DEEP REINFORCEMENT LEARNING FOR PLAYING 2.5D FIGHTING GAMES

Yu-Jhe Li[*1], *Hsin-Yu Chang*[*2], *Yu-Jing Lin*[2], *Po-Wei Wu*[2], *and Yu-Chiang Frank Wang*[1]

[1] Graduate Institute of Communication Engineering, National Taiwan University
[2] Department of Computer Science & Information Engineering, National Taiwan University

ABSTRACT

Deep reinforcement learning has shown its success in game playing. However, 2.5D fighting games would be a challenging task to handle due to ambiguity in visual appearances like height or depth of the characters. Moreover, actions in such games typically involve particular sequential action orders, which also makes the network design very difficult. Based on the network of Asynchronous Advantage Actor-Critic (A3C), we create an OpenAI-gym-like gaming environment with the game of Little Fighter 2 (LF2), and present a novel A3C+ network for learning RL agents. The introduced model includes a Recurrent Info network, which utilizes game-related info features with recurrent layers to observe combo skills for fighting. In the experiments, we consider LF2 in different settings, which successfully demonstrates the use of our proposed model for learning 2.5D fighting games.

Index Terms— Deep reinforcement learning, 2.5D, game

1. INTRODUCTION

Deep reinforcement learning (RL) has been widely utilized in recent topics and applications in computer vision, robotics, and machine learning. One of the successful examples is the Atari 2600 games. More specifically, Schaul et al. [1] and Nair et al. [2] advance Deep Q Network (DQN) for learning Atari games. Recently, Schulman et al. [3] and Gu et al. [4] advance policy gradient to improve the training efficacy. Some other works [5, 6, 7, 8] have also made great success in fine-tuning models for deep reinforcement learning.

With impressive progress on the learning of Atari games, games in more complex forms are further explored by the researchers. For example, Wu and Tian [9] apply different deep RL models to learn first-person shooting games, and Tessler et al. [10] utilize transfer learning for lifelong learning problems in Minecraft. Also, Silver et at. [11] use the deep neural network to learn GO and defeat the world-class players. Leibo et al. [12] further investigate the behaviors of the learned agents, and apply DQN for train multi-agent games. Some non-2D works contain a trial of 3D first player shooting (FPS) game **Doom** [13]. Lample et al. [14] designed their new model to successfully train the agent in 3D games. Besides, Vinyals et al. [15] also extended their work on playing the non-2D strategy game: **StartCraft 2**, and their improved method in training 3D game is also impressive.

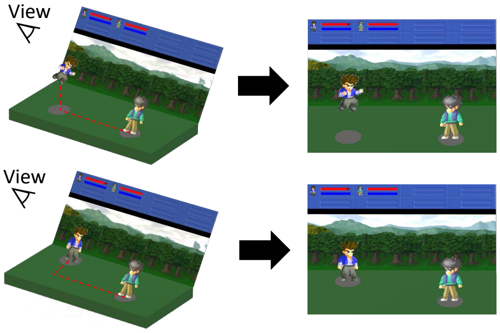


Fig. 1: With orthographic projection, objects in varying distances in 2.5D games would exhibit the same size, which makes the learning of such games very challenging. Note that the left character in the upper image is jumping at a closer distance, while that in the lower one is at a farther distance.

Nevertheless, existing deep RL models are typically designed for handling games in 2D or limitedly in 3D. Extension of such learning models for games beyond 2D would not be trivial. In this paper, we focus on learning 2.5D fighting games, in which characters are manipulated in 3D mode while they are displayed in 2D. We note that, in most 2.5D games, 3D objects in a scene are orthographically projected to the 2D screen. Thus, the size of the object in a farther distance would still be similar to that of a closer one. Fig. 1 shows example frames from a 2.5D fighting game: Little Fighter 2 (LF2) [16]. We see that, the ambiguity between distance and height brings difficulties in training agents. Thus, utilizing the locations of the characters and memorizing their temporal states become vital for RL training. Moreover, characters in LF2 are with different combinations of skills, which make the learning even more challenging. As a result, direct uses of existing RL models would not be applicable.

In this paper, we present a novel RL architecture based on the Asynchronous Advantage Actor-Critic (A3C) network [17], and focus on learning the 2.5D fighting game of LF2. In addition to input screen frames, our proposed network takes a variety of information as input features. Together with the long short-term memory (LSTM) components, this allows us to exploit information across and beyond input screen frames, while the aforementioned ambiguity and challenges can be properly addressed. We refer to as proposed network as A3C+, and our main contributions are summarized as follows:

[*] Indicate equal contribution.

978-1-4799-7061-2/18/$31.00 ©2018 IEEE

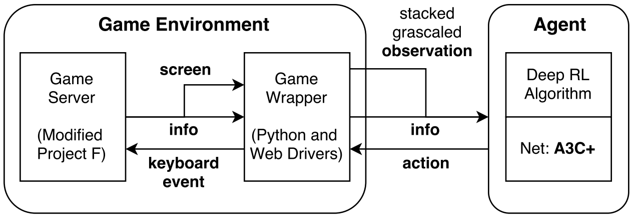


Fig. 2: Our framework describing the game environment and the agent. The agent interacts with the game server through the game wrapper, which makes allows the agents to take actions and obtain rewards from the environment.

- We build an OpenAI-gym-like 2.5D game environment in which we successfully wrap the game server for agent learning and interaction.
- Based on A3C, we propose an A3C+ which takes not only screen frames but also game-related features in a recurrent learning architecture.
- We show that our proposed architecture favors 2.5D games, supported by quantitative and qualitative evaluation results.

2. ENVIRONMENT SET-UP FOR 2.5D FIGHTING GAME: LITTLE FIGHTER 2 (LF2)

A major contribution of our work is the environment set-up for the freeware PC 2.5D fighting game of LF2 [16]. As depicted in Fig. 2, this process consists of three parts: game server, wrapper, and the agent. We apply the open-source Project F [18], which is a web-based implementation of LF2, as our game server. Next, we wrap this game into an OpenAI-gym-like environment [19] *env* with Python and web drivers (PhantomJS). The set-up details can be found at [20].

2.1. Observations for RL

In order to learn the proposed RL-based network architecture, an array of 4-stacked, gray-scaled successive screenshots is given as the observed state to the agent at each step, which reveals the motion information of the agents in a scene. In order to perform this process efficiently, we apply the frame-skipping technique of [21], with skip frame parameter k set and fixed as 4. To be more precise, the agent learned by our model would take an action at every four successive frames.

2.2. Action Space and Rewards

The action space of LF2 comprises 8 basic actions and 8 combo skills. While *idle, up, right, down, left, attack, jump,* and *defend* constitute the basic action, combo skills are represented by sequences of basic actions. The role *Firen*, for example, makes explosion by taking *defend, up,* and *jump* successively. We note that, a set combo skills with 8 different combinations are defined as [*defend* + *up/right/down/left* + *attack/jump*], while different kinds of combo skills are available for different characters.

In 2.5D fighting games, the goal is to defeat the adversary character in terms of the remaining health points (HP) of the two characters. We define the reward as such differences. Thus, a positive reward is given by any damage to the opponent character, while any injury to our agent results in a negative reward. Thus, our agent would learn through the training process by taking proper actions to beat the opponent.

3. OUR PROPOSED MODEL

3.1. Brief Review of Asynchronous Advantage Actor-Critic (A3C)

Since our proposed deep RL architecture is based on A3C [17], we now briefly review this state-of-the-art RL model for the sake of completeness. For deep RL, the agent is learned via a deep neural network with parameter θ, which defines a policy π_θ and an estimate of the value function V_θ. The agent receives observation s_t at each time step t, selects an action a_t with probability $\pi_\theta(a_t|s_t)$, and then receives a reward r_t from the game environment. The goal of the RL agent is to maximize the return $R_t = \sum_{k=0}^{\infty} \gamma^k r_{t+k+1}$, where γ is a discount factor. The parameters of the policy and the value are learned using Asynchronous Advantage Actor-Critic (A3C) described by Mnih et al. [17]. A3C is a policy gradient method with an approximate gradient ascent on $\mathbf{E}[R_t]$. The update can be seen as $\bigtriangledown_{\theta'} log\pi_\theta(a_t|s_t;\theta)A(a_t,s_t;\theta)$, in which $A(a_t,s_t;\theta)$ is an estimate of advantage function given by $R_t - V(s_t;\theta)$, where $V(s_t;\theta)$ is a value function estimate of the expected return $\mathbf{E}[R_t]$. Based on the recent models of [17, 22], an additional entropy regularization is applied, with the complete version of the objective function as $\bigtriangledown_{\theta'} log\pi_\theta(a_t|s_t;\theta)(R_t - V(s_t;\theta)) + \beta \bigtriangledown_{\theta'} \pi_\theta(s_t;\theta^{'})log\pi_\theta(s_t;\theta^{'})$. We note that, the hyperparameter β controls the strength of the entropy regularization term.

3.2. Our A3C+ for Learning 2.5D Fighting Games

With A3C applied to basic 2D games like Atari, how to reflect features vital for 2.5D games (e.g., character location) is still a challenging task. In our proposed network of **A3C+**, which extends A3C and includes a novel **Recurrent Info network** (i.e., recurrent layers with introduced 2.5D game-related features) as illustrated in Fig. 3. We now discuss the A3C+ components in the following subsections.

3.2.1. LSTM for recurrent learning

Previously, Hausknecht and Stone [5] presented a deep recurrent Q network, which improves DQN with recurrent layers for observing and learning temporal information. Similar ideas have been utilized in standard A3C [17], showing

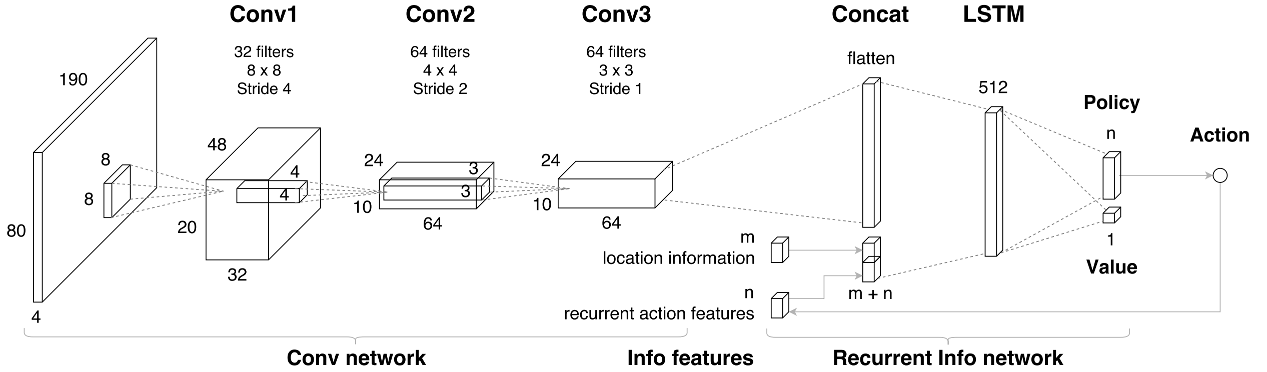


Fig. 3: Our network of A3C+. Note that the introduced recurrent info network takes 2.5D game-related and fighting-action related features, aiming to better describe game environments and to allow our agent for winning the fighting game.

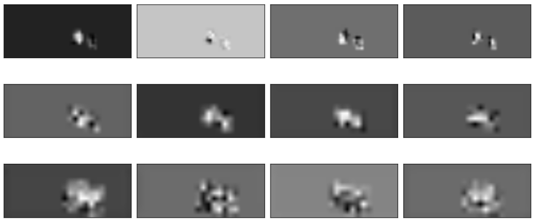


Fig. 4: Example visualization of convolution layers in Conv network for LF2 using standard A3C (top to bottom: Conv1, Conv2, Conv3 respectively).

preferable results in RL tasks. In view of the above success, we also advance recurrent layers in our A3C+. More specifically, we utilize the LSTM with the observation of the features related to 2.5D games (see Sect. 3.2.2).

3.2.2. Info features for 2.5D games

As noted earlier, standard A3C cannot easily describe game-related features like identification, actions, or height/distance of characters (see Fig. 4 as example). To solve these problems, we take the **location information** of both our agent and the opponent into consideration. To be more precise, we extract the coordinates of both characters from the game environment, as well as their heights, depths, the distance between them. These representations are concatenated with the CNN visual feature, which would be the inputs to the LSTM module as depicted in Fig. 3. We believe that, when learning complicated 2.5D fighting games, it is vital to allow the design network to observe the above game-related features.

3.2.3. Actions of the agents in info features

In fighting games like LF2, effective hits on the opponents involve sequential orders actions (e.g., an agent needs to jump to the back of the opponent and then punch the opponent immediately to gain the highest reward). To better train our agent using the proposed network, we particularly consider **recurrent action features** as parts of the inputs to the recurrent info network. With such joint feature observation and the use of LSTMs, we aim at learning proper 2.5D game-related and action-order related features, which would allow our agent to survive longer and/or defeat the opponent.

4. EXPERIMENTS

4.1. Implementation Details and Settings

As mentioned earlier, we consider Little Fighter 2 (LF2) as the 2.5D fighting game in our work. In LF2, each character has different skills in terms of offense. For evaluation simplicity, we consider the role of *Davis*, which is the main character of this game, as our training agent. To confirm the effectiveness of the proposed **Recurrent Info network** in our A3C+ network, we take the **Conv network** [17] in the standard A3C as the baseline network. And we perform ablation tests to verify the contributions of the proposed network components (i.e., uses of info features and recurrent layers of LSTM).

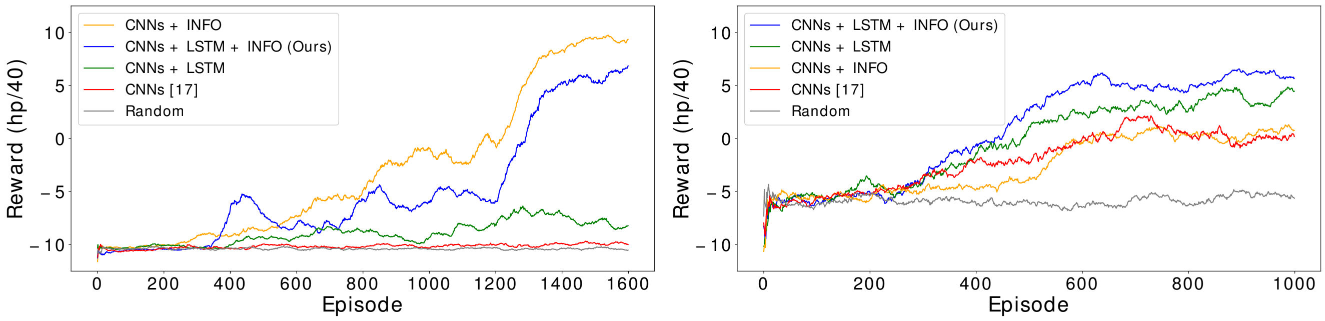
In our evaluation, two different settings are considered:

- Basic: Only 8 basic actions allowed (see Sect. 2.2).
- Advanced: 8 actions in the basic setting, plus 8 combo skills without constraints on the magic points (MP).

We note that, for the advanced setting, each LF2 agent has a total of 8 combo skills. When utilizing our A3C+ for such fighting games, we wish to verify that the agent would not only learn the basic actions, it would learn the correlation between them and produce combo skills as well. As for the opponent in the game, we consider a rule-based AI. That is, it is always able to use all 16 actions in either of the two settings.

4.2. Evaluation

We start with health points (HP) as 500 for both our agent and the opponent, with reward scaled to $1/40$ of the HP differences. In other words, the highest reward our agent can achieve per episode is between -12.5 and +12.5. To make the



(a)

(b)

Fig. 5: Average rewards over episodes when training A3C models in (a) basic and (b) advanced settings.

evaluation comparisons more complete, we also consider a baby-agent, which simply takes a random action at each step and thus can be viewed as a naive baseline model.

4.2.1. *Training stage*

Fig. 5 shows the rewards observed by different models using basic and advanced settings. In Fig. 5 (a), our agent was only trained to take 8 basic actions, and no combination skills were allowed. Thus, the use of info features in our A3C+ was sufficient to achieve promising rewards, when comparing to the use of our full model (i.e., info features plus LSTM recurrent layers). More importantly, advancing recurrent layers or not, models without observing info features would not be able to produce satisfactory results. This confirms that, the use of 2.5D game-related features would be beneficial in learning 2.5D fighting games, and our proposed network architecture with such features would be preferable.

On the other hand, Fig. 5 (b) shows the agent training in the advanced setting. From this figure, we observe that the use of both info feature and the LSTM recurrent layers would be preferable, since this full version of our A3C produced the highest rewards compared to other baseline models or controlled experiments. This is because that, the combo skills may receive rewards after several screen frames, and thus delayed rewards are more likely to be observed and taken into consideration for the actions in the past. The network models with only info features but without the recurrent layer can only observe the information from the previous action, which would not be sufficient produce preferable actions or combo skills. In other words, when combination skills are allowed, it would be preferable to learn actions in proper sequential orders so that it becomes more likely to defeat the opponent.

4.2.2. *Testing stage*

In addition to the comparisons of the observed rewards during training, we further present the evaluation of testing. To be more specific, we present the winning rate (WR) of the learned agent, based on trained models for 1000 episodes. The results are listed in Table 1. Recall that the opponent in our experiment is a rule-based AI taking both basic actions and combo skills during fighting. From Table 1, we see that the use of the full version of our A3C+ achieved the most satisfactory WR. For the basic setting (i.e., only basic actions allows without combo skills), the use of recurrent layers did not exploit additional sequential action information, and thus the use of info features would be sufficiently satisfactory (i.e., WR as 99.3%). From the above experiments, the effectiveness of our A3C+ for learning 2.5D fighting games can be successfully verified.

Table 1: Winning rate (WR) of the deep RL agents trained in different settings.

	Model Setting		Game Setting	
Agent	LSTM	INFO	Basic	Advanced
Ours	✓	✓	95.5%	**95.9%**
Ours w/o LSTM		✓	**99.3%**	61.3%
CNNs w/ LSTM	✓		11.9%	86.3%
CNNs [17]			7.3%	57.6%
Random			4.9%	10.6%

5. CONCLUSION

As the first to advance deep reinforcement learning for 2.5D fighting games, our proposed network extends Asynchronous Advantage Actor-Critic (A3C) and exploits game-related info features with recurrent layers. This not only alleviates the problem of visual ambiguity of the characters during fighting, it also allows us to observe proper action orders for producing combo action skills. With our set-up of game environment and the game of LF2, our experiments confirmed that the use of our game-related info features is crucial in 2.5D fighting games, while the full version of ours (i.e., with the proposed recurrent info networks) produces the best winning rate and performs favorably against recent or baseline deep reinforcement learning models.

Acknowledgments This work was supported by the Ministry of Science and Technology of Taiwan under grant MOST 107-2634-F-002-010.

6. REFERENCES

[1] Tom Schaul, John Quan, Ioannis Antonoglou, and David Silver, "Prioritized experience replay," *arXiv preprint arXiv:1511.05952*, 2015.

[2] Arun Nair, Praveen Srinivasan, Sam Blackwell, Cagdas Alcicek, Rory Fearon, Alessandro De Maria, Vedavyas Panneershelvam, Mustafa Suleyman, Charles Beattie, Stig Petersen, et al., "Massively parallel methods for deep reinforcement learning," *arXiv preprint arXiv:1507.04296*, 2015.

[3] John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov, "Proximal policy optimization algorithms," *arXiv preprint arXiv:1707.06347*, 2017.

[4] Shixiang Gu, Timothy Lillicrap, Zoubin Ghahramani, Richard E Turner, and Sergey Levine, "Q-prop: Sample-efficient policy gradient with an off-policy critic," *arXiv preprint arXiv:1611.02247*, 2016.

[5] Matthew Hausknecht and Peter Stone, "Deep recurrent q-learning for partially observable mdps," *CoRR, abs/1507.06527*, 2015.

[6] Emanuel Todorov, Tom Erez, and Yuval Tassa, "Mujoco: A physics engine for model-based control," in *Intelligent Robots and Systems (IROS), 2012 IEEE/RSJ International Conference on*. IEEE, 2012, pp. 5026–5033.

[7] Paul Christiano, Jan Leike, Tom B Brown, Miljan Martic, Shane Legg, and Dario Amodei, "Deep reinforcement learning from human preferences," *arXiv preprint arXiv:1706.03741*, 2017.

[8] Tejas D Kulkarni, Karthik Narasimhan, Ardavan Saeedi, and Josh Tenenbaum, "Hierarchical deep reinforcement learning: Integrating temporal abstraction and intrinsic motivation," in *Advances in Neural Information Processing Systems*, 2016, pp. 3675–3683.

[9] Yuxin Wu and Yuandong Tian, "Training agent for first-person shooter game with actor-critic curriculum learning," 2016.

[10] Chen Tessler, Shahar Givony, Tom Zahavy, Daniel J Mankowitz, and Shie Mannor, "A deep hierarchical approach to lifelong learning in minecraft.," in *AAAI*, 2017, pp. 1553–1561.

[11] David Silver, Aja Huang, Chris J Maddison, Arthur Guez, Laurent Sifre, George Van Den Driessche, Julian Schrittwieser, Ioannis Antonoglou, Veda Panneershelvam, Marc Lanctot, et al., "Mastering the game of go with deep neural networks and tree search," *Nature*, vol. 529, no. 7587, pp. 484–489, 2016.

[12] Joel Z Leibo, Vinicius Zambaldi, Marc Lanctot, Janusz Marecki, and Thore Graepel, "Multi-agent reinforcement learning in sequential social dilemmas," in *Proceedings of the 16th Conference on Autonomous Agents and MultiAgent Systems*. International Foundation for Autonomous Agents and Multiagent Systems, 2017, pp. 464–473.

[13] Michał Kempka, Marek Wydmuch, Grzegorz Runc, Jakub Toczek, and Wojciech Jaśkowski, "Vizdoom: A doom-based ai research platform for visual reinforcement learning," in *Computational Intelligence and Games (CIG), 2016 IEEE Conference on*. IEEE, 2016, pp. 1–8.

[14] Guillaume Lample and Devendra Singh Chaplot, "Playing fps games with deep reinforcement learning," pp. 2140–2146, 2017.

[15] Oriol Vinyals, Timo Ewalds, Sergey Bartunov, Petko Georgiev, Alexander Sasha Vezhnevets, Michelle Yeo, Alireza Makhzani, Heinrich Küttler, John Agapiou, Julian Schrittwieser, et al., "Starcraft ii: a new challenge for reinforcement learning," *arXiv preprint arXiv:1708.04782*, 2017.

[16] Little Fighter 2, "http://www.littlefighter.com," 1999-2008.

[17] Volodymyr Mnih, Adria Puigdomenech Badia, Mehdi Mirza, Alex Graves, Timothy Lillicrap, Tim Harley, David Silver, and Koray Kavukcuoglu, "Asynchronous methods for deep reinforcement learning," in *International Conference on Machine Learning*, 2016, pp. 1928–1937.

[18] F.LF Project, "http://www.projectf.hk," 2011-2015.

[19] Greg Brockman, Vicki Cheung, Ludwig Pettersson, Jonas Schneider, John Schulman, Jie Tang, and Wojciech Zaremba, "Openai gym," *arXiv preprint arXiv:1606.01540*, 2016.

[20] Lf2gym, "https://github.com/elvisyjlin/lf2gym," 2018.

[21] Marc G Bellemare, Yavar Naddaf, Joel Veness, and Michael Bowling, "The arcade learning environment: An evaluation platform for general agents.," 2013.

[22] Ronald J Williams and Jing Peng, "An efficient gradient-based algorithm for on-line training of recurrent network trajectories," *Neural computation*, vol. 2, no. 4, pp. 490–501, 1990.