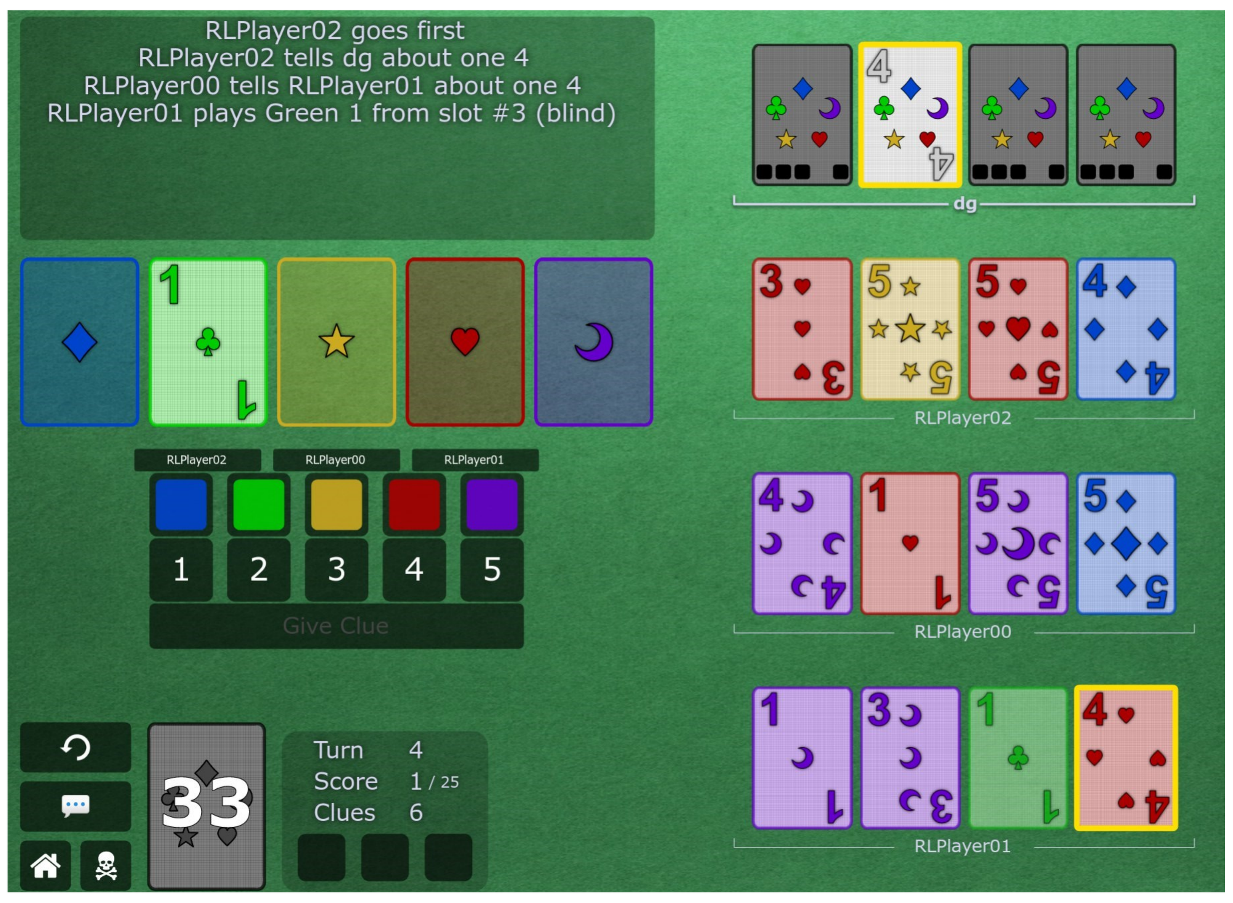
Learning Hanabi

Alexander Dittmann, Dennis Grunwald, Jason Harris, Sascha Lange

Technical University Berlin



Abstract

While Reinforcement Learning (RL) recently achieved super-human results in single player games (Silver et al. 2016), multi-player settings still pose a challenge (Bard et al. 2019). The co-operative multiplayer game Hanabi has great significance, because it incorporates aspects of theory of mind. In this project, we evaluated the performance of multiple state-of-the RL agents in both self- and ad-hoc play. Observing the gameplay of the trained agents, we tried to improve the agents' performances by shaping the re-ward system of Hanabi. Furthermore, we trained agents with rule-based agents in a multi agent setting to find out whether they can adapt to them.

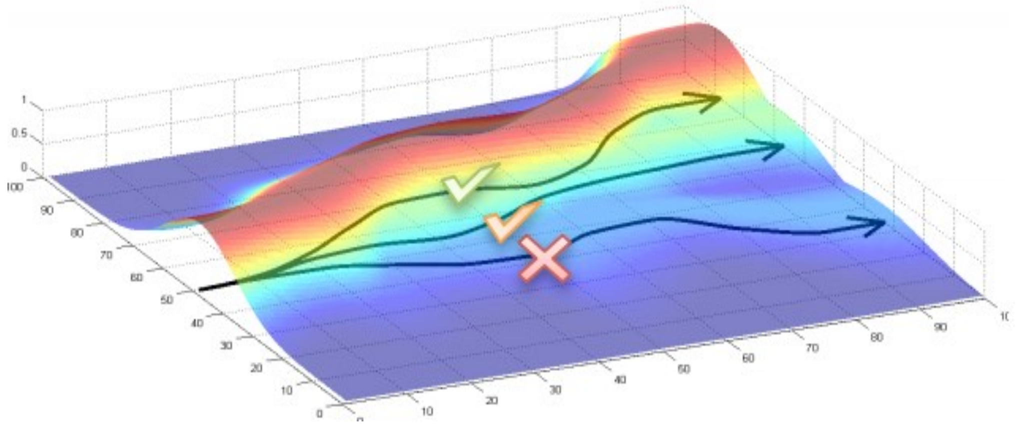
Theory

Rainbow Agent (Hessel et al. 2017)

- Q-learning update rule: $Q(S_t, A_t) \leftarrow Q(S_t, A_t) + \alpha[R_{t+1} + \gamma \max_a Q(S_{t+1}, a) - Q(S_t, A_t)]$
- Additional features: multi-step learning, prioritized replay buffers, distributional RL

REINFORCE Agent (Williams 1992)

- Objective: $\max_\theta \mathrm{E}[\sum_{t=0}^{H} R(s_t) \,|\pi_\theta]$
- Update rule : $\theta \leftarrow \theta + \alpha\gamma^t G \nabla_\theta \ln \pi(A_t, S_t, \theta)$
- With gradient clipping: Proximal Policy Optimization (PPO, Schulman et al. 2017)



Experimental Setup

Environment:

- 4 Players, full game
- State size: 1041, actions: 38

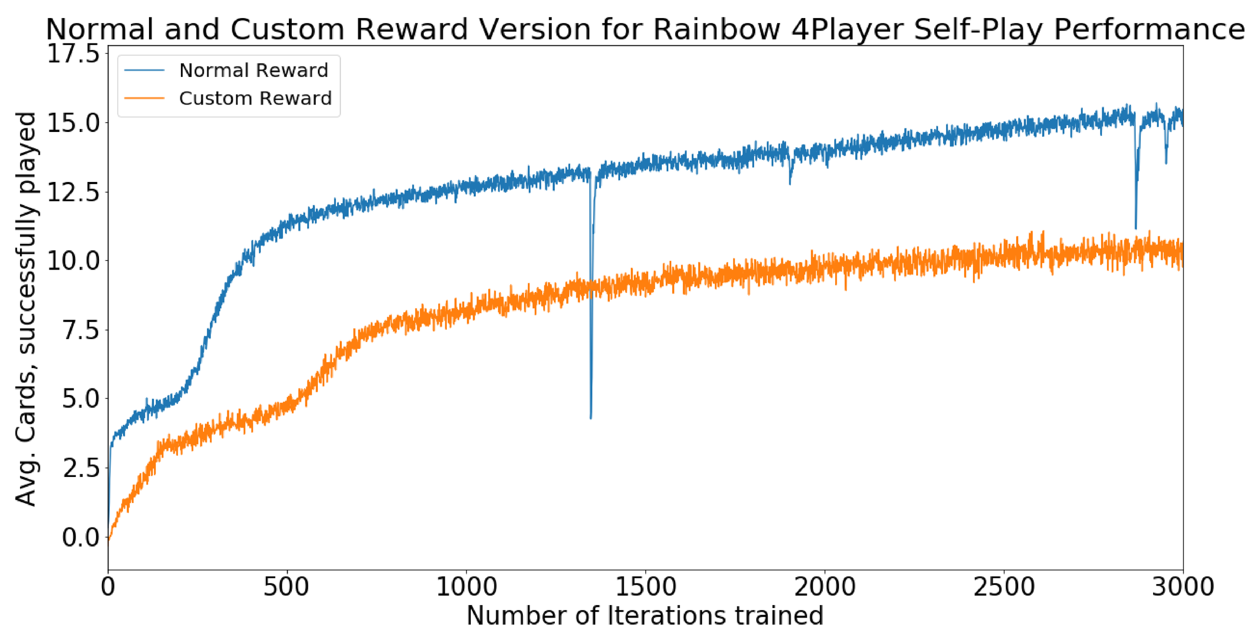
Training:

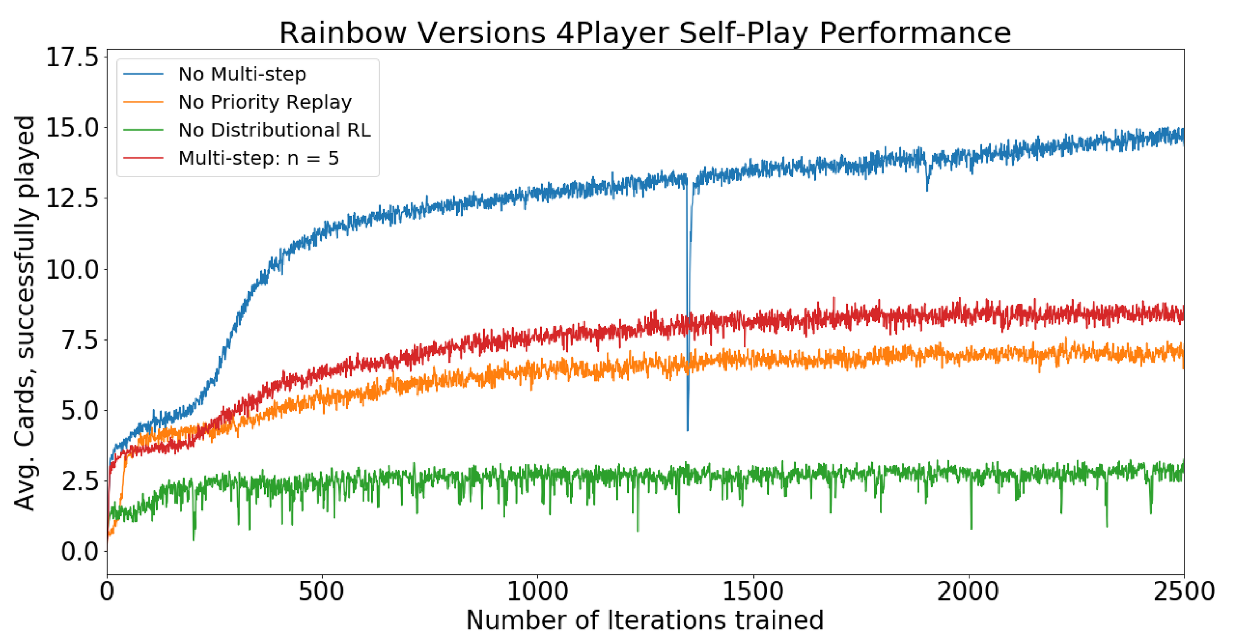
- Rainbow variants + PPO + REINFORCE
- Self-play + ad-hoc with rule-based player

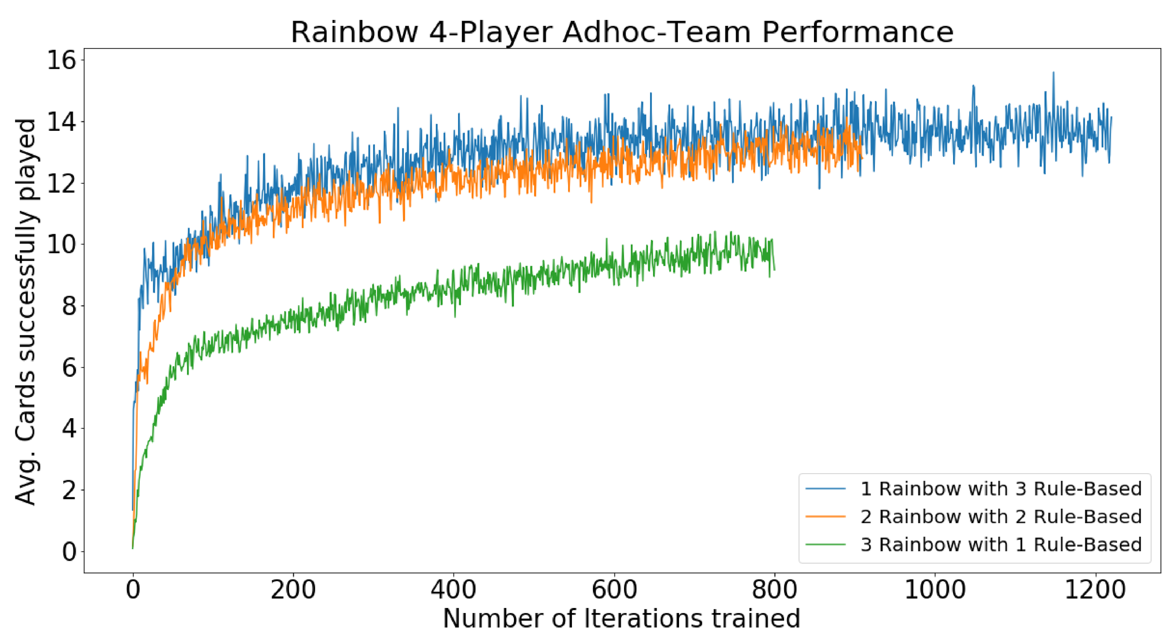
Evaluation:

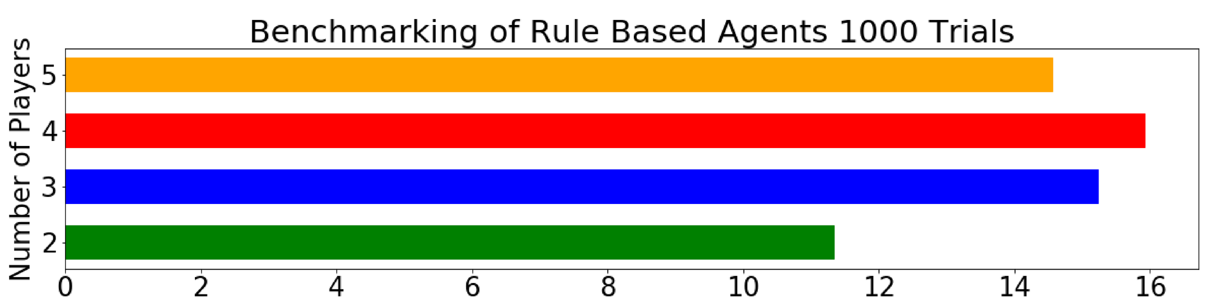
- Graphical user interface
- Self-play and ad-hoc performance

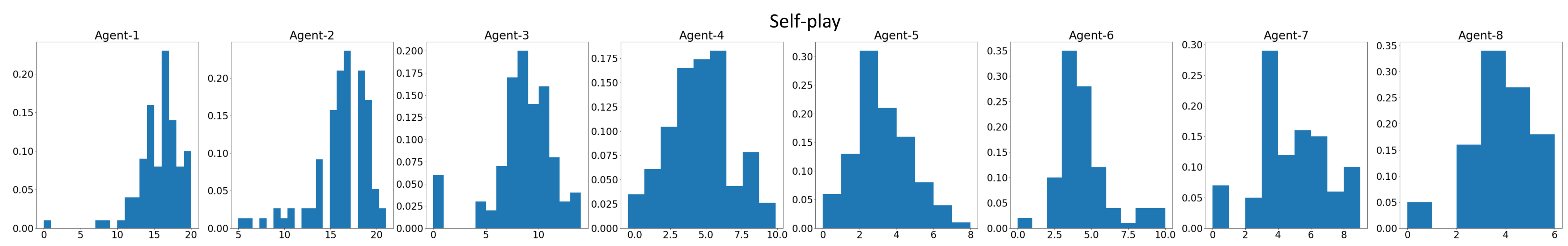
Results

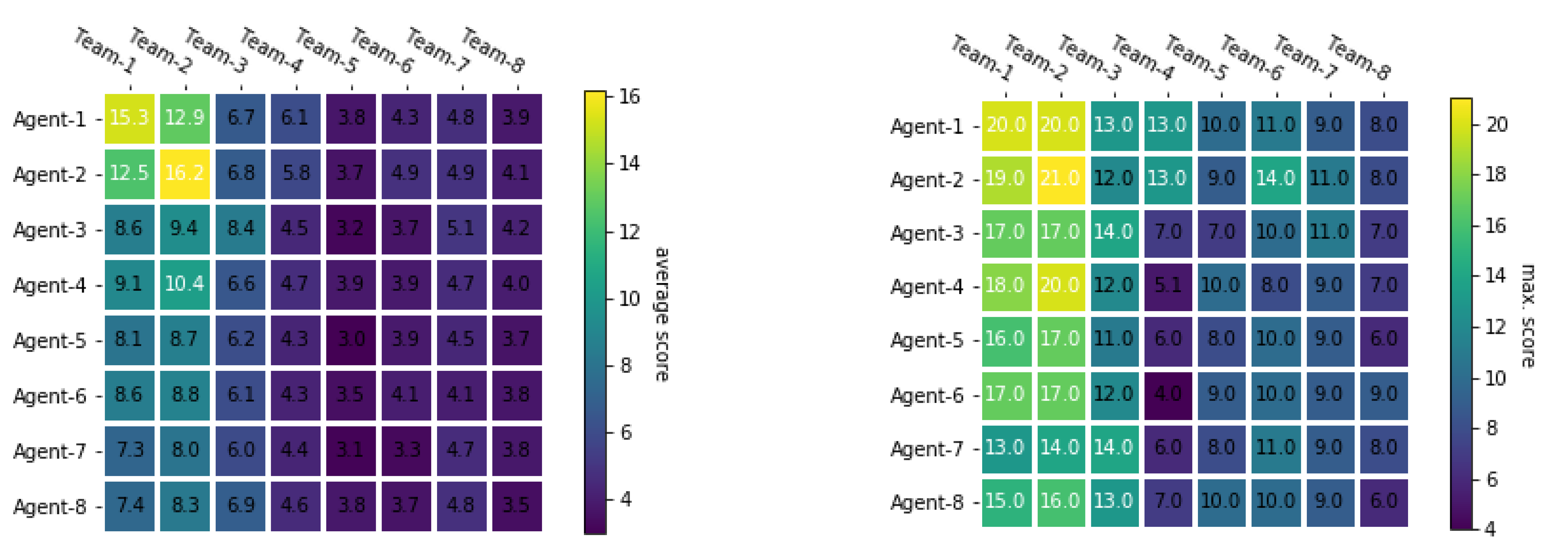












Agent-1: Rainbow (1st run), Agent-2: Rainbow (2nd run), Agent-3: Rainbow Custom Reward, Agent-4: Rainbow with 3 Rule-based, Agent-5: Rainbow with 2 Rule-based, Agent-6: Rainbow with 1 Rule-based, Agent-7: PPO-Agent, Agent-8: REINFORCE-Agent

Conclusion

We created an experimental end-to-end setup. We trained several state-of-the-art RL agents, performed evaluation and provided a graphical user interface to interpret the agents' strate-gies. We trained these agents with rule-based players, showing that they were able to adapt to predefined rule play. From self-play training we concluded, that it is not always useful to incorporate as many state-of-the-art-solutions as possible into the agent: Learning was less successful when using e.g. multi-step updates. The results of training A3C-agents obtained by Deepmind (Bard et. Al 2019) suggested that policy-gradient-methods would generally outper-form DQN approaches. However, we showed that this is not the case for Hanabi, even when using state-of-the-art policy gradient methods, such as PPO. Additionally, the ad-hoc perfor-mance reduces significantly, when evaluating agents that have converged to a certain score, supporting the claim that ad-hoc play requires additional degrees of flexibility.

References

Bard et al. 2019, "The Hanabi Challenge: A New Frontier for AI Research". arXiv preprint arXiv:1902.00506

Guadarrama et al. 2018, "TF-Agents: A library for Reinforcement Learning in TensorFlow". URL: https://github.com/tensorflow/agents

Hessel et al. 2018, "Rainbow: Combining improvements in deep reinforcement learning." Thirty-Second AAAI Conference on Artificial Intelligence.

Schulman et al. 2019, "Proximal policy optimization algorithms." arXiv preprint arXiv:1707.06347

Silver et al. 2016, "Mastering the Game of Go with Deep Neural Networks and Tree Search." Nature 529, no. 7587: 484-489.

Sutton et al. 2018, "Reinforcement learning: An introduction". MIT press

Williams 2019, "Simple statistical gradient-following algorithms for connectionist reinforcement learning." Machine learning 8.3-4 : 229-256

Hanabi Learning Environment: https://github.com/deepmind/hanabi-learning-environment