Playing FPS Games with Deep Reinforcement Learning

Guillaume Lample,∗ Devendra Singh Chaplot∗

https://ojs.aaai.org/index.php/AAAI/article/view/10827

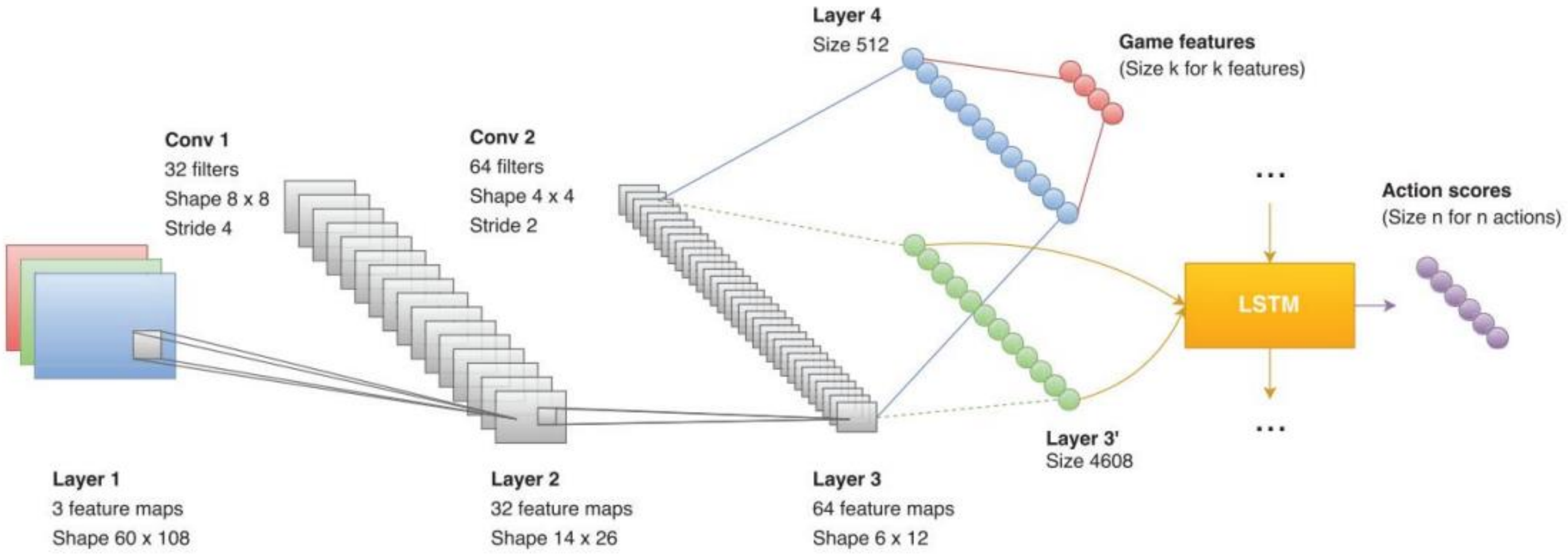
Introduction

- Almost all game AI is for 2D environments, but introduced AI in this paper is for 3D environment.
- Especially they talk about AI for FPS game.
- Their model can play as well as human in deathmatch scenario.

Difficulties

- 3D game AI is much more challenging than most 2D games, such as navigating through a map, collecting items, recognizing and fighting enemies etc.
- So they divides AI into two parts: navigation and action.
- Navigation is exploring the map to collect items and find enemies.
- Action is fighting enemies when they are observed.
- 3D games rarely return complete observations.
- However, the current screen is sufficient to infer the course of action.

Deep Recurrent Q-Networks (DRQN)



Deep Recurrent Q-Networks (DRQN)

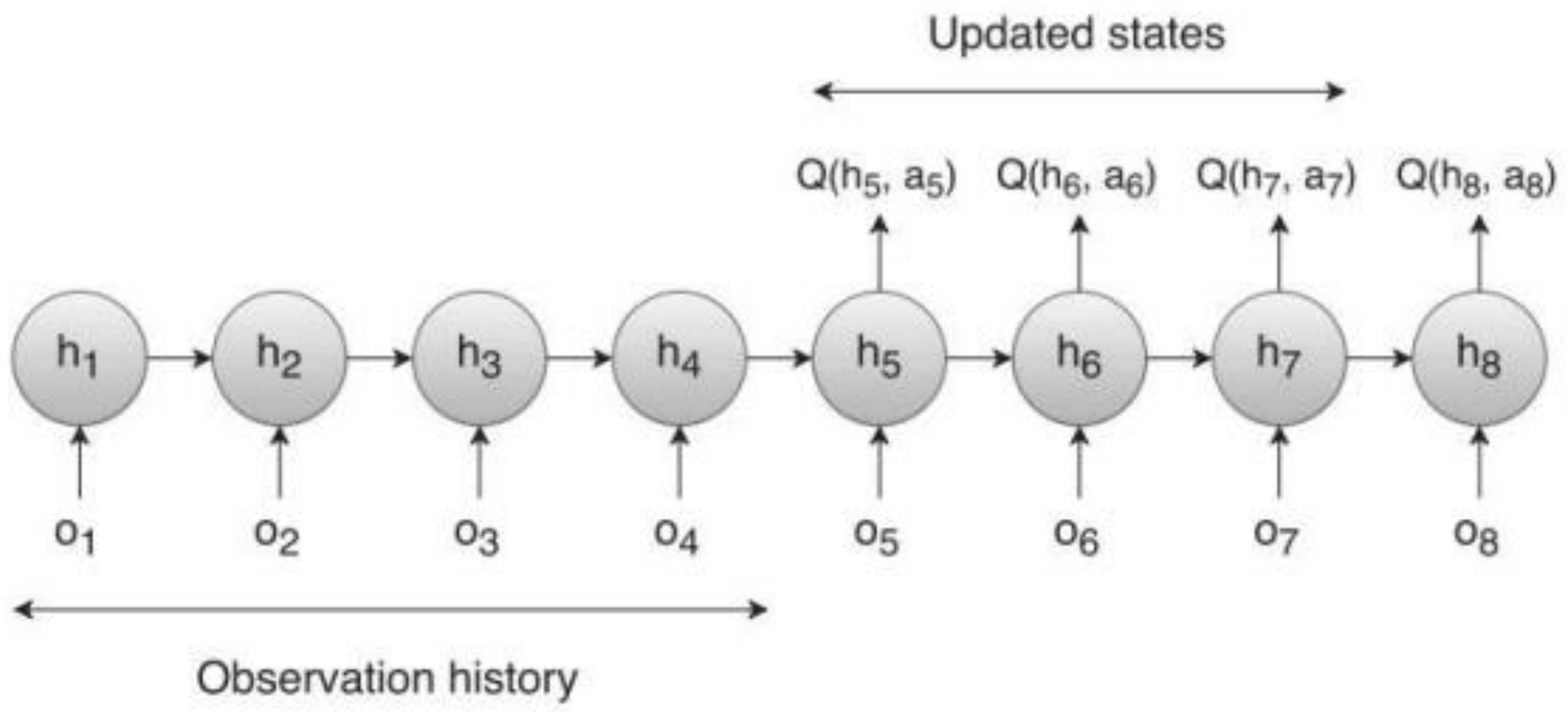
- The input image is given to two convolutional layers.
- The output of the convolutional layers is split into two streams.
- The first bottom flattens the output and feeds it to a LSTM, as in the DRQN model.
- The second one top projects it to an extra hidden layer.
- During the training, the game and Q-learning objectives are trained jointly.

Results

- Although this model achieved good performance in relatively simple scenarios.

- But giving a penalty for using ammo did not help: with a small penalty.

Divide and conquer



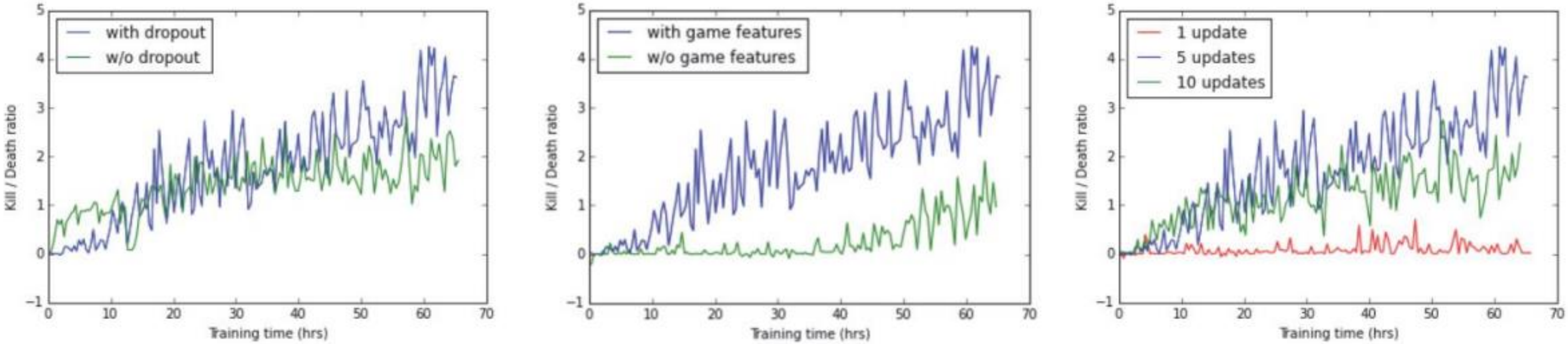
Divide and conquer

- There are advantages of splitting the task into two phases and training a different network for each phase.
- First, this makes the architecture modular and allows different models to be trained and tested independently for each phase.
- More importantly, using two networks also mitigates "camper" behaviour. (stay in one area of the map and wait for enemies)

Reward shaping

- Rewards based solely on score are likely to slow learning.
- Therefore, we will set an intermediate reward.
- Negative reward for losing health.
- Negative reward for losing ammunition.
- Positive rewards for item acquisition.

Plot of K/D score



Left: with and without dropout
Middle: with and without game features
Right: with different of updates in the LSTM

Results

	Limited Deathmatch		Full Deathmatch			
	Known Map		Train maps		Test maps	
Evaluation Metric	Without navigation	With navigation	Without navigation	With navigation	Without navigation	With navigation
Number of objects	14	46	52.9	92.2	62.3	94.7
Number of kills	167	138	43.0	66.8	32.0	43.0
Number of deaths	36	25	15.2	14.6	10.0	6.0
Number of suicides	15	10	1.7	3.1	0.3	1.3
Kill to Death Ratio	4.64	5.52	2.83	4.58	3.12	6.94

Results and Analyze

- K/D scores are much better when using training with map navigation
- Setting weapons weaker increases the benefit of picking up items, this was a good decision

Conclusion

- These method introduced in this paper lead to dramatic improvements over the standard DRQN model when applied to complicated tasks like a deathmatch.
- The proposed model is able to outperform built-in bots as well as human players and demonstrated the generalizability of our model to unknown maps.