Arcade Learning Environment

Challenges With Empirical Evaluation

Jesse Farebrother

CMPUT 607

Outline

- Background
- Evolving Methodologies
- Measuring Performance
- Moving Forwards

The Arcade Learning Environment: An Evaluation Platform for General Agents

Marc G. Bellemare MG17@CS.UALBERTA.CA
University of Alberta, Edmonton, Alberta, Canada

Yavar Naddaf YAVAR@EMPIRICALRESULTS.CA
Empirical Results Inc., Vancouver,
British Columbia, Canada

Joel Veness VENESS@CS.UALBERTA.CA
Michael Bowling BOWLING@CS.UALBERTA.CA
University of Alberta, Edmonton, Alberta, Canada

UNIVERSITY OF ALBERTA

Human-level control through deep reinforcement learning

Volodymyr Mnih[1]*, Koray Kavukcuoglu[1]*, David Silver[1]*, Andrei A. Rusu[1], Joel Veness[1], Marc G. Bellemare[1], Alex Graves[1], Martin Riedmiller[1], Andreas K. Fidjeland[1], Georg Ostrovski[1], Stig Petersen[1], Charles Beattie[1], Amir Sadik[1], Ioannis Antonoglou[1], Helen King[1], Dharshan Kumaran[1], Daan Wierstra[1], Shane Legg[1] & Demis Hassabis[1]

Evolving Methodologies

Revisiting the Arcade Learning Environment: Evaluation Protocols and Open Problems for General Agents

Marlos C. Machado MACHADO@UALBERTA.CA
University of Alberta, Edmonton, Canada

Marc G. Bellemare BELLEMARE@GOOGLE.COM
Google Brain, Montréal, Canada

Erik Talvitie ERIK.TALVITIE@FANDM.EDU
Franklin & Marshall College, Lancaster, USA

Joel Veness AIXI@GOOGLE.COM
DeepMind, London, United Kingdom

Matthew Hausknecht MATTHEW.HAUSKNECHT@MICROSOFT.COM
Microsoft Research, Redmond, USA

Michael Bowling MBOWLING@UALBERTA.CA
University of Alberta, Edmonton, Canada
DeepMind, Edmonton, Canada

2015

“We also show that these algorithms produce competitive results when learning policies [on the ***Arcade Learning Environment****]”*

2016

“We perform most of our experiments using the ***Arcade Learning Environment****”*

2017

“We also ran [algorithm] on the ***Arcade Learning Environment*** *benchmark”*

2018

“We evaluated [algorithm] on the ***Arcade Learning Environment****”*

2019

"Atari-57 is a collection of 57 classic [Atari] games. The ***ALE****, exposes them as [RL] environments”*

2015

```lua
local alewrap = require"alewrap"

local env = alewrap.GameEnvironment{
  game_path = "Pong.bin"
}
```

2016

```python
from ale_py import ALEInterface

env = ALEInterface()
env.loadROM("Pong.bin")
```

2017

```python
import gym

env = gym.make("Pong-v0")
```

2018

```python
import gym

env = gym.make("Pong-v4")
```

2021

```python
import gym

env = gym.make("ale-py:Pong-v5")
```

Behind The Scenes

```python
import gym

env = gym.make("ale-py:Pong-v5")
```

Frame Skip	5
Stochasticity	Repeat Actions
Frame Post-Processing	Max-Pool
All-Actions	TRUE
Deterministic	TRUE
Distribution	Arcade Learning Environment

```lua
local alewrap = require"alewrap"

local env = alewrap.GameEnvironment{
  game_path = "Pong.bin"
}
```

Frame Skip	**4**
Stochasticity	**Start State**
Frame Post-Processing	**Phosphor**
All-Actions	**FALSE**
Deterministic	**FALSE**
Distribution	**DeepMind Xitari**

Behind The Scenes

```python
import gym

env = gym.make("ale-py:Pong-v5")
```

Frame Skip	5
Stochasticity	Repeat Actions
Frame Post-Processing	Max-Pool
All-Actions	TRUE
Deterministic	TRUE
Distribution	Arcade Learning Environment

```python
import gym

env = gym.make("Pong-v0")
```

Frame Skip	**4**
Stochasticity	Repeat Actions
Frame Post-Processing	Max-Pool
All-Actions	**FALSE**
Deterministic	**FALSE**
Distribution	**OpenAI atari-py**

Behind The Scenes

```python
import gym

env = gym.make("ale-py:Pong-v5")
```

Frame Skip	5
Stochasticity	Repeat Actions
Frame Post-Processing	Max-Pool
All-Actions	TRUE
Deterministic	TRUE
Distribution	Arcade Learning Environment

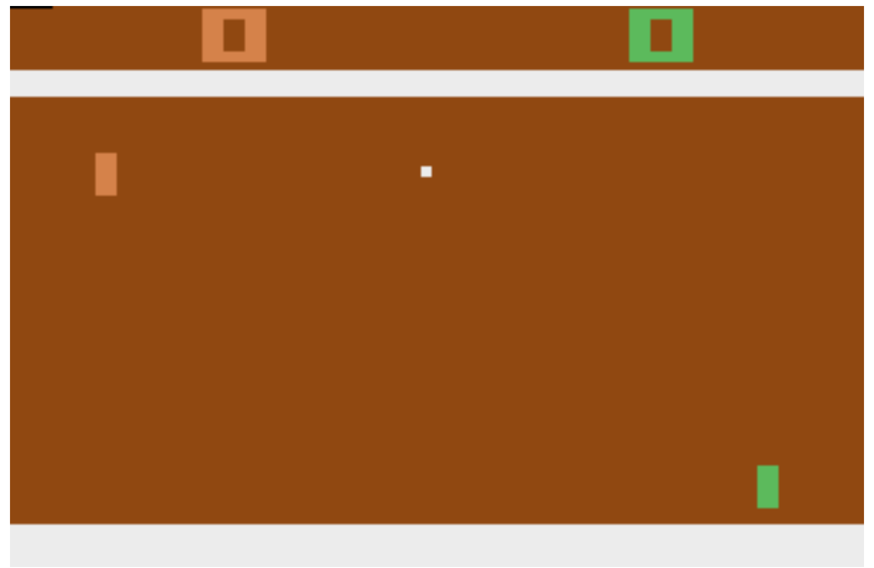
```python
import gym

env = gym.make("Pong-v4")
```

Frame Skip	**Uniform(2, 5)**
Stochasticity	**Frame Skip**
Frame Post-Processing	Max-Pool
All-Actions	**FALSE**
Deterministic	**FALSE**
Distribution	**OpenAI atari-py**

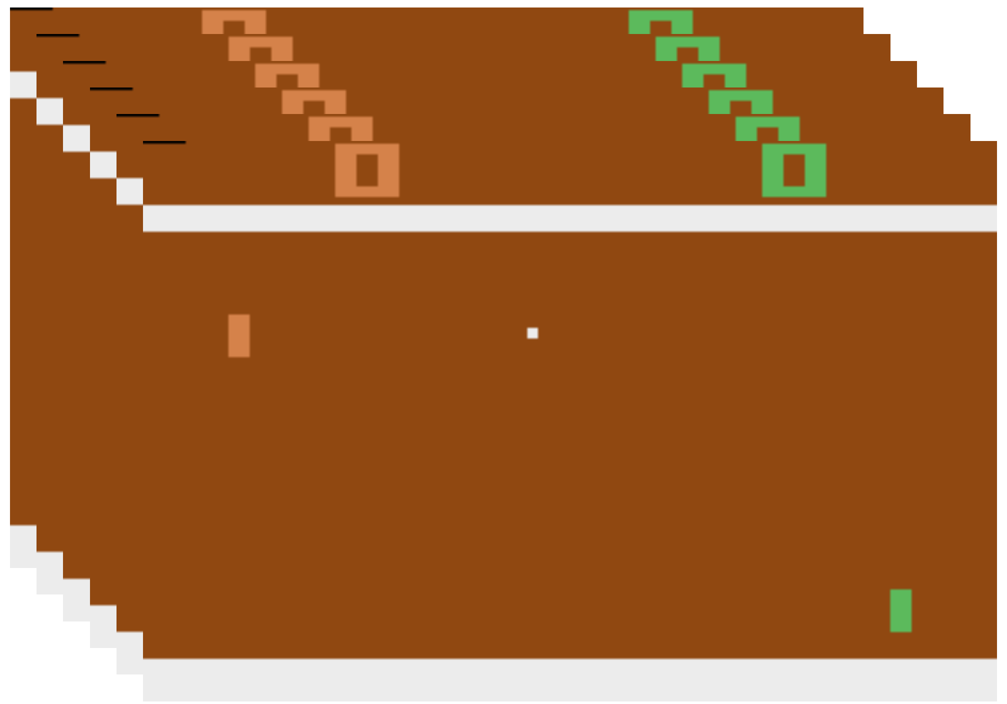
What's the extent of the issue?

Would it have a perceptible impact on results?

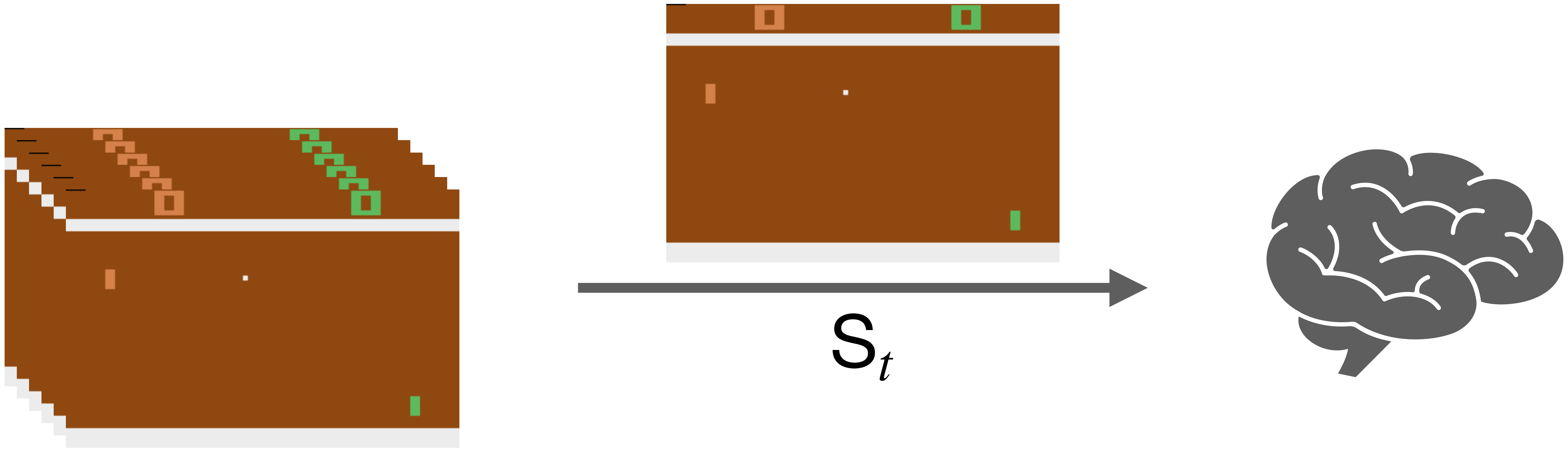
Frame Skip



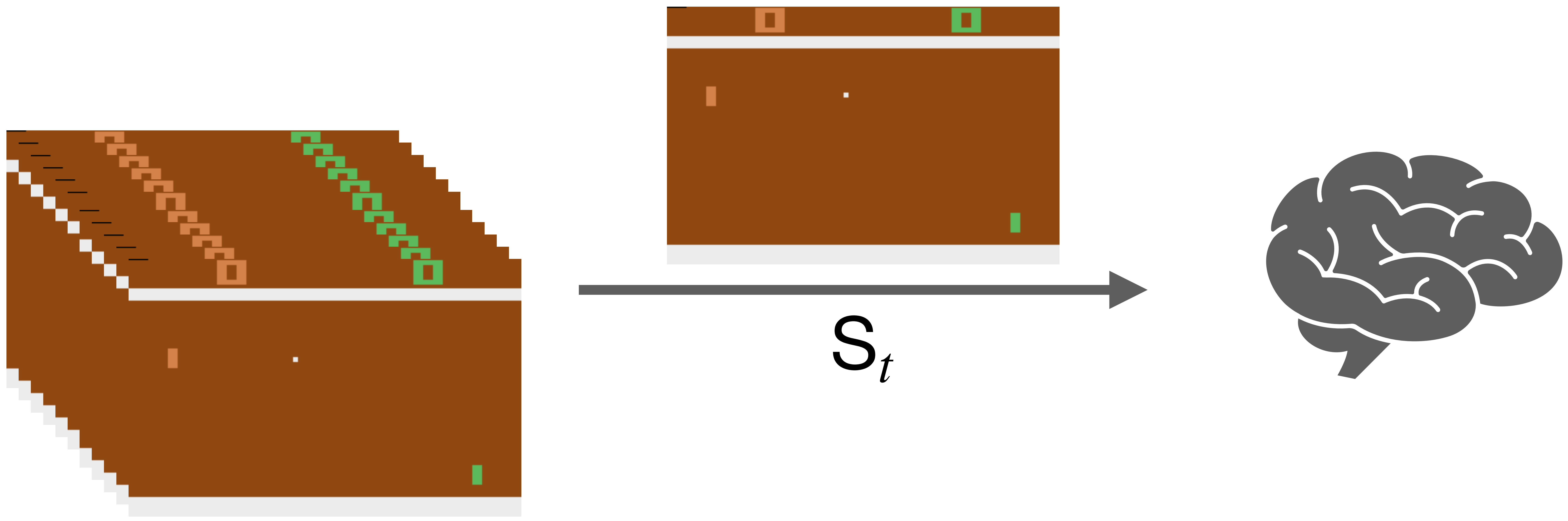
Frame Skip



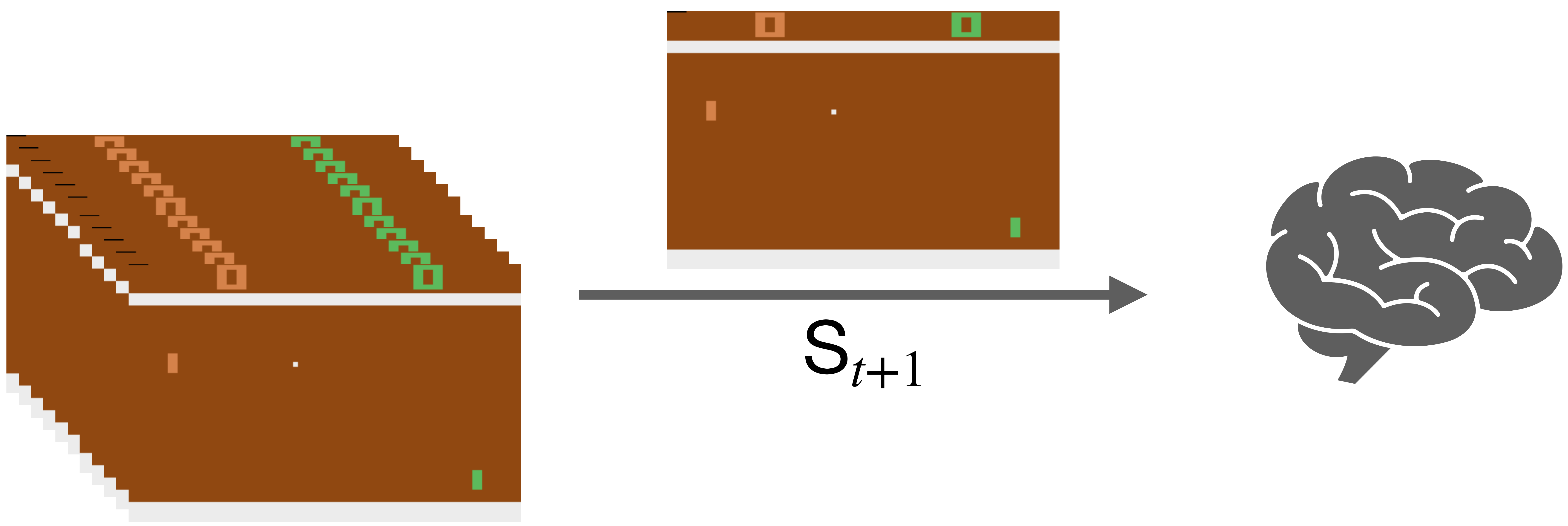
Frame Skip



Frame Skip

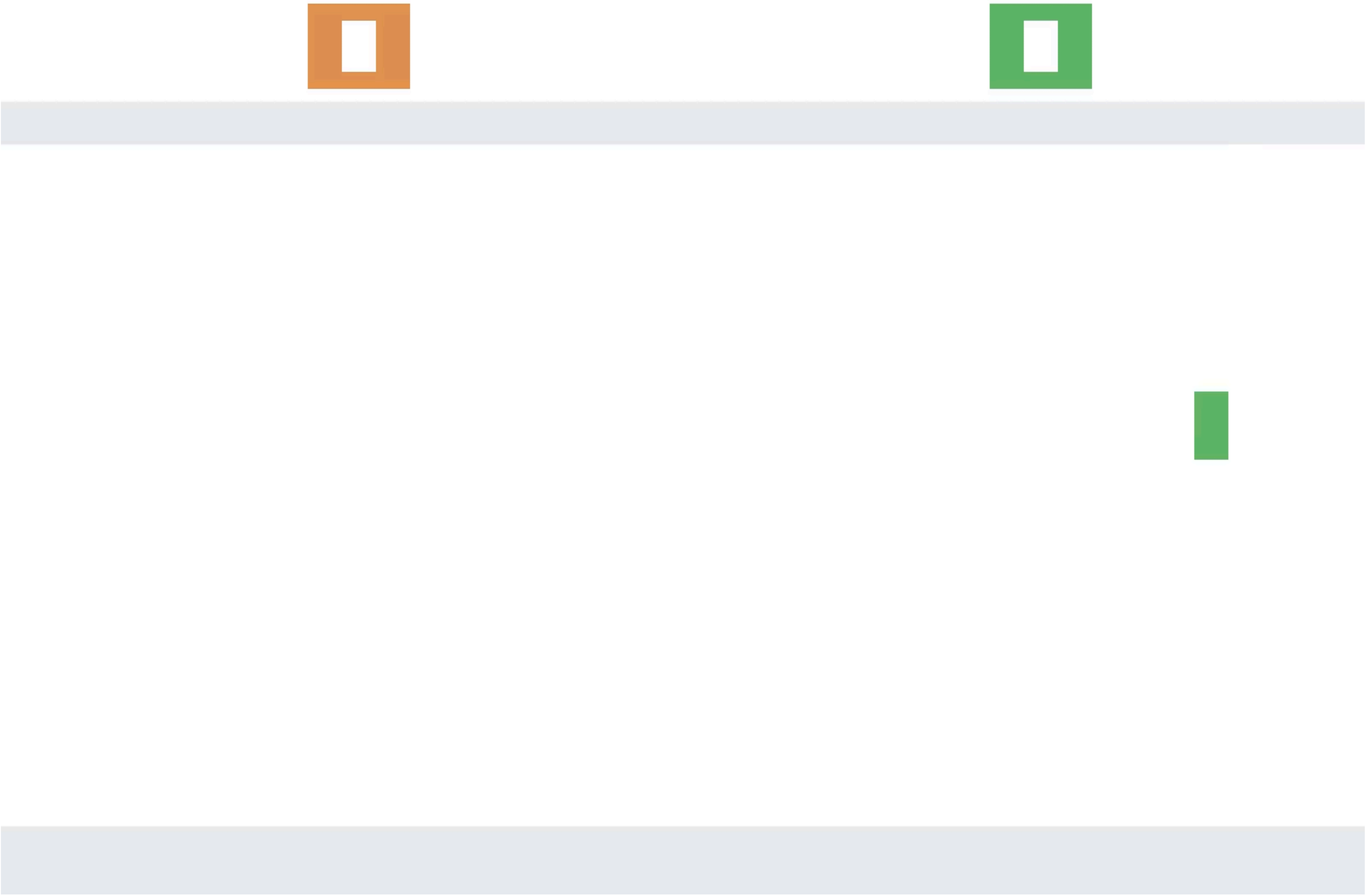


Frame Skip

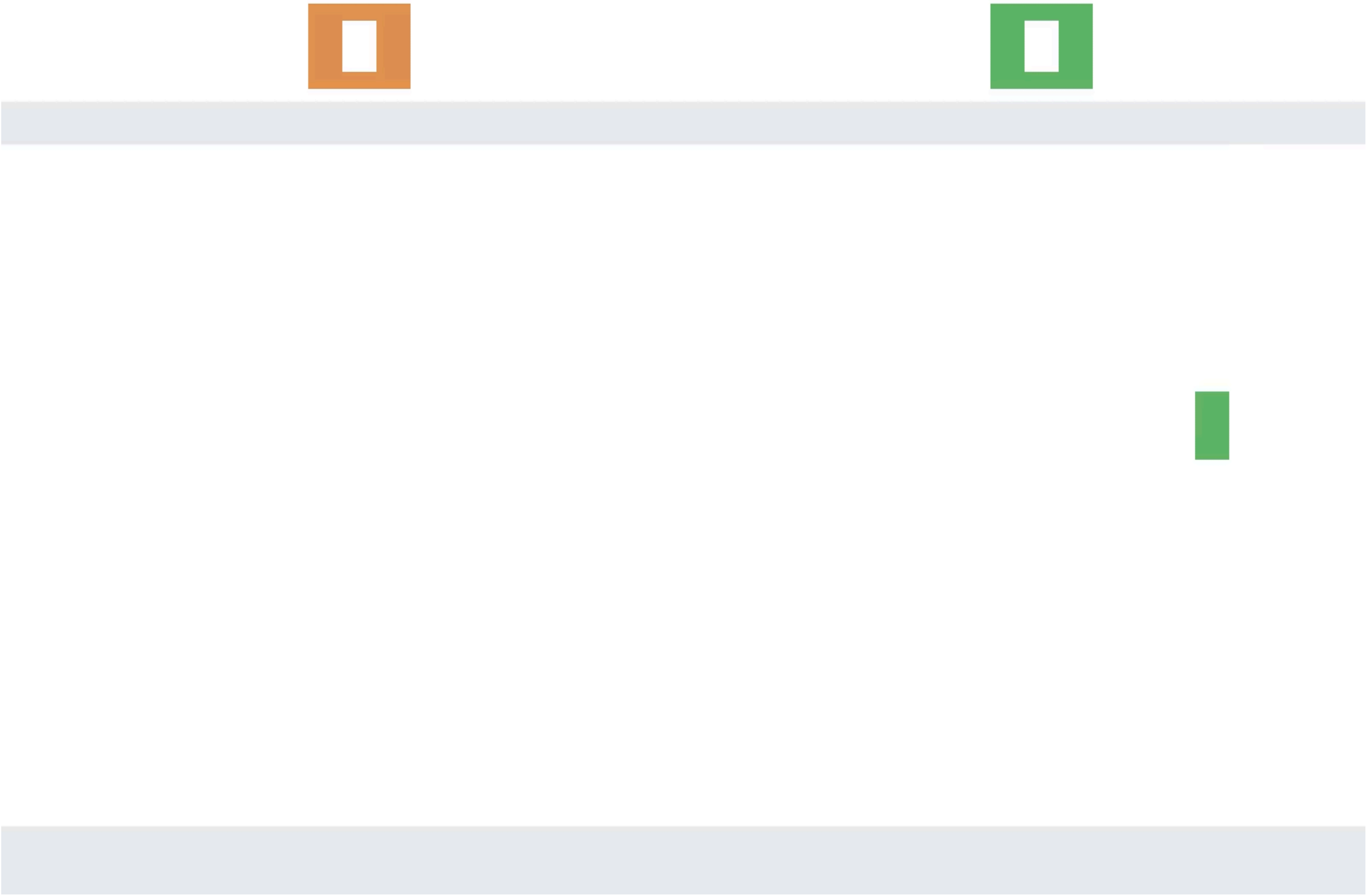


Frame Skip

How do we fare?



Frame Skip: 4

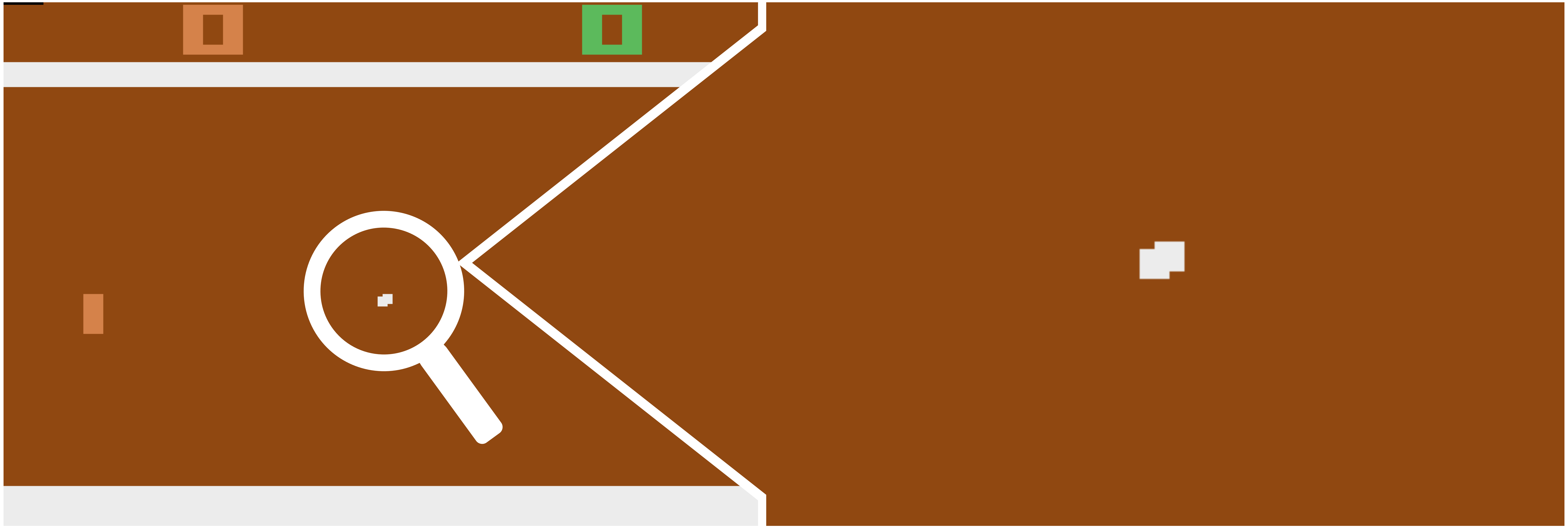


Frame Skip: 5

Trained Rainbow Agent @ 200M Frames (Castro et al., 2018)

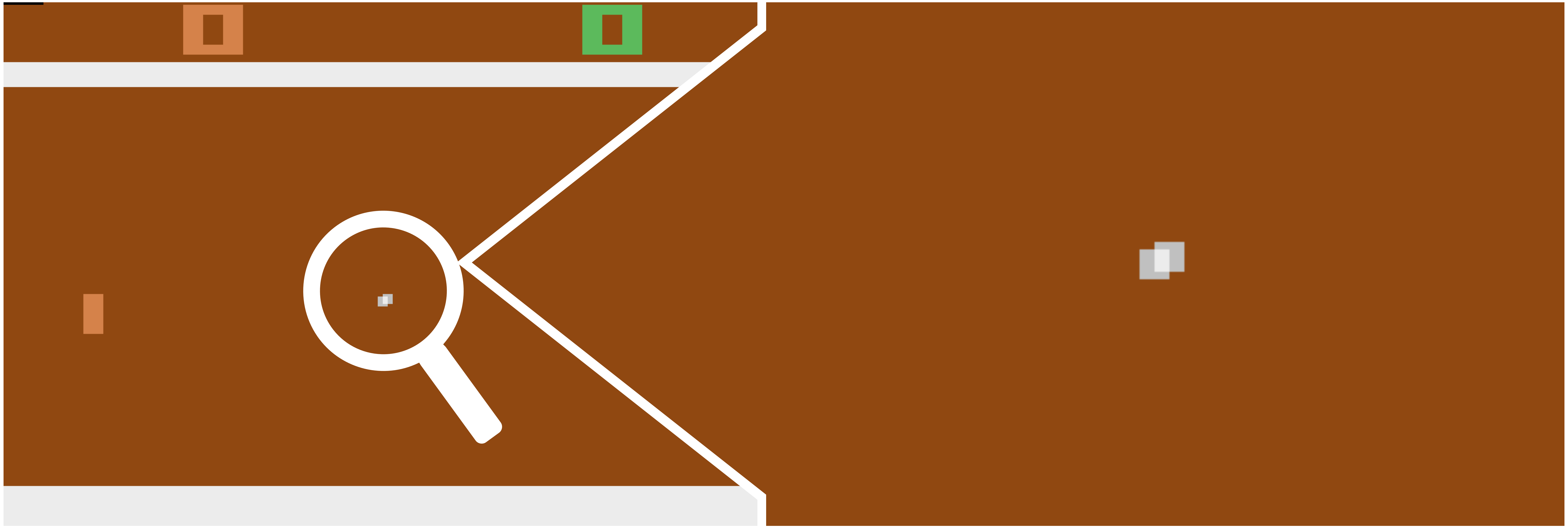
Frame Post-Processing

Max-Pool



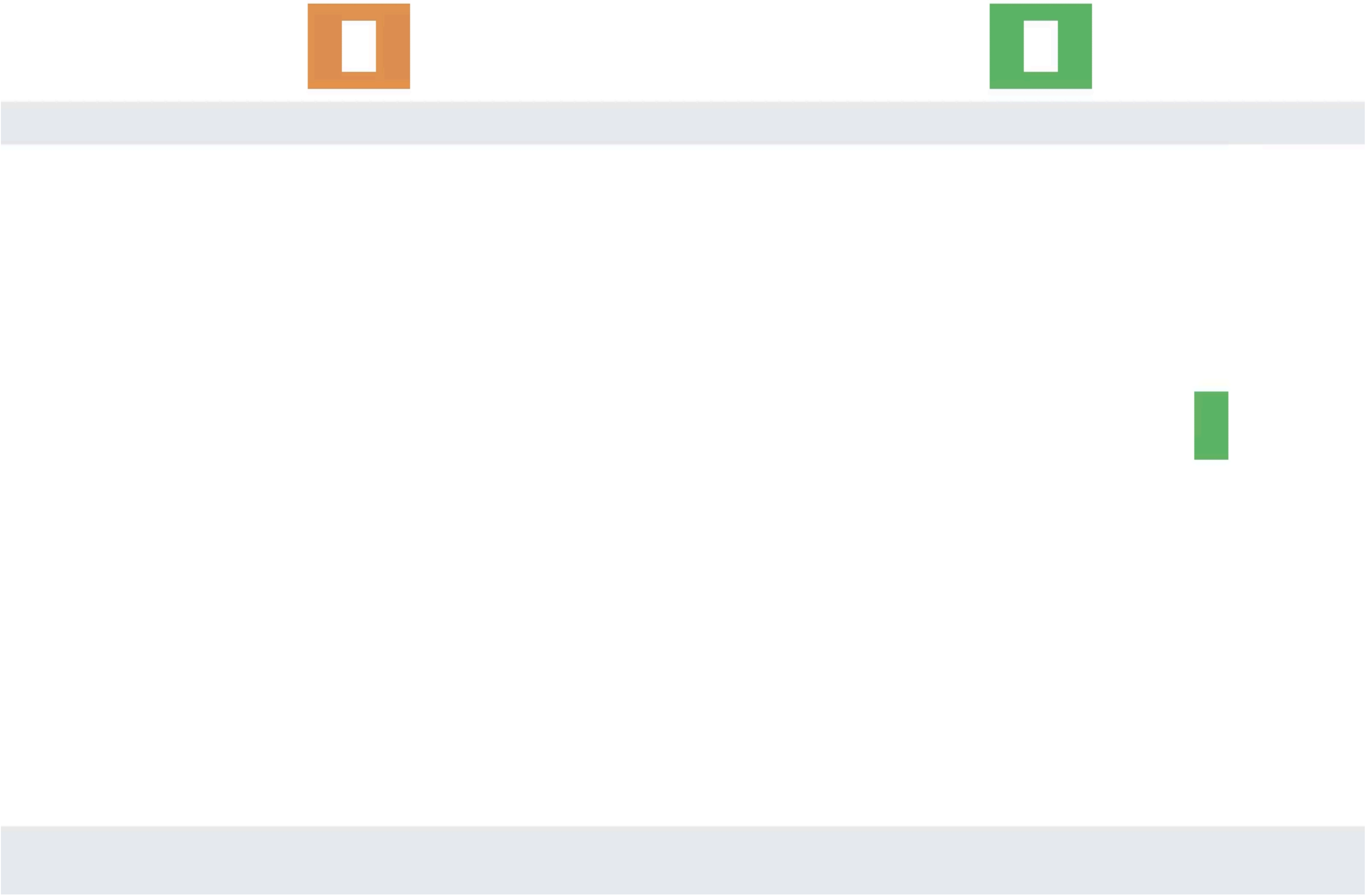
Frame Post-Processing

Phosphor Blend (Colour Averaging)

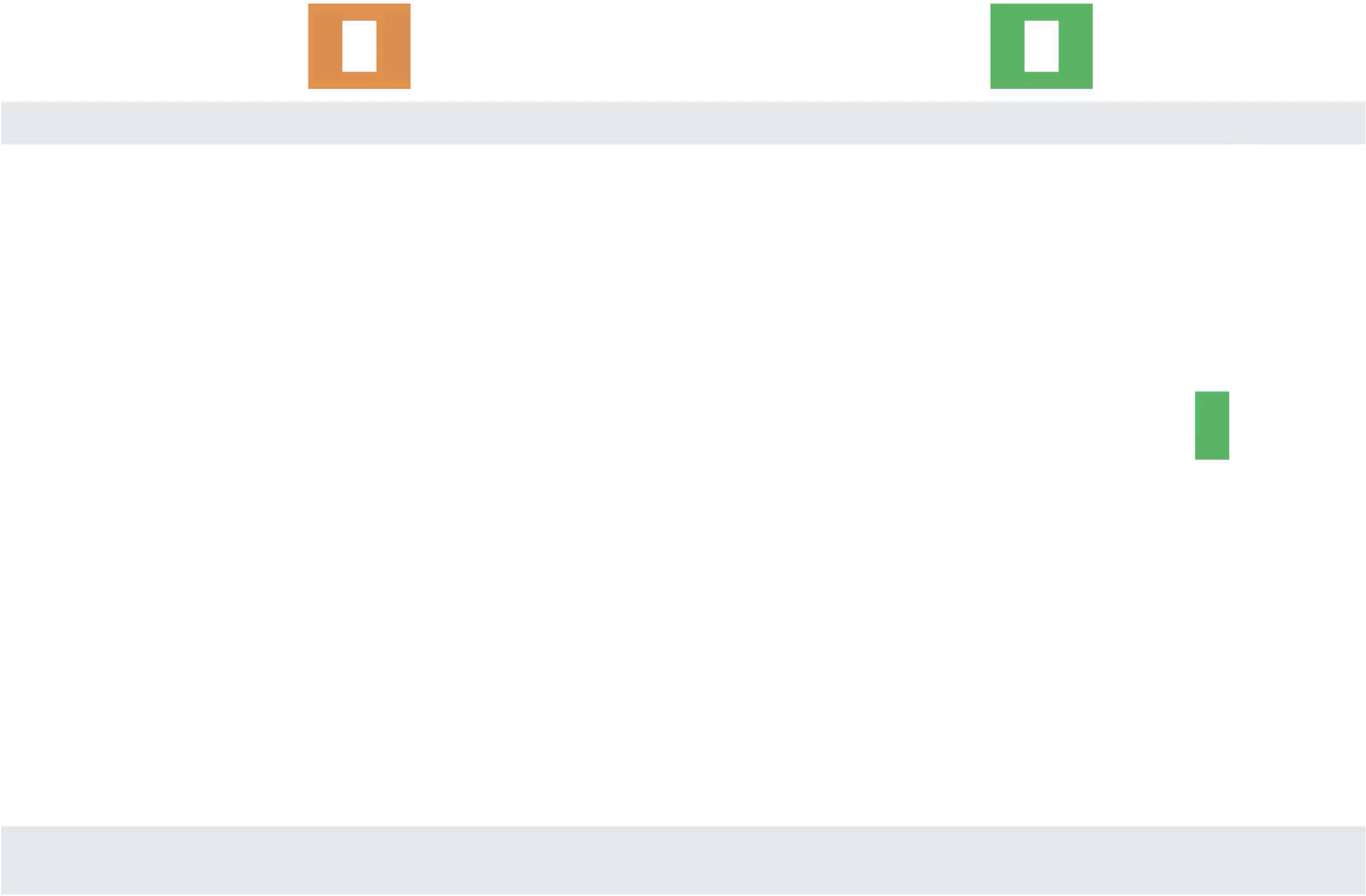


Frame Post-Processing

How do we fare?



Post-Processing: Max-Pool



Post-Processing: Phosphor

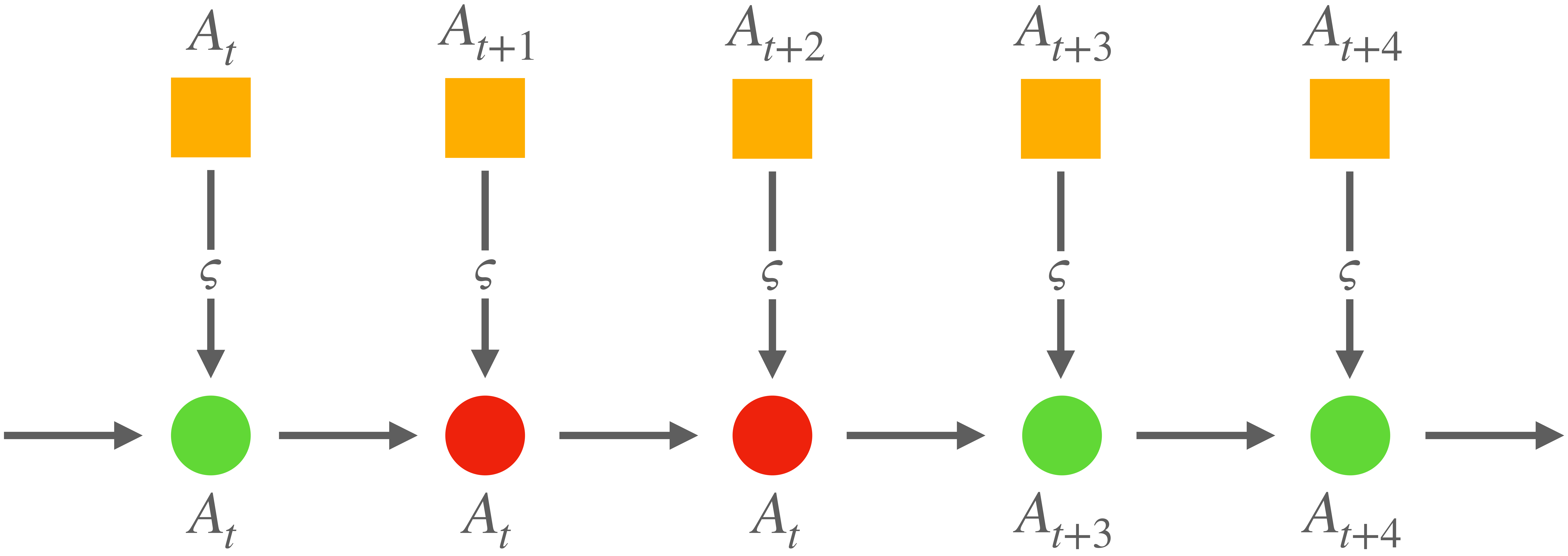
Trained Rainbow Agent @ 200M Frames (Castro et al., 2018)

More Confounders…

Stochasticity

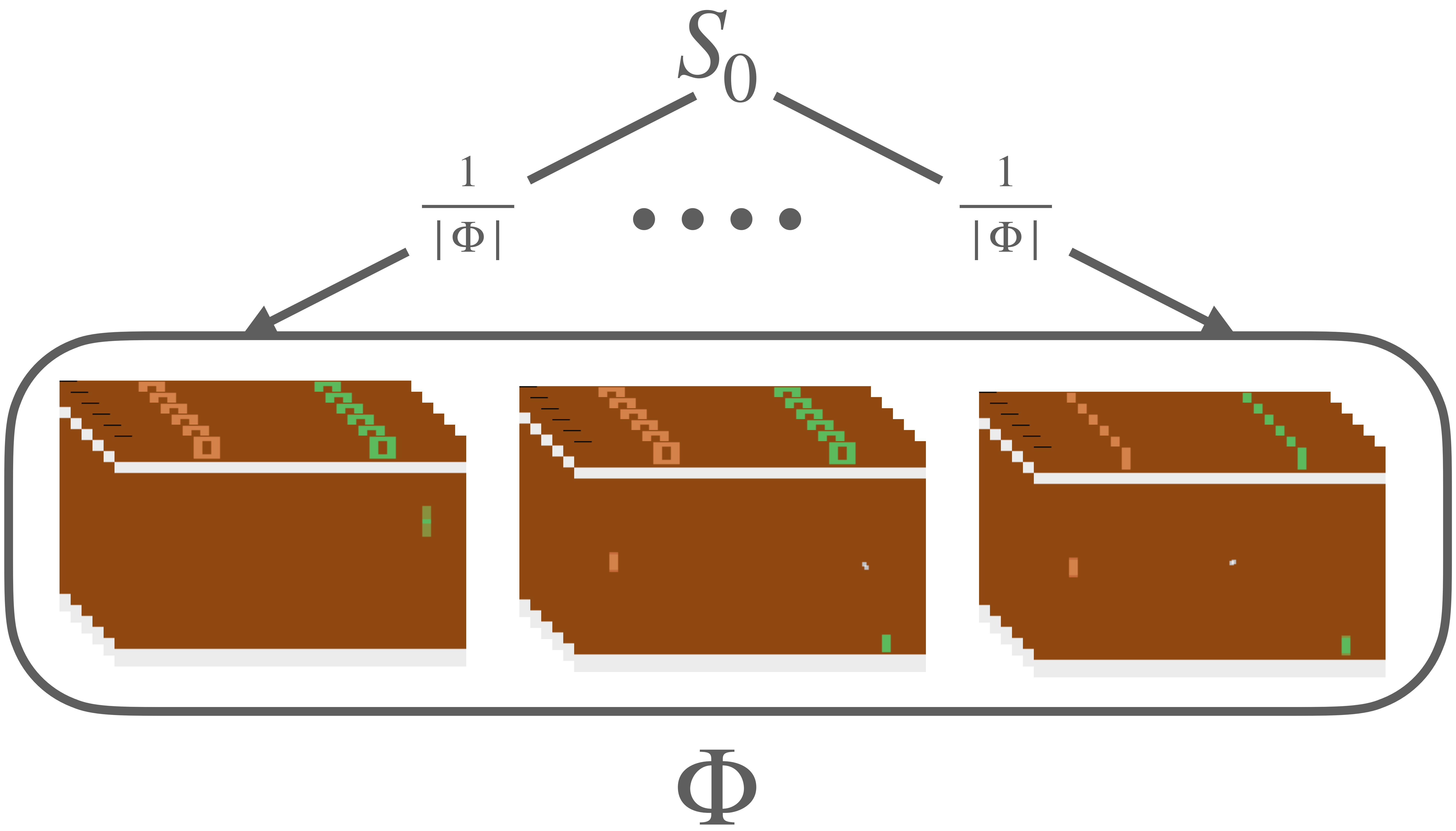
Sticky Actions

$$A_t = \begin{cases} A_t & \text{w.p. } \varsigma \\ A_{t-1} & \text{w.p. } 1 - \varsigma \end{cases}$$



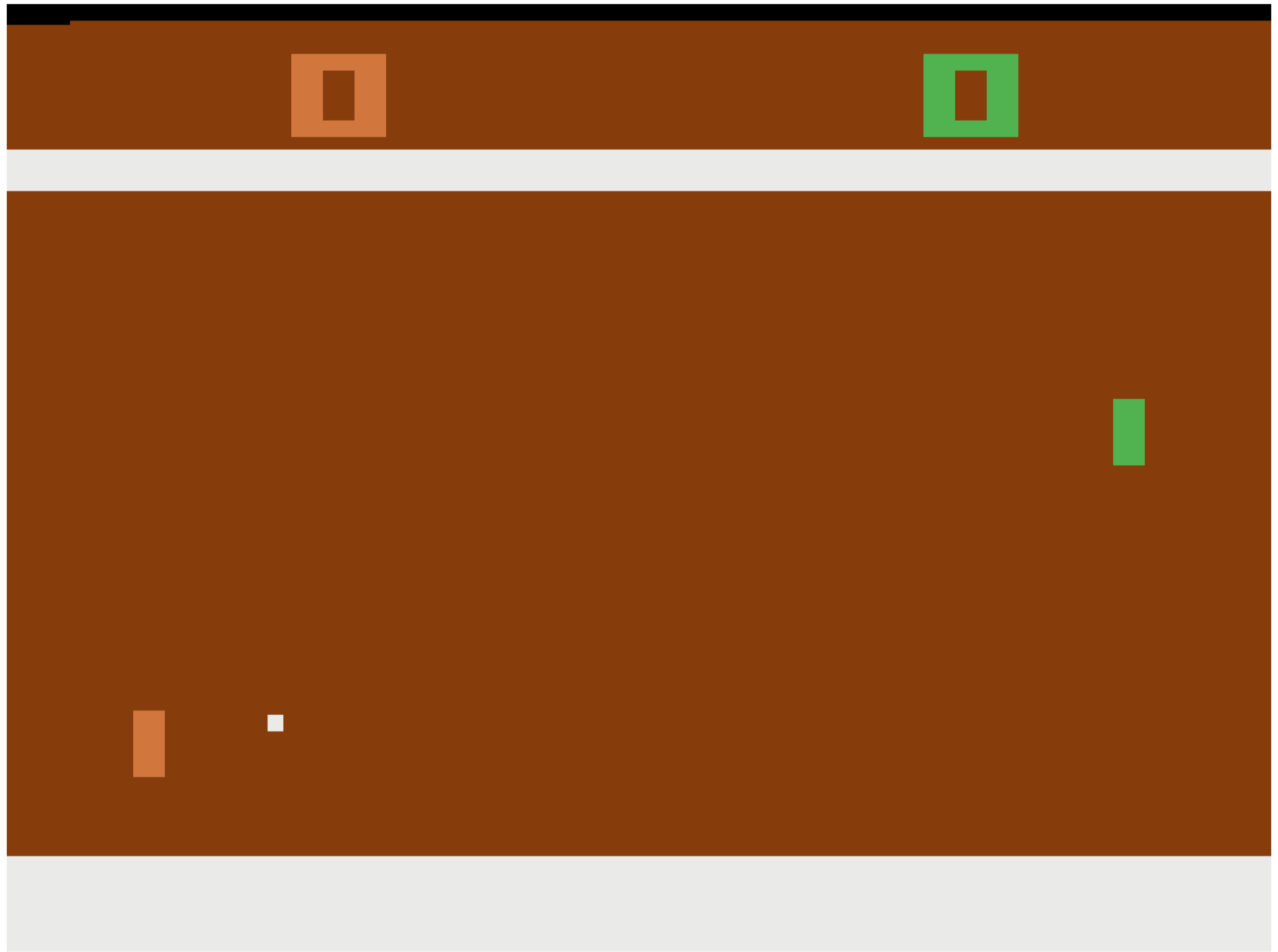
Stochasticity

Random Starts



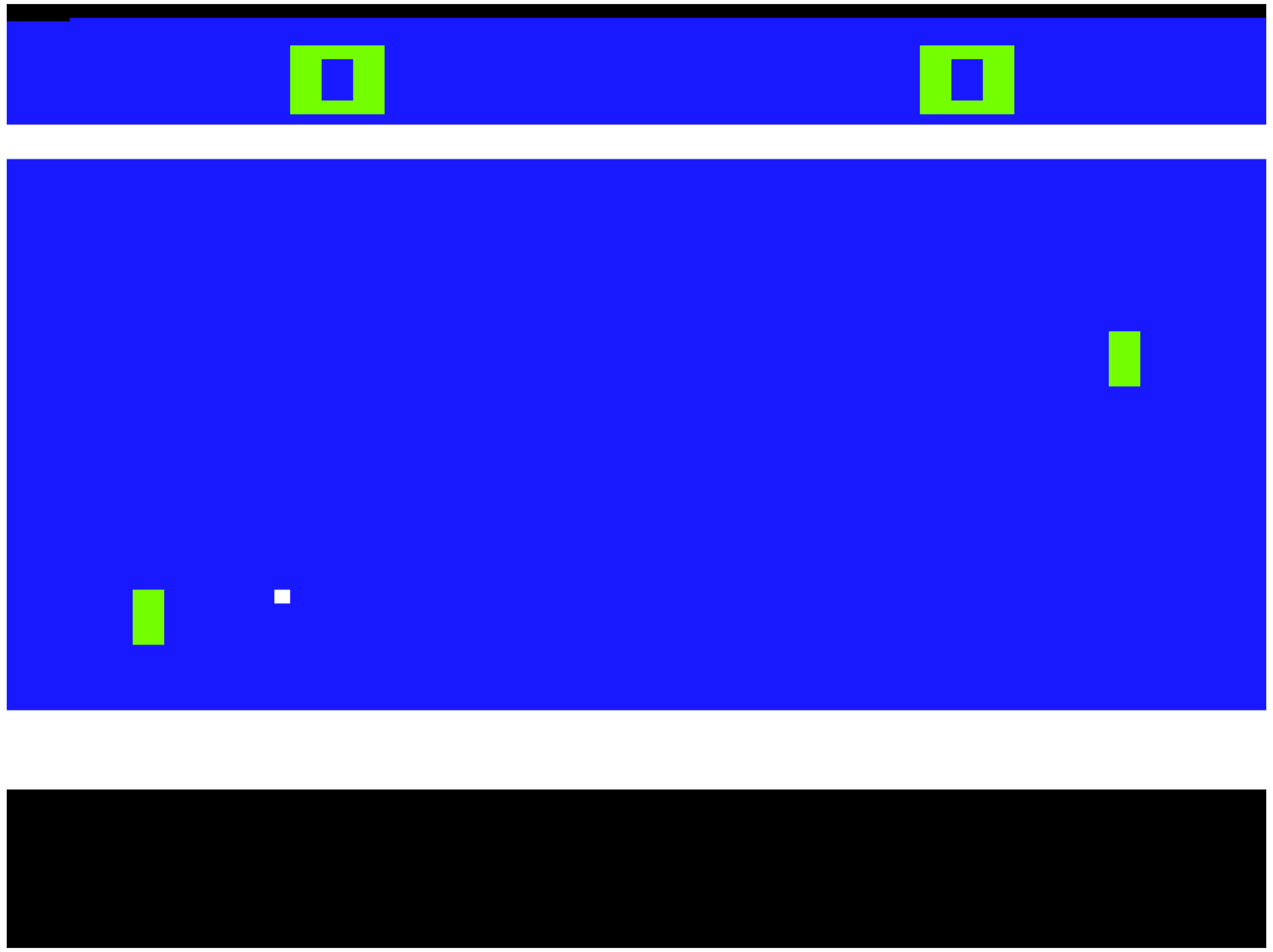
ROM Differences

NTSC



- 60 Hz
- 525 Scanlines
- Wider colour gamut

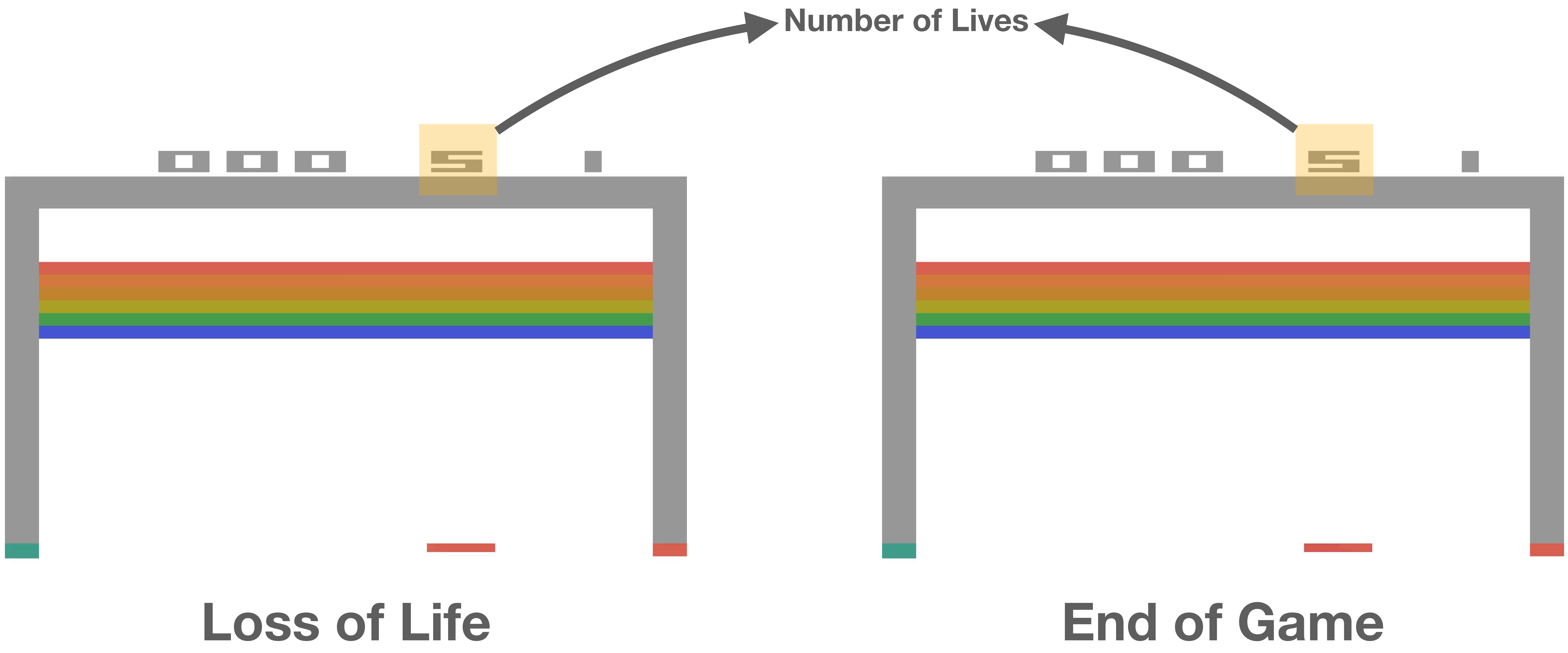
PAL/SECAM



- 50 Hz
- 625 Scanlines
- Limited colour gamut

Terminal State

Loss of life?



Measuring Performance

Score?

- Hard to analyze for those unfamiliar with the benchmark.
- **Return distributions** can be **multimodal** leading to large variance with few seeds.
- **Naive baselines** can help to ground results, especially on obscure games.

Game	10M frames		50M frames		100M frames		200M frames	
Alien	600.5	(23.6)	1,426.6	(81.6)	1,952.6	(216.0)	2,742.0	(357.5)
Amidar	91.6	(10.5)	414.2	(53.6)	621.6	(92.6)	792.6	(220.4)
Assault	688.9	(16.0)	1,327.5	(83.9)	1,433.9	(126.6)	1,424.6	(106.8)
Asterix†	1,732.6	(314.6)	3,122.6	(96.4)	3,423.4	(213.6)	2,866.8	(1,354.6)
Asteroids	301.4	(14.3)	458.1	(28.5)	458.0	(18.9)	528.5	(37.0)
Atlantis	6,639.4	(208.4)	51,324.4	(8,681.7)	291,134.7	(31,575.2)	232,442.9	(128,678.4)
Bank Heist	32.3	(6.5)	448.2	(104.8)	740.7	(130.6)	760.0	(82.3)
Battle Zone	2,428.3	(200.4)	10,838.4	(1,807.6)	15,048.5	(2,372.0)	20,547.5	(1,843.0)
Beam Rider †	693.9	(111.0)	4,551.5	(849.1)	4,977.2	(292.2)	5,700.5	(362.5)
Berzerk	434.5	(51.2)	457.5	(9.4)	470.0	(24.5)	487.2	(29.9)
Bowling	28.7	(0.8)	29.4	(1.8)	32.8	(3.6)	33.6	(2.7)
Boxing	18.6	(3.8)	71.7	(2.7)	77.9	(0.5)	72.7	(4.9)
Breakout	14.2	(1.2)	75.1	(4.3)	57.9	(14.6)	35.1	(22.6)
Carnival	588.5	(47.0)	2,131.6	(534.3)	4,621.9	(191.0)	4,803.8	(189.0)
Centipede	3,075.2	(381.1)	2,280.0	(184.2)	2,555.2	(195.1)	2,838.9	(225.3)
Chopper Command	841.4	(144.3)	2,104.8	(327.7)	3,288.1	(339.2)	4,399.6	(401.5)
Crazy Climber	43,716.6	(2,571.2)	80,599.6	(4,209.8)	64,807.3	(26,100.0)	78,352.1	(1,967.3)
Defender	2,409.9	(78.6)	2,525.7	(124.0)	2,711.6	(96.8)	2,941.3	(106.2)
Demon Attack	154.8	(11.5)	3,744.6	(688.9)	4,556.5	(947.2)	5,182.0	(778.0)
Double Dunk	-20.9	(0.3)	-18.4	(1.2)	-15.6	(1.6)	-8.7	(4.5)
Elevator Action	6.7	(13.3)	4.5	(9.0)	4.7	(9.4)	6.0	(10.4)
Enduro	473.2	(22.3)	578.0	(79.6)	597.4	(153.1)	688.2	(32.4)
Fishing Derby	-63.1	(7.8)	7.5	(4.1)	12.2	(1.4)	10.2	(1.9)
Freeway†	13.8	(8.1)	31.7	(0.7)	32.4	(0.3)	33.0	(0.3)
Frostbite	241.8	(30.8)	292.5	(28.8)	274.3	(8.8)	279.6	(13.9)
Gopher	679.6	(35.2)	2,233.7	(123.1)	2,988.8	(514.4)	3,925.5	(521.4)
Gravitar	79.5	(8.0)	109.3	(3.1)	118.5	(22.0)	154.9	(17.7)
H.E.R.O.	1,667.9	(1,107.8)	11,564.0	(3,722.4)	14,684.7	(1,840.6)	18,843.3	(2,234.9)
Ice Hockey	-15.1	(0.3)	-8.9	(1.7)	-4.4	(2.0)	-3.8	(4.7)
James Bond	30.7	(6.0)	191.4	(144.9)	517.2	(35.8)	581.0	(21.3)
Journey Escape	-2,220.0	(176.1)	-2,409.7	(341.2)	-2,959.0	(383.9)	-3,503.0	(488.5)
Kangaroo	298.6	(56.1)	8,878.8	(2,886.1)	12,846.9	(688.3)	12,291.7	(1,115.9)
Krull	4,424.7	(492.7)	6,035.6	(248.6)	6,589.8	(264.4)	6,416.0	(128.5)
Kung-Fu Master	9,468.1	(1,975.9)	17,537.4	(1,128.8)	17,772.3	(3,423.3)	16,472.7	(2,892.7)
Montezuma's Revenge	0.2	(0.4)	0.2	(0.4)	0.0	(0.0)	0.0	(0.0)
Ms. Pac-Man	1,675.5	(41.9)	2,626.1	(139.8)	2,964.9	(100.8)	3,116.2	(141.2)
Name This Game	2,265.6	(171.0)	4,105.4	(932.3)	4,105.6	(653.5)	3,925.2	(660.2)
Phoenix	1,501.2	(278.1)	3,174.0	(543.5)	2,607.1	(644.1)	2,831.0	(581.0)
Pitfall!	-24.9	(14.8)	-28.2	(13.0)	-23.3	(9.6)	-21.4	(3.2)
Pong	-15.9	(1.0)	12.2	(1.0)	15.2	(0.7)	15.1	(1.0)
Pooyan	2,278.9	(273.7)	3,528.9	(256.3)	3,387.8	(182.8)	3,700.4	(349.5)
Private Eye	81.6	(15.6)	60.4	(92.4)	1,447.4	(2,567.9)	3,967.5	(5,540.6)
Q*bert	674.7	(53.6)	3,142.1	(1,238.7)	7,585.4	(2,787.4)	9,875.5	(1,385.3)
River Raid	3,166.2	(125.2)	8,738.1	(500.0)	10,733.1	(229.9)	10,210.4	(435.0)
Road Runner	14,742.2	(1,553.4)	37,271.7	(1,234.5)	41,918.4	(1,762.5)	42,028.3	(1,492.0)
Robotank	4.1	(0.3)	28.4	(1.4)	38.0	(1.6)	58.0	(6.4)
Seaquest†	311.5	(36.9)	1,430.8	(162.3)	1,573.4	(561.4)	1,485.7	(740.8)
Skiing	-20,837.5	(1,550.2)	-17,545.5	(4,041.5)	-13,365.1	(800.7)	-12,446.6	(1,257.9)
Solaris	1,030.2	(40.3)	977.7	(112.5)	783.4	(55.3)	1,210.0	(148.3)
Space Invaders †	211.6	(14.8)	686.6	(37.0)	787.2	(173.3)	823.6	(335.0)
Star Gunner	603.0	(28.0)	1,492.3	(79.7)	11,590.5	(4,658.9)	39,269.9	(5,298.8)
Tennis	-23.8	(0.1)	-23.9	(0.1)	-23.9	(0.0)	-23.9	(0.0)
Time Pilot	1,078.8	(60.3)	1,068.1	(138.8)	1,330.7	(177.1)	2,061.8	(228.8)
Tutankham	56.5	(10.0)	64.9	(12.6)	65.1	(11.9)	60.0	(12.7)
Up and Down	4,378.4	(172.5)	6,718.3	(671.2)	5,962.8	(618.7)	4,750.7	(1,007.5)
Venture	24.4	(46.9)	21.4	(15.1)	4.4	(5.4)	3.2	(4.7)
Video Pinball	4,009.3	(271.9)	7,817.0	(1,884.4)	16,626.2	(3,740.6)	15,398.5	(2,126.1)
Wizard of Wor	184.2	(22.0)	1,377.4	(71.0)	1,440.6	(237.3)	2,231.1	(820.8)
Yar's Revenge	7,261.4	(777.1)	10,344.8	(452.4)	10,312.3	(528.9)	13,073.4	(1,961.8)
Zaxxon	53.5	(51.0)	672.3	(748.5)	1,638.2	(784.0)	3,852.1	(1,120.7)

Scores Can Mislead

Which agent do you chose?

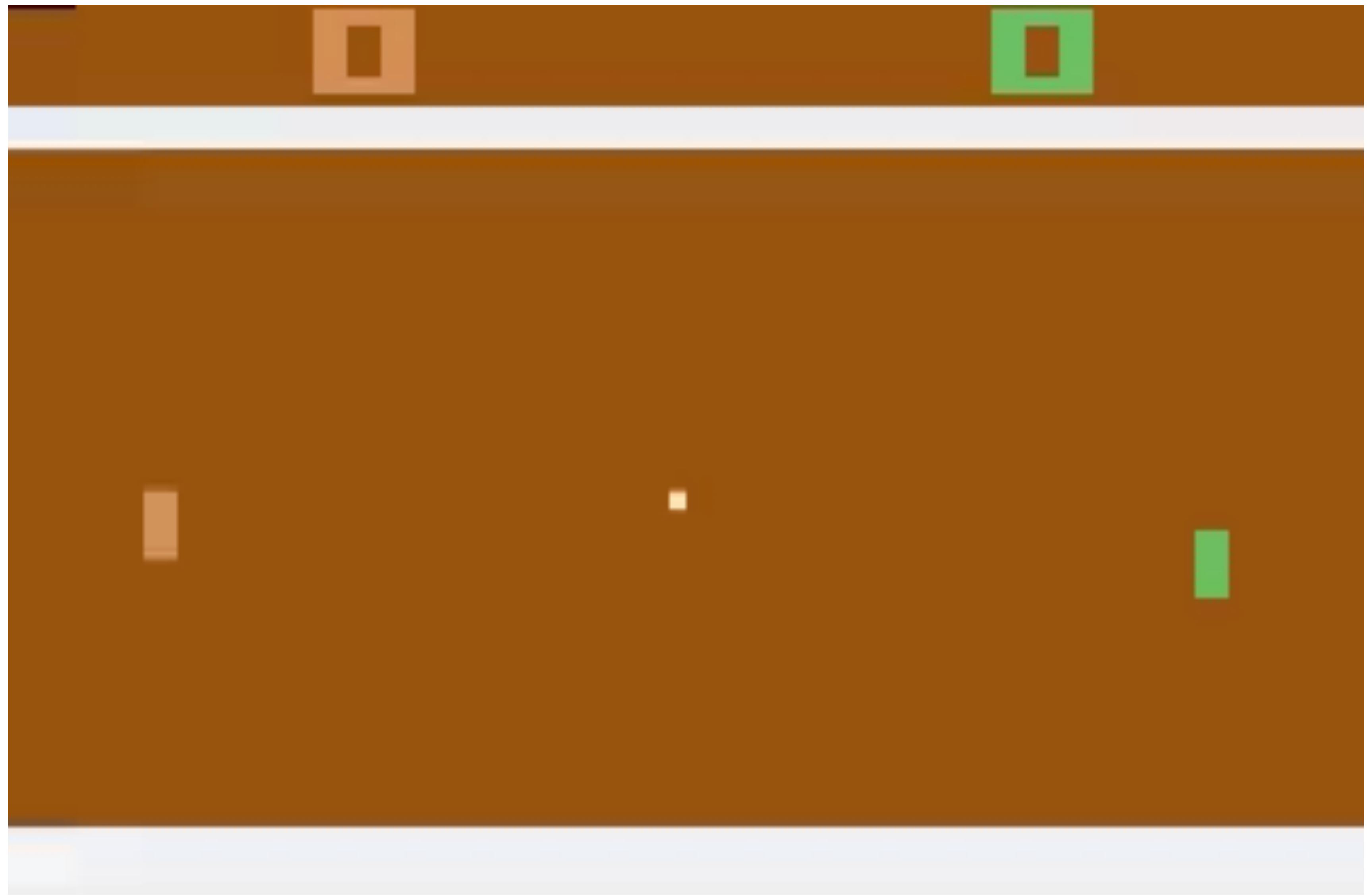
	Cumulative Reward
My Agent	21
A3C	11

Scores Can Mislead

Which agent do you chose? My Agent?

	Cumulative Reward
My Agent	21
A3C	11

My Agent



Computation

Frames

2 Billion Frames

	Gravitar	Montezuma's Revenge	Pitfall!	PrivateEye	Solaris	Venture
RND	**3,906**	**8,152**	-3	8,666	3,282	**1,859**
PPO	3,426	2,497	0	105	3,387	0
Dynamics	3,371	400	0	33	3,246	1,712
SOTA	2,209[1]	3,700[2]	**0**	**15,806**[2]	**12,380**[1]	**1,813**[3]
Avg. Human	3,351	4,753	6,464	69,571	12,327	1,188

Table 1: Comparison to baselines results. Final mean performance for various methods. State of the art results taken from: [1] (Fortunato et al., 2017) [2] (Bellemare et al., 2016) [3] (Horgan et al., 2018)

100 Million Frames

Citing results using different methodologies

Table 3: Performance of the proposed algorithm, $\text{DQN}_e^{\text{MMC}}$+SR, compared to various agents on the "hard exploration" subset of Atari 2600 games. The DQN results reported are from Machado et al. (2018a) while the $\text{DQN}_{\text{CTS}}^{\text{MMC}}$, $\text{DQN}_{\text{PIXELCNN}}^{\text{MMC}}$ and RND results were obtained through personal communication with the authors of the corresponding papers. Burda et al. did not evaluate RND in FREEWAY. When available, standard deviation is reported between parentheses. See text for details.

	DQN		$\text{DQN}_e^{\text{MMC}}$		$\text{DQN}_{\text{CTS}}^{\text{MMC}}$	$\text{DQN}_{\text{PIXELCNN}}^{\text{MMC}}$	RND		$\text{DQN}_e^{\text{MMC}}$+SR	
FREEWAY	32.4	(0.3)	29.5	(0.1)	29.2	29.4	-	-	29.4	(0.1)
GRAVITAR	118.5	(22.0)	1078.3	(254.1)	199.8	275.4	790.0	(122.9)	457.4	(120.3)
MONT. REV.	0.0	(0.0)	0.0	(0.0)	2941.9	1671.7	524.8	(314.0)	1395.4	(1121.8)
PRIVATE EYE	1447.4	(2,567.9)	113.4	(42.3)	32.8	14386.0	61.3	(53.7)	104.4	(50.4)
SOLARIS	783.4	(55.3)	2244.6	(378.8)	1147.1	2279.4	1270.3	(291.0)	1890.1	(163.1)
VENTURE	4.4	(5.4)	1220.1	(51.0)	0.0	856.2	953.7	(167.3)	1348.5	(56.5)

100 Million Frames

50M Frames 1-actor: ~ 24 hours

200M Frames 1-actor: ~ 4 days

2B Frames 1-actor: ~ 40 days

Computation

Seeds

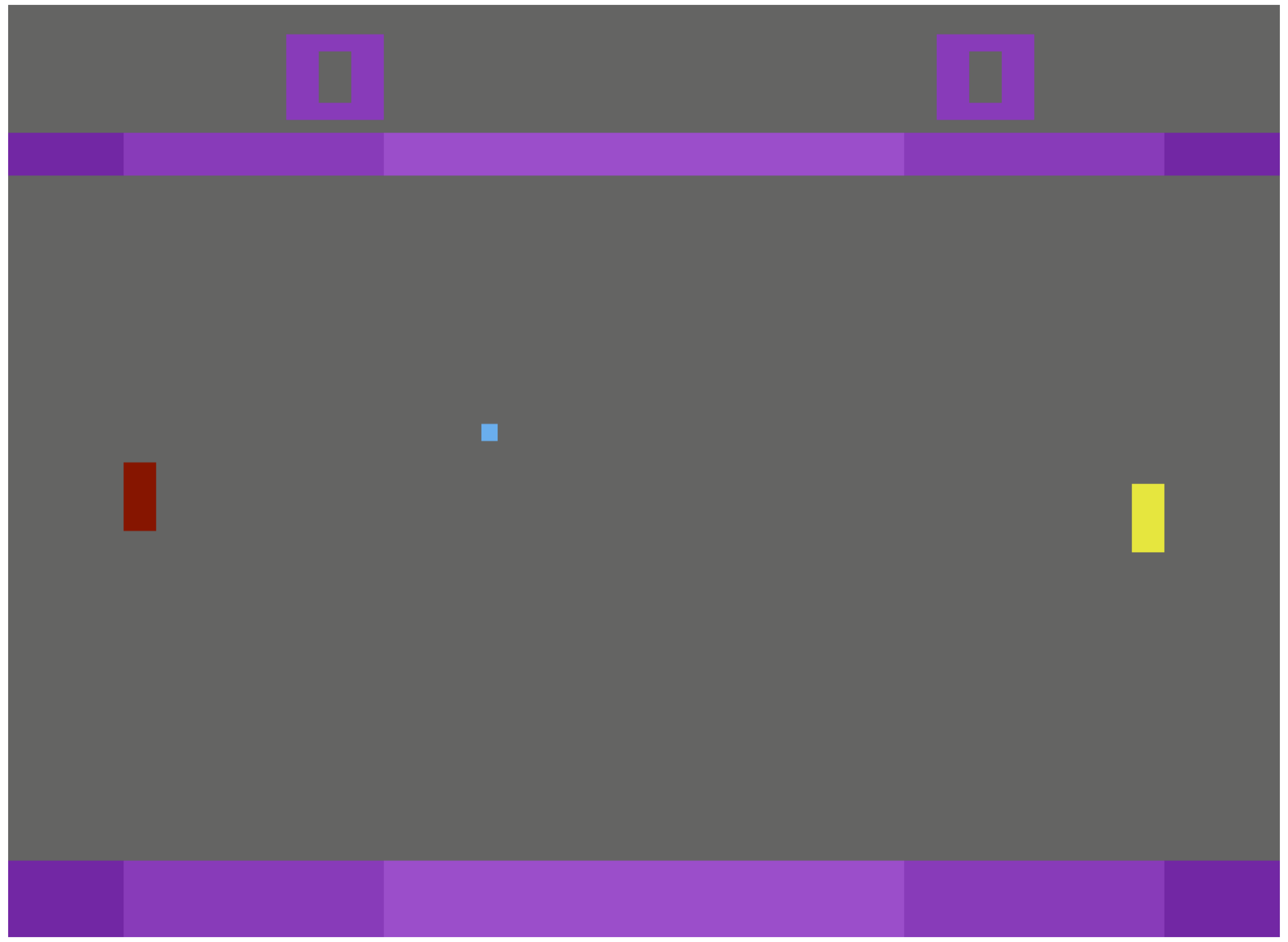
Algorithm	# Seeds
DQN	—
Double DQN	6
Prioritized Replay	8
Duelling DQN	—
PPO	3
TRPO	1
A3C	5
ACER	—
RAINBOW	—

- Computation is expensive
- The entire benchmark is massive…
- Do we need results on every game?

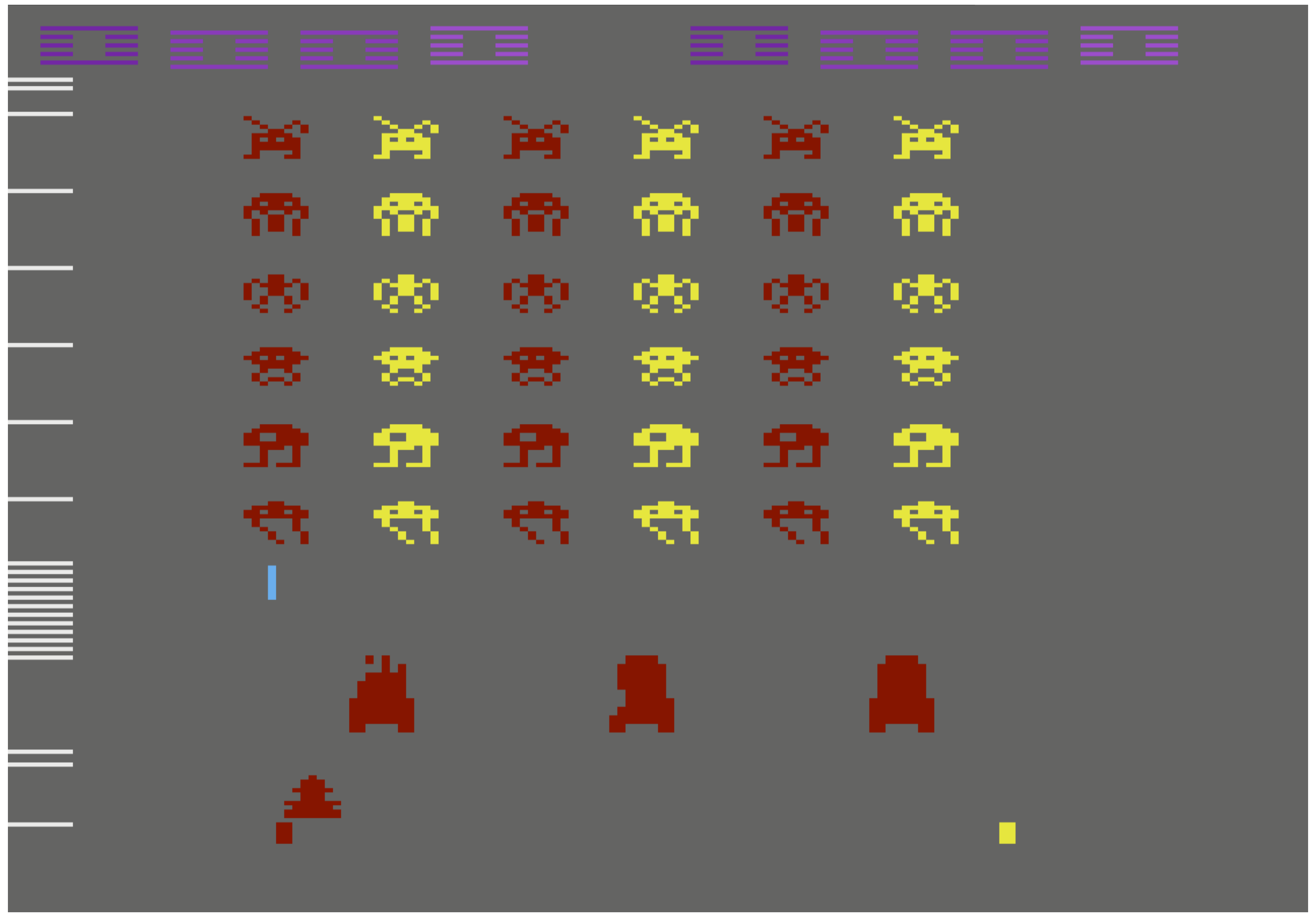
Moving Forwards

Reducing Computation

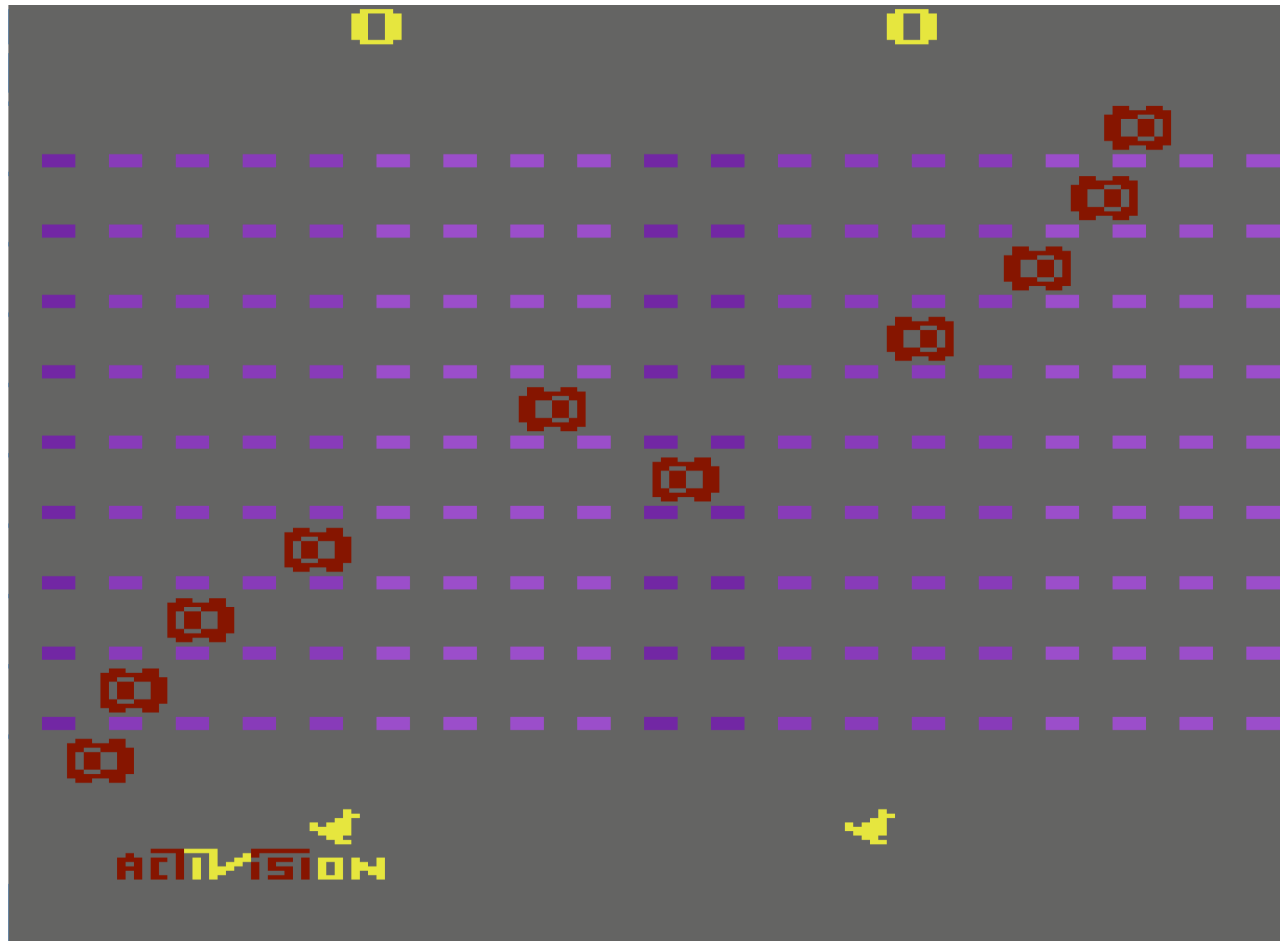
3-bit Color



Pong



Space Invaders



Freeway

Reducing Fragmentation

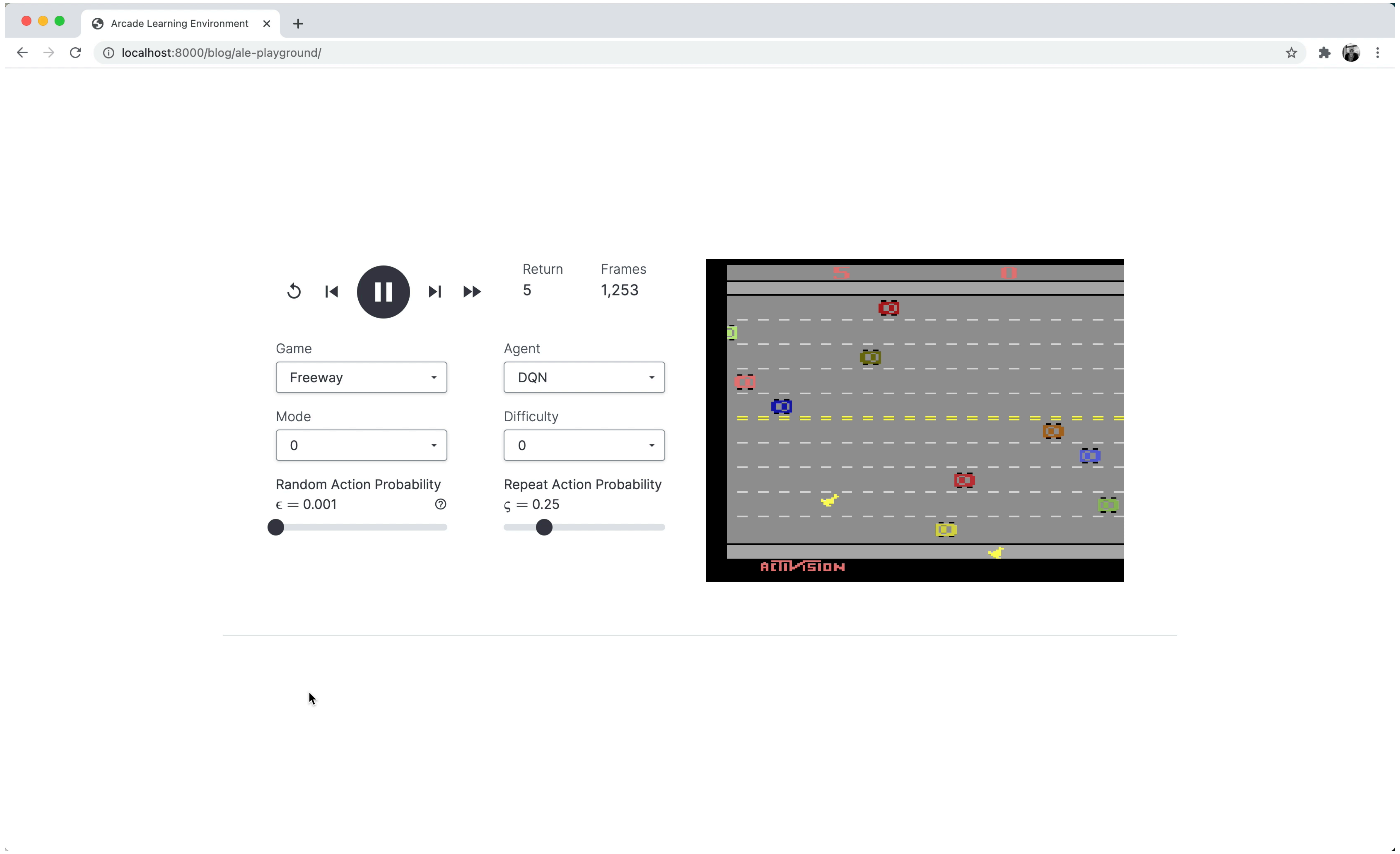
ale-py

- Soon you can use upstream ALE in Gym!
- This will now follow all of the best practices!
- No need to compile the ALE from scratch!
- ROM checksum validation!

```python
import gym

env = gym.make("ale-py:Pong-v5")
```

Accessible Visualizations

ale-ts



Takeaways

- **Hyperparameters**, **hyperparameters**, **hyperparameters**. Report **ALL** environment hyperparameters.
- Benchmarks evolve. **Fragmentation** can become an issue.
- Take **caution** when **directly citing results**. What methodologies were used? If possible, gather data yourself.
- Be **wary** of **results with few seeds**, there’s well known games that have multimodal return distributions.

Thanks!