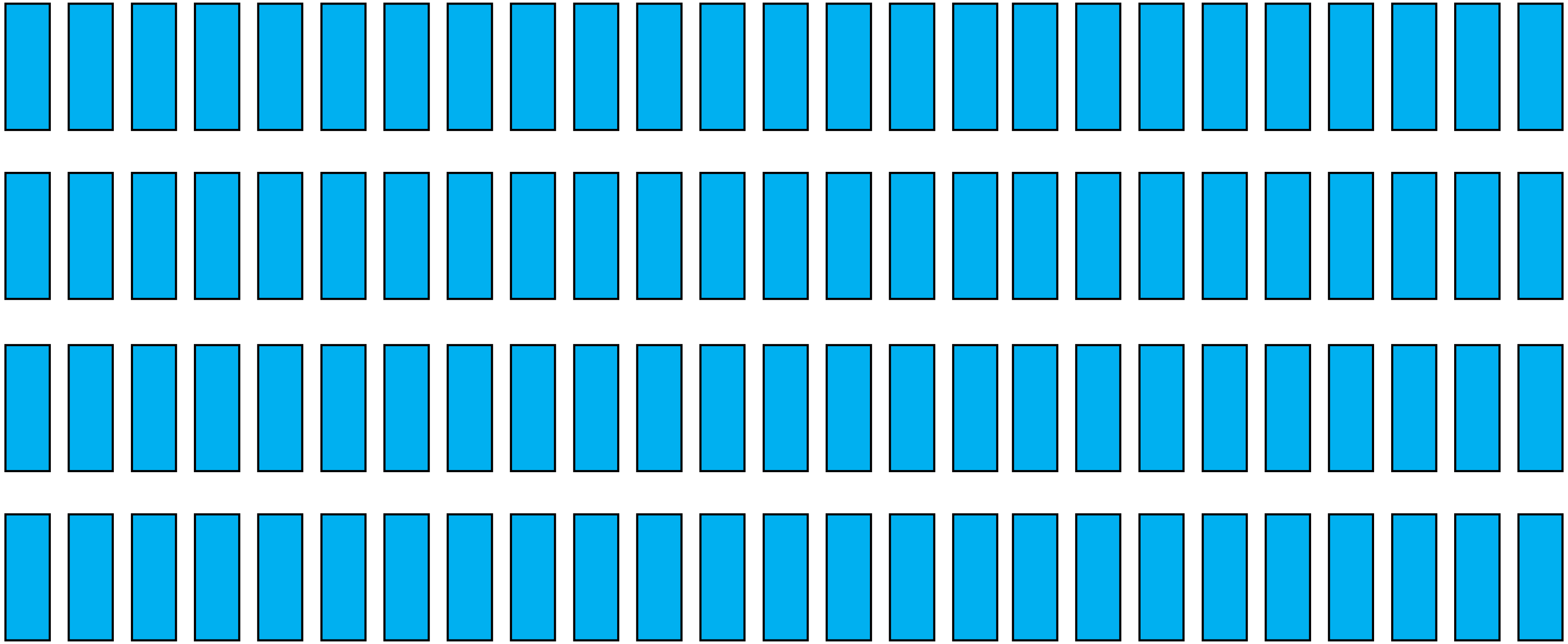


Learner

Roll-out Shapes

Zacharias Holland et al. 2018

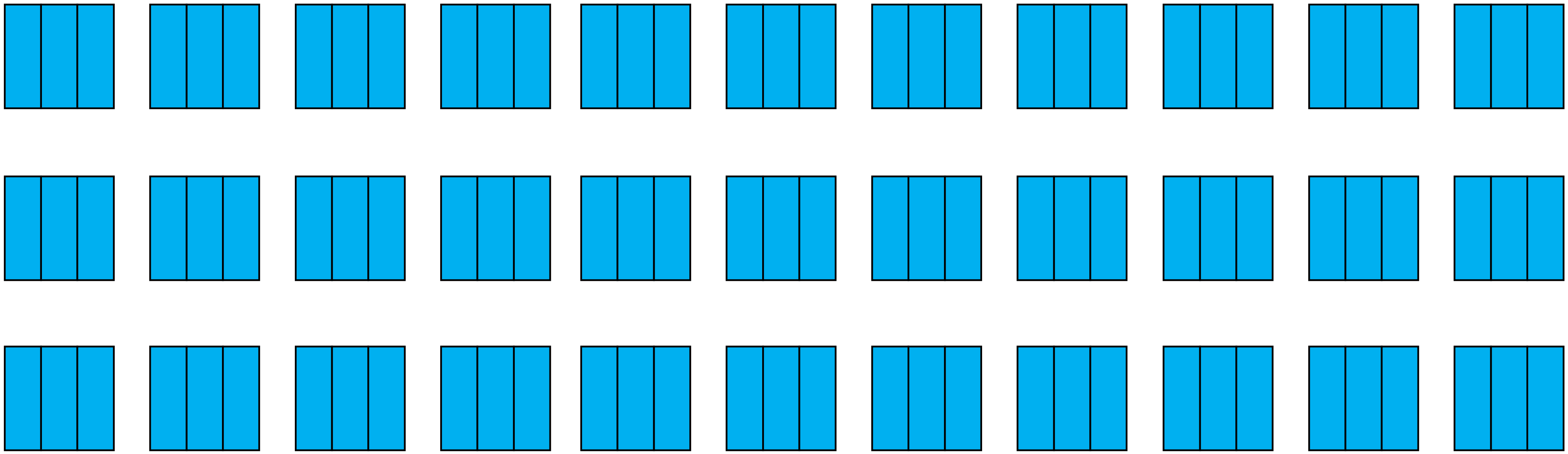
Roll-out Shapes



100 x 1

Zacharias Holland et al. 2018

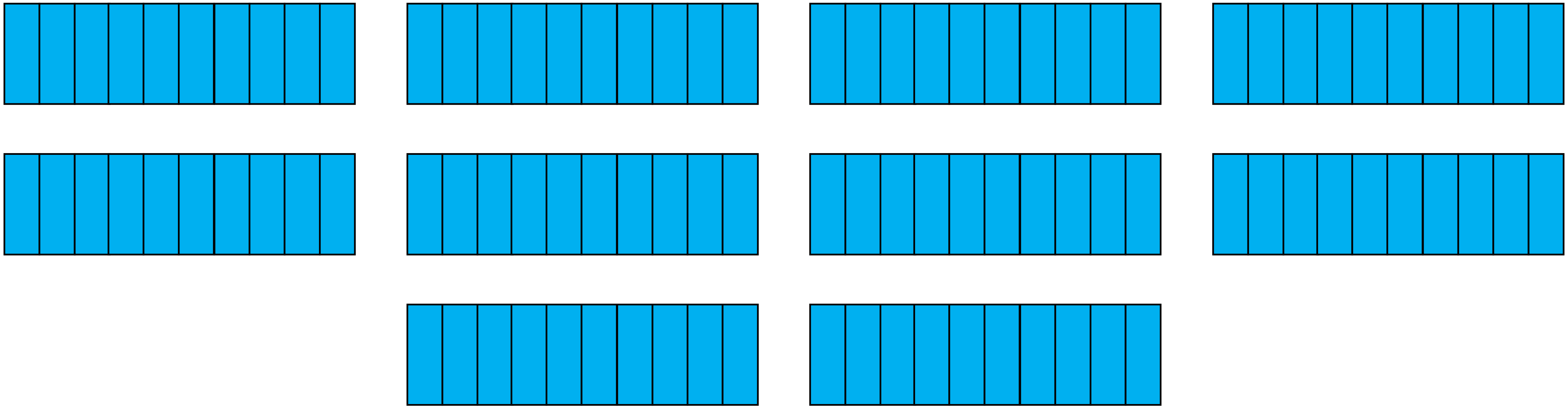
Roll-out Shapes



33 x 3

Zacharias Holland et al. 2018

Roll-out Shapes

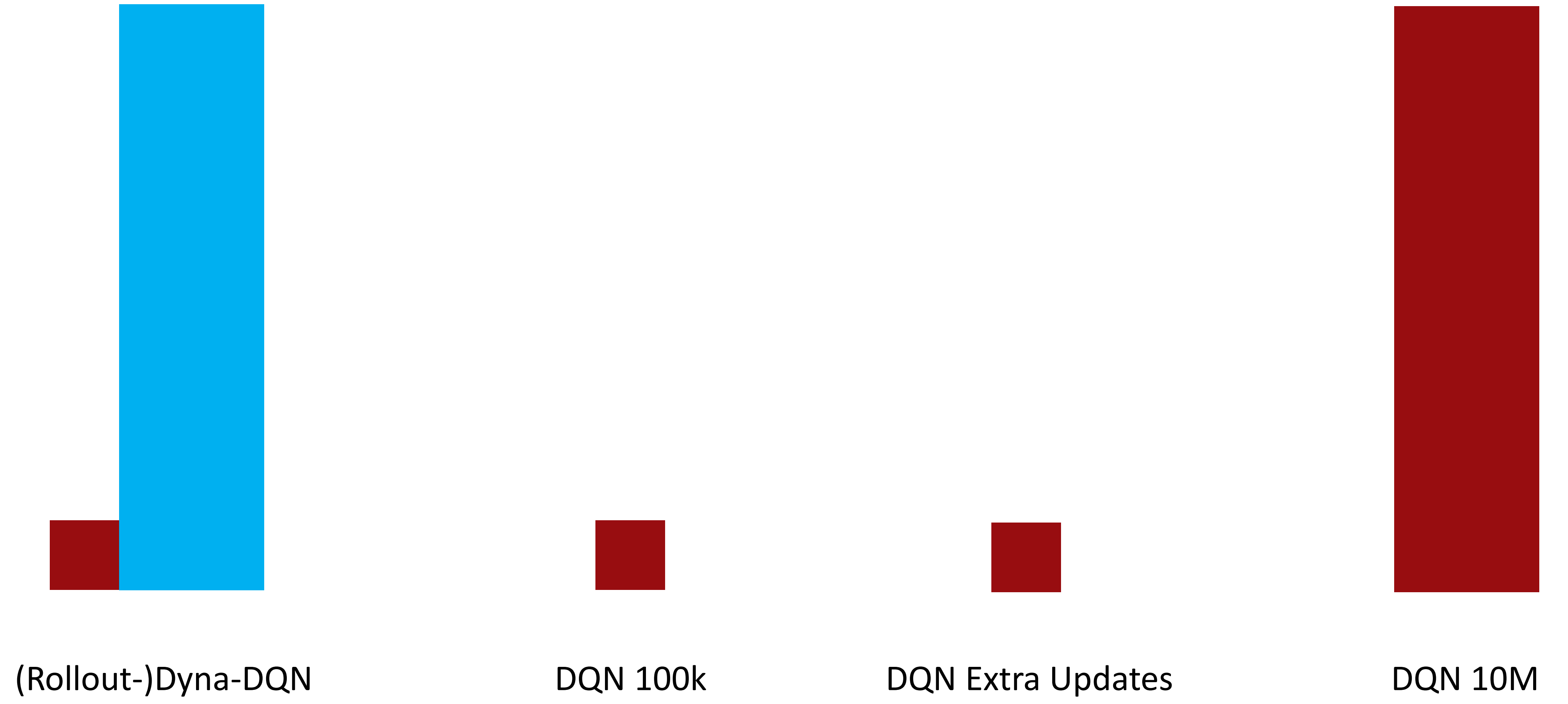


10 x 10

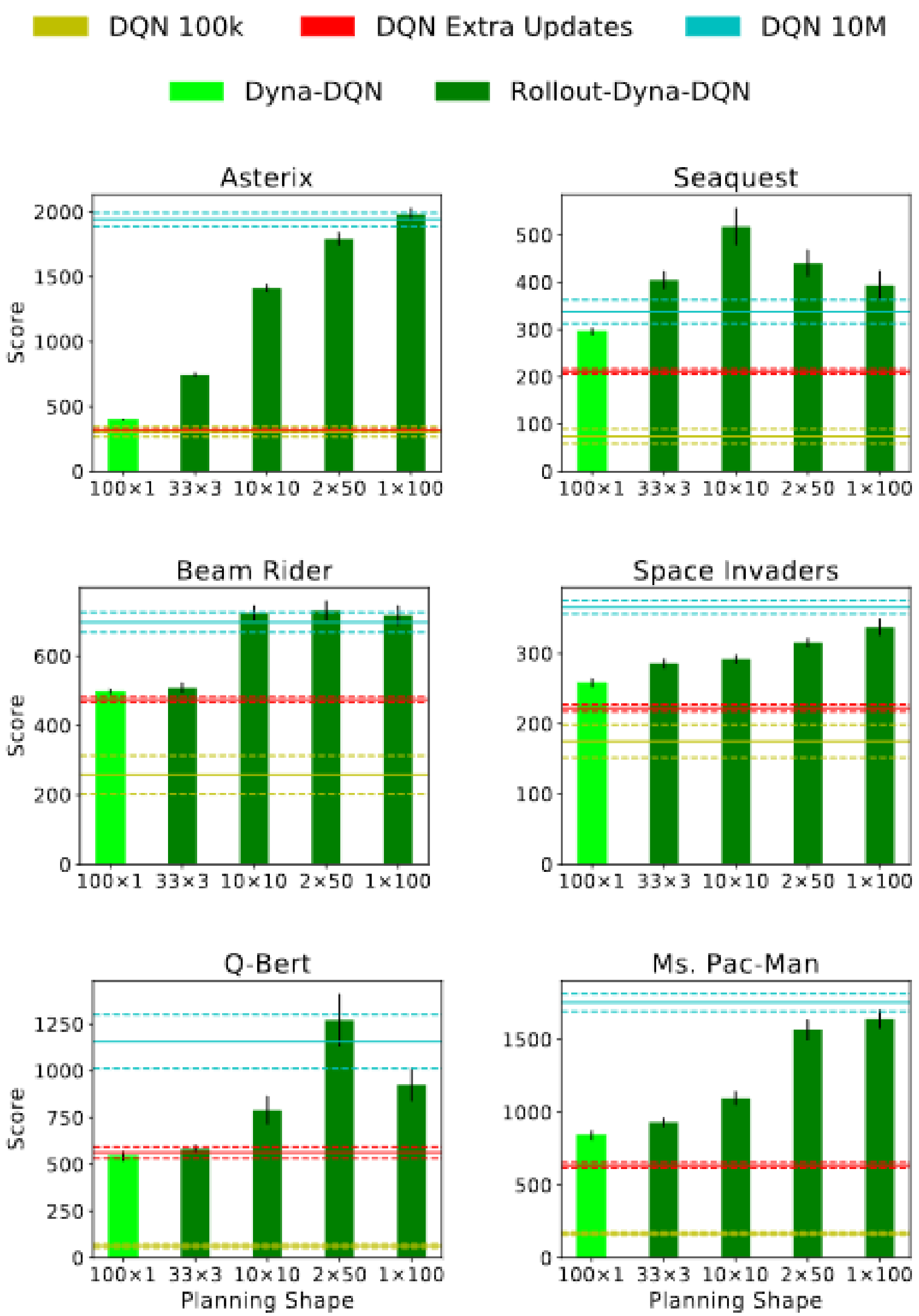
Zacharias Holland et al. 2018

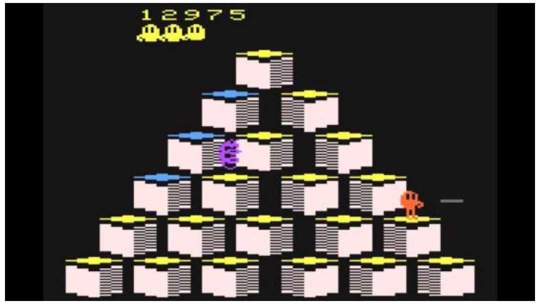
Experiments

Number of Samples & Benchmarks



Zacharias Holland et al. 2018





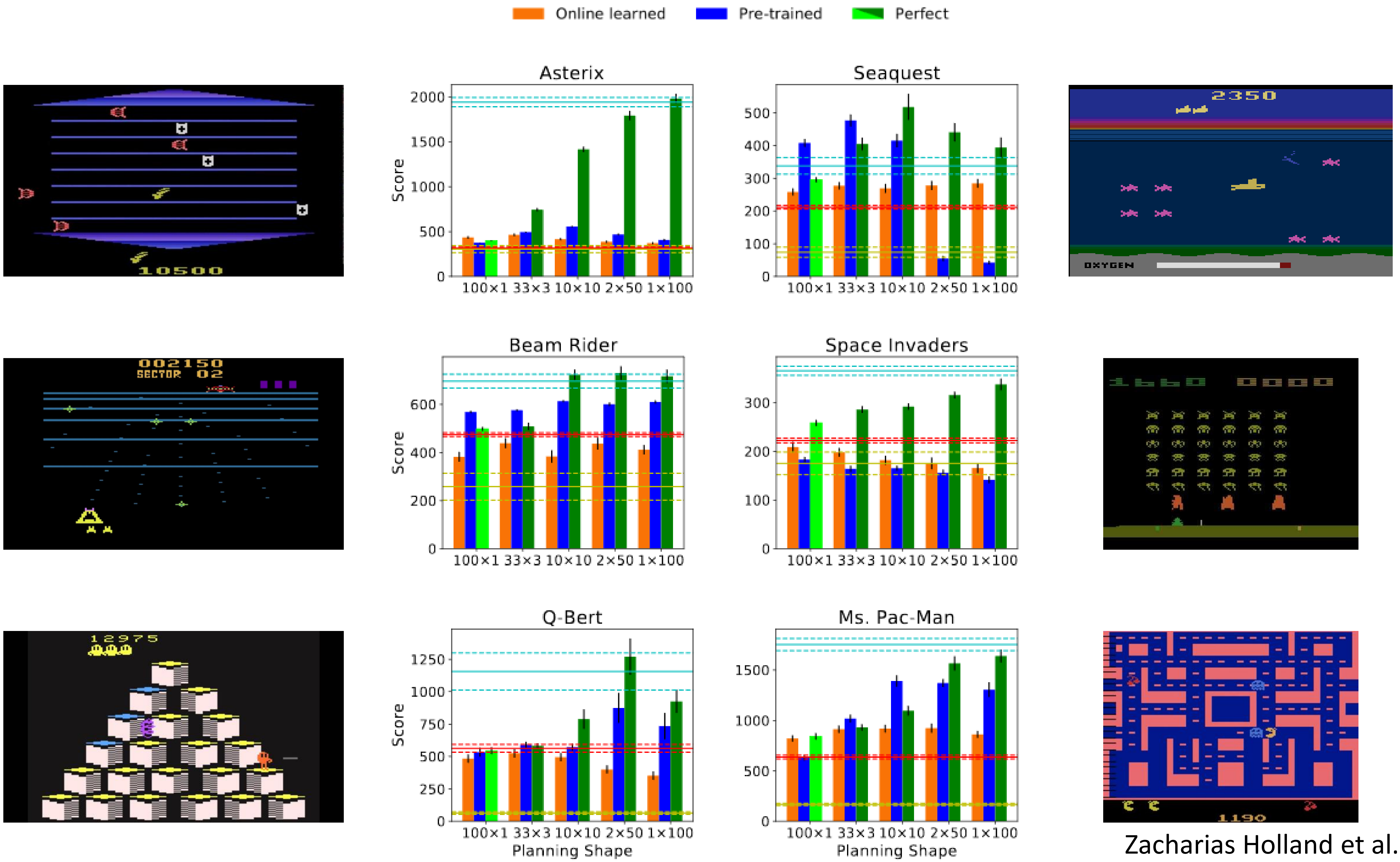
Zacharias Holland et al. 2018

Relaxing the model

One model is pretrained
on expert data

One model learns in an online
fashion

Zacharias Holland et al. 2018



Zacharias Holland et al. 2018

World Models

(Ha & Schmidhuber 2018, arXiv:1803.10122)

This thought bubble is our goal

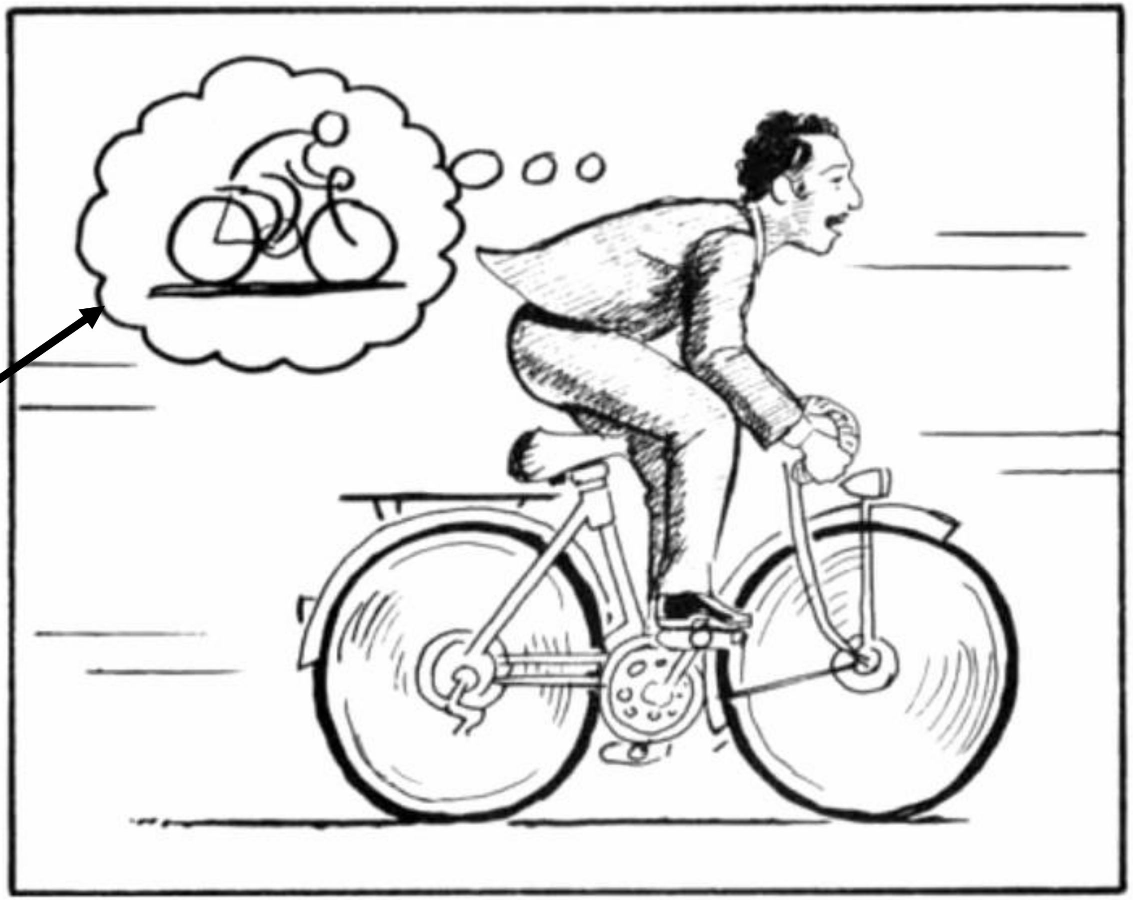
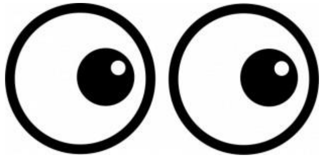


Figure 1. A World Model, from Scott McCloud's *Understanding Comics.* (McCloud, 1993; E, 2012)

Ha et al. 2018

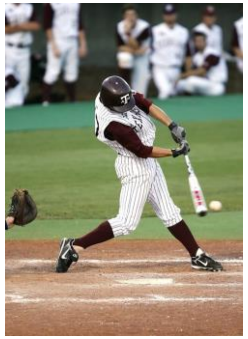
Learner over all

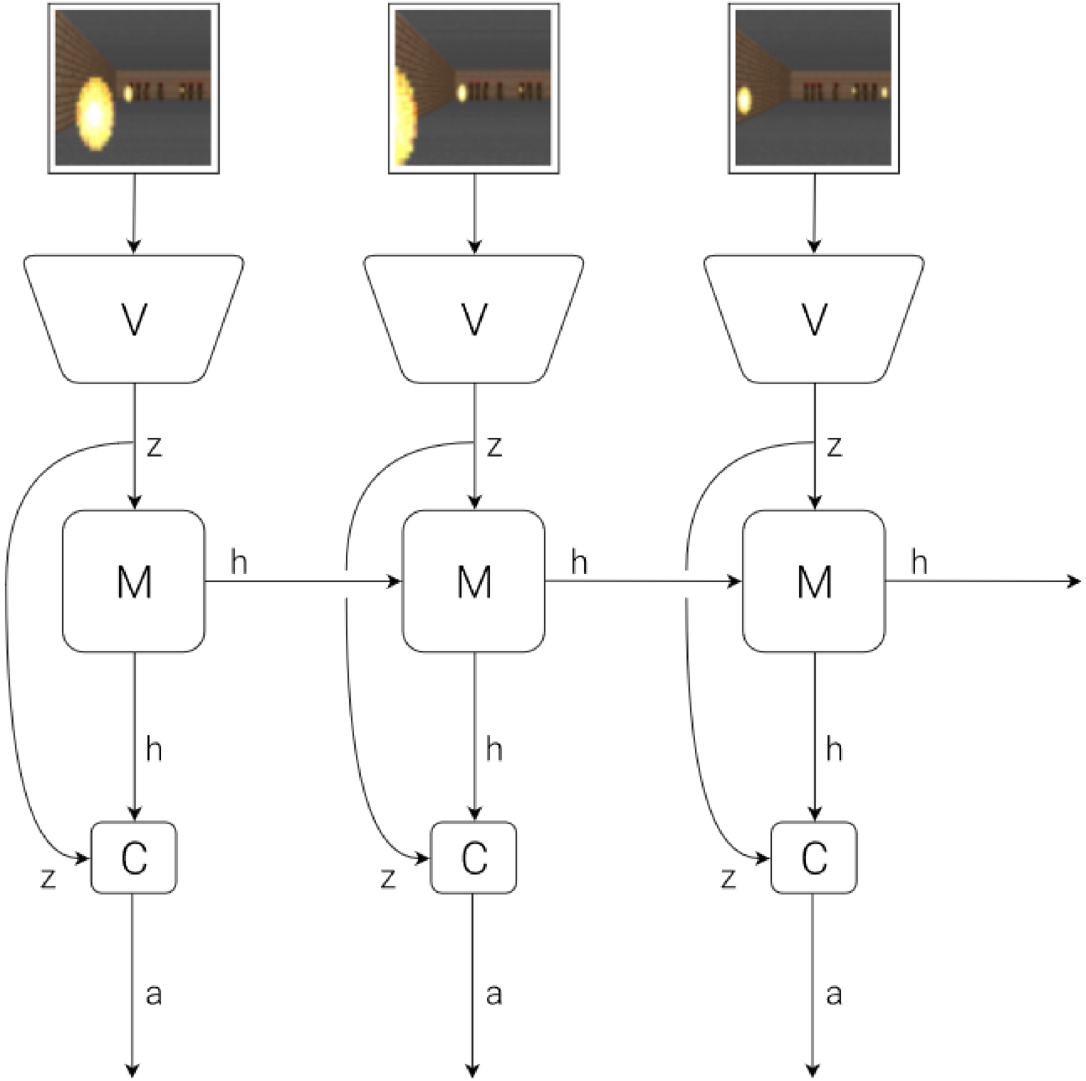
Vision



Memory

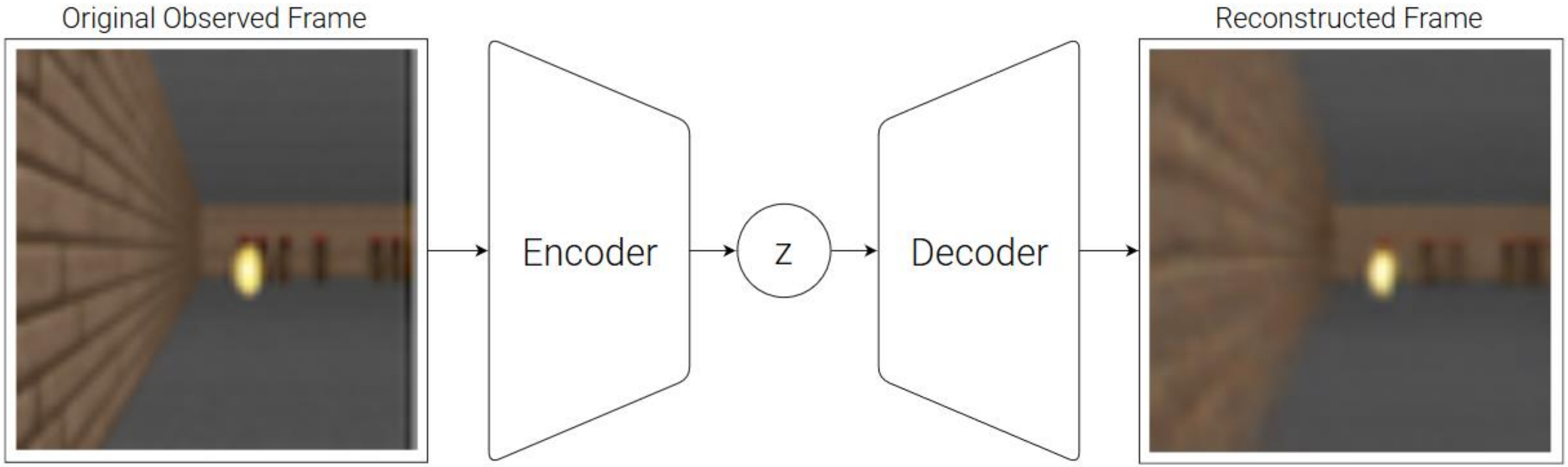
Controller





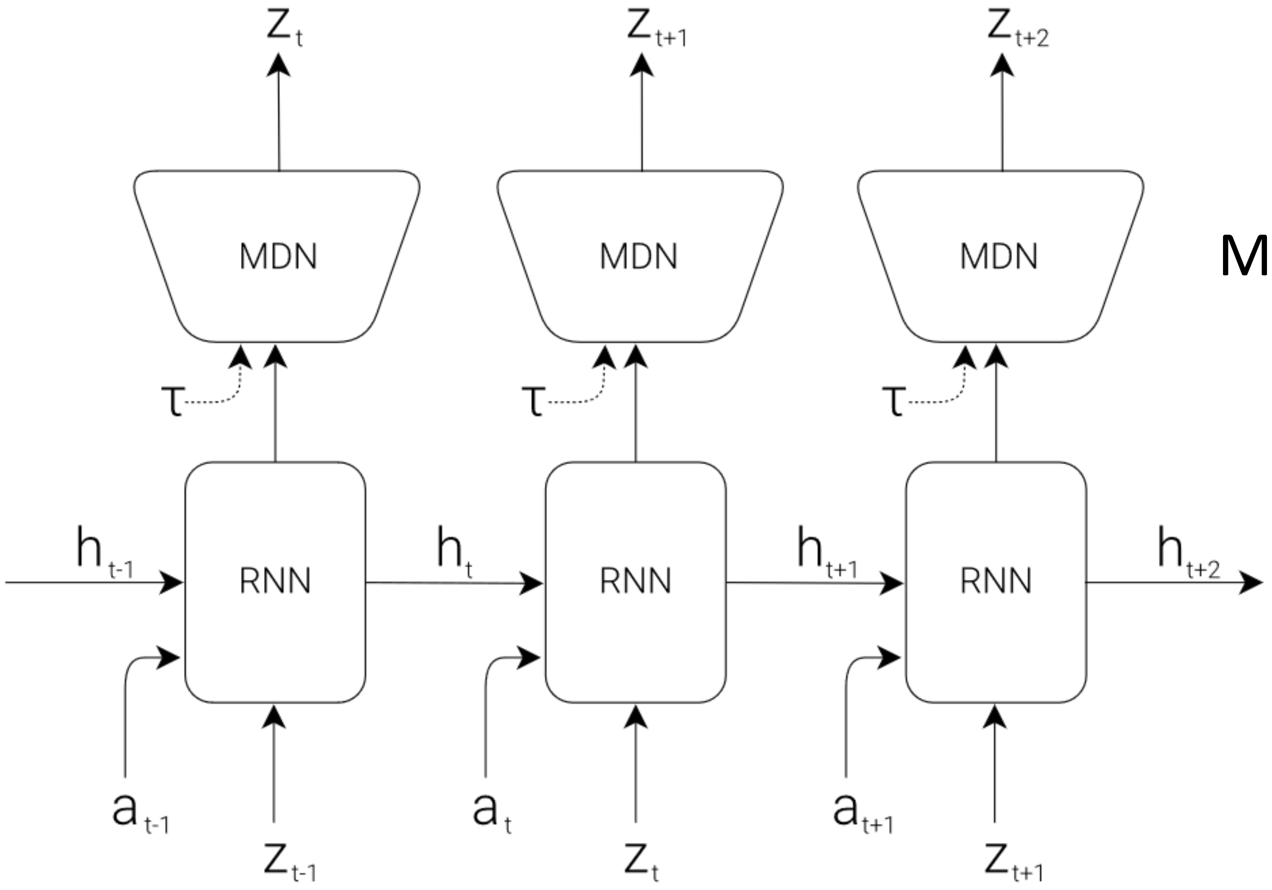
Ha et al. 2018

Vision (V) Model



Ha et al. 2018

Memory (M) Model

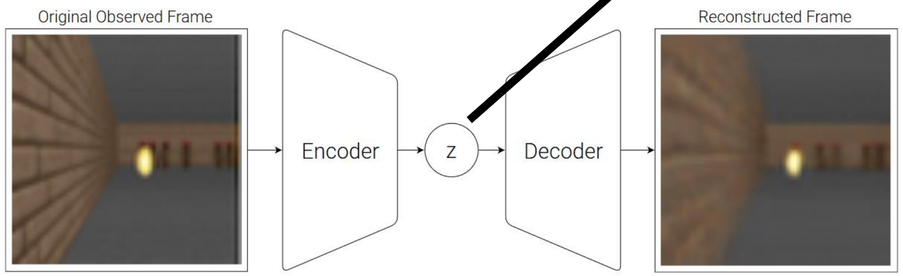


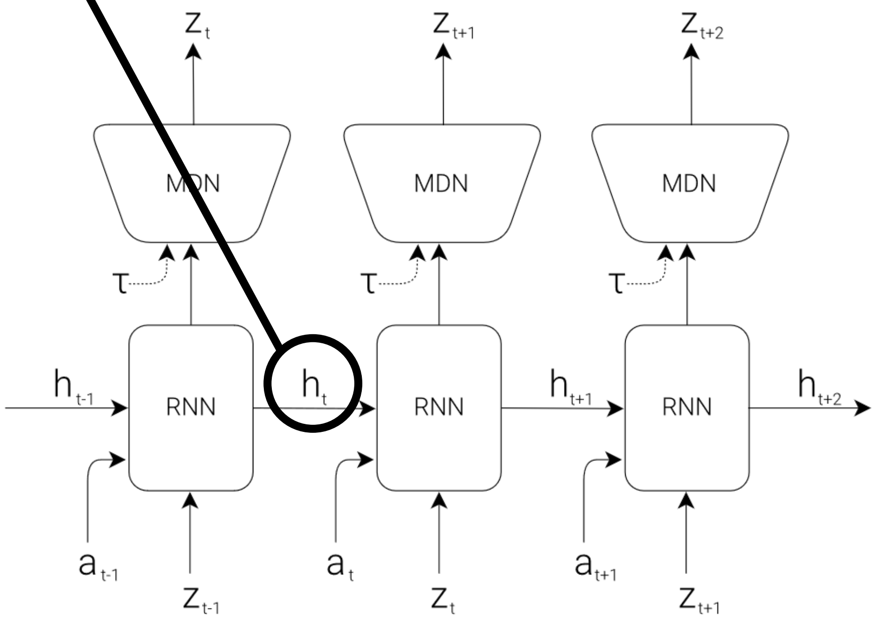
Mixture Density Network

Ha et al. 2018

Controller (C) Model

$$a_t = W_c \, [z_t \; h_t] \; + b_c$$





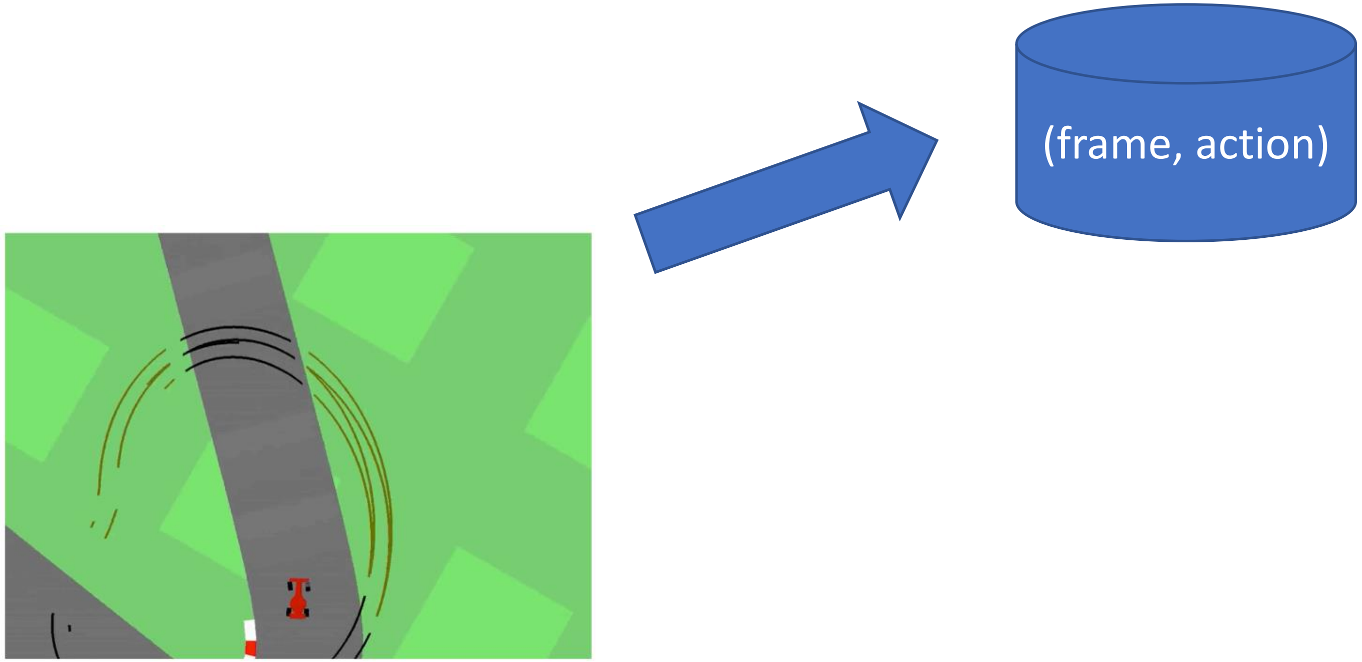
Ha et al. 2018

Experiments

Experiment 1 – Training the model

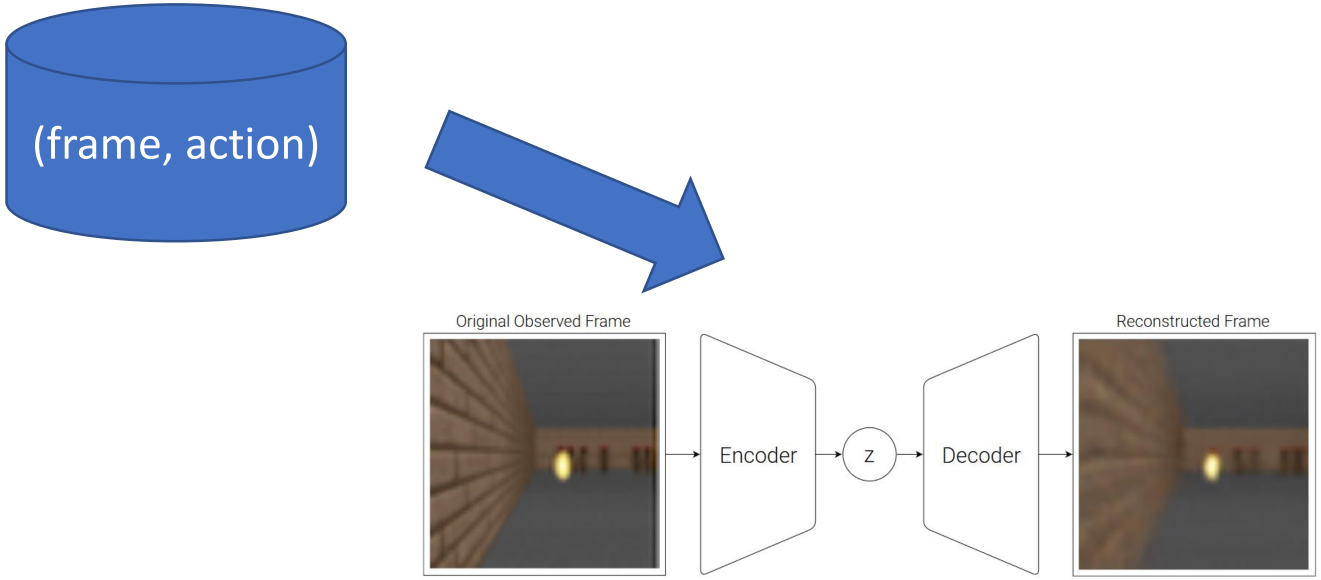
Ha et al. 2018

Experiment 1 – Training the model



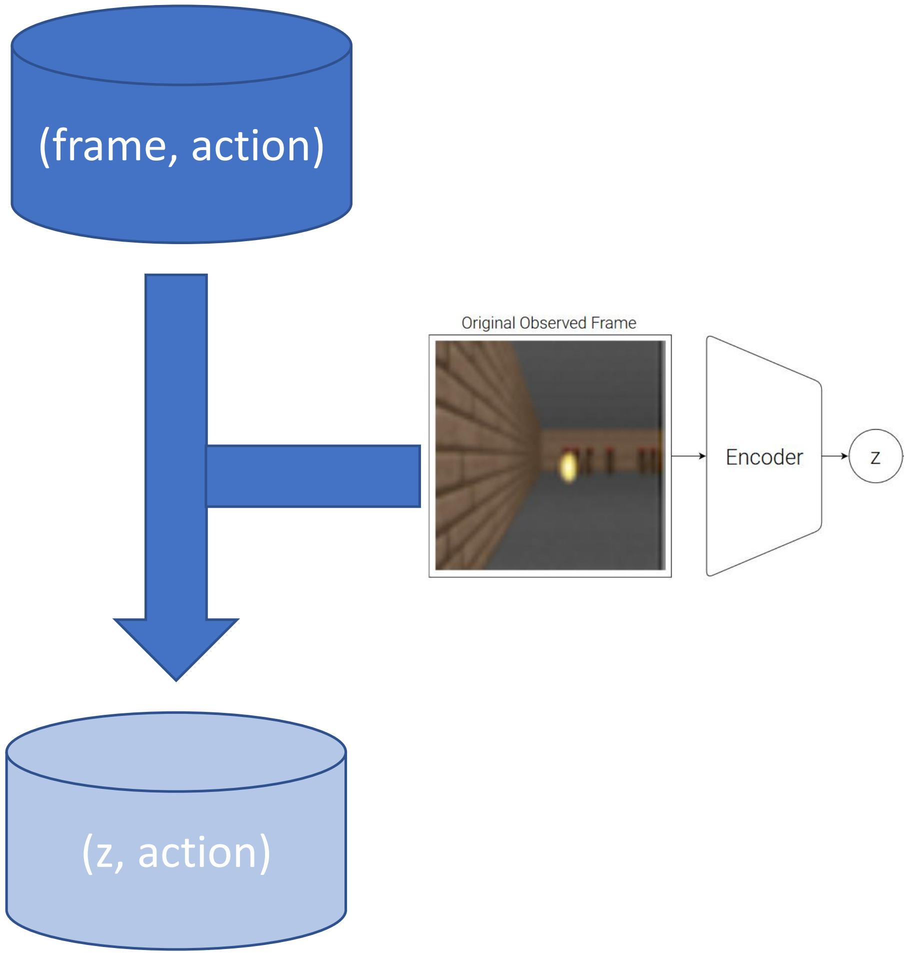
Ha et al. 2018

Experiment 1 – Training the model



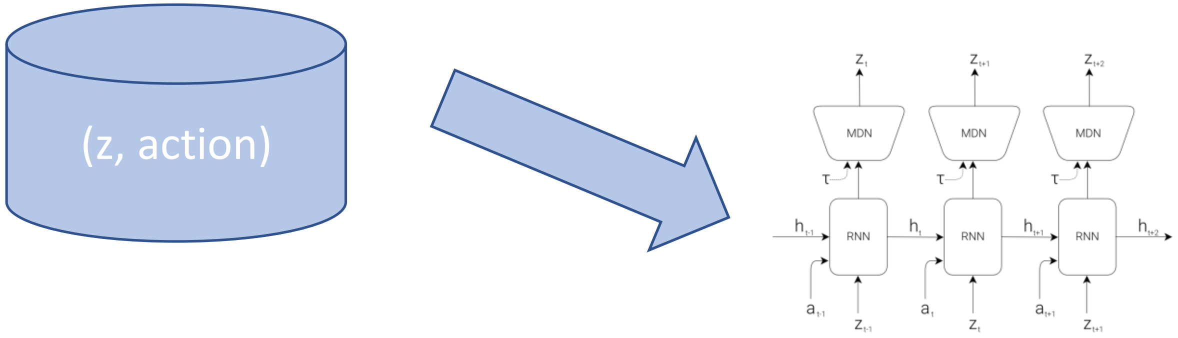
Ha et al. 2018

Experiment 1 – Training the model



Ha et al. 2018

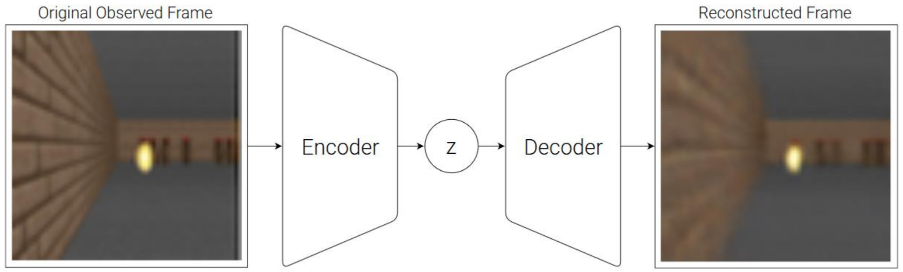
Experiment 1 – Training the model



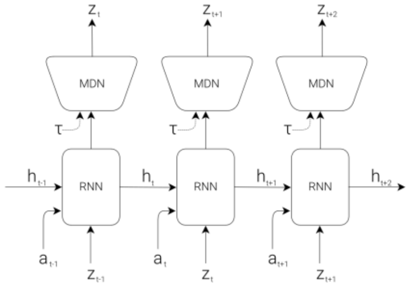
Ha et al. 2018

Experiment 1 – Training the model

(frame, action)



(z, action)

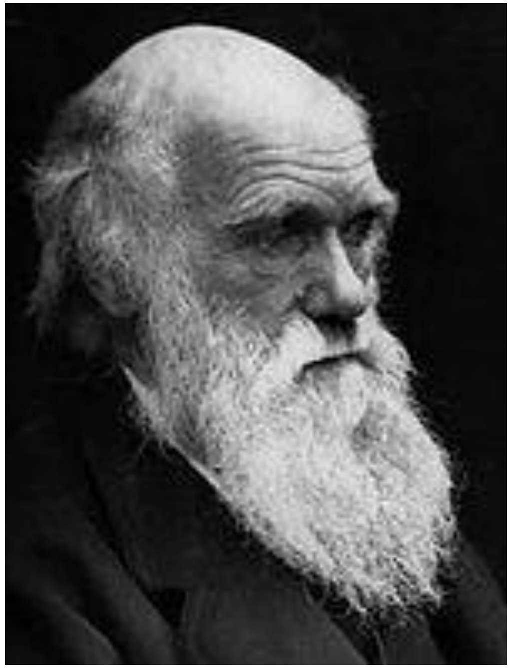


Ha et al. 2018

Experiment 1 – Training the Controller

$$a_t = W_c \, [z_t \; h_t] \; + b_c$$

Evolution!



Ha et al. 2018

Experiment 1

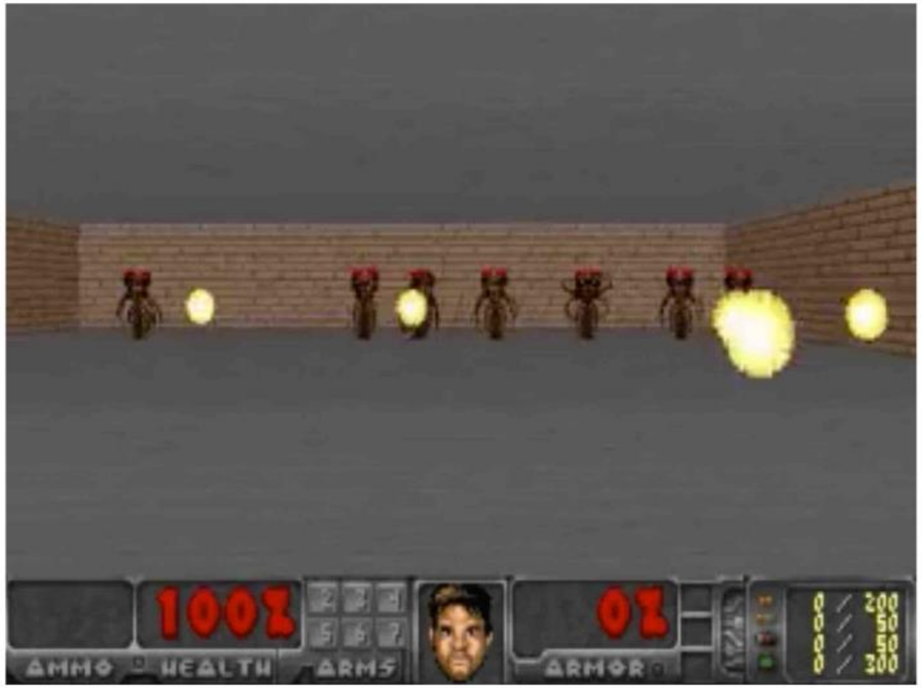
Removed memory Model

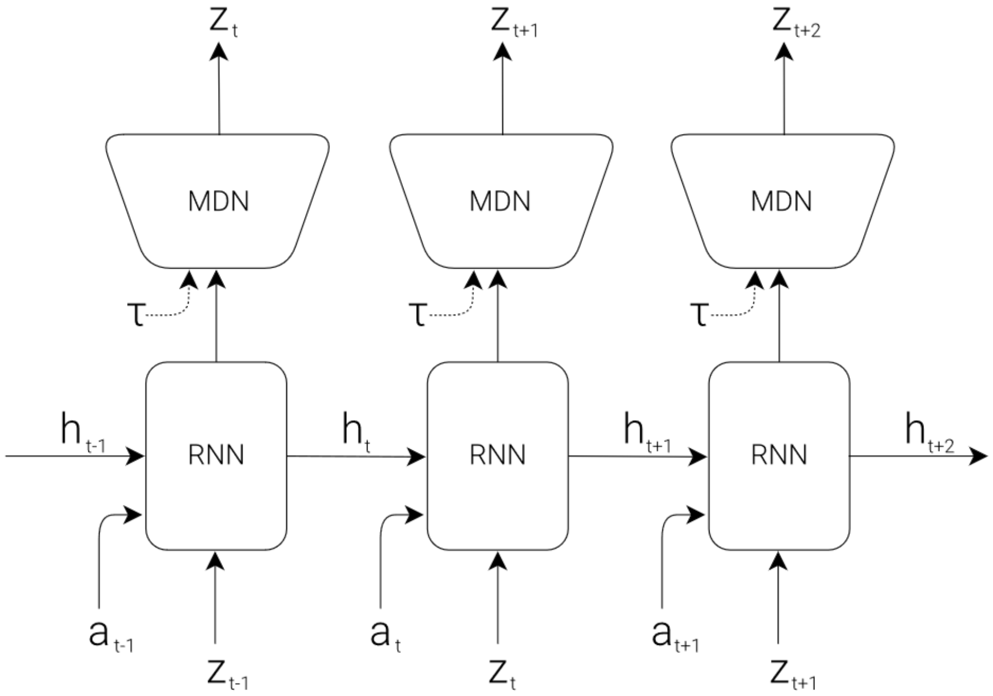
Removed memory Model, but with hidden layer in Controller

METHOD	AVG. SCORE
DQN (PRIEUR, 2017)	343 ± 18
A3C (CONTINUOUS) (JANG ET AL., 2017)	591 ± 45
A3C (DISCRETE) (KHAN & ELIBOL, 2016)	652 ± 10
CEOBILLIONAIRE (GYM LEADERBOARD)	838 ± 11
V MODEL	632 ± 251
V MODEL WITH HIDDEN LAYER	788 ± 141
FULL WORLD MODEL	$\mathbf{906 \pm 21}$

Ha et al. 2018

Experiment 2 – train in dream world





Ha et al. 2018

Experiment 2 – what about rewards?

Maximize survival time

Train MDN-RNN (M) to model $P(z_{t+1}, \underline{d_{t+1}} \mid a_t, z_t, h_t)$.

Memory model predicts death

Ha et al. 2018

Experiment 2 – Problems



Too low temperature -> no shooting
Too high temperature -> chaos

Controller exploits inaccuracies of the model

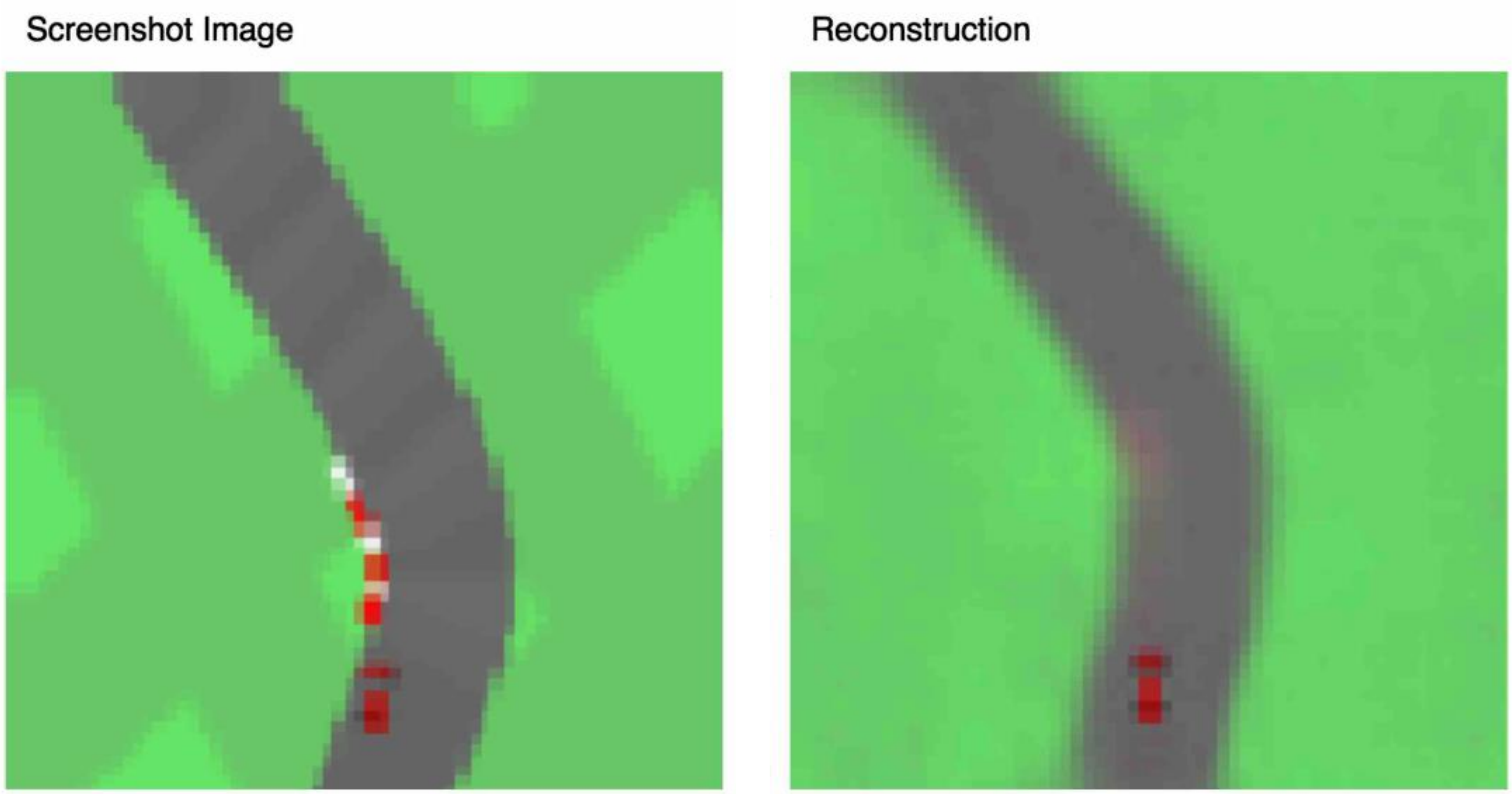
Ha et al. 2018

Experiment 2

TEMPERATURE τ	VIRTUAL SCORE	ACTUAL SCORE
0.10	2086 ± 140	193 ± 58
0.50	2060 ± 277	196 ± 50
1.00	1145 ± 690	868 ± 511
1.15	918 ± 546	1092 ± 556
1.30	732 ± 269	753 ± 139
RANDOM POLICY	N/A	210 ± 108
GYM LEADER	N/A	820 ± 58

Ha et al. 2018

World Models – Shortcoming



Latent representation z optimized for reconstruction and not for task solving

Ha et al. 2018

See you on Piazza