Tencent AI Lab

Mastering Complex Control in MOBA Games with Deep Reinforcement Learning

Deheng Ye, Zhao Liu, Mingfei Sun, Bei Shi, Peilin Zhao, Hao Wu, Hongsheng Yu, Shaojie Yang, Xipeng Wu, Qingwei Guo, Qiaobo Chen, Yinyuting Yin, Hao Zhang, Tengfei Shi, Liang Wang, Qiang Fu, Wei Yang, Lanxiao Huang

Tencent AI Lab

2020.01.02

Tencent AI Lab

Action control of heroes

"Honor of Kings" tested
(Chinese: 王者荣耀)

Mastering $\boxed{\text{Complex Control}}$ in $\boxed{\text{MOBA Games}}$ with $\boxed{\text{Deep Reinforcement Learning}}$

The method
system-level & algorithm-level

Paper link: https://arxiv.org/abs/1912.09729

Overview

Tencent AI Lab

- Introduction
- Method
  - System
  - Algorithm
- Experiments
- Conclusion & future work

Overview

Tencent AI Lab

- Introduction
- Method
  - System
  - Algorithm
- Experiments
- Conclusion & future work

Introduction

Tencent AI Lab

- Recent development in Game AI

2016.03 | 2019.02 | 2017 - 2019 | 2017 - present

DeepMind: AlphaGo

DeepMind： AlphaStar

OpenAI
Five

OpenAI：Dota2

Tencent：JueWu (V1, SL based, 2018.12)

Tencent：JueWu (V2, RL based, 2019.08)

- Tencent AI Lab – Game AI Center

Go (2016 - present) | **MOBA (2017 - present)** | **3D-FPS (2018 - present)**

Introduction

Tencent AI Lab

- MOBA 1v1 games
  - Two-agent, one vs. another
  - Many game units
    - Turrets, creeps, heroes, etc.
  - Pure arena for competing one's ability of **action control** (micro-management)
    - 5v5 games focus more on team strategy

Honor of Kings Game UI Illustration

Introduction

Tencent AI Lab

- The game is complex
  - Enormous action space
  - Enormous state space
  - Real time
  - Playing method
    - Complicated action control
    - Vary from hero to hero
  - Target selection
    - Hard to decide which game unit(s) to attack/protect
  - Little high-quality human data
    - 1v1 mainly for practicing heroes, while 5v5 as formal matches
    - Supervised learning infeasible

Table 1: Comparing Go and MOBA 1v1

Game	**Go 1v1**	**MOBA 1v1**
Action space	$250^{150} \approx 10^{360}$ (250 pos available, 150 decisions per game on average)	10^{18000} (100+ discretized actions, 9,000 frames per game)
State space	$3^{361} \approx 10^{170}$ (361 pos, 3 states each)	$2^{2000} \approx 10^{600}$ (2 heroes, (1000+ pos)*(2+ states))
Human player data	rich, high-quality	little
Peculiarity	long-term tactics	real-time, complex control

Introduction

- The game is complex
  - Enormous action space
  - Enormous state space
  - Real time
  - Playing method
    - Complicated action control
    - Vary from hero to hero
  - Target selection
    - Hard to decide which game unit(s) to attack/protect
  - Little high-quality human data
    - 1v1 mainly for practicing heroes, while 5v5 as formal matches
    - Supervised learning infeasible

Large-scale system for exploration

Unified modeling

Self-play

Overview

Tencent AI Lab

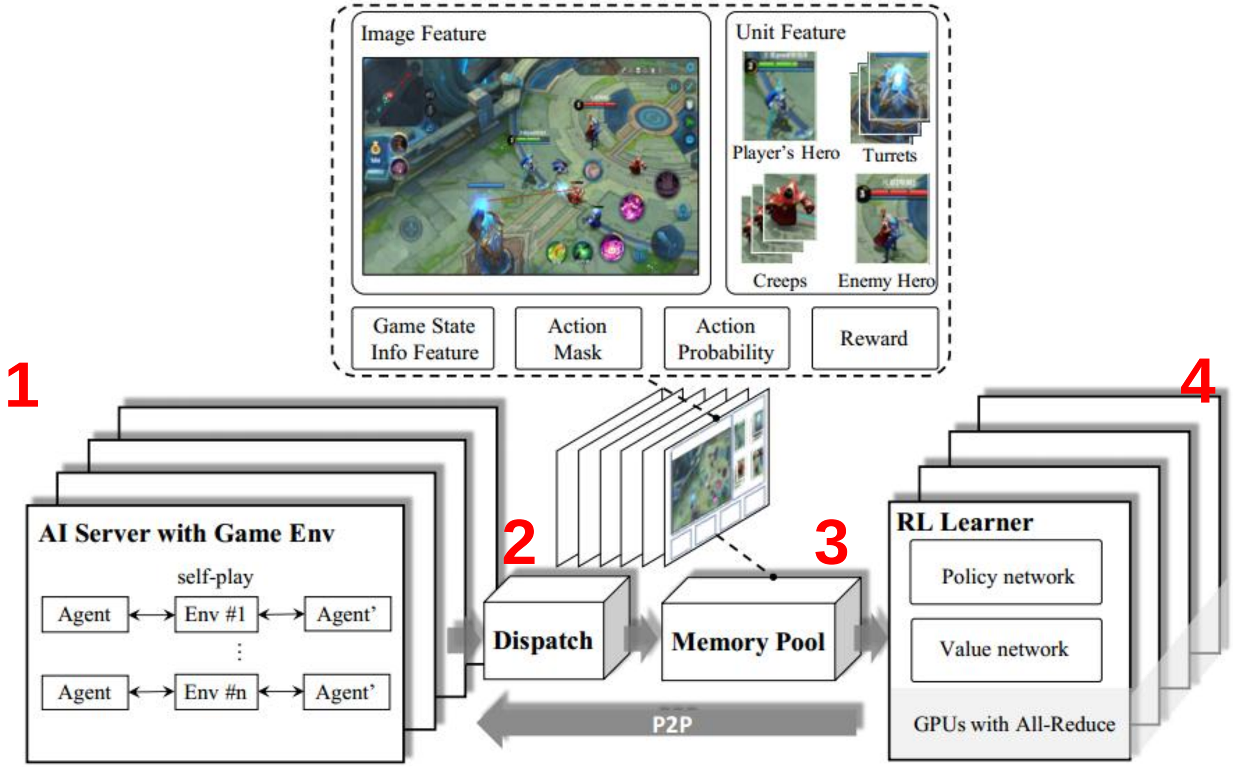
- Introduction
- Method
  - System
  - Algorithm
- Experiments
- Conclusion & future work

Method: overview

- Deep reinforcement learning system
  - Large-scale
  - Off-policy

- Algorithm
  - Multi-modal feature design
  - Actor-critic neural network
  - Multiple action control strategies
  - Dual-clip PPO

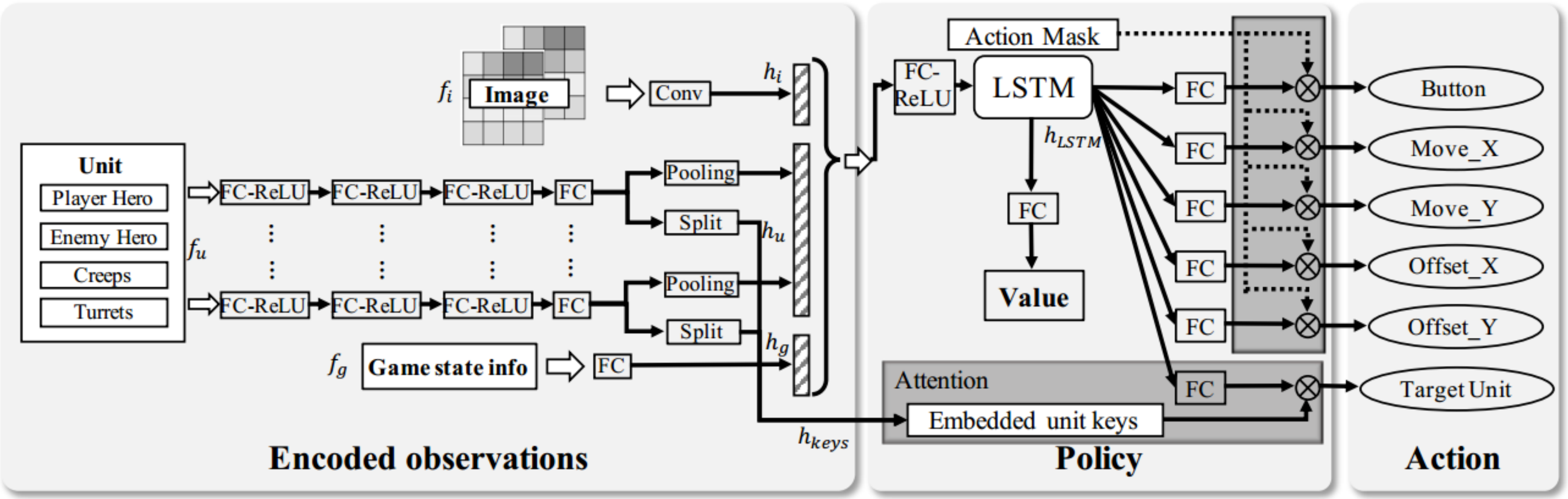
Method: system



- Large-scale
  - Support up to 1000+ GPU cards, 500,000+ CPUs tested in our Beta Environment
- Off-policy
  - Actor highly decoupled from Learner

- System architecture
  1. AI Server
     - Actor, where self-play happens
     - Interact with GameCore
  2. Dispatch Server
     - Data collect, compress & transmit
  3. Memory Pool
     - For data storage
     - Feed data to RL Learner
  4. RL Learner
     - For training reinforcement learning model
     - Model sysn to AI Server via P2P

Method: algorithm



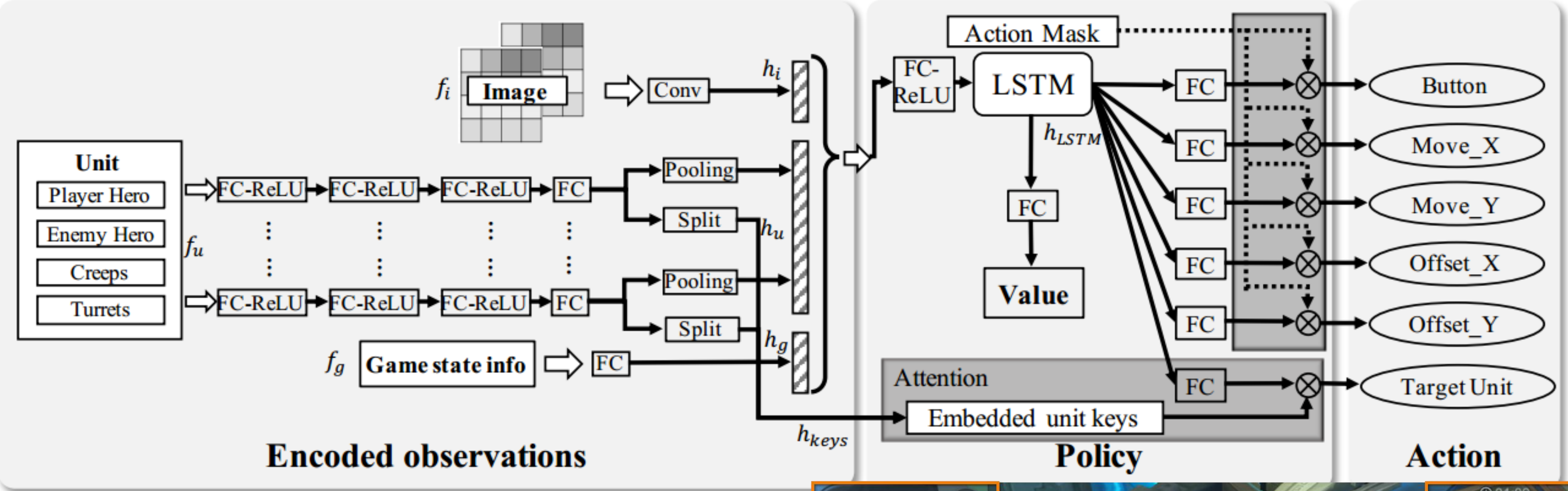
Input: observations/features

Internal: neural network model

Output: hero actions

Method: algorithm

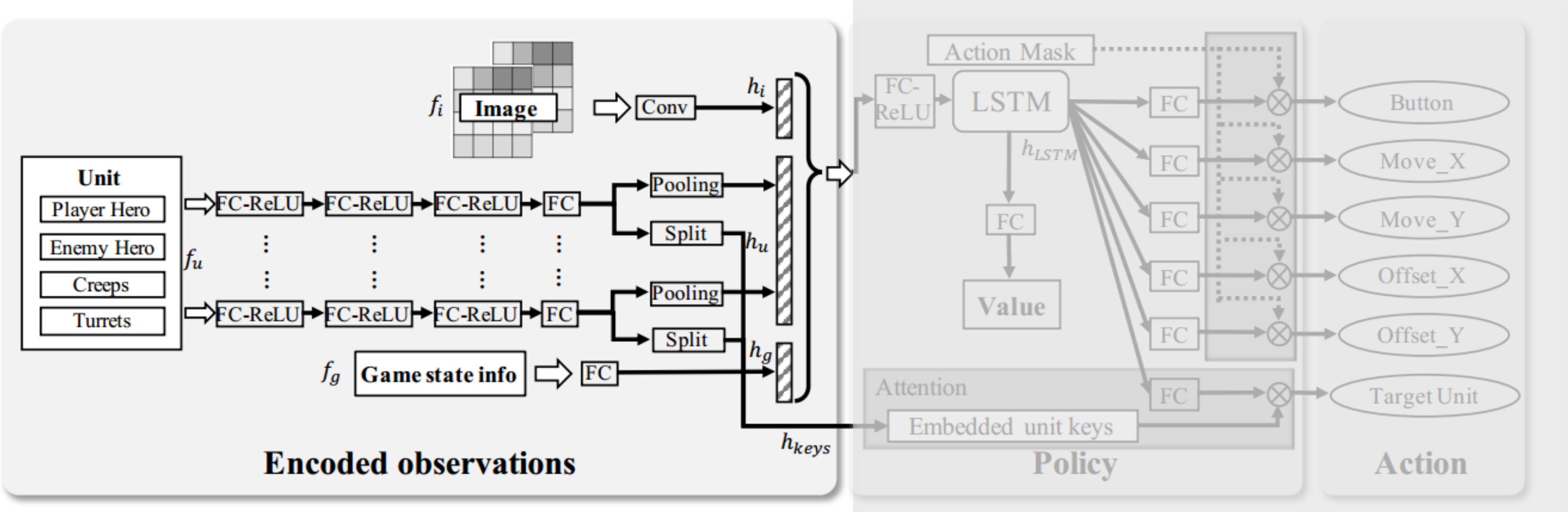
Tencent AI Lab



Input:

- Observable game unit attributes
  - Heroes, creeps, turrets, etc.
- Observable game states
- Local-view image-like channels

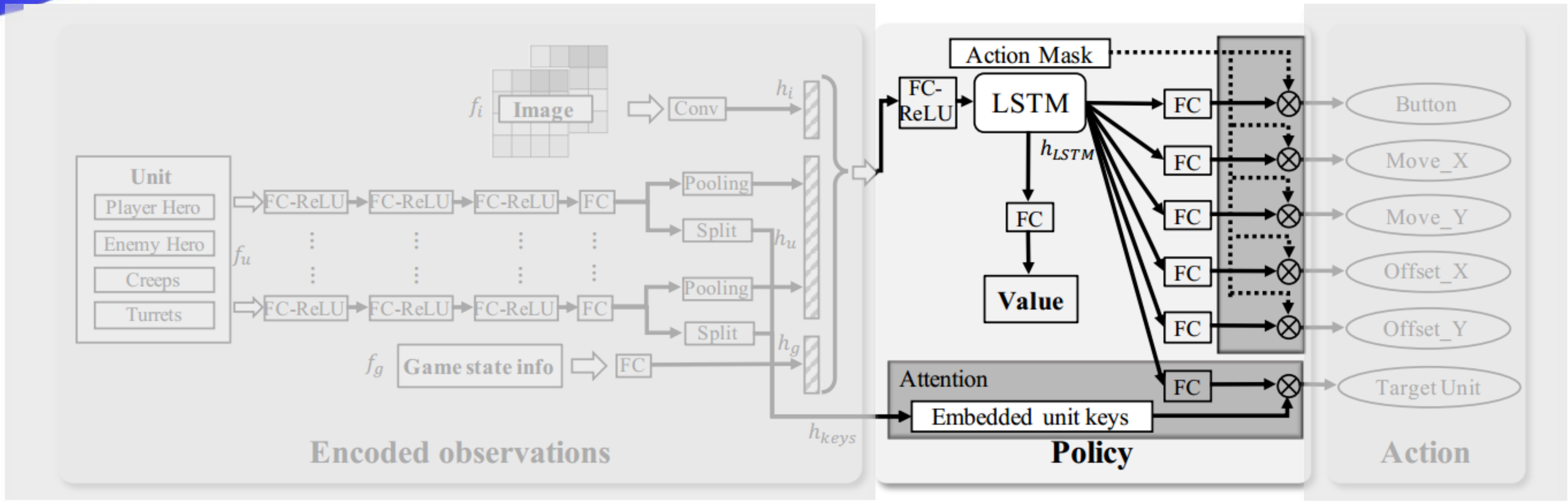
Method: algorithm



Internal:

- Feature/observation encoding
  - FC/ReLU layers, Conv layer, Pooling, Split
  - Weight sharing across same types of units

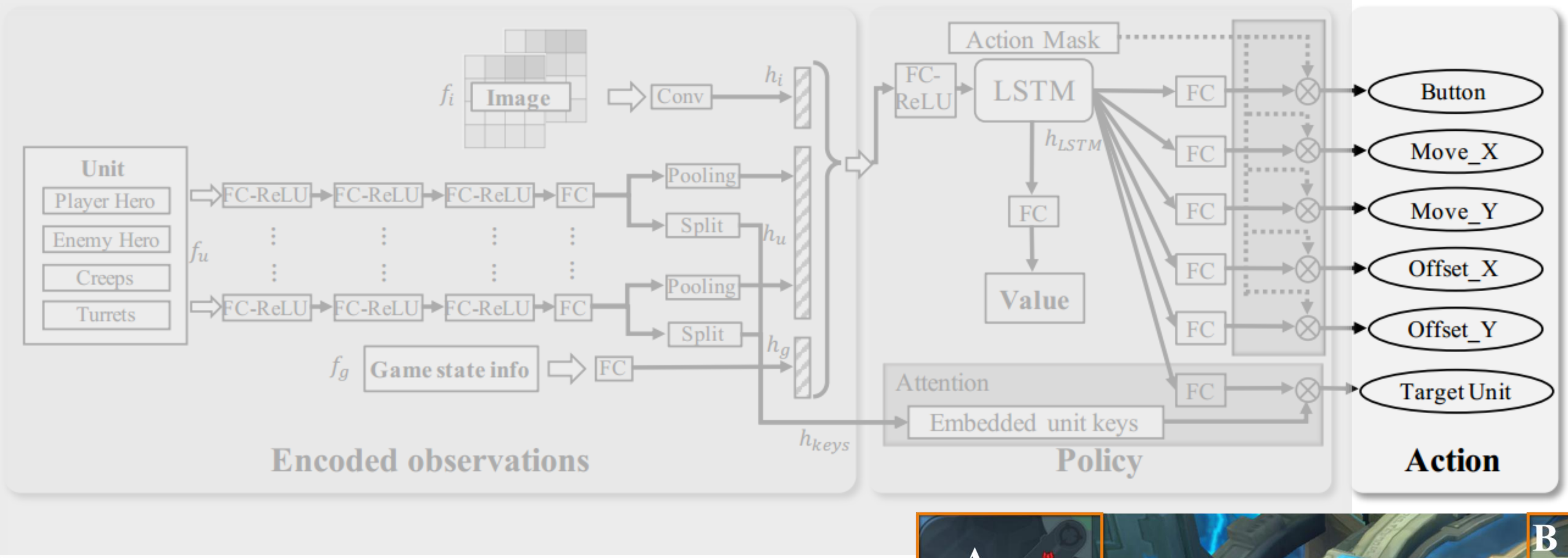
Method: algorithm



Internal:

- LSTM
- Action mask
  - For pruning RL exploration
- Target Attention
- Actor-critic network
  - Policy & value share parameter

Method: algorithm

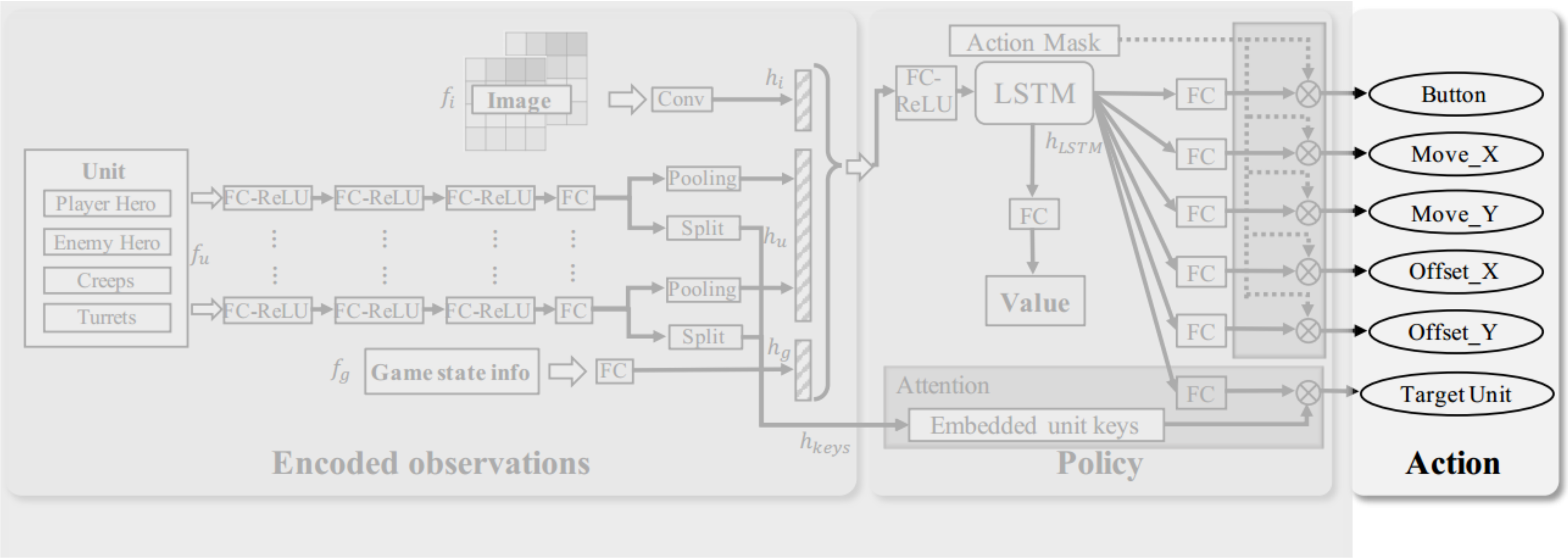


Output:

- Hierarchical, multi-label
  - First, predict **which action** to take, i.e., Button
    - E.g., move
  - Second, predict **how to execute** that action
    - E.g., the direction to move

  - What about label correlations?

Method: algorithm

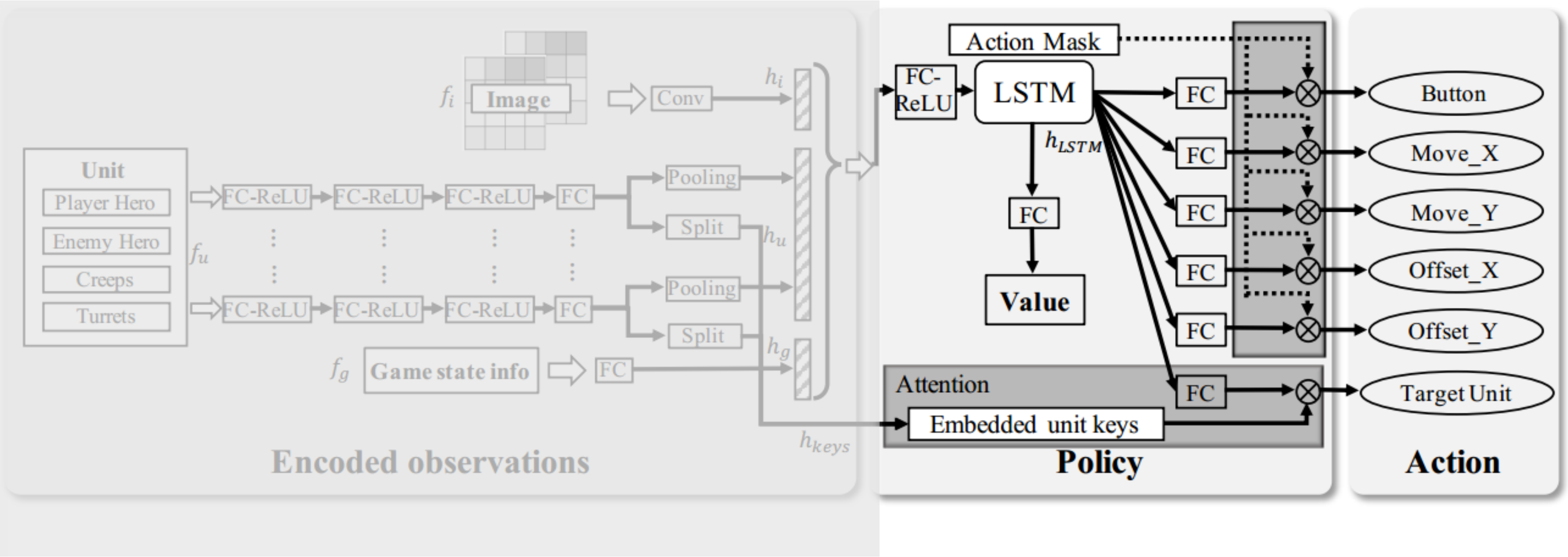


Output:

- Control dependency decoupling
  - Action **labels have correlations**, but are treated **independently**
    - To simplify episode sampling & objective optimization (See next slide)

Method: algorithm

Tencent AI Lab



Objective optimization

- Multi-label PPO (proximal policy optimization)

$$maximize_{\theta} \sum_{label\_i} E_{s_t, a_t \sim \pi_{\theta_{old}}} \left[\frac{\pi_{\theta}\left(a_t^{label\_i} | s_t\right)}{\pi_{\theta_{old}}\left(a_t^{label\_i} | s_t\right)} \left(R - V_{\theta_{old}}\left(s_t\right)\right) \right] - \frac{1}{2} E_{s_t \sim \pi_{\theta_{old}}} \left[\left(R - V_{\theta}\left(s_t\right)\right)^2 \right]$$

Label treated independently

Policy loss
We use PPO (to be continued)

Value loss

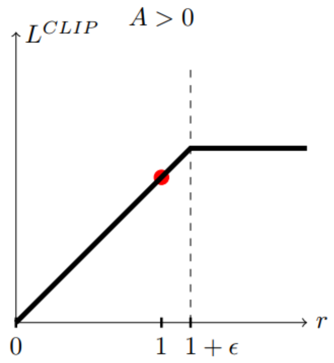
Method: algorithm

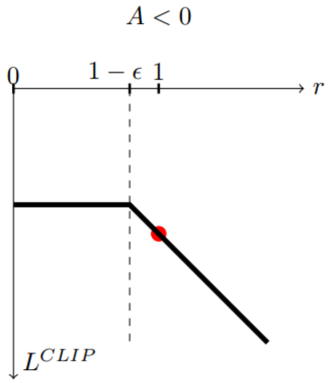
Objective optimization (continued)

Standard PPO [1]:

$$L^{clip}(\theta) = E_t\left[min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}\boxed{(R-V)}, clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)},1-\varepsilon,1+\varepsilon\right)(R-V)\right)\right]$$

A: Advantage





The problem:

- **large-scale & off-policy** setting → policy deviations

$$\text{when } \pi_\theta(a_t^{(i)}|s_t) \gg \pi_{\theta_{\text{old}}}(a_t^{(i)}|s_t) \text{ and } \hat{A}_t < 0 \implies \frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{\text{old}}}(a_t|s_t)}\hat{A}_t \ll 0$$

[1] Schulman et al. Proximal Policy Optimization Algorithms

Method: algorithm

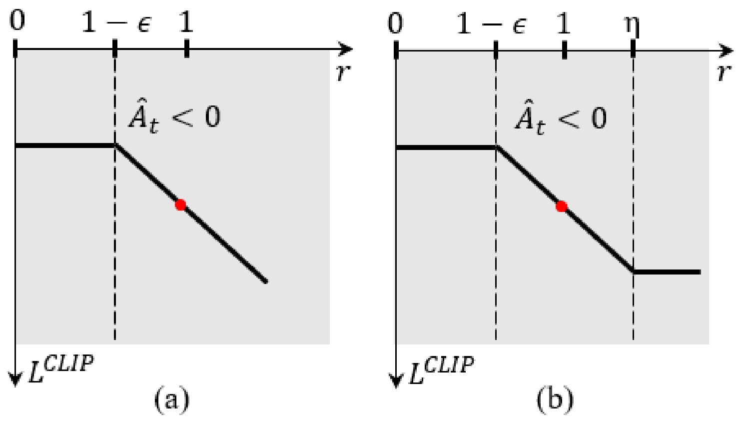
Objective optimization (continued)

Standard PPO:

$$L^{clip}(\theta) = E_t\left[min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}(R-V), clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon\right)(R-V)\right)\right]$$

Our proposed PPO: **dual-clip PPO**

$$L^{clip}(\theta) = E_t\left[max\left(min\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}(R-V), clip\left(\frac{\pi_\theta(a_t|s_t)}{\pi_{\theta_{old}}(a_t|s_t)}, 1-\varepsilon, 1+\varepsilon\right)\right), \eta(R-V)\right)\right]$$



(a) standard PPO (b) Dual-clip PPO

Overview

Tencent AI Lab

- Introduction
- Method
  - System
  - Algorithm
- Experiments
- Conclusion & future work

Experiments: setup

Tencent AI Lab

- System
  - 40+ GPU cards & 15000+ CPU cores used to train one hero
  - 80,000 samples per second per GPU
  - FP16 for data transmission
- Algorithm
  - LSTM
    - time step 16, unit size 1024
  - Discount factor 0.998
  - Generalized advantage estimation (GAE)
    - Lambda 0.95
  - Dual-clip PPO
    - Two clip parameters are 0.2 and 3, respectively

Experiments: results

Tencent AI Lab

- Evaluating the **upper limit** of control ability
  - Match results between AI & top professional human players
    - Best of five (BO5)
    - Tested on different types of heroes
      - Mage, warrior, Marksman, etc.
    - Tested by several top professionals

Hero	DiaoChan	DiRenjie	LuNa	HanXin	HuaMulan
Hero Type	Mage	Marksman	Warrior+Mage	Assassin	Warrior
Score	3:0	3:0	3:0	3:1	3:0
Kill	5.0:1.3	2.3:0.7	2.7:1.0	2.5:1.5	4.0:1.3
Game Length	6’56”	6’23”	7’53”	6’41”	6’48”
Gold/min	852.7:430.6	869.3:606.6	969.7:724.0	954.1:754.2	945.2:654.2
Exp/min	900.0:573.0	895.3:661.7	979.0:817.2	965.4:802.5	921.4:723.1

Experiments: results

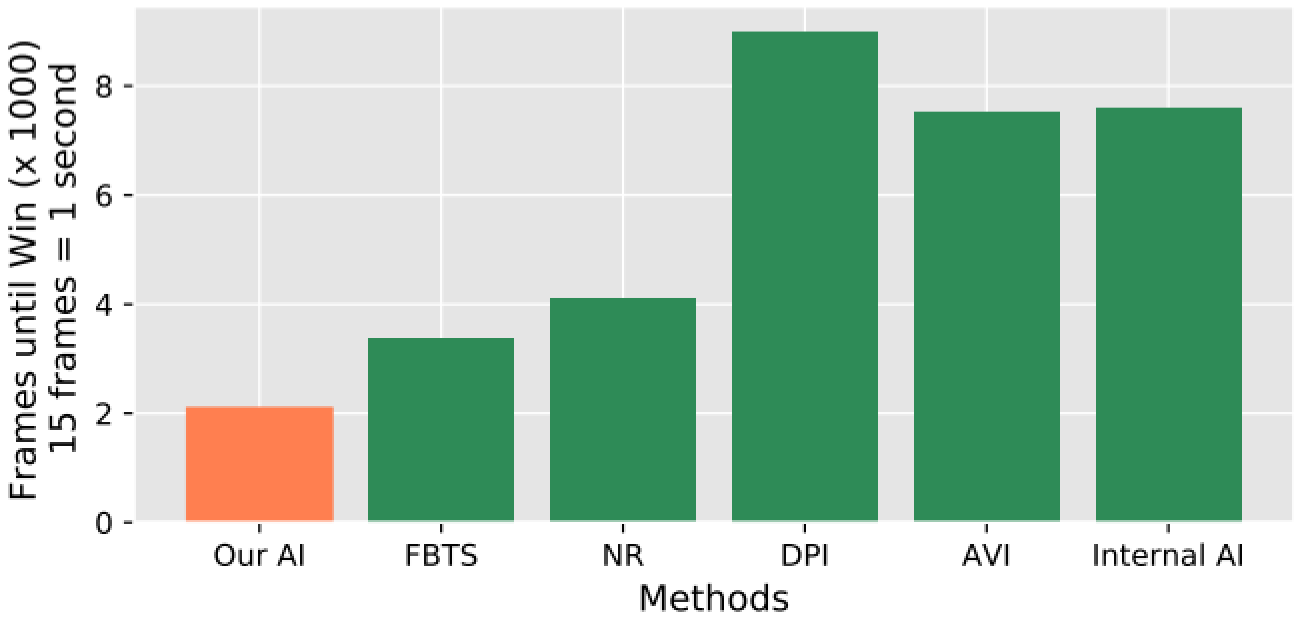
Tencent AI Lab

- Evaluating the **robustness** of control ability
  - 2,100 public matches (AI vs. a **diversity** of top human players)
  - Multiple heroes that require very **diverse** playing method

Hero Name	Hero Type	#Matches	#Win	Rate
DiaoChan	Mage	445	445	100%
DiRenJie	Marksman	264	264	100%
HuaMuLan	Warrior	256	256	100%
HanXin	Assassin	221	220	99.55%
LuNa	Warrior+Mage	260	260	100%
HouYi	Marksman	79	78	98.70%
LuBan	Marksman	354	354	100%
SunWukong	Assassin	221	219	99.09%
		2100	2096	99.81%

Experiments: results

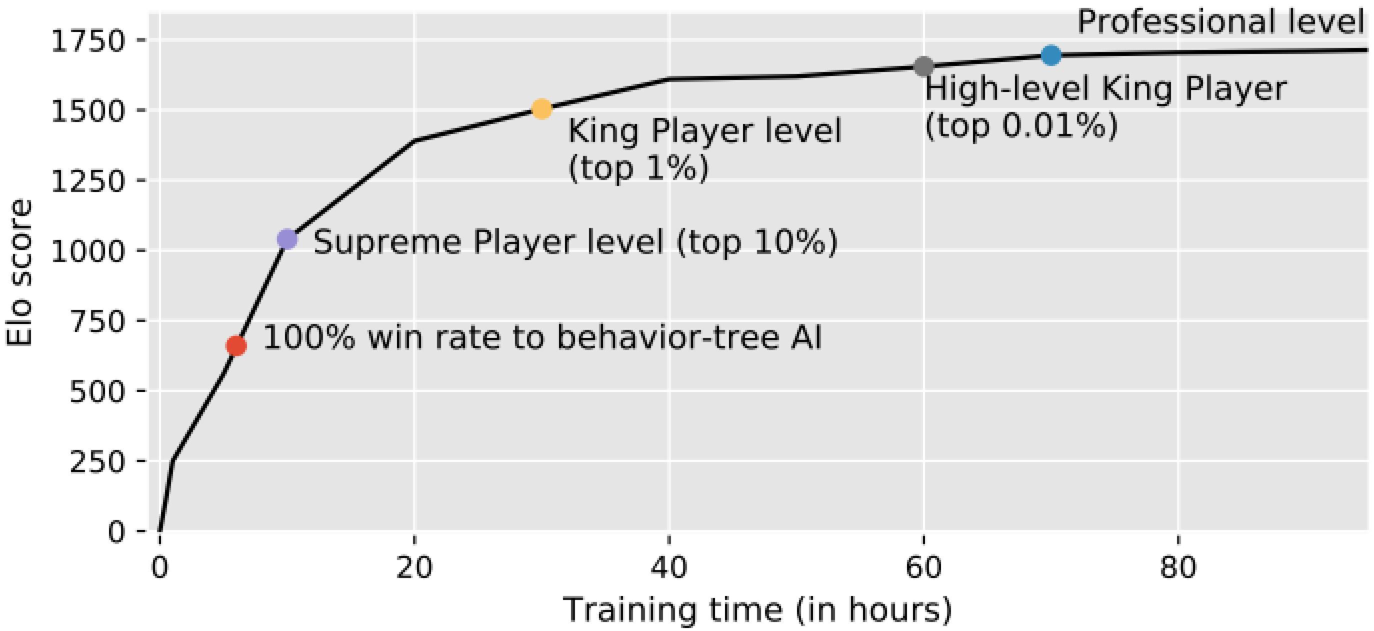
- Comparison with baseline methods
  - Our method vs. MCTS and its variants
  - Measuring average time length to defeat the same set of opponents



Feedback-based tree search for reinforcement learning. ICML 2018

Experiments: results

- The growth of our AI
  - Elo rate



Experiments: results

- Ablation
  - AM: action mask
  - TA: target attention
  - LSTM
  - Base: Full w/o AM TA LSTM

Item	Win rate vs Base	Time to converge
Base	-	80 h
Base + AM	50.5%	**65 h**
Base + TA	**75%**	90 h
Base + LSTM	73%	100 h
Full version	90%	80 h

Overview

Tencent AI Lab

- Introduction
- Background
- Method
    - System
    - Algorithm
- Experiments
- Conclusion & future work

Conclusions

- **Action control** of different MOBA heroes
  - Complex, a big challenging to AI research

- We develop a **super-human AI agent** which has mastered the complex action control in MOBA 1v1 games

- Our deep reinforcement learning framework
  - System design
  - Algorithm design
    - Multi-modal feature design
    - Actor-critic neural network
    - Multiple action control strategies
    - Dual-clip PPO

Future work

- Our ongoing **Open-Platform** Plan (开放平台计划)
  - Open-source our framework & algorithm
  - Honor of Kings GameCore accessibility
  - Computing resources (CPUs/GPUs) for public use

Link: https://mp.weixin.qq.com/s/jaZJtkljVBib0mj1iOJQbg

Future work

- Our ongoing **Open-Platform** Plan (开放平台计划)
  - Open-source our framework & algorithm
  - Honor of Kings GameCore accessibility
  - Computing resources (CPUs/GPUs) for public use

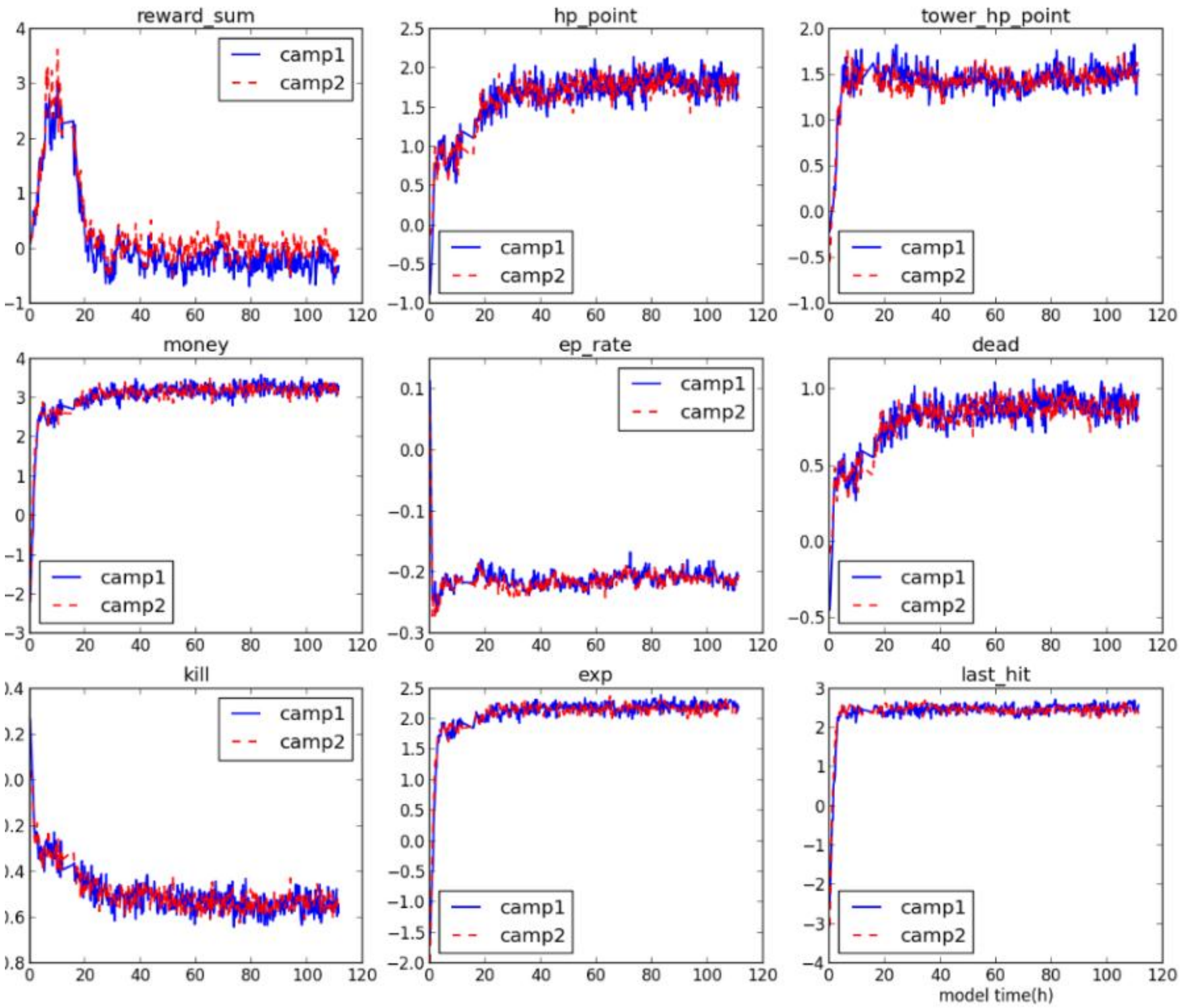
Link: https://mp.weixin.qq.com/s/jaZJtkljVBib0mj1iOJQbg

Appendix

Tencent AI Lab

- Reward design

Reward	Weight	Type	Description
hp_point	2.0	dense	the health point of hero
tower_hp_point	10.0	sparse	the health point of turrets and base
money (gold)	0.008	dense	the gold gained
ep_rate	0.8	dense	the rate of mana
death	-1.0	sparse	being killed
kill	-0.5	sparse	kill an enemy hero
exp	0.008	dense	the experience gained
last_hit	0.5	sparse	last hitting to enemy units



We are recruiting!

Tencent AI Lab

- Computer Science background in a related domain
  - Machine learning in general
- Strong problem solving & analyzing
- Strong programming skills
  - C++/Python/Linux Shell
- Experience in academic research
  - Having peer-reviewed publications is a plus
- Knowledge in Game AI & RL

Send your CV to: dericye@tencent.com