

# Heisenbot: A Rule-Based Game Agent for Gin Rummy

Matthew Eicholtz, Savanna Moss, Matthew Traino, Christian Roberson

Florida Southern College

111 Lake Hollingsworth Drive, Lakeland, Florida 33801

meicholtz@flsouthern.edu, smoss@mocs.flsouthern.edu, mtraino@mocs.flsouthern.edu, croberson@flsouthern.edu

## Abstract

Games are an excellent tool for undergraduate research in artificial intelligence because they typically have clear objectives, a limited action space, and well-defined constraints. Nonetheless, games involving chance and imperfect information offer unique challenges for optimizing gameplay. In this paper, we analyze one such card game, gin rummy, and propose an artificial intelligence player based on empirically driven strategies. Our approach separates gameplay into three disjoint policies for drawing, discarding, and knocking. On each turn, decisions are influenced by offensive considerations as well as defensive moves. Tournament-style simulations enable us to determine statistically which combination of policies achieves the highest win rate. Our resulting player, dubbed Heisenbot, is competitive against strong baseline strategies.

## Introduction

Games can serve as a solid framework for students looking to participate in undergraduate research. They provide well-defined problems with clear rulesets and a limited set of decisions, which are both useful traits for developing artificial intelligence agents. Gin rummy is a particularly interesting game in this regard because there has not been significant research developed in this space. While the rules of the game are fairly simple, the strategy involved has much depth and complexity. In this paper, we propose a novel artificial intelligence player capable of competing in an ELO tournament against other players.

This paper is organized into several sections. The Background section provides an explanation of the rules for gin rummy and an overview of artificial intelligence in games. In the Methods section, we examine the three major policies needed to implement an artificial intelligence player: a draw policy, a discard policy, and a knock policy. We also describe implementation details for our simulation setup. Next, the Results section provides a breakdown of experimental tests and analysis. Finally, we discuss conclusions and considerations for future work.

Copyright © 2021, Association for the Advancement of Artificial Intelligence (www.aaai.org). All rights reserved.

## Background

### Gin Rummy Rules and Terminology

Gin rummy is a non-cooperative, symmetric, sequential, zero-sum, imperfect information game played between two players. The goal is to score 100 points by forming groups of three or more cards before the opponent over several hands of play. To start each hand, ten cards from a standard 52-card deck are dealt to each player and one card is placed face up to form a discard pile. Players alternate turns drawing either the top card from the discard pile or from the draw pile and then discarding a card from their hand. A player cannot discard a card if it was drawn from the top of the discard pile on that turn. The dealer switches each hand, and if the dealer declines the face-up discard on the first turn, then the opponent can opt to take the discard and start the round themselves.

Although there are several variants of gin rummy, here are the key terms and scoring mechanisms used in the current EAAI Undergraduate Research Challenge:

**Melds** During play, each player attempts to form combinations of three or more cards known as melds. There are two meld types in gin rummy: (1) *sets* consist of cards of the same rank and different suits; and (2) *runs* contain cards of the same suit with adjacent ranks. For the purpose of this game aces are considered rank one and are adjacent to twos, but not to kings. There are 65 unique meld sets and 240 unique meld runs of ten or fewer cards, for a total of 305 possible melds. Cards cannot belong to more than one meld in a hand, even though there may exist multiple possible collections of melds for a given hand. Knowing the state of a card (i.e. discarded, in our opponent's hand) can have important implications for possible melds that are available. For example, if the 4 of clubs is buried in the discard pile, then 80% of club sets and 52% of club runs are eliminated from the game. Sevens contribute to the most run melds (160 out of 240), whereas aces and kings contribute to the fewest run melds (32 out of 240).

**Deadwood** Any unmelded cards in a player's hand are considered deadwood. Deadwood cards are counted against the player and the sum of all deadwood cards is the score of the hand. Aces have a deadwood value of one, number cards have a deadwood value equal to their rank, and face cards all have a deadwood value of ten.

**Knocking** If a player has a total deadwood score of ten or less, that player is able to knock which signals the end of the round. At this point the opponent is given an opportunity to layoff any cards in their deadwood that can extend melds in the player’s hand; these cards do not count against the final deadwood score for the opponent. If all ten cards in the knocking player’s hand contribute to melds, then the player goes gin. In this case, the opponent may not layoff any deadwood cards.

**Scoring** If a player has gone gin, they score 25 points plus the total of any remaining deadwood in the opponent’s hand. If the player has knocked instead, both players total their deadwood. If the knocking player has a deadwood score equal to or less than the opponent, the player scores the value of the difference in their deadwood totals. If the knocking player has a higher deadwood score than the opponent, then the opponent has successfully undercut the player and receives both an undercut bonus of 25 points plus the difference in deadwood scores.

**Notation** In this paper, we will use two-character rank-suit notation to describe specific cards in a standard 52-card deck. Card ranks include {A, 2, 3, 4, 5, 6, 7, 8, 9, T, J, Q, K} and suits include {C, D, S, H}. So, 7H is the Seven of Hearts and AS is the Ace of Spades, for example. A sample sorted ten-card deal is [2C, QC, 3D, 6D, KD, AS, 3S, 9H, TH, KH].

## Artificial Intelligence in Games

Artificial intelligence has been studied in the context of games for many decades. Arguably the most successful implementations have focused on perfect information games like Chess (Hsu 1999; Silver et al. 2018), and Go (Silver et al. 2017). In recent years, a growing body of work examines imperfect information games like Poker (Moravčík et al. 2017) and StarCraft II (Vinyals et al. 2019). These breakthroughs in artificial intelligence research have come through various techniques such as reinforcement learning from games of self-play (Silver et al. 2017) and deep convolutional neural networks (Maddison et al. 2014).

Gin rummy is an example of an imperfect information game because each player does not know what cards their opponent holds. This makes prediction of the game state nontrivial and determination of the value of a card (for drawing or discarding purposes) quite challenging. To our knowledge, there is very limited existing literature on computer strategies for gin rummy. One work (Kotnik and Kalita 2003) explored reinforcement learning and evolutionary techniques in this context, inspired by the success of an AI player for backgammon (Tesauro 1995). Their research presented several challenges including long training times, human-tuned parameters, and tournament agents that may not generalize well. Our goal in this paper is to add to this line of research by testing a number of simple rule-based policies against each other in tournament-style fashion and analyzing which strategies achieve the most success.

## Methods

While there are many potential approaches to playing gin rummy, any strategy can be boiled down to three recurring, sequential decisions: (1) drawing a card, (2) discarding a card, and (3) choosing to knock or not. For our AI player, we consider these decisions independently. Choosing to knock is the only instance that should influence prior decisions (in this case, discarding the highest deadwood), but our discard strategies are biased toward higher deadwood in general, so independence is not a harmful assumption. In this section, we describe each of the three policies in detail.

### Draw Policy

Determining which card to draw at the start of a turn is a binary decision: a player must either take the known face-up discard or draw from the remaining deck of face-down cards. This may seem like an easy decision to make in general (e.g. draw the discard if it helps your hand, otherwise draw from the deck), yet it can have critical implications for the outcome of a hand for many reasons. First, choosing to draw the discard enables the opponent to know with certainty *at least one* of the cards in the player’s hand (and perhaps more depending on game state). In general, we argue that it is advantageous for all cards in a player’s hand to be unknown to the opponent, especially early in the round. Second, declining the discard automatically removes several potential melds from the round. During early stages of the hand, the range of ineligible melds varies from 12 (4 sets and 8 runs for aces) to 44 (4 sets and 40 runs for sevens). However, later in the game, the top discard may already be ineligible for most remaining melds; in fact, this is often the reason it gets discarded in the first place. Third, drawing from the top of the deck introduces chance into a player’s hand. At the start of a round, there are 31 cards in the draw pile, so there is a 1/31 probability of drawing a specific card. Later in the round, the probability of drawing unseen cards increases significantly to a maximum of 1/3.

The balance of these nuanced effects makes it difficult to determine an optimal draw policy. There are many potential factors at play including the round, number of turns taken, current deadwood, current points, permanently inaccessible cards, knowledge of the opponent’s possible melds, and the value of the face-up card in relation to the player’s current hand. In our tournament simulations, we focus on the last feature listed and test the following policies:

**Random** A *random* draw policy selects a card randomly, choosing the top discard or the top card of the deck with equal probability. Alternatively, the decision could remain random, but with distinct probabilities so that a player favors drawing from the deck, for example. While random decision-making may seem foolish (and we hypothesize that drawing randomly with equal probability is a suboptimal strategy), it does have the advantage of keeping the opponent guessing as to what the player’s strategy is. A defensive-minded opponent may choose to keep high deadwood longer if it thinks the player needs it based on a discard that was drawn early. Also, some studies have shown that randomization (to some degree) can be a key component of optimal

player strategy in games with imperfect information (Koller and Pfeffer 1995; Kuhn 1950).

**Immediate Value** A draw policy that prioritizes *immediate value* only chooses to take the face-up discard if it instantly creates or contributes to a meld in hand. Otherwise, draw from the face-down deck. This is the strategy of the experimental simple player provided by the organizers of the research challenge. The advantages of this policy are that: (1) nonrandom cards selected are guaranteed to lower our deadwood and increase our chances for winning a hand; and (2) knowledge gained by the opponent about cards in hand is less important because adjacent cards, while still desirable, are not required for melds. Nonetheless, an immediate value draw policy may be shortsighted due to declining discards that may have provided value on subsequent turns while simultaneously eliminating melds from the round (e.g., passing on KH with KS in hand, then drawing KC randomly from deck).

**Future Value** A draw policy that prioritizes *future value* takes into account the potential benefit of the top discard when deciding where to draw from. Like the immediate value policy, this strategy still picks up the discard if it immediately creates or contributes to a meld in hand. In addition, the future value policy always draws the discard if it is an ace or two because such cards represent the lowest possible deadwood. Having low deadwood, even if unmeldable, is a good strategy because it increases the probability of knocking (even with more deadwood *cards*), increases the probability of undercutting if the opponent knocks, or at the very least reduces the points earned by a knocking opponent. Finally, the future value policy draws discards if they form triangles early in the game. A triangle is a group of three cards that include two cards of the same rank and different suits with a third card that is adjacent to one of the other cards (e.g., [2C, 2D, 3D], [6H, 7H, 7D]). A triangle offers multiple options for drawing a single card and completing a meld with two of the triangle cards. However, triangles become less important as the game progresses, so we limit drawing a triangle card to the first five moves of the round.

**Heisenbot Draw Strategy** For our Heisenbot player, we incorporate the future value draw policy as described above, which prioritizes existing melds, triangles, and low deadwood cards.

### Discard Policy

Of the three policies needed to implement the AI player, discarding is perhaps the most complex to implement. Discards can be made offensively as a player attempts to better the value of their hand, or defensively to try and avoid handing useful cards to the opponent. Other factors include the phase of the game (e.g. early, middle, or late in game), state information for cards in the deck, and the quality of the player's hand. Here, we explore some of the strategies researched for our player.

**Offensive Play** When playing an offensive-style game, the focus for discards is to remove cards that provide the least

value for the player's hand and improve the player's position. This does not take into consideration the impact of a discard on the hand of an opposing player. There are several possible metrics that can be used to assess the value of cards for discard: (1) deadwood value of the card, (2) number of possible melds for the card, and (3) involvement in triangles in hand, to name a few.

**Defensive Play** When playing a defensive-style game, the primary concern is how each discard can impact the opponent's hand since they will have an opportunity to pick up the discard on the next turn. A defensive player will try to avoid discarding low deadwood since the value of such cards is helpful even in a losing hand. Furthermore, if the opponent has one or more melds with low cards, discarding low deadwood can quickly shift a game in favor of the opponent. Also, defensive players will track any known cards picked up by the opponent and avoid discarding suit- or rank-adjacent cards so that the opponent has difficulty completing melds.

**Safety Counts** One strategy for discarding is to evaluate possible discards and count the number of possible ways that card could be melded into an opponent's hand. For example consider the six of diamonds (6D). There are six possible melds this card can be involved in: [4D, 5D, 6D], [5D, 6D, 7D], [6D, 7D, 8D], [6D, 6S, 6C], [6D, 6S, 6H], and [6D, 6C, 6H]. If any of these cannot be formed without a card from the player's hand or the discard pile they are not possible for the opponent if the six of diamonds is discarded. This count of possible melds the opponent can form with the prospective discard is the safety count. The best card to discard is a card with a safety count of zero since there are no possible melds that can be formed from that card. Lower safety counts are considered "safer" cards to discard.

**Prognostication** Another important factor in defensive play is trying to reliably know what cards are in the opponent's hand. Cards that are in the player's hand or in the discard are not possible cards for the opponent to hold. Any card that the opponent picks up from the discard is definitely held by the opponent. But this information alone isn't enough to give great insight into the opponent's hand because most opponents do not regularly pick up face-up cards from the discard pile. In order to more quickly try to identify probable and improbable cards some sort of prediction model can be used to guess cards in the opponent's hand.

One simple model for predicting opponent cards is to track a likelihood value for each card in the deck with respect to the opponent holding it in their hand. Likelihood is an integer ranging from -5 to 5, with -5 meaning it is impossible for the opponent to hold the card and 5 meaning the card is known to be in the opponent's hand. This range of values was selected by trial and error over several iterations of development and is summarized in Table 1. Note that the magnitude of values is less important than the relative difference between discardable cards. All unknown cards start with a neutral value of 0. When the opponent picks up a card from the discard pile, that card is marked with a value of 5 because the opponent is known to be in possession of the

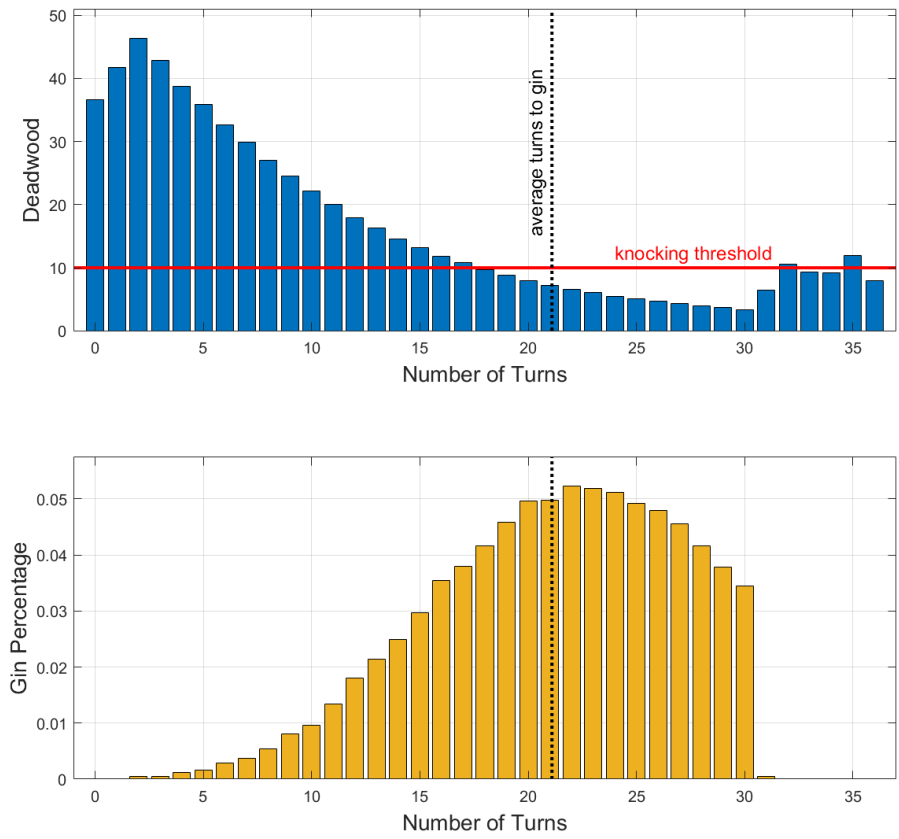


Figure 1: Simulation results for 10,000 games of the simple AI strategy against itself. Turns are counted together for both players, so 20 turns means each player played 10 times. Artifacts beyond 30 turns are due to forced knocking (if possible) beyond that point in the round to ensure the round is not invalidated due to too few cards remaining in the draw pile.

Likelihood	Description
-5	Impossible card for opponent to hold
-4 to -1	Unlikely card for opponent to hold
0	Neutral or unknown card
1 to 4	Likely card for opponent to hold
5	Known card for opponent

Table 1: Breakdown of prognostication values and meaning. Likelihood of a card may increase when an opponent draws from the discard pile and may decrease when an opponent discards a card.

card. In addition, neighboring cards with adjacent ranks or same rank and different suits are marked with an increased score. Cards in the same rank but a different suit and neighbors with the same suit but a rank difference of 1 are given a score increase of 2. Cards with the same suit but a rank difference of 2 are given a score increase of 1. Any cards that

are known to be held by the player or in the discard pile are skipped while marking since they are already labeled correctly as impossible or known.

It is also possible to gain insight about the likelihood of cards being in the opponent’s hand by the cards they discard. If a card is discarded, neighbors of the discard are marked as less likely to be held by the opponent. The discarded is marked as a -5 since the opponent is known to not possess it anymore. Cards in the same rank but a different suit and neighbors with the same suit but a rank difference of 1 are given a score decrease of 2 because they are less likely to be in the opponent’s hand. Cards with the same suit but a rank difference of 2 are given a score decrease of 1. Any cards that are known to still be held by the opponent or that are in the discard pile are skipped while marking since they are already labeled correctly as impossible or known. The likelihood score for cards in the deck can factor in when deciding which card to discard. Cards that are less likely to complete a meld for the opponent are more desirable to discard and can be identified by checking the likelihood score of cards that neighbor the potential discard.

**Heisenbot Discard Strategy** For our Heisenbot player, we focus on a mixture of offensive and defensive play. Generally we will not discard a melded card unless there are no unmelded cards in our hand. For our remaining cards, they are separated into three categories: deadwood, doubles, and triangles. A triangle offers several possible options for drawing a single card and completing a meld with two of the triangle cards. A double is a set of two cards of either the same suit or of adjacent ranks that are not part of a triangle in our hand. All other unmelded cards are considered deadwood for discard purposes. As the turn count increases for a hand, doubles are converted down to deadwood for discard decisions. As cards are assigned to each of these three groups they are also evaluated for a safety count value. The best option for discard, if available, would be a card in the deadwood set. If no deadwood cards are in the hand, a double card is selected. If there are no double cards or deadwood cards, a triangle card is selected. To determine which card in the selected set should be discard the safety counts of the matching cards are considered. Whichever card has the lowest safety count is selected since the opponent has the least ways to use the card if picked up. If multiple cards have the same safety count the card with the highest rank is chosen to break the tie.

### Knock Policy

The primary goal in developing a strong knock policy was to emulate the nuances in logic that human players consider when deciding whether or not to knock at a certain point in the round. There are several factors which influence that decision in various circumstances and as such must be broken down to find the logic behind them. It is important to note that all of the following policy variants only include legal knocking, which according to the rules of the game employed here, means having deadwood less than or equal to 10.

**Simple** The most basic knock policy is to knock as soon as possible. Indeed, this is the general strategy of many novice human players as well as the simple AI player provided by tournament organizers. While this is a decent baseline strategy and will result in some hands won, it also gives way to low deadwood differentials as well as undercutting in many cases because of important factors that it overlooks. For example, the number of turns taken in the round greatly impacts the value of knocking quickly. Knocking with high (7-10) deadwood in the first few turns of a round may be advantageous because the opponent has not had ample time to strategically create melds whereas later in the game it is more likely to cause undercutting. Also, knocking with high deadwood but low card count (e.g., only deadwood is JS) is a poor decision because in most cases there is a high probability of drawing lower deadwood on your next turn. Furthermore, being in a position to knock indicates a higher likelihood of scoring positive points. If a player is significantly behind in points, it may be best not to squander the opportunity to go gin and close the gap in points.

To ground these ideas in empirical data, we simulated 10,000 games between the simple AI player and itself. It is

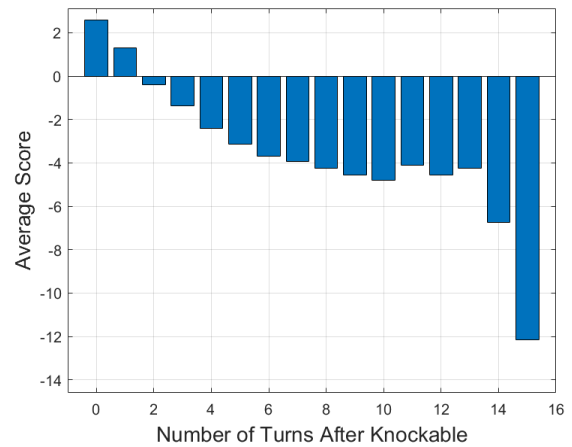


Figure 2: Average score versus turns after first ability to knock for the simply AI strategy in simulation of 10,000 games (>250,000 rounds). Negative scores indicate impact of undercutting as the player waits too long to knock.

important to note that these simulation results should not be generalized without caution to all gin rummy players. However, given that this aspect of the work was conducted early on, the simple player was the best strategy we had access to and it performed comparably against novice human players. Figure 1 depicts the distribution of deadwood and the likelihood of going gin as a function of turns in the game (i.e. 2 turns means 1 turn per player). This plot emphasizes the benefit of knocking early if able, but not as quickly mid-game because the opponent is more likely to have low deadwood and undercut. On average, the simple player was able to knock after 18 turns (9 per player) and go gin after 21 turns (10-11 per player). That last statistic provides some evidence that waiting 1-2 turns after you are able to knock may allow you to go gin and maximize points per round.

To explore that idea a bit further, Figure 2 plots the average score of the simple player as a function of turns after the first ability to knock in a round. This graph reiterates the idea of diminishing returns after waiting more than a couple turns to knock. However, undercutting disproportionately decreases the score, so there may be other factors to consider when searching for the optimal knock policy. Also, these simulations only reflect outcomes for the simple player; while we suspect similar trends for other AI strategies, the exact profiles may differ and require validation through future simulations.

**Greedy** A viable extension of the simple knock strategy is to lower the internal threshold for knocking in an attempt to: (1) maximize point differential; (2) increase the likelihood of going gin; and (3) decrease the likelihood of being undercut. By trial and error, we found that choosing to knock only when deadwood is less than or equal to 5 yielded higher winning scores than the standard threshold of 10 on average, all other policies constant.

**Rule-Based** The previous two policies are solely based on a single threshold for deadwood. However, there are other factors to consider that may improve performance outcomes. Here, we investigated several handcrafted rulesets in a tree structure to yield a more complex decision theoretic. For example, the current game score, the number of turns in the round, and the number of deadwood *cards* (as opposed to *points*) all can have an impact on the value of knocking.

**Heisenbot Knock Strategy** For our Heisenbot player, we opt for the rule-based decision structure for knocking. The specific rules we implemented include, in order of precedence:

- If we have no deadwood, go gin.
- If we are ahead by 30 points or more in the game, do not knock to try and win the game faster.
- If we are behind by 30 points or more in the game, do not knock because small point differentials will have negligible effect.
- If we have taken fewer than 4 turns, knock because our opponent is more likely to have high deadwood.
- If we have taken more than 13 turns, knock because our opponent may be close to going gin anyways or they are stuck with high deadwood.
- If we have less than or equal to 5 deadwood points, knock because of diminishing returns for waiting too long.
- If we have more than 5 deadwood points, but fewer than 3 deadwood cards, do not knock because it is highly likely that our deadwood will decrease more quickly.

## Results

While the flow of this paper may suggest we selected the policies for Heisenbot *a priori*, those decisions were primarily driven by empirical tests. Here, we demonstrate some of those tests with a tournament of the following five players:

1. The simple AI player provided by tournament organizers (Simple).
2. The simple AI player adjusted for greedy knocking (Greedy).
3. The simple AI player adjusted to always draw from the deck (Always Draw).
4. The simple AI player adjusted for complex rule-based knocking (Rule-Based Knocking).
5. Our custom AI player described in this paper (Heisenbot).

Every player competed in head-to-head matchups with each of the other four players. Every matchup comprised 20,000 games. Win percentages for head-to-head matchups are depicted in Figure 3. Note that the strength of a given player is indicated by the win percentages in their respective row. Clearly, the policy of always drawing from the deck is a poor choice because the Always Draw player is the weakest in the tournament. The strongest player is Heisenbot, with an average win percentage of 68.8%. Compared to the baseline (Simple) strategy, Heisenbot wins 60.5% of the time,

	Simple	Greedy	Always Draw	Rule-Based Knocking	Heisenbot
Simple		46.2	89.1	43.4	39.5
Greedy	53.8		94.6	46.1	38.5
Always Draw	10.9	5.4		7.1	2.3
Rule-Based Knocking	56.6	53.9	92.9		44.6
Heisenbot	60.5	61.5	97.8	55.4	

Figure 3: Head-to-head win percentage over 20,000 games per matchup. Numbers indicate the percentage of wins for the player listed in the row against a given opponent listed in the corresponding column.

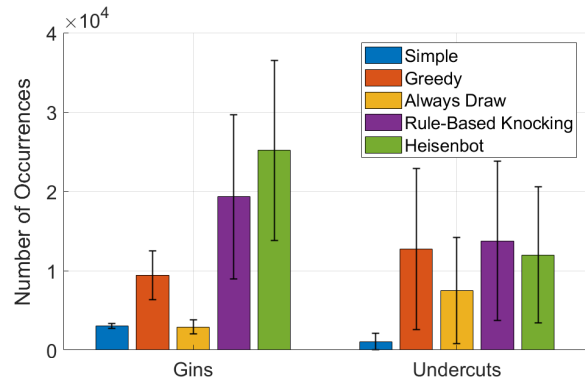


Figure 4: Tournament statistics including the number of times a player went gin or undercut their opponent. Larger values indicate stronger players. Error bars mark one standard deviation.

which is a decent margin. The next best competitor is the Rule-Based Knocking player, winning almost 45% of games against Heisenbot. This is likely because the knock policies of these two players are most similar (and the most complex). Perhaps this suggests that knocking is the most critical decision to make in the game.

Figure 4 analyzes the effect of policy decisions on earning bonus points from going gin or undercutting an opponent. Heisenbot outperforms the other players in going gin, which is likely caused by the modified draw and discard policies that prioritize triangles and doubles. However, Heisenbot is not the best in terms of undercutting, despite having similarly high numbers to the Greedy and Rule-Based Knocking players.

Another way of analyzing the tournament statistics is to examine the points earned per round for a given player. Figure 5 plots this data, showing the average points a player



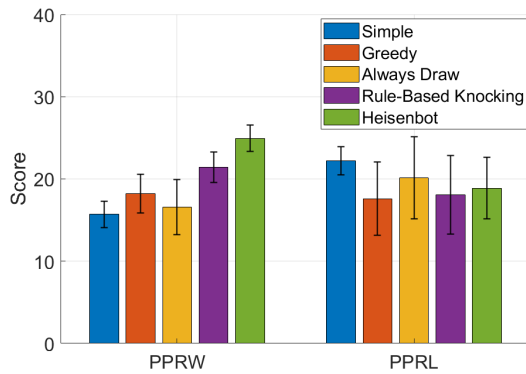


Figure 5: Tournament statistics including the points per round wins (PPRW) and points per round losses (PPRL). PPRL refers to the average score for an opponent who wins. Therefore, high PPRW and low PPRL are desired. Error bars mark one standard deviation.

earns when they win (PPRW) alongside the average points a player’s opponent earns when they win (PPRL). The optimal player should have high PPRW from high deadwood differential, going gin, or undercutting. On the other hand, the optimal player should also have low PPRL, indicating that their opponents do not earn many points for a win. Heisenbot scores the most points per round, on average, when it wins ( $PPRW = 25.0 \pm 1.6$ ), while the Simple player scores the least on average for a win. This reinforces the idea that the basic strategy of knocking as soon as you can is suboptimal. Although Heisenbot has a low PPRL ( $18.9 \pm 3.8$ ), it is not as low as the Greedy player ( $17.6 \pm 4.5$ ) or the Rule-Based Knocking player ( $18.0 \pm 4.8$ ). Heisenbot has room for improvement in this regard, although it is worth noting the higher standard deviation of players with lower PPRL.

The tournament results presented in this paper are limited by the sequential refinement of AI strategies within the competing players and the fact that improvements were made solely based on performance against similar agents. Indeed, our initial modifications were developed by examining how the Simple agent played against itself. From this perspective, it is possible that Heisenbot may be “overfit” to playing against simple strategies and have flaws which more advanced players may be able to exploit. Specifically, we hypothesize that AI players built on deep learning models could learn to predict the best card to discard more accurately than our discrete likelihood model. Also, if a computer opponent played many games against Heisenbot, it could in theory discover some of the Heisenbot’s knocking policies. Nonetheless, even knowing the full set of specific rules for knocking, the authors had a difficult time beating Heisenbot in friendly matches consistently.

## Conclusions and Future Work

In this paper, we outline the development of Heisenbot, a rule-based AI player for the card game gin rummy. Heisenbot prioritizes future value in drawing cards, discards based

on deadwood with no value to either player, and knocks using a structured set of empirically-driven rules. In a small tournament against simple AI competitors, Heisenbot performs well, winning almost 70% of games. However, there is still much room for improvement. Future areas of interest include deep reinforcement learning (Heinrich and Silver 2016) or deep convolutional networks (Moravčík et al. 2017), because these approaches can capitalize on patterns that are difficult for human players to articulate.

## References

- Heinrich, J.; and Silver, D. 2016. Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*.
- Hsu, F.-h. 1999. IBM’s deep blue chess grandmaster chips. *IEEE Micro* 19(2): 70–81.
- Koller, D.; and Pfeffer, A. 1995. Generating and solving imperfect information games. In *IJCAI*, 1185–1193. Citeseer.
- Kotnik, C.; and Kalita, J. K. 2003. The significance of temporal-difference learning in self-play training TD-rummy versus EVO-rummy. In *Proceedings of the 20th International Conference on Machine Learning (ICML-03)*, 369–375.
- Kuhn, H. W. 1950. A simplified two-person poker. *Contributions to the Theory of Games* 1: 97–103.
- Maddison, C. J.; Huang, A.; Sutskever, I.; and Silver, D. 2014. Move evaluation in Go using deep convolutional neural networks. *arXiv preprint arXiv:1412.6564*.
- Moravčík, M.; Schmid, M.; Burch, N.; Lisý, V.; Morrill, D.; Bard, N.; Davis, T.; Waugh, K.; Johanson, M.; and Bowling, M. 2017. Deepstack: Expert-level artificial intelligence in heads-up no-limit poker. *Science* 356(6337): 508–513.
- Silver, D.; Hubert, T.; Schrittwieser, J.; and Hassabis, D. 2018. AlphaZero: Shedding new light on the grand games of chess, shogi and Go. *DeepMind blog*.
- Silver, D.; Schrittwieser, J.; Simonyan, K.; Antonoglou, I.; Huang, A.; Guez, A.; Hubert, T.; Baker, L.; Lai, M.; Bolton, A.; et al. 2017. Mastering the game of go without human knowledge. *nature* 550(7676): 354–359.
- Tesauro, G. 1995. Temporal difference learning and TD-Gammon. *Communications of the ACM* 38(3): 58–68.
- 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.