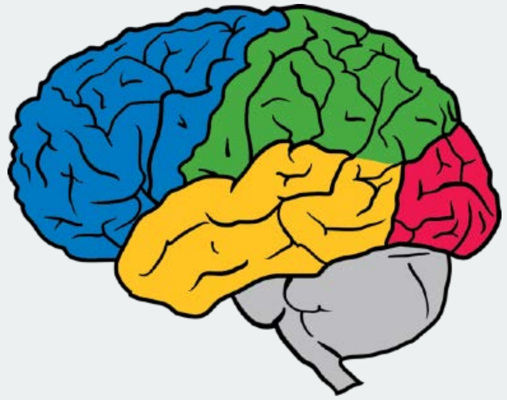
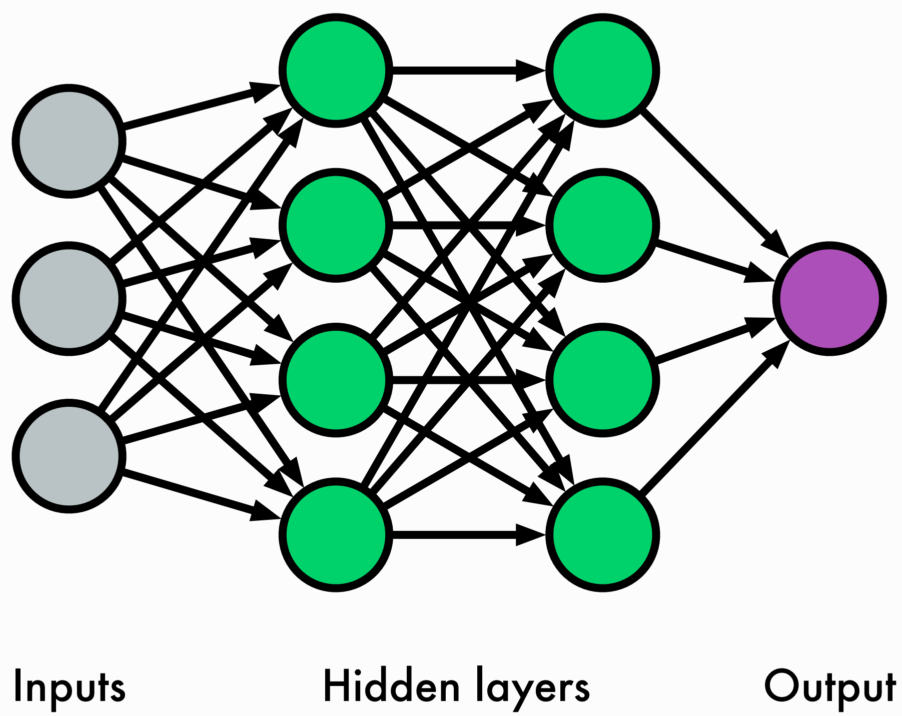
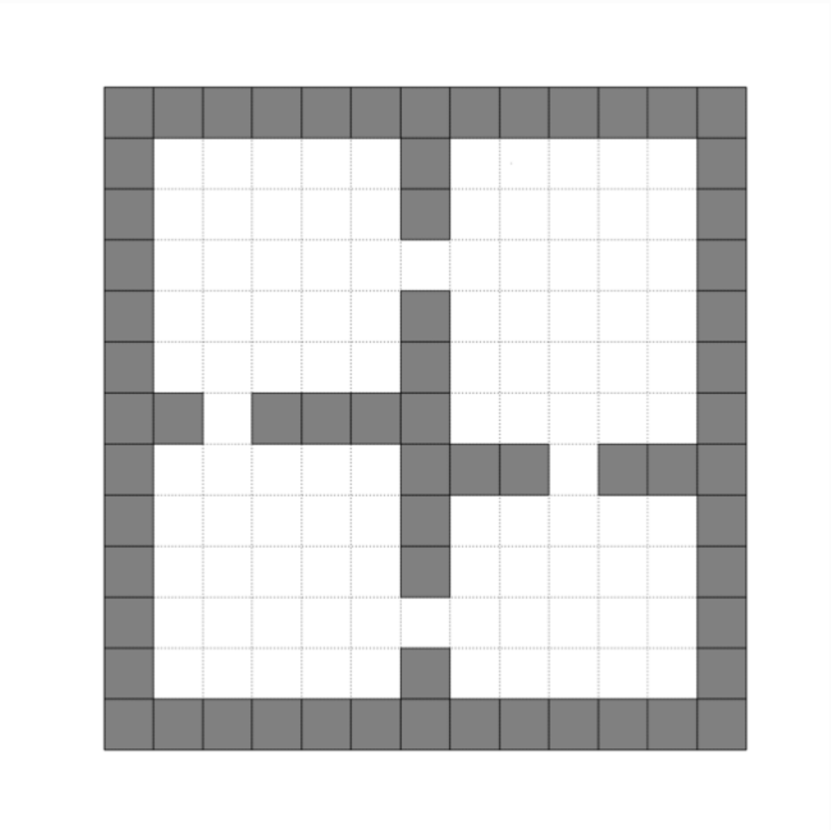
Deep Reinforcement Learning and the Atari 2600

Marc G. Bellemare
Google Brain



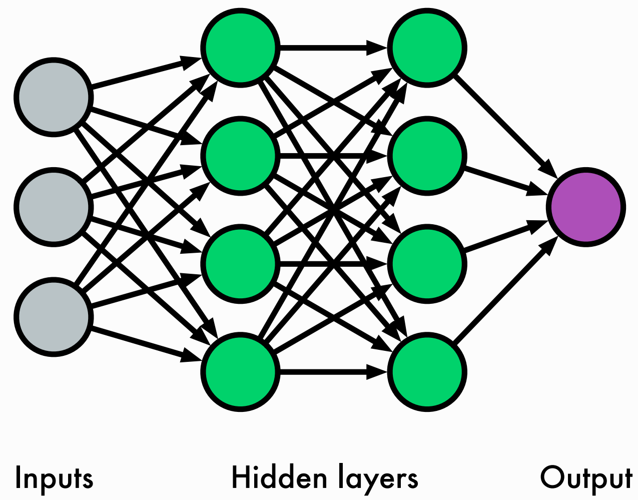
Deep Reinforcement learning

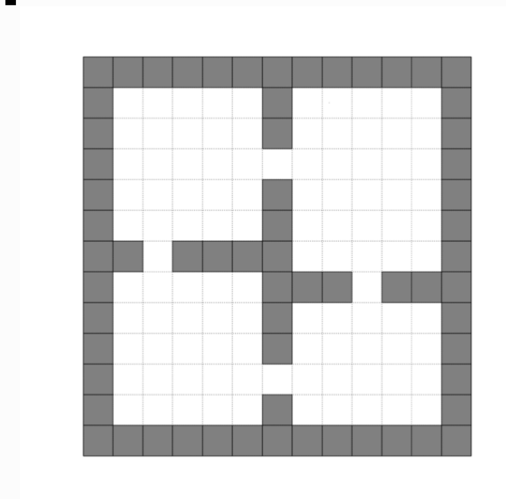


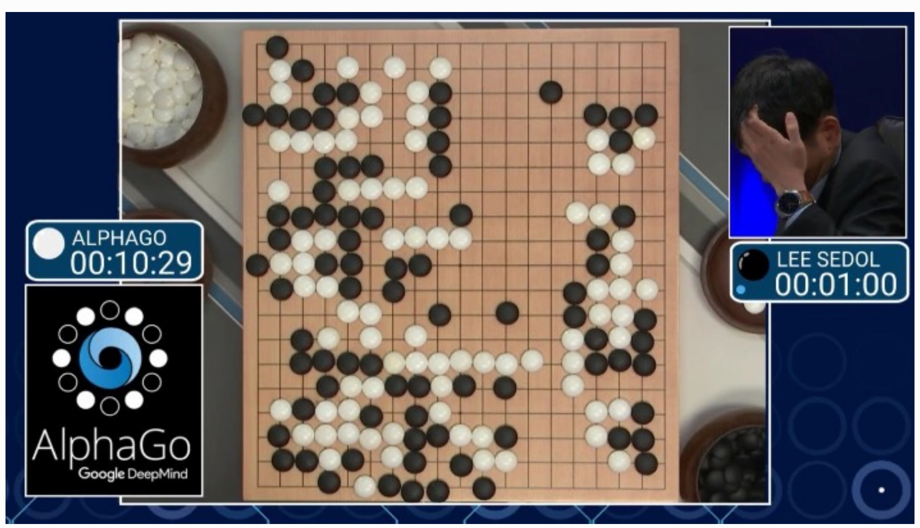


Backgammon (Tesauro, 1994)
Elevator control (Crites and Barto, 1996)
Helicopter control (Ng et al., 2003)
Transfer in tic-tac-toe (Rivest and Precup, 2003; Bellemare et al., unpublished)
...
Atari game-playing (Mnih et al., 2015)
Go (Silver et al., 2016)

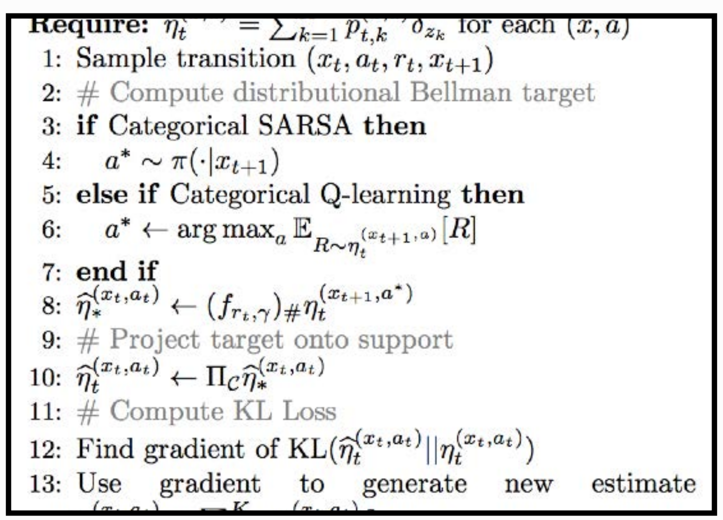
Deep RL







Challenge domains

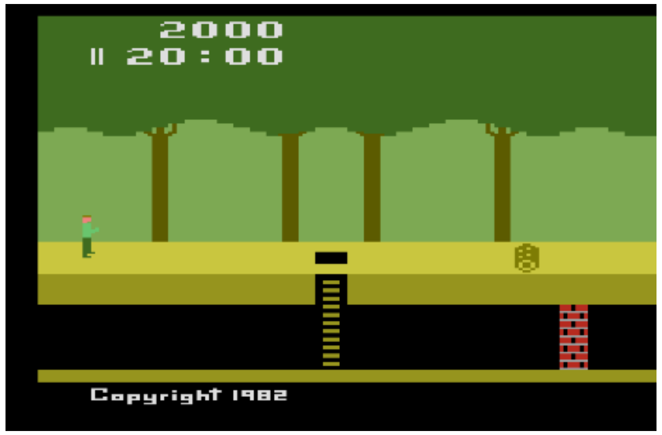


Require: $\eta_t^{(x,a)} = \sum_{k=1} p_{t,k}^{(x,a)} \delta_{z_k}$ for each (x, a)
1: Sample transition (x_t, a_t, r_t, x_{t+1})
2: # Compute distributional Bellman target
3: **if** Categorical SARSA **then**
4: $\quad a^* \sim \pi(\cdot|x_{t+1})$
5: **else if** Categorical Q-learning **then**
6: $\quad a^* \leftarrow \arg\max_a \mathbb{E}_{R\sim\eta_t^{(x_{t+1},a)}}[R]$
7: **end if**
8: $\widehat{\eta}_*^{(x_t,a_t)} \leftarrow (f_{r_t,\gamma})_\# \eta_t^{(x_{t+1},a^*)}$
9: # Project target onto support
10: $\widehat{\eta}_t^{(x_t,a_t)} \leftarrow \Pi_{\mathcal{C}} \widehat{\eta}_*^{(x_t,a_t)}$
11: # Compute KL Loss
12: Find gradient of $\mathrm{KL}(\widehat{\eta}_t^{(x_t,a_t)} || \eta_t^{(x_t,a_t)})$
13: Use gradient to generate new estimate

Algorithms

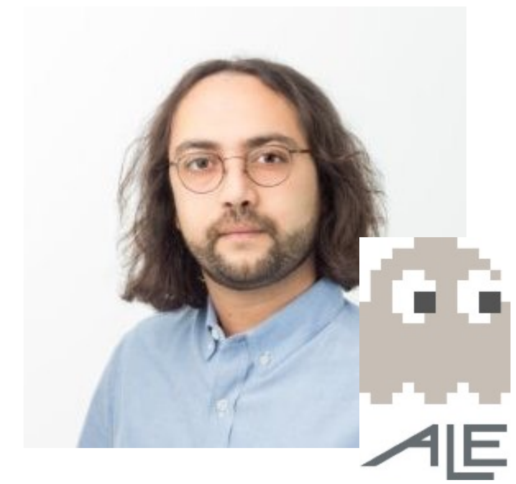
Some challenges in Deep RL

- Training data is not i.i.d.
- Hard to get confidence intervals
- Potential divergence issues (e.g. van Hasselt et al., 2015)
- State space often unknown
- Model-free methods reign supreme (simulation a plus)

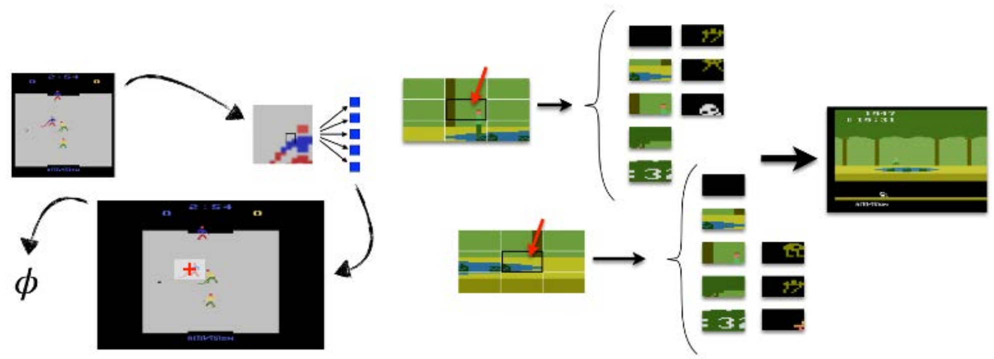


Diuk *et al.*, 2008

Barbados RL Workshop, 2008

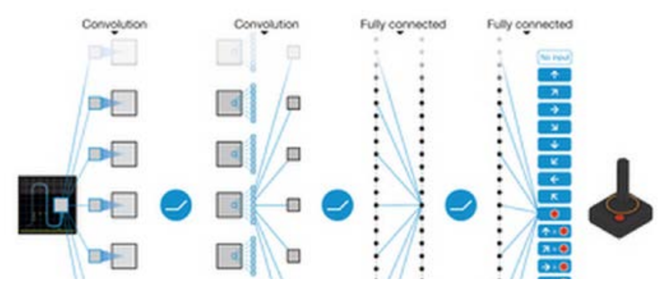


Naddaf, 2008



Bellemare *et al.*, 2011-2013

MGB, Naddaf,
Veness, Bowling 2013

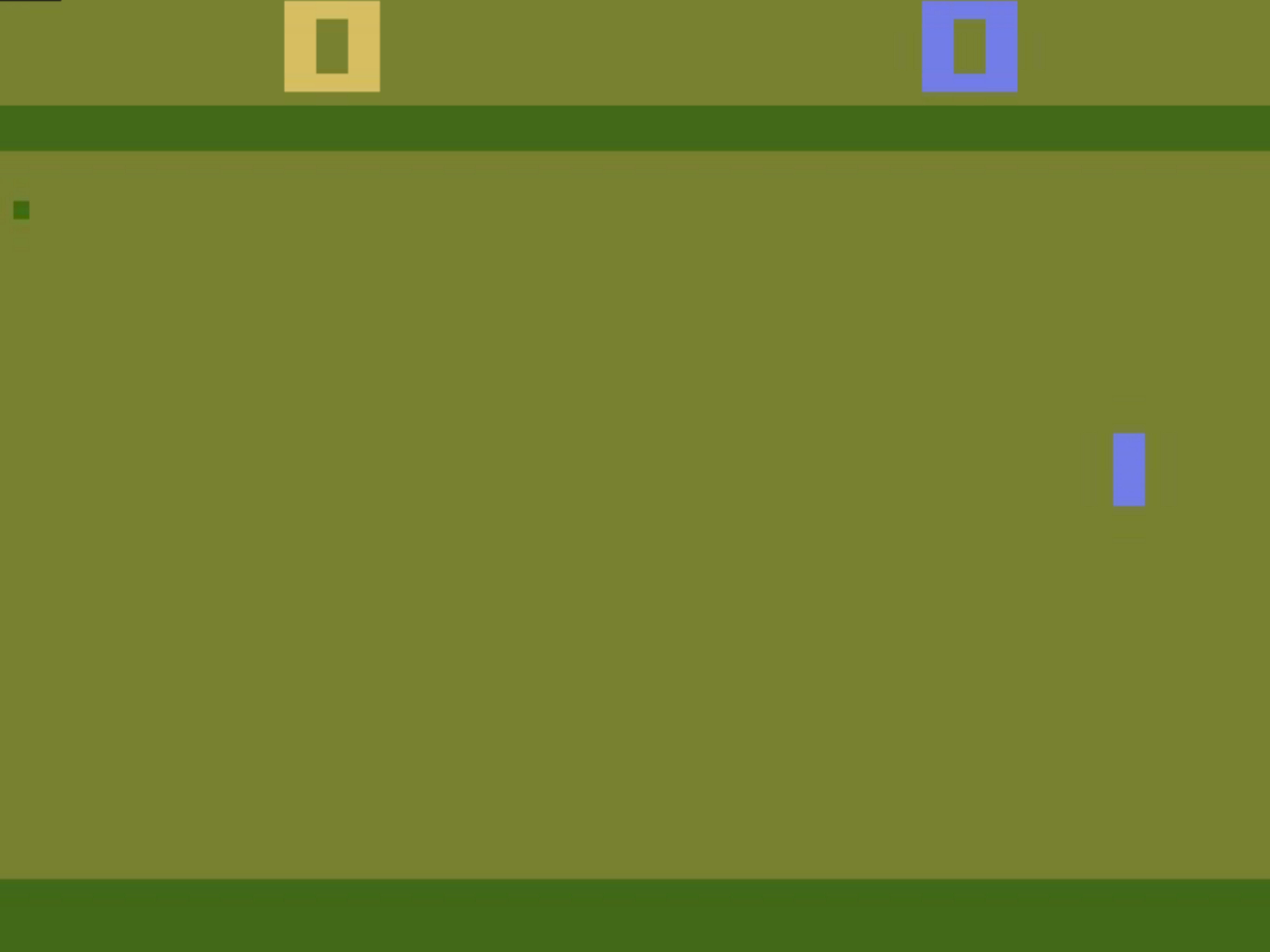


Mnih *et al.*, 2013, 2015

Hausknecht *et al.* 2013

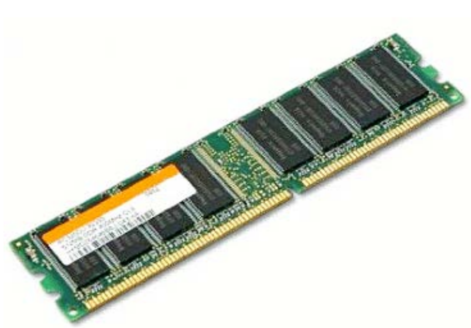
Lipovetsky *et al.* 2015

. . .

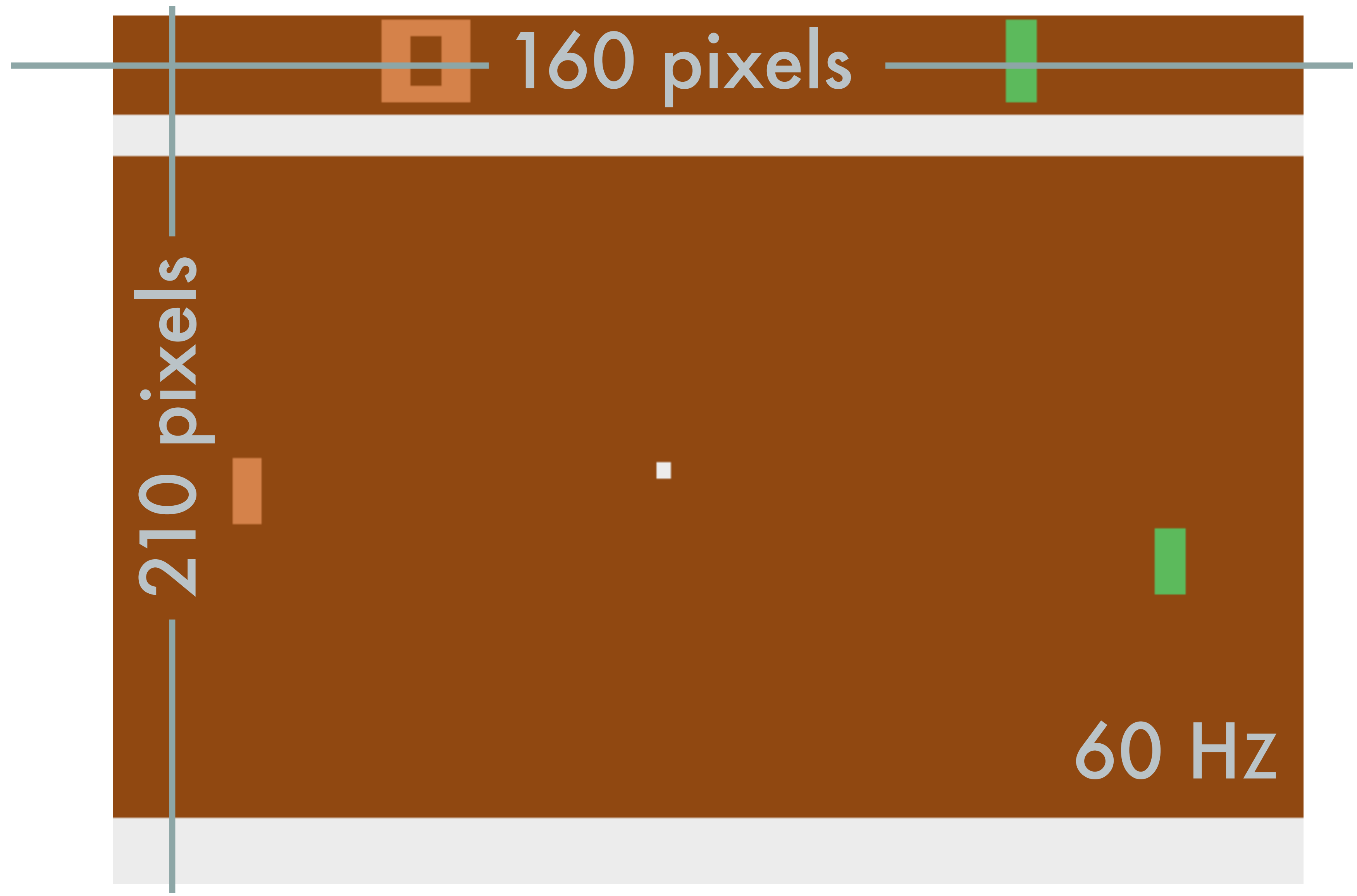

0
0

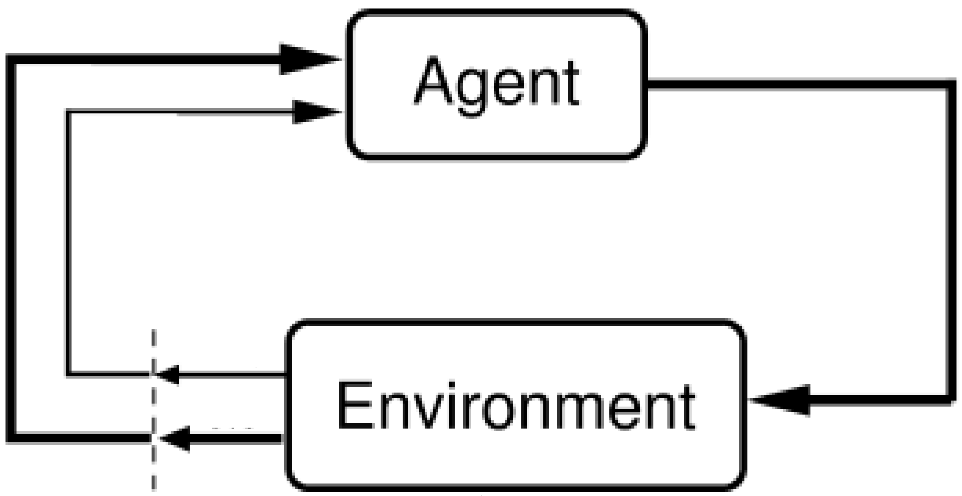
1977

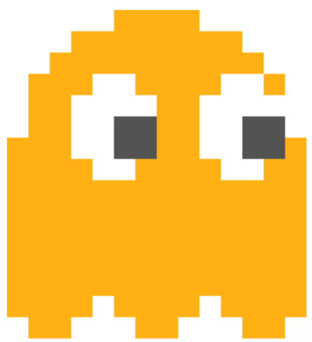
18



128 BYTES



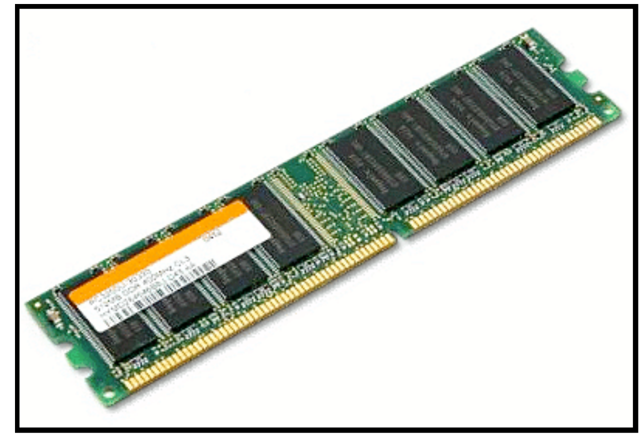


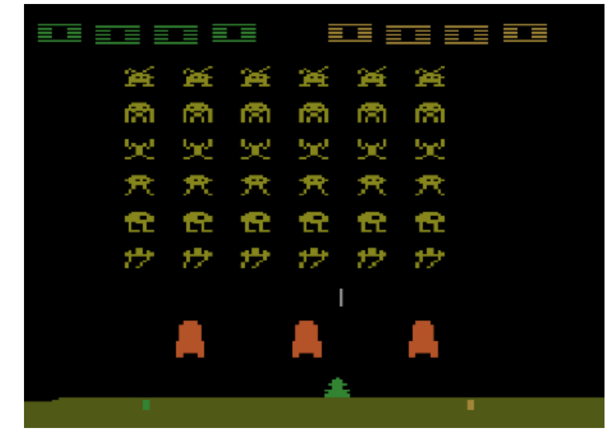


ALE

Stella
A multi-platform Atari 2600 VCS emulator

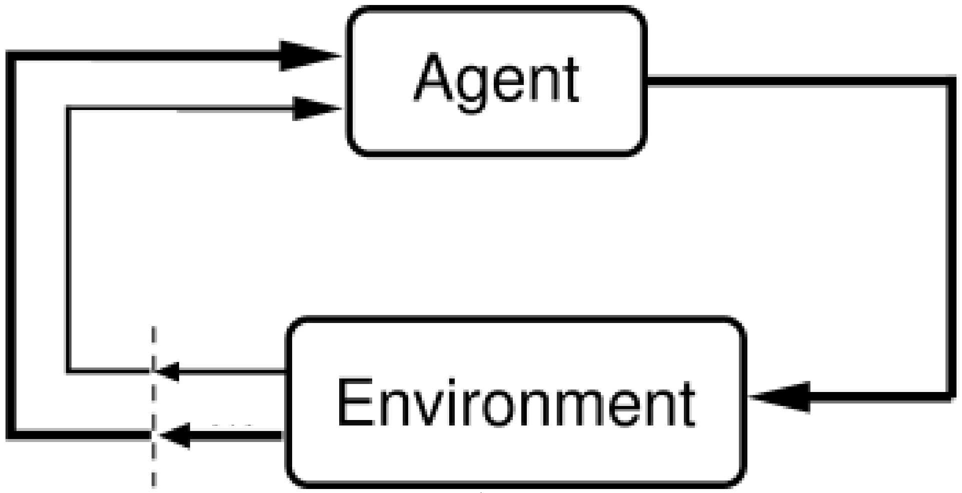
Observation

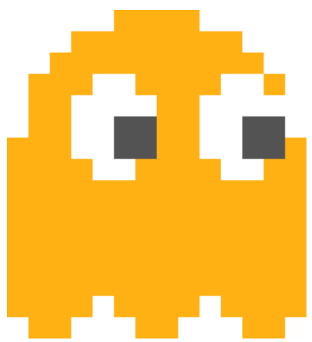




Action

The Arcade Learning Environment (Bellemare et al., 2013)

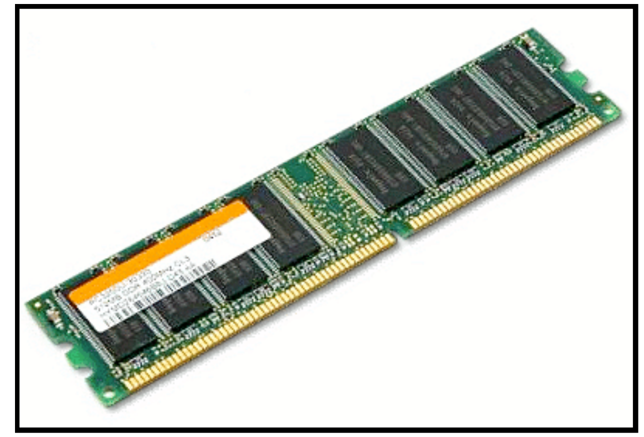


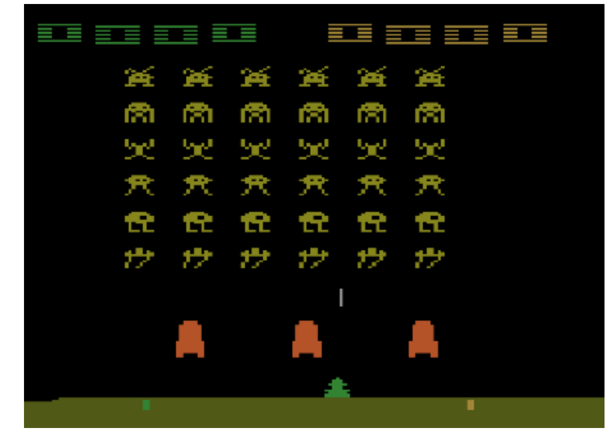


ALE

Stella
A multi-platform Atari 2600 VCS emulator

Observation

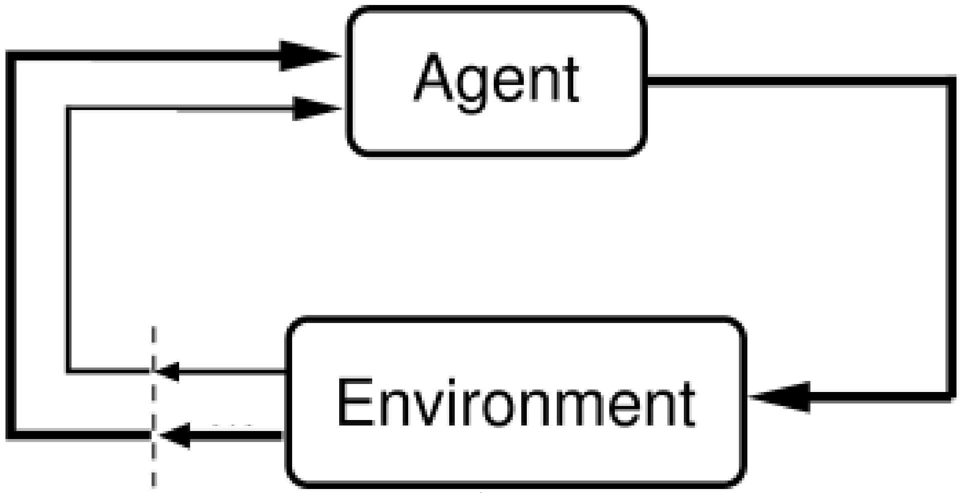


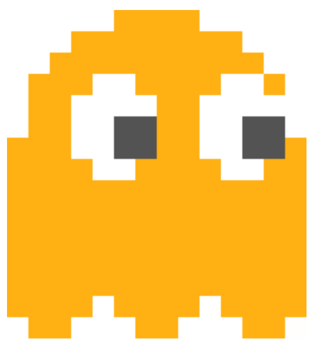


Action

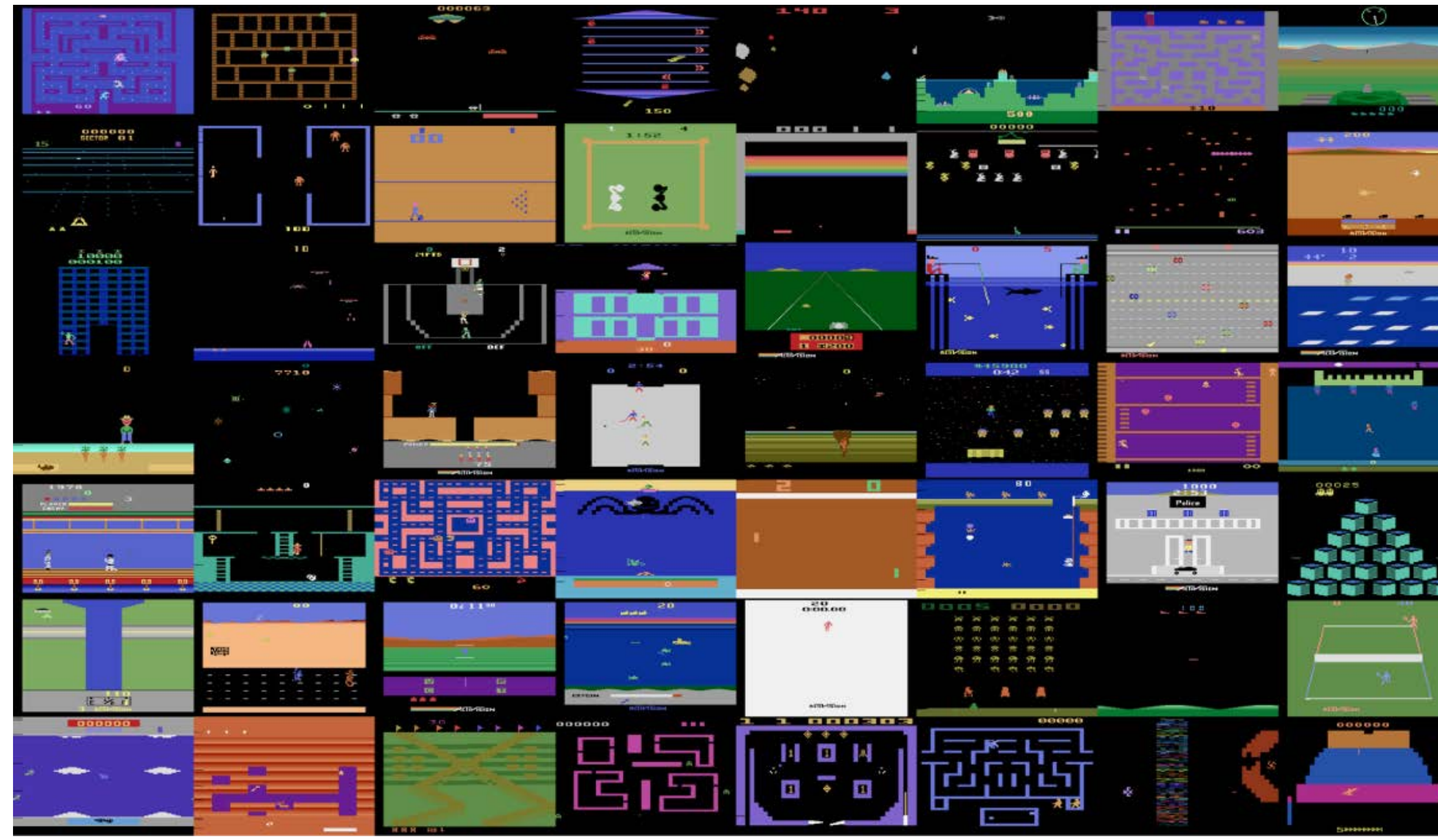
Reward

The Arcade Learning Environment (Bellemare et al., 2013)



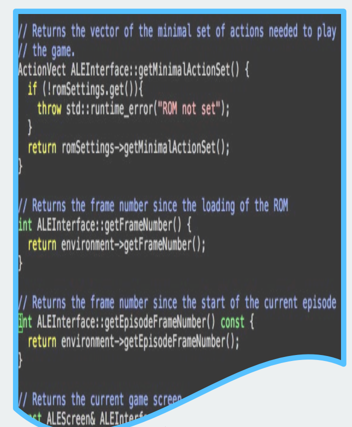


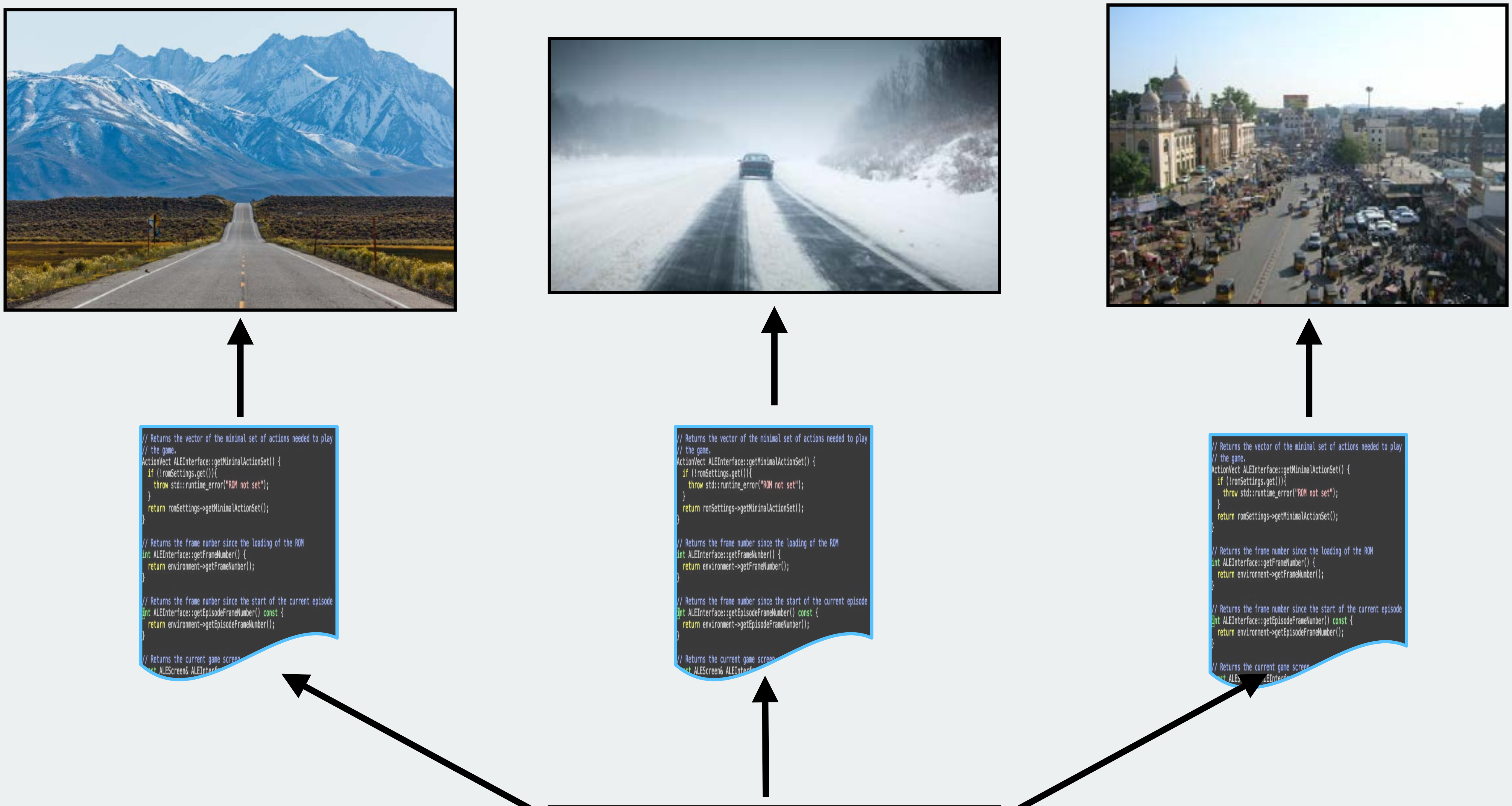
ALE



Stella
A multi-platform Atari 2600 VCS emulator

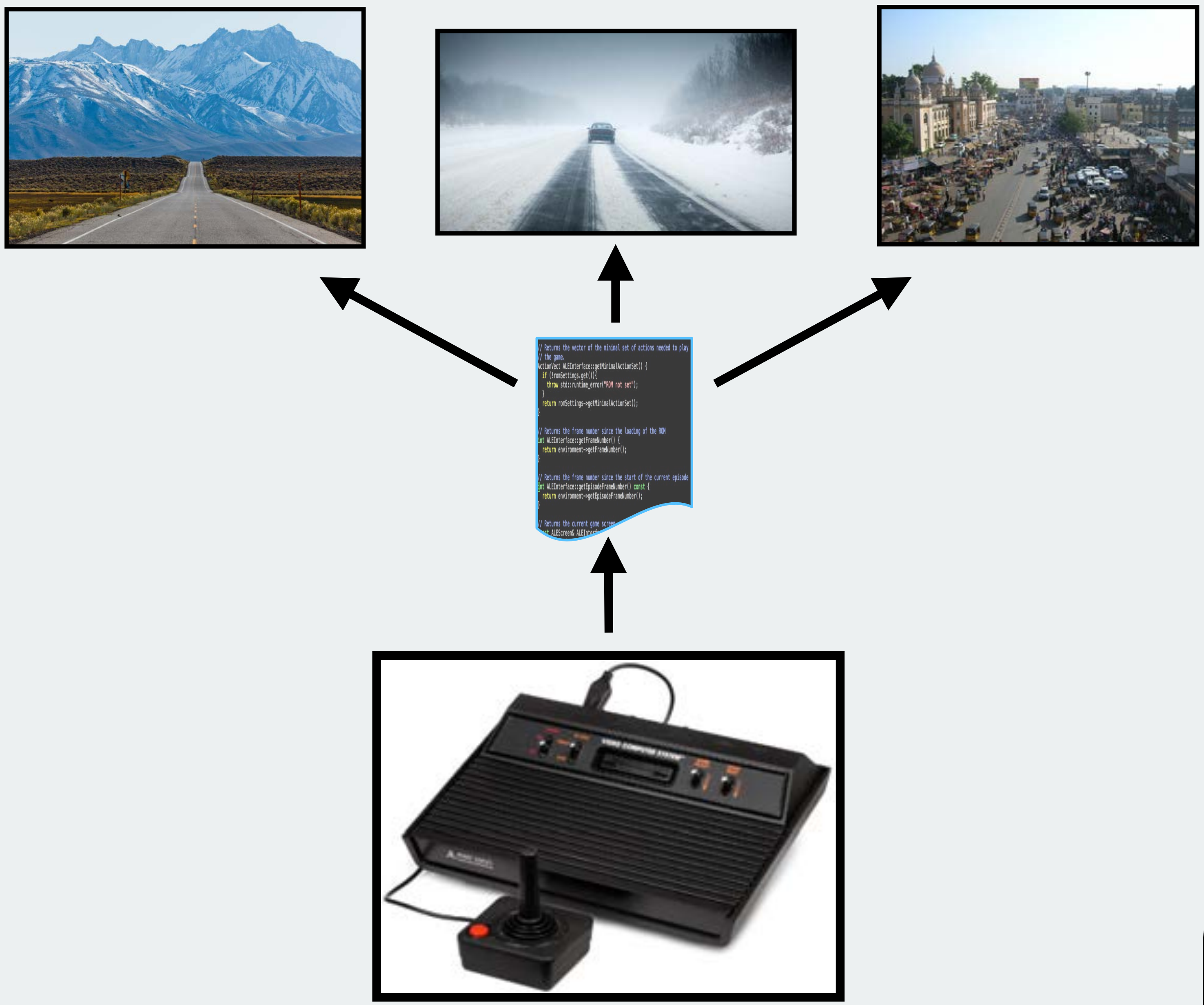
General competency





Narrow competency

CC Wikipedia

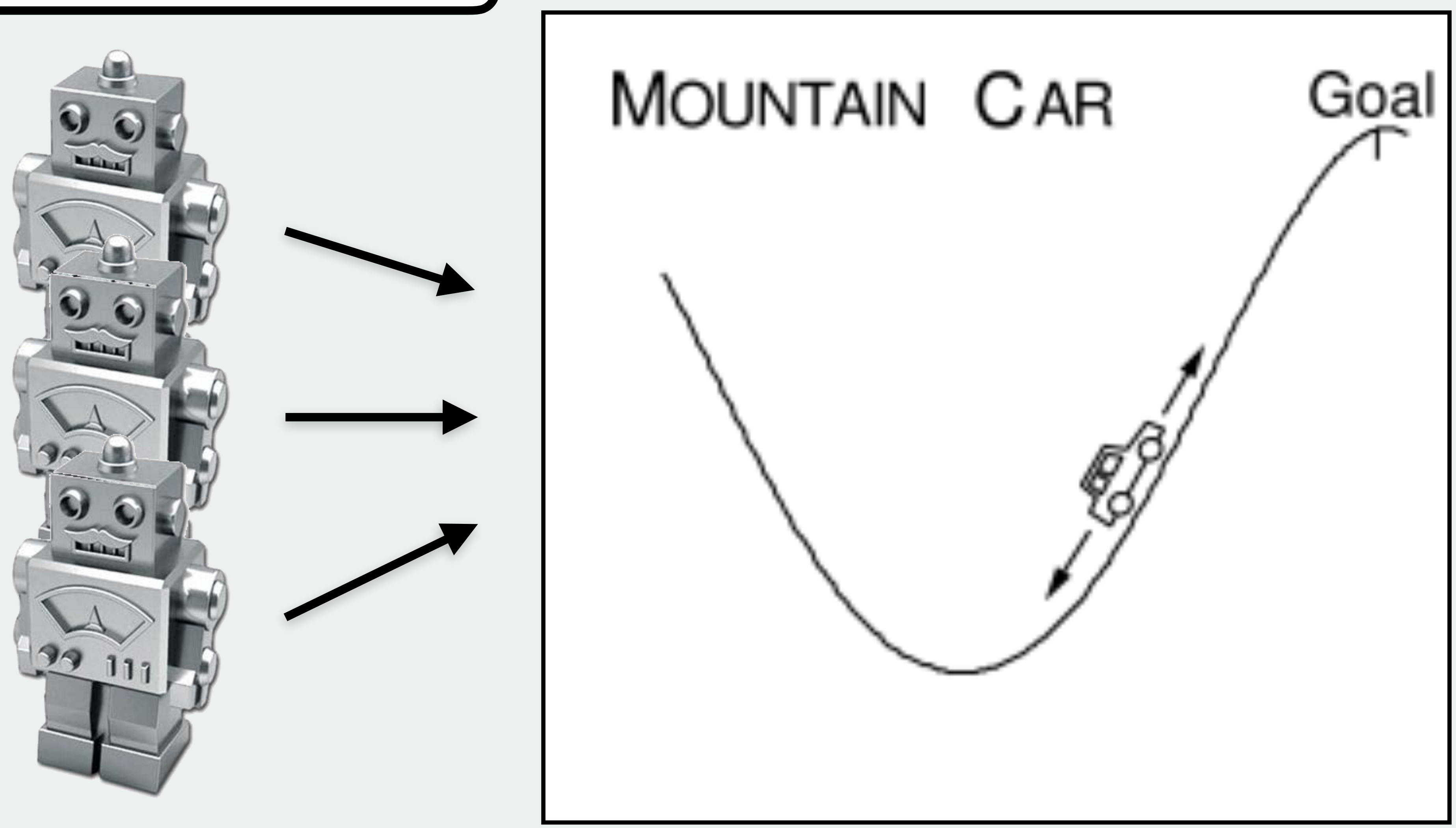


Diverse

Interesting

Independent

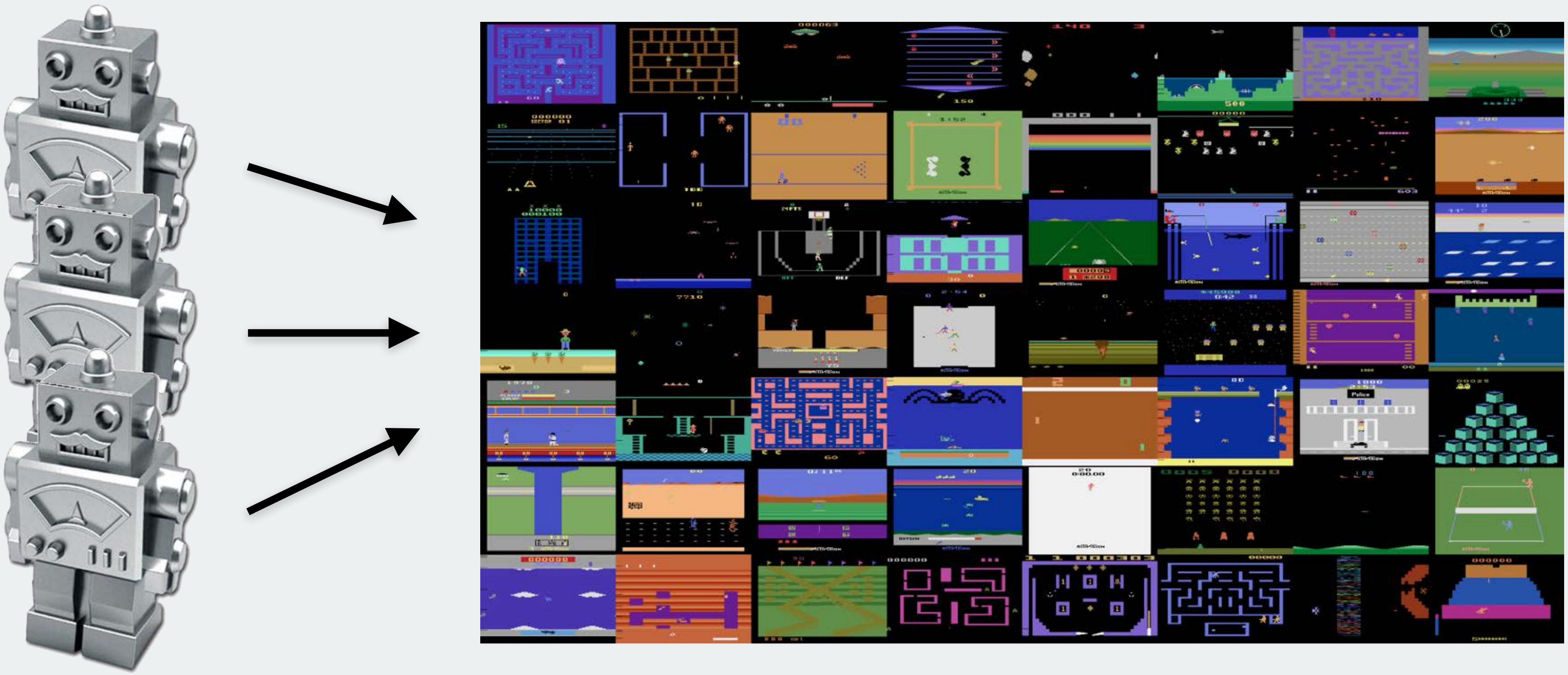
Diverse



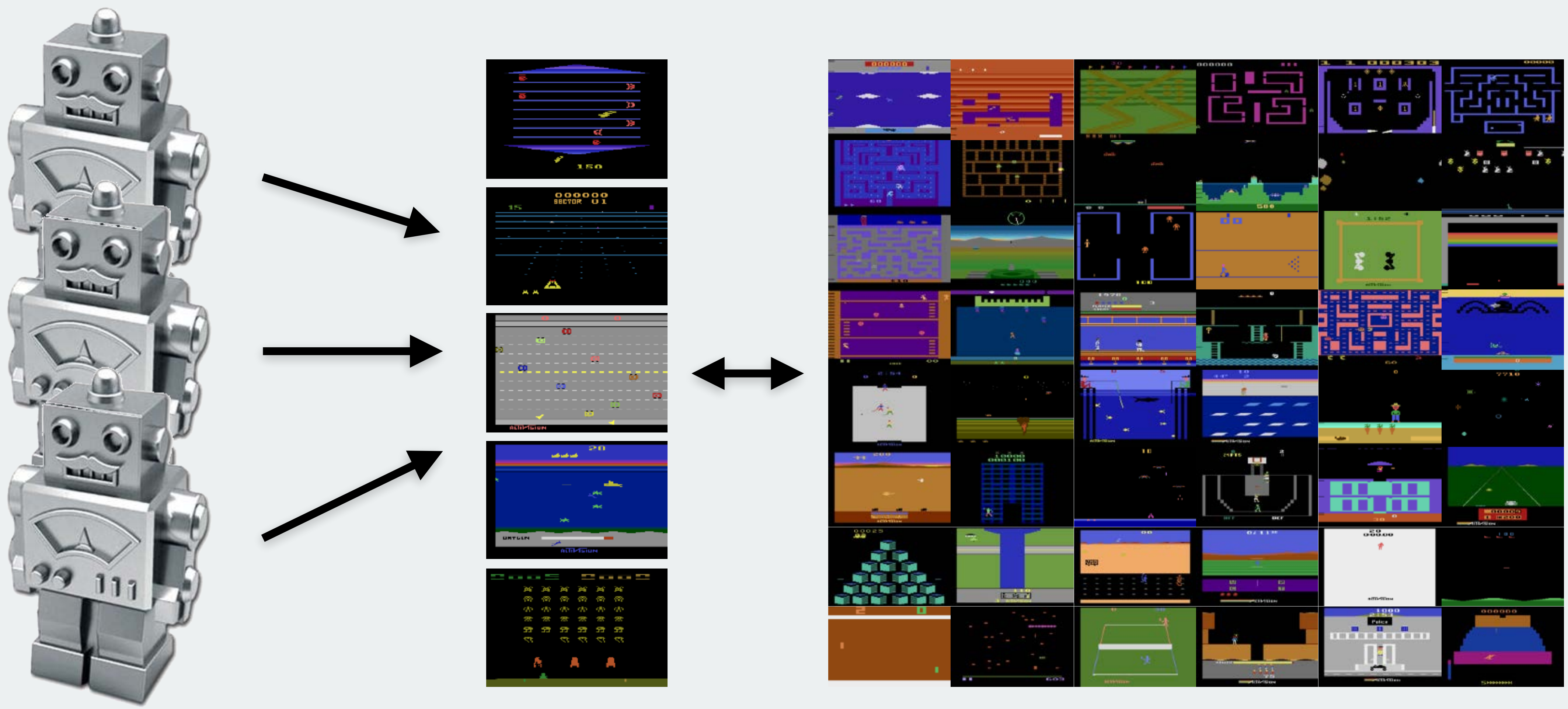
"We argue that reinforcement learning is particularly vulnerable to environment overfitting and propose as a remedy generalized methodologies [...]"

Whiteson et al., 2011

Diverse



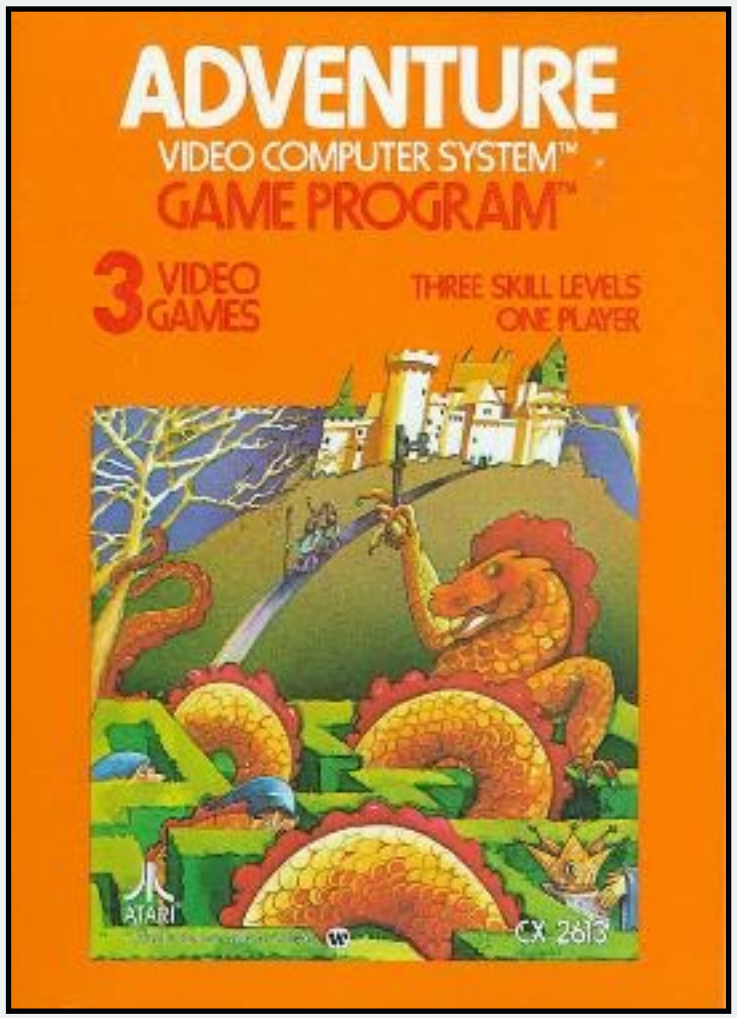
Diverse



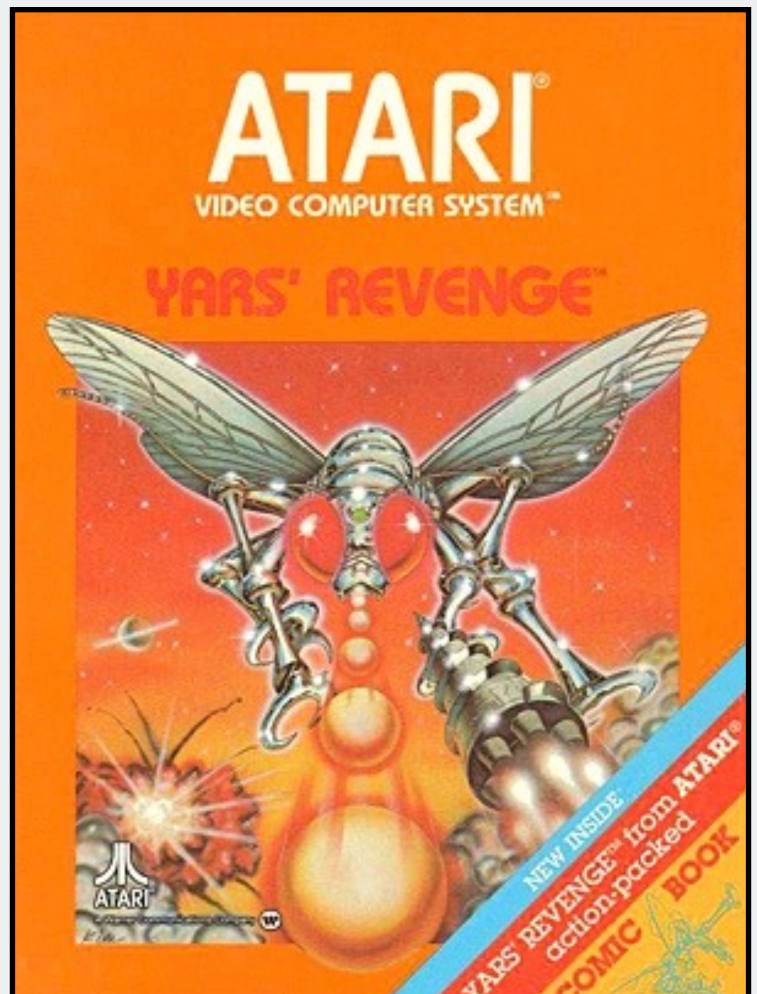
Interesting
(to people)

Independent

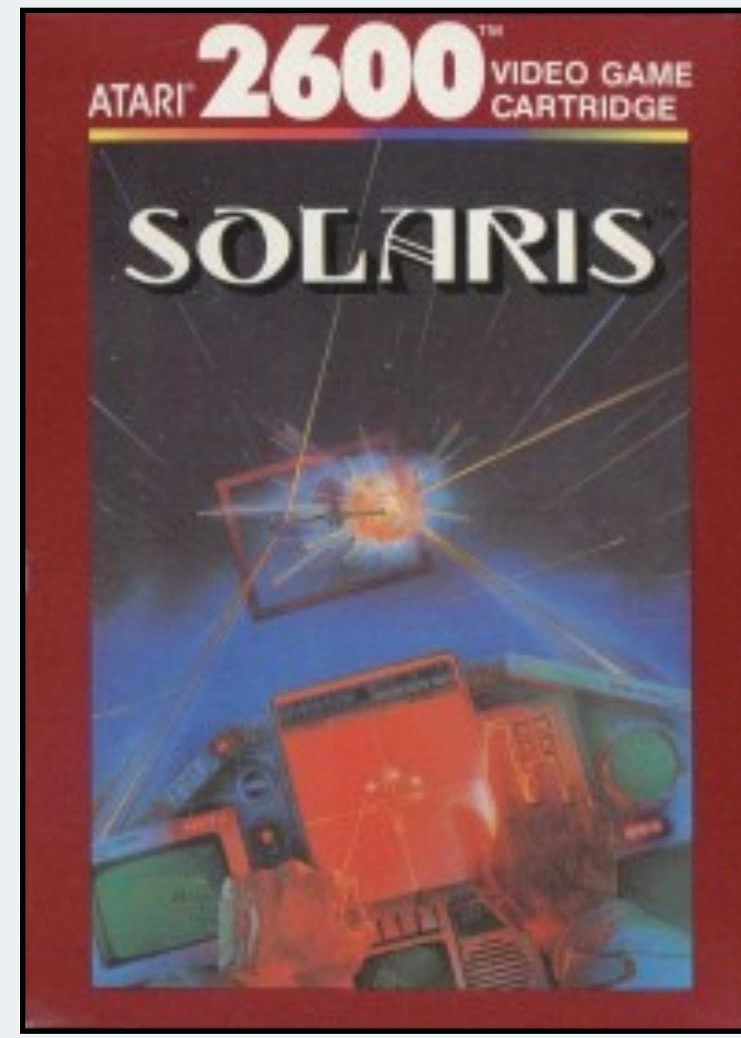
(by and for people)



1979

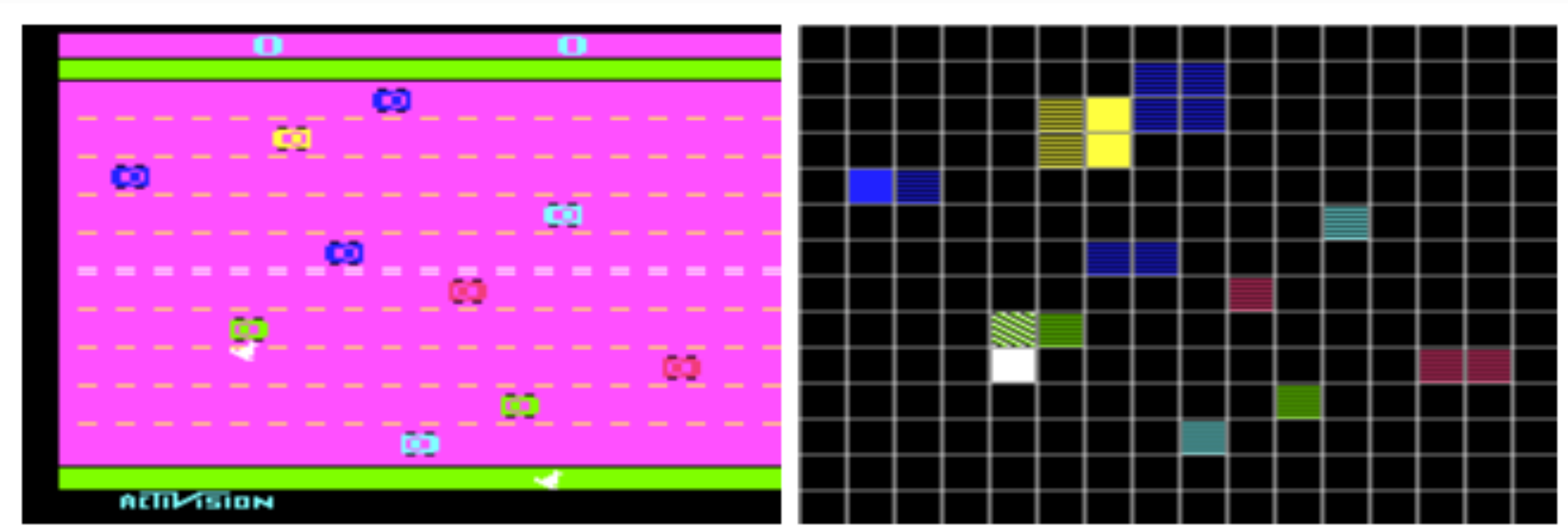


1982

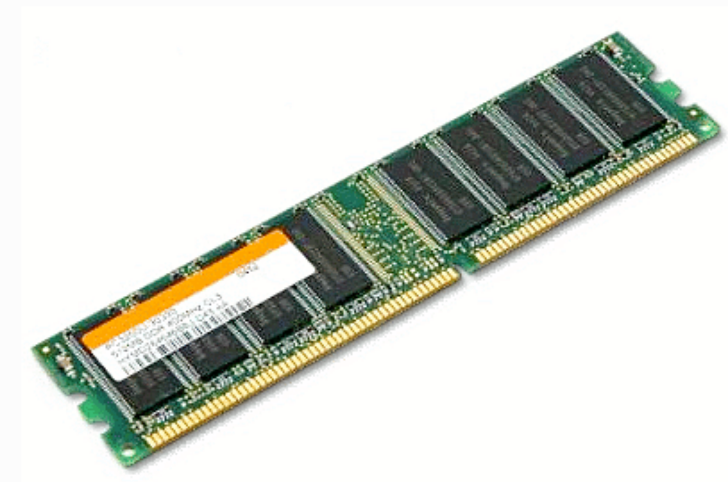


1986

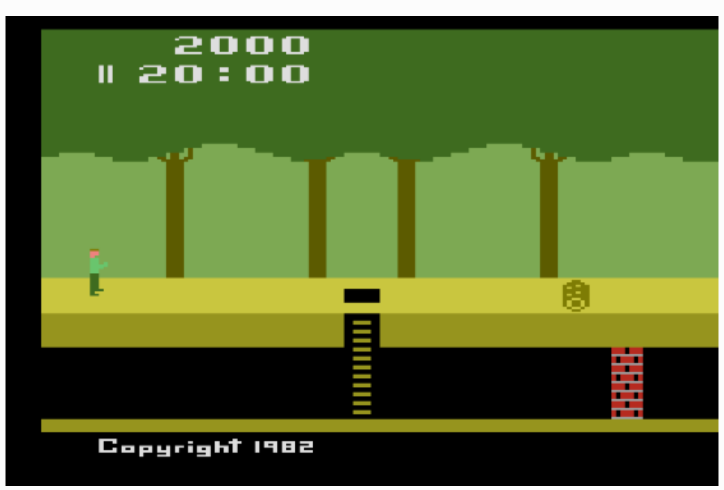
EARLY ATTEMPTS (2010-2013)



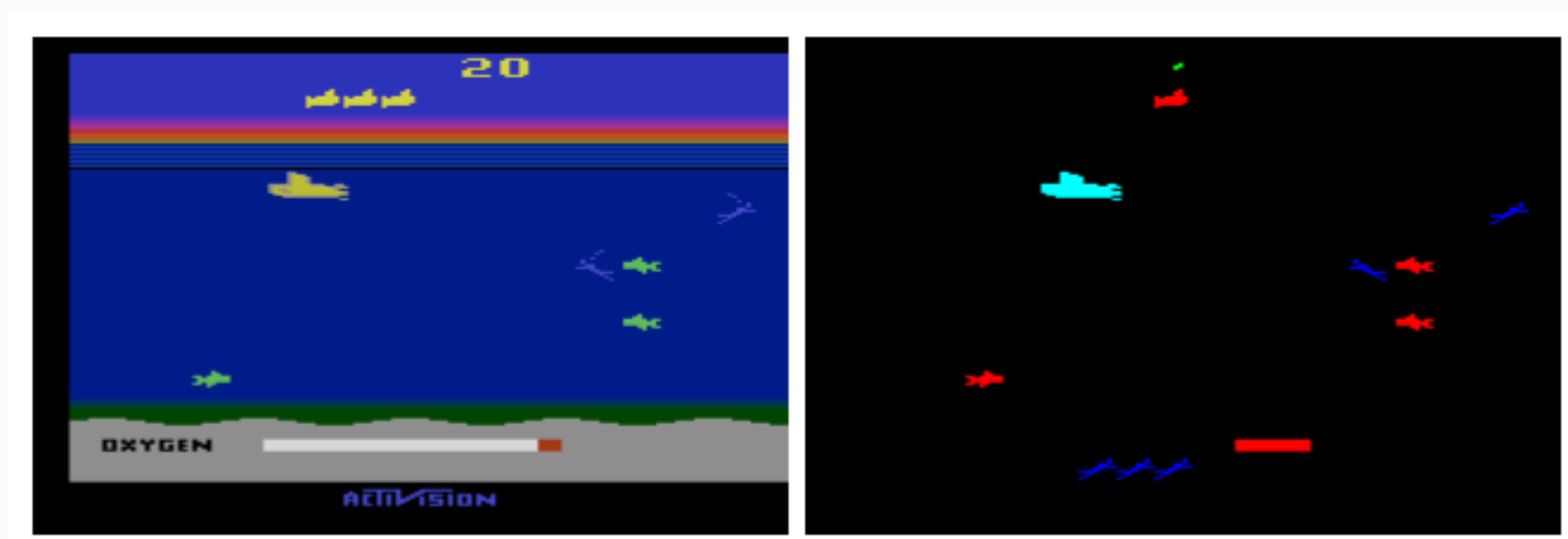
BASS/Basic Features



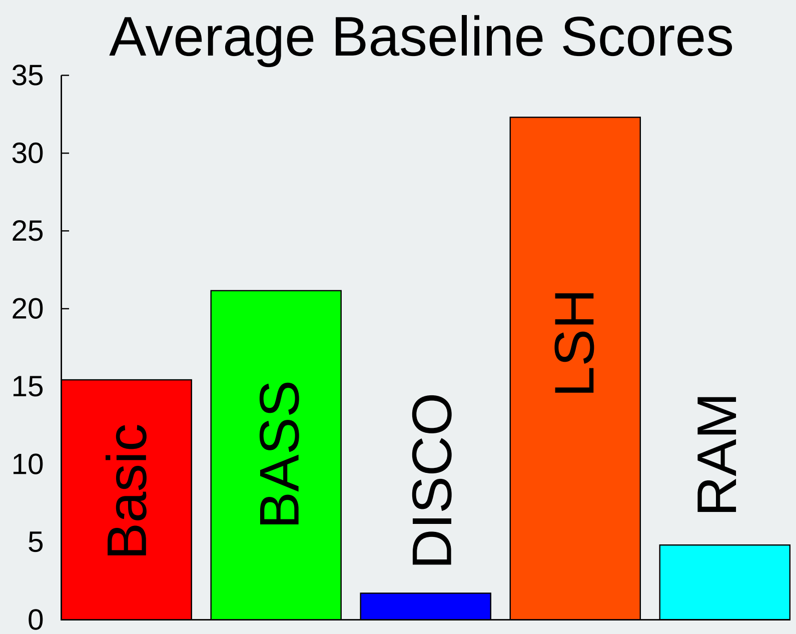
RAM

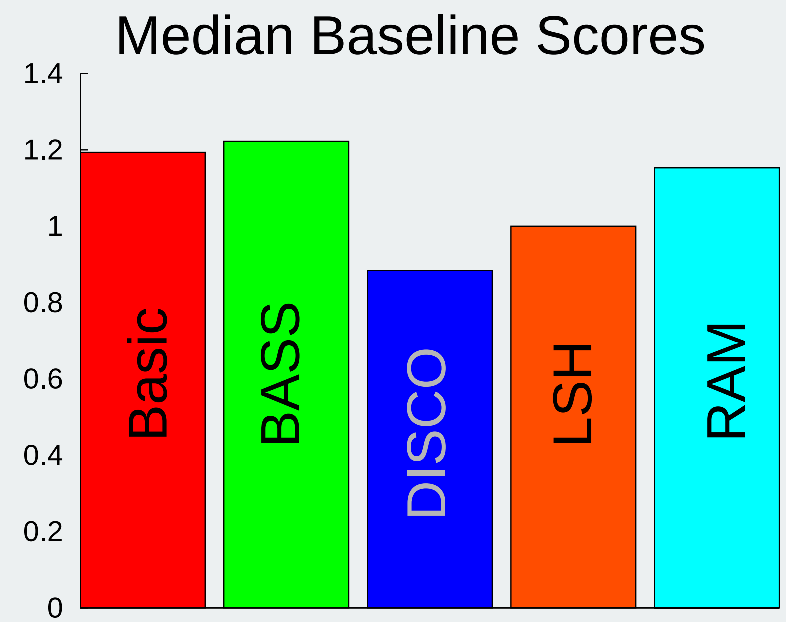


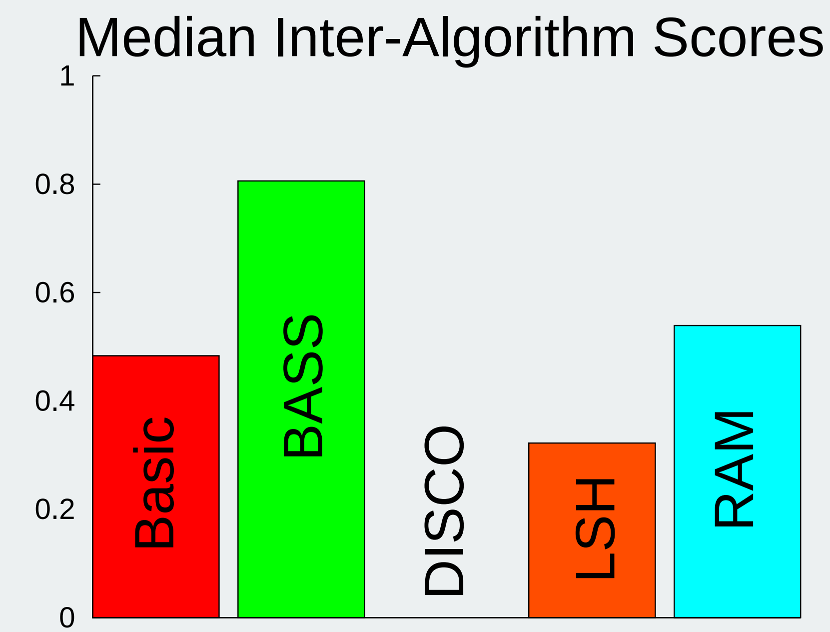
LSH on pixels



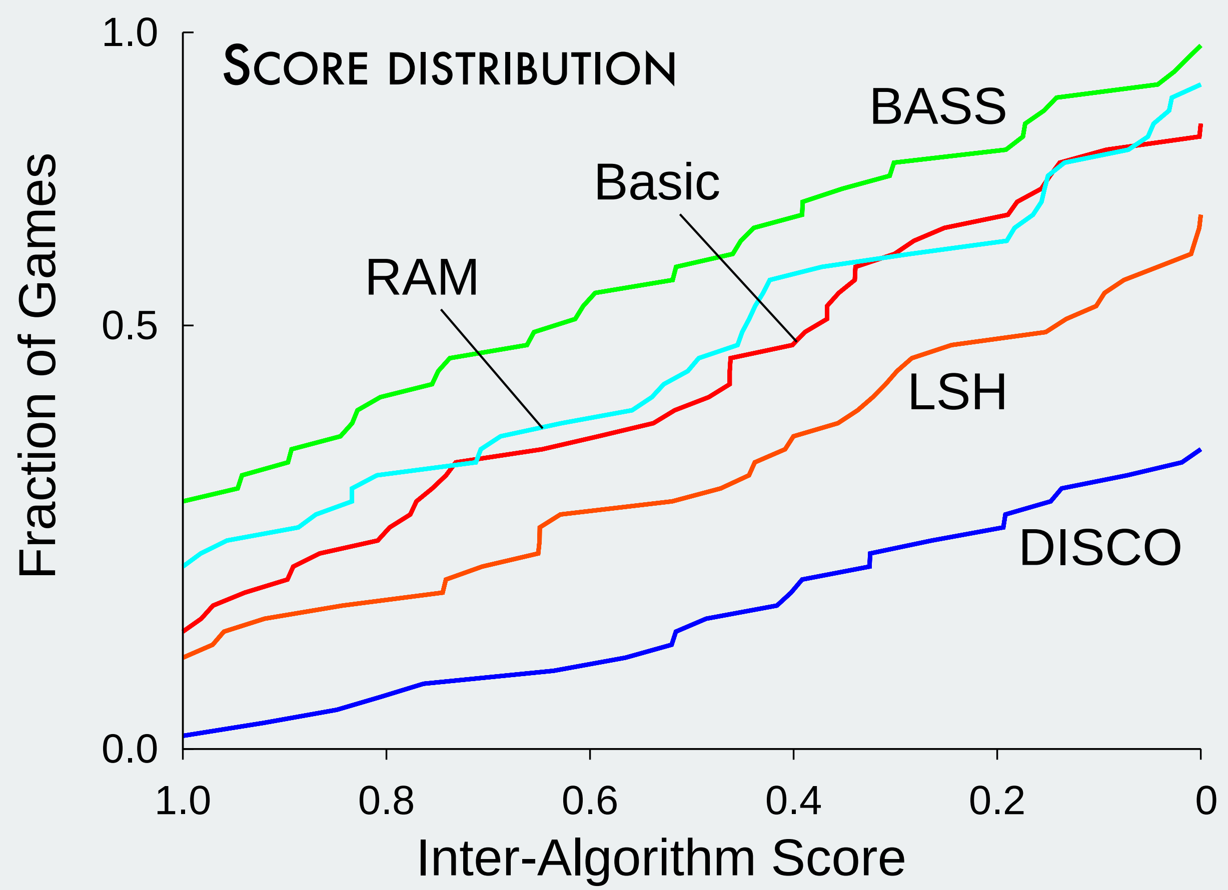
DISCO: Object Detection


Average Baseline Scores
35
30
25
20
15
10
5
0
Basic
BASS
DISCO
LSH
RAM


Median Baseline Scores
1.4
1.2
1
0.8
0.6
0.4
0.2
0
Basic
BASS
DISCO
LSH
RAM


Median Inter-Algorithm Scores
1
0.8
0.6
0.4
0.2
0
Basic
BASS
DISCO
LSH
RAM


OXYGEN
ACTIVISION


SCORE DISTRIBUTION
1.0
0.5
0.0
Fraction of Games
BASS
Basic
RAM
LSH
DISCO
1.0
0.8
0.6
0.4
0.2
0
Inter-Algorithm Score

DEEP Q-NETWORKS (DQN)

Mnih et al., 2015

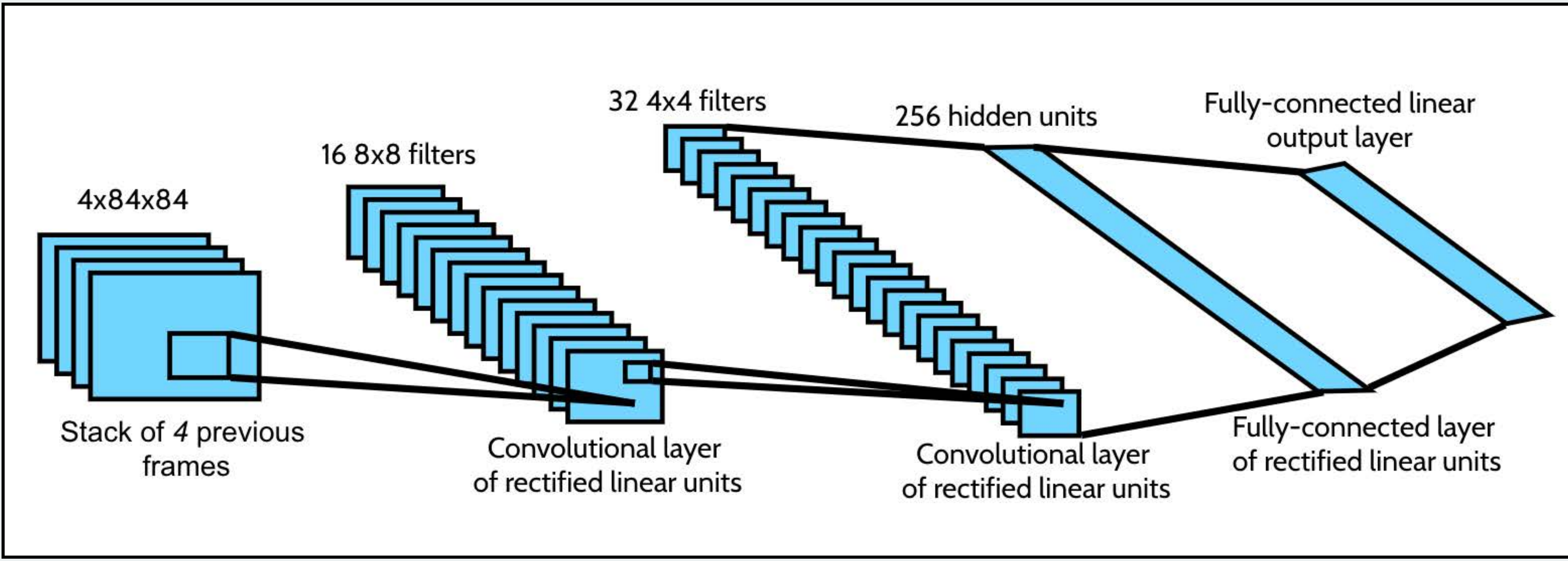
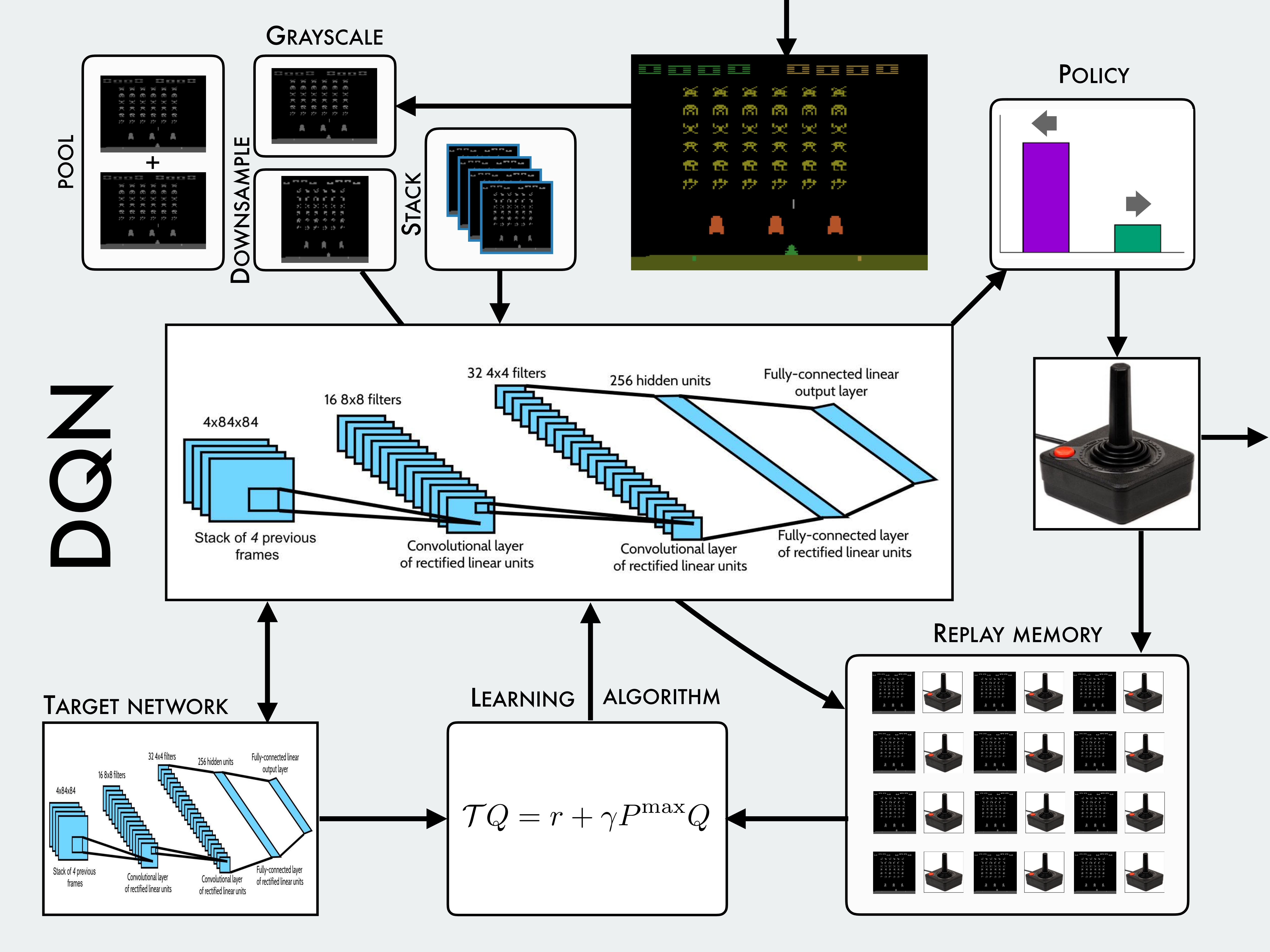
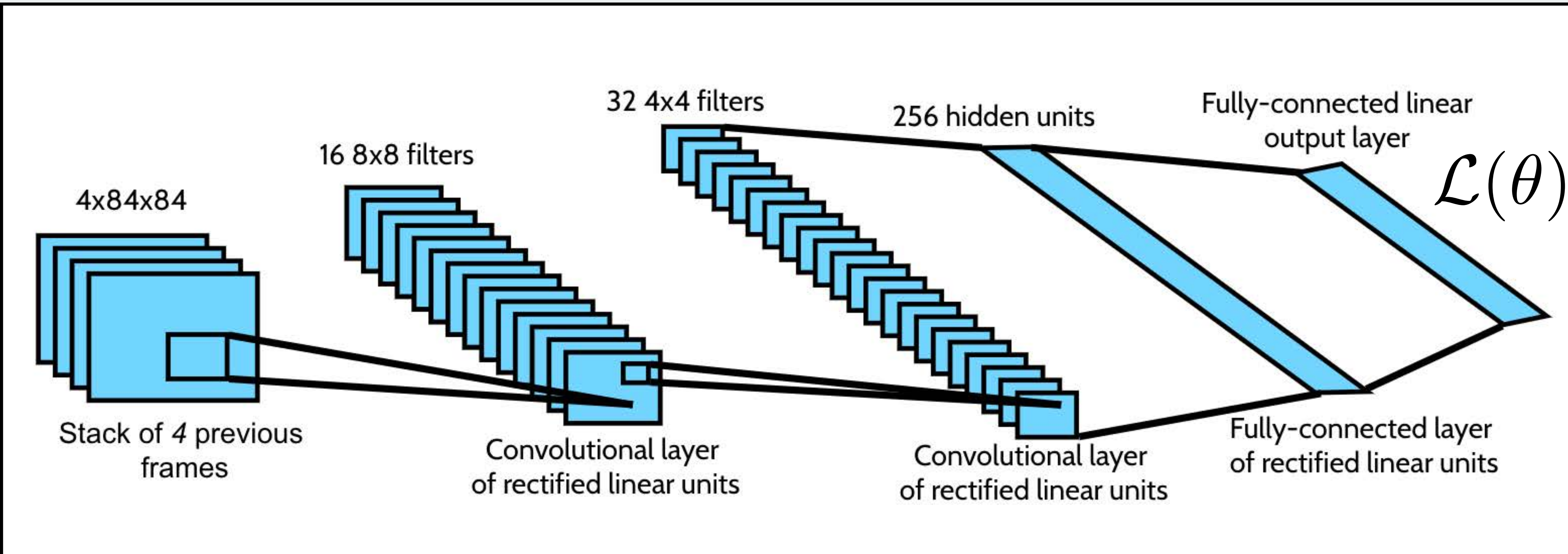


Image credit: Volodymyr Mnih



GRAYSCALE
POOL
DOWNSAMPLE
STACK
POLICY
DQN
4x84x84
16 8x8 filters
32 4x4 filters
256 hidden units
Fully-connected linear output layer
Stack of 4 previous frames
Convolutional layer of rectified linear units
Convolutional layer of rectified linear units
Fully-connected layer of rectified linear units
TARGET NETWORK
LEARNING ALGORITHM
REPLAY MEMORY
$\mathcal{T}Q = r + \gamma P^{\max} Q$

$$\nabla_\theta \mathcal{L}(\theta)$$

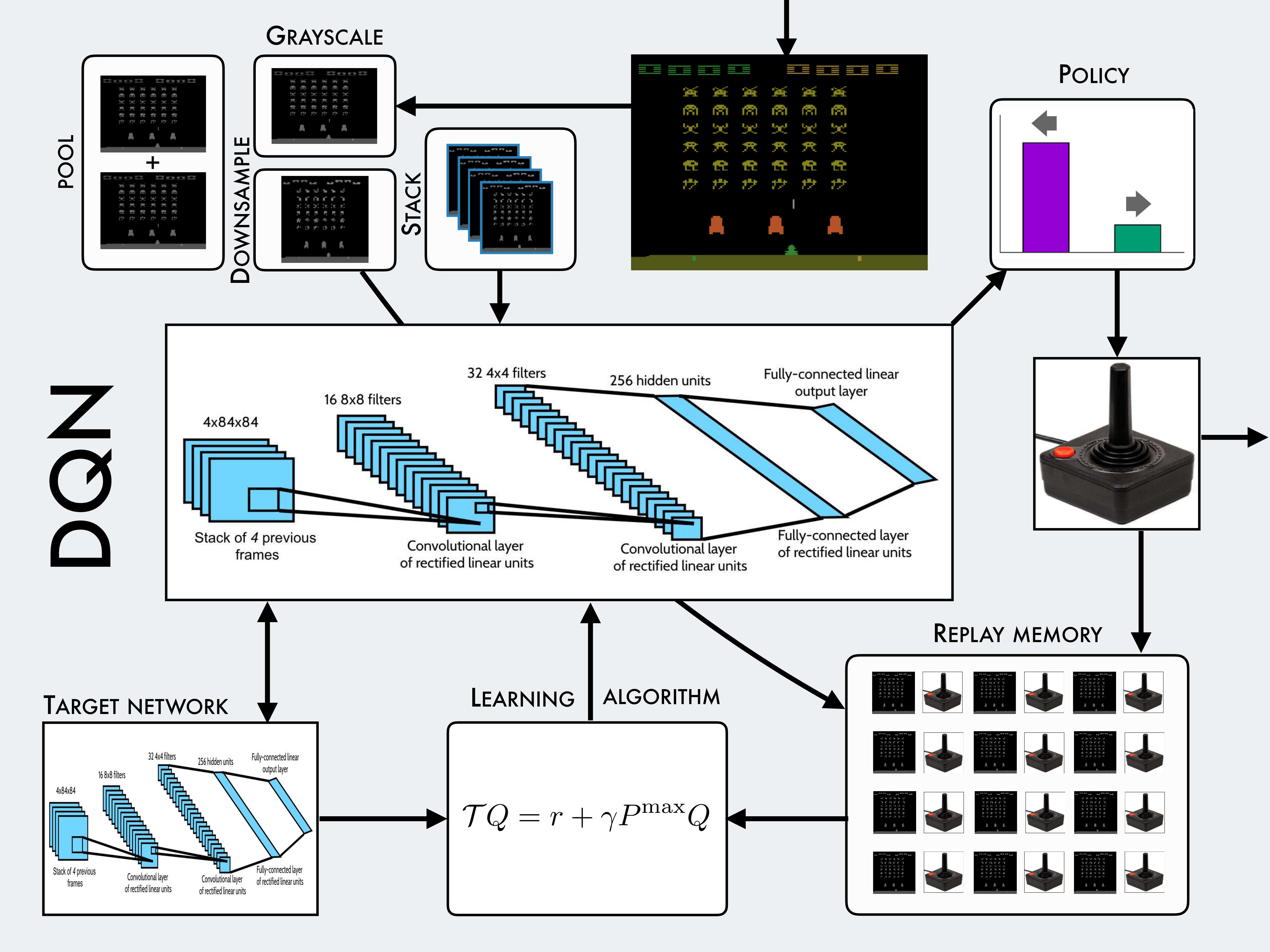


$$\mathcal{L}(\theta) = \frac{1}{2}\left(r + \gamma \max_{a' \in \mathcal{A}} Q_{\tilde{\theta}}(x', a') - Q_\theta(x, a)\right)^2$$

LEARNING ALGORITHM

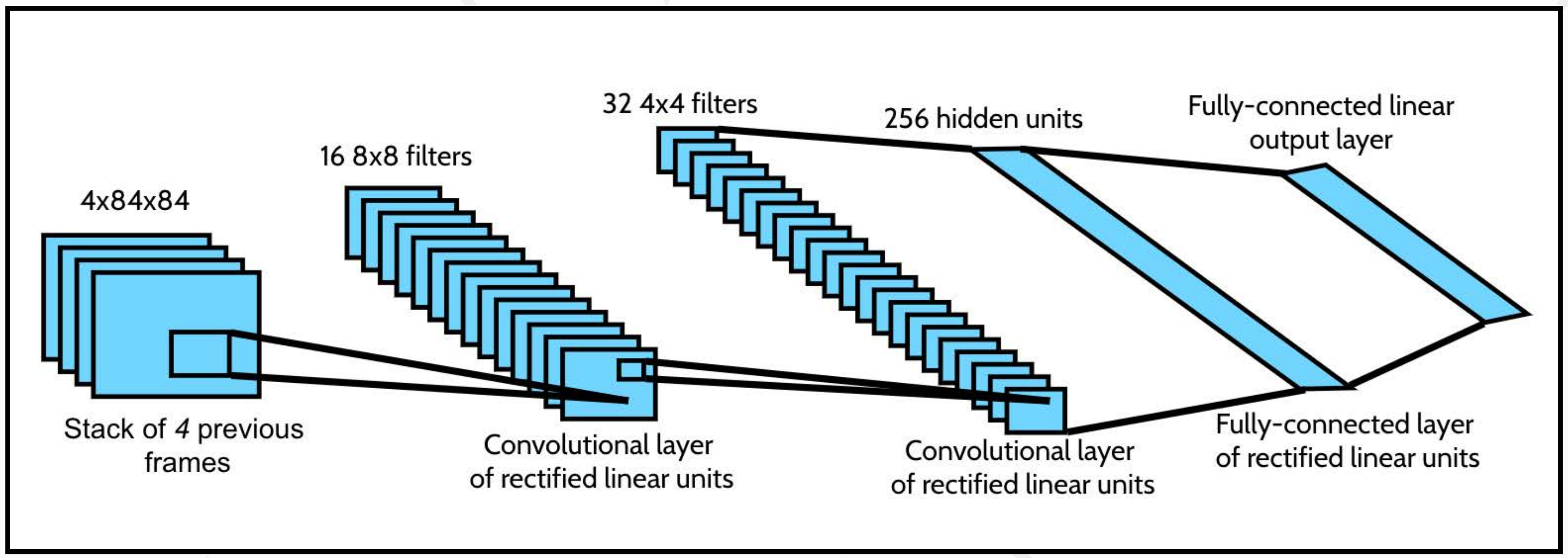
$$\mathcal{T}Q = r + \gamma P^{\max} Q$$

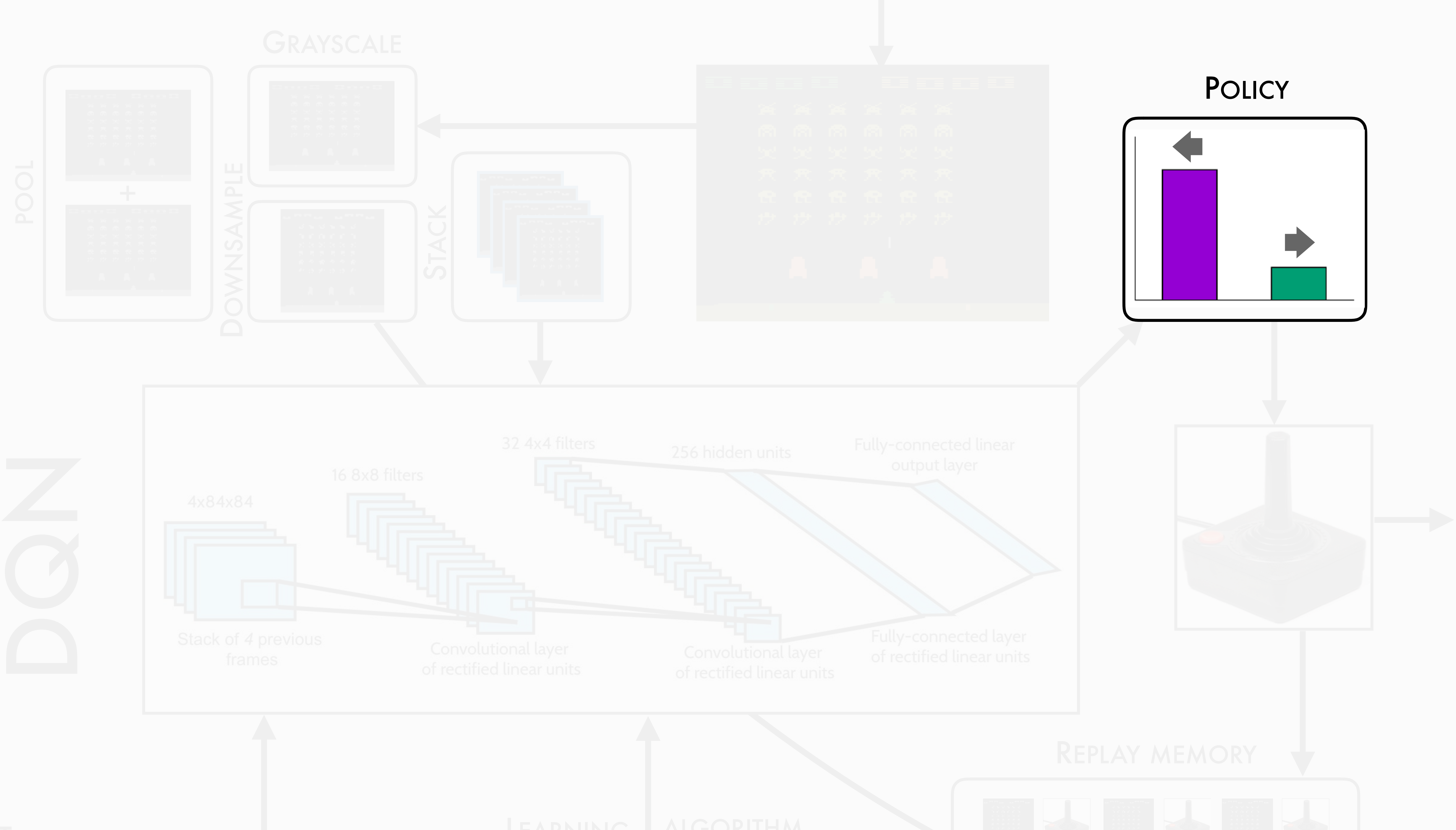
Bellemare, Ostrovski, Guez, Thomas, Munos, 2015



Grayscale
Pool
+
Downsample
Stack
Policy
DQN
4x84x84
16 8x8 filters
32 4x4 filters
256 hidden units
Fully-connected linear output layer
Stack of 4 previous frames
Convolutional layer of rectified linear units
Convolutional layer of rectified linear units
Fully-connected layer of rectified linear units
Replay memory
Target network
4x84x84
16 8x8 filters
32 4x4 filters
256 hidden units
Fully-connected linear output layer
Stack of 4 previous frames
Convolutional layer of rectified linear units
Convolutional layer of rectified linear units
Fully-connected layer of rectified linear units
Learning algorithm
$\mathcal{T}Q = r + \gamma P^{\max} Q$

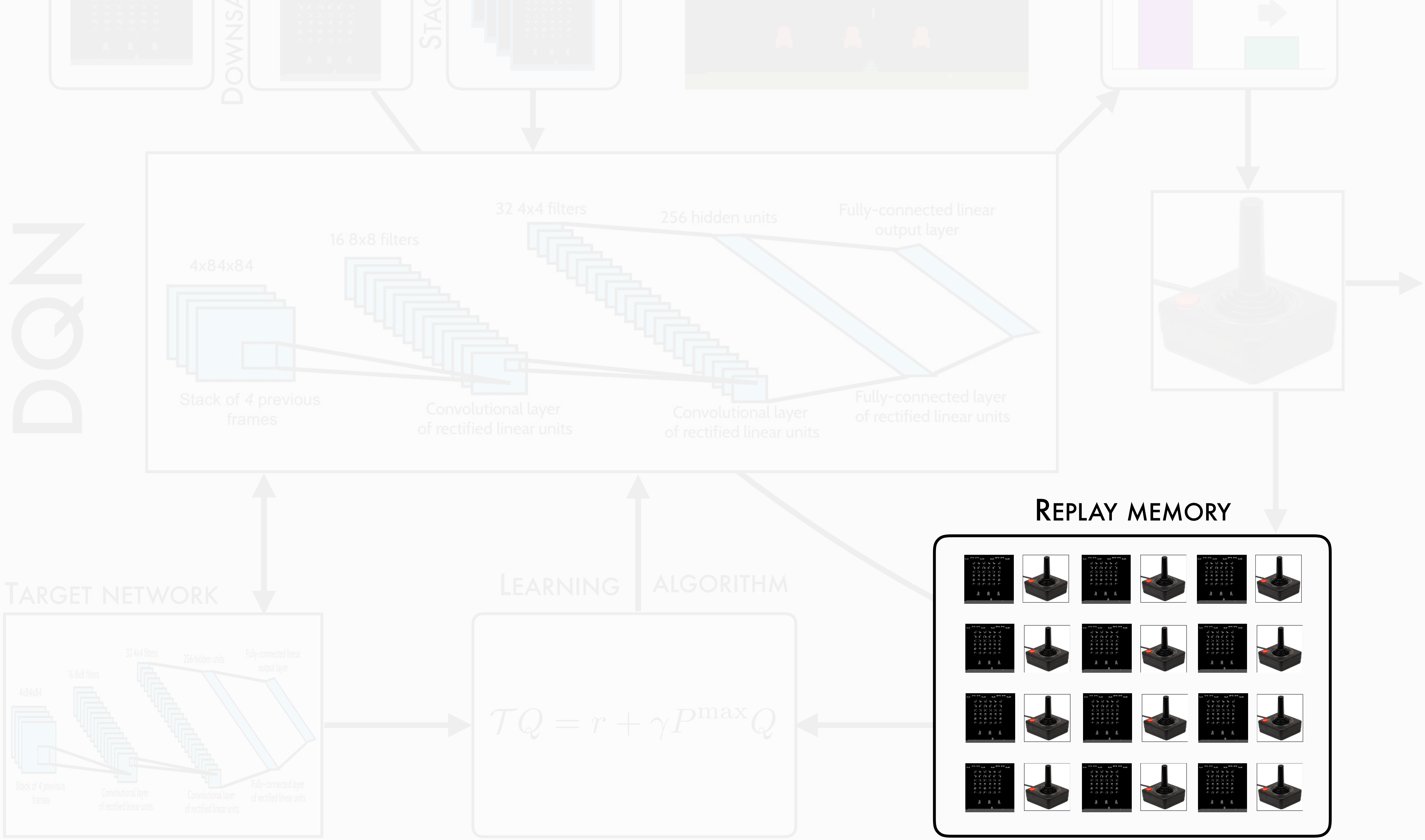
LSTM layer (Hausknecht and Stone, 2015)
Recurrent network (Mnih et al., 2016)



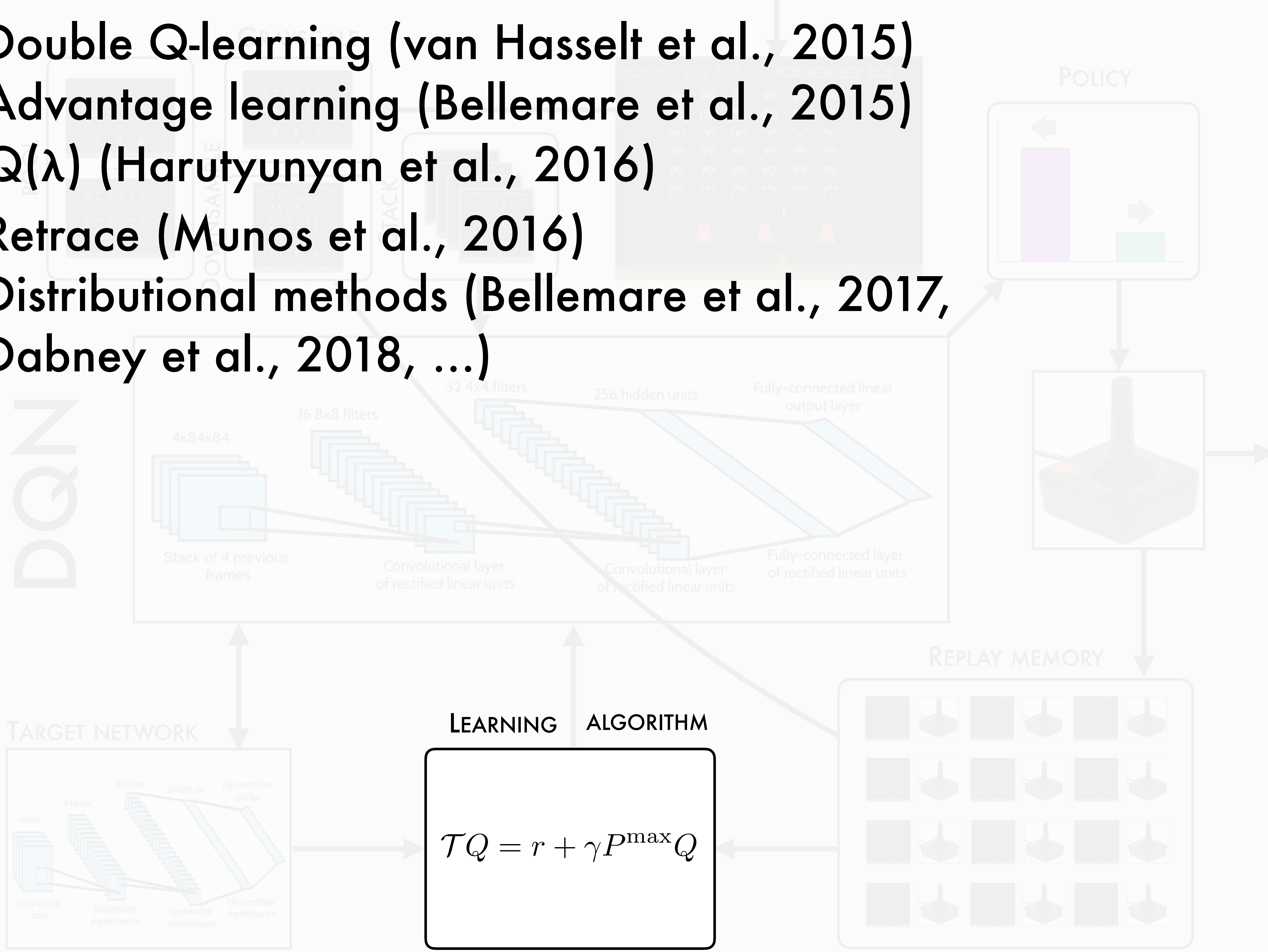


Dueling networks (Wang et al., 2016)
Asynchronous actor-critic (Mnih et al., 2016)
Bootstrap Q-functions (Osband et al., 2016)
Noisy networks (Fortunato et al.; Plappert et al., 2017)

Prioritized replay (Schaul et al., 2016)
Importance sampling (Gruslys et al., 2018)
Off-policy corrections (Gelada et al., in prep.)



Double Q-learning (van Hasselt et al., 2015)
Advantage learning (Bellemare et al., 2015)
Q(λ) (Harutyunyan et al., 2016)
Retrace (Munos et al., 2016)
Distributional methods (Bellemare et al., 2017, Dabney et al., 2018, ...)



1. Distributional reinforcement learning

2. Exploration with pseudo-counts

$\boxed{x}$

$$\mathbf{E}\, R(x):$$

⚀⚀

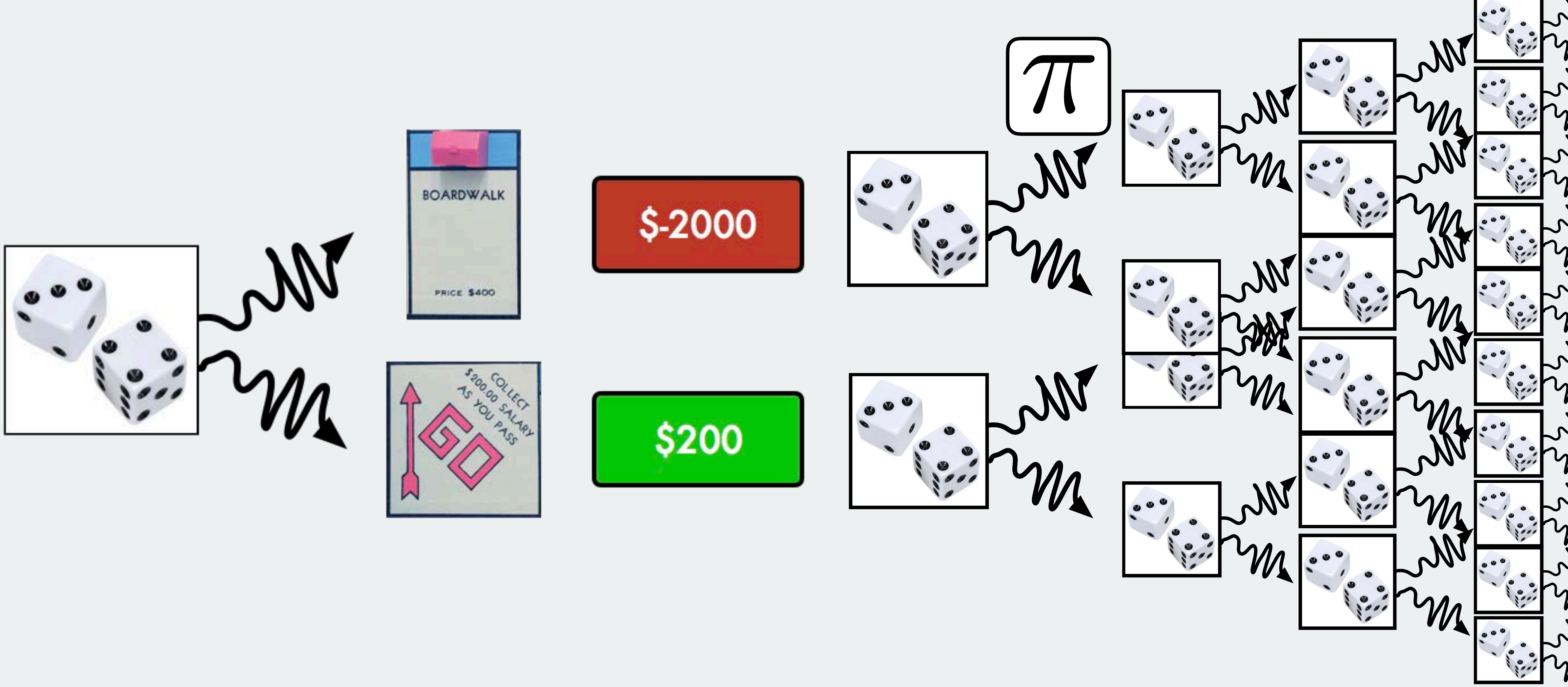
$\boxed{R(x)}$

$$\frac{1}{36}(-2000)$$

⚀⚁-⚅⚅

$$+\frac{35}{36}(200)$$

$$139$$

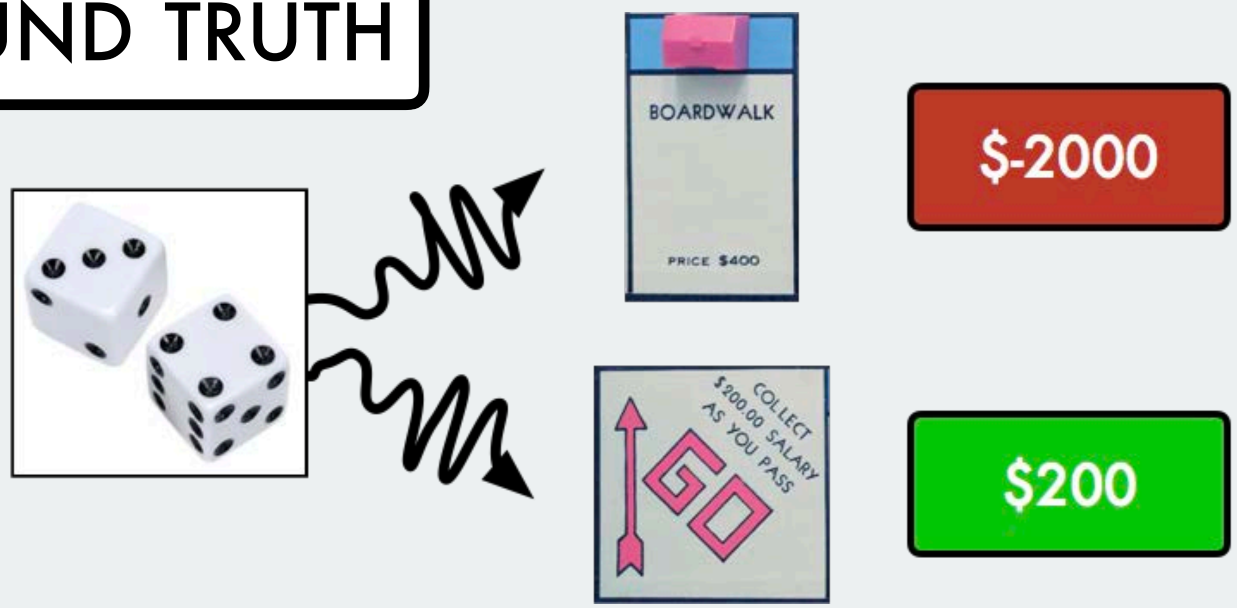


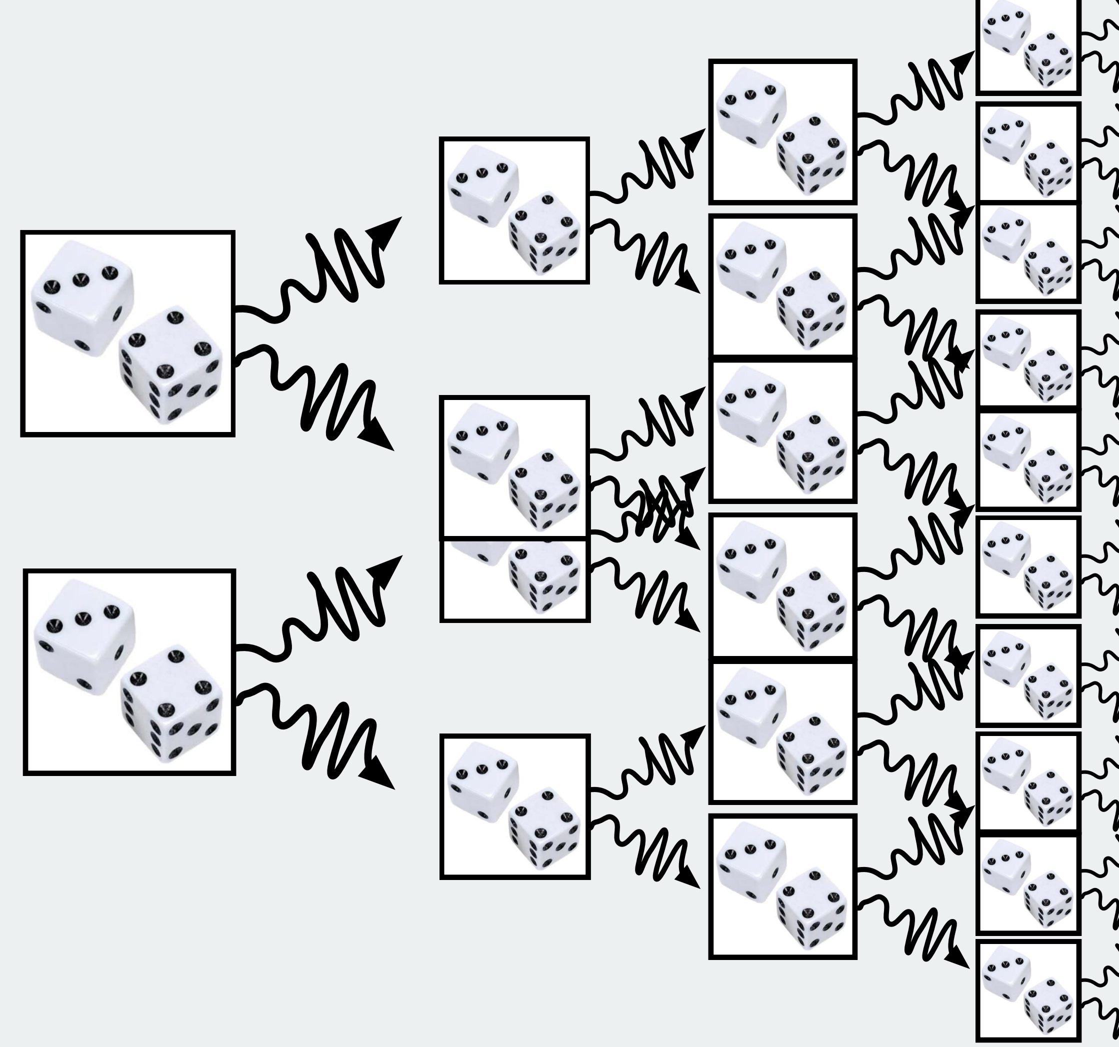
$$R(x_0) + \gamma R(x_1) + \gamma^2 R(x_2) \dots$$

Bellman equation

$$V^\pi(x) = \mathbf{E}\, R(x) + \gamma \mathop{\mathbf{E}}_{x' \sim P^\pi} V^\pi(x')$$

Ground truth





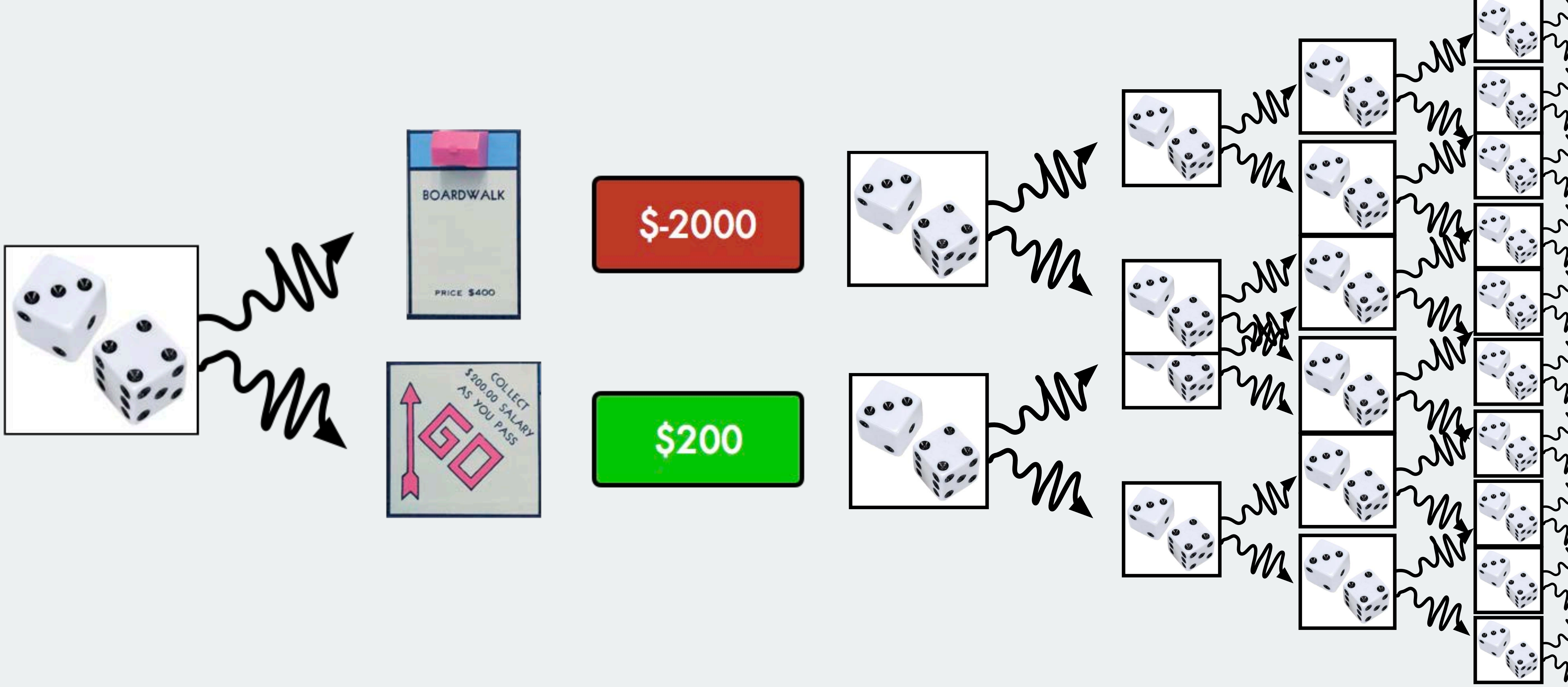
$$V^\pi(x) = \mathbf{E}\, R(x) + \gamma \underset{x' \sim P^\pi}{\mathbf{E}} V^\pi(x')$$

Implied model

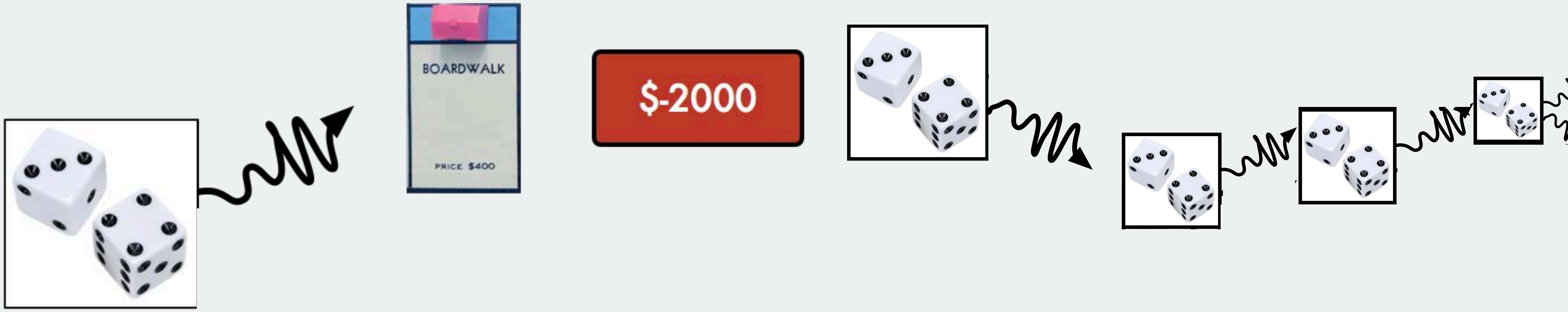
Schoknecht 2002; Parr et al., 2008; Sutton et al. 2008

Implied model

$$
\begin{aligned}
V^\pi(x) &= \mathbf{E}\left[R(x_0) + \gamma R(x_1) + \gamma^2 R(x_2) + \cdots \mid x_0 = x, a_t \sim \pi\right] \\
&= \mathbf{E}\left[R(x_0)\right] + \gamma\,\mathbf{E}\left[R(x_1)\right] + \gamma^2\,\mathbf{E}\left[R(x_2)\right] + \cdots .
\end{aligned}
$$



VENESS, BELLEMARE, HUTTER, CHUA, DESJARDINS (2015)



$$V^{\pi}(x) = \mathbf{E}\left[R(x_0) + \gamma R(x_1) + \gamma^2 R(x_2) + \cdots\right]$$

Veness, Bellemare, Hutter, Chua, Desjardins (2015)

Bellman equation

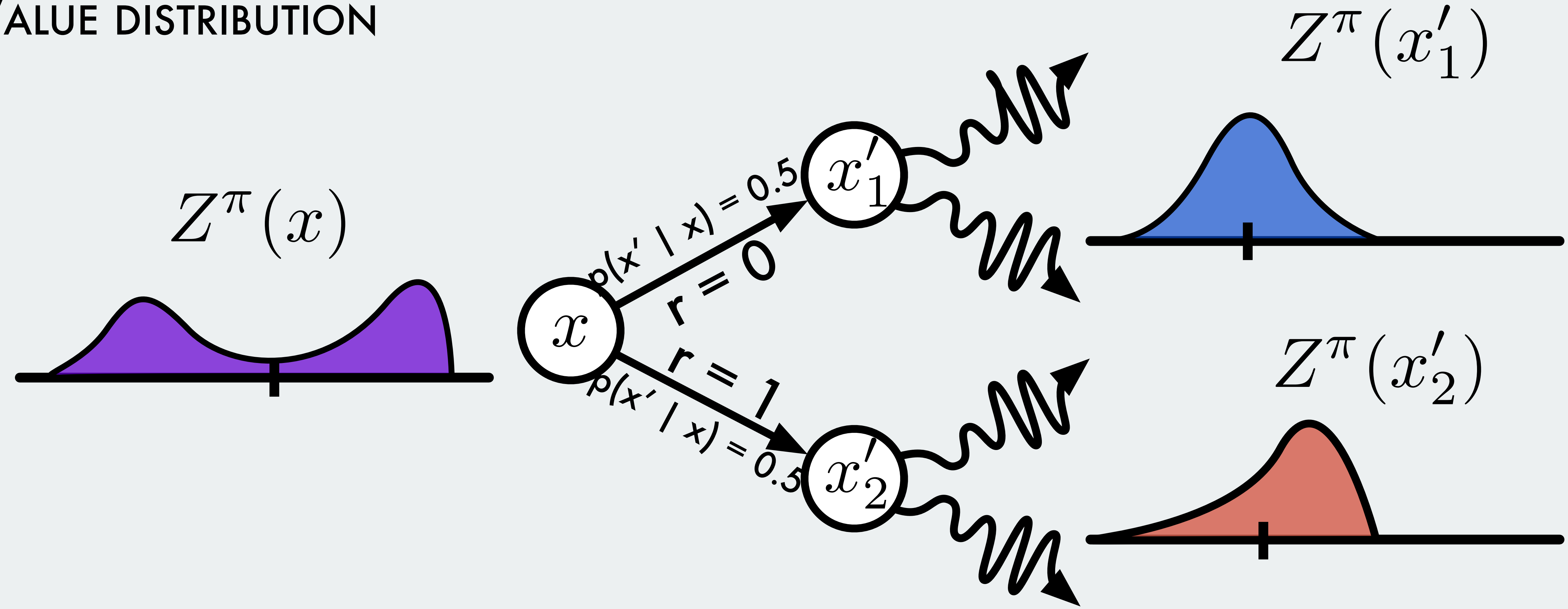
$$\cancel{\mathbf{E}}\left[Z^{\pi}(x)\right] = \cancel{\mathbf{E}}\,R(x) + \gamma\,\cancel{\mathbf{E}}\left[Z^{\pi}(X')\right]$$

Distributional Bellman equation

$$Z^{\pi}(x) \stackrel{D}{=} R(x) + \gamma Z^{\pi}(X')$$

$$Z^\pi(x) \stackrel{D}{=} R(x) + \gamma Z^\pi(X')$$

VALUE DISTRIBUTION



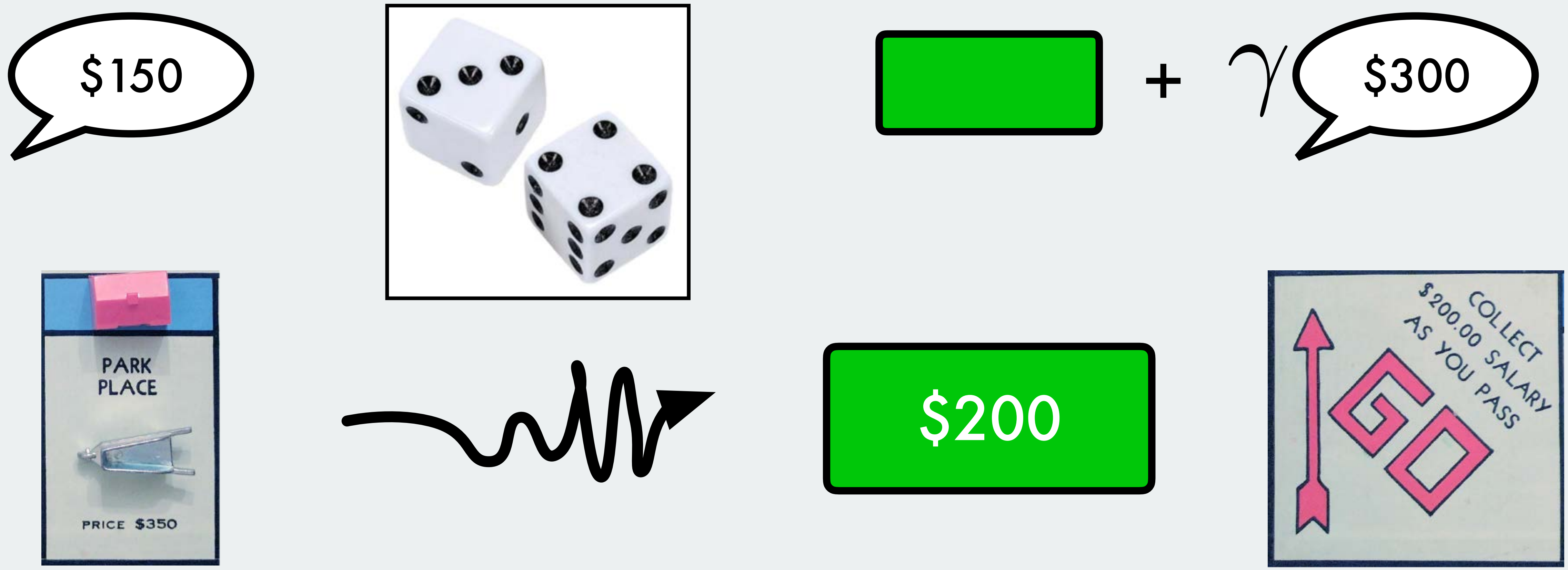
Bellman (1957): Bellman equation for mean

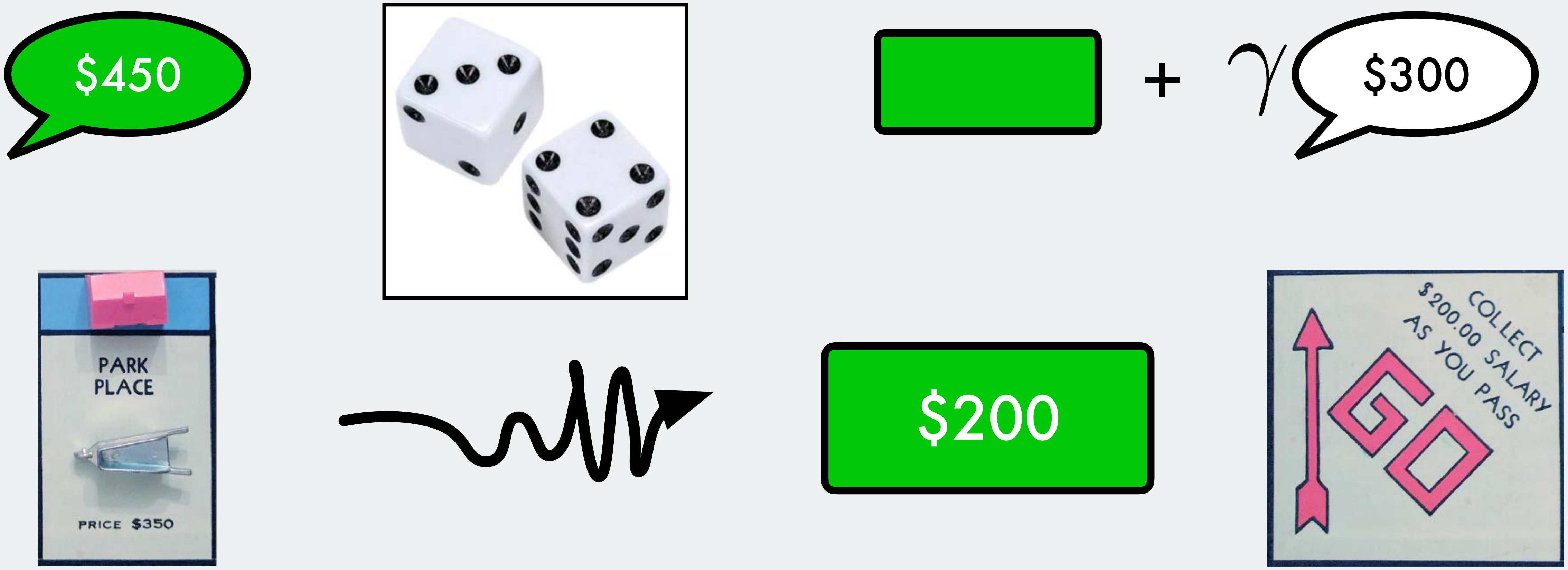
Sobel (1982): ... for variance

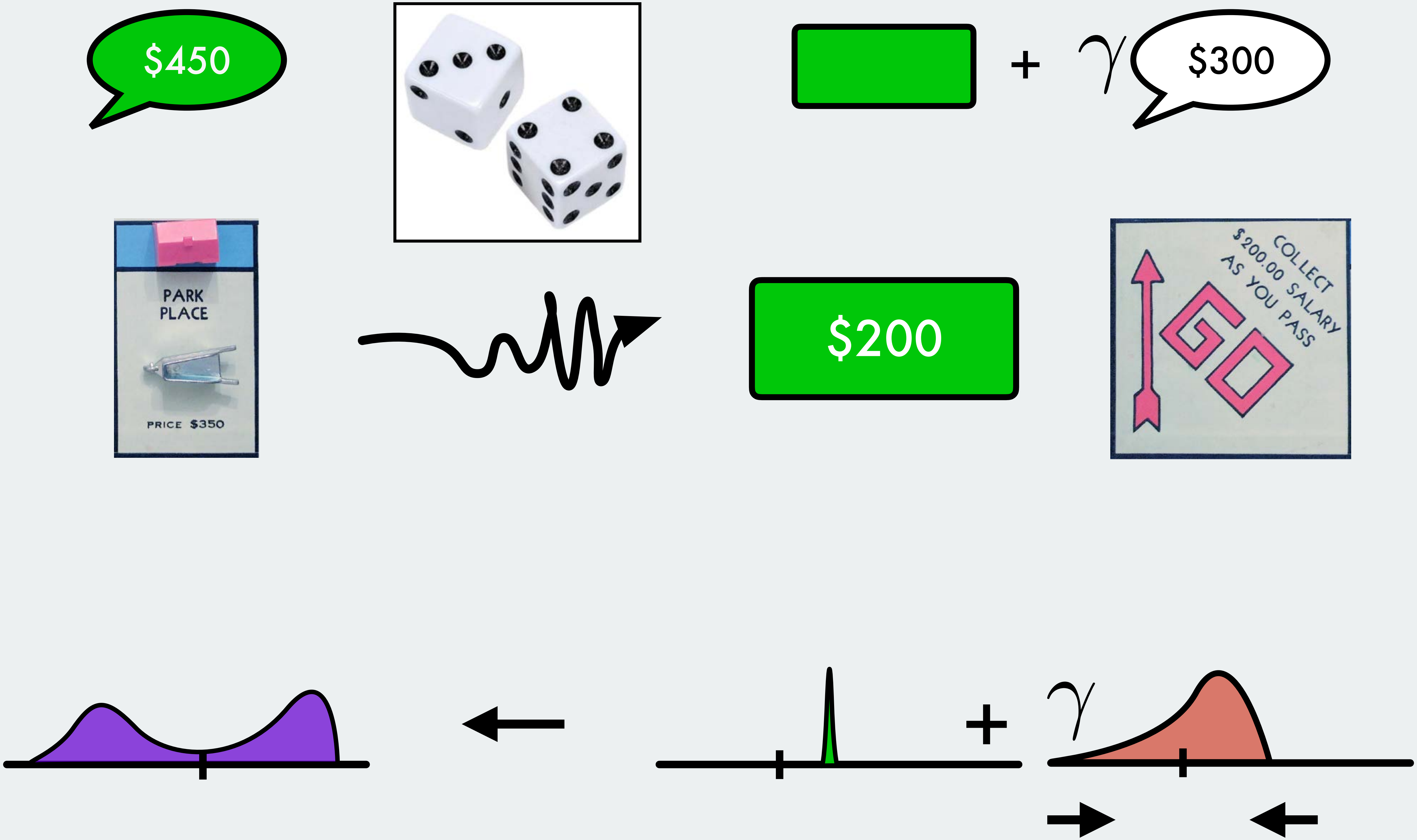
Engel (2003): ... for Bayesian uncertainty

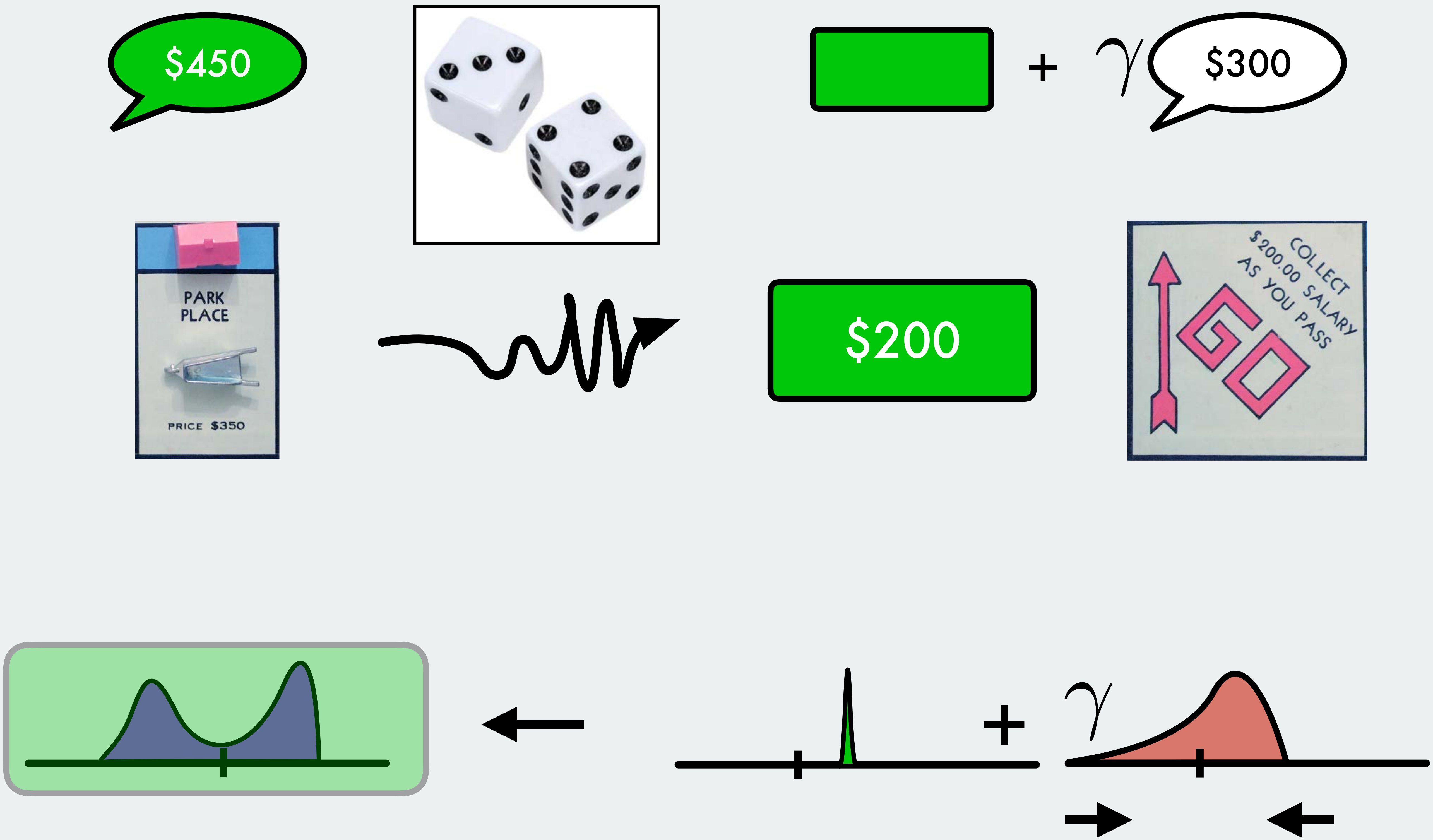
Azar et al. (2011), Lattimore & Hutter (2012): ... for higher moments

Morimura et al. (2010, 2010b): ... for densities

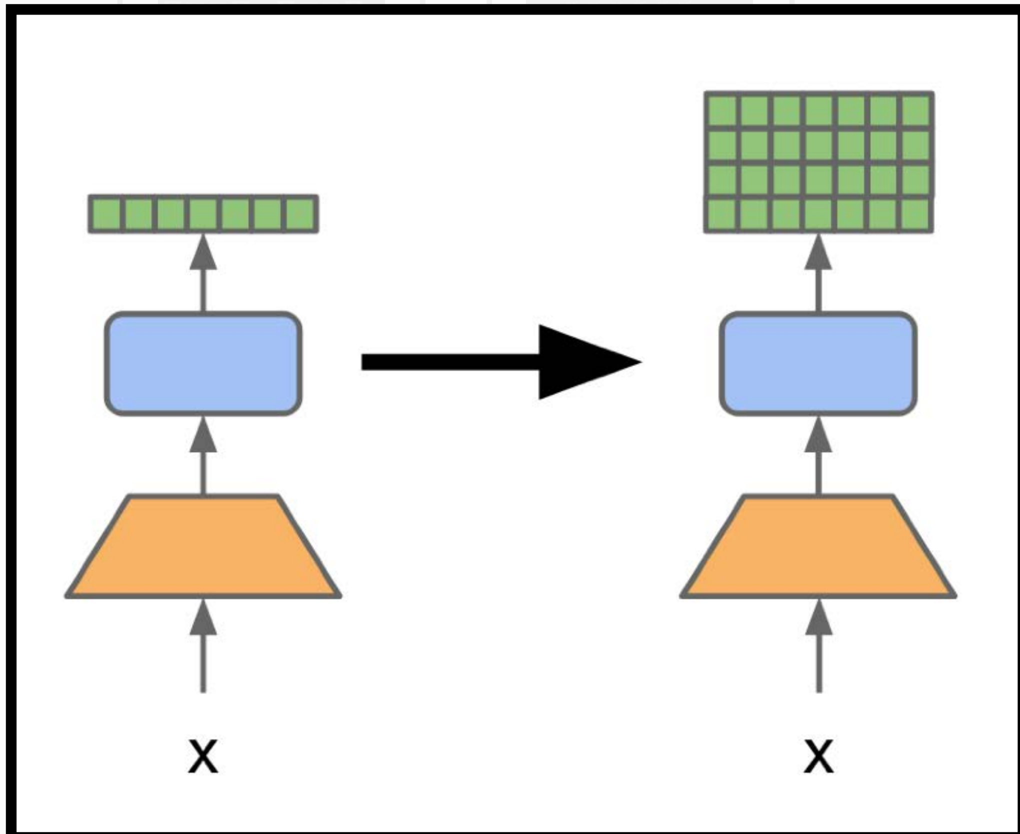

$150
+ γ
$300
PARK PLACE
PRICE $350
$200
COLLECT $200.00 SALARY AS YOU PASS
GO


$450
+ γ
$300
PARK PLACE
PRICE $350
$200
COLLECT $200.00 SALARY AS YOU PASS
GO

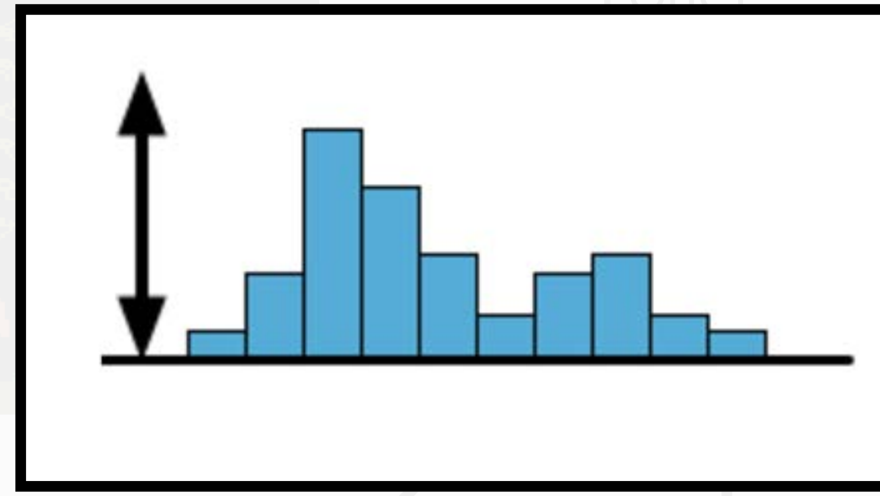

$450
+ γ
$300
PARK PLACE
PRICE $350
$200
COLLECT
$200.00 SALARY
AS YOU PASS
GO
+ γ

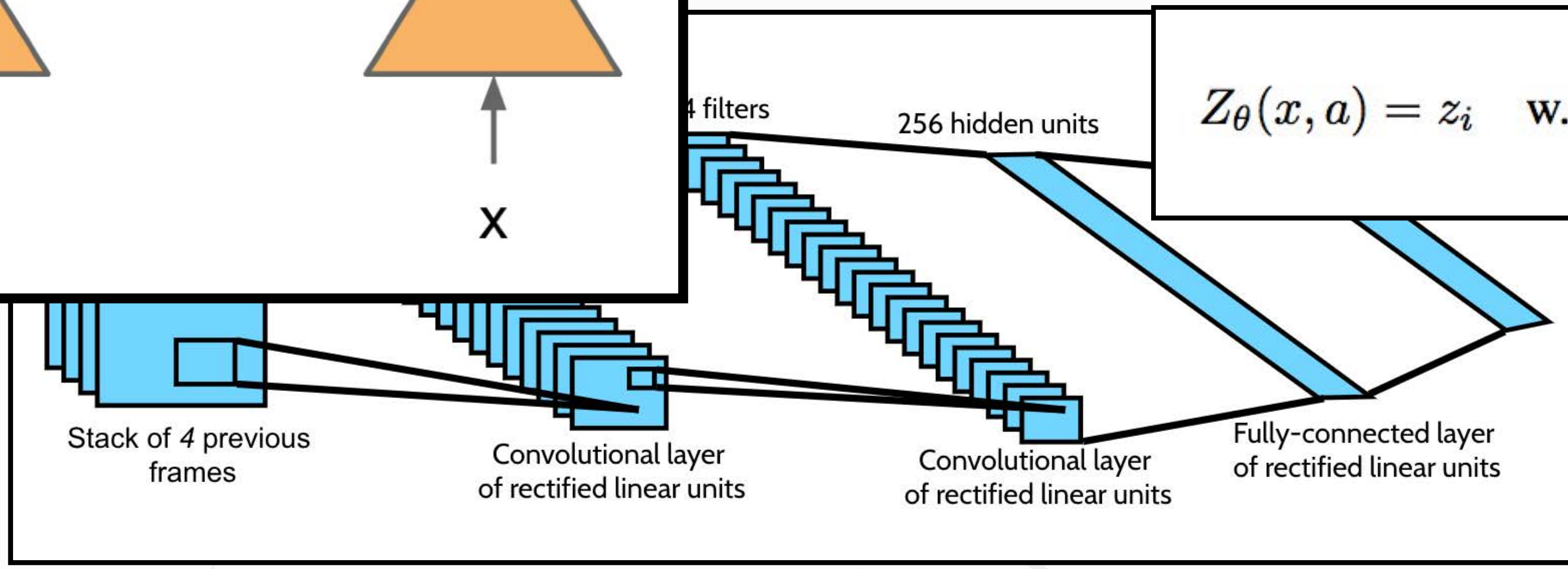

$450
$300
γ
PARK PLACE
PRICE $350
$200
COLLECT $200.00 SALARY AS YOU PASS
GO
γ

$Q(x, a)$ $\qquad$ $p(z \mid x, a)$



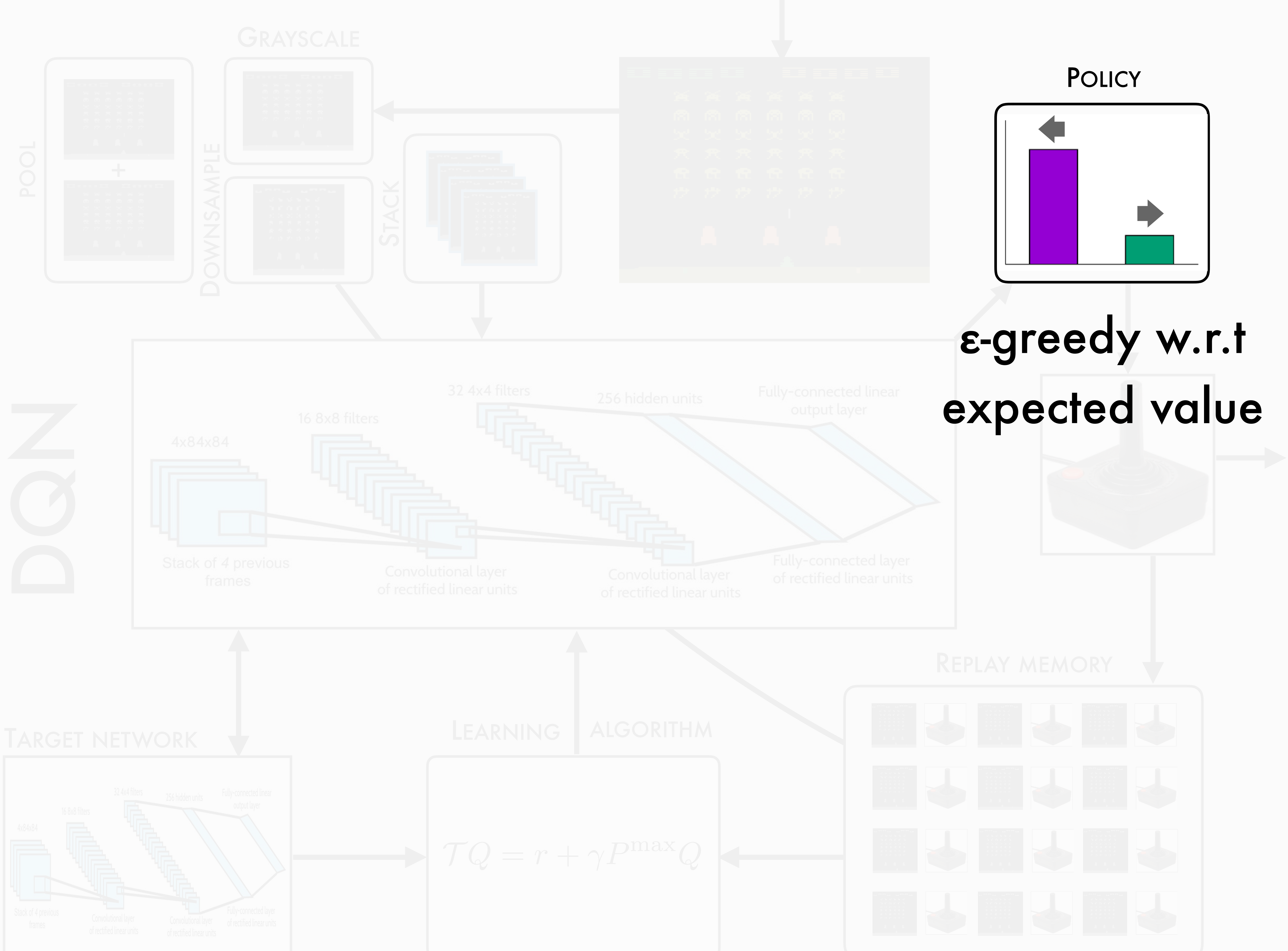
Discrete distribution





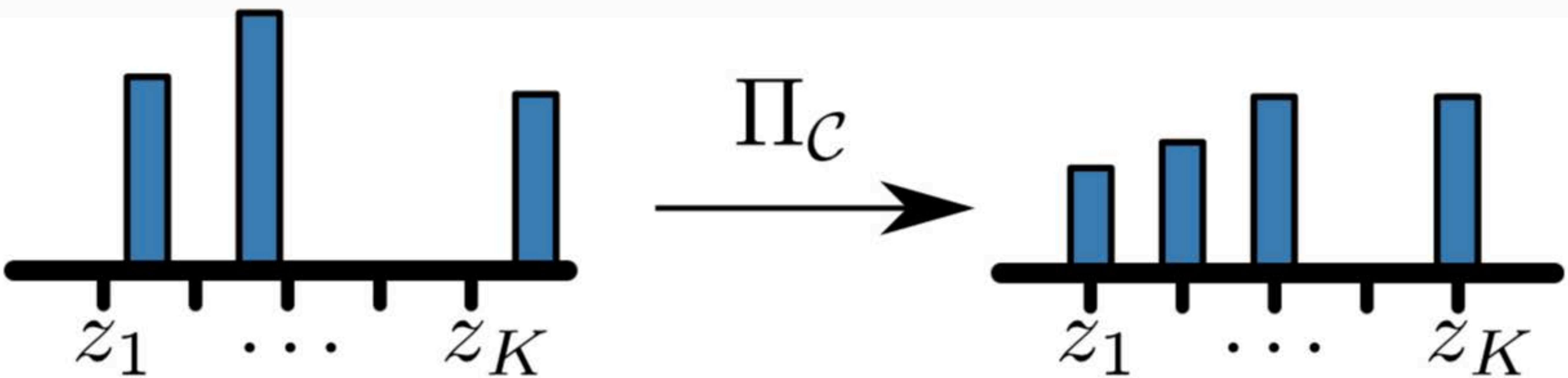
$$Z_\theta(x, a) = z_i \quad \text{w.p.} \quad p_i(x, a) := \frac{e^{\theta_i(x,a)}}{\sum_j e^{\theta_j(x,a)}}.$$

C51



ε-greedy w.r.t expected value

APPROXIMATION



PROJECTION

$$\boxed{\mathcal{T}^\pi Z = R + \gamma P^\pi Z}$$

1. From x, a, sample a transition:

$$r, X', A' \sim R(x,a), P(\cdot \,|\, x,a), \pi(\cdot \,|\, X')$$

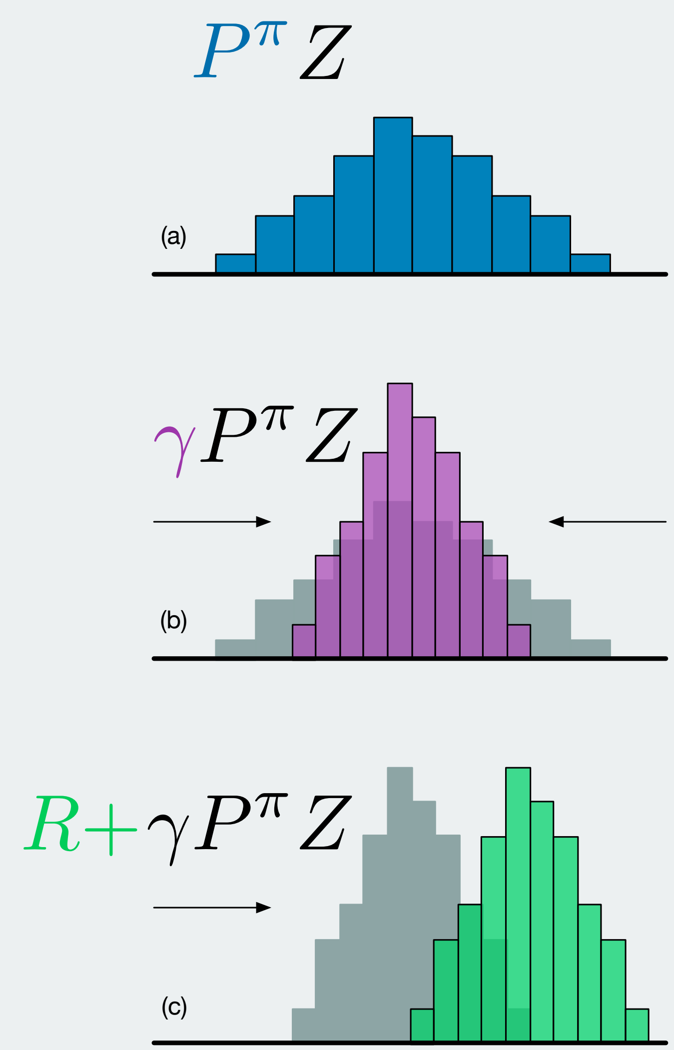
2. Compute sample backup

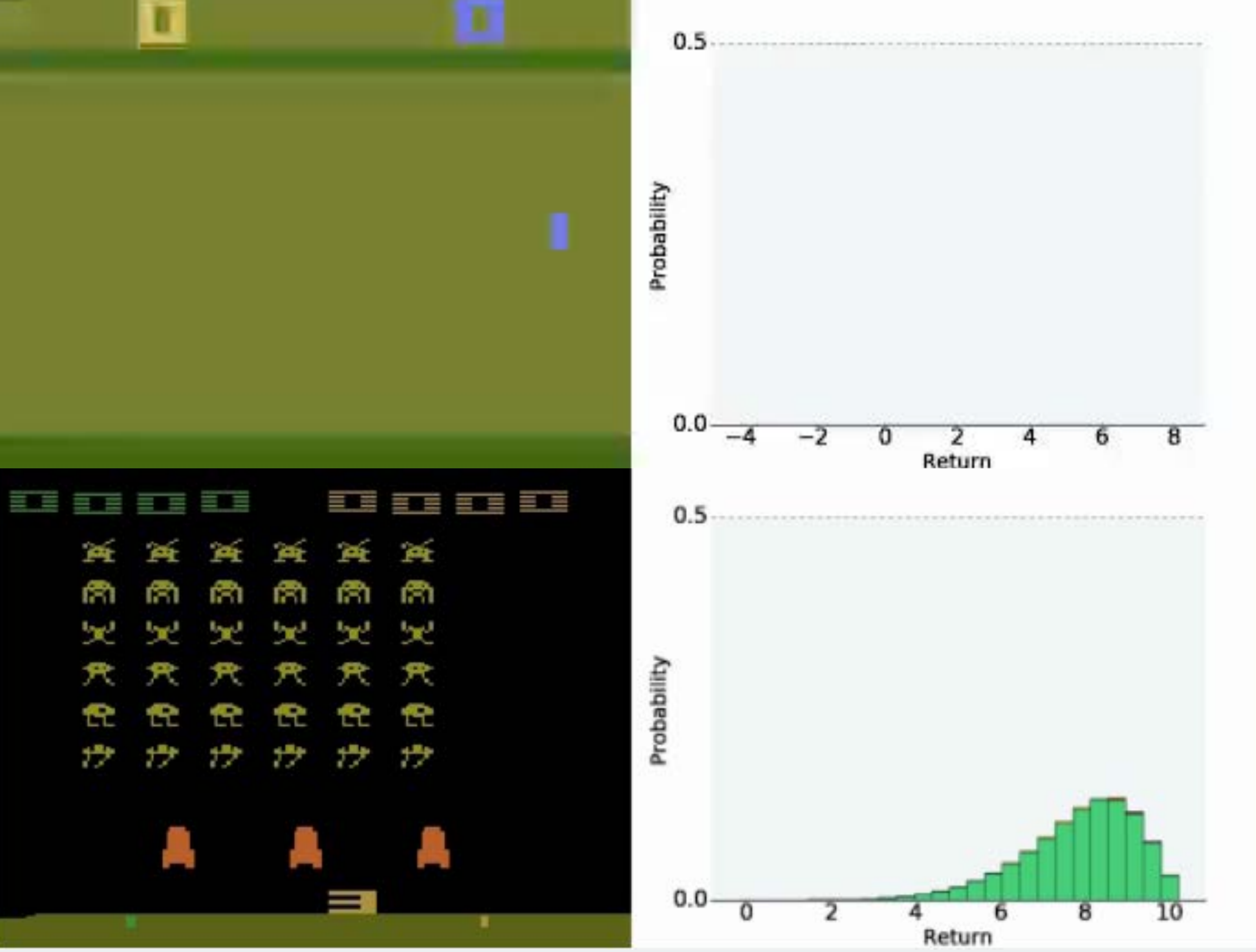
$$\hat{\mathcal{T}}^\pi Z(x,a) := r + \gamma Z(X', A')$$

3. Project onto approximation support:

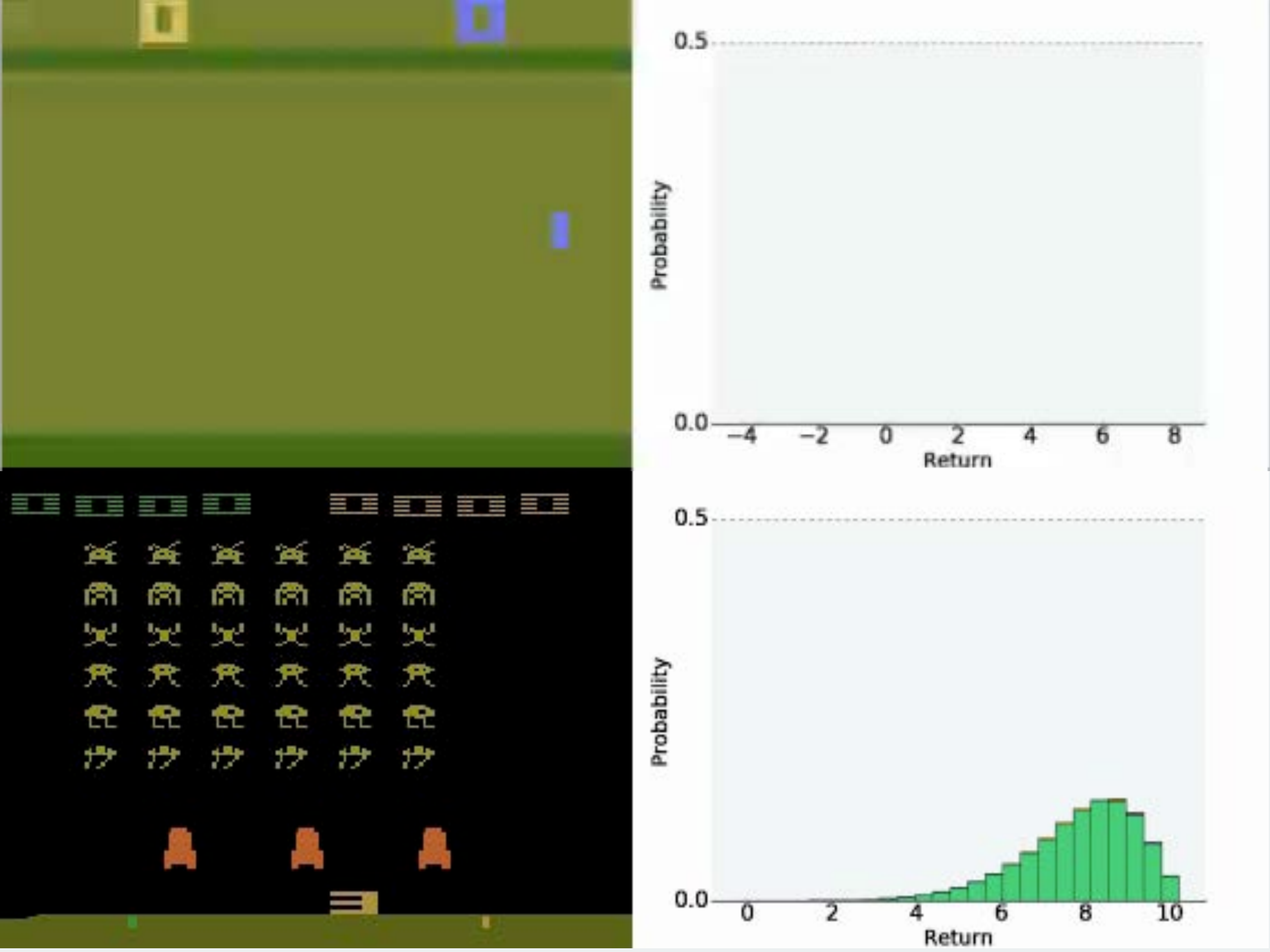
$$\Phi \hat{\mathcal{T}}^\pi Z(x,a)$$

4. Update towards projection, i.e. take a KL-minimizing step

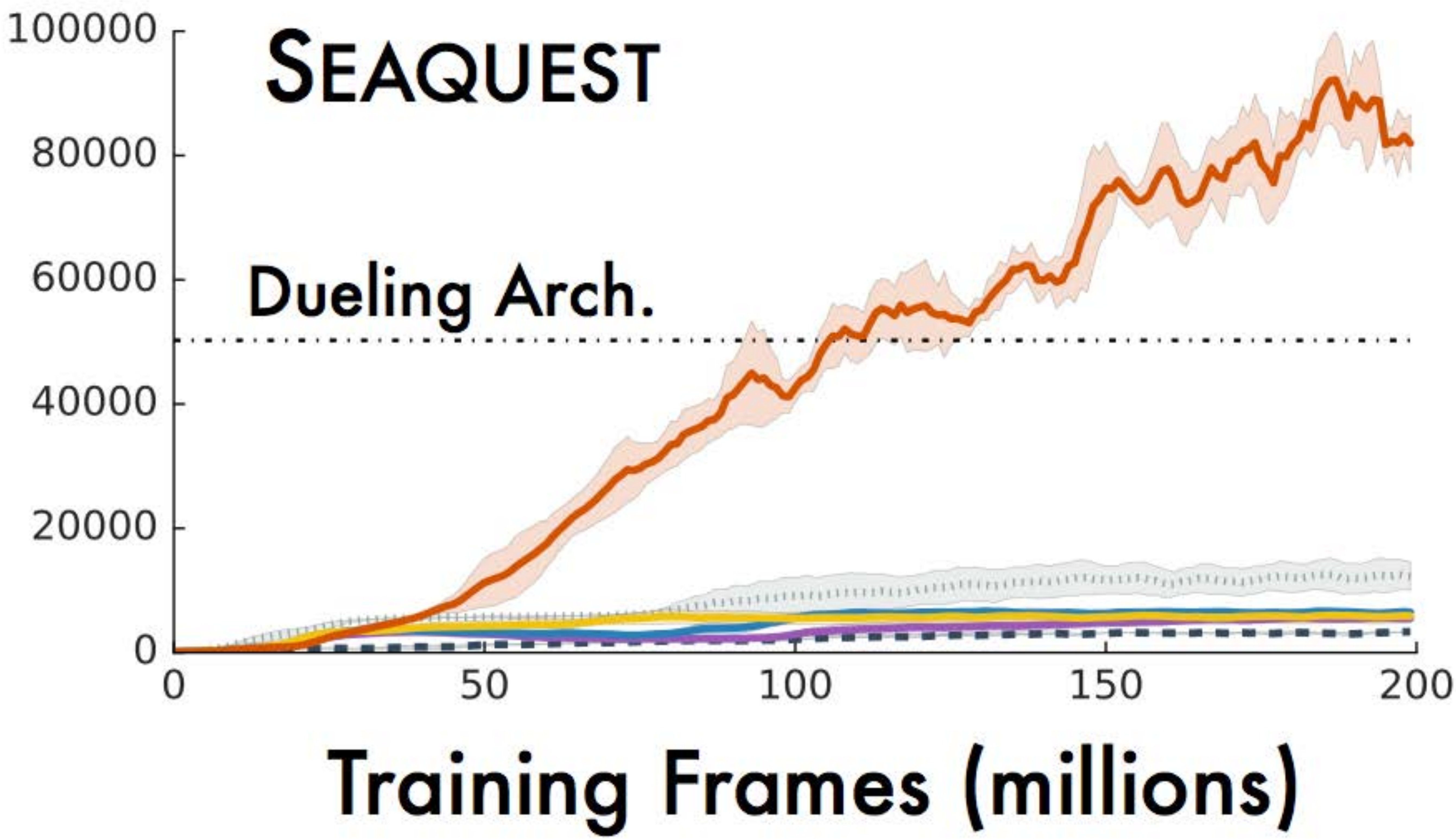




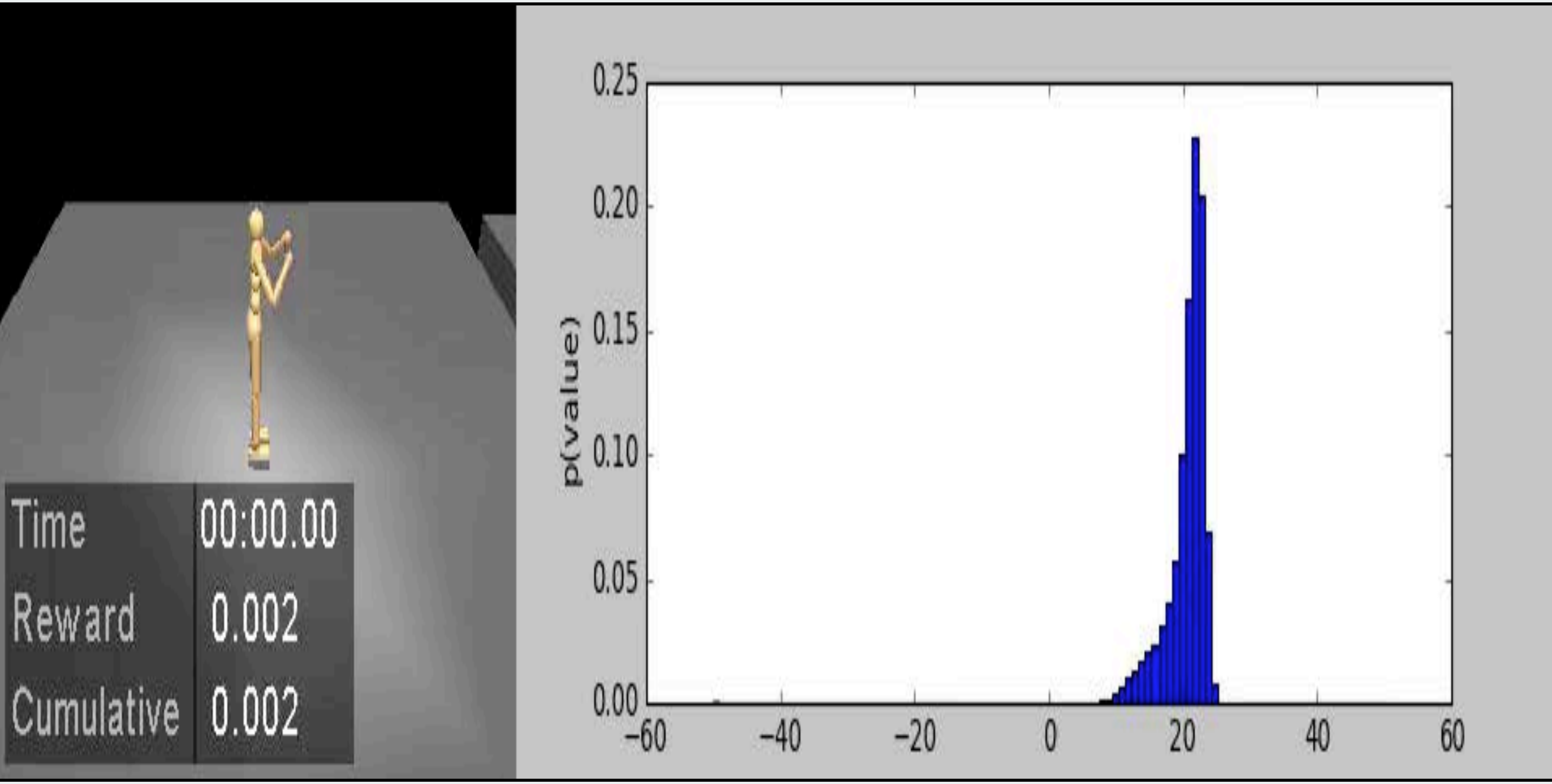
0.5
Probability
0.0
−4
−2
0
2
4
6
8
Return
0.5
Probability
0.0
0
2
4
6
8
10
Return


0.5
Probability
0.0
−4
−2
0
2
4
6
8
Return
0.5
Probability
0.0
0
2
4
6
8
10
Return

	Mean	Median	> H.B.	> DQN
DQN	228%	79%	24	0
DDQN	307%	118%	33	43
DUEL.	373%	151%	37	50
PRIOR.	434%	124%	39	48
PR. DUEL.	592%	172%	39	44


SEAQUEST
Dueling Arch.
100000
80000
60000
40000
20000
0
0
50
100
150
200
Training Frames (millions)


Time
00:00.00
Reward
0.002
Cumulative
0.002



Time
00:00.00
Reward
0.002
Cumulative
0.002
p(value)
0.25
0.20
0.15
0.10
0.05
0.00
−60
−40
−20
0
20
40
60

Distributional perspective

Bellemare et al., 2016

Quantile Regression

Dabney et al., 2017

Implicit Quantile Networks

Dabney et al., 2018

Distributional Actor-Critic

Barth-Maron et al., 2018

Generative Quantile Networks

Ostrovski et al., 2018

The Rainbow Agent

Hessel et al., 2018

Cramer distance

Bellemare et al., 2017

Analysis of C51

Rowland et al., 2017

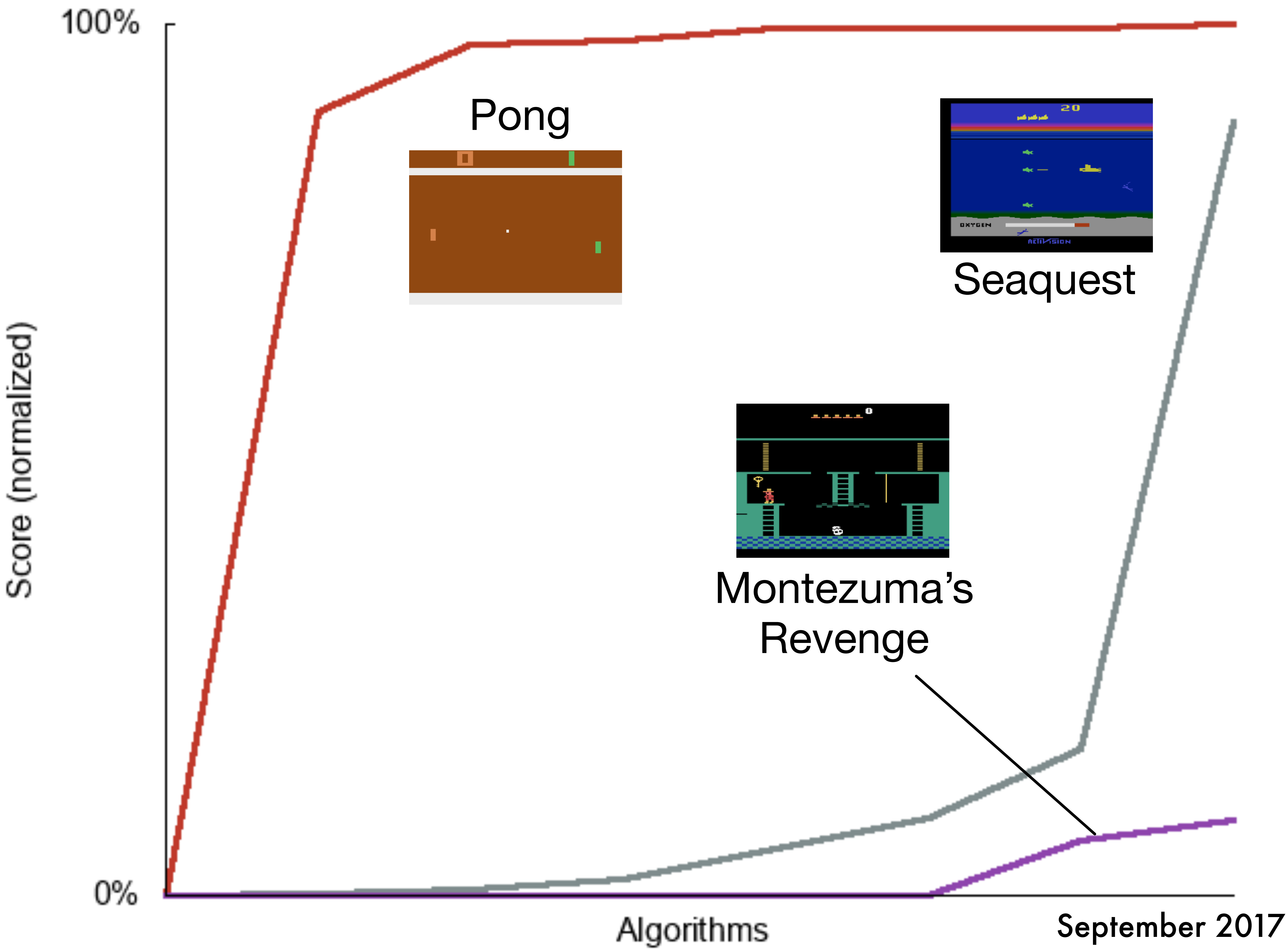
"As Expected"

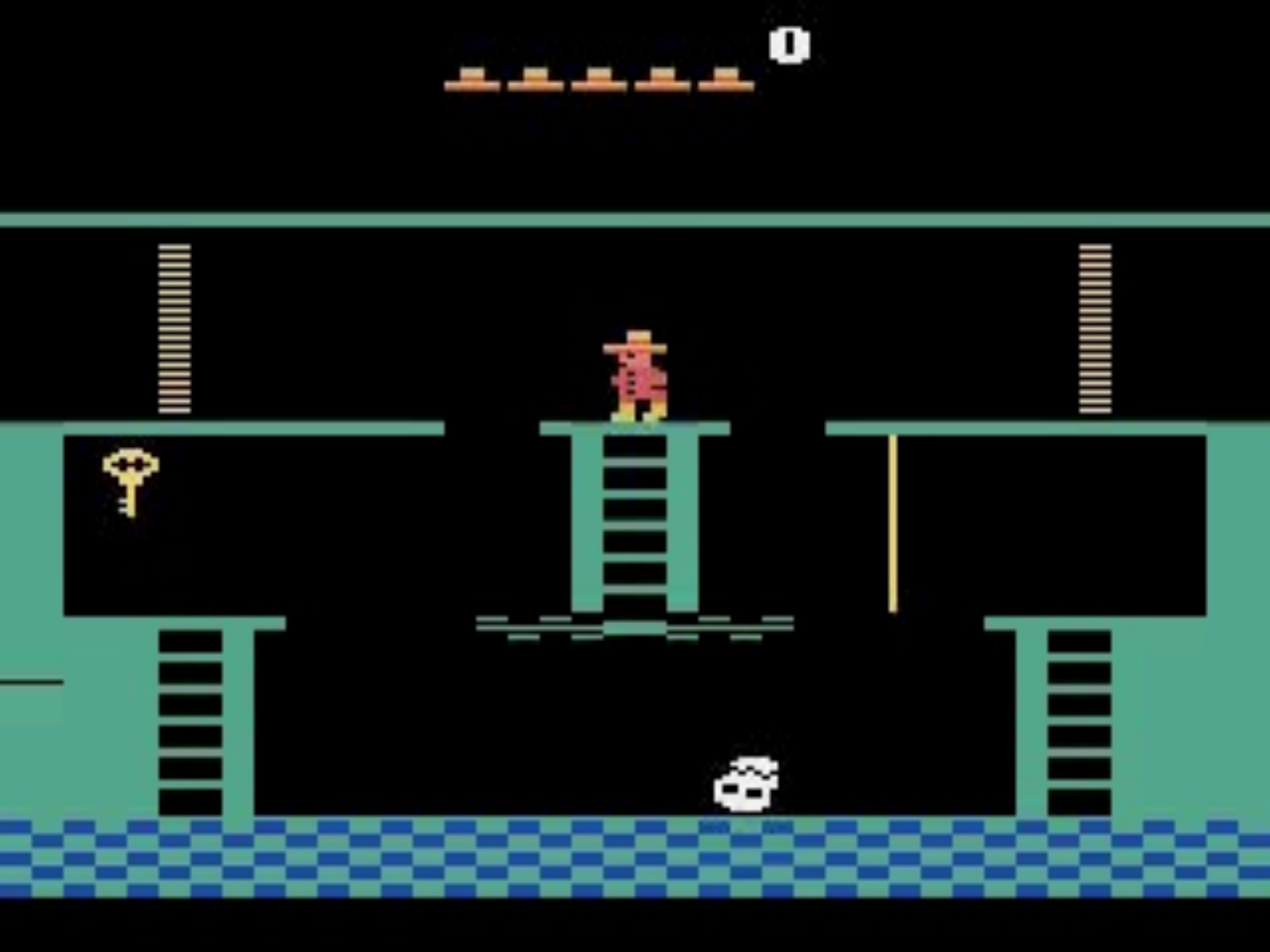
Lyle et al., 2018

PGMRL Workshop, ICML

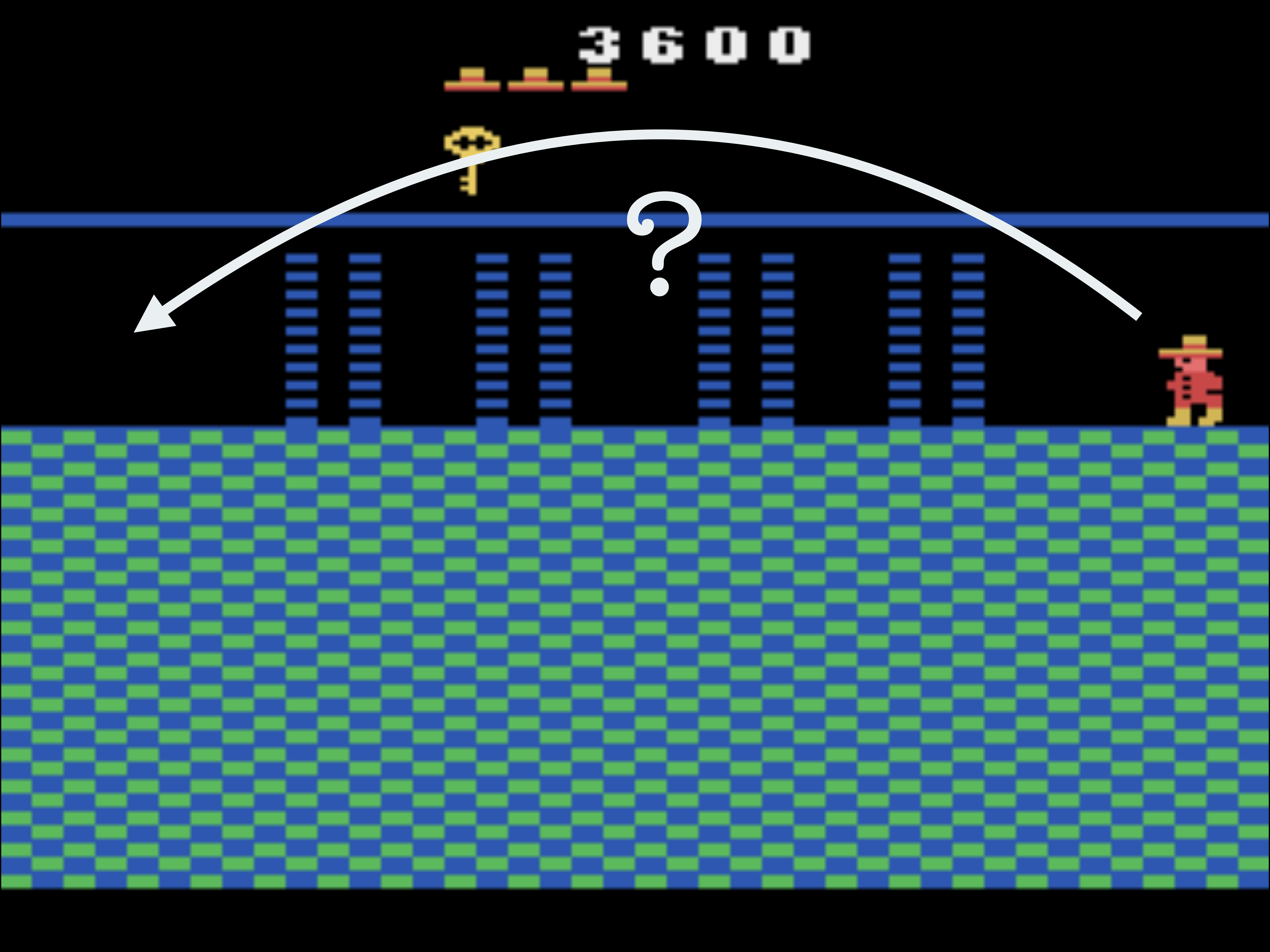
1. Distributional reinforcement learning

2. Exploration with pseudo-counts


100%
0%
Score (normalized)
Algorithms
September 2017
Pong
Seaquest
Montezuma's Revenge


0


0
3600
6900
8100


3600
?

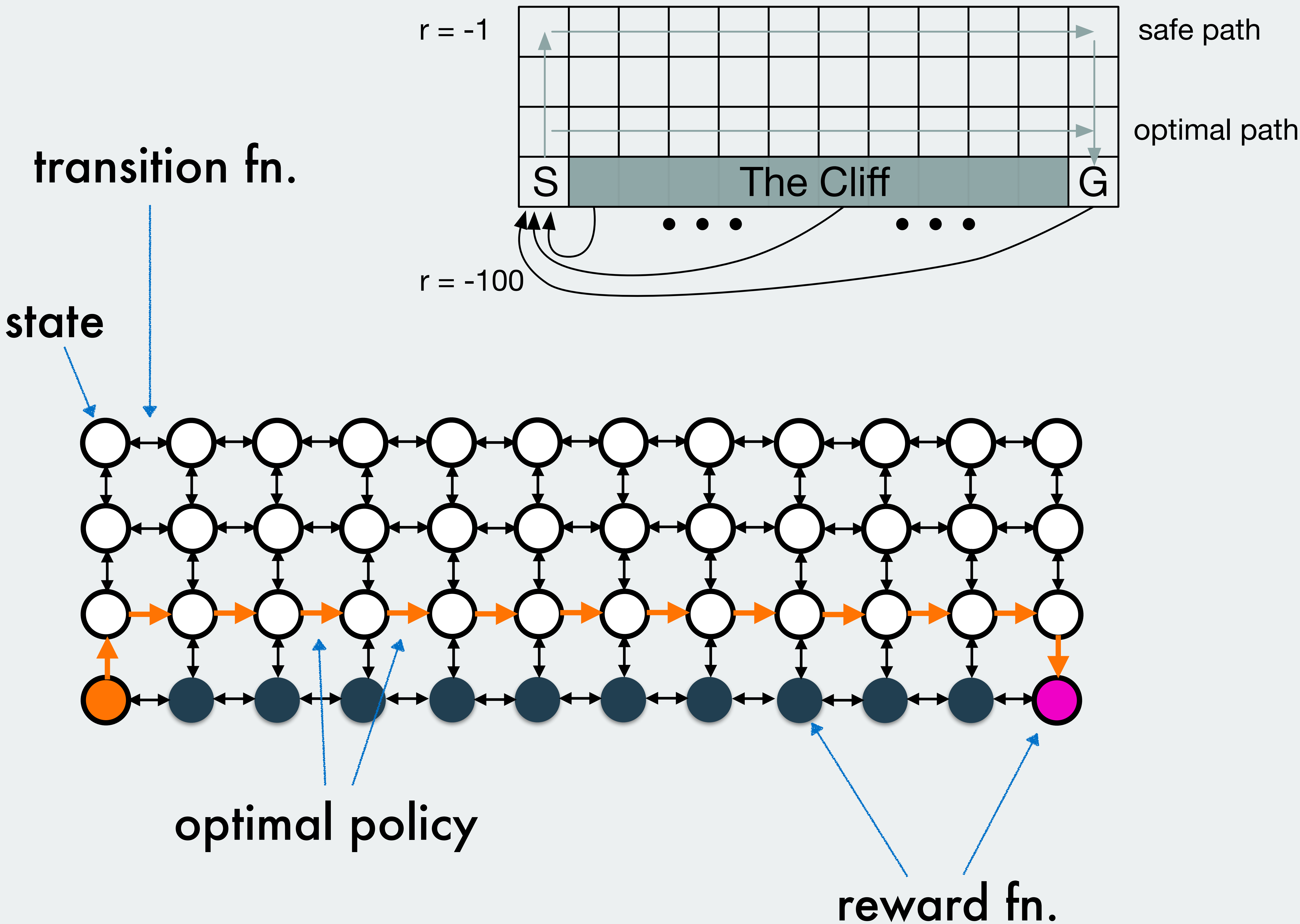
Exploration

Bellman equation:

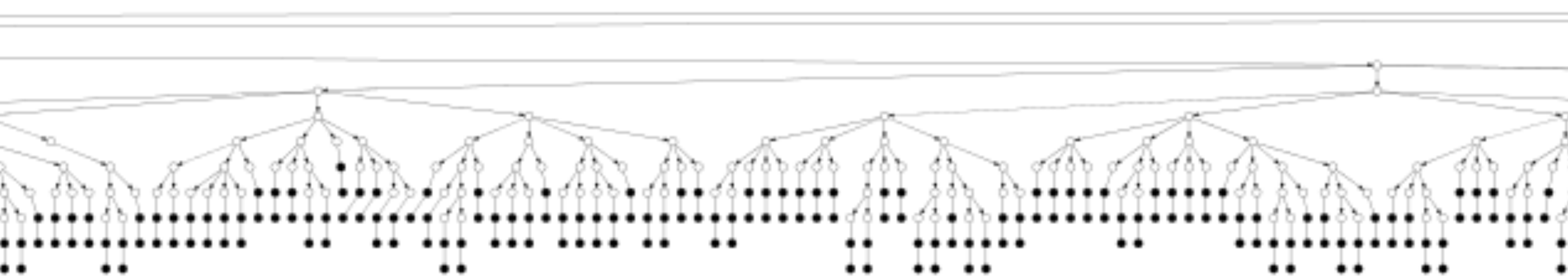
$$Q(x,a) = r(x,a) + \gamma \mathop{\mathbf{E}}_{x' \sim P} \max_{a' \in \mathcal{A}} Q(x',a')$$

Exploration bonus (Strehl and Littman, 2008):

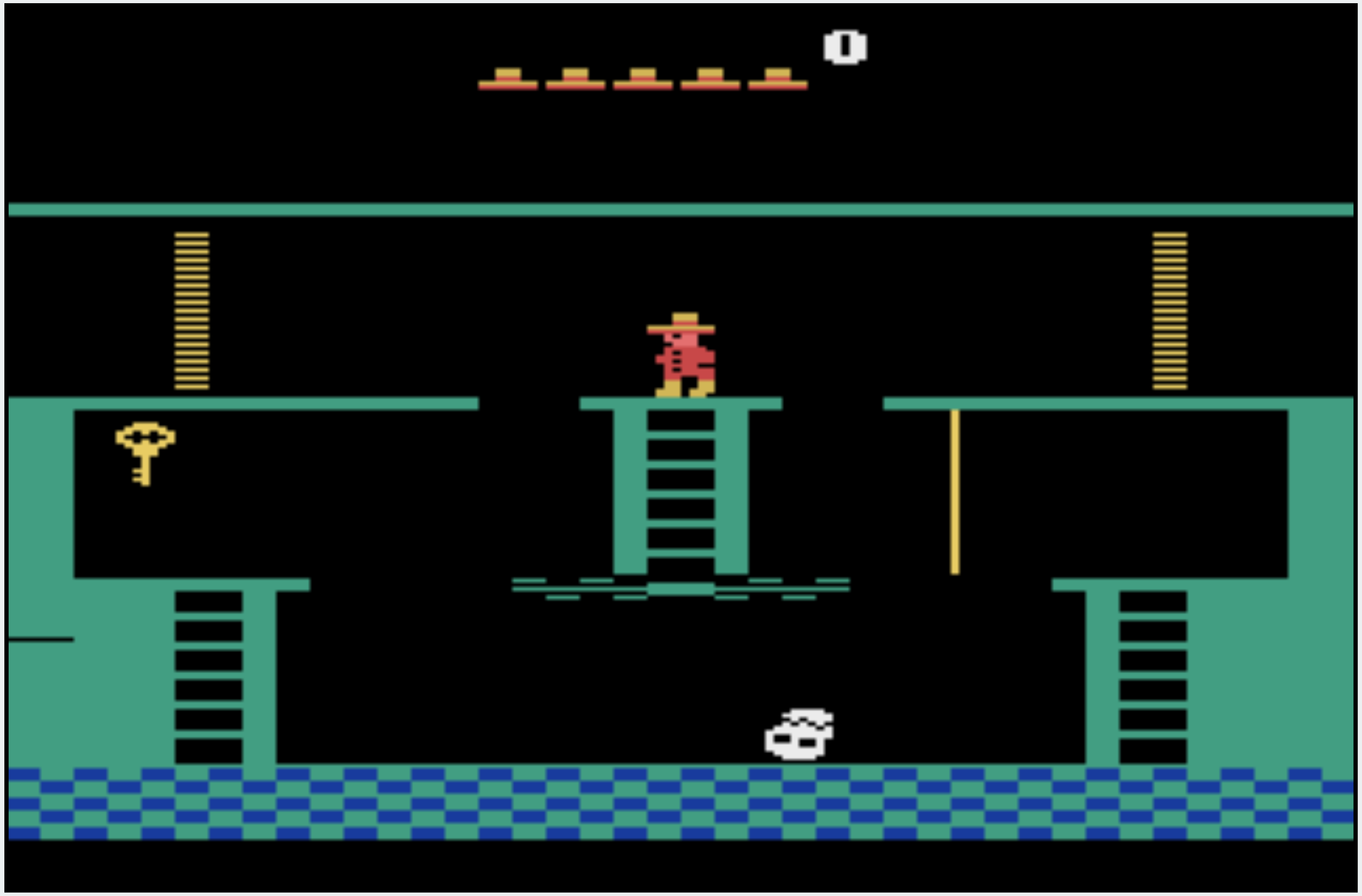
$$Q(x,a) = \hat{r}(x,a) + \gamma \mathop{\mathbf{E}}_{x' \sim \hat{P}} \max_{a' \in \mathcal{A}} Q(x',a') + \frac{\beta}{\sqrt{N(x,a)}}$$



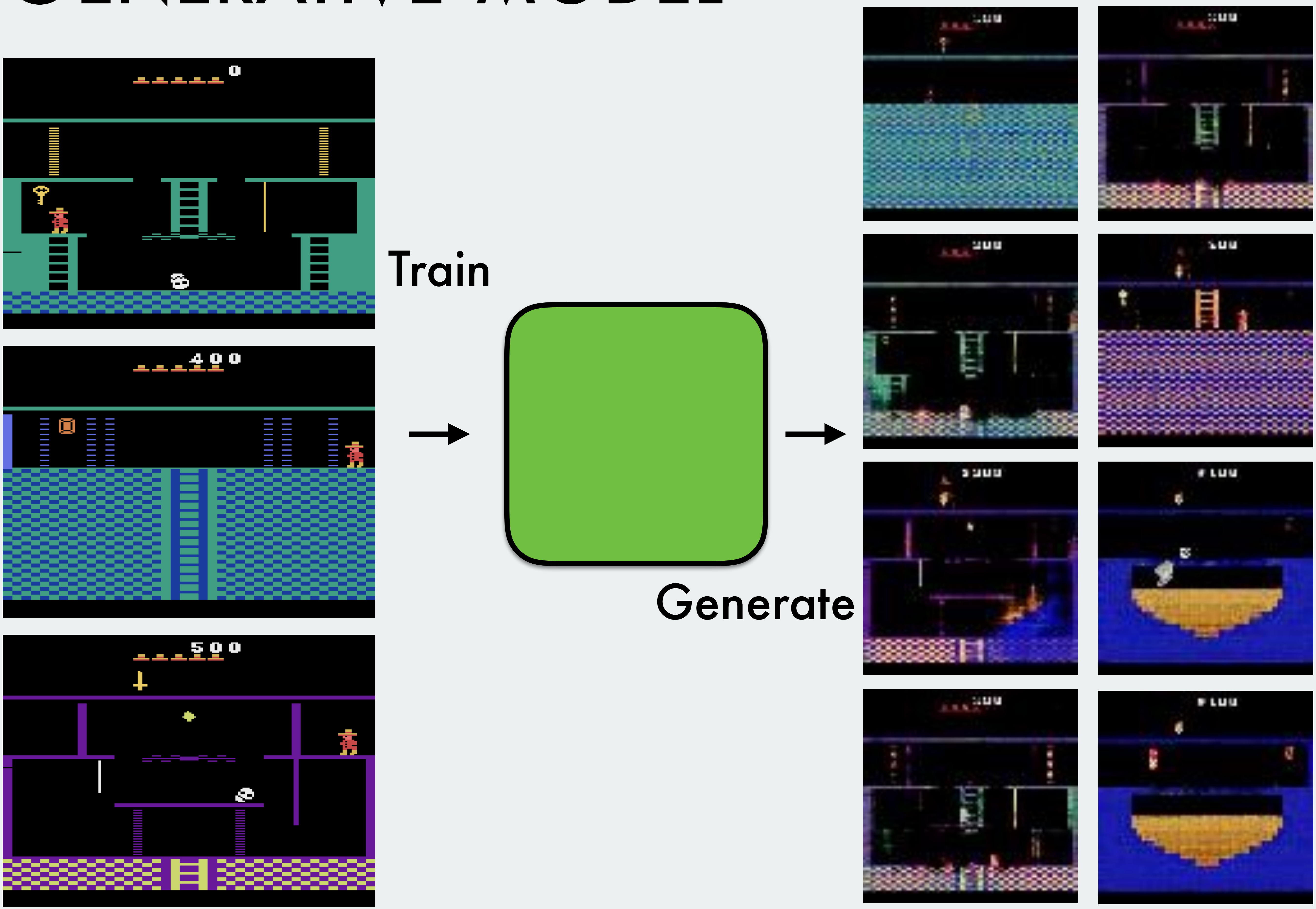
r = -1
safe path
optimal path
S
The Cliff
G
r = -100
transition fn.
state
optimal policy
reward fn.



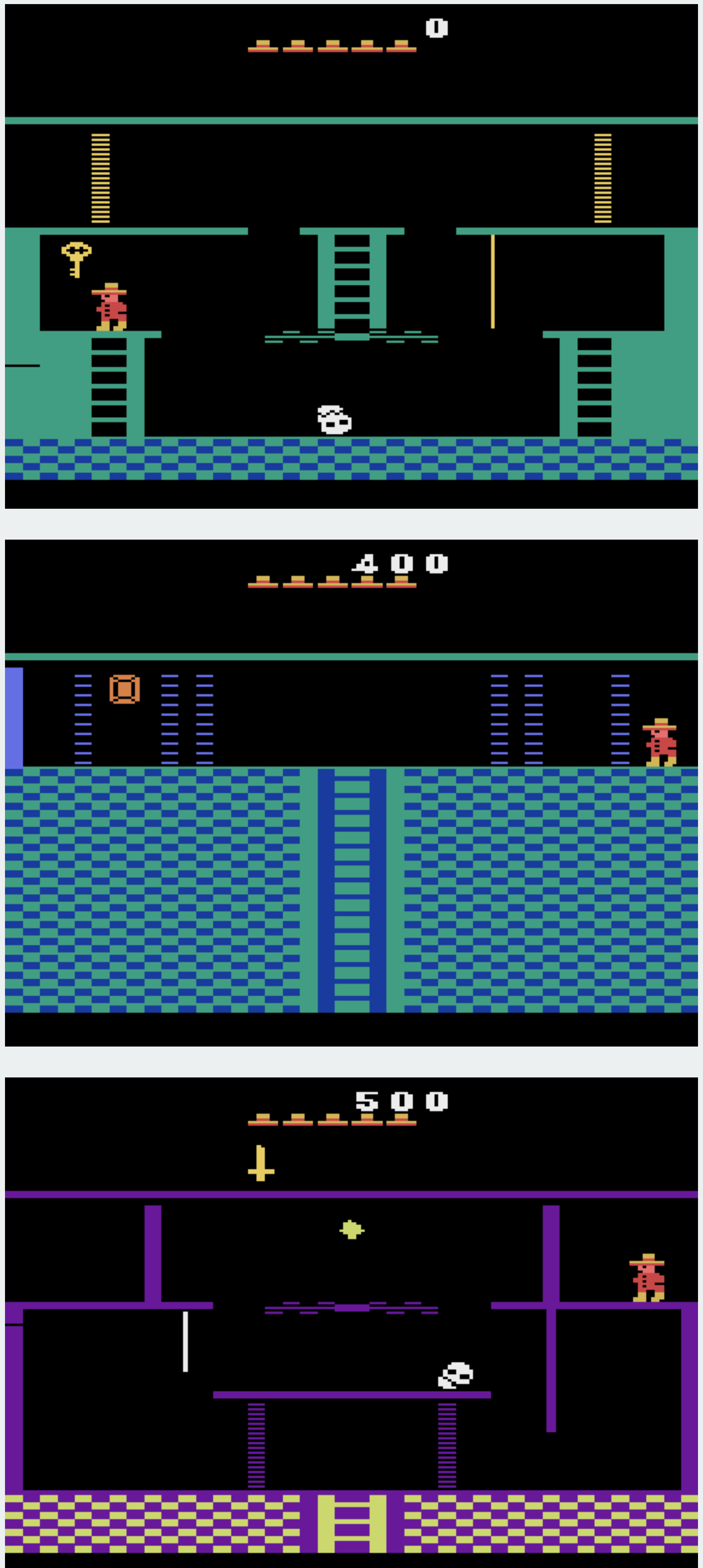
Most observations experienced *once*: up to 90% singletons (Blundell et al., 2016)



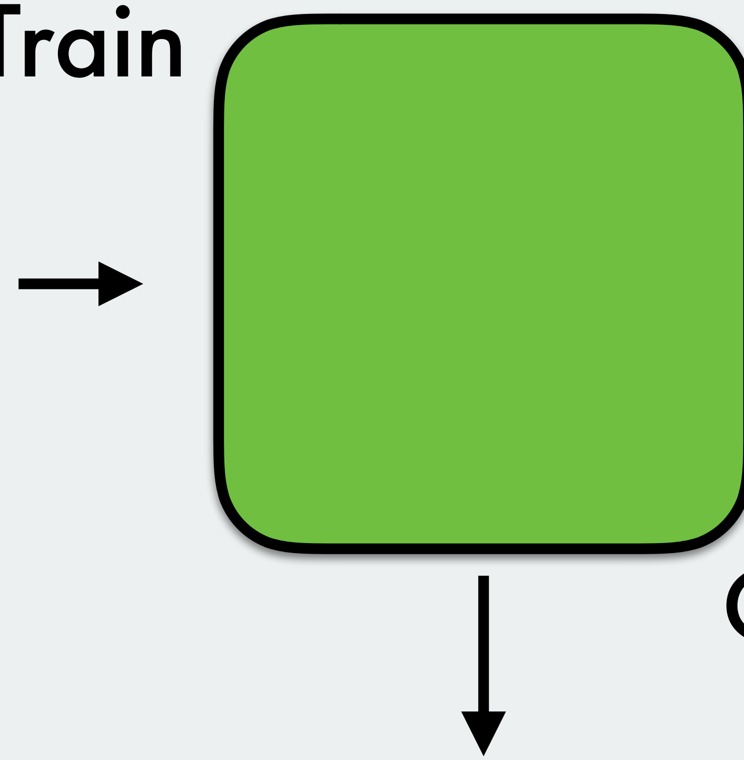
GENERATIVE MODEL



Density model



Train



Query

$\rho_n(x)$

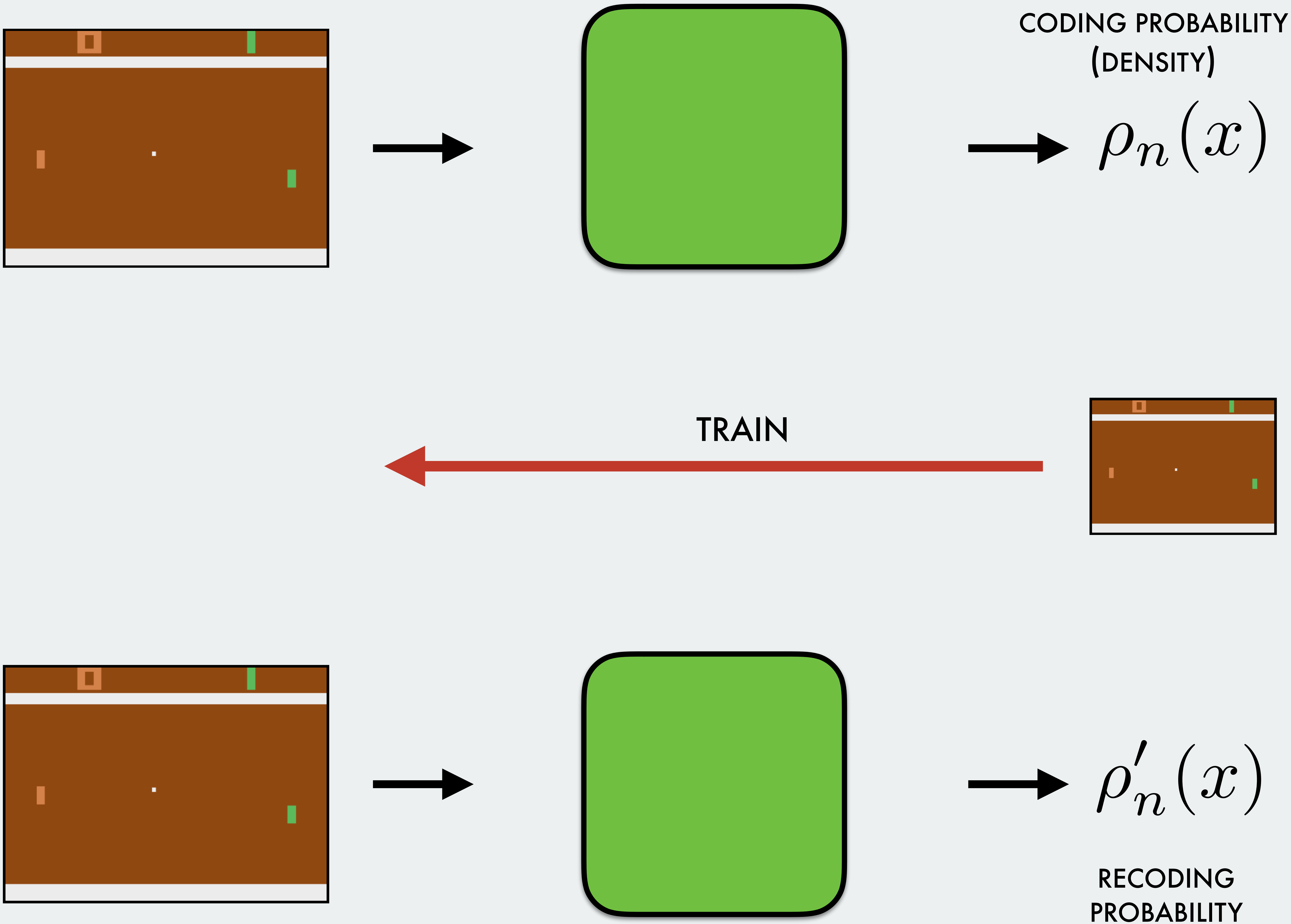
Assumptions

1. density $\propto$ frequency
2. model quality = generalization

$$\rho_n(x) = \frac{\hat{N}_n(x)}{\hat{n}} \qquad \rho'_n(x) = \frac{\hat{N}_n(x) + 1}{\hat{n} + 1}$$

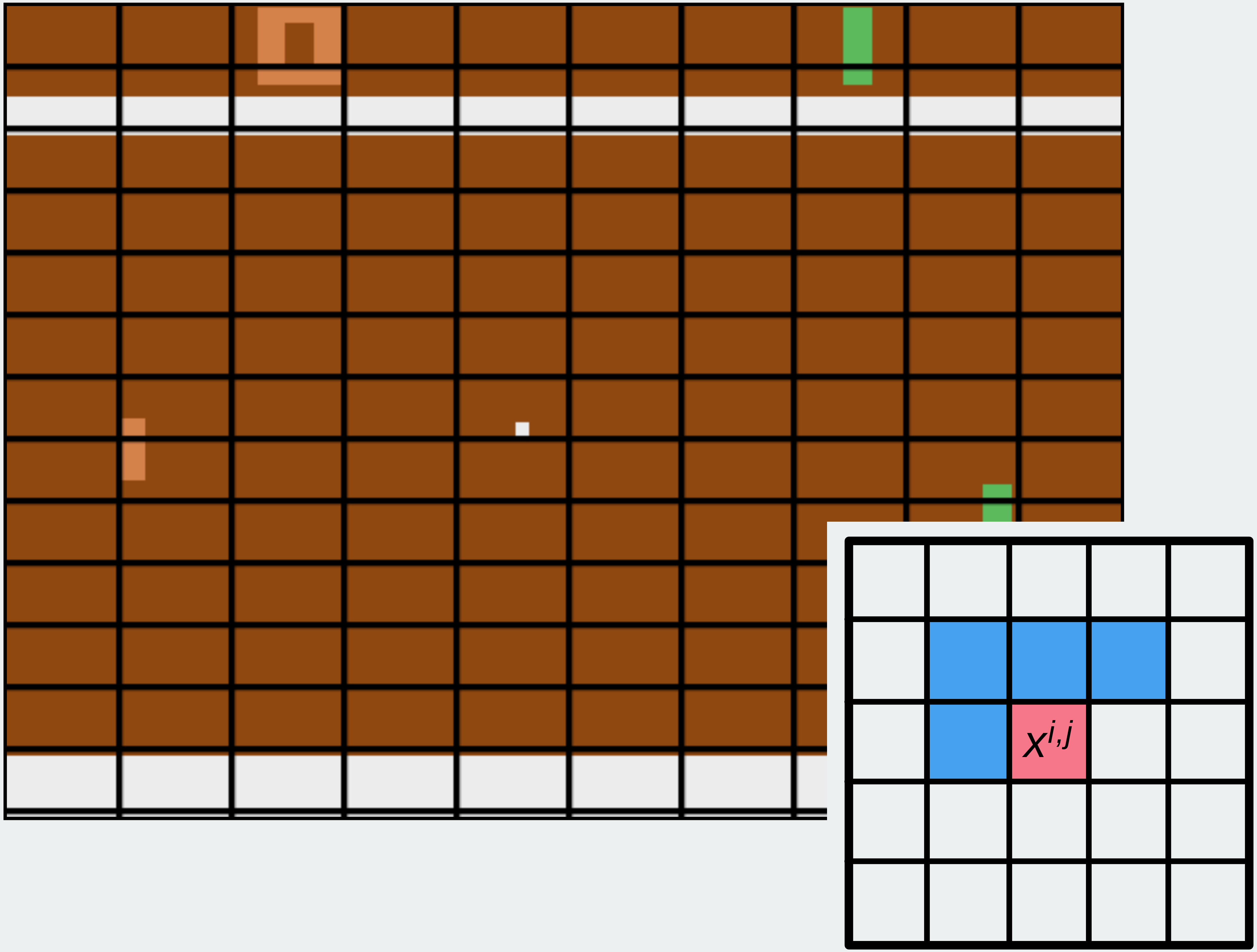
$$\hat{N}_n(x) = \rho_n(x)\,\frac{1 - \rho'_n(x)}{\rho'_n(x) - \rho_n(x)}$$

Bellemare, Srinivasan, Ostrovski, Schaul, Saxton, Munos, 2016

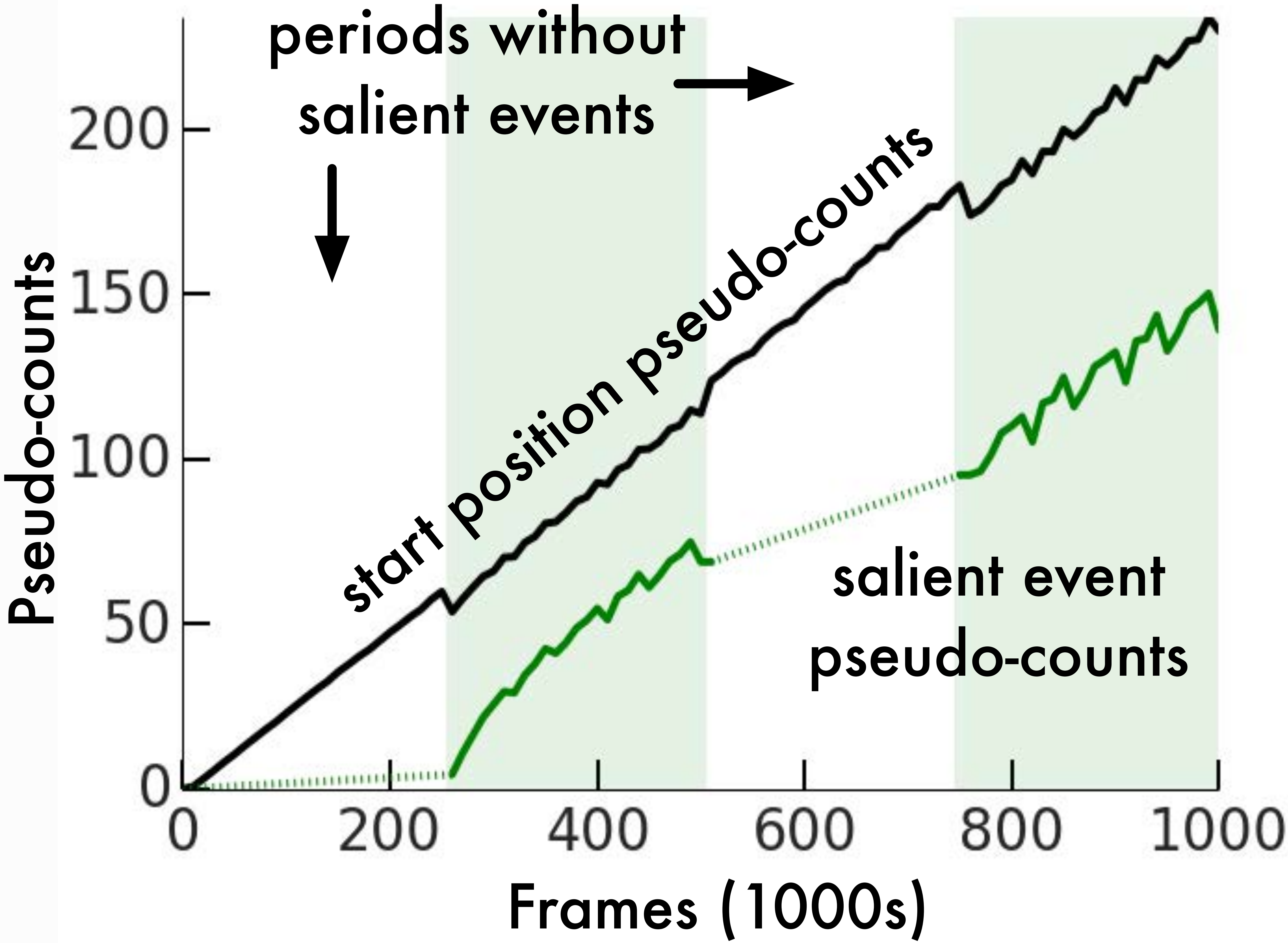

CODING PROBABILITY
(DENSITY)
$\rho_n(x)$
TRAIN
$\rho'_n(x)$
RECODING
PROBABILITY

The "CTS" model

$$\rho_n(x) := \prod_{i=1}^{K} \rho_n^i(x^i \mid x^{<i})$$




ACTIVISION



periods without
salient events
start position pseudo-counts
salient event
pseudo-counts
Pseudo-counts
0
50
100
150
200
0
200
400
600
800
1000
Frames (1000s)

Exploration

Bellman equation:

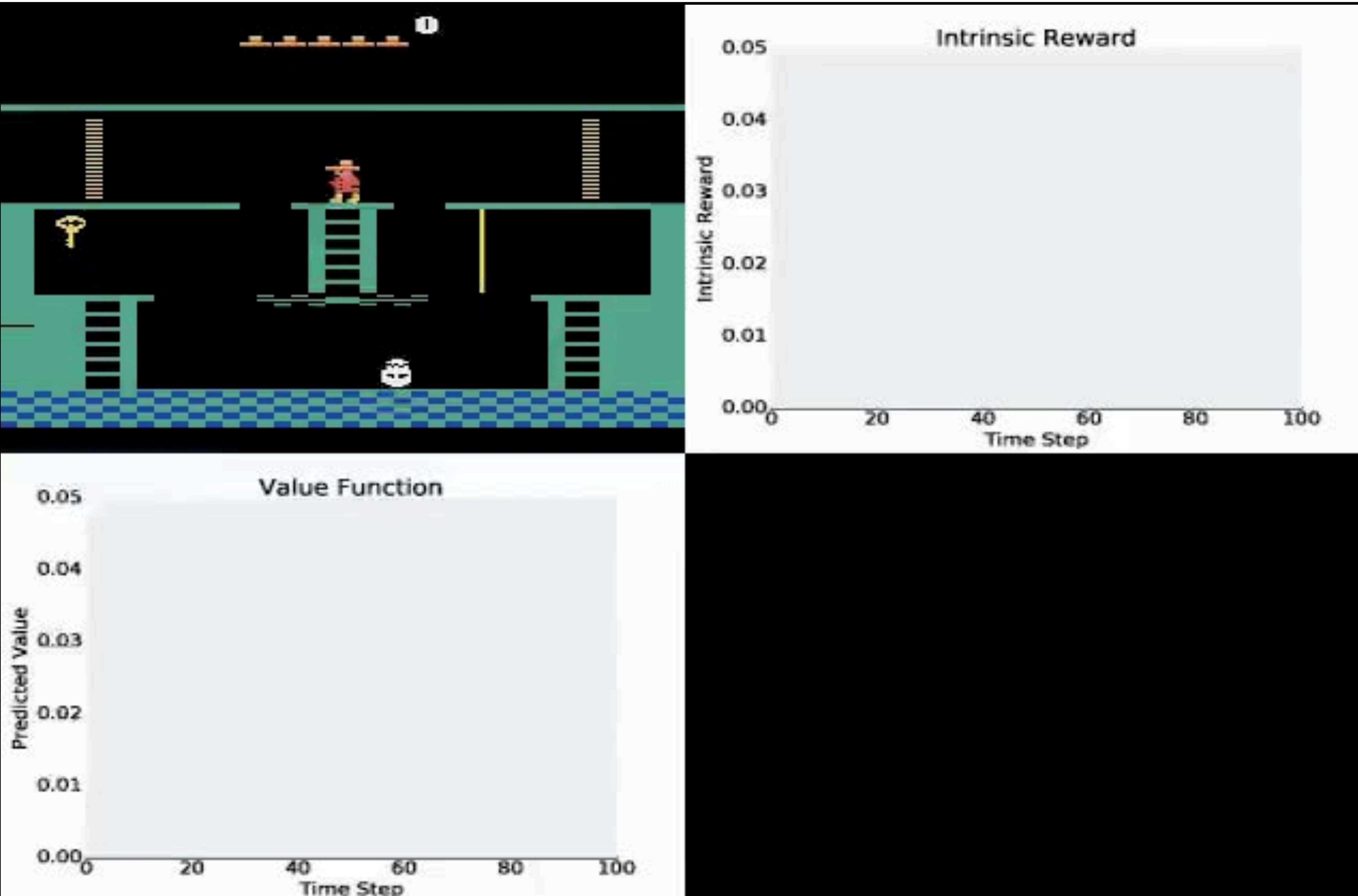
$$Q(x,a) = r(x,a) + \gamma \mathop{\mathbf{E}}_{x' \sim P} \max_{a' \in \mathcal{A}} Q(x',a')$$

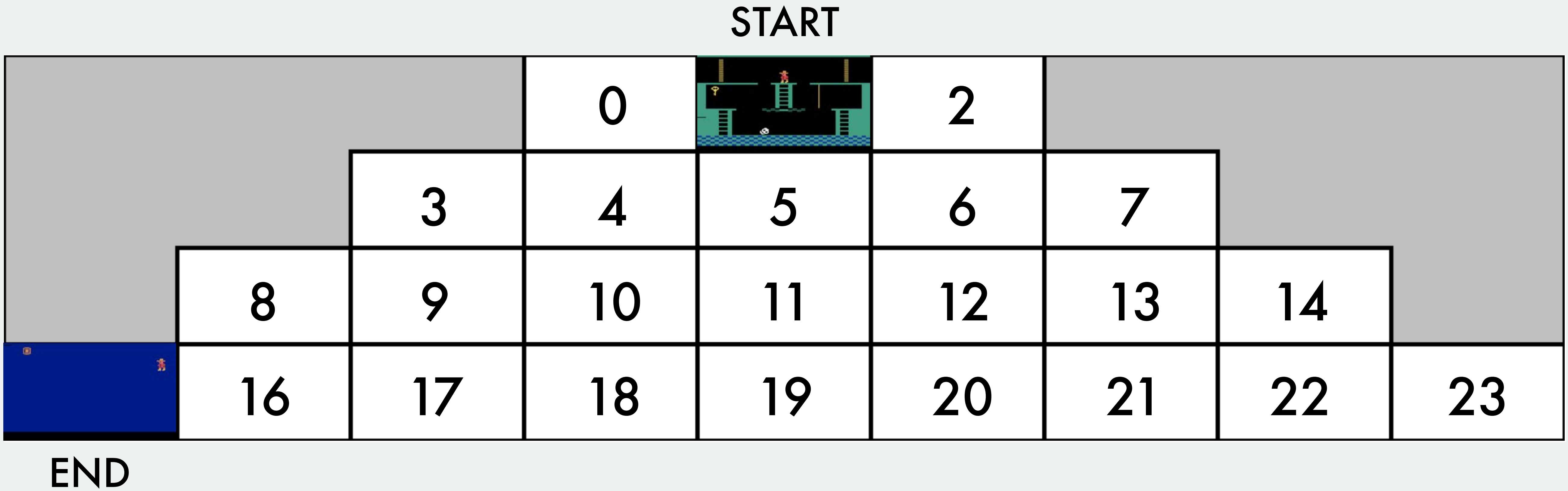
Exploration bonus (Strehl and Littman, 2008)

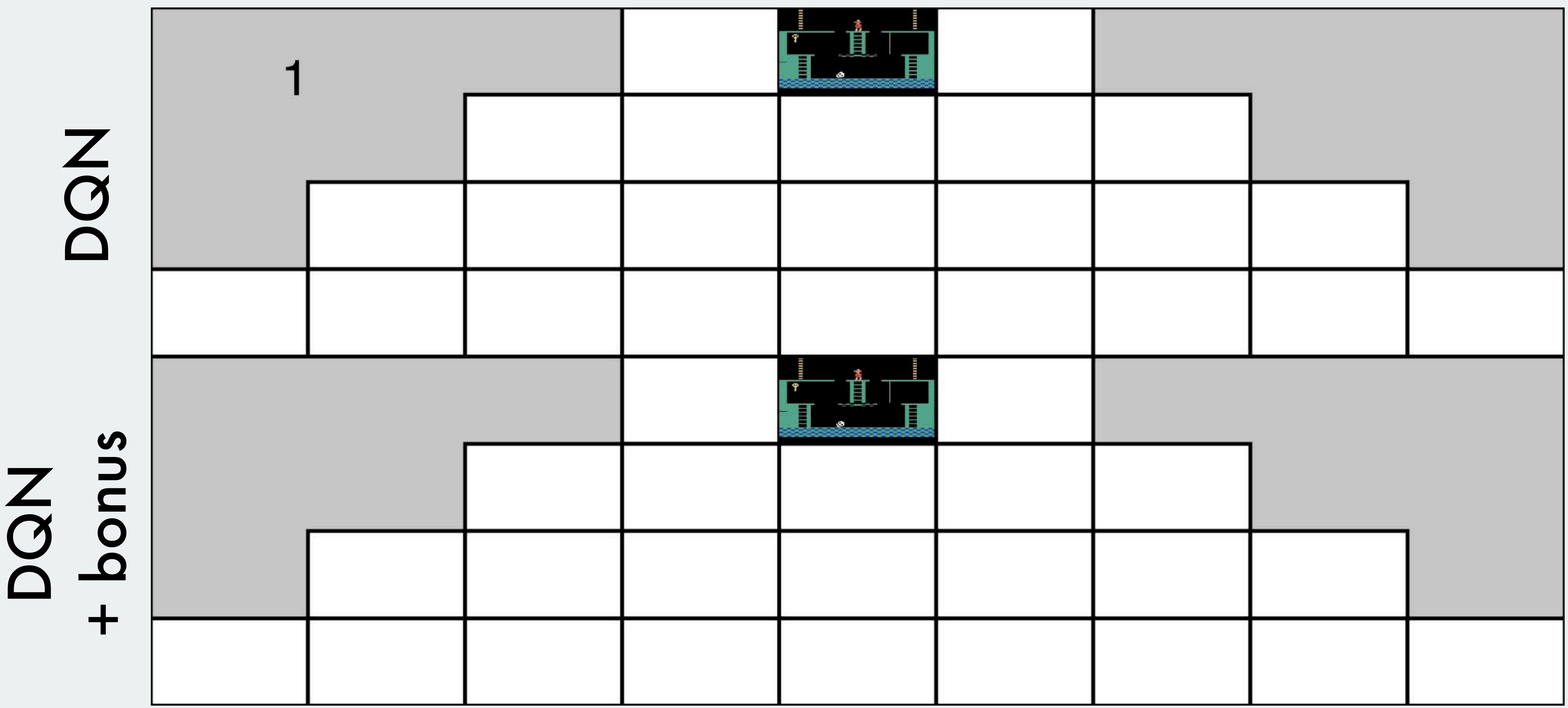
$$Q(x,a) = \hat{r}(x,a) + \gamma \mathop{\mathbf{E}}_{x' \sim \hat{P}} \max_{a' \in \mathcal{A}} Q(x',a') + \frac{\beta}{\sqrt{N(x,a)}}$$

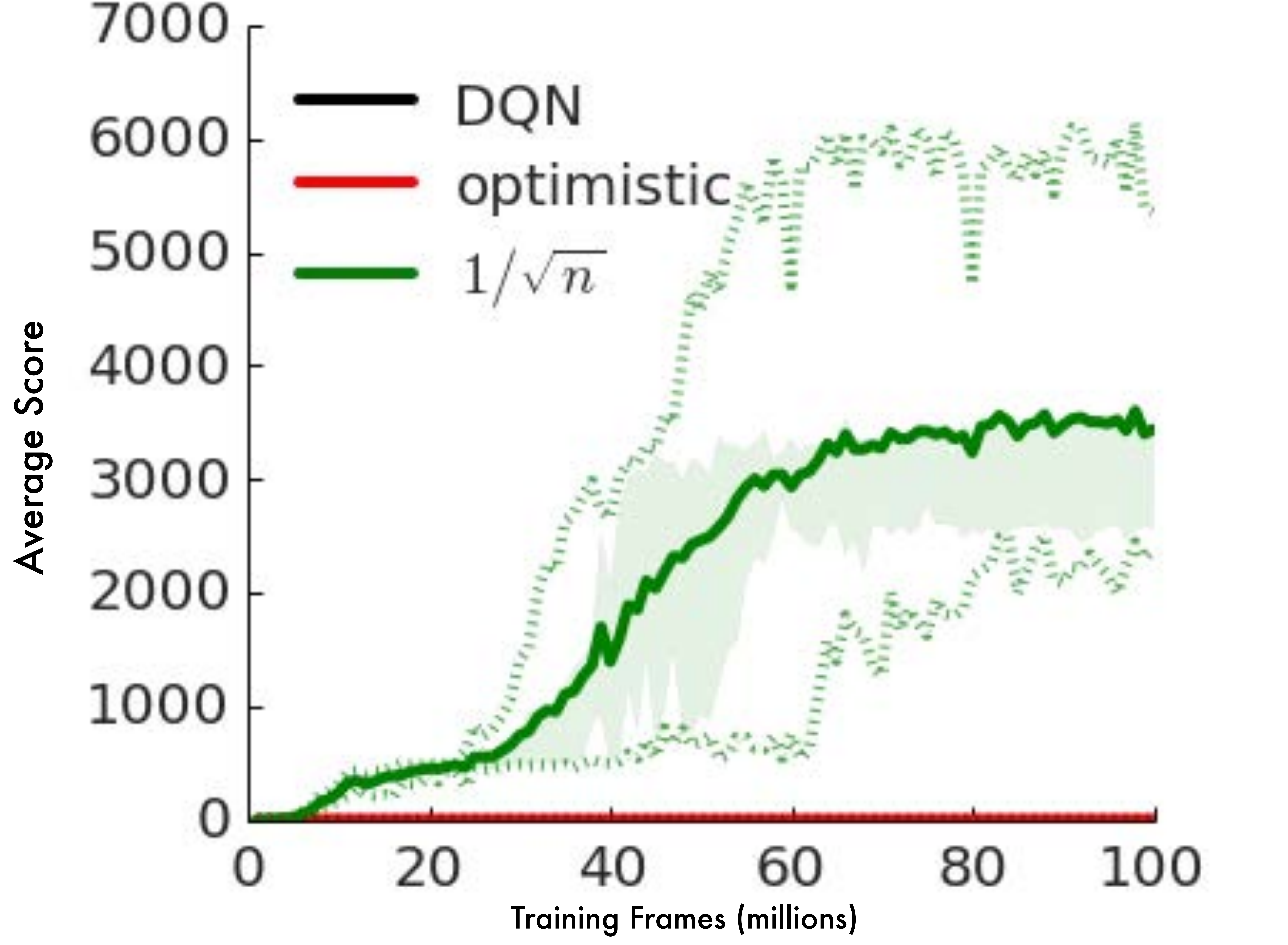
Pseudo-count bonus

$$Q(x,a) = \hat{r}(x,a) + \gamma \mathop{\mathbf{E}}_{x' \sim \hat{P}} \max_{a' \in \mathcal{A}} Q(x',a') + \frac{\beta}{\sqrt{\hat{N}(x)}}$$


Intrinsic Reward
0.05
0.04
0.03
0.02
0.01
0.00
Intrinsic Reward
0
20
40
60
80
100
Time Step
Value Function
0.05
0.04
0.03
0.02
0.01
0.00
Predicted Value
0
20
40
60
80
100
Time Step


START
0
2
3
4
5
6
7
8
9
10
11
12
13
14
16
17
18
19
20
21
22
23
END


1
DQN
DQN + bonus


DQN
optimistic
$1/\sqrt{n}$
Average Score
Training Frames (millions)
0
1000
2000
3000
4000
5000
6000
7000
0
20
40
60
80
100

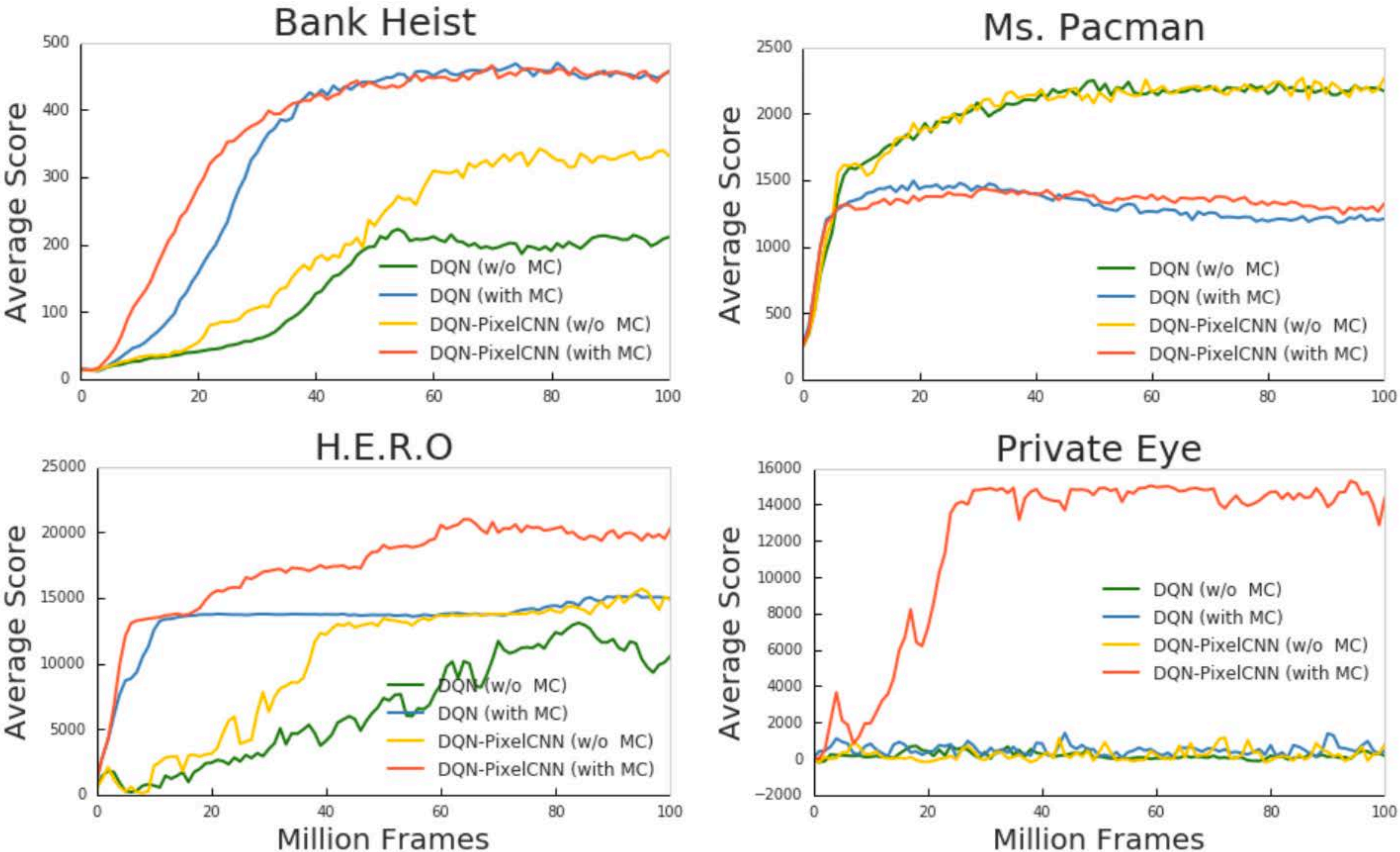
Credit Assignment Issues in Exploration

- Another important key factor: fast credit assignment

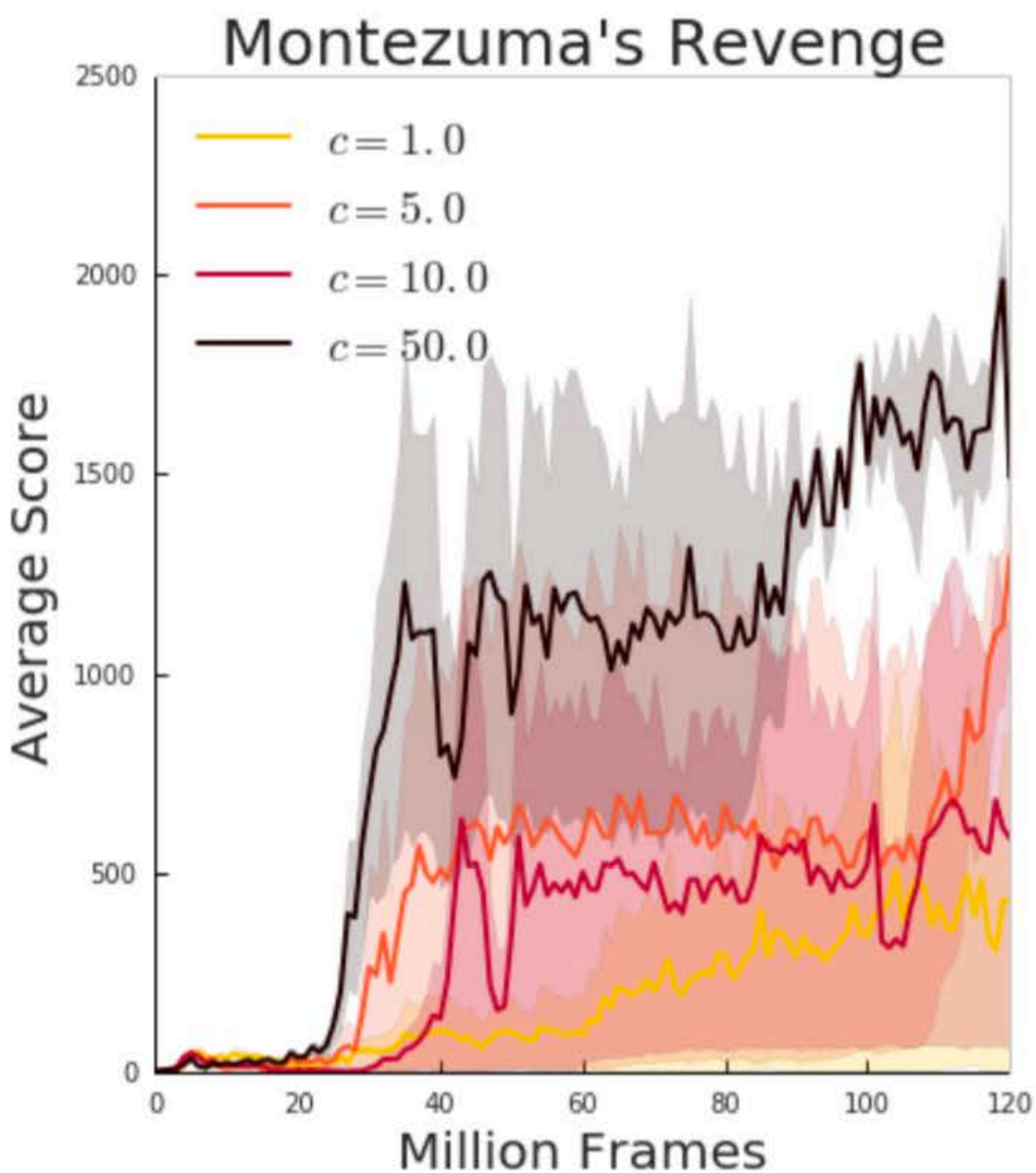
$$\tilde{Q}(x_t, a_t) = q \times \left[r_+(x_t, a_t) + \gamma \max_{a' \in \mathcal{A}} Q(x_{t+1}, a') \right] + (1 - q) \times \left[\sum_{i=0}^{\infty} \gamma^i r_+(x_{t+i}, a_{t+i}) \right]$$

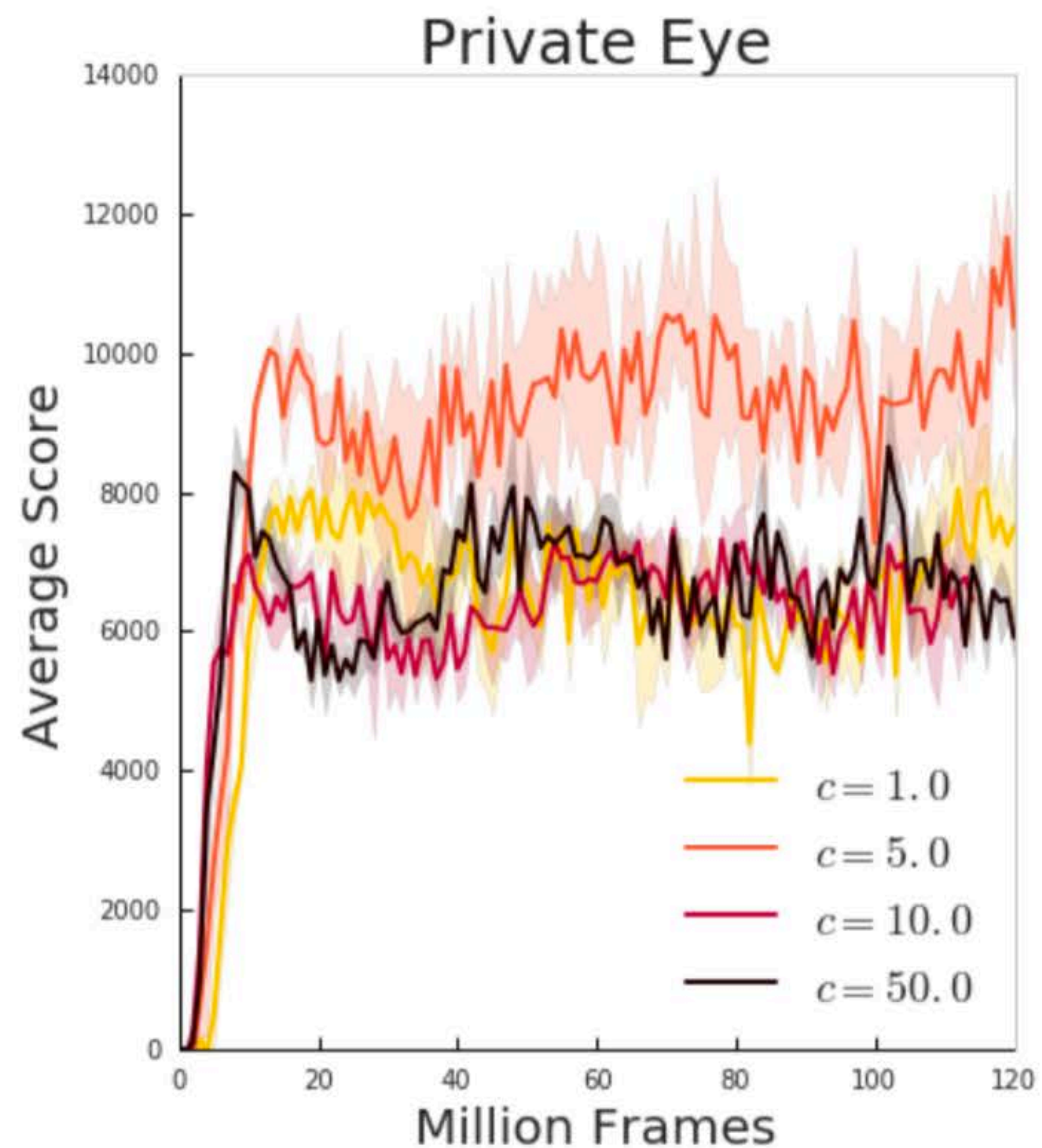
Mixed Monte-Carlo (MMC) update

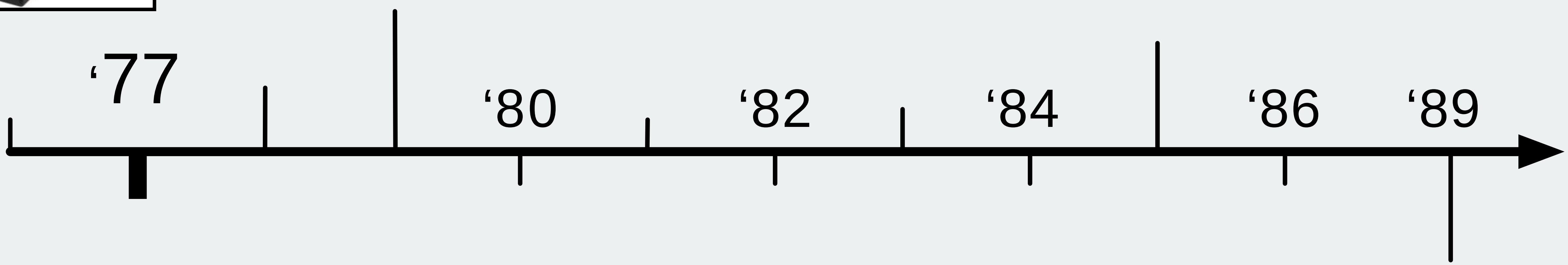
EFFECT OF MIXED MONTE CARLO UPDATE



REMOVING EXTRINSIC REWARDS






‘77
‘80
‘82
‘84
‘86
‘89

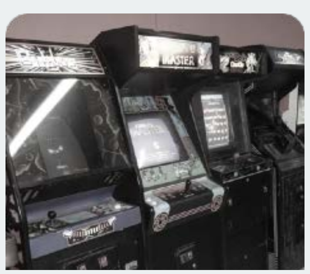
■ Super-human agents
■ Scoring exploit (grey)
□ Sub-human agents

■ Breakout

'77

■ Pong
■ Video Pinball

■ Space Invaders



■ Asteroids
■ Bowling

'80

■ Battlezone
■ Berzerk
■ Boxing
■ Carnival
■ Centipede
■ Crazy Climber
■ Fishing Derby
■ Phoenix
□ Skiing

■ Freeway
■ Ice Hockey
■ Tennis
□ Ms. Pac-Man
□ Venture

'82

■ Amidar
■ Atlantis
■ Chopper Cmd.
■ Demon Attack
■ Gopher
■ Name this Game
■ Pooyan
■ River Raid
■ Star Gunner
■ Time Pilot
■ Yar's Revenge
■ Zaxxon
■ Tutankham (grey)
□ Alien
□ Gravitar
□ Journey Escape
□ Kangaroo
□ Pitfall!

■ Assault
■ Asterix
■ Beam Rider
■ Elevator Action
■ Enduro
■ James Bond
■ Seaquest
■ Up N' Down
■ Bank Heist (grey)
■ Krull (grey)
□ Frostbite
□ Private Eye
□ Q*Bert

'84

■ Kung-Fu Master (grey)
□ H.E.R.O.
□ Montezuma's Revenge

■ Road Runner (grey)

'86

□ Solaris

'89

■ Double Dunk

Deep Reinforcement Learning and the Atari 2600

Marc G. Bellemare
Google Brain