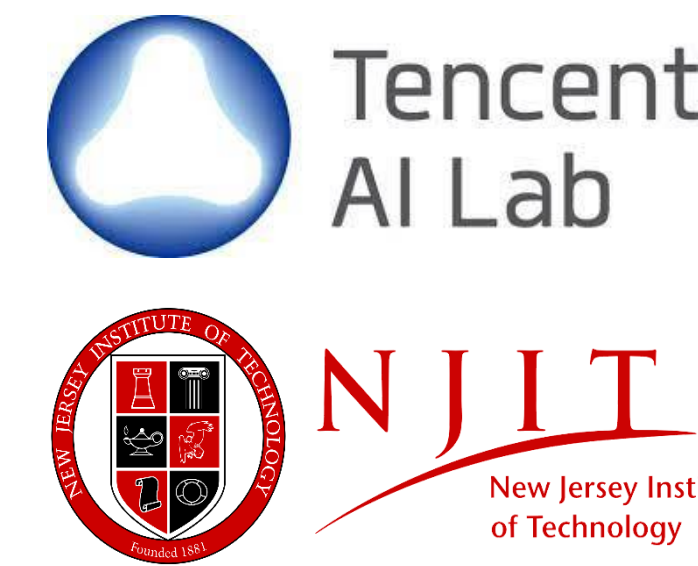


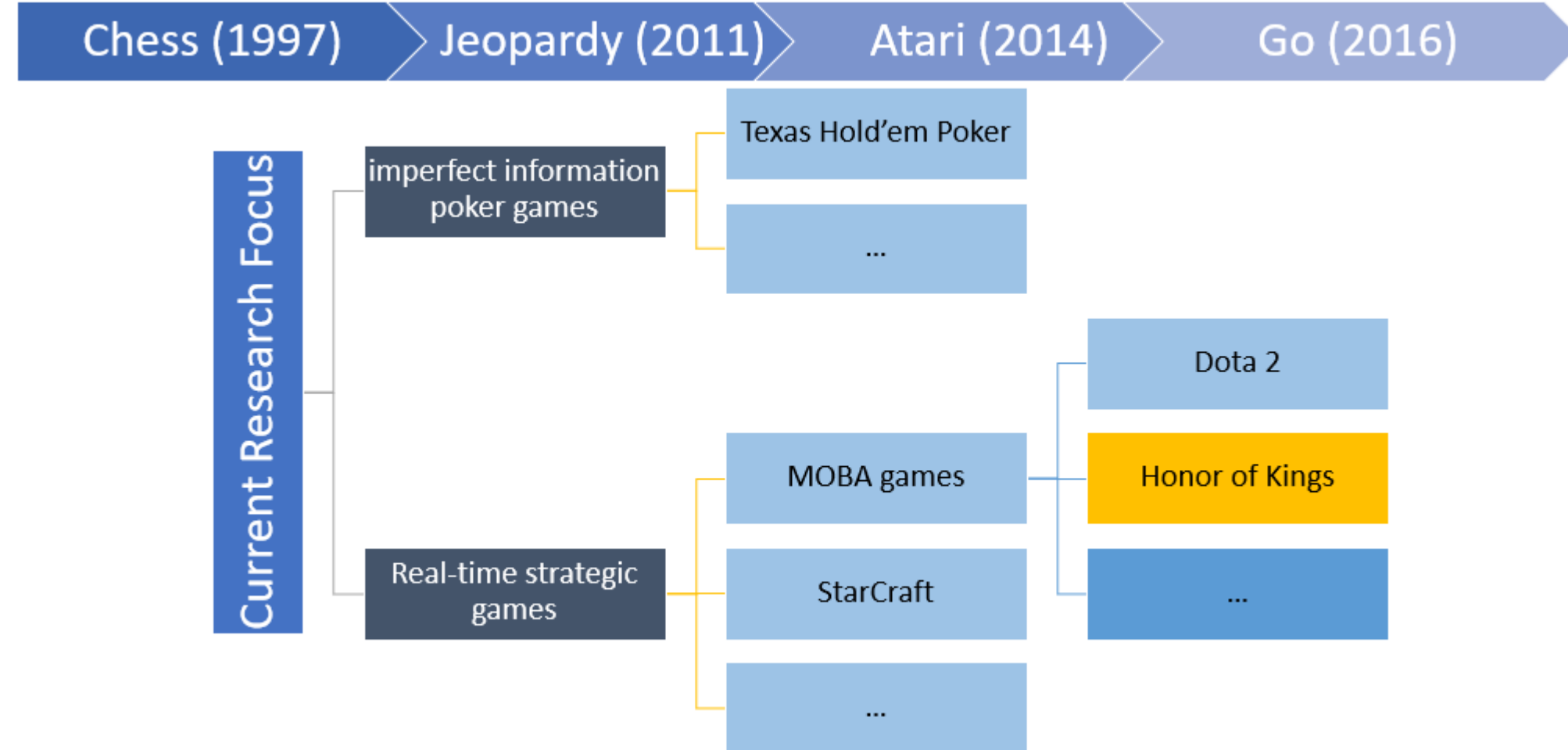
Honor of Kings Arena: an Environment for Generalization in Competitive Reinforcement Learning

Hua Wei*, Jingxiao Chen*, Xiyang Ji*, Hongyang Qin, Minwen Deng, Siqin Li, Liang Wang, Weinan Zhang, Yong Yu, Lin Liu, Lanxiao Huang, Deheng Ye, Qiang Fu, Wei Yang

hua.wei@njit.edu, timemachine@sjtu.edu.cn, wnzhang@sjtu.edu.cn, yyu@apex.sjtu.edu.cn, (xiyangji, hongyangqin, danierdeng, gracesqli, enginewang, linliu, jackiehuang, dericye, leonfu, willyang)@tencent.com



Game as AI Testbeds



Multi-player Online Battle Arena (MOBA) Game: Role Play and Multi-player

Mechanics from MOBA

Role/hero play

- MOBA games have different roles/heroes and each role has different actions

Multi-player

- MOBA games usually involves in two or more parties, each party consists of one or more players

Challenges for AI

Generalization

- Good AI model needs to perform stably well in controlling the actions of different heroes against different opponent heroes.

Multi-agent

- Good AI model need to coordinate well between different players

Honor of Kings: an Appealing Environment for AI

- Popularity
- Existing research interest
 - 10+ related papers in top AI venues
 - NeurIPS, ICML, AAI, IJCAI, TNNLS, ...



In the main screen, there are four parts:

- a mini-map (A),
- a dashboard that records the number of KDAs (kill/death/assist) (B)
- a movement controller (C.1)
- skill controller buttons (C.2)

Generalization Challenge in Honor of Kings Arena (HoK)



Generalization across opponents



Generalization across targets



Benchmarking Results

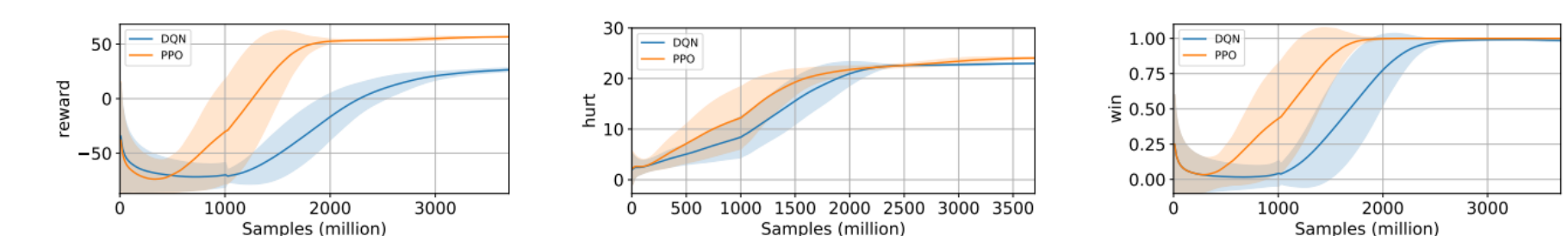


Figure 4: Different evaluation metrics on the Honor of Kings Arena for DQN and PPO w.r.t. the number of training samples. Error bars represent standard deviation. PPO performs better than DQN.

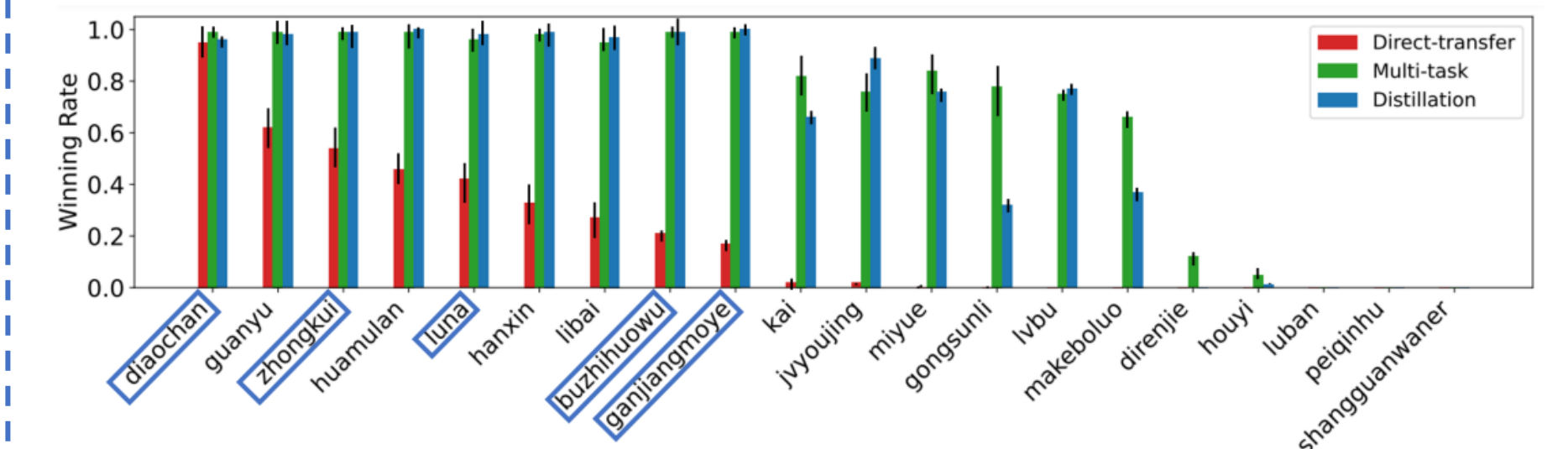


Figure 6: Win rate of a well-trained model from task "Diaochan (RL) vs. Diaochan (BT)" transferred to tasks "Diaochan (RL) vs. different opponent heroes (BT)". The agent is trained to control Diaochan against Diaochan controlled by BT, and tested to control Diaochan against different heroes controlled by BT. Red: Directly transferring the model to control Diaochan and compete with different opponent heroes. Green: Multi-task training on five tasks "Diaochan (RL) vs. Diaochan/Buzhihuowu/Luna/Ganjiangmoye/Zhongkui (BT)" and testing the model on twenty tasks. Blue: Distilling the model trained from five tasks "Diaochan (RL) vs. Diaochan/Buzhihuowu/Luna/Ganjiangmoye/Zhongkui (BT)" and testing the model on twenty tasks. The policy trained on Diaochan could not generalize to all tasks with different opponent heroes. Blue rectangles highlights the five tasks used in multi-task and distillation. The error bars indicate the standard deviation under five seeds.

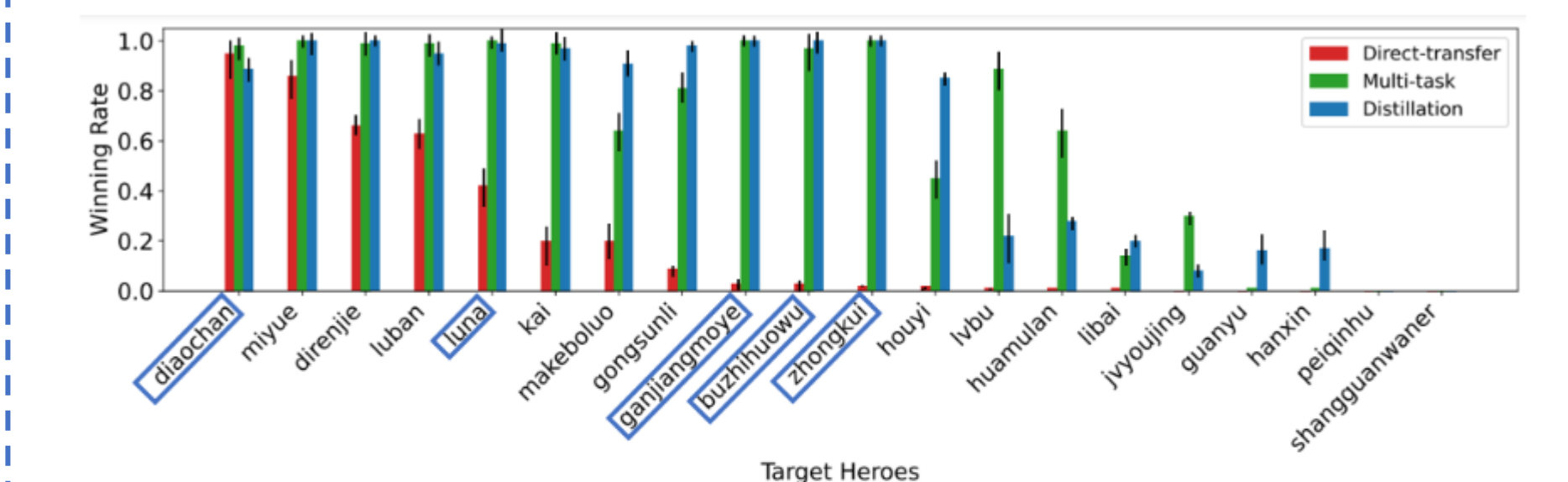


Figure 7: Win rate of a well-trained model from task "Diaochan (RL) vs. Diaochan (BT)" transferred to tasks "Different target heroes (RL) vs. Diaochan (BT)". The agent is trained to control Diaochan against Diaochan controlled by BT, and tested to control different heroes against Diaochan controlled by BT. Red: Directly transferring the model to control Diaochan and compete with different opponent heroes. Green: Multi-task training on five tasks "Diaochan/Buzhihuowu/Luna/Ganjiangmoye/Zhongkui (RL) vs. Diaochan (BT)" and testing the model on twenty tasks. Blue: Distilling the model trained from five tasks "Diaochan/Buzhihuowu/Luna/Ganjiangmoye/Zhongkui (RL) vs. Diaochan (BT)" and testing the model on twenty tasks. The policy trained on Diaochan could not generalize to all tasks with different target heroes. Blue rectangles highlights the five tasks used in multi-task and distillation. The error bars indicate the standard deviation under five seeds.

Honor of Kings Arena (HoK): Provided Resources

- HoK: OpenAI Gym-like, authorized game environments
- Baseline models, including behavior-tree (BT) and RL models
- Replay tool

```

1 from hok import HoK1v1
2
3 # load environment
4 env = HoK1v1.load_game(game_config)
5
6 # init agents
7 agents = []
8 for i in range(env_num_agents):
9     agents.append(Agent(agent_config))
10
11 # start an episode
12 obs, _, _, info = env.reset()
13 total_reward = 0.0
14 done = False
15 while not done:
16     actions = []
17     for i in range(env_num_agents):
18         action = agents[i].process(obs[i])
19         actions.append(action)
20     done, _, _, info = env.step(actions)
21

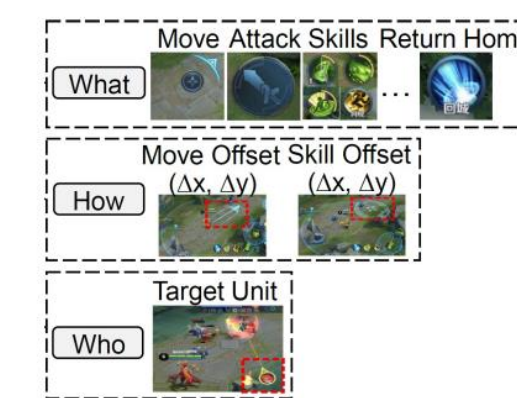
```

Table 3: State information

Feature Class	Field	Description	Dimension
HeroStatePublic	Hero Status	HP, MP level, exp, position, skills, etc	49 * 2
	Hero skills	Specific attributes of hero skills, like buff mark layers, etc	53 * 2
HeroStatePrivate	Diaochan-related	skill position and buffs phases and status of normal attack	11
	Luna-related	strengthened normal attack related	7
	Jyuyoujing-related	attack stage	9
VecCreeps	Status	HP, camp, attack range, etc	12 * 4
	Position	Absolute and relative coordinates, distance	6 * 4
VecTurrets	Status	Hp, camp, attack range, etc	12 * 4
	Position	Absolute and relative coordinates, distance	6 * 4
VecCampsWholeInfo	Current Period	Divide game time into 5 periods	5
Total	Sum	All features	491

Table 5: Reward Design

Reward	Weight	Type	Description
hp_point	2.0	dense	the rate of health point of hero
tower_hp_point	10.0	dense	the rate of health point of tower
money (gold)	0.006	dense	the total gold gained
ep_rate	0.75	dense	the rate of mana point
death	-1.0	sparse	being killed
kill	-0.6	sparse	killing an enemy hero
exp	0.006	dense	the experience gained



Code: https://github.com/tencent-ailab/hok_env
 Documentation: <https://aiarena.tencent.com/hok/doc/>