Supplemental Material: Beyond Value: CHECKLIST for Testing Inferences in Planning-Based RL

Kin-Ho Lam, Delyar Tabatabai, Jed Irvine, Donald Bertucci, Anita Ruangrotsakun, Minsuk Kahng, Alan Fern

Oregon State University
{lamki, tabatase, jed.irvine, bertuccd, ruangroc, minsuk.kahng, alan.fern}@oregonstate.edu

1 Agent Learning Approach Details

To learn the required components we use model-free RL to learn a Q-function $Q(s, a)$ that estimates the value of action a in abstract state s. Specifically, since this is a two-player game, we use a tournament-style self-play strategy, where a pool of previously trained model-free agents, each with their own Q-function, is used to play against a currently learning agent. The agent is trained until it achieves a high win-percentage against the pool or training resources are expended. This is similar to the pool-based self-play strategy employed by AlphaStar for the full game of StarCraft 2 (Vinyals et al. 2019).

To train each agent we use a variant of DQN (Mnih et al. 2015) called Decomposed Reward DQN (Juozapaitis et al. 2019), allowing us to learn a Q-function that returns a vector of probabilities over the different endgame possibilities (e.g. winning by destroying the opponents top base). The sum of the win-condition probabilities for a specific player is the overall value of the action for that player (i.e. the win probability). This vector provides more insight into the agent's decision making and part of the visualization in our explanation interface. The Q-function of the best agent in the pool (typically the last trained agent) is used as the learned action ranking for search. In addition, it is also used for the state-evaluation function $V(s)$ by letting the state value to be the value of the best action, that is $V(s) = \arg\max_a Q(s, a)$.

We represent the Q-function using a 3 layer feed-forward neural network with an input consisting of features describing the abstract state and action. The network outputs the predicted value vector of the state-action pair. Self-play training was conducted for two days, after which the learned model-free agent appeared to be quite strong, likely better than most humans with some game experience.

To learn the dynamics model used for search we formed a training set of abstract state transitions observed in games between agents during pool-based training in addition to games involving random agents to further increase data diversity. Each data instance was of the form $(s, a_f, a_e, s', \vec{r})$ giving the current abstract state, friendly action, enemy action, next abstract state, and decomposed-reward vector respectively. Here the reward vector is the zero vector for all states, except at the end of the game where it is the zero vector for a loss and a one-hot encoding of the win condition otherwise. We designed a feed-forward neural network that takes s, a_f, and a_e as input to predict s' and $\vec{r}$. Note that this approximates the dynamics as a deterministic function. While the actual dynamics are stochastic due to unit level randomization of damage, in aggregate, a deterministic model appears to be adequate for strong play.

The final planning-based agent using the learned components is able to win 80% to 90% of games against the different model-free agents added to the pool during training. This shows that despite potential inaccuracies in the learned components, look-ahead search is able to provide significant improvement over the model-free agents.

References

Juozapaitis, Z.; Koul, A.; Fern, A.; Erwig, M.; and Doshi-Velez, F. 2019. Explainable reinforcement learning via reward decomposition. In *Proceedings of the IJCAI 2019 Workshop on Explainable Artificial Intelligence*, 47–53.

Mnih, V.; Kavukcuoglu, K.; Silver, D.; Rusu, A. A.; Veness, J.; Bellemare, M. G.; Graves, A.; Riedmiller, M.; Fidjeland, A. K.; Ostrovski, G.; et al. 2015. Human-level control through deep reinforcement learning. *Nature*, 518(7540): 529–533.

Vinyals, O.; Babuschkin, I.; Czarnecki, W. M.; Mathieu, M.; Dudzik, A.; Chung, J.; Choi, D. H.; Powell, R.; Ewalds, T.; Georgiev, P.; et al. 2019. Grandmaster level in StarCraft II using multi-agent reinforcement learning. *Nature*, 575(7782): 350–354.