DQNViz: A Visual Analytics Approach to Understand Deep Q-Networks

Junpeng Wang[1], **Liang Gou**[2], Han-Wei Shen[1], Hao Yang[2]

1. The Ohio State University
2. Visa Research

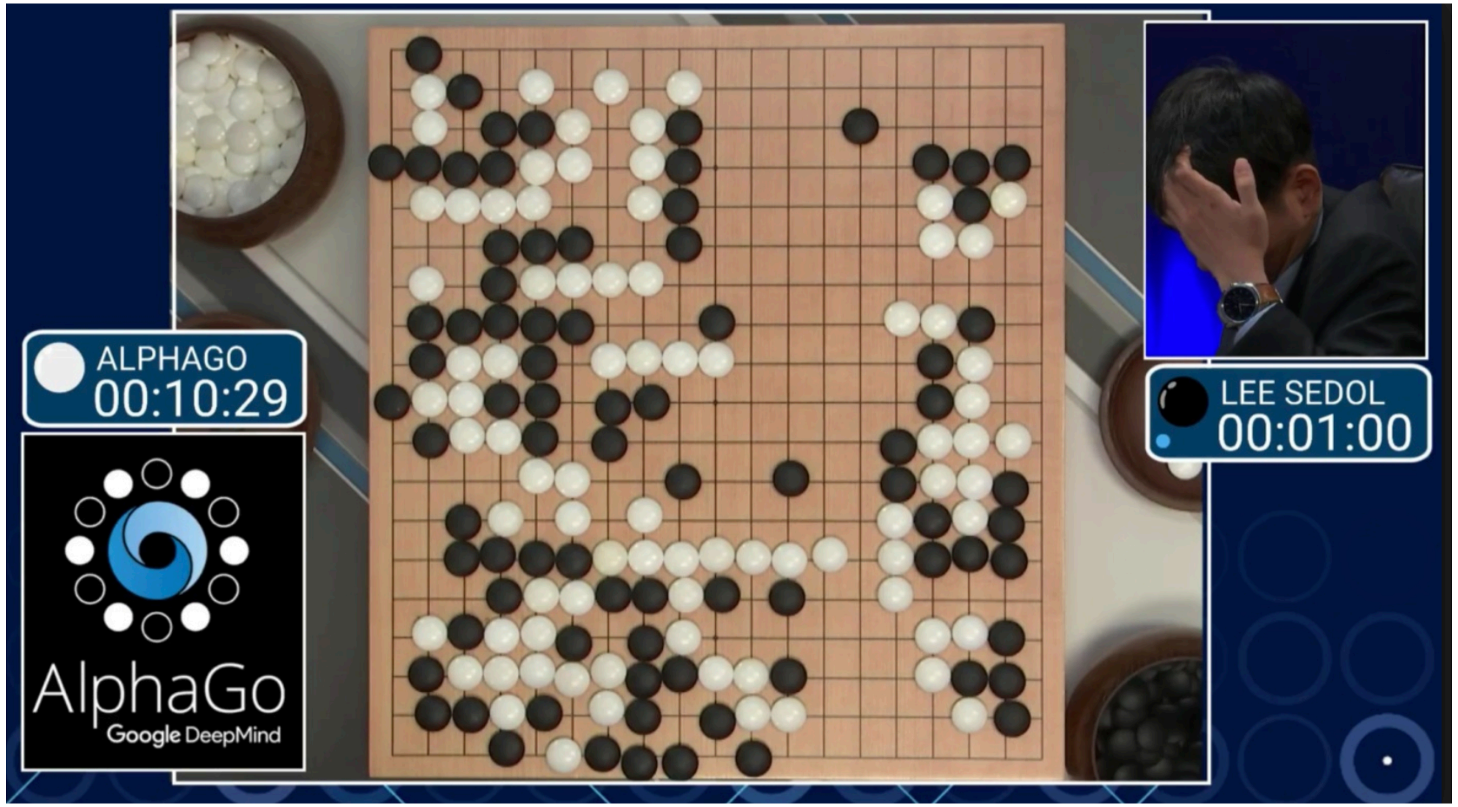
VIS 2018

THE OHIO STATE UNIVERSITY

VISA Research

Introduction

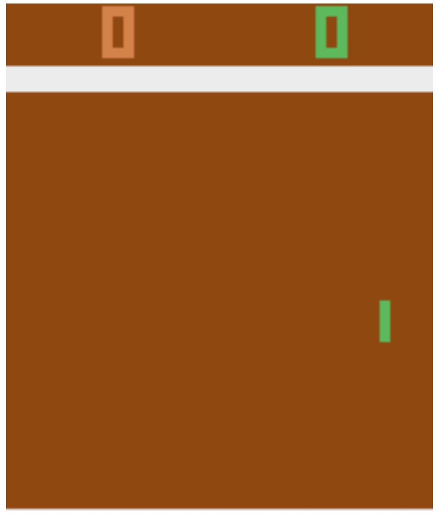
- Deep Reinforcement Learning + AlphaGo



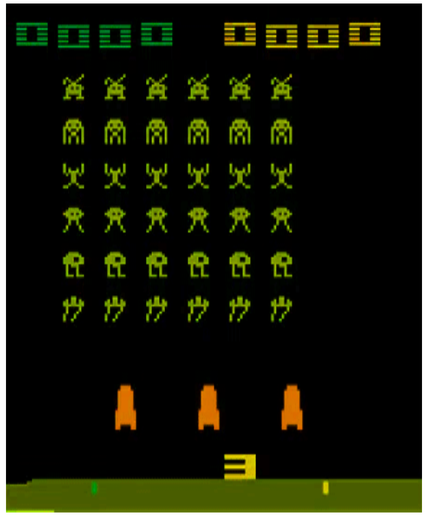
Introduction

- Deep Reinforcement Learning + AlphaGo
- Deep Q Networks + Atari Games

Breakout



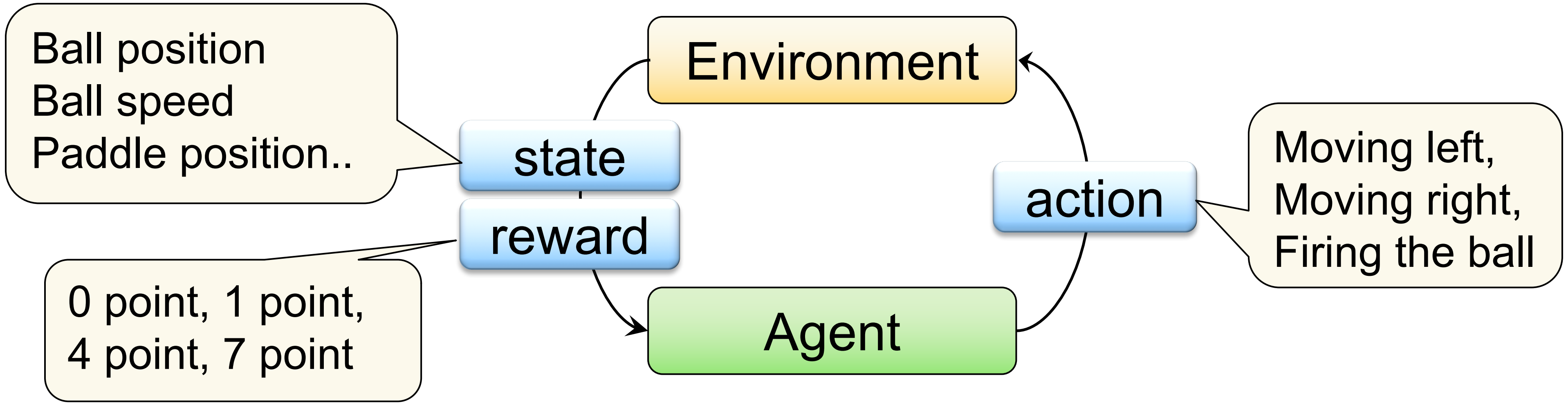
Pong



Space-Invader

Introduction

- Deep Reinforcement Learning + AlphaGo
- Deep Q Networks + Atari Games



$$a_1, s_1, r_1, a_2, s_2, r_2, \dots a_n, s_n, r_n$$

Background: Q-Learning with Bellman Equation

Data: $s_0, a_0, r_1, s_1, a_1, r_2, \ldots, r_n, s_n$

Total reward: $R = r_1 + r_2 + \ldots + r_n$

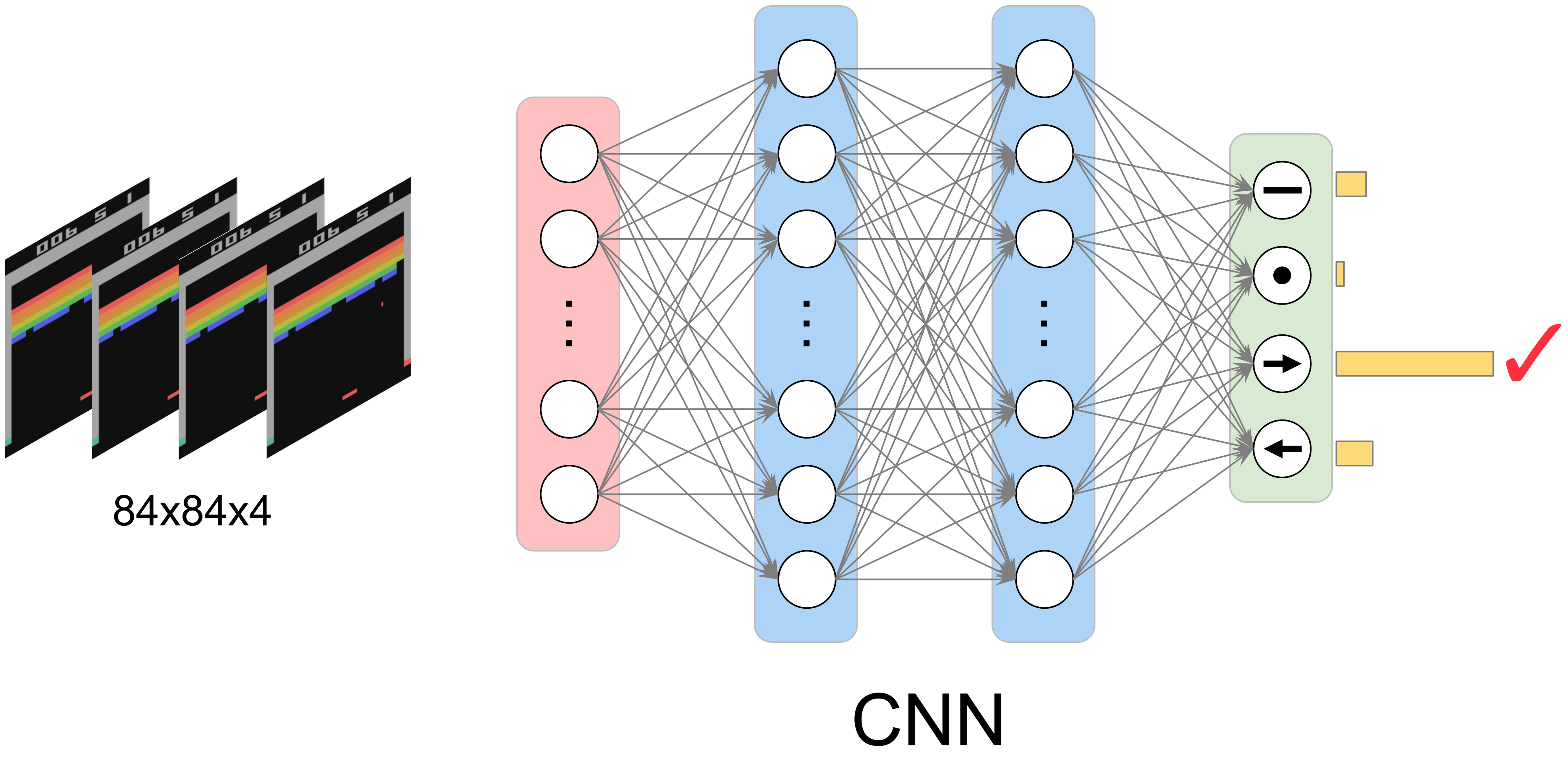
Future reward: $R_t = r_t + r_{t+1} + \ldots + r_n$

Discounted Future reward: $R_t = r_t + \gamma r_{t+1} + \gamma^2 r_{t+2} \ldots + \gamma^{n-t} r_n$

Bellman equation: $Q(s,a) = r + \gamma \max_{a'} Q(s', a')$

Why deep neural network? s is too complex: $256^{84 \times 84 \times 4}$

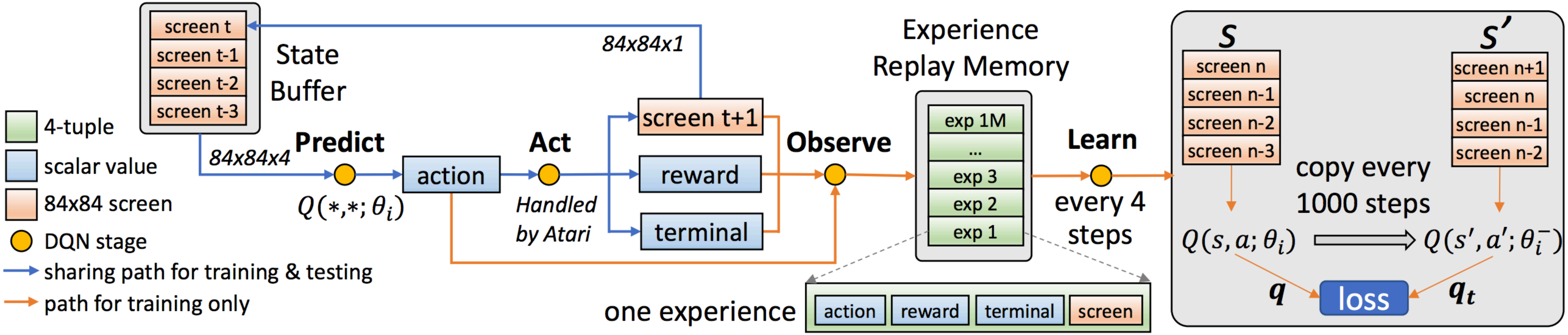
Background: Deep Q(uality)-Network (DQN)



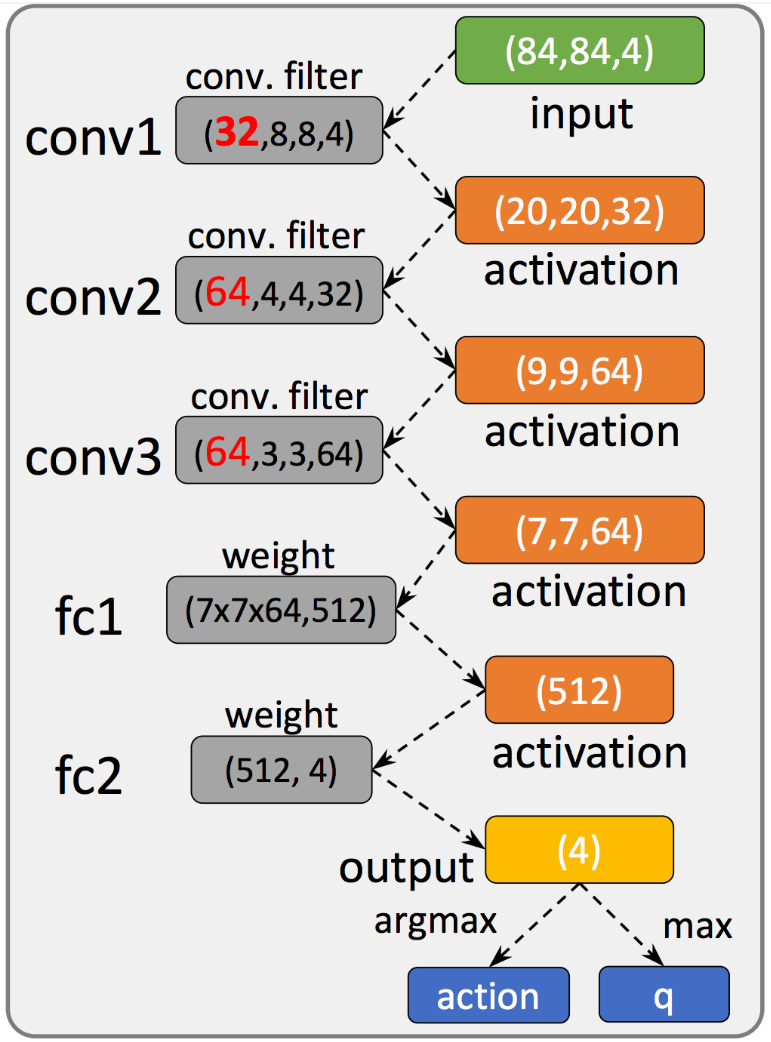
Objective:

maximize the total game reward

Background: Deep Q(uality)-Network (DQN)



Background: Deep Q(uality)-Network (DQN)



$$L_i(\theta_i) = \mathbb{E}_{(s,a,r,s')\sim ER}\left[\left(r + \gamma \max_{a'} Q(s', a'; \theta_i^-) - Q(s, a; \theta_i)\right)^2\right]$$

a random sample from the experience replay memory: $(s, a, r, s') \sim ER$

$s = \{sn_{t-3}, sn_{t-2}, sn_{t-1}, sn_t\} \Rightarrow DQN(\theta_i) \Rightarrow$

$output_t = \{1.91, 1.72, \mathbf{1.95}, 1.83\}$

$q = \max(output_t) = 1.95$

$a = argmax(output_t) = 2 \ (right)$

$r = 1 \qquad \gamma = 0.99$

$s' = \{sn_{t-2}, sn_{t-1}, sn_t, sn_{t+1}\} \Rightarrow DQN(\theta_i^-) \Rightarrow$

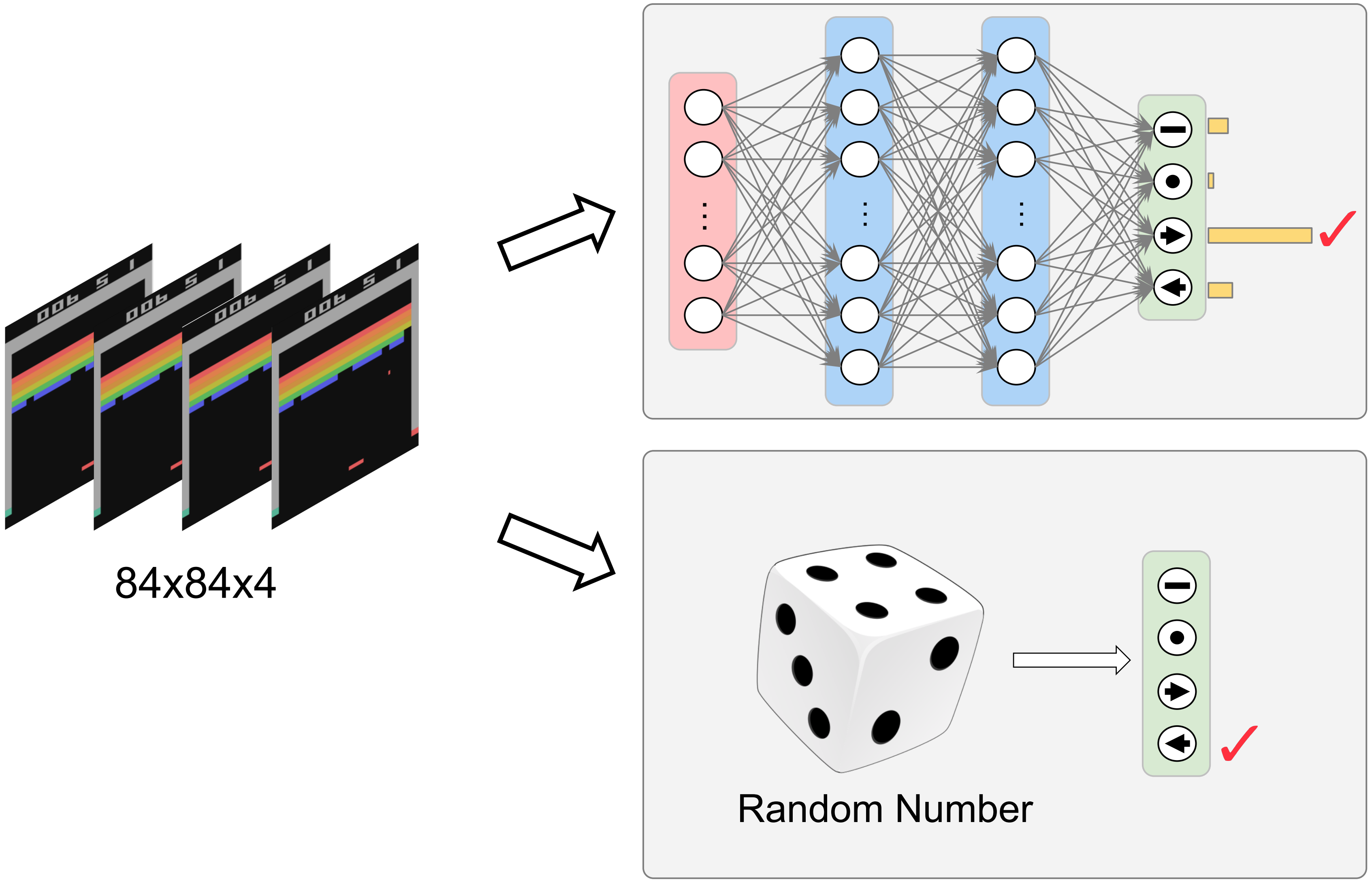
$output_{t+1} = \{1.35, 1.39, 1.32, \mathbf{1.43}\}$

$q_t = r + \gamma * \max(output_{t+1})$

$= 1 + 0.99 * 1.43 = 2.4175$

$$loss = (q - q_t)^2 = q_{diff}^2 = (1.95 - 2.4175)^2 = 0.4675^2 = 0.2186$$

Background: Exploration and Exploitation dilemma



moving right 95%

moving left 5%

VIS 2018

Challenges

- Long-time blind training process (understand the model)
  - What strategies are really learned?
  - When are those strategy learned?
  - Which part of the neural network learned those strategies?

- Proper choice of different hyper-parameters (improve the model)
  - E.g., the random rate for the tradeoff between exploration and exploitation

Contribution: DQNViz

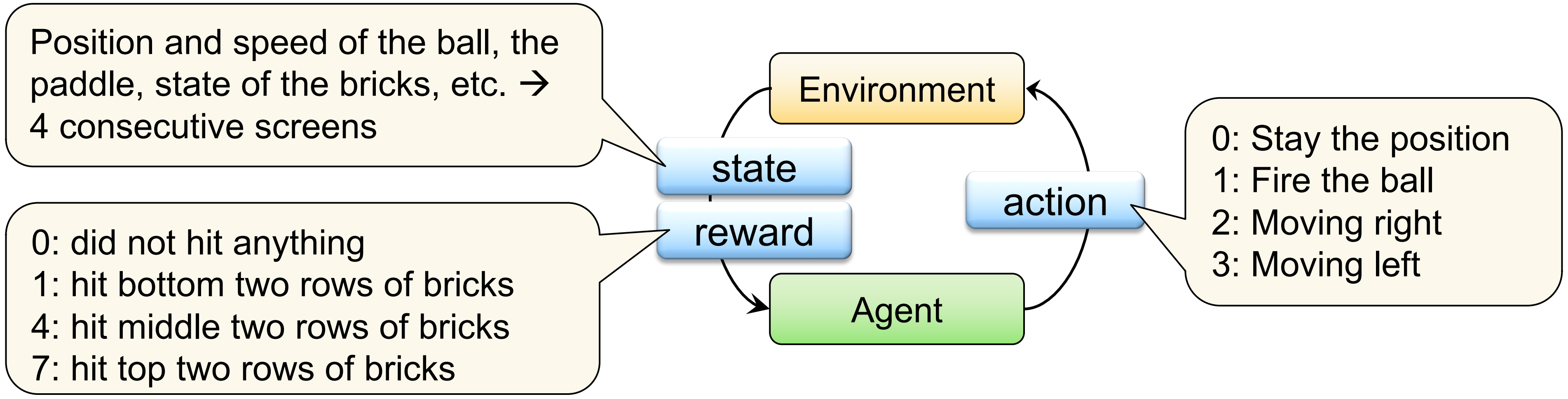
- Visual Analytics System (DQNViz):
  - Effective visual summary
  - Efficient (movement/reward) pattern mining

- Improve DQN training by optimizing the random actions
  - Pattern detection algorithm based on DQNViz

Part I: effective visual summary and efficient pattern mining

DQNViz: The Breakout Game

7 points
4 points
1 point



$$\underline{a_1}, \underline{s_1}, \underline{r_1}, \underline{a_2}, \underline{s_2}, \underline{r_2}, \ldots \underline{a_n}, \underline{s_n}, \underline{r_n}$$

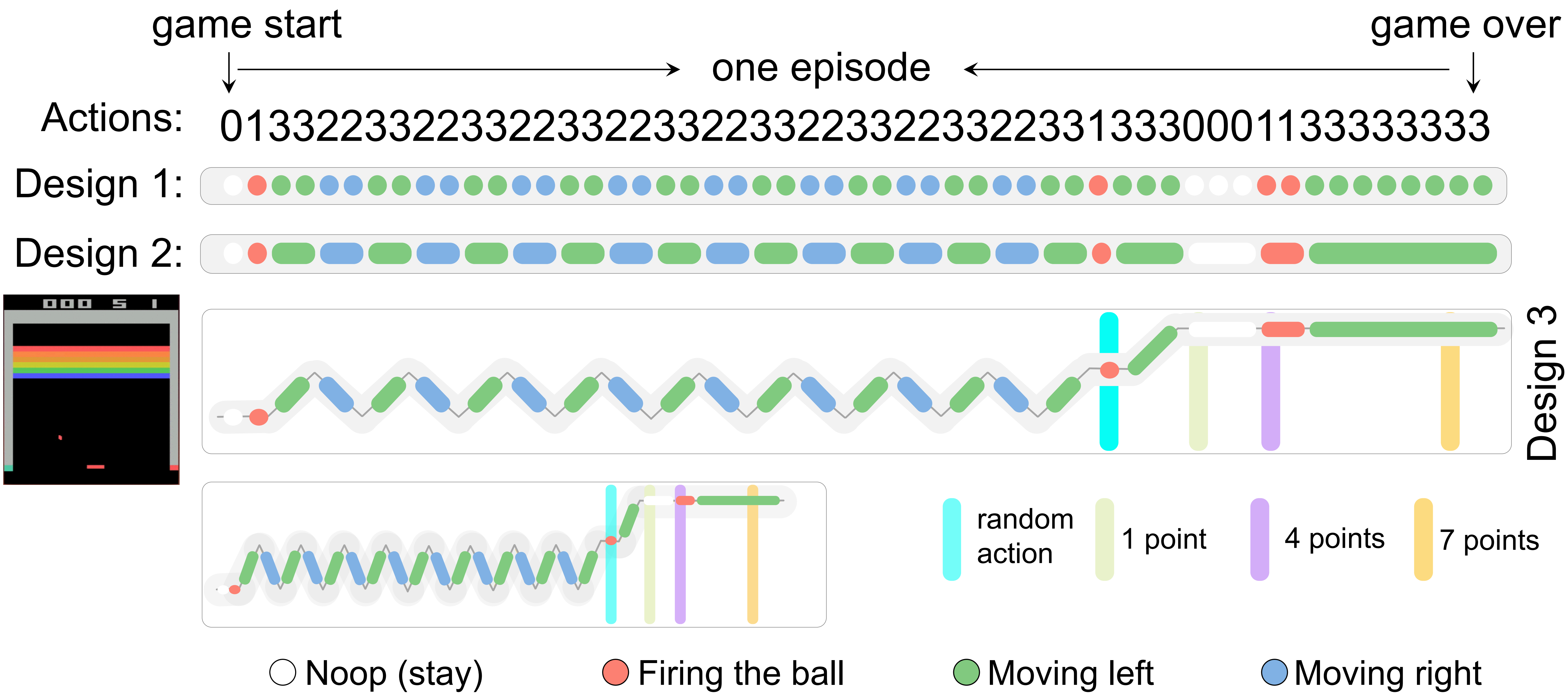
Example Data:

Action sequence: 000001222233331231231100

State sequence: (84x84x3)(84x84x3)…(84x84x3)

Reward sequence: 1000000040000004000000700

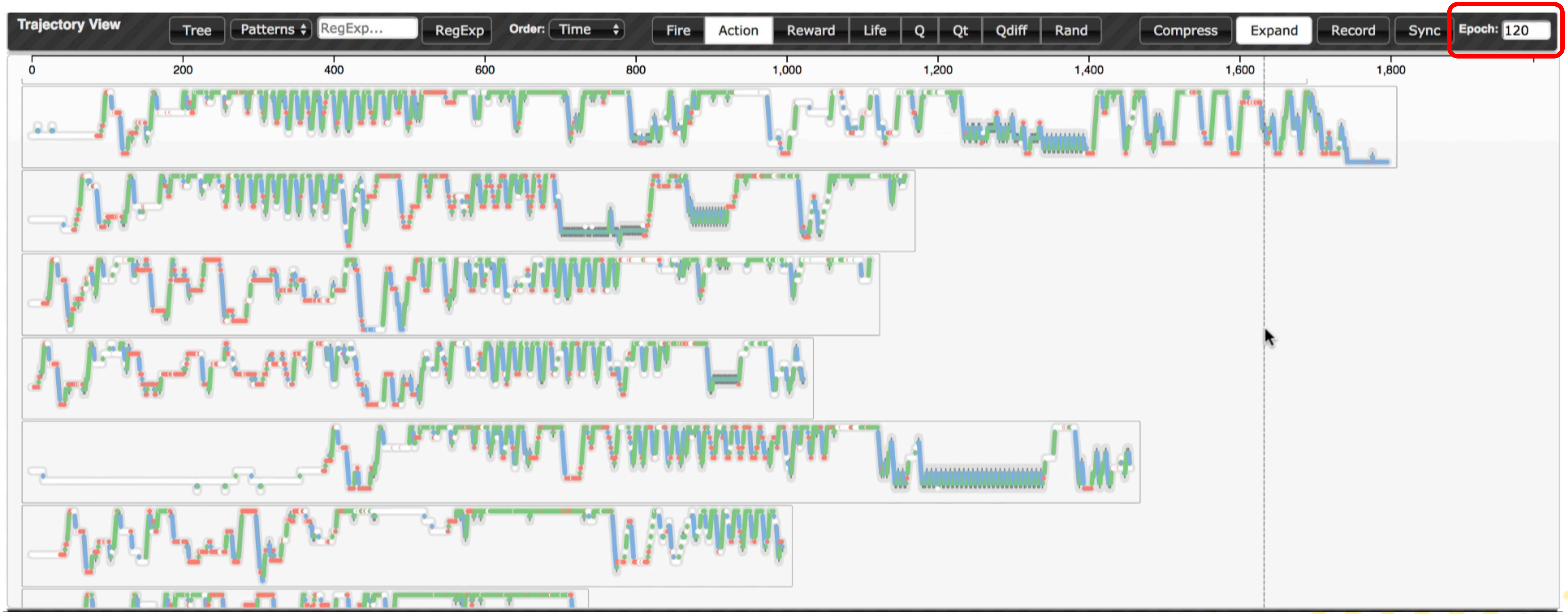
VIS 2018


game start
one episode
game over
Actions:
0133223322332233223322332233223313330001133333333
Design 1:
Design 2:
Design 3
random action
1 point
4 points
7 points
Noop (stay)
Firing the ball
Moving left
Moving right

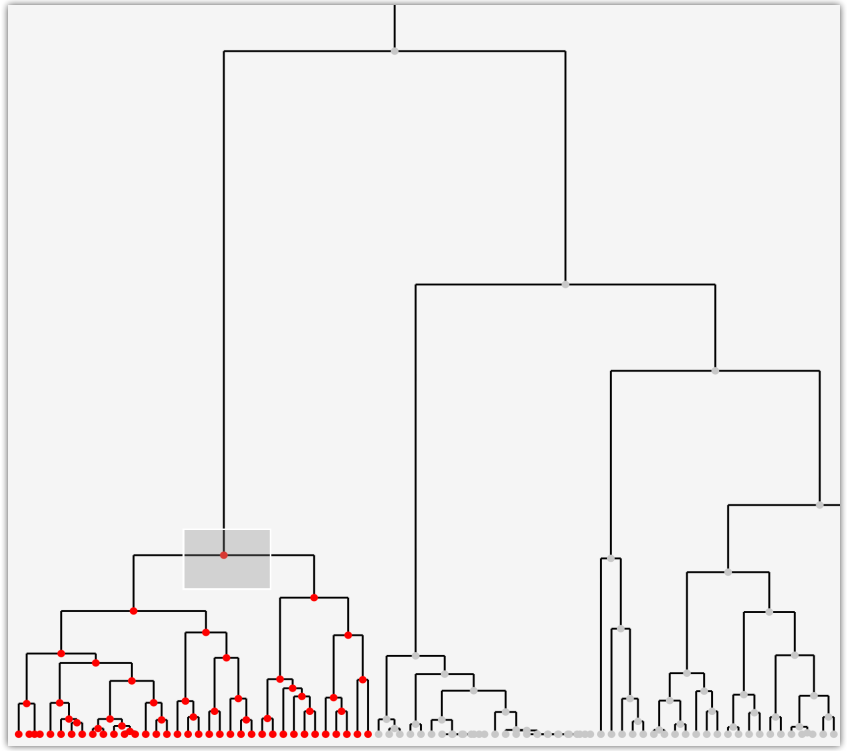
VIS 2018

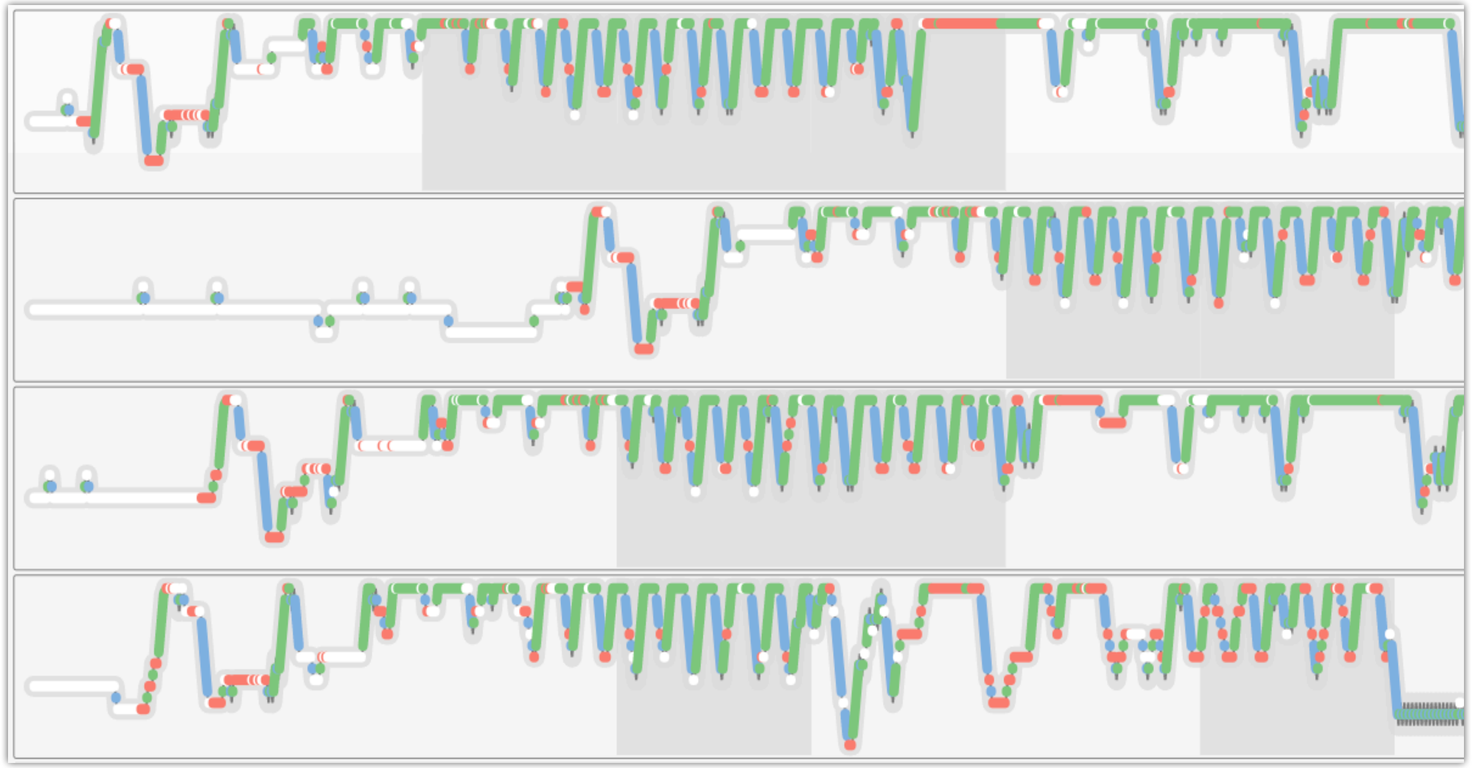
DQNViz: The Episode/Trajectory View

25000 game steps (actions)

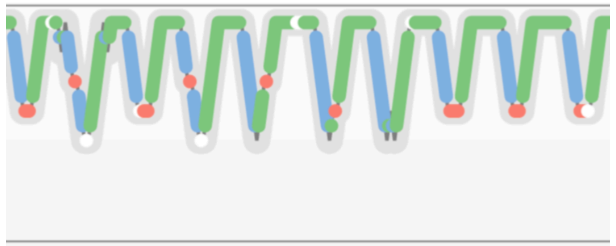
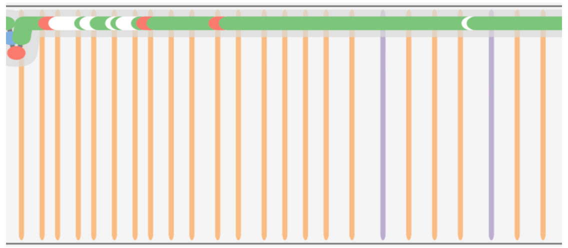
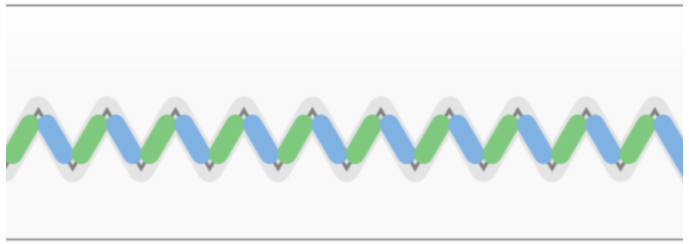


DQNViz: Pattern Mining





DQNViz: Pattern Mining

Action Sequence				
Video Replay	Digging Pattern	Bouncing Pattern	Hesitating Pattern	Repeating Pattern
Regular Expression	10+10+40+40+70+7 reward pattern	(70+){5, } reward pattern	(20*30*){5, } action pattern	3{30,} action pattern

Statistics View
Epoch: 120
Epoch View
Episode count: 20
Step: [127 , 1989]
Reward: [3 , 404]
average_reward
(120, 262)
nr_games
(120, 19)
average_q
(120, 10.4)
loss
(120, 0.000773)
bouncing
(120, 26)
action_distribution
step
Left
Noop
Fire
Right
reward_distribution
count
7 pts
1 pt
4 pts
action
reward
Trajectory View
Tree
Patterns
RegExp...
RegExp
Order:
Step
Fire
Action
Reward
Life
Q
Qt
Qdiff
Rand
Compress
bouncing
(120, 26)

VIS 2018

Guided Back-propagation

$$h^{l+1} = \max\{0, h^l\}$$

Forward pass

$$h^l \begin{bmatrix} 1 & -1 & 5 \\ 2 & -5 & -7 \\ -3 & 2 & 4 \end{bmatrix} \rightarrow \begin{bmatrix} 1 & 0 & 5 \\ 2 & 0 & 0 \\ 0 & 2 & 4 \end{bmatrix} h^{l+1}$$

$$\frac{\partial L}{\partial h^l} = [\![h^l > 0]\!] \frac{\partial L}{\partial h^{l+1}}$$

Backward pass: backpropagation

$$\begin{bmatrix} -2 & 0 & -1 \\ 6 & 0 & 0 \\ 0 & -1 & 3 \end{bmatrix} \leftarrow \begin{bmatrix} -2 & 3 & -1 \\ 6 & -3 & 1 \\ 2 & -1 & 3 \end{bmatrix} \frac{\partial L}{\partial h^{l+1}}$$

$$\frac{\partial L}{\partial h^l} = [\![h^{l+1} > 0]\!] \frac{\partial L}{\partial h^{l+1}}$$

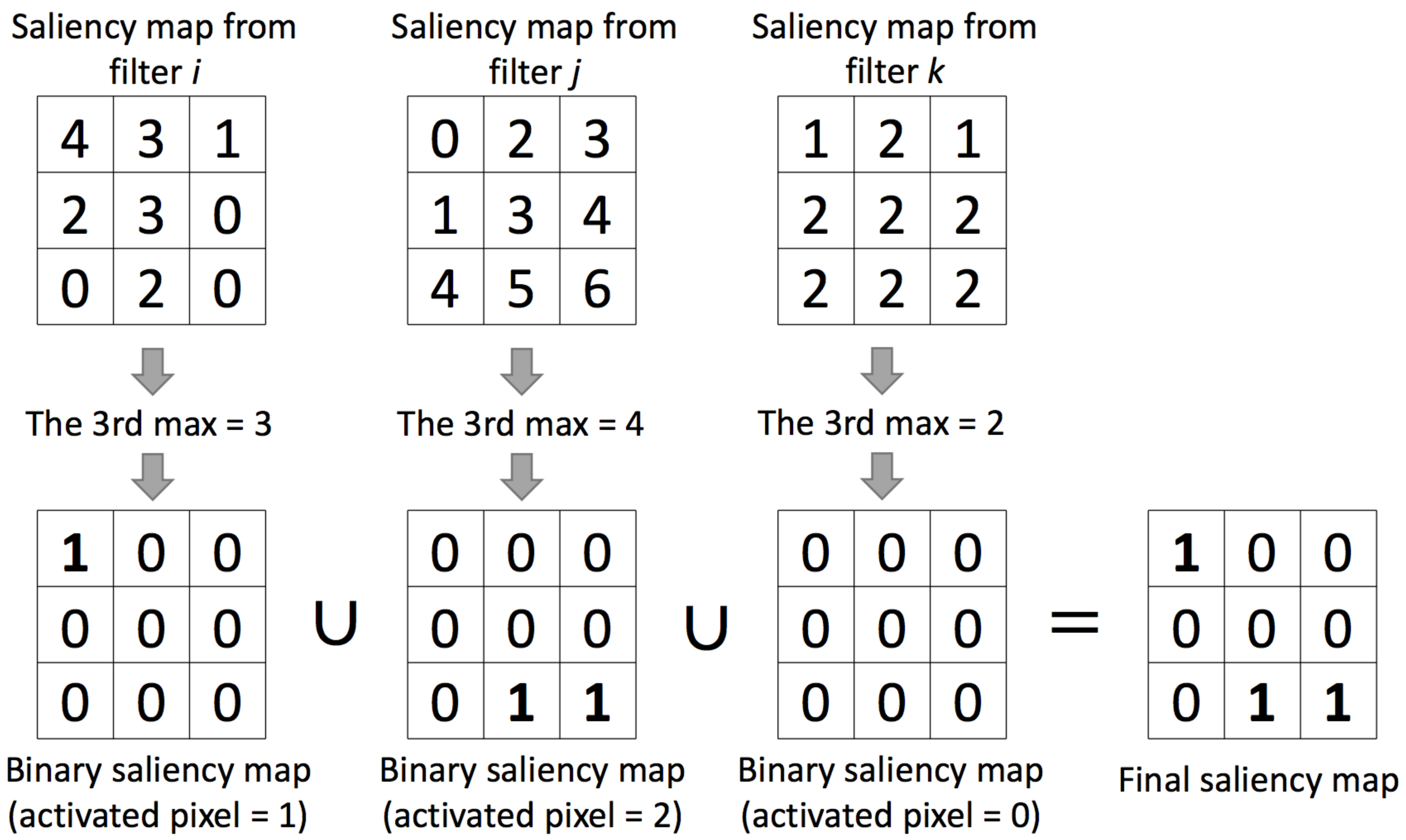
Backward pass: "deconvnet"

$$\begin{bmatrix} 0 & 3 & 0 \\ 6 & 0 & 1 \\ 2 & 0 & 3 \end{bmatrix} \leftarrow \begin{bmatrix} -2 & 3 & -1 \\ 6 & -3 & 1 \\ 2 & -1 & 3 \end{bmatrix} \frac{\partial L}{\partial h^{l+1}}$$

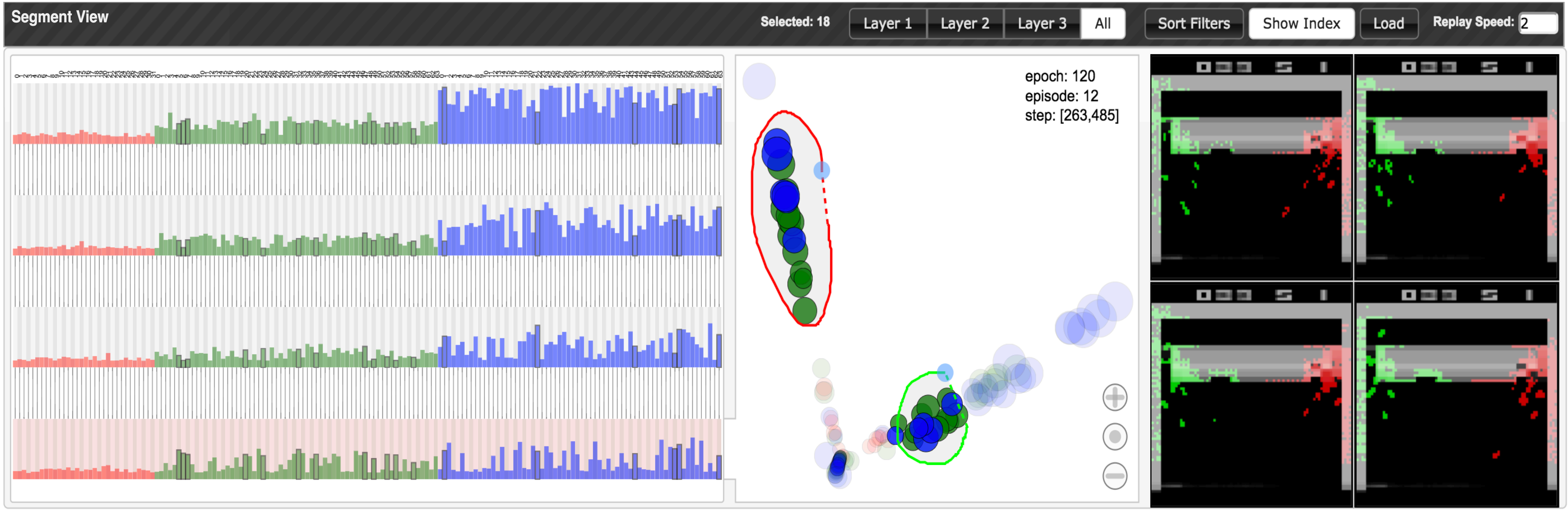
$$\frac{\partial L}{\partial h^l} = [\![(h^l > 0) \& \& (h^{l+1} > 0)]\!] \frac{\partial L}{\partial h^{l+1}}$$

Backward pass: *guided backpropagation*

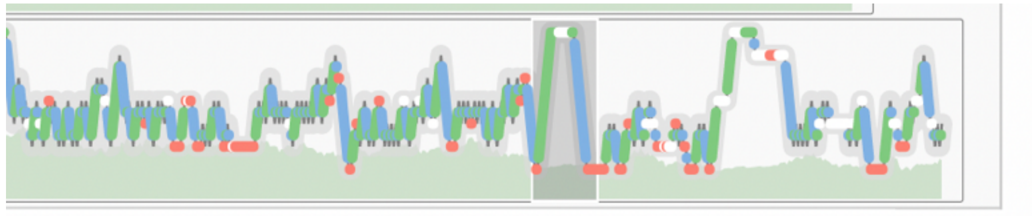
$$\begin{bmatrix} 0 & 0 & 0 \\ 6 & 0 & 0 \\ 0 & 0 & 3 \end{bmatrix} \leftarrow \begin{bmatrix} -2 & 3 & -1 \\ 6 & -3 & 1 \\ 2 & -1 & 3 \end{bmatrix} \frac{\partial L}{\partial h^{l+1}}$$


Saliency map from filter *i*
4 3 1
2 3 0
0 2 0
The 3rd max = 3
1 0 0
0 0 0
0 0 0
Binary saliency map (activated pixel = 1)
∪
Saliency map from filter *j*
0 2 3
1 3 4
4 5 6
The 3rd max = 4
0 0 0
0 0 0
0 1 1
Binary saliency map (activated pixel = 2)
∪
Saliency map from filter *k*
1 2 1
2 2 2
2 2 2
The 3rd max = 2
0 0 0
0 0 0
0 0 0
Binary saliency map (activated pixel = 0)
=
1 0 0
0 0 0
0 1 1
Final saliency map

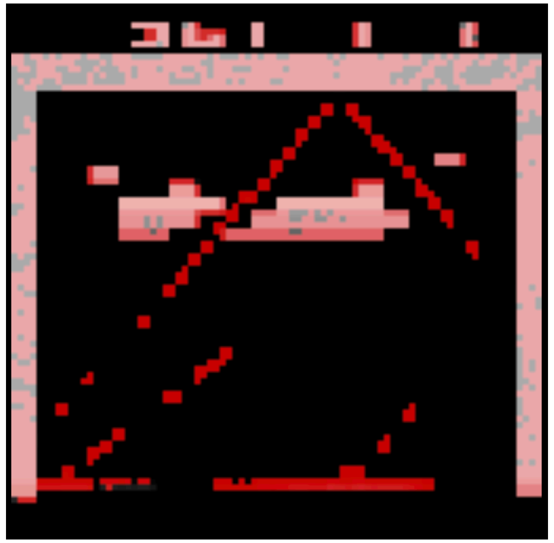
Guided Back-propagation



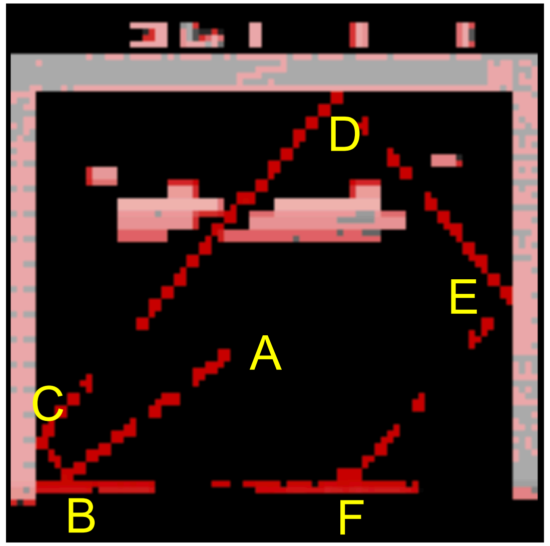
Guided Back-propagation



Epoch 0



Epoch 10

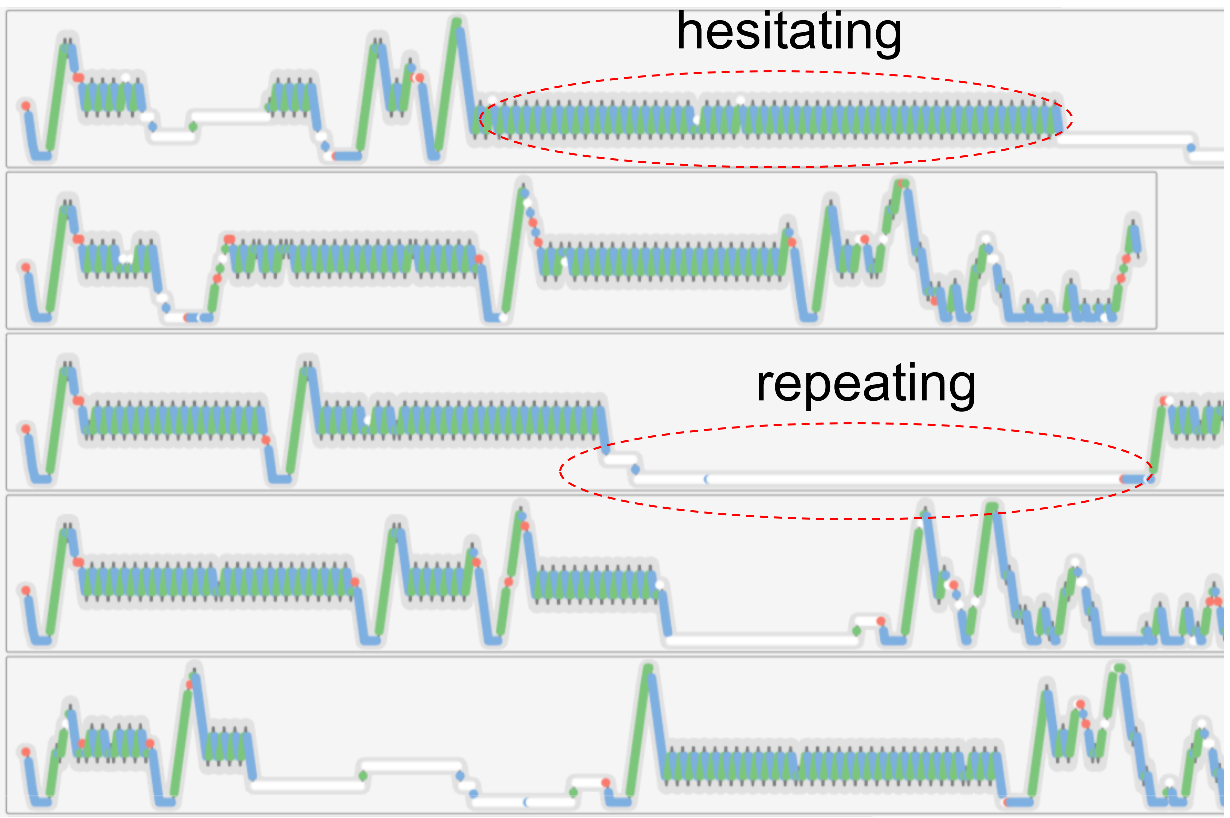


Epoch 120

VIS 2018

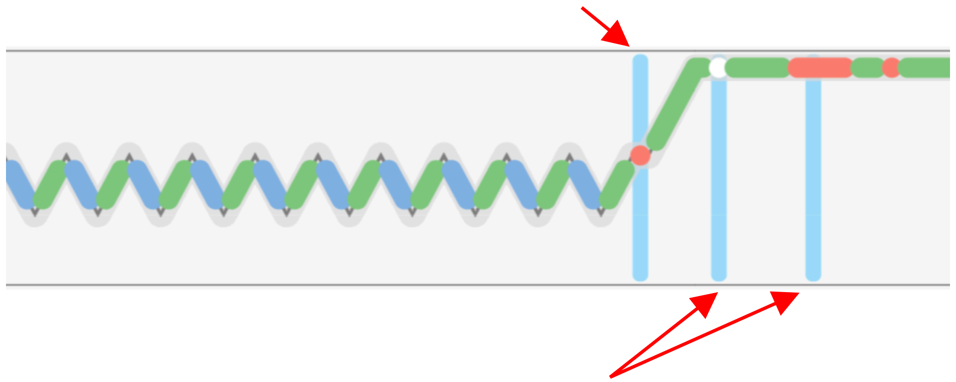
Part II: Optimize random actions to improve the DQN model training.

Improve Random Actions



An unsuccessful training due to the hesitating and repeating pattern

Random action terminating the hesitating pattern



unnecessary random action, which may break an intentional move

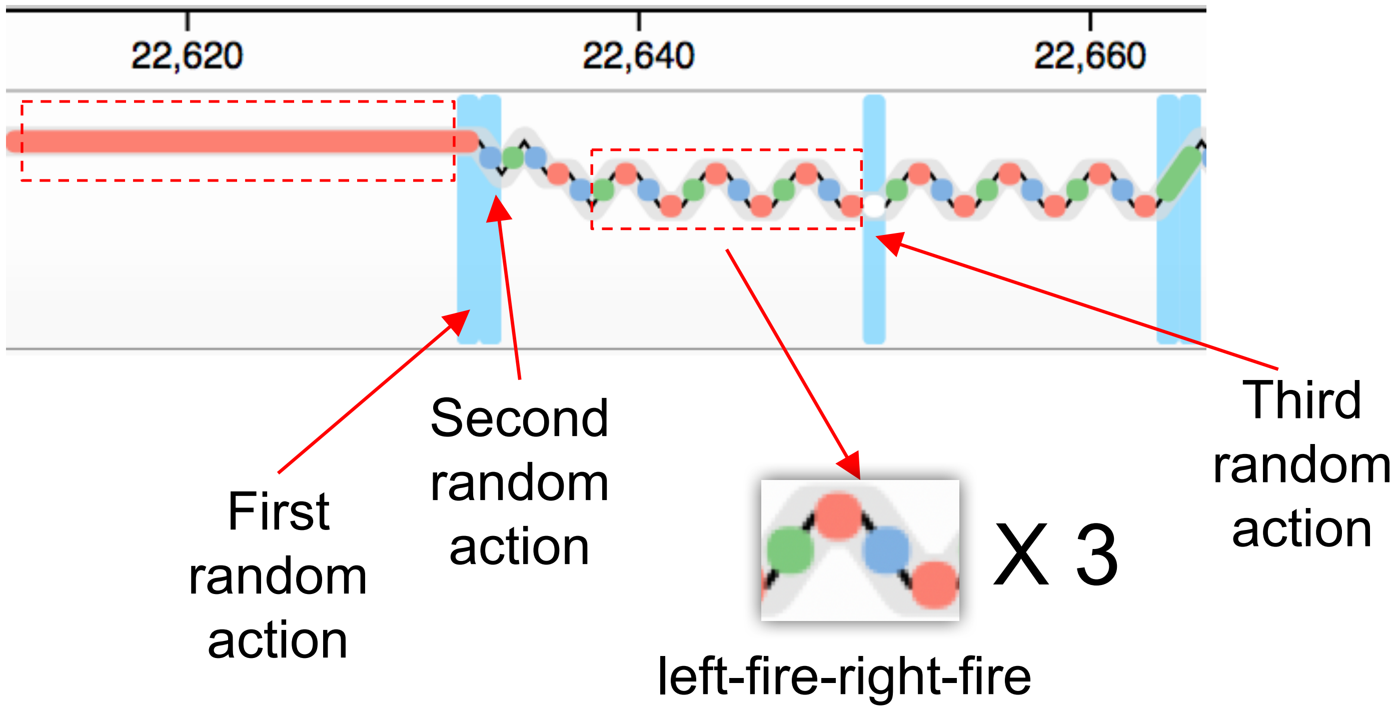
Random actions break the agent's intentional move and cause a life-loss.

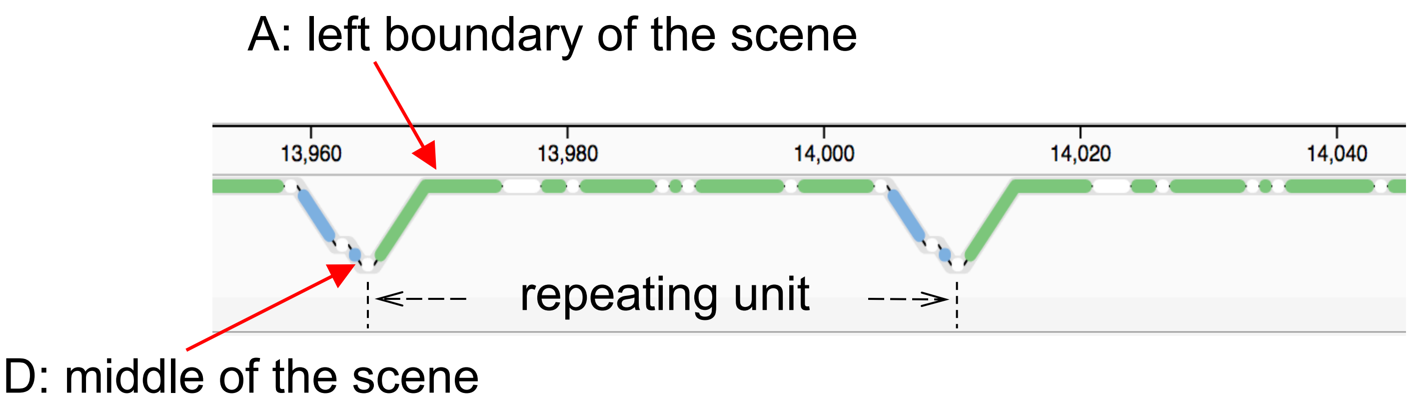
VIS 2018

Improve Random Actions

○ Noop (stay) ● Firing the ball

● Moving left ● Moving right





A-B-C-D-C-B-A

Quantitative Evaluation

Result of averaging over 10 runs

Algorithm	# of Steps	# of episodes	Total rewards	# or random steps
Random rate 5%	25,000	16.6	4198.6	1269.4
Our RegExp Alg.	25,000	11.4	4899.2	503
Random rate 2%	25,000	9.9	3780.8	492.1

Conclusion

- We present DQNViz, a visual analytics system that provides effective multi-level visual summaries of the large multi-faceted data generated from DQN trainings.

- Based on our analysis of the training data, we identified typical movement and reward patterns of the agent, and those patterns have helped in controlling the random actions of the DQN model.

Thanks! Questions?

This work was supported in part by US Department of Energy Los Alamos National Laboratory contract 47145, UT-Battelle LLC contract 4000159447, NSF grants IIS-1250752, IIS1065025, and US Department of Energy grants DE-SC0007444, DEDC0012495, program manager Lucy Nowell.

VIS 2018

The Ohio State University

VISA Research