

# MinAtar: An Atari-Inspired Testbed for Thorough and Reproducible Reinforcement Learning Experiments

Authored by:

Tian Tian and Kenny Young in Univ of Alberta Edmonton

This research is maybe preprint as far as I searched.

So, this might not have publisher.

Published on arXiv in March 2019.

# BackGround : About ALE

- Arcade Learning Environment (ALE) is based on Atari 2600



# BackGround : About ALE

- ALE is widely used for evaluating reinforcement learning (RL) agents for following main reasons:
  1. Diversity:
    - ... There are many kinds of games, allowing researchers to test to broad applicability of their algorithms.
  2. Diverse challenges:
    - ... Each game presents unique exploration challenges. So it offers complex tasks that RL agents need to learn.
  3. Natural curriculum:
    - ... The difficulty of the games gradually increases as time is passes or as the agent improves. When they master certain tasks, they receive new and more challenging task.

# BackGround : About ALE

- ALE has mainly two challenges: the representation learning problem and the behavioural learning problem.
  1. The representation learning problem.
    - ... The challenge of extracting meaningful information from raw pixel inputs.  
Extracting meaningful information require considerable computation resource. ( also take a lot of time )
  2. The behavioural learning problem.
    - ... The challenge of learning by understanding complex and delayed associations between actions and rewards.

# BackGround : AIM

- They tried to create the new environment, called MinAtar, to overcome ALE's limitations and enable more efficient and reproducible RL experiments.
- It tries to enable more reproducible and more thorough experiments.
  1. Reproducibility
    - ... Whether a certain experiment can be reproduced by other researchers.
  2. More thorough experiments
    - ... To evaluate a certain algorithm in variety of conditions.
- So, a major its goal is to reduce the complexity of the representation learning problem while maintaining the mechanics of ALE as much as possible.

## Related work

- RAM representation: Simplifies representation, but loses spatial structure.
- DeepMind control suite (built upon the MuJoCo):  
These two have continuous action-spaces, comparing discrete in ALE.
- The Pygame Learning Environment (PLE):  
The visual complexity is similar to ALE.
- Talvitie: Reduce representation, but lose spatial structure.

-> These are little different from MinAtar work.

# Method : MinAtar

	Abstract	
	MinAtar	ALE
Reduced spatial demension	10*10 grid.	64*64 grid.
Reduced action space	6 actions.	18 actions.
Simplified rewards	1 or 0 (there is an exception.)	more than 100.
Meaningful input pixels	each pixel has meaning, (ball, enemy, etc...)	raw color channels.
Reduced partial observability	moving direction can be understood from even 1 frame.	sometime 1 frame do not represent moving direction.
Simplified game mechanics	omitted some game mechanics.	
Added stochasticity		

# Method : MinAtar

- In MinAtar, all environment maintain the main mechanics of ALE and reduce their representation complexity.
- There are five environments:
  1. Asterix : Collect treasures while avoiding enemies.
  2. Breakout : Destroy blocks by bouncing a ball against them.
  3. Freeway : Cross the road while avoiding cars.
  4. Seaquest : Rescue divers while managing oxygen and avoiding enemies.
  5. Space Invaders : Defeat arians.



# Method : MinAtar

- Asterix : Collect treasures while avoiding enemies.



# Method : MinAtar

- Asterix : Collect treasures while avoiding enemies.



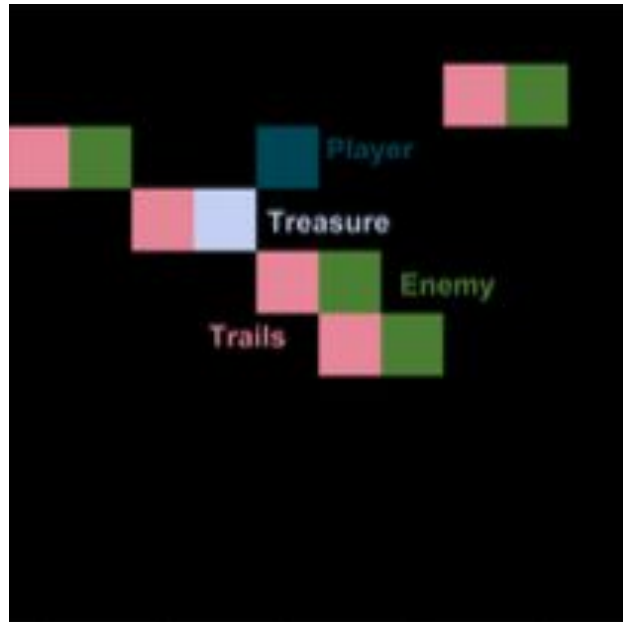
# Method : MinAtar

- Asterix : Collect treasures while avoiding enemies.








# Method : MinAtar

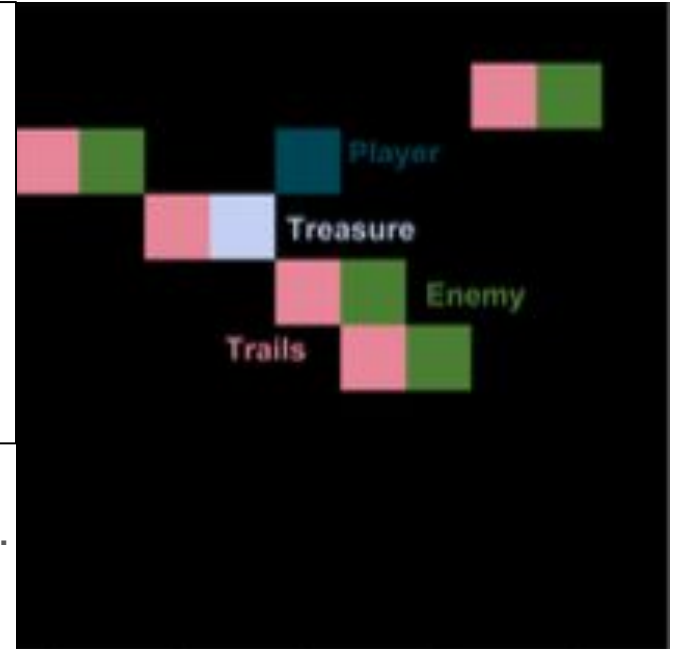
- Asterix : Collect treasures while avoiding enemies.



# Method : MinAtar

- Asterix : Collect treasures while avoiding enemies.
- Action : 
- Rewards :  
+ 1 per picking up treasure.
- Termination :  
When player hit to enemy
- Difficulty :  
Periodically increased in speed and spawn rate.

 : Trails  
( For Direction )  
 : Enemy  
 : Player  
 : Treasure



# Method : MinAtar

- Breakout : Destroy blocks by bouncing a ball against them.



# Method : MinAtar

- Breakout : Destroy blocks by bouncing a ball against them.



# Method : MinAtar

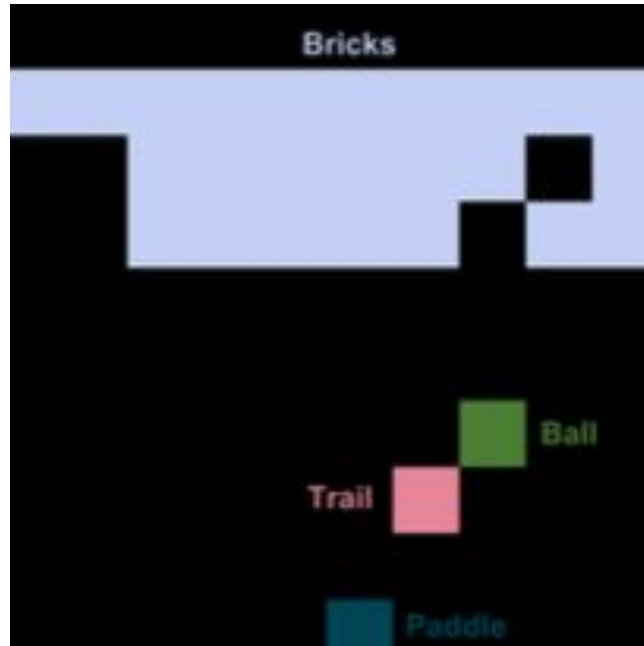
- Breakout : Destroy blocks by bouncing a ball against them.










# Method : MinAtar

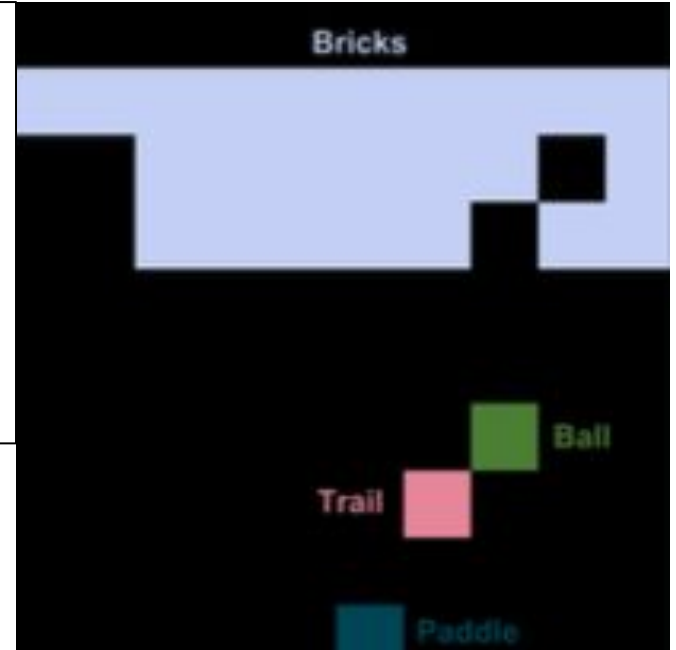
- Breakout : Destroy blocks by bouncing a ball against them.



# Method : MinAtar

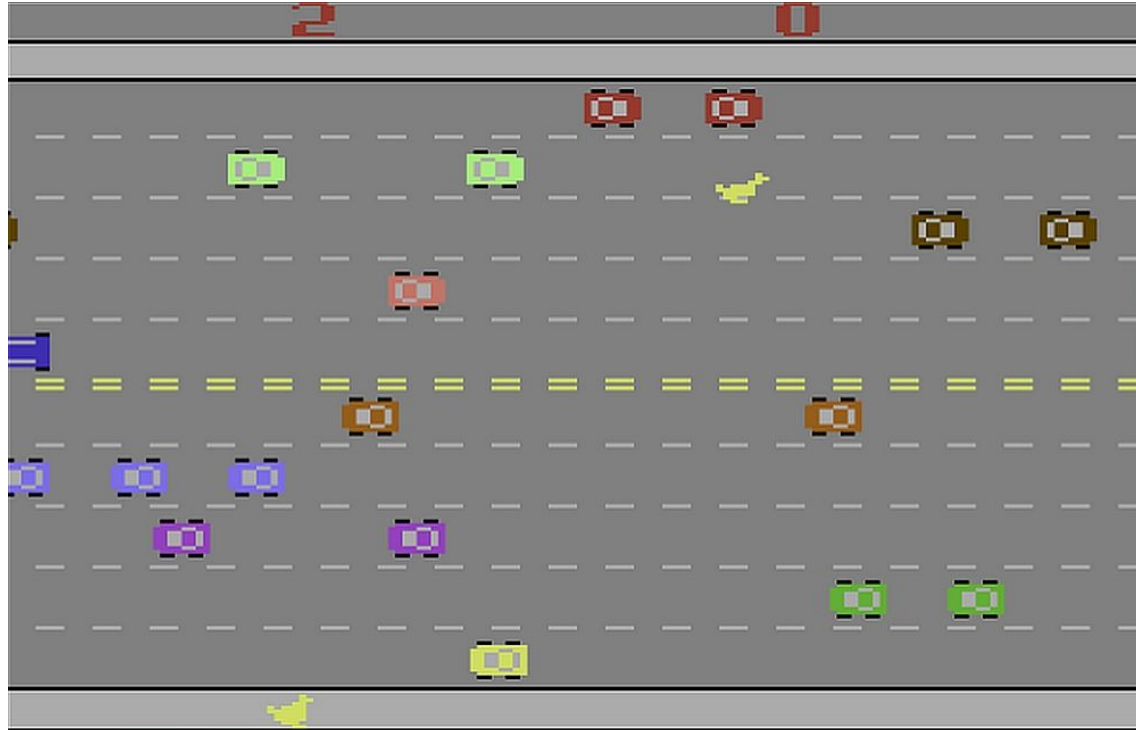
- Breakout : Destroy blocks by bouncing a ball against them.
- Actions : 
- Rewards :  
+1 per each brick broken.
- Termination :  
The ball hits the bottom.
- When all bricks are cleared, another 3 rows are added.

 : Trails  
( For Direction )  
 : Ball  
 : Paddle  
 : Bricks



# Method : MinAtar

- Freeway : Cross the road while avoiding cars.



# Method : MinAtar

- Freeway : Cross the road while avoiding cars.

5 \* 5



# Method : MinAtar

- Freeway : Cross the road while avoiding cars.

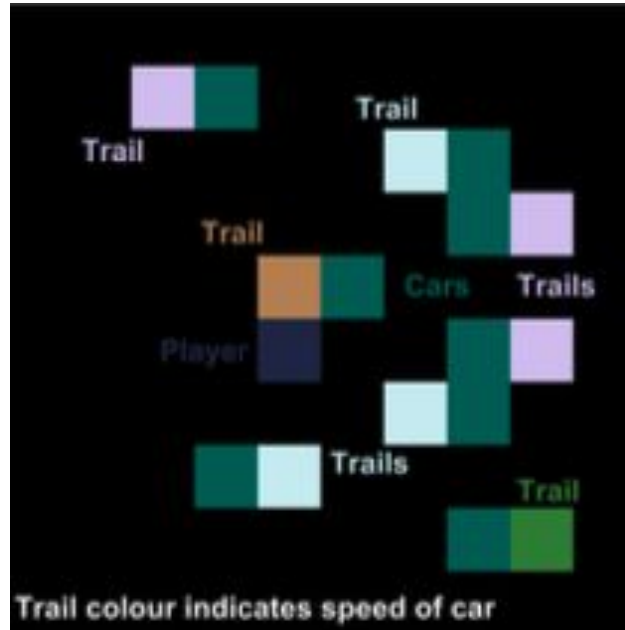
4 \* 4




# Method : MinAtar

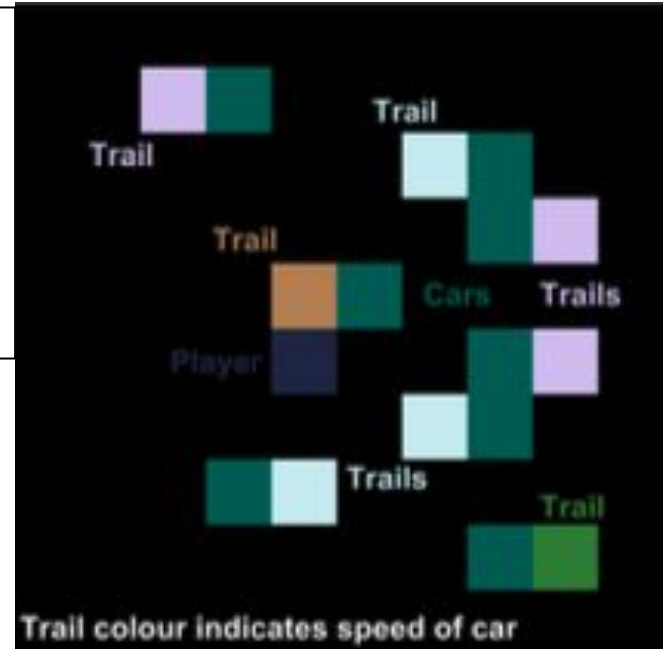
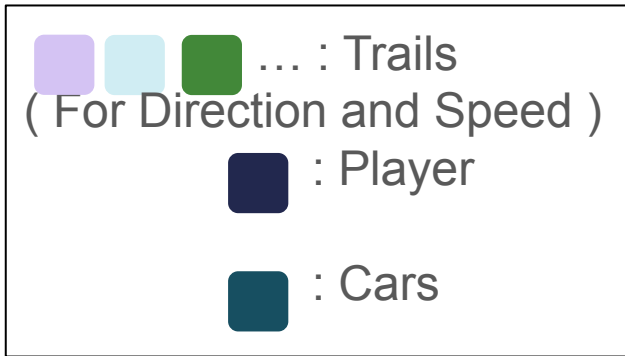
- Freeway : Cross the road while avoiding cars.

4 \* 4



# Method : MinAtar

- Freeway : Cross the road while avoiding cars.
- Actions : 
- Rewards :  
+1 per each reaching the goal.
- Termination :  
After 2500 frames have elapsed.
- For trails, there are 5 color that means speed.



# Method : MinAtar

- Seaquest : Rescue divers while managing oxygen and avoiding enemies.





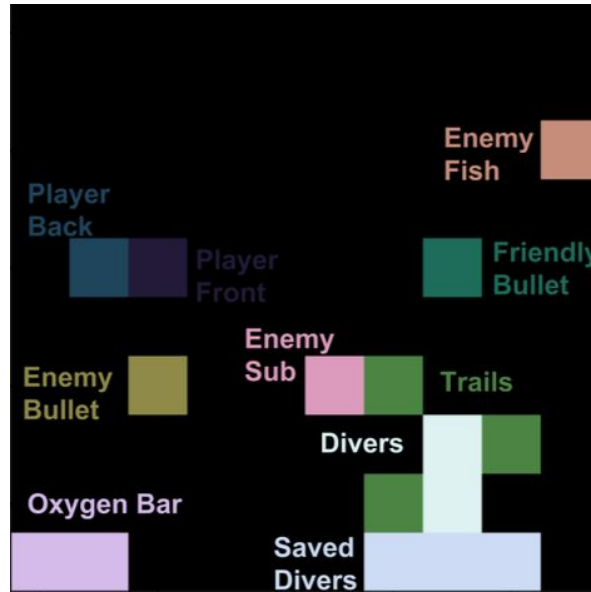
# Method : MinAtar

- Seaquest : Rescue divers while managing oxygen and avoiding enemies.




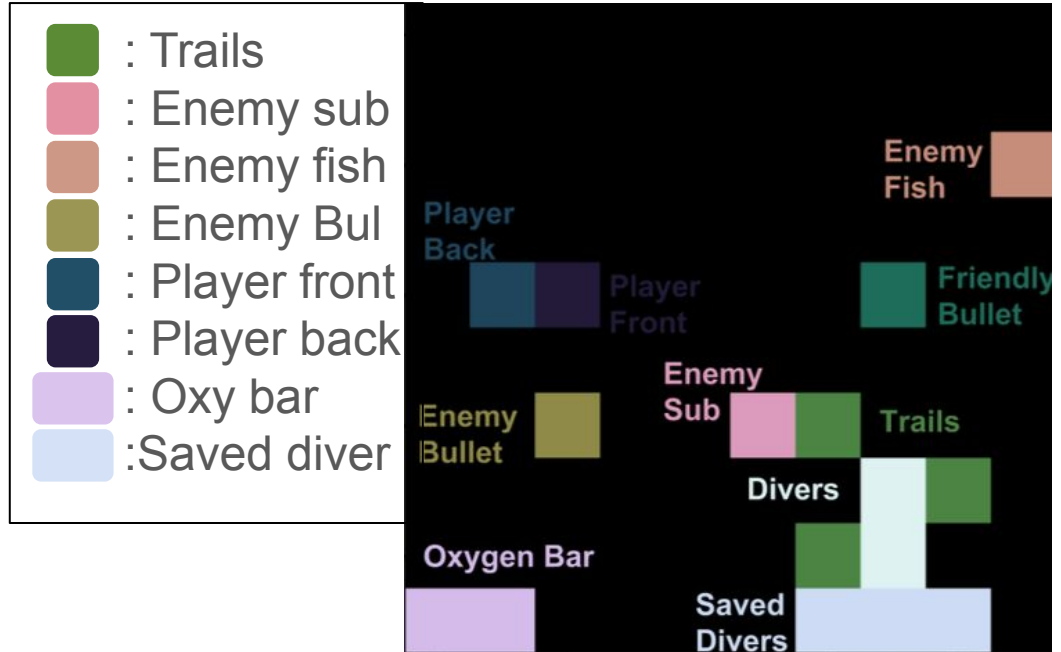
# Method : MinAtar

- Seaquest : Rescue divers while managing oxygen and avoiding enemies.



# Method : MinAtar

- Seaquest : Rescue divers while managing oxygen and avoiding enemies.
- Actions : , and fire.
- Rewards :
  1. +1 per hit to enemy
  2. When the player recharges oxygen in specific timing based on the amount.
- Termination :
  1. Hit to enemy or bullet.
  2. run out oxygen.



# Method : MinAtar

- Space Invaders : Defeat arians.



# Method : MinAtar

- Space Invaders : Defeat arians.



# Method : MinAtar

- Space Invaders : Defeat arians.



# Method : MinAtar

- Space Invaders : Defeat arians.



## Method : MinAtar

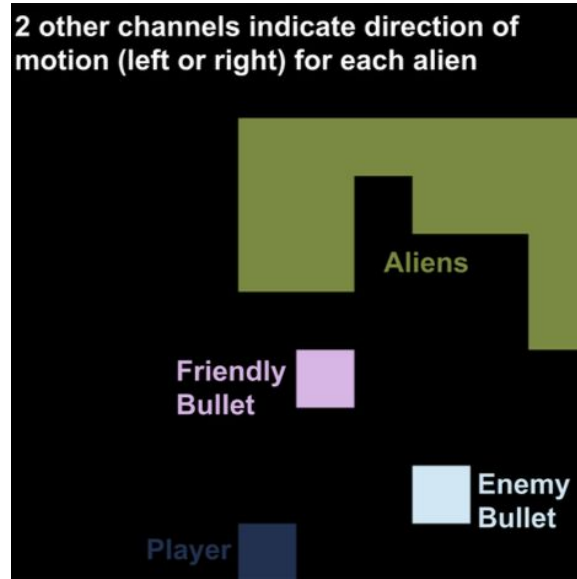
- Space Invaders : Defeat arians.






# Method : MinAtar

- Space Invaders : Defeat arians.




# Method : MinAtar

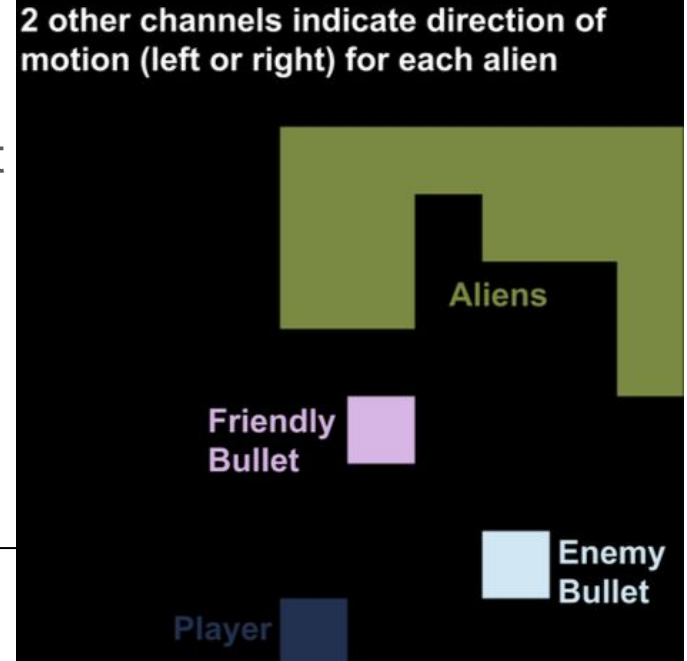
- Space Invader : Destroy aliens.
- Actions :  , fire
- Rewards :  
+1 per each part of aliens broken.
- Termination :  
The player hit to enemy or bullet.

 : Player Bullet

 : Enemy Bullet

 : Player

 : Aliens  
( Color of it means the way of aliens)



# Method : MinAtar, Important factors

1. About Seaquest, Asterix, and Space Invaders:  
The difficulty of game improve as each certain game event occurs.
2. About the game except of Breakout:  
These are partially observable. It means that some information are not encoded in the state, including the timing of object movement and the current difficulty level.
3. About Seaquest and freeway:  
These require high difficulty level of exploration.  
For example, the timing of recharging oxygen is difficult.

# Experiments

- Environments : 5 (as I explained before).
- Algorithms : Totally 4 (DQN, DQN-sub, AC( $\lambda$ ), AC( $\lambda$ )-sub)\*  
\* DQN: Deep Q-Network, AC: Actor-Critic with Eligibility Traces  $\lambda$
- Random Seeds : 30
- Training Frames : 5,000,000 compared to 50,000,000 in the conventional approaches in ALE.

# Experiments: About DQN

- Architecture:
  - A convolutional layer: 16 3\*3 kernels, and stride is 1.
  - A fully connected hidden layer: 128 units (neurons). It is one-quarter of original DQN.
- Hyper Parameter:
  - Replay buffer size: For save experience.
  - Target network update frequency:
  - $\epsilon$ -decay time:  $\epsilon$  means exploration rate.  
It means the decay rate of random action probability.
  - Recharging time of replay buffer: Number of frames required to initialize the replay buffer.
  - All of these parameters are one tenth of original DQN in this research.
  - To decide  $\alpha$ , the parameter of step size, they tried to use much variety of value as following:  
 $\alpha = 1, \frac{1}{2}, \frac{1}{4}, \dots$  (it is shown and explained in the next slide)
- No frame skipping
- For comparison, they also use the DQN without replay buffer.

# Experiments: About DQN

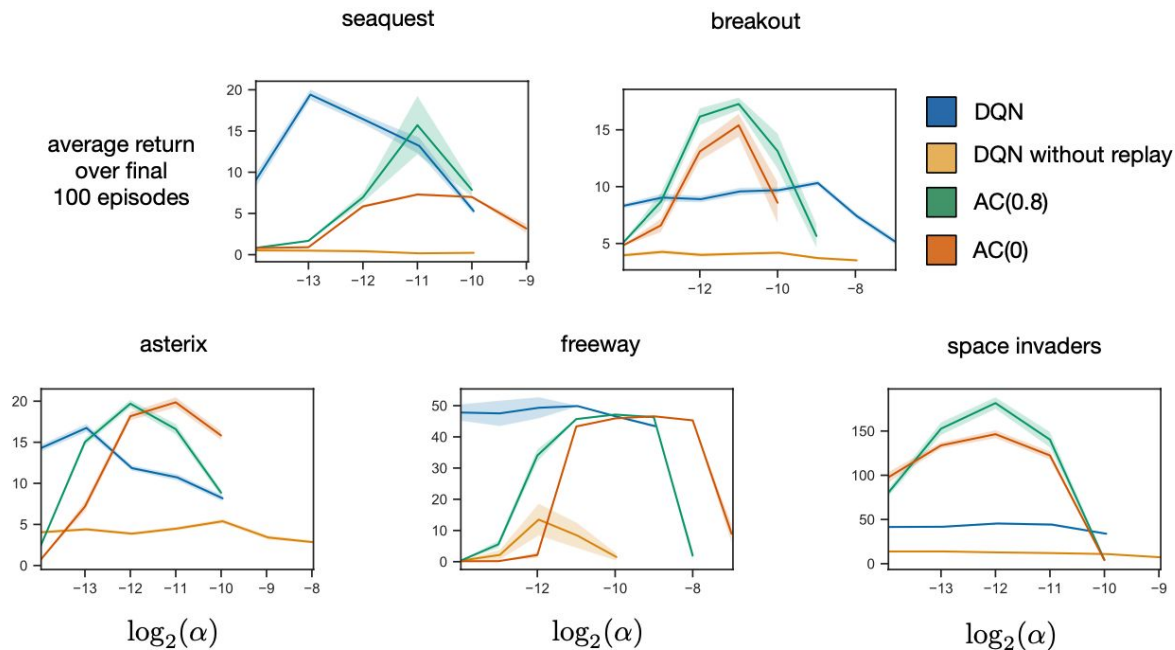


Figure 2: Average return over the final 100 episodes v.s.  $\log_2(\alpha)$  for all agents. All curves are averaged over 30 runs.

# Experiments: About $AC(\lambda)$

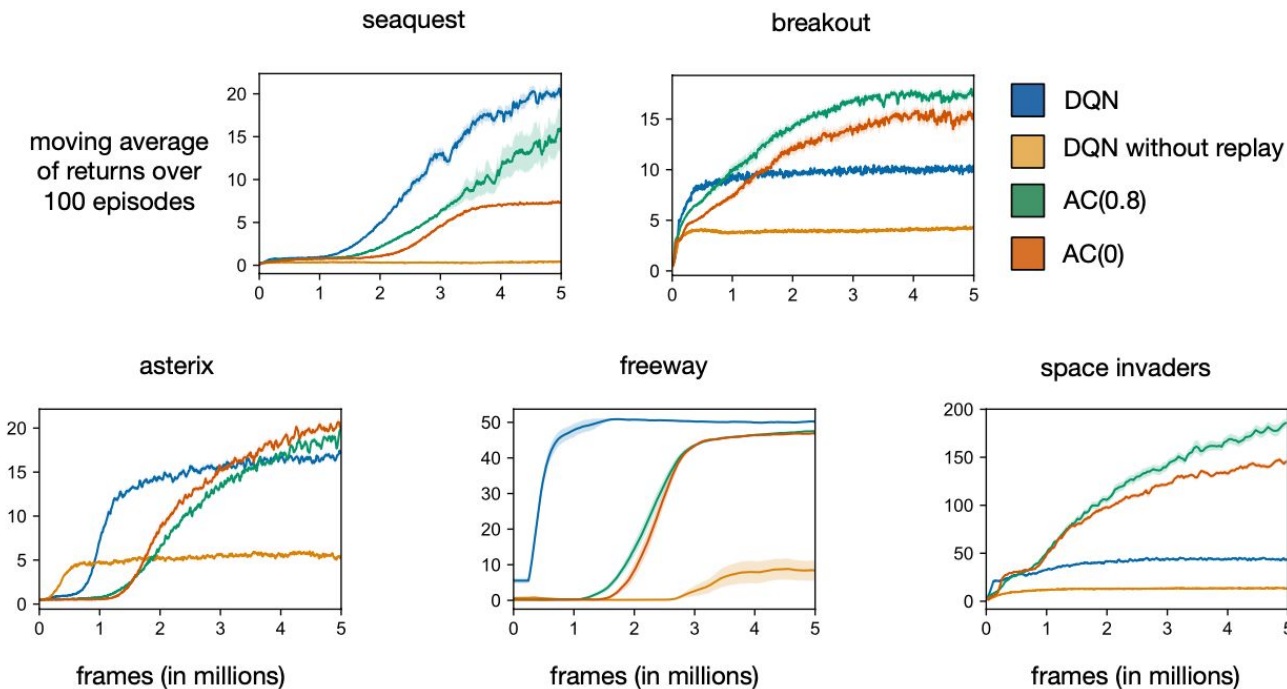
- Architecture:
  - $\lambda$  ( $0 \sim 1$ ): trace decay parameter, the parameter that adjust effects of past gradient information.
  - Activation function: They used SiLU (Sigmoid-weighted Linear Unit) and dSiLU (SiLU with derivative) instead of ReLU (Rectified Linear Unit).  
These two are known as the function which is efficient in ALE research.
  - Optimizer: RMSProp is used for stabilization learning.  
They adjust the smoothing factor from 0.95 to 0.999.  
It is for determines how much influence to retain from past gradients.  
Also, they set the minimum square gradient to 0.0001 for stabilization learning.
- They used two patterns of  $\lambda$ , 0.8, and 0.
  - $\lambda = 0$ : AC update actions from only current information.
  - $\lambda = 0.8$ : AC update actions from current and past information.

# Results

- In DQN training with Seaquest, the total time per a frame is 8 millisecond in single CPU, comparing 5 millisecond in GPU.
  - > It means CPU can be feasible instead of GPU and decreased cost of computer resource.



# Results



In initial:  
DQN > AC( $\lambda$ )

In the long run:  
AC( $\lambda$ ) > DQN

The benefit of replay

The benefit of  $\lambda$

Figure 3: Average return v.s. training frames for all games with the optimal alpha value for each agents. All points are the average of 30 runs with error bars showing standard error in the mean.

# Results

- MinAtar demonstrated some qualitative interesting behaviours as following:

Environment	abstract
Breakout	Agents learned to clear a one side column of the bricks in order to trap the ball above. (It is important technique to achieve reward efficiently)
Seaquest	Some agents going up for air when oxygen was low. (They learned one of difficult exploring challenge)
Seaquest	Agents Maintained a horizontally centered position on the screen. This behaviour maximizes the time to respond to threats since enemies emerge from the sides of the screen.

# Results

- Generally, MinAtar can indicate strength and weakness of algorithms. So, it makes analysis easier.
- Also, with MinAtar researchers can conduct experiment with relatively low cost for computer resources. Thanks to it, they can try experiment in much variety of seeds and parameter. So they can obtain reliable results.

# Conclusions

- MinAtar is a novel evaluation platform, which has higher reproducibility and enable thorough experiment.
- As a future work, they are going to add some difficult environment from ALE, such as Montezuma's Revenge, Pitfall, and so on.
- Also, they would like to evaluate another method such as A2C, Double DQN, in MinAtar.

**Thank you for listening!**