Proceedings of the Sixteenth AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment (AIIDE-20)

Behavioral Evaluation of Hanabi Rainbow DQN Agents and Rule-Based Agents

Rodrigo Canaan,[1] Xianbo Gao,[1] Youjin Chung,[1] Julian Togelius,[1] Andy Nealen,[2] Stefan Menzel[3]
[1]NYU Tandon School of Engineering, New York, USA
[2]University of Southern California, Los Angeles, USA
[3]Honda Research Iinstitute Europe GmbH, Offenbach, Germany
{rodrigo.canaan, xg656, yjc433, julian.togelius}@nyu.edu, nealen@usc.edu, stefan.menzel@honda-ri.de

Abstract

Hanabi is a multiplayer cooperative card game, where only your partners know your cards. All players succeed or fail together. This makes the game an excellent testbed for studying collaboration. Recently, it has been shown that deep neural networks can be trained through self-play to play the game very well. However, such agents generally do not play well with others. In this paper, we investigate the consequences of training Rainbow DQN agents with human-inspired rule-based agents. We analyze with which agents Rainbow agents learn to play well, and how well playing skill transfers to agents they were not trained with. We also analyze patterns of communication between agents to elucidate how collaboration happens. A key finding is that while most agents only learn to play well with partners seen during training, one particular agent leads the Rainbow algorithm towards a much more general policy. The metrics and hypotheses advanced in this paper can be used for further study of collaborative agents.

Introduction

Hanabi (Bauza 2010) is a cooperative card game with hidden information and a limited communication channel that has recently been proposed as an ideal testbed for Artificial Intelligence (AI) agents that are capable of reasoning about other players' beliefs and intentions (Osawa 2015; Walton-Rivers et al. 2017; Bard et al. 2020). Of particular importance is the *ad-hoc teamplay* setting, where teams of agents are required to play with no previous coordination of strategies (Stone et al. 2010). This is because if coordination is allowed before the game, players can agree on *conventions* that encode a set of assumptions on how all participants should play the game. This allows each player to infer additional information about hidden aspects of the game state.

Strategies based on human-created conventions such as *hat-guessing* (Cox et al. 2015; Wu 2016; Bouzy 2017) can achieve very high average scores, above 24 out of a maximum of 25, and a high percentage of perfect games. These conventions, however, rely on an arbitrary assignment of meaning to features of the game such as card colors. This makes them unsuited for playing with players that did not agree to the convention, as is the case in ad-hoc teamplay. It has also been shown that many recent agents based on Deep Reinforcement Learning such as $ACHA$ (Bard et al. 2020), BAD (Foerster et al. 2018) and SAD (Hu and Foerster 2019) can "learn" similar conventions with no human guidance when trained in a self-play setting. Thus, these agents are unsuited for ad-hoc teamplay, including situations where teams are composed of independently trained instances of the same model.

While the ad-hoc teamplay problem has been less studied than self-play, our work builds on the ideas and experiments of our recent workshop paper (Canaan et al. 2020a). There, we trained versions of a Rainbow DQN agent developed by (Bard et al. 2020) by pairing them with seven human-inspired rule-based agents by previous authors (Osawa 2015; van den Bergh et al. 2016; Walton-Rivers et al. 2017). Since these rule-based agents do not rely on specialized conventions like hat-guessing, we believed an agent trained using one of these partners had the potential to learn policies that also apply well to unseen rule-based agents. However, we observed that not only did the policies learned with one rule-based partner rarely transfer well to playing with other partners, these strategies were also quite weak when evaluated in a self-play setting.

In this paper we similarly train Rainbow DQN agents using six of the seven rule-based agents as partners but also investigate the fact that some of these partners enable Rainbow to learn more general policies than others. We do this by gathering behavioral metrics such as Communicativeness and Information Per Play (IPP) defined in (Canaan et al. 2019; 2020b), among others. These metrics help shed light on game events beyond a simple reporting of scores. They also help formulate hypotheses about the relationships between each agent's behavioral characteristics, their ad-hoc performance and the performance of reinforcement learning agents trained using them as partners. Due to additional training time, some agents' performances are also remarkably different than reported in (Canaan et al. 2020a).

All code for running and visualizing the agents is available to other researchers in our online repository [1].

[1]https://github.com/rocanaan/hanabi-ad-hoc-learning

Hanabi: The Game

Hanabi is a game where a team of 2-5 players attempts to play cards in a correct sequence. Each card has a numerical rank from 1 to 5 and one of five colors. Players try to build five piles, one for each color, in ascending rank order. However, each player does not see the contents of their own hands, only those of their partners. Each correctly played card scores 1 point for the team (up to a maximum of 25), but each incorrectly played card depletes one of the team's lives. If all lives are lost, the team's score is reduced to zero and the game ends (using the *strict* scoring scheme favored in this paper. In the alternative *lenient* scheme, the team keeps whatever score they had leading up to the loss).

Players are allowed to spend an action to provide information to other players. This consists on selecting another player and pointing to all the cards in that player's hand with a chosen rank or color. This spends a hint token from a shared pool. Any other form of communication between players is disallowed. Players can also spend an action to discard a card and recover a hint token, but since the number of cards in the deck is limited, this risks making some piles impossible to complete. If the draw deck is depleted, the game ends after one last round of actions by each player.

It is useful to distinguish between grounded information, provided by the game rules, and non-grounded information inferred from a belief about other player's intentions. For example, a hint such as "your third card is Red" identifies that card as red and the remaining cards as non-red on a grounded level, but may also imply additional information, such as that it is likely playable, if we assume that our partner is likely to prioritize hints about playable cards. These assumptions can improve a team's performance, but can also backfire if they are not shared by the whole group.

Related Work

Many of the early AI agents for playing Hanabi could be defined by a sequence of rules, where each rule takes a condition (e.g. another player has a playable card) and, if that condition is satisfied, returns a corresponding action (e.g. hint that card's rank). In (Walton-Rivers et al. 2017), the authors propose to evaluate agents based on their ability to play well with diverse partners, including rule-based agents first proposed by (Osawa 2015) and (van den Bergh et al. 2016). Later, they organized a competition (Walton-Rivers et al. 2017) based on this type of evaluation, where in the Mixed track, participants would be paired with agents that were kept secret up to the competition. The competition ran at the 2017 CIG and 2018 CoG conferences and the organizers provided a Java framework that allowed participants to use pre-implemented rules to define their own agents. The rule-based agents used in this paper were originally provided in the competiton framework as samples.

Another framework for playing Hanabi is the Hanabi Learning Environment (HLE), by (Bard et al. 2020). HLE is a Python framework focused on reinforcement learning (RL), and features the original implementation of the Rainbow agent used in this paper, although trained only in a self-play setting. The authors remarked that Rainbow appeared to learn compatible policies across different training runs, whereas another agent introduced in that work, $ACHA$, has better self-play performance but does not perform well in that ad-hoc scenario. More recent RL agents such as BAD (Foerster et al. 2018) and SAD (Hu and Foerster 2019) perform even better at self-play, with similar problems when paired with independently trained versions of themselves if trained naively. A combination of SAD with a multi-agent search protocol proposed by (Lerer et al. 2019) is the current state-of-the-art in self-play Hanabi, with average scores of 24.61 in the 2-player version.

The Hanabi Rainbow agent bundled with the HLE framework is a multi-agent Hanabi version of the Rainbow DQN Atari agent (Hessel et al. 2018) which combines many extensions to the "vanilla" DQN agent that had been proposed over time, and evaluates which extensions actually improve the algorithm through a series of ablation tests. While the extensions that define Rainbow could in principle be applied to a variety of neural network architectures, we focus on the simple feedforward architecture provided by HLE.

This paper builds on a workshop paper (Canaan et al. 2020a), where we re-implemented the rule-based agents in the HLE and trained Rainbow agents using both self-play and rule-based agents as partners. Evaluation consisted on playing with all rule-based agents as well as self-play, where we observed that performance rarely transfers well to agents not seen during training. We extend that work with an investigation on why certain agents, if used as training partners, lead to the RL agent learning policies that transfer better or worse to pairings with other agents. This investigation is based on two behavioral metrics defined by (Canaan et al. 2020b) for evolving diverse agents using MAP-Elites (Mouret and Clune 2015), along with newly introduced metrics.

An algorithm called Other Play (Hu et al. 2020) was recently proposed as a solution to the problems of agents that do not play well with independently trained versions of themselves, such as SAD. Other Play re-labels game features (such card colors) under symmetries of a POMDP during training, to avoid the agent leaning on conventions that violate these symmetries. It is not explicitly designed to play well with the rule-based agents under consideration in this paper (which do not use conventions based on color), but was shown to achieve good scores with human partners.

Methods

All games simulated for this paper, unless otherwise specified, used the 2-player version of Hanabi, with the strict scoring scheme, randomizing the starting player.

Selection of Rule-based Agents

We selected six of the seven rule-based agents implemented in the Hanabi Learning Environment by (Canaan et al. 2020a) to train our Rainbow agents. We excluded $LegalRandom$ from our experiments since it is not an agent from which any meaningful learning can be expected, especially using the strict scoring scheme. We provide below a few details about these agents that we deem relevant for the purpose of cooperating with them. A full description of these agents is out of the scope of this paper, but can be seen at (Walton-Rivers et al. 2017), where the agents $IGGI$, $Piers$ and $Flawed$ were also introduced. $Internal$ and $Outer$ are originally from (Osawa 2015) and VDB is from (van den Bergh et al. 2016).

$Flawed$ is a poor agent by design, which gives random hints and plays cards based on a very low threshold of playability. However, still attempts to play whatever card has the highest probability of being playable, so some level of success is achievable if its partner gives it abundant information, but this is unlikely to be achieved by any agent that did not have $Flawed$ as training partner.

$Internal$ and $Outer$ are simple agents, prioritizing playing cards that are for sure playable and discarding cards that are no longer useful, followed by giving hints about playable cards, and then about other cards. The only difference between them is that $Internal$ has no memory of the hints other players have received and is liable to give repeated hints. The only other agent with this property is $Flawed$.

The rules used by $IGGI$ do not cover all possible game states and will rarely return a random action. Other than this edge case, however, $IGGI$ is the only agent who will never give a hint that does not contain at least one playable card. All other agents have hint rules that activate even if the receiving player has no playable cards, such as hinting cards that should be discarded for extra hint tokens. This makes $IGGI$ potentially very predictable, as all hints it gives (except for the mentioned edge case) contain at least one playable card.

$Piers$ and VDB's distinguishing feature are their probabilistic play rules: they compute the probability of a card being playable by eliminating all cards incompatible with the grounded information received so far. They then play their most likely playable card if this probability is above the threshold of 60%.

Finally, the agents, except $Flawed$, have hint rules with deterministic "tiebreakers" in case multiple cards apply: they usually scan the other players' cards from oldest to newest, then find the first applicable (e.g. playable) card. $Internal$ chooses randomly between hinting that card's color or rank, while the others prioritize rank over color. This also makes these agents more predictable than might seem at first glance.

Training of Rainbow Agents

The architecture and training procedure we use for this paper are the same (other than the choice of training partner) as used in (Bard et al. 2020): value distributions are approximated using distributional reinforcement learning (Bellemare, Dabney, and Munos 2017) by a 2-layer Multi-Layer Perceptron with 512 nodes per hidden layer and the action is selected ϵ-greedily over the expected value of each of the 20 possible actions (although not all actions are necessarily legal at each game state). The input space consists of a binary vector of size 658 representing the current game state plus the most recent action. All hyperparameters were the same as those used in (Bard et al. 2020).

The game state representation encodes features of the game that are directly seen by the agent, such as the cards already played and discarded, the number of hint tokens, life tokens and cards in the deck, and the rank and color of all cards in other players' hand. The representation also keeps track of hints received by all players, so the agent knows at all times the possible rank and color of cards in its hand and also which information is known by other players.

However, the representation doesn't keep track of game history beyond the last action, and the neural network used by the agent is feed-forward with no recurrent connections. As such, the agent is not expected to learn policies dependent on a long history of actions. Therefore, our inquiry on how well this agent plays with other teammates is not out of hope that it is able to identify and adapt to different teammates, since it lacks the long-term memory needed for it. Rather, it is based on the observation that the agent, when trained through self-play, seems to learn strategies that work well among independently trained instances of the policy, which might be a sign that these strategies rely less on arbitrary conventions and more on grounded information, and thus might play well with the chosen rule-based agents.

We trained agents in nine different regimes: $R_{Internal}$, R_{Outer}, R_{Flawed}, R_{IGGI}, R_{Piers} and R_{VDB} are each trained with a single rule-based agent as partner. R_{SP} trains through pure self-play. R_{All} plays each training game with a randomly chosen rule-based agent, but plays no training game in a self-play setting. The report by (Canaan et al. 2020a), however, assigns very poor self-play score to this agent, and for this reason we add a ninth type of agent, R_{All+SP}, which has a 1/7 chance of either playing with a copy of itself or with a random rule-based agent. For each regime, we trained four independent instances of the agent.

We allowed each instance to train for a total of 200 million game actions. Since the Rainbow agents only perform updates on their own actions, this corresponds to 100 million training steps for the agents trained exclusively with rule-based partners, the same number of training steps used in (Bard et al. 2020). This also means our R_{SP} trained for twice as many training steps as in (Bard et al. 2020). However, as seen in the Results section, performance of R_{SP} did not improve much after the first half of training. By comparison, (Canaan et al. 2020a) were only trained their R_{SP} agents for 30 to 60 million game steps, and the remaining agents for 27.5 million game steps.

We used a Linux machine with a Intel Core i7-5930K 3.5 GHz Processor with six cores and Cuda 9.1 with three GTX 1080 GPUs. Training time depended on the agent being trained, ranging from around 200 to 300 game steps per second, for a total of 8 to 12 days of training per agent, with up to 9 agents being trained simultaneously.

Collection of Behavioral Metrics

While game scores can tell us *which* pairs of agents play well or poorly with each other, they give little insight into *why* these agents succeed or fail. For this reason, while we evaluate each of the trained Rainbow agents with the six rule-based agents and self-play (regardless of which agents the Rainbow agent played with during training), we also collect a number of behavioral metrics that help shed light on this problem and formulate hypotheses.

We suspect that most failures to cooperate happen due to a combination of the Rainbow agent not understanding the hints given by its partner (and thus committing too many mistakes or not playing cards that would have been playable) or, conversely, due to the Rainbow agent giving hints in ways that its partners do not understand.

For this reason, we collect behavioral metrics relating to the cards played and hints given by both players. We start with the metrics of "Communicativeness" and "Information Per Play", defined by (Canaan et al. 2020b):

- Communicativeness is defined as the ratio of hints given by the agent to the number of turns where a hint action was available.
- Information per Play (IPP) is defined as the average number of pieces of information (a number between 0 and 2 corresponding to no information, only rank, only color or both rank and color) known by the agent about each card it played.

In particular, we expect agents with very low IPP to "overfit": they will learn to infer a lot hidden information from its training partner's actions and play cards correctly with little information, in a way that does not work with other agents.

We go beyond these metrics and also collect the number of cards correctly played by both players in a match-up, the number of mistakes made by both players and the number of games where the group "bombed out" by losing three lives. We then average these statistics among match-ups that were seen during training as well as among all match-ups. The gap between an agent's correctly played cards and mistakes when playing with its training partner and unseen partners can yield similar insights as IPP, and the corresponding number for the other player might indicate if the Rainbow agent is giving hints in ways that are helpful to them.

Results

Before training the Rainbow agents, we played 1000 games between each pair of the 6 rule-based agents for a total of 36 pairings and 36000 games. The score in these match-ups could signal to differences and incompatibilities between the strategies used by these agents. Table 1 reports the average score and standard deviations (SD) of each match-up.

We also highlight the mean self-play score of these agents in the rightmost column. These values are useful as benchmarks since, if a Rainbow agent achieves higher scores than the value of this column for a certain agent, it means it learned to play at least as well with that partner as the corresponding rule-agent plays with itself. On average, the six rule-based agents have a self-play score of 12.10.

Note that $Flawed$, as expected, has scores close to zero across the board, and that most agents' best scores are not with themselves, but with "stronger" partners, especially $Piers$. The best overall pairing is between $Piers$ and VDB at over 17 points.

During these games we also collected behavioral metrics for the rule-based agents, which we make available in our repository but omit here for space considerations.

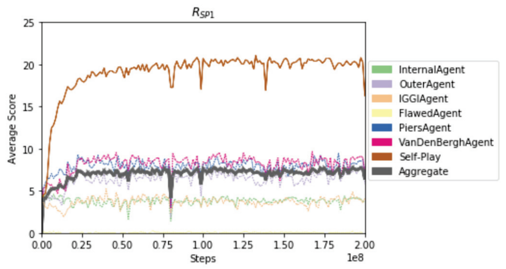

Figure 1: Training curve of the first instance of R_{SP}, showing evaluation performance with all 6 rule-based agents and self-play over training time. This example was chosen to exemplify a sharp drop in performance in the last episodes of training (confirmed in an independent round in evaluation). This instability means that the final version of each instance (whose score we report) is not necessarily the best achieved during training.

Performance of Trained Rainbow Agents

For each of the 9 training regimes described in the previous section we trained 4 instances of Rainbow agent with different random seeds, for a total of 36 instances. For each instance, we performed small evaluation rounds every 1.25 million steps of training, consisting of 100 games paired with each of the 6 rule-based agents as well as a copy of itself (for a total of 700 games across 7 match-ups for each evaluation round during training). Figure 1 shows the resulting training curve for the first instance of R_{SP}. Note that due to oscillations in this curve, the final checkpoint, reported for each instance, is not necessarily the one with maximum performance over all training checkpoints.

After training completed, we performed a larger evaluation round of 1000 games per match-up (for a total of 7000 games for each of the 9*4 instances). The mean score and SD of each match-up (averaged over the 4 different instances of each Rainbow agent) are shown on table 2. Comparing this table with table 1, we see that all Rainbow agents that trained with a single rule-based partner (above the dashed line in table 2) achieved higher average score with that partner than that partner's baseline self-play performance (rightmost column of table 1). However, with the exception of $R_{Internal}$, aggregate and self-play performance for most of these agents was quite poor. Note that although aggregate performance includes self-play, none of the agents above the dashed line (except $R_{Internal}$) would achieve 12.10 aggregate score even discounting self-play.

On the other hand, $R_{Internal}$ represents an example of agent trained with a single rule-based partner, but that achieves good score across the remaining rule-based agents (above the rule-based agent's self-play baseline) and a reasonable self-play score of 11.26 which, although not competitive with R_{SP} or more recent state-of-the-art self-play agents, is much higher than the corresponding score for $Rainbow_{Internal}$ of 3.91 reported (with fewer training steps) in our original workshop paper (Canaan et al. 2020a).

On the opposite extreme, R_{IGGI} also represents an inter-

	Internal	Outer	IGGI	Flawed	Piers	VDB	Self-Play
Internal	10.04 (2.05)	11.84 (2.14)	12.59 (1.98)	0.04 (0.60)	13.71 (2.17)	13.37 (2.53)	10.04 (2.05)
Outer	11.86 (2.20)	13.8 (2.15)	15.38 (1.85)	0.07 (0.84)	15.48 (2.00)	14.77 (2.92)	13.8 (2.15)
IGGI	12.64 (2.02)	15.41 (1.86)	15.99 (4.61)	0.17 (1.30)	16.77 (2.60)	16.48 (3.34)	15.99 (4.61)
Flawed	0.04 (0.59)	0.02 (0.36)	0.24 (1.47)	0.00 (0.00)	0.18 (1.44)	0.23 (1.56)	0.00 (0.00)
Piers	**13.67 (2.31)**	**15.51 (1.95)**	**16.61 (2.91)**	0.15 (1.10)	16.99 (1.97)	**17.14 (2.46)**	**16.99 (1.97)**
VDB	13.32 (2.56)	14.76 (2.99)	16.55 (3.47)	**0.18 (1.35)**	**17.18 (2.38)**	15.81 (4.41)	15.81 (4.41)
Average	10.26 (1.95)	11.89 (1.91)	12.89 (2.72)	0.10 (0.87)	13.38 (2.09)	12.97 (2.87)	12.10 (2.53)

Table 1: Mean scores and standard deviations obtained when playing 1000 games for all pairings of the rule-based agents. Values across the main diagonal correspond to the self-play scores and are repeated in the Self-Play column for ease of visualization. Values on either side of the main diagonal represent two independent sets of games between the same pairs of agents. The maximum score in each column is highlighted in bold.

	$Internal$	$Outer$	$IGGI$	$Flawed$	$Piers$	VDB	$Self$	Aggregate
$R_{Internal}$	**13.86 (3.6)**	14.65 (3.9)	15.97 (3.3)	0.06 (0.8)	15.05 (4.4)	14.64 (5.4)	11.26 (6.4)	12.21 (4.0)
R_{Outer}	7.75 (6.8)	**17.38 (2.6)**	16.2 (4.9)	0.00 (0.1)	12.71 (7.9)	13.36 (7.7)	5.03 (7.7)	10.35 (5.4)
R_{IGGI}	2.10 (4.4)	1.80 (4.9)	**18.25 (2.9)**	0.00 (0.0)	7.01 (8.8)	7.66 (8.7)	0.63 (2.9)	5.35 (4.7)
R_{Flawed}	4.13 (1.6)	5.65 (1.8)	4.30 (2.2)	**4.11 (3.7)**	7.52 (2.4)	7.74 (2.7)	3.24 (2.1)	5.24 (2.4)
R_{Piers}	3.37 (5.5)	7.14 (7.9)	17.09 (3.7)	0.01 (0.2)	**17.68 (4.0)**	13.26 (7.6)	7.52 (8.2)	9.44 (5.3)
R_{VDB}	6.06 (5.8)	10.58 (6.9)	16.27 (3.3)	0.02 (0.3)	13.91 (7.0)	**18.16 (3.5)**	4.42 (6.2)	9.92 (4.7)
R_{SP}	3.02 (2.6)	5.06 (3.8)	4.25 (2.8)	0.05 (0.5)	7.43 (4.9)	7.14 (5.5)	**19.53 (4.8)**	6.64 (3.5)
R_{All}	13.50 (3.4)	16.03 (3.1)	17.45 (2.9)	0.47 (1.5)	17.07 (3.6)	16.78 (3.9)	12.89 (6.3)	13.46 (3.5)
R_{All+SP}	13.11 (3.4)	15.64 (3.1)	16.85 (3.2)	0.41 (1.5)	16.66 (3.7)	16.32 (3.9)	15.77 (4.7)	**13.54 (3.3)**
Average	7.43 (4.1)	10.44 (4.2)	14.07 (3.2)	0.57 (0.1)	12.78 (5.2)	12.78 (5.4)	8.92 (5.5)	9.57 (4.1)

Table 2: Mean scores and SD for each match-up. Values are averaged across the 4 different runs of each type of Rainbow agent per match-up. Each match-up was played 1000 times for each run, so each match-up was played 4000 times in total. Agents above the dashed line were trained with a single rule-based agent as partner. The maximum score in each column is highlighted in bold.

Type of Rainbow Agent	$R_{Internal}$	R_{Outer}	R_{IGGI}	R_{Flawed}	R_{Piers}	R_{VDB}	R_{SP}	R_{All}	R_{All+SP}
Score with partner	13.86	17.38	18.25	4.11	17.68	18.16	19.53	13.54	13.55
Aggregate score	12.21	10.35	5.35	5.24	9.44	9.92	6.64	13.54	13.46
Comm with partner	0.60	0.38	0.57	0.21	0.47	0.48	0.42	0.43	0.45
Aggregate comm	0.65	0.45	0.46	0.43	0.43	0.54	0.51	0.43	0.46
IPP with partner	0.65	0.91	0.33	0.79	0.54	0.68	0.28	0.64	0.62
Aggregate IPP	0.63	0.56	0.41	0.77	0.54	0.65	0.71	0.64	0.60
Correct plays with partner	7.51	12.13	9.23	1.89	9.97	9.23	10.21	7.74	7.87
Correct plays by partner	6.71	5.36	9.18	4.00	8.19	9.19	7.67	6.21	6.43
Aggregate correct plays	6.37	7.23	5.29	2.29	6.69	6.06	4.41	7.74	7.80
Correct plays by all partners	6.81	5.48	4.91	3.31	5.33	5.67	3.28	6.21	6.55
Mistakes with partner	1.77	1.36	1.19	0.27	0.89	1.05	0.71	0.98	1.13
Mistakes by partner	0.00	0.00	0.08	1.64	0.92	0.54	0.55	0.73	0.67
Aggregate mistakes	1.15	1.39	1.81	0.69	1.43	1.22	1.28	0.98	1.11
Mistakes by all partners	0.72	0.72	0.66	0.70	0.73	0.75	0.66	0.73	0.73
Games bombed with partner	0.04	0.01	0.01	0.38	0.04	0.02	0.07	0.15	0.15
Games bombed in aggregate	0.20	0.36	0.66	0.09	0.42	0.35	0.30	0.15	0.15

Table 3: Behavioral Evaluation of Rainbow agents. Metrics "with partner" and "Aggregate" always refer to the Rainbow agent's own behavior, either with their training partners or across all seven match-ups. "by partner" refers to the behavior of the training partners of that agent in its match-up with the agent (e.g. the behavior of $IGGI$ when playing with R_{IGGI}). "by all Partners" refers to the aggregate behavior of the other player across all match-ups (including those with the training partners). Comm is short for Communicativeness. Data was collected in the same sets games as table 2. The same metrics were also collected for match-ups involving only rule-based agents and are available in our repository, but are not shown for space considerations.

esting case. When paired with $IGGI$, it achieves the highest score of any pairing between a Rainbow agent and a rule-based agent (18.25), but has the lowest self-play score of any Rainbow agent and aggregate score comparable to R_{Flawed}!

With regards to the agents below the dashed line, R_{SP} achieves an average self-play score of 19.53, similar to the scores reported in (Bard et al. 2020), but with poor aggregate performance. Independently trained instances of R_{SP} tend to play well with each other, as remarked by (Bard et al. 2020) and reproduced in (Canaan et al. 2020a), so we do not linger on this topic here.

The R_{All} and R_{All+SP} agents achieve good scores with all partners except $Flawed$ (an aggregate of 13.46 and 14.54 respectively), as well as decent self-play scores (12.89 and 15.77). This differs from the result of (Canaan et al. 2020a) where the corresponding score for R_{all} (with fewer training steps) was 5.62.

Behavioral Analysis

Before looking at the behavioral data, an examination of the rules that compose the $Internal$ and $IGGI$ agents can give us some insight as to why $R_{Internal}$'s strategy generalizes well to agents not seen during training and why R_{IGGI}'s does not. $IGGI$ is the only of the six rule-based agents that never gives a hint that does not involve a playable card (apart from a rare edge case). A good strategy for playing with $IGGI$ would probably favor immediately playing hinted cards, even when lacking information, which might backfire with other rule-based agents, all of which can give hints even if their partner lacks a playable card.

$Internal$, on the other hand, is the only rule-based agent that does not keep track of past hints and has no bias for hinting at rank over color (or vice-versa) nor at hinting about a card in a specific slot if it is not playable. Therefore, while $Internal$ prefers to hint at playable cards, its remaining hints might seem very random. A good strategy for playing with $Internal$ might thus be more focused on reasoning about grounded information. This might explain why agents trained with $Internal$ play so well with others.

While we collected individual behavioral metrics for each individual match-up, table 3 summarizes the results for the Rainbow agents. The most surprising result is that all Rainbow agents have IPP below 1, which means they often play cards that were never hinted by their partners. This is much lower than the IPP of the rule-based agents themselves, between 1.5 and 2. While we were surprised by the IPP of all Rainbow agents, R_{SP} and R_{IGGI} have the lowest values for this metrics at 0.28 and 0.33, respectively, when playing with their training partners. This means over half the cards they play have never been hinted at. While we can not necessarily infer causality, these are also the two agents with the lowest aggregate scores other than R_{Flawed}.

R_{IGGI} is also the one that makes the most mistakes in the aggregate (1.81), which reinforces the hypothesis that it might be blindly trusting that every card hinted by the other player is playable. R_{SP} does not make many mistakes in the aggregate compared to other Rainbow agents, so it is possible that the drop in performance might be simply due to failing to realize that certain cards are playable.

While low IPP might explain why R_{IGGI} and R_{SP} have low aggregate scores, it doesn't explain why $R_{Internal}$'s is so high, as its IPP is similar to various other Rainbow agents. A distinguishing characteristic of $R_{Internal}$ is that it sees the highest number of correct plays made by all partners out of all Rainbow agents (6.81). In other words, $R_{Internal}$ seems to have learned to give hints that benefit the other player in all match-ups, not just $Internal$.

VDB and $Piers$ are the only rule-based agents that use probabilistic rules to play cards. It should be possible for an agent to learn to give them hints in a way that they would play cards very efficiently, and indeed R_{VDB} and R_{Piers} see very high numbers of correct plays by its training partners (9.19 and 8.19 per match, on average). However, this does not transfer to games with other agents (5.67 and 5.33).

R_{Outer} also does not see its companions play many correct cards per game (5.36 with $Outer$ and 5.48 in aggregate), which is curious as both $Outer$ and $Internal$ have the same "Play" rules. On the other hand, R_{Outer} plays the most correct cards with outer (12.13) than any Rainbow agent in their respective training match-ups. R_{Outer} also has the lowest communicativeness (0.38) except for R_{Flawed}. What we suspect is that $Outer$ gives hints in such a helpful manner that R_{Outer} learns a strategy that focuses more on reacting to its partners' hints and playing cards than actually giving hints. The fact that $Internal$ is a harder partner to decypher, with its randomized, memory-less hints, might reward the agent for learning a more balanced strategy between hints and plays, which would transfer better to other match-ups.

Discussion

While obviously more research is needed, we may allow ourselves to speculate a little about the implications of these results for teaching transferably collaborative behavior. It seems that the right amount of reliability is an important factor. The contrast between $IGGI$ and $Internal$ is instructive here; agents trained with $IGGI$ play very well with that agent but that policy barely transfers at all; in contrast, the less predictable $Internal$ agent apparently instils a policy in its partner that transfers well. However, a highly unpredictable agent such as $Flawed$ leads its partner to develop a policy that doesn't work well with any partner.

Some ways to continue this line of research involve: (1) training other types of neural network controllers, including recurrent ones such as SAD (Foerster et al. 2018), under similar regimes, (2) creating variants of the rule-based agents to test hypotheses about what makes an agent a good partner for learning a general policy, (3) evolving agents or sets of agents specifically for the purpose of being a good partner in this sense.

Acknowledgments

Rodrigo Canaan, Andy Nealen and Julian Togelius gratefully acknowledge the financial support from Honda Research Institute Europe (HRI-EU).

References

Bard, N.; Foerster, J. N.; Chandar, S.; Burch, N.; Lanctot, M.; Song, H. F.; Parisotto, E.; Dumoulin, V.; Moitra, S.; Hughes, E.; et al. 2020. The hanabi challenge: A new frontier for ai research. *Artificial Intelligence* 280:103216.

Bauza, A. 2010. Hanabi. Asmodée Èditions.

Bellemare, M. G.; Dabney, W.; and Munos, R. 2017. A distributional perspective on reinforcement learning. In *Proceedings of the 34th International Conference on Machine Learning-Volume 70*, 449–458. JMLR. org.

Bouzy, B. 2017. Playing hanabi near-optimally. In *Advances in Computer Games*, 51–62. Springer.

Canaan, R.; Togelius, J.; Nealen, A.; and Menzel, S. 2019. Diverse agents for ad-hoc cooperation in hanabi. In *2019 IEEE Conference on Games (CoG)*, 1–8. IEEE.

Canaan, R.; Gao, X.; Chung, Y.; Togelius, J.; Nealen, A.; and Menzel, S. 2020a. Evaluating rl agents in hanabi with unseen partners. *AAAI'20 Reinforcement Learning in Games Workshop*.

Canaan, R.; Gao, X.; Togelius, J.; Nealen, A.; and Menzel, S. 2020b. Generating and adapting to diverse ad-hoc cooperation agents in hanabi. *arXiv preprint arXiv:2004.13710*.

Cox, C.; De Silva, J.; Deorsey, P.; Kenter, F. H.; Retter, T.; and Tobin, J. 2015. How to make the perfect fireworks display: Two strategies for hanabi. *Mathematics Magazine* 88(5):323–336.

Foerster, J. N.; Song, F.; Hughes, E.; Burch, N.; Dunning, I.; Whiteson, S.; Botvinick, M.; and Bowling, M. 2018. Bayesian action decoder for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1811.01458*.

Hessel, M.; Modayil, J.; Van Hasselt, H.; Schaul, T.; Ostrovski, G.; Dabney, W.; Horgan, D.; Piot, B.; Azar, M.; and Silver, D. 2018. Rainbow: Combining improvements in deep reinforcement learning. In *Thirty-Second AAAI Conference on Artificial Intelligence*.

Hu, H., and Foerster, J. N. 2019. Simplified action decoder for deep multi-agent reinforcement learning. *arXiv preprint arXiv:1912.02288 to be presented at ICLR 2020*.

Hu, H.; Lerer, A.; Peysakhovich, A.; and Foerster, J. 2020. Other-play" for zero-shot coordination. *arXiv preprint arXiv:2003.02979*.

Lerer, A.; Hu, H.; Foerster, J.; and Brown, N. 2019. Improving policies via search in cooperative partially observable games. *arXiv preprint arXiv:1912.02318 to be presented at AAAI 2020*.

Mouret, J.-B., and Clune, J. 2015. Illuminating search spaces by mapping elites. *arXiv preprint arXiv:1504.04909*.

Osawa, H. 2015. Solving hanabi: Estimating hands by opponent's actions in cooperative game with incomplete information. In *AAAI workshop: Computer Poker and Imperfect Information*, 37–43.

Stone, P.; Kaminka, G. A.; Kraus, S.; and Rosenschein, J. S. 2010. Ad hoc autonomous agent teams: Collaboration without pre-coordination. In *Twenty-Fourth AAAI Conference on Artificial Intelligence*.

van den Bergh, M. J.; Hommelberg, A.; Kosters, W. A.; and Spieksma, F. M. 2016. Aspects of the cooperative card game hanabi. In *Benelux Conference on Artificial Intelligence*, 93–105. Springer.

Walton-Rivers, J.; Williams, P. R.; Bartle, R.; Perez-Liebana, D.; and Lucas, S. M. 2017. Evaluating and modelling hanabi-playing agents. In *Evolutionary Computation (CEC), 2017 IEEE Congress on*, 1382–1389. IEEE.

Wu, J. 2016. State of the art hanabi bots + simulation framework in rust. https://github.com/WuTheFWasThat/hanabi.rs. Access: 05/14/2018.