Drafting in Collectible Card Games via Reinforcement Learning

Ronaldo e Silva Vieira
Student
Dep. de Ciência da Computação
Univ. Federal de Minas Gerais
Belo Horizonte, Brazil
ronaldo.vieira@dcc.ufmg.br

Anderson Rocha Tavares
Co-advisor
Instituto de Informática
Univ. Federal do Rio Grande do Sul
Porto Alegre, Brazil
artavares@inf.ufrgs.br

Luiz Chaimowicz
Advisor
Dep. de Ciência da Computação
Univ. Federal de Minas Gerais
Belo Horizonte, Brazil
chaimo@dcc.ufmg.br

***Abstract*—Collectible card games (CCGs), such as *Magic: the Gathering* and *Hearthstone*, are played by tens of millions of players worldwide and are challenging for humans and artificial intelligence (AI) agents alike. To win, players must be proficient in two interdependent tasks: deck-building and battling. We present three deep reinforcement learning approaches for deck-building in the *arena mode* that differ in considering past information when making new choices. We formulate the problem in a game-agnostic manner and perform experiments on *Legends of Code and Magic*, a CCG designed for AI research. Results show that our trained draft agents outperform the currently best draft agents of the game and do so by building very different decks. Moreover, a *Strategy Card Game AI* competition participant improves from tenth to fourth place when using our best draft agent to build decks. This work is a step towards strong and fast game-playing AI for CCGs, one of the current academic AI milestones that would enable thorough playtesting of new cards before they are released – a long-standing problem in the CCG industry.**

***Index Terms*—reinforcement learning, collectible card game, deck-building**

I. Introduction

Collectible card games (CCG), such as *Magic: the Gathering* and *Hearthstone*, are adversarial turn-taking two-player games played by tens of millions of players worldwide. Their usually large and intricate set of rules make them more challenging than traditional card games for both human players and artificial intelligence (AI) agents. In fact, despite the recent breakthroughs in game-playing AI that led to superhuman performance on games like Go [1] and Texas Hold'em Poker [2], currently, humans remain unchallenged in CCGs.

The advent of human- or superhuman-level game-playing agents for collectible card games would enable more challenging opponents for amateur and professional players and more efficient playtesting tools. Such tools could have a critical role in game-balancing, one of the toughest challenges the CCG industry faces. Although CCG designers conduct careful qualitative playtesting processes before releasing new cards [3], banning or weakening some of them due to unforeseen imbalances in the game is a fairly common event.

From an AI standpoint, the act of playing a CCG is divided in two interdependent tasks: deck-building and battling. Many deck-building and battling modes exist, and different AI solutions may be adequate for each of them. In this thesis, we tackle deck-building in the *arena* mode of CCGs, also known as *drafting*. Despite its popularity among players, to the best of our knowledge, there is currently a single published work on the topic [4].

Our overall research goal is to investigate whether deep reinforcement learning (DRL) methods can achieve competitive performance in arena mode compared to the current state-of-the-art. Our specific research goals are: (i) to encourage further research on the topic by developing a game-agnostic methodology, reproducible experiments, and reusable open-source tools; (ii) to advance the state-of-the-art of drafting, by presenting AI agents that achieve greater win rates than the best ones so far; and (iii) to study the differences between drafting agents through extensive experiments.

We formulate drafting as a Markov decision process and propose three DRL approaches trained in self-play that differ on handling information from previously drafted cards when making a new choice. Performance is measured via the win rate of the built decks when used by fixed battling agents in a large set of matches. We use *Legends of Code and Magic* (LOCM) [5] as a testbed for our experiments, and reimplement the game as OpenAI Gym [6] environments to speed up experiments and facilitate the use of DRL algorithms. Nevertheless, we define our approaches in an entirely game-agnostic manner to foster their application to other CCGs.

Our resulting drafting agents significantly outperformed the state-of-the-art for LOCM when paired with two fixed battling agents of different skill levels. Results also show that our drafters built very different decks than those of the other agents. Moreover, we show that a participant of the *Strategy Card Game AI* (SCGAI) competition, held at the IEEE CoG 2019 conference, improves from tenth to fourth place when using our best drafting agent (from 34% to 46.1% in win rate).

II. The drafting problem

Most commercial CCGs feature an arena mode. While there are differences among them, they all share the same core structure: as soon as a player signs-up, the game starts a draft in which the player builds a deck incrementally during n turns. At each draft turn, the game presents k cards sampled without replacement from a pool of possible cards, and the player must choose one of them to add to their deck. At the end of the draft, the player has a deck containing n cards. The values of k and n and the probability distribution used to sample the cards are game-specific factors.

After the draft, the player uses the n-card deck to battle other players. The player is then rewarded proportionally to the number of battles won. We define the problem of choosing cards during a draft in arena mode aiming to maximize the win rate in the subsequent battles as a (k, n)-*draft*.

To tackle the (k, n)-draft problem with reinforcement learning, we formulate it as a Markov decision process (MDP). We define two different state representations: **history-aware**, which considers the k presented cards in the current turn t plus the $t-1$ cards chosen so far in the previous turns; and **history-oblivious**, which considers only the k presented cards. At any state, the possible actions are choosing either of the k cards, i.e., $\mathcal{A} = \{1, 2, \ldots, k\}$. The transition function follows the draft rules, and rewards are proportional to the win rate of the built deck in battles played by a fixed battle agent. We define the MDP in greater detail in the thesis [7, Sect. 4.1].

Finding a solution to a (k, n)-draft MDP with either state representation is equivalent to developing a strategy to draft in the arena mode of a CCG with the specified k and n parameters. In other words, an agent can draft in arena following any policy $\pi : \mathcal{S} \times \mathcal{A} \to [0, 1]$ that maps every state in $\mathcal{S}$ to a probability distribution over every action in $\mathcal{A}$.

III. Our approaches

We propose three approaches that differ in state representation and the type of neural network used by the deep reinforcement learning algorithm. The first approach, **History**, uses a multilayer perceptron (MLP) network [8] and the history-aware state representation. This representation enumerates all previously chosen cards, enabling the DRL agent to leverage card synergies when drafting new cards. The second approach, **LSTM**, uses the history-oblivious states but relies on a layer of long short-term memory (LSTM) [9] units to retain information about past picks without explicitly enumerating them. **Immediate**, the last approach, uses the history-oblivious states and an MLP network, not considering past picks but reasoning in a much smaller state space. Despite the differences, all approaches share the same training methodology.

In each draft turn, we convert the game state to a numeric vector containing the relevant features of all cards in the state normalized to the range of $[-1, 1]$, and then feed it to the network. Considering a card feature extraction process which maps a single card in the game state to p features, History's network input consist of $p(k + n)$ values, while Immediate and LSTM's consist of pk values. The network then outputs its policy, which we use to choose a card back in the game. Fig. 1 illustrates this process. Once the deck is complete, a selected battling agent plays matches with the deck, returning a reward signal proportional to its win rate.

In the thesis, we discuss applying our methodology to arena-like modes of popular CCGs, including values of k and n, feature extraction and additional challenges [7, Sect. 4.4].

IV. Experiments and results

We instantiate our methodology on *Legends of Code and Magic* (LOCM) [5], a collectible card game designed for AI research. Playing LOCM is equivalent to a $(3, 30)$-draft where decks are used in a single battle, and both players are presented with the same cards during the draft. Since there is a known advantage of playing first in battles [10], drafters in LOCM's literature use different strategies depending on whether the battler will play first or second. Thus, we trained two separate instances of the DRL algorithm specialized at, respectively, drafting for first and second players. We use a self-play variant frequently used when the agents are in asymmetric roles [11].

We chose the Proximal Policy Optimization (PPO) [12] algorithm to train our approaches after preliminary tests with other DRL algorithms using the *stable-baselines* [13] library. Each training session consists of 30,000 LOCM matches, where we stop and evaluate our agents 12 times with 1,000 LOCM matches. We run all matches using our reimplemented game engine [14] and OpenAI Gym [6] environments, and repeat all experiments with two different fixed battling agents of varying skill levels, named *max-attack* (MA) and *greedy* (GR). For each combination of our three approaches (History, LSTM, and Immediate) and battlers (MA and GR), we optimized the network architecture and PPO hyperparameters via the Tree of Parzen Estimators [15] using the *hyperopt*[1] library.

In our first experiment, we compared the performance of our three approaches. With the win rates obtained in the evaluations, we compiled average learning curves for each combination [7, Fig. 5.1]. The results show better performance by the Immediate approach in all scenarios, followed by LSTM. Although History achieves better performance than LSTM early in training, it also seems to settle earlier, while the latter improves and eventually reaches better win rates. Along with results from further experiments yet to be published (using 1,000,000-match training sessions), this may suggest that LOCM's rules may be too simple to enable the emergence of relevant card synergies or that the companion battle agents might not be able to leverage those synergies.

Our second experiment evaluated our best resulting draft agents against two baseline and three state-of-the-art draft strategies. The baselines are a *random* agent, which chooses at random, and *max-attack*, which chooses the card with the highest attack power. The state-of-the-art agents are named *coac*, *closet-ai*, and *icebox*, past champions of LOCM-based AI competitions, which choose the highest-ranked card of a previously calculated card ranking. We carried out round-robin tournaments containing all drafting agents. Every pair of agents faced each other in 20,000 matches using the same random seeds and switching first and second players halfway.

We extracted win rates from every pairwise match-up. Fig. 2 shows the pairwise win rates (vs. a specific agent) and the average win rates (vs. all agents) in both round-robin tournaments. The results show that our approaches outperform all selected draft strategies using either battlers. As in the previous

[1]https://github.com/hyperopt/hyperopt

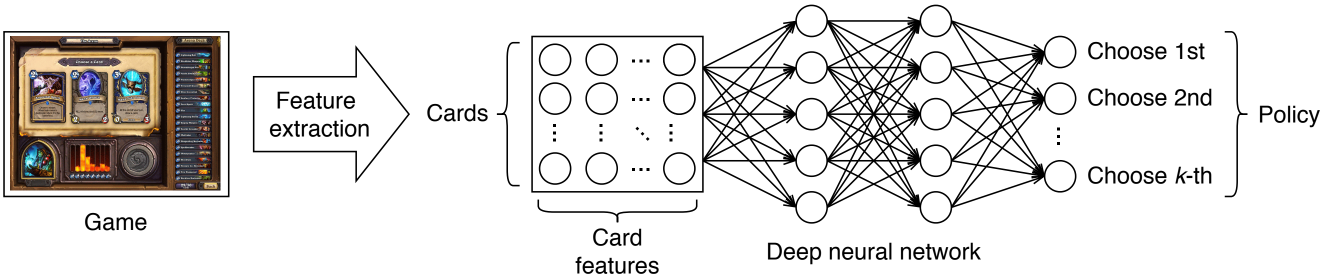

Fig. 1. The interaction between our agent and the game. From the game, we extract features of the cards in the current draft turn. The features of all cards contained in the state representation are concatenated and fed as input to a deep neural network, which outputs values from which we build a policy. This policy is then used to act in the game (i.e., choose a card).

	random	*max-atk*	*coac*	*closet-ai*	*icebox*	**History**	**LSTM**	**Immed.**	Avg.
random	50	45.83	45.31	40.18	34	30.11	24.31	19.51	34.18
max-atk	54.17	50	51.24	43.54	37.7	33.3	27.08	20.28	38.19
coac	54.7	48.76	50	43.12	37.96	34.28	27.56	21.54	38.27
closet-ai	59.82	56.46	56.88	50	44.06	39.5	31.94	23.36	44.57
icebox	66	62.3	62.04	55.94	50	44.86	38.82	30.04	51.43
History	69.9	66.7	65.72	60.5	55.14	50	42.88	37.98	56.97
LSTM	75.7	72.92	72.44	68.06	61.18	57.12	50	43.82	64.46
Immed.	80.49	79.72	78.46	74.14	67.24	62.02	56.18	50	71.18

(a) Round-robin tournament using the MA battler.

	max-atk	*random*	*closet-ai*	*icebox*	*coac*	**History**	**LSTM**	**Immed.**	Avg.
max-atk	50	41.67	38.08	35.86	28.84	27.78	25.76	18.84	30.98
random	58.34	50	45.87	45.39	36.73	36.51	34.5	28.06	40.77
closet-ai	61.92	54.14	50	49.48	41.02	39.82	37.64	28.66	44.67
icebox	64.14	54.61	50.52	50	43.34	40	39.04	30.16	45.97
coac	71.16	63.27	58.98	56.66	50	49.1	47.9	43.04	55.73
History	72.22	63.49	60.18	60	50.9	50	48.1	39.2	56.3
LSTM	74.24	65.51	62.36	60.96	52.1	51.9	50	42.62	58.53
Immed.	81.16	71.94	70.32	68.86	56.96	60.8	57.38	50	66.77

(b) Round-robin tournament using the GR battler.

Fig. 2. Performance of our agents (highlighted in bold) plus the baseline and state-of-the-art agents in two round-robin tournaments, each using a different battler. Each cell represents the win rate (in %) of the row's drafter on 20,000 matches against the column's drafter. In the last column, we show the average win rate of the drafters in the tournament. The agents are ordered in ascending order by average win rate.

experiment, Immediate achieved the best performance using both MA and GR, followed by LSTM and then by History. The differences in win rate between our agents and all others were statistically significant with $p < 0.0001$, as verified by paired t-tests, except for History and *coac* when using GR ($p = 0.253$).

Besides the win rate, other aspects show differences between our trained drafting agents and the selected baseline and state-of-the-art drafters. We measured the average mana curve[2] of the decks built by each agent in the tournaments [7, Fig. 5.3]. The results show that all our drafting agents, especially those trained alongside the GR battler, tend to favor cards with low resource costs. Also, there were no significant differences between the curves of the first and second player networks we trained ($p > 0.998$ on paired t-tests). Moreover, as expected, the selected agents produced very distinct curves.

We also measured the choice similarity between agents, i.e., the percentage of times that a pair of agents, when faced with the same card alternatives, agreed on their choice [7, Fig. 5.4]. Our trained agents shared most of the highest similarity values. However, an unexpected result was that the *icebox* and *max-attack* agents shared the highest similarity value, despite their dramatic difference in win rate. The *random* agent had about 33.33% of choice similarity with all agents since it chooses one of the three presented cards at random.

We also applied Principal Component Analysis [16] to reduce the high-dimensional choice vector of each agent to three dimensions and enable a visual representation of their card choices. We then applied K-means [17] and silhouette analysis [18] to separate the 3D points into clusters. In the resulting 3D plot [7, Fig. 5.5], our agents were placed near each other, but only those who shared the same battler in training remained in the same cluster. They were also far from all other selected agents.

Our third experiment reenacted the 2019 edition of the *Strategy Card Game AI* (SCGAI) competition held at the IEEE CoG conference. It reunites the state-of-the-art LOCM-playing bots (draft + battle) in a round-robin tournament and presents an opportunity to evaluate our draft strategies against opponents with diverse battling agents. Using the source code of the competition available at GitHub[3], we re-executed the tournament, using the *max-attack* battler twice: once paired with the original *max-attack* drafter, as in the original tournament, and once paired with our best drafting agent. Fig. 3 shows the performance of *max-attack*'s original and improved versions in the tournament. While the original bot ranked tenth (win rate of 34%), the improved version reached the fourth place (win rate of 46.1%) solely by drafting better, despite its very simple companion battling agent.

[2]The mana curve is a prevalent metric among players. It is a histogram that bins cards according to the amount of resource needed to play them.

[3]https://legendsofcodeandmagic.com/COG19

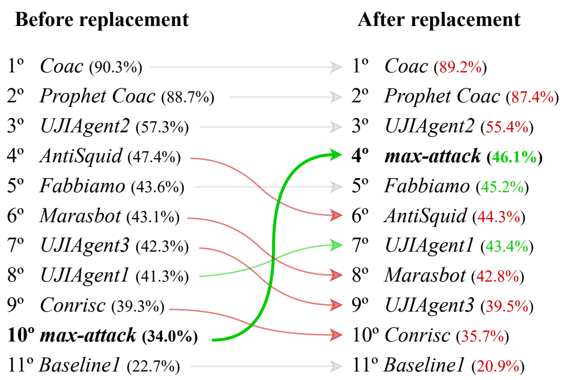

Fig. 3. Tournaments with agents from the IEEE CoG 2019 *Strategy Card Game AI* competition before and after the substitution of *max-attack*'s drafting agent.

V. Conclusion

This thesis tackles deck-building in arena mode of CCGs. We modeled the problem in a game-agnostic way and proposed three deep reinforcement learning approaches to tackle it. We measured the performance of our agents by observing the win rate of the decks they built in battles carried out by fixed battle agents in *Legends of Code and Magic*, a CCG designed for AI research. According to our experiments, our draft agents outperformed all selected baseline and state-of-the-art drafters when paired with two different fixed battle and did so by building very different decks that focused on low-cost cards.

Our agents learn drafting strategies without domain knowledge or labeled data, and once trained, their decision is fast, having no noticeable delay compared to the state-of-the-art agents who look up a ranking of cards. Furthermore, our drafters could handle unforeseen card sets because they rely on a feature-based card representation, in contrast with the current state-of-the-art agents, which rank the cards based on their IDs. Our best drafting agent also improved the performance of a participant of the 2019 edition of the *Strategy Card Game AI* (SCGAI) competition, showing that it can achieve good performance against a diverse set of unseen adversaries.

In general, all experiments suggested a positive response to our research question: deep reinforcement learning methods can achieve competitive drafting performance in collectible card games compared to the best works available. A surprising result was that considering past choices when choosing new cards did not improve the agents' performance.

Future work involves constructing a reinforcement learning battler and applying our methodology to more complex commercial CCGs, where considering past choices is fundamental.

VI. Contributions

We consider this thesis a step towards superhuman-level AI for collectible card games. It advances the state-of-the-art in the understudied subject of drafting, while fostering further work via its reproducibility, applicability to other games and open-source tools. We produced the following artifacts:

- A short paper published on **SBGames 2019**, with preliminary results [19].
- A full paper published on **SBGames 2020**, awarded as the **best paper** in the Computing Track [20].
- A paper submitted to **Entertainment Computing**, extending [20] with additional experiments and improved results (currently under review).
- A submission to the **Strategy Card Game AI Competition**, held at the IEEE CEC 2020 conference, which achieved the third place.[4,5]
- A fully-working open-source game engine for LOCM, and OpenAI Gym environments encompassing LOCM's drafting and battling tasks [14].
- A collection of ready-to-use competitive drafting strategies along with instructions on how to use them.[6]

References

[1] D. Silver *et al.*, "Mastering the game of go without human knowledge," *Nature*, vol. 550, no. 7676, 2017.
[2] N. Brown and T. Sandholm, "Superhuman AI for multiplayer poker," *Science*, vol. 365, no. 6456, pp. 885–890, 2019.
[3] B. Hawley, "Checking in with play design," 2019. [Online]. Available: https://magic.wizards.com/en/articles/archive/play-design/checking-play-design-2019-04-24
[4] J. Kowalski and R. Miernik, "Evolutionary approach to collectible arena deckbuilding using active card game genes," in *CEC 2020*. IEEE, 2020.
[5] J. Kowalski and R. Miernik, "Legends of Code and Magic," https://legendsofcodeandmagic.com, 2018, accessed: 2020-03-27.
[6] G. Brockman *et al.*, "OpenAI Gym," *CoRR*, vol. abs/1606.01540, 2016.
[7] R. Vieira, A. R. Tavares, and L. Chaimowicz, "Drafting in collectible card games via reinforcement learning," Master's thesis, UFMG, 2020.
[8] S. Pal and S. Mitra, "Multilayer perceptron, fuzzy sets, and classification." *IEEE transactions on neural networks*, vol. 3, no. 5, 1992.
[9] S. Hochreiter and J. Schmidhuber, "Long short-term memory," *Neural computation*, vol. 9, no. 8, pp. 1735–1780, 1997.
[10] CodinGame community, "Legends of Code & Magic (CC05) - Feedback & Strategies," https://www.codingame.com/forum/t/legends-of-code-magic-cc05-feedback-strategies/50996/63, 2018.
[11] B. Baker, I. Kanitscheider, T. M. Markov, Y. Wu, G. Powell, B. McGrew, and I. Mordatch, "Emergent tool use from multi-agent autocurricula," in *ICLR 2020*, 2020.
[12] J. Schulman *et al.*, "Proximal policy optimization algorithms," *CoRR*, vol. abs/1707.06347, 2017.
[13] A. Hill *et al.*, "Stable baselines," https://github.com/hill-a/stable-baselines, 2018.
[14] R. Vieira, L. Chaimowicz, and A. R. Tavares, "OpenAI Gym environments for Legends of Code and Magic," https://github.com/ronaldosvieira/gym-locm, 2019.
[15] J. Bergstra, D. Yamins, and D. D. Cox, "Making a science of model search: Hyperparameter optimization in hundreds of dimensions for vision architectures," in *ICML 2013*. JMLR.org, 2013.
[16] I. T. Jolliffe and J. Cadima, "Principal component analysis: a review and recent developments," *Phil. Trans. A*, vol. 374, no. 2065, 2016.
[17] S. P. Lloyd, "Least squares quantization in PCM," *IEEE Trans. Inf. Theory*, vol. 28, no. 2, 1982.
[18] P. J. Rousseeuw, "Silhouettes: a graphical aid to the interpretation and validation of cluster analysis," *Journal of computational and applied mathematics*, vol. 20, pp. 53–65, 1987.
[19] R. Vieira, L. Chaimowicz, and A. R. Tavares, "Reinforcement learning in collectible card games: Preliminary results on Legends of Code and Magic," in *SBGames 2019*, 2019.
[20] R. Vieira, A. R. Tavares, and L. Chaimowicz, "Drafting in collectible card games via reinforcement learning," in *SBGames 2020*. IEEE, 2020.

[4] https://legendsofcodeandmagic.com/CEC20/
[5] https://github.com/ronaldosvieira/reinforced-greediness
[6] https://github.com/ronaldosvieira/gym-locm/tree/1.0.0/gym_locm/trained_models