FANTASYCLASH: BENCHMARKING STRATEGIC REASONING IN FANTASY SPORTS

Anonymous authors
Paper under double-blind review

ABSTRACT

An intelligent agent that operates among its peers must reason strategically: it must model how its counterparts will respond to its own action. Existing evaluations of strategic reasoning in LLM agents rely on stylized matrix games, board and card games, or dialogue-based negotiation, which lack the complex dynamics that govern the uncertainty real-world agents tackle. We present FantasyClash, a benchmark for strategic reasoning under real-world uncertainty. The benchmark is built on the popular fantasy sports format, where agents (1) bid for athletes in an auction, (2) field weekly lineups from their pools, and (3) bet on their teams' performance relative to others. Crucially, the outcomes are determined by the corresponding athletes' performances in real games, which imbues the game with high-dimensional, real-world dynamics. Evaluating 8 frontier LLMs, we find that agents vary substantially in their innate ability to play the game. We further write detailed descriptions of 7 strategies and verify them as effective prompts for improving an agent's performance. We empirically identify intransitive cycles among a subset of strategy-prompted agents, illustrating strategic depth where success depends on inferring counterparts' strategies. Furthermore, we find that agents rarely act on what their opponents do, but some agents exploit an opponent's strategy once they see its full text. Finally, we characterize the relative strengths of the 8 agents through a large-scale evaluation with an Elo rating system.

1 INTRODUCTION

Intelligent agents must confront uncertainty and adapt to changing environments. Reflecting this view, existing evaluations for large language model (LLM) agents measure their ability to resolve uncertainties (Krishnamurthy et al., 2024; Barres et al., 2025; Yang et al., 2026) and adapt to changes (Park et al., 2025; Asawa et al., 2026; Mittal et al., 2026). However, this only captures a fraction of the complexity a real-world agent contends with. When the environment is populated by other agents, each pursuing their own objectives, those counterparts not only change but change *in response* to the agent's actions. Therefore, an agent must reason strategically: modeling the counterparts' strategies and anticipating how they will adapt to its own actions. This *strategic reasoning* under uncertainty, long studied by classical game theory (von Neumann & Morgenstern, 1947; Nash, 1950), is a hallmark of intelligence.

Considerably fewer evaluations focus on strategic reasoning. One line of work evaluates LLMs in toy 2×2 matrix games such as Prisoner's Dilemma or Battle of the Sexes (Akata et al., 2025; Wang et al., 2025). Beyond toy settings, others implement classic board and card games from chess to tic-tac-toe (Feng et al., 2023; Guertler et al., 2025), or focus on dialogue-based negotiation in games such as Diplomacy or Werewolf (FAIR et al., 2022; Xu et al., 2024). These settings offer stylized and convenient testbeds for evaluating basic strategic reasoning of LLM agents, wherein the environment dynamics that govern the uncertainty are fundamentally simple, insofar that they can be fully described in a manual. Realistic strategic settings are rarely so straightforward. Often, real-world uncertainty and environment shifts stem from two distinct but intertwined sources: (a) a leviathanic environment with high-dimensional latent that operates on its own convoluted dynamics, largely indifferent to actions of the agent, and (b) the immediate peers of the agents, who not only interact with the environment in service of their own goals, but also actively observe, react, and adapt their strategies to the agent's actions. From geopolitics to oligopoly, real-world strategic

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

reasoning tackles complexity arising from the *interplays* of these two components. For example, an airline must plan its future capacity by considering *rivals' plans* in relation to *future travel demand*: maintaining capacity when rivals cut flights may capture market share if demand holds up, but might leave its flights unprofitable if demand falls more than expected. Current evaluations of strategic reasoning lack the complex environmental dynamics, which is central to the difficulty that intelligent agents tackle in strategic settings.

This paper presents FantasyClash, a benchmark designed to evaluate strategic reasoning in the presence of both adaptive opponents and complex real-world dynamics. We implement a widely-played multi-agent game format, fantasy sports, which proceeds in three stages: (1) an auction stage in which agents bid on players to build their pools, (2) a weekly play stage where agents select players from the pool to compete as a team, and (3) a betting stage in which agents bet on the performance of their selected players relative to the others' teams. Crucially, these players correspond to real athletes, and their in-game performance in any given week is tied to their *actual performances* during that week. This design grounds the game's uncertainty in real-world events, so that agents' strategic decisions necessarily need to account for complex external dynamics. FantasyClash therefore evaluates strategic reasoning and real-world dynamics as intertwined components.

Besides grounding uncertainty in the real world, fantasy sports has additional properties that make it a uniquely fruitful setting for studying strategic reasoning. First, a variety of strategies are known to the community. Some agents spend most of their budget on a few elite athletes and fill the rest of the roster with minimum-price players, an approach known as *stars and scrubs*, while others build a more balanced roster (Cockcroft, 2012). An agent can also bid on athletes it does not want in order to drain rivals' budgets, a tactic that auction theory shows can pay off when bidders face budget constraints (Benoît & Krishna, 2001). Some agents discount hot streaks, since short runs of strong performance tend not to persist (Efron & Morris, 1975; Ding et al., 2025).

Importantly, there is no dominant strategy: due to the bidding stage, if many agents employ a similar strategy, the competition for the same players drives up prices and thereby diminishes the strategy's advantages. As a result, it is fundamentally necessary to infer the capabilities and intent of other agents to perform well in the long run. Second, due to general popularity of sports and its well-established analytical culture, fantasy sports offers uniquely rich sources of public information for agents to develop their judgment and strategies. For one, numerous detailed historical statistics are available across many sports, which can be used both as the foundation of game variety and rich sources of history on which agents can base their judgment. Fantasy play has itself become a subject of quantitative research, from lineup optimization for hockey contests (Hunter et al., 2016) to models of how opponents choose their teams (Haugh & Singal, 2021). Beyond statistics, skilled human players observe athletes in real games, form a nontrivial understanding of the player styles and the sport, and use it to predict how those athletes will perform in the future. FantasyClash offers a testbed for agents to likewise combine statistics, news, and their own understanding to compete in a uniquely complex strategic game.

Our main contributions are as follows:

- **Benchmark.** We introduce FantasyClash, a strategic reasoning benchmark where strategy and complex real-world dynamics are fundamentally intertwined.
- **Agent evaluation.** Evaluating 8 frontier LLMs and 3 rule-based baselines, we find that agents differ widely in raw ability.
- **Strategies as levers.** We develop detailed descriptions of 7 strategies and show that 5 improve both money and rank when given as prompts to a capable agent. These results from interpretable strategies can support future understanding of agents' natural tendencies or generating strategic behavior for finetuning.
- **No (known) dominant strategy.** We demonstrate the presence of non-transitive cycles in performance among specific bidding strategies, highlighting situations where multi-agent interactions can require agents to model their opponents.
- **Limited opponent modeling.** We show that agents rarely act on what their opponents do, even when we prompt them to track their opponents. However, some agents exploit an opponent's strategy once we show them its full text.

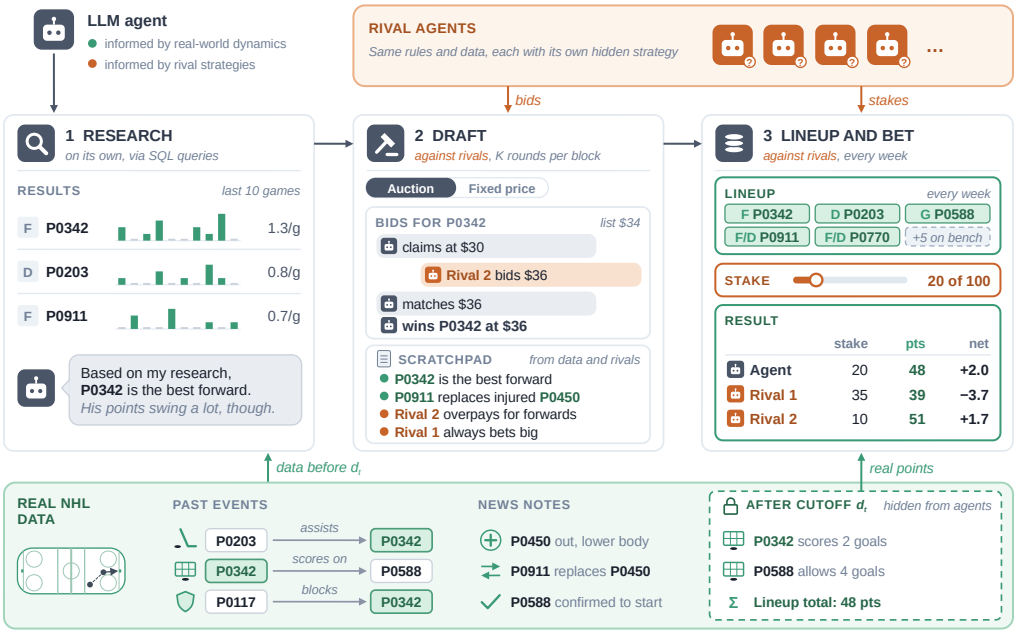


Figure 1: Overview of FantasyClash setup. FantasyClash requires an agent to reason about the interplay between (a) real-world dynamics, such as injuries and volatile athlete performance, and (b) strategies of other agents, which shape bids and stakes. Agents must reason through these deeply intertwined components: proper draft and bids on athlete depend on both their expected points from real athlete performance and what rivals will pay for them.

2 Related Work

Evaluating agents under uncertainty. Recent benchmarks evaluate whether LLM agents can actively acquire information, resolve uncertainty over multiple turns, and adapt to changing environments (Krishnamurthy et al., 2024; Barres et al., 2025; Yang et al., 2026; Park et al., 2025; Asawa et al., 2026; Mittal et al., 2026). However, these works often focus on uncertainty that is static or evolves independently of the agent's actions. A qualitatively different kind of uncertainty arises when other agents exist, whose strategies and intent can change *in response* to the agent's actions. In turn, the agent needs to infer what others intend to do and anticipate how its own choices might change their behavior. This need for strategic reasoning is a central distinction between our work and conventional multi-turn benchmarks.

Strategic reasoning in LLMs. Existing evaluations study LLMs' strategic capabilities through well-understood matrix games (Akata et al., 2025; Wang et al., 2025), common board games (Feng et al., 2023; Guertler et al., 2025), or social, communication-based games (FAIR et al., 2022; Xu et al., 2024). Although these settings capture strategic interactions between agents with competing objectives, they lack the complex external dynamics that underlie real-world strategic settings. For example, a firm in an oligopolistic market needs to anticipate both its rivals' strategies and shifts in customer demand or supply-chain conditions. Financial market benchmarks (Yu et al., 2025; Zeng et al., 2026) represent the other extreme, where agents' returns depend entirely on external market dynamics but not on each others' strategies. FantasyClash better represents real-world strategic settings by tightly intertwining complex dynamics and strategic decisions: choosing the best action requires an agent to predict future athletes' performances and account for its opponents' strategies.

3 FANTASYCLASH

3.1 GAME FORMULATION

FantasyClash is a multi-agent fantasy hockey environment built on games from historical NHL seasons (Figure 1). Environment rules closely follow and are grounded in real-world fantasy sports (Anagnostopoulos et al., 2016; Boudreau & Shunda, 2016; Allen, 2022; Hunter et al., 2016; Haugh & Singal, 2021). We define a run to cover a window of consecutive weeks from one season, and it is split into draft blocks of several weeks each. At the start of each block, every agent drafts a roster from active players. In each week, every agent chooses a lineup from its roster and places a stake on that lineup. Each agent tries to end the run with as much money as possible.

Each week begins at a cutoff date. Agents make each block's and week's decisions using only data that predates that cutoff date. Each week is scored on the games played in the seven days after the cutoff. A *rule set* gives a weight to each box-score statistic, e.g., goals. Only the players in an agent's lineup score. The total fantasy points of agent i for a week, which we call its lineup points, are

$$\mathsf{LineupPoints}_i = \sum_{p \in L_i} \sum_{\text{statistic } c} \mathsf{Weight}_c \times \mathsf{Total}_{p,c}$$

where L_i is the set of players in agent i's lineup and $\mathsf{Total}_{p,c}$ is player p's total for statistic c in that week's games. The inner sum is player p's fantasy points for the week. Appendix C.3 lists the statistics and their weights.

Draft. Each agent gets a fixed draft budget per block that is separate from the betting bankroll. The draft budget can only be used to acquire players, and any unspent budget is lost at the end of the block. Each player has a public list price. The list price is based on the player's fantasy points over the 60 days before the cutoff and over the previous season. A random error of 20% is also added to each player's list price, so the list price is only a rough guide to the player's strength. Agents are not told how list prices are computed.

In our experiments, players are sold in an auction held in rounds, with the order of agents shuffled each round. In each round, every agent gets one *turn* to acquire a player and one *steal* to challenge another agent's claim. On its turn, an agent claims a player at a price it names, which can be below the list price. Any agent that has not yet acquired a player this round can use its steal to place a secret bid above the claim. If challenged, the claimant either matches the highest bid and keeps the player, or lets the challenger take the player at its bid. An agent cannot make a claim or bid that would leave it unable to fill its roster.

Choose Lineup. Each week, an agent chooses a starting lineup from its roster, and only the players it starts score. A lineup has a fixed set of slots. Each slot either requires a player of one position (forward, defenseman, or goalie) or accepts an athlete of any position. In our default setting, an agent drafts 10 athletes and starts 5 of them each week. The 5 slots are one forward, one defenseman, one goalie, and two slots that accept any skater. The agent's weekly LineupPoints are the sum of FantasyPoints over the athletes it starts.

Bet. Every agent starts the run with a betting bankroll. Each week, every agent stakes an amount between a fixed minimum and its whole bankroll. In our default setting, agents stake before any lineup is revealed, so each agent knows its rivals' rosters but not which players they will start. All stakes go into a common pot, and the whole pot is paid back out in proportion to $\mathsf{Stake} \times \mathsf{LineupPoints}$. Agent i's net gain for the week is its payout from the pot minus its stake,

$$\mathsf{NetGain}_i = \underbrace{\mathsf{Pot} \times \frac{\mathsf{Stake}_i\,\mathsf{LineupPoints}_i}{\sum_j \mathsf{Stake}_j\,\mathsf{LineupPoints}_j}}_{\text{payout from the pot}} - \underbrace{\mathsf{Stake}_i}_{\text{stake paid in}}$$

An agent therefore gains money only when its lineup points exceed $\bar{w} = \sum_j \mathsf{Stake}_j \mathsf{LineupPoints}_j / \sum_j \mathsf{Stake}_j$, the stake-weighted average of lineup points. When one agent stakes much more than the others, $\bar{w}$ moves close to that agent's lineup points, so the gain from a bet depends on rivals' stakes as well as on their lineups. An agent whose bankroll falls below the minimum stake is eliminated.

3.2 Data and Interface

Data. The data is taken from sixteen NHL seasons (2010 - 2026), and contains numeric and textual data. The numeric data comes from the NHL's public APIs and includes box scores, play-by-play records, and player statistics for every game. (National Hockey League, 2026b;a) The text data are 127,000 dated news notes from Daily Faceoff (Daily Faceoff, 2026), a fantasy hockey news site. The notes report events such as injuries, lineup changes, and confirmed goalie starts. Language models may have seen these data during training. We therefore replace every player and team with an opaque token, e.g., `P0342`. Appendix D lists the sources and describes how we prepared the data.

Agent interface. All agents in a run receive the same system prompt, covering game rules and data schema. At each decision, e.g., a draft turn, an agent also receives the state it needs for that decision, such as its remaining budget. All other information must be researched by the agent through SQL queries of the database. The agent has a fixed research budget that limits the number of queries the agent can make for each decision. We log every step, including the reason an agent gives for each decision, so any decision can be checked afterward. Appendix C gives all prompts.

An agent does not keep the rows returned by its queries from one week to the next. However, the agent keeps a record of the queries it ran and the decisions it made, including those from earlier draft blocks. The agent also sees every opposing agent's past rosters, stakes, and results. Furthermore, agents have a private scratchpad that they can write to (up to a fixed length) at any point in the game. An agent therefore has to write down anything it wants to remember such as what it has learned about a rival. Each agent keeps the same anonymous ID across blocks, so an agent can link a returning opponent to its past behavior.

Generalizability Most settings of FantasyClash can be adjusted. The simplest changes are to the roster size, the lineup slots, and the pool of opponents. The draft can be an auction or a fixed-price draft, in which agents take turns buying players at their list price (Appendix B.1). The rule set can give any weight to any box-score statistic, and under a different rule set, different players are valuable. By default, we use the *balanced* rule set, in which the best forward, the best defenseman, and the best goalie score about the same number of points. Other settings control what agents know and remember. For example, agents can keep either a full record of their past decisions or only a scratchpad. Appendix E.1 gives the full list of settings.

4 Experimental Setup

4.1 Run Configurations

Unless we state otherwise, each run lasts 12 weeks and is split into three draft blocks of 4 weeks. All runs use the *auction* draft and the *balanced* rule set. The draft budget is 200 per block, bankrolls start at 100, and the minimum stake is 5. An agent may make 15 queries before each draft, 2 on each draft turn, and 5 in each week. We refer to each participant in a run as a *seat*, because several seats in one run can use the same model.

Metrics. We report three run-level metrics. 1) *Money* is the bankroll the seat holds at the end of the run. 2) *Fantasy points* (FP) are the total points scored by the players the seat started, summed over all weeks. 3) *Rank* is the seat's finishing position by money among the seats in its run. We use two evaluation settings. In the fixed-pool setting, the same seats play every run, so we can compare their ranks directly. In the varied-pool setting, the seats change from run to run, so ranks from different runs are not comparable.

To compare agents across varied pools, we fit a Plackett-Luce model (Luce, 1959; Plackett, 1975) to the order in which seats finish each draft block by money. We fit the scores by maximum a posteriori estimation with a Gaussian prior that has a standard deviation of 800 on the Elo scale. We report the scores on the usual Elo scale and fix the `stats` baseline, which plays in nearly every run, at 1500. We compute 95% intervals with a bootstrap that resamples whole runs. Note that since we observe non-transitive performance among strategies (Section 5.3), a scalar Elo rating should be viewed as summarizing average field performance.

Table 1: **Fixed-pool evaluation.** The same six seats play 10 runs of 12 weeks each, with the auction draft and the balanced rule set. The best value in each performance column is in bold. Reported with $\pm SE$.

Agent	Rank	Money	FP	Cost ($)
GPT-5.6 Terra	**2.1** ±0.4	**115.1** ±4.3	**2153** ±39	3.79 ±0.13
GPT-5.6 Luna	2.7 ±0.5	108.2 ±6.2	2032 ±45	**0.34** ±0.01
`projection`	3.0 ±0.5	105.9 ±4.7	2048 ±51	(N/A)
Claude Sonnet 4.5	3.1 ±0.5	104.7 ±5.7	2029 ±71	12.51 ±0.34
DeepSeek V3.2	4.6 ±0.4	90.3 ±3.8	1875 ±64	8.52 ±0.27
Qwen3 235B	5.5 ±0.3	75.8 ±3.4	1709 ±47	5.52 ±0.19

4.2 Agent Evaluation

We evaluate 8 LLMs: GPT-5.6 Terra, GPT-5.6 Luna, Claude Opus 4.5, Claude Sonnet 4.5, Claude Haiku 4.5, DeepSeek V3.2, Qwen3 235B, and Llama 4 Maverick. We also introduce 3 deterministic models. The `stats` agent ranks players by fantasy points per game over the past month and stakes a constant amount each week. The `projection` agent extends `stats`, adjusting its ranking by using the news notes, e.g., when an athlete is injured, and it sizes its stake by how much its projected points vary. The `oracle` agent starts the best possible lineup each week using the actual results of the week, and it stakes a constant amount.

Strategies. A strategy is a block of text that we add to an agent's system prompt. Every strategy starts from a shared section of *fundamentals*, which explains what the rules mean in practice, e.g., that a player's value grows with the number of games the player's team plays. Each strategy then adds a *philosophy*, which is a set of rules drawn from fantasy practice or from research on auctions that tells the agent what to do in different situations. We study seven strategies in three groups and refer to each by a two-letter short form.

Valuation strategies change how an agent judges a player's worth. Mean Reversion (MR) trusts long-run production over recent streaks, because short runs of strong play tend not to last (Efron & Morris, 1975; Ding et al., 2025). Schedule Tiebreak (ST) uses the schedule to break close ties between similar players (Allen, 2022). *Market* strategies change how an agent responds to prices. Room Reader (RR) spends more or less depending on how much rivals overpay, since overpaying follows a regular pattern in real fantasy auctions (Boudreau & Shunda, 2016). Bargain Hunter (BH) buys rivals' cheap claims but never pays more than the list price. *Contest* strategies set fixed rules for claims and challenges. Loyalist (LO) never challenges and always defends its own claims up to a fixed price. Decoy claims players it does not want so that rivals spend their steal on them, and Tax Collector raises its bids against rivals that keep defending their claims. All three draw on research about bidding with limited budgets (Benoît & Krishna, 2001; Anagnostopoulos et al., 2016). Appendix C.2 describes each strategy in detail and gives its full prompt.

5 Results

5.1 Agents Differ in Raw Capability

We first compare five language models and the `projection` baseline over ten runs in the fixed-pool setting (Table 1). The agents succeed and fail in different parts of the game. Terra drafts well, sets strong lineups, and is the most likely to change a later decision in accordance with the performance of others, but it stakes too little when its roster is strong. Luna turns its rosters into money by sizing its stakes well, but repeatedly leaves better players off its lineup, even after naming this mistake in its own notes. Sonnet reasons best about schedules and bets well yet inconsistently, but it is slow to change its lineup when a player's situation changes. DeepSeek researches carefully but often fails to commit, leaving its draft budget unspent. Qwen writes one plan on its notes such as calling for a balanced roster, but then follows another, spending most of its budget on goalies. Appendix A.1 describes each agent in detail.

Table 2: **A cycle among three strategies.** Runs won counts the runs in which the first strategy finishes ahead of the second. The other columns give the first strategy's mean advantage over the second, with the standard error after ±.

Pairing	Runs won	Money	Places	FP
DECOY > TAX COLLECTOR	11/12	+9.2 ±3.2	+1.6 ±0.4	+141 ±50
TAX COLLECTOR > LOYALIST	11/12	+8.7 ±2.9	+1.4 ±0.5	+100 ±47
LOYALIST > DECOY	12/12	+26.0 ±6.6	+1.9 ±0.3	+264 ±68

Human Baseline As a reference point, one person experienced in fantasy sports took one seat in the fixed pool for three runs, with the same interface, information and query budgets as the models and no time limit. They finished first in all three, with mean money 127.8 ± 5.1.

5.2 STRATEGIES (MOSTLY) ENHANCE PERFORMANCE

We evaluate 7 strategies. Each run has four seats: one Terra seat with the strategy, two Terra seats without a strategy, and the `projection` baseline. We use Terra because it follows strategy prompts most closely, and Appendix A.5 reports results with other models. Figure 2 shows that five of the seven strategies improve both money and rank, and LOYALIST and SCHEDULE TIEBREAK give the largest gains in money. DECOY and TAX COLLECTOR lose money and places, and both of them depend on specific behavior from rivals. DECOY gains only when a rival challenges its decoy. TAX COLLECTOR gives up any of its own players that are challenged at more than 0.2 times the list price, and ordinary bids often exceed that line. Section 5.3 shows that each of these two strategies wins against the right opponent.

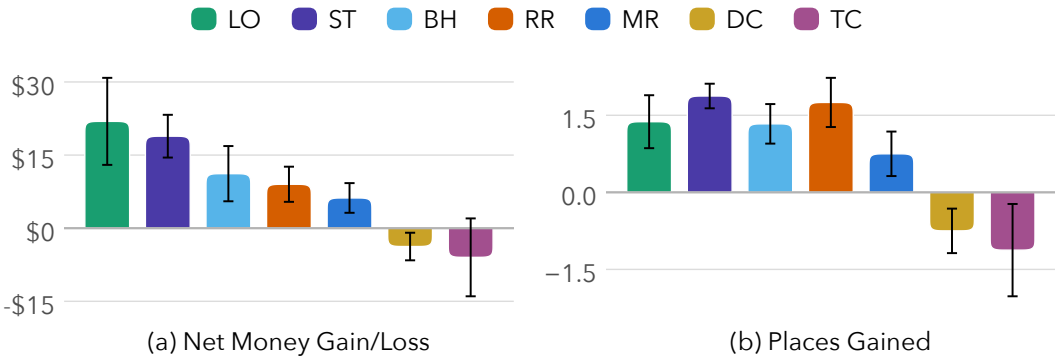


Figure 2: **Strategy performance.** Gain of the Terra seat that plays each strategy over the Terra seats without a strategy in the same runs, in (a) money and (b) places. Five of the seven strategies improve both. DECOY and TAX COLLECTOR lose, because each depends on specific behavior from rivals (Section 5.3). Error bars show one standard error.

5.3 EMPIRICAL INTRANSITIVITY AMONG STRATEGIES.

Rating an agent with one number assumes that strength can be ordered. That is, if A beats B and B beats C, then A beats C. We find two ways in which this assumption fails in FantasyClash.

Strategies Fall in Cycles Three strategies beat one another in a cycle (Table 2). They value skaters in the same way and differ mainly in how they handle contested claims. DECOY beats TAX COLLECTOR because the decoy draws TAX COLLECTOR's challenge, and when TAX COLLECTOR wins that challenge, it has spent its pick for the round on a player that DECOY never wanted. TAX COLLECTOR beats LOYALIST because LOYALIST promises to match every challenge up to a fixed price, so its rivals choose what it pays, and TAX COLLECTOR raises its bids against seats that keep matching. LOYALIST beats DECOY on plain value, because a decoy that no rival takes is a wasted pick. No strategy in the cycle is stronger than the other two, and a single rating cannot express this.

Strategies Suffer From Crowding Out A strategy can lose its advantage as more seats use it. A strategy outperforms seats without a strategy because it exploits some source of advantage, and that advantage erodes when copies compete for the same source. We compare runs with one strategy seat ($k = 1$) to runs with two ($k = 2$) where the other seats are Terra seats without a strategy and `projection`. From $k = 1$ to $k = 2$, a SCHEDULE TIEBREAK seat loses 7.2 ± 3.5 in money, 18 ± 12 in fantasy points, and 1.2 ± 0.9 places in rank. A BARGAIN HUNTER seat loses 4.3 ± 3.4 in money, 14 ± 13 in fantasy points, and 1.6 ± 0.7 places.

The causes differ between strategies. SCHEDULE TIEBREAK copies rarely bid on the same players, and they lose because they share the same weak opponents, whose losses are the source of the strategy's gains. BARGAIN HUNTER copies all target the same claim in each round and bid against each other. Appendix A.4 gives further analysis and runs.

5.4 DO AGENTS REASON ABOUT OTHERS?

As in any social-deduction game, a successful agent should 1) observe opposition to create a profile of their gameplay and 2) act to exploit what they have learned about opposing players. We find that, when unprompted, agents rarely take the first step and almost never take the second.

Opponent modeling and exploitation is rare. Across 79 runs involving seats with assigned strategy, agents rarely noted anything about a rival (only in $62/570$ seat blocks), mostly recording standings like *"field d leads 102.68"*. Surprisingly, only $119/14099$ of explanations agents have written for their actions mention a rival; and only one uses the rival's past behavior. That seat noted that a rival "has consistently wagered aggressively" and kept its own bet small. The other 118 compare the agent's current roster with a rival's roster without motivating an action. Normalizing each seat's rank in a block so that the winner has rank 0 and the last seat has rank 1, seats with a note about a rival have a mean normalized rank of 0.55 ± 0.04, and the other seat-blocks have a mean of 0.50 ± 0.02.

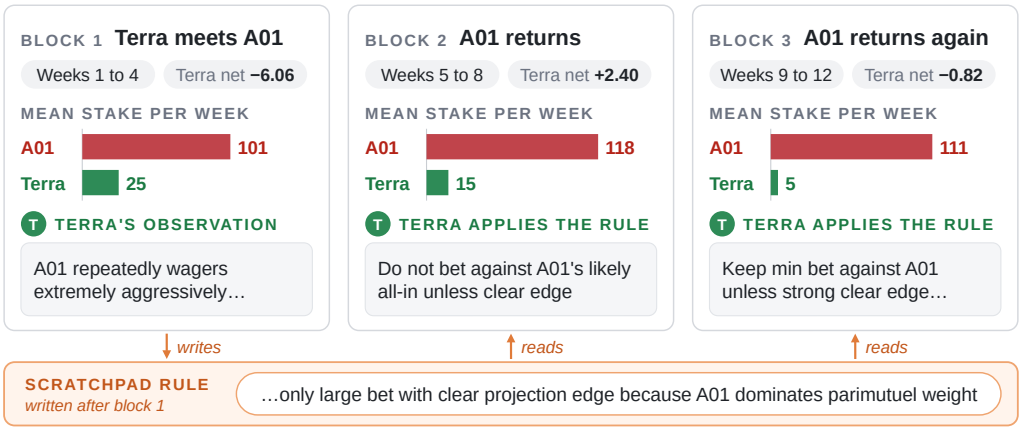


Figure 3: **Terra adapts to a returning opponent.** A01 is Astra's (openai:gpt-6-astra) anonymous label, and it stakes its whole bankroll in 11 of 12 weeks. After block 1, Terra writes a rule in its scratchpad and then stakes the minimum in 7 of the next 8 weeks.

Agents do not act on opponents' strategies. To measure whether agents are able to track opposing agents, we prompt agents to track each opponent and to use what they learn against them. Across 5 runs, we observe that references to opponent IDs rise 3.4 times. Several models keep notes on specific opponents across blocks. For example, Sonnet builds a dedicated opponent section comparing rosters and schedules, Opus records how each opponent bets, and DeepSeek repeatedly describes the leading opponent's roster, whereas Qwen records almost no opponent IDs even when prompted. However, seats prompted to track rivals have a mean normalized rank of 0.51 ± 0.05, and unprompted seats have a mean of 0.52 ± 0.04. We observe that the failure point is that agents rarely act on what they record. The clearest case is a run with two Sonnet seats: they track each other more than any other seats in the experiment, yet neither changes its play in response. Further examples are found in Appendix A.2.

Table 3: **Exploiting a disclosed LOYALIST strategy.** In each pair, a Terra seat plays LOYALIST against an exploiter that either does not see the strategy (blind) or receives its full text (disclosed). Margins are the LOYALIST seat's result minus the exploiter's.

Exploiter	Condition	Unused ($)	FP margin	Money margin
Opus (2013)	Blind	95	+196	+6.4
	Disclosed	4	+50	−1.8
Terra (2015)	Blind	190	+123	+14.4
	Disclosed	107	+16	−6.8

The only exception is Terra in a prompted run against Astra. After the first draft block, Terra recorded that Astra staked its whole bankroll in several weeks. Terra therefore wrote a rule to bet big only with a clear projected edge. When Astra returned, Terra cited this history in its bet reasons and staked the minimum in 7 of 8 weeks. Terra's net improved from −6.06 to +2.40 and −0.82. Figure 3 shows Terra's stakes and its scratchpad notes.

Disclosing a strategy helps some agents exploit it. To measure whether agents can ever exploit opponent strategies, we modify the strategy runs, giving one of the seats without a strategy the full text of its opponent's strategy. We observe disclosure leads to exploitation only when the strategy contains a rule that an opponent can steer. For TAX COLLECTOR, disclosure adds little. TAX COLLECTOR gives up any player challenged at more than 0.2 times the list price, but rivals already challenge at about 0.56 times the list price without knowing the rule, and exploiter Luna's bids do not change once the rule is disclosed.

Conversely, with LOYALIST, since it matches any challenge up to 1.5 times the list price, an opponent who knows this cap can bid just below it to make LOYALIST pay more for its own players, or just above it to take them. After disclosure, Opus does both against a Terra seat playing LOYALIST, and Terra's lead over Opus falls from 196 to 50 fantasy points (Table 3). When Terra is the exploiter, it finds the same approach and places 9 of its 11 challenges between 1.49 and 1.52 times the list price, compared with 0 of 8 when blind. The exploitation shows clearly in the draft, where the LOYALIST seat pays more and leaves less budget unspent.

5.5 LARGE-SCALE EVALUATION

To compare agents, we fit a Plackett-Luce model from Section 4 to 113 draft blocks from 37 auction-draft runs, covering 21 distinct sets of competing models (Figure 4). Since strategy performance is intransitive (Section 5.3), these ratings summarize average strength against the pools we ran rather than a fixed ordering. The `oracle` won all 8 blocks it played, so the data give no upper limit on its rating, and we show only its lower bound.

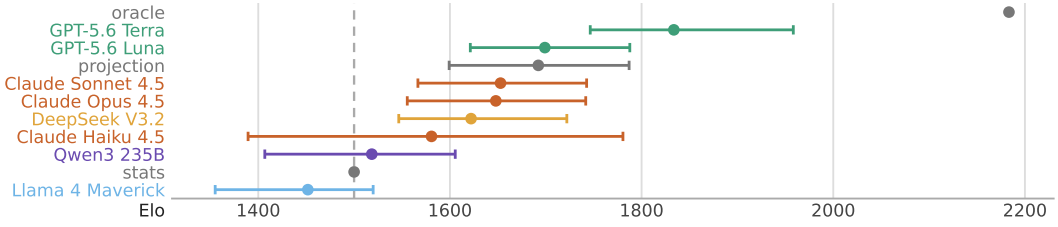


Figure 4: **Ratings of all agents.** Plackett-Luce ratings on the Elo scale, fit to 113 draft blocks from 37 auction runs, with `stats` fixed at 1500 (dashed line). The bars are 95% bootstrap intervals over runs. The `oracle` won every block it played, so we show only the lower bound of its rating.

AI USE STATEMENT

We used generative AI tools to write code for the benchmark and to help compute and interpret results. We have not used generative AI tools to develop theoretical models or conceptual frameworks,

to formulate mathematical claims or write proofs, to propose hypotheses, or to design the methodology or experiments. Translation is not applicable to this work. Additionally, we used generative AI tools to draft and edit parts of the paper for readability and to suggest its structure. The LLM agents that we evaluate are the subject of our study, and all of their outputs come from the experiments that we describe. The authors reviewed all AI-assisted text and figures, and we take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

Ethics statement

Our data come from public NHL statistics and public fantasy hockey news. We replace every player and team name with a token before agents see the data. No real money is involved, and all betting happens inside the simulated game. One volunteer experienced in fantasy sports played three runs as a human baseline. They were unpaid, and we collected no personal information about them.

Reproducibility statement

Appendix C gives every prompt that an agent can receive, including the full text of each strategy. Appendix D describes the data sources and the processing steps, and it gives the formula for list prices. Appendix E.1 lists every configuration setting, and Table 7 lists the season windows of the rated runs. We release all code in Appendix F.

References

Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *Nature Human Behaviour*, 9(7):1380–1390, 2025. doi: 10.1038/s41562-025-02172-y.

Sean Allen. Fantasy hockey: How to add, drop and stream to max out your team. ESPN, https://www.espn.com/fantasy/hockey/story/_/id/35196481/espn-nhl-how-add-drop-stream-max-your-team, 2022. Accessed September 2026.

Aris Anagnostopoulos, Ruggiero Cavallo, Stefano Leonardi, and Maxim Sviridenko. Bidding strategies for fantasy-sports auctions. In Yang Cai and Adrian Vetta (eds.), *Web and Internet Economics (WINE 2016)*, volume 10123 of *Lecture Notes in Computer Science*, pp. 102–115. Springer, 2016. doi: 10.1007/978-3-662-54110-4_8.

Parth Asawa, Christopher M. Glaze, Gabriel Orlanski, Ramya Ramakrishnan, Benji Xu, Asim Biswal, Vincent Sunn Chen, Frederic Sala, Matei Zaharia, and Joseph E. Gonzalez. Continual learning bench: Evaluating frontier AI systems in real-world stateful environments, 2026. URL https://arxiv.org/abs/2606.05661.

Victor Barres, Honghua Dong, Soham Ray, Xujie Si, and Karthik Narasimhan. τ^2-bench: Evaluating conversational agents in a dual-control environment, 2025. URL https://arxiv.org/abs/2506.07982.

Jean-Pierre Benoît and Vijay Krishna. Multiple-object auctions with budget constrained bidders. *The Review of Economic Studies*, 68(1):155–179, 2001. doi: 10.1111/1467-937X.00164.

James W. Boudreau and Nicholas Shunda. Sequential auctions with budget constraints: Evidence from fantasy basketball auction drafts. *Journal of Behavioral and Experimental Economics*, 62: 8–22, 2016. doi: 10.1016/j.socec.2016.03.002.

Tristan H. Cockcroft. 2012 fantasy football draft kit: Auction strategy guide. ESPN, https://www.espn.com/fantasy/football/ffl/story?page=nfldk2k12auctionstrategy, 2012. Accessed September 2026.

Daily Faceoff. NHL player news. https://www.dailyfaceoff.com/hockey-player-news, 2026. Accessed September 2026.

Likang Ding, Ivor Cribben, Armann Ingolfsson, and Monica Tran. Do NHL goalies get hot in the playoffs? A multilevel logistic regression analysis. *Journal of Sports Analytics*, 2025. doi: 10.1177/22150218251314737.

Bradley Efron and Carl Morris. Data analysis using Stein's estimator and its generalizations. *Journal of the American Statistical Association*, 70(350):311–319, 1975. doi: 10.1080/01621459.1975.10479864.

FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, Athul Paul Jacob, Mojtaba Komeili, Karthik Konath, Minae Kwon, Adam Lerer, Mike Lewis, Alexander H. Miller, Sasha Mitts, Adithya Renduchintala, Stephen Roller, Dirk Rowe, Weiyan Shi, Joe Spisak, Alexander Wei, David Wu, Hugh Zhang, and Markus Zijlstra. Human-level play in the game of Diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074, 2022. doi: 10.1126/science.ade9097.

Xidong Feng, Yicheng Luo, Ziyan Wang, Hongrui Tang, Mengyue Yang, Kun Shao, David Henry Mguni, Yali Du, and Jun Wang. ChessGPT: Bridging policy learning and language modeling. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2023. URL https://openreview.net/forum?id=pvdm4B6JMK.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. TextArena, 2025. URL https://arxiv.org/abs/2504.11442.

Martin B. Haugh and Raghav Singal. How to play fantasy sports strategically (and win). *Management Science*, 67(1):72–92, 2021. doi: 10.1287/mnsc.2019.3528.

David Scott Hunter, Juan Pablo Vielma, and Tauhid Zaman. Picking winners in daily fantasy sports using integer programming. *arXiv preprint arXiv:1604.01455*, 2016.

Akshay Krishnamurthy, Keegan Harris, Dylan J. Foster, Cyril Zhang, and Aleksandrs Slivkins. Can large language models explore in-context? In *Advances in Neural Information Processing Systems*, volume 37, 2024. doi: 10.52202/079017-3818.

R. Duncan Luce. *Individual Choice Behavior: A Theoretical Analysis*. John Wiley & Sons, New York, 1959.

Daksh Mittal, Tommaso Castellani, Thomson Yen, Naimeng Ye, Fangyu Wu, Minghui Chen, Tiffany Cai, Emmanouil Koukoumidis, William Zeng, and Hongseok Namkoong. LatentGym: A testbed for cross-task experiential learning with controllable latent structure, 2026. URL https://arxiv.org/abs/2606.15306.

John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950. doi: 10.1073/pnas.36.1.48.

National Hockey League. NHL stats API. https://api.nhle.com/stats/rest, 2026a. Accessed September 2026.

National Hockey League. NHL web API. https://api-web.nhle.com/v1, 2026b. Accessed September 2026.

Chanwoo Park, Xiangyu Liu, Asuman E. Ozdaglar, and Kaiqing Zhang. Do LLM agents have regret? A case study in online learning and games. In *The Thirteenth International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=qn9tBYQHGi.

Carolyn Pitchik and Andrew Schotter. Perfect equilibria in budget-constrained sequential auctions: An experimental study. *The RAND Journal of Economics*, 19(3):363–388, 1988.

R. L. Plackett. The analysis of permutations. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 24(2):193–202, 1975. doi: 10.2307/2346567.

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, New Jersey, second edition, 1947.

Haochuan Wang, Xiachong Feng, Lei Li, Yu Guo, Zhanyue Qin, Dianbo Sui, and Lingpeng Kong. TMGBench: A systematic game benchmark for evaluating strategic reasoning abilities of LLMs, 2025. URL https://arxiv.org/abs/2410.10479.

Yuzhuang Xu, Shuo Wang, Peng Li, Fuwen Luo, Xiaolong Wang, Weidong Liu, and Yang Liu. Exploring large language models for communication games: An empirical study on Werewolf, 2024. URL https://arxiv.org/abs/2309.04658.

Gilbert Yang, Yaqin Chen, Thomson Yen, and Hongseok Namkoong. Benchmarking in-context experiential learning through repeated product recommendations, 2026. URL https://arxiv.org/abs/2511.22130.

Haofei Yu, Fenghai Li, and Jiaxuan You. LiveTradeBench: Seeking real-world alpha with large language models, 2025. URL https://arxiv.org/abs/2511.03628.

Zhiyuan Zeng, Jiashuo Liu, Siyuan Chen, Tianci He, Yali Liao, Yixiao Tian, Jinpeng Wang, Zaiyuan Wang, Yang Yang, Lingyue Yin, Mingren Yin, Zhenwei Zhu, Tianle Cai, Xinjie Chen, Zehui Chen, Jiecao Chen, Yantao Du, Xiang Gao, Jiacheng Guo, Liang Hu, Jianpeng Jiao, Xiangsheng Li, Jingkai Liu, Shuang Ni, Zhoufutu Wen, Ge Zhang, Kaiyuan Zhang, Xin Zhou, Jose Blanchet, Xipeng Qiu, Mengdi Wang, and Wenhao Huang. FutureX: An advanced live benchmark for LLM agents in future prediction. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=z28PLIEj6l.

A Qualitative Observations

A.1 Pooled Agent Observations

Terra Overall, Terra had the best performance across all models. It had the highest average FP and led the field in FP in most of recorded runs. Of all models, Terra used its scratchpad most effectively, allowing learned lessons to reliably change its subsequent decisions. For instance, in run 9 Terra's block regret went from 108 $\rightarrow$ 191 $\rightarrow$ 51.5, learning how to select a better lineup. Terra's main weakness is in underbetting on strong rosters. For instance, in run 4 it led the field with 2,138 FP but wagered only a total of $76 all season and finished 3rd. Furthermore, in drafting, Terra occasionally failed to produce reliable backups for positions. For example, in run 7 its goalie, P0083, got injured and scored zero in the final two weeks. Due to the lack of a backup goalie, Terra could not replace the position and gained 0 FP from its goalie position in the lineup.

Sonnet Sonnet had the highest peak performance but was unstable. It was particularly strong at schedule reasoning: in runs 7 and 8, it consistently combined schedule strength with recent form and goalie-start risk. Sonnet also tracked opponents without explicit prompting. However, its performance varied sharply across seasons, ranging from last place in run 3 to first place in run 8, with an Elo SD of 189. Sonnet's main weakness was responding slowly when conditions changed. It tended to keep cold players in the lineup after identifying their decline. In run 10 block 2, it repeatedly started P1085 and P0672 and incurred 276 regret. Its scratchpad was also unreliable. In run 9 it mis-stated its own regret, and in run 10 it mislabeled both the block and the standings.

Luna Luna was effective at converting competitive rosters into bankroll, even when it did not lead in FP. It won runs 2 and 4 despite scoring fewer FP than Terra. It also drafted high-ceiling rosters: it had the highest roster ceiling in run 5 and $2,438$ potential FP in run 9. Luna was comparatively good at identifying data-quality problems, giving the clearest warning in run 5 that roster-history joins could duplicate games. It also reduced its betting exposure when performance deteriorated, as seen in runs 9 and 10.

Luna's main weakness was lineup selection, with low 81.1% capture, largely because it repeatedly started expensive drafted players over stronger bench options. It explicitly diagnosed this problem in runs 8, 9, and 10 but did not consistently change its policy. In run 9 it accumulated 619 regret and only 74.6% capture, including one week in which it started the inactive P0847 for zero while leaving 84.9 FP on the bench. In run 10 it started P0352 for zero throughout block 1. Its execution was also highly variable week to week: in run 7 it selected an exactly optimal lineup in week 6, then incurred 93 regret one week later.

DeepSeek DeepSeek often showed competent local reasoning but struggled to convert research into a complete end-to-end plan. It would often hit tool call limit by trying to update scratchpad at each step. It could improve lineup decisions within a block: in run 7 block 1, its regret decreased from $58.8 \rightarrow 41.6 \rightarrow 20.3 \rightarrow 11.1$. Its lineup conversion was also occasionally competitive; in run 9 its 82.8% capture exceeded Sonnet's, and in run 10 it selected exactly optimal lineups in weeks 5 and 6. DeepSeek generally controlled downside by betting conservatively when evidence was weak, wagering only \$113 in total in run 8. Its strongest performance came in run 10, where it spent its full draft budget in every block and finished second.

DeepSeek's main weakness was failing to turn analysis into decisive roster construction. It left substantial budget unspent in runs 1, 2, 4, 5, and 9, including deficits of \$51, \$60, and \$40 in individual drafts. It could also consume its available tool calls on research before committing to a plan, a failure mode that eventually required a harness-level fix. Even when its lineup capture was reasonable, its roster ceiling was often too low; in run 9 it lost money in every block. DeepSeek also failed to validate anomalous tool outputs. In run 6 it calculated that P0522 had 35 games in one week, stored this result in memory, and later started P0522 over a player who scored 74.7 FP. Its betting could also fail badly when its underlying assessment was wrong, including a -22.4% ROI in run 4, where it finished last.

Qwen Qwen had the weakest aggregate performance, although it occasionally produced strong individual weeks. It won the most weeks in run 5, with three weekly victories, and won block 2 in run 7. It also recorded the lowest regret in run 3. However, this did not necessarily indicate strong lineup reasoning: in several cases, including its exactly optimal weeks in run 10, its roster was so constrained that there were few meaningful alternatives, making an apparently optimal lineup easier to achieve.

Qwen's main weakness was failing to translate its own stated strategy into consistent constraints on behavior. Its notes frequently called for balanced roster construction, yet it repeatedly concentrated heavily in goalies: in run 1 it spent \$194 of its \$200 budget on four goalies, and in run 10 it constructed six-goalie rosters twice. It also showed poor price discipline, paying \$30 in run 8 for P0465 despite a listed value of \$3.87. Qwen was especially vulnerable to bad data: it stored impossible schedule estimates such as "10+ games weekly" in run 5, 15 games in one week in run 7, and 216 games over four weeks in run 9. These errors persisted because its scratchpad tended to accumulate stale information rather than replace it with corrected beliefs. Its betting results were correspondingly weak, including ROIs of -23.1% in run 1 and -31.7% in run 8, and it finished last in seven of ten runs.

A.2 Opponent Modeling Under Prompted Observation

We test whether agents build and use models of specific opponents across repeated auctions. While prompting agents to record opponent observations increases tracking frequency, the magnitude of this effect varies significantly by model and yields no overall performance improvement. Most strategic exploitation stems from the immediate auction state rather than persistent memory of returning opponents. Only one agent, Terra, clearly adapts its strategy to a specific identity; we present this as a descriptive example rather than evidence that opponent memory raises payoff.

Prompting increases opponent tracking, but unevenly across models. With the modeling prompt, references to other agents' IDs rise from 10 to 34 overall: from 1 to 11 for Sonnet, 4 to 8 for Opus, 5 to 13 for DeepSeek, and 0 to 2 for Qwen. The prompt is neither necessary nor sufficient: Sonnet tracks opponents extensively even without it, while two prompted Qwen instances fail to record any opponent IDs. Tracking also does not improve when opponents are copies of the same model: Sonnet instances track one another closely but never build targeted counterstrategies from their logs.

More tracking does not improve performance. Agents that write more extensively about their opponents do not achieve better outcomes: in the one matched pair of runs, the four LLM seats' cumulative gain was $-\$62.58$ without the prompt and $-\$76.46$ with it. The most common use of opponent observations is to copy lessons that work in general, not to exploit a particular rival. Agents extract portable rules from successful opponents, such as prioritizing elite goalies or build-

ing defense-heavy rosters. For example, DeepSeek explicitly notes that it should follow Oracle's drafting approach.

A.3 General Strategy Observations

A.3.1 Model Compliance and Execution Fidelity

Terra follows given rules consistently across time. We verified every draft decision made by Terra seats assigned a strategy against prompt instructions using auction logs (claim prices, contest actions, match decisions). Rule compliance reached **95–100%** across all testable conditions:

- Loyalist claimed at minimum price ($750/750$), never contested ($0/730$ rounds), and matched challenges up to $1.5\times$ list ($305/305$).
- Tax Collector folded above $0.2\times$ list ($194/197$).
- Bargain Hunter contested at half list ($98/99$) and matched only up to list ($125/125$).
- Decoy never contested ($320/320$).

Compliance showed no decay across runs: Loyalist maintained 100% compliance across Blocks 1, 2, and 3, while Tax Collector achieved 100%, 98%, and 100% across the same blocks. Because strategy instructions are appended directly to the system prompt on every call, the context remains persistent.

A.3.2 Drivers of Strategy Departures in Other Models

An analysis of transcripts across DeepSeek (27 strategy seats), Luna ($\approx$ 190 strategy seats), and Sonnet (5 strategy seats) indicates that strategy departures stem from two primary root causes:

Inability to Execute Strategy Requirements

- **Research Failures:** Schedule-dependent strategies rely on counting upcoming games per team. DeepSeek repeatedly generated SQL joins that aggregated games across multiple seasons, resulting in impossible single-week metrics (e.g., *"P1173 (11 games!)"*). It also executed invalid queries repeatedly despite schema hints. When research failed, Sonnet preserved visible strategy rules (flat bets, sub-list bids, walking away from challenges) but evaluated players at *"list minus* $\approx 25\%$*"* rather than computing proper valuations.
- **Budget Bookkeeping Errors:** Unspent draft capital is destroyed at the end of a draft. DeepSeek's *Value Hunter* left \$92–\$154 of its \$200 budget unspent per draft (vs. \$0–\$38 for control seats). Luna's *Value Hunter* forfeited \$35 per block (vs. \$0.84 for unprompted Luna) and frequently bid conservatively *"to preserve budget for the final roster slot"* while filling that exact final slot.
- **Unit and Roster Misinterpretations:** DeepSeek misinterpreted fantasy points per game as dollar valuations (e.g., treating *"13.68 FP/game"* as a \$13–\$15 value). When assigned to target star players, it allocated \$167 of \$200 to two elite goalies despite roster rules permitting only one starting goalie.

Strategy Retention and Behavioral Drift Because strategy text is included in every prompt and restated in scratchpads ($21/22$ Luna *Contrarian* seats), agents do not lose context. Fixed numeric rules (e.g., Terra's contest rules, Luna's *Sharp* caps, and *Grinder/Value Hunter* bet sizes) remain stable across blocks. Behavioral drift occurs primarily in high-variance profiles following losses:

- **Star Chaser:** Stake sizes dropped from 44% to 26–29% of bankroll between Blocks 1 and 3, while top-two spend shares declined from 0.71 to 0.55.
- **Contrarian:** Average stakes decreased from 42% to 31% of bankroll.
- **Profile Mentions:** Explicit profile references in decision logs declined over time (*Value Hunter*: $0.65 \rightarrow 0.54$; *Table Shark*: $0.25 \rightarrow 0.11$).

Table 4: Strategies screened on more than one model. Entries are the strategy seat's final bankroll minus the mean of the unprompted seats of the same model in the same room, averaged over rooms, ± one standard error. Luna's screens use three rooms and its intervals are correspondingly wide.

Strategy	Terra	DeepSeek	Luna
SCHEDULE TIEBREAK (ST)	$+18.88 \pm 4.39$	$+13.42 \pm 5.71$	$+2.16 \pm 5.50$
MEAN REVERSION (MR)	$+12.70 \pm 4.70$	—	$+6.75 \pm 10.05$
ROOM READER (RR)	$+9.01 \pm 3.61$	—	$+3.57 \pm 13.70$

This drift is driven by self-generated reflection notes overriding initial profile instructions. For example, a *Contrarian* scratchpad noted: *"Minimum 5; avoid aggressive bets absent clear lineup edge"*, leading to minimum $5 bets throughout Block 3. Winning seats maintained original profile parameters. Conversely, scratchpads can aid recovery: DeepSeek applied SCHEDULE TIEBREAK more effectively in Block 2 than Block 1 by preserving schedule insights across rounds.

A.4 CROWDING OUT

Crowding can manifest at two levels. **Behavioral interference** occurs when copies directly alter one another's opportunities, such as by targeting the same players or driving up prices. **Payoff erosion** occurs when a strategy's financial advantage shrinks. We track $\mathsf{PriceRatio} = \mathsf{PricePaid}/\mathsf{ListPrice}$, defined as the price a seat pays for an athlete relative to their list price, averaged over the seat's acquisitions in a block.

SCHEDULE TIEBREAK The strategy's advantage fell at higher saturation, but not because of behavioral interference: copies did not bid against one another. Strategy mechanism evaluates usable starts relative to each agent's own roster, so copies with different rosters tend to value different players; indeed, direct stealing among copies did not occur. Scarcity came from a *dilution of a shared field-level opportunity*: when another strategy copy enters, each copy competes indirectly for the same weak opponents even without contesting the same players.

ROOM READER The strategy also lost its advantage as saturation increased in the Projection field, accompanied by deterioration in both roster quality and realized scoring. The entry's edge is buying below market when the room overpays early or turns passive, and that discount largely disappeared: strategy seats' PriceRatio rose from 0.84 at $k = 1$ to 0.95 at $k = 2$ (SE = 0.04). The money moved accordingly. Projection remained exploitable and in fact lost more per block (-0.89 to -4.87), but the surviving unprompted seat captured that value (-1.20 to +3.09) rather than the strategy seats. Thus each additional disciplined copy removes one sloppy bidder and competes for the same underpriced players, so acquisition prices move toward list for every copy.

BARGAIN HUNTER This strategy produced the clearest behavioral interference. Because copies use similar valuations, they frequently targeted the same opportunities: 61% of contests were copy-versus-copy, and the PriceRatio advantage fell by 42%. As such, strategy players' acquisition advantage fell as players of BARGAIN HUNTER increased.

A.5 STRATEGIES ACROSS MODELS

The main text screens strategies on Terra because it follows strategy prompts most closely. Table 4 reports strategy performance when applied to alternative models. The same strategy yields varying returns depending on the executing model: SCHEDULE TIEBREAK provides $+\$18.88 \pm 4.39$ per run to Terra and $+\$13.42 \pm 5.71$ to DeepSeek, but only $+\$2.16 \pm 5.50$ to Luna. A strategy measured on an agent that fails to effectively translate rules into decisions understates the strategy's latent value. Consequently, a single performance metric per strategy is no more generalizable across models than model performance is across experimental fields. Most of this was due to the inability of certain agents to execute strategy requirements as described in Appendix A.3.

B Additional Experiments

B.1 The Fixed-Price Draft

The main text reports results fitted on auction runs. The fixed-price draft is a different game: a player costs his list price to whoever takes him first, so there is no contest, no premium to pay, and no budget pressure from rivals. We therefore fit it separately rather than pooling the two, and report it here. Figure 5 gives the ranking, estimated identically to the auction ranking in Figure 4: Plackett–Luce over draft blocks ranked by net money, with `stats` anchored at 1500 and intervals derived from a cluster bootstrap over whole runs.

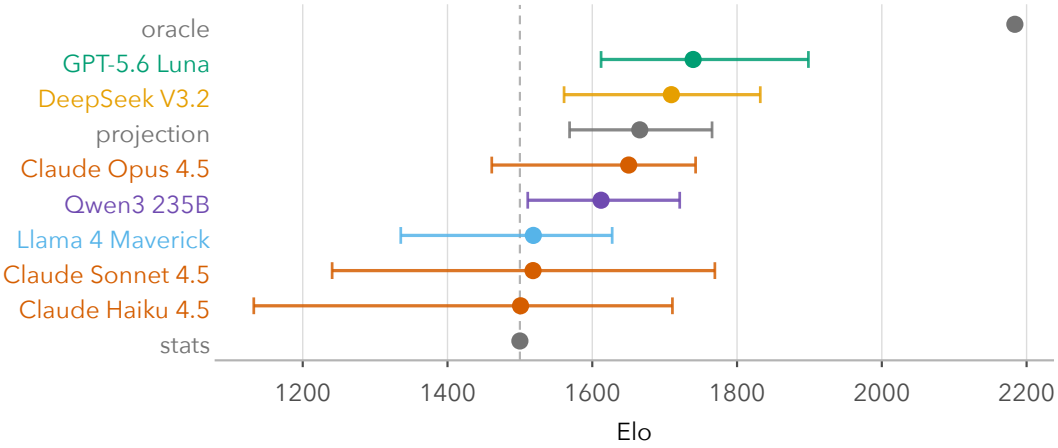


Figure 5: **Agent ranking under the fixed-price draft.** Plackett–Luce ability over 19 runs and 52 blocks, with `stats` anchored at 1500. Marks are coloured by vendor; scripted seats are grey. Hollow marks survive under 95% of the run-level bootstrap resamples. Compare Figure 4, which is the same estimator on the auction corpus.

Models differ substantially in draft format sensitivity. Luna, DeepSeek, and Qwen performed significantly better in static drafts, whereas Haiku experienced lower performance (Table 5).

Table 5: Model performance breakdown across Auction vs. Static (Fixed-Price) draft modes.

Model	**Auction** (Norm. Rank / Net per Block)	**Static** (Norm. Rank / Net per Block)
Luna	0.41 / +$1.9 (68 blocks)	0.29 / +$7.4 (35 blocks)
DeepSeek	0.56 / −$3.4 (72 blocks)	0.34 / +$3.2 (30 blocks)
Qwen	0.64 / −$4.8 (59 blocks)	0.53 / −$2.0 (31 blocks)
Opus	0.40 / +$4.0 (43 blocks)	0.42 / +$3.7 (10 blocks)
Haiku	0.49 / +$0.6 (25 blocks)	0.67 / −$4.8 (12 blocks)
Maverick	0.78 / −$9.7 (27 blocks)	0.70 / −$9.8 (20 blocks)
Sonnet	0.44 / +$2.0 (69 blocks)	0.54 / +$1.5 (6 blocks)
`stats`	0.66 / −$7.4 (69 blocks)	0.65 / −$7.2 (49 blocks)

C Prompts

Appendix C gives every prompt an agent can receive. Section C.1 covers the environment prompt, assembled once per seat; Section C.2 pairs each strategy's description with the exact text appended for it; Section C.3 lists the scoring rubrics.

C.1 Environment Prompts

Each agent receives one system prompt, built from the fragments below in this order: arena instructions, database research rules, table list, roster rules, draft rules, auction parameters, information

visibility, changing opponents, scratchpad, weekly rules, cross-block memory, strategy block, and opponent strategies. Bracketed labels such as [only if . . .] mark fragments or sentences that depend on the run configuration; values in braces such as {N} are filled from it. The weekly line-up and betting prompts arrive later, as separate user messages.

Seat	Environment prompts	Fundamentals	Strategy text
Unassigned	Yes	No	No
Baseline (`fundamentals`)	Yes	Yes	No
Assigned a strategy	Yes	Yes	Yes

C.1.1 Arena Instructions

Prompt for Arena Instructions

Objective: You are one of several agents competing in a multi-week fantasy hockey tournament. Your score is the money you hold at the end.
Players and teams are identified by code: every player is a `P####`, every team a `T####`. Those codes are their identities—there is nothing behind them to look up. Everything you know comes from your tools, as of a fixed cutoff.

What Arrives:

- STATE OF AFFAIRS, once per draft block: the run's parameters, the standings, and the prior-week history this run allows you to observe.
- Each turn: your roster and open slots, every opponent's roster, the tokens already taken, and the pick number.

Tools: Use only the tools supplied with the current request. Their definitions are the authority on parameters and whether a call is free. Research calls spend from your tool-call budget, and a rejected research call still spends one.
Make at most one tool call per response. Wait for its result before deciding which tool call, if any, is worth making next.

Rules:

- Once a player is awarded to an agent, that player is unavailable to everyone else; no roster may contain a player twice.
- The eligible pool is players who appeared in a box score in the last 30 days.
- A pick that is taken, off-pool, or illegal is rejected and costs a retry.
- Refer to players and teams only as `P####` / `T####`.

Figure 6: Core arena prompt defining the objective, observable state, tool-use constraints, and game rules.

C.1.2 Database Research

Prompt for Database Research

Research Surface: Use the database tools supplied with the request.

Query Rules: `db_query` accepts one read-only `SELECT`: no writes, `PRAGMA`, `ATTACH`, or semicolons.

Result Limits: At most 200 rows return, and an identical repeated query returns nothing new. Database views are already filtered to your cutoff; `fixtures` is the exception and includes upcoming games without future results.

Tables: Call the schema tool for what a column means and the values it takes.
{one line per queryable table: `table` — column, column, . . . , read from the live database}

`notes` — the only text in the database: [only if notes are enabled] `details` and `fantasy_details` are free text written about one player on one day: injuries, line-up changes, call-ups, confirmed goalie starts, usage changes. Nothing else here reports any of that—the other fifteen relations are box scores and tracking numbers, and they cannot tell you that a player is hurt or has moved onto the top power play.
Match with `player_token`, filter with `category` and `date`, and read `details`. `LIKE` works. Visibility follows publication date, not the date a note is about, so a note published before your cutoff about an upcoming game is legitimately yours to read.

Figure 7: Prompt defining the database research interface, its query and information constraints, and the table list.

C.1.3 Roster and Lineup Rules

Prompt for Roster and Lineup Rules

Position Encoding: In the data the `position` column holds C/L/R = forward; D = defenseman; G = goalie.

Fixed-Slot Roster: You draft {N} players into fixed slots: {slot list}.
Slot rules: {slot} takes {a forward — a defenseman — a goalie — any position}, ...
Every player you draft plays every week.

Squad Larger Than the Starting Line-up: You draft {N} players. The draft is position-free: any player fits any slot.
Each week, including weeks between drafts, you start {M} of them: {slot} takes {...}, ... Only those score. Your last picks are restricted to positions you still need, so that a legal starting line-up always remains possible.

Figure 8: Roster prompt variants for runs with fixed slots and runs with a squad larger than the weekly starting line-up.

C.1.4 Draft Rules

Prompt for Fixed-Price Draft

Budget: Drafting is paid. You hold two separate pots and they never mix:

- Betting bankroll—what you wager each week, and your final score.
- Draft budget—refreshed at the start of every draft block, spent only on acquiring players. Anything you do not spend is forfeited. It does not carry to the next block and it never becomes betting money.

Prices: Every player has a list price, and you pay it when you draft them. Look prices up with `draft_price_lookup`—free, never costs tool budget. Call it with no arguments for the full undrafted market, or pass `player_tokens` to check specific players.
Market price is not the same as player value; use your research tools to judge value.

Roster Feasibility: You may not strand yourself. A pick that would leave you unable to fill your remaining slots is rejected, and the rejection tells you your ceiling for that pick. Your remaining budget and a per-slot ceiling appear in the draft state each turn.

`draft_price_lookup` Tool Description: Draft-money prices and your remaining draft budget. FREE—never costs tool budget. Call it with NO arguments to get the full price board (every undrafted player and its price), or pass `player_tokens` to price specific players. This is the market price—what a player costs, which is not the same as what a player is worth.

Note Returned With Each Lookup: Fixed draft: you pay exactly this price and players cannot be taken from you. Unspent draft money is forfeited at the end of the block.

Figure 9: Prompt for the fixed-price draft mechanism and the price-lookup tool.

Prompt for Auction Draft

Budget: Drafting is paid and contested. You hold two separate pots and they never mix:

- Betting bankroll—what you wager each week, and your final score.
- Draft budget—refreshed at the start of every draft block, spent only on acquiring players. Anything you do not spend is forfeited. It does not carry over and it never becomes betting money.

Pricing: Every player has a list price, but it is a market reference rather than a reserve. You may claim below list price, subject to this run's minimum legal claim price, but a low claim is easier for a rival to steal. `draft_price_lookup` is free: call it with no arguments for the full undrafted market, or pass `player_tokens` for specific players.

Prompt for Auction Draft (continued)

Market price is not the same as player value; use your research tools to judge value.

Round Structure: Order is randomized every round. Within a round you hold two separate rights:

- your turn—you will acquire exactly one player this round;
- your steal—you may contest one other agent's claim this round.

When it is your turn you name a player and the price you offer. Then every agent that still has both a turn and an unused steal bids secretly against your claim. The highest bid becomes the challenge.

Challenge Resolution: If you are challenged you either match the challenge price and keep the player at that higher price, or decline—in which case the challenger takes the player and you get a fresh turn this round to pick someone else. The claim you lost costs you nothing.
If you are the challenger and win, you get the player at your own bid and that was your turn. If you lose because they matched, you keep your turn and will still pick this round, but your steal is spent and you cannot contest again until next round.
Agents who already have their player this round cannot bid.

Roster Feasibility: You may not strand yourself: a claim or bid that would leave you unable to fill your remaining slots is rejected.

Auction Parameters: The minimum legal claim price is {price_min} draft money. A steal bid must raise the standing claim by at least {steal_min_increment}.

`draft_price_lookup` Tool Description: Draft-money prices and your remaining draft budget. FREE—never costs tool budget. Call it with NO arguments to get the full price board (every undrafted player and its price), or pass `player_tokens` to price specific players. This is the market price—what a player costs, which is not the same as what a player is worth.

Note Returned With Each Lookup: Auction draft: this is the market's list price, not a floor. You name your own price for a player and may bid under the list price—but a rival can then claim the player by outbidding you, and you only keep them by matching. Unspent draft money is forfeited at the end of the block.

Figure 10: Prompt for the auction draft mechanism, including claims, steals, challenge resolution, auction parameters, and the price-lookup tool.

C.1.5 Information Visibility and Memory

Prompt for Information Visibility and Memory

Information Visibility:
Private: STATE OF AFFAIRS contains only your own prior outcomes and bankroll. Opponent rosters remain visible during the live draft.
Standings: STATE OF AFFAIRS contains your own prior outcomes plus every agent's bankroll, but not opponents' prior rosters, bets, or fantasy points.
Full: STATE OF AFFAIRS contains every agent's prior roster, bet, fantasy points, net result, and bankroll.

Changing Opponents and Bankroll Resets: [only if opponents rotate between blocks] THE FIELD CHANGES BETWEEN DRAFT BLOCKS. This tournament has a fixed pool of participants and several draft blocks, and at each redraft you may be assigned a different group of opponents. Participant labels are anonymous but stable: the same label always denotes the same participant, so an opponent who disappears and returns is the SAME opponent, and it remembers the blocks it played.
Each block seats {arena_size} agents drawn from the pool. [only if `anchors` > 0] {anchors} of them play every block, and the rest rotate in and out.
Each block lasts {redraft_every} scoring weeks. Your betting bankroll resets to {bankroll} at the next redraft; unused betting money does not carry forward. Each block's gain or loss accumulates into the final standings. Your private scratchpad persists across blocks.
STATE OF AFFAIRS names who is seated for the current block and how many blocks each opponent has played, so you can tell a newcomer from a returning rival.
[only if `suggest_opponent_memory`] Opponents may have recurring drafting and betting tendencies. Note in your scratchpad who you faced and what they did, against their anonymous label, and apply it the moment they come back.

Scratchpad: [only if the scratchpad is enabled] `scratchpad_write` appends private notes that persist across picks and draft blocks. It is free and the current scratchpad is shown on every draft turn. It is the

Prompt for Information Visibility and Memory (continued)

only place a finding of yours survives a block boundary verbatim, so write down what you would not want to pay to look up twice.

Cross-Block Memory:
Full history, scratchpad on: You keep the record of every question you asked and every decision you made in prior blocks. You do NOT keep the rows those queries returned: the data is still in the database and can be queried again, but the output itself is gone. Copy anything you want to keep—a ranking you built, a rate you computed, a player you rejected and why—into your scratchpad before the block ends.
Full history, scratchpad off: You keep the record of every question you asked and every decision you made in prior blocks. You do NOT keep the rows those queries returned: the data is still in the database and can be queried again, but the output itself is gone. Nothing you have not restated in your own words will survive.
Scratchpad only, scratchpad on: You keep ONLY your scratchpad. Every query you ran, every result and every decision is discarded at the block boundary—the notes you kept as you went are the whole of what survives. Copy anything you want to keep—a ranking you built, a rate you computed, a player you rejected and why—into your scratchpad before the block ends.
Scratchpad only, scratchpad off: You keep ONLY your scratchpad. Every query you ran, every result and every decision is discarded at the block boundary—the notes you kept as you went are the whole of what survives. Nothing you have not restated in your own words will survive.

Figure 11: Prompt variants governing information visibility, opponent rotation, persistent scratchpad notes, and cross-block memory.

C.1.6 Weekly Decisions

Prompt for Weekly Decision Context

Decision Order:
[squad larger than line-up, bet after line-up] You hold your squad for several weeks. Every week you name a starting line-up from it—only the players you start score—and then place a bet.
[squad larger than line-up, bet before line-up] You hold your squad for several weeks. Every week you place a bet, and then name the starting line-up from it—only the players you start score.
[squad equals line-up] Every week you place a bet on the team you are holding.

Opponent Information: Everyone's squad is public—the draft happens in the open.
[bet after] Line-ups are revealed BEFORE betting, so you bet knowing exactly which players every agent is fielding.
[bet before] Betting happens BEFORE anyone names a line-up, so you know each rival's squad but not which of them they will start—and they do not know yours.

Research Context: Each week opens with the block so far in front of you: every question you asked in your survey, your draft and your earlier weeks, and every decision you made. The ROWS those queries returned are not carried—re-query what you still need, and note that you can see what you have already asked, so asking it again costs you a call for nothing.
[weekly budget $>$ 0] You get {weekly_tool_budget} research calls for the week, shared between those decisions, so you can check what the draft could not tell you: how many games each of your players has this week, who they face, and how they have been going since you drafted them. The week's line-up and bet share one conversation, so anything you look up for one is still there for the other.
[weekly budget $=$ 0] You get no research calls during a week: decide from your squad summary, the standings, and your scratchpad.

Figure 12: Prompt defining the weekly decision sequence, observable opponent information, and research budget.

Prompt for Weekly Line-up Selection

Set this week's starting lineup. You own more players than you can start; only the players you start score this week, and the rest sit out.
Slot rules: {slot} takes {a F — a D — a G — any position}; .... Every starting slot must be filled by a different player you own.
Choose on this week's prediction window and this week's scoring weights—who plays the most games, who faces the softest opponents, and which categories the scoring rewards. A player who was right last week may be wrong this week.

Prompt for Weekly Line-up Selection (continued)

[research budget left this week] You may research first: use your tools to check this week's schedule and recent form before answering, then return ONLY the JSON. Each call spends from this week's budget, which the line-up and the bet share.
[no research budget left] Do not call tools.

Output Format: Return ONLY JSON:

```
{"lineup":{"{slot}":"<player_token>", ...},
 "reasoning":"<short>"}
```

using the `player_token` values from `your_squad`.

Figure 13: Prompt for weekly starting-line-up selection.

Prompt for Weekly Betting

Now place this week's bet from your bankroll. Settlement is parimutuel: every agent antes its bet into a shared pot, and the pot is split in proportion to (bet × your team's actual fantasy points).
You profit only if your team beats the bet-weighted field average—betting big when you underperform loses money. Your final score for the whole run is the money you end with.
[research budget left this week] You may research first: use your tools to check this week's schedule and recent form before answering, then return ONLY the JSON. Each call spends from this week's budget, which the line-up and the bet share.
[no research budget left] Do not call tools.

Output Format: Return ONLY JSON:

```
{"bet":<number>,"reasoning":"<short>"}
```

where `bet > 0` and `bet <= your bankroll`.

Figure 14: Prompt for weekly parimutuel betting.

C.1.7 Strategy Block

The strategy block is appended to the end of the system prompt for the baseline seat and for every seat assigned a strategy; unassigned seats do not receive it. For an assigned seat, the strategy text (Figures 17–23) is inserted directly after the header, prefixed with a dash.

Prompt for Strategy Block

Header:
[baseline seat] HOW TO PLAY. You have not been assigned any particular strategy or style, and you are not expected to adopt one. What follows is the set of fundamentals a competent player works from. Play them properly and consistently, and where a decision is close take the simpler and safer side of it.
[assigned a strategy] YOUR ASSIGNED STRATEGY. You have been directed to play this strategy. This is how you think about the game: let it drive how you value players, how you spread your budget, how much risk you accept and how you bet. Follow it deliberately, in the draft and in every weekly decision, and prefer the choice it implies over the choice that merely looks safe.
[assigned a strategy] - {strategy text}

Fundamentals:
THE FUNDAMENTALS.

WHAT A PLAYER IS WORTH

- The scoring table decides it. Points come from particular statistics at particular weights, so a table that pays for blocks and saves makes different players valuable than one that pays for goals. Read it before you form an opinion about anybody, and read it again if it changes between blocks.
- Points accrue per game, so the number of times a player's team plays inside the scoring window multiplies everything else about them.
- A player who does not play scores nothing. Injuries, suspensions, healthy scratches and who is actually starting in goal decide whether a player produces at all; ice time, line placement and power-play usage decide how much.

Prompt for Strategy Block (continued)

- A handful of games is mostly noise, and short runs of extreme production usually regress. Recent evidence is worth more for what it tells you about role, a promotion up the lineup, an injury to the man ahead, a new linemate, than for what it says about scoring rate.

THE AUCTION

- You must be able to fill every slot. Whatever you commit early, a legal roster still has to be completed afterwards. Work out what your remaining slots will cost before you bid, not after.

THE WEEK

- You own more players than you can start, and the line-up is a decision you make again every week. Points scored by a player sitting on your bench do not count. Slots constrain who can go where, so check that the players you want to start can legally fill the slots you have.
- You are eliminated when your bankroll falls below the minimum bet, and an eliminated agent stops playing.

THE OTHER AGENTS

- You can see what they own, what they paid for it, and what they wager each week. That is free information, there whether or not you choose to act on it, and it is the only direct evidence you have about how the rest of the room is thinking.
- [only if `opponent_tracking`; off by default] Keep track of how each opponent plays—what they chase, what they overpay for, how they size their bets—and carry that read from block to block. An opponent you have faced before will probably play the same way again, so a rival who bid recklessly for stars in the last block is likely to do it in the next one. Adjust to what you have learned about the agents actually in front of you rather than replaying the same block each time.

MEMORY

- Query results do not survive the reset between draft blocks; your notes do. Anything you worked out has to be written down before the block ends or it is gone, and re-deriving it later costs research calls you could have spent on something you do not already know.

Closing Instruction:
[baseline seat] In the `reasoning` you give with every pick and bet, say which fundamentals drove the decision. Do not invent a grand strategy to justify a choice—if the answer is that the player was available, played four games and fit the slot, say exactly that.
[assigned a strategy] In the `reasoning` you give with every pick and bet, say where you are applying it and how. If you decide against it on a particular decision, say so and why—a departure you can justify is more useful than silent compliance.

Figure 15: Strategy block appended to the system prompt of the baseline seat and of every assigned seat: header, strategy text (assigned seats only), shared fundamentals, and closing instruction.

C.1.8 Opponent Strategies

Prompt for Opponent Strategies

[only for seats listed in `expose_strategies`; appended after the strategy block. Only opponents with an assignment are listed.]
OPPONENT STRATEGIES. You know what each opponent was told to do. Use it: anticipate which players they will chase, and take or avoid them accordingly.

Headline Detail: [`expose_strategies_detail` = `headline`; one line per assigned opponent]

- Agent {agent_id} was directed to play: {first sentence of the opponent's strategy, title-cased, e.g. Sell The Flavour Of The Week To The Room}
- [baseline opponent] Agent {agent_id} was directed to play: No Assigned Strategy (Fundamentals Only)

Full Detail: [`expose_strategies_detail` = `full`; one block per assigned opponent]
--- Agent {agent_id} was directed to play: ---
{the opponent's full strategy text}
[baseline opponent] No assigned strategy: it plays the fundamentals only.

Figure 16: Prompt revealing opponents' assigned strategies, in headline and full detail.

C.2 STRATEGIES

Each strategy below is given first in prose and then as the exact text appended to the seat's system prompt. Strategies come in three families: *valuation* changes how a seat judges an athlete's worth, *market* how it responds to prices, and *contest* how it claims and challenges.

C.2.1 MEAN REVERSION (MR), VALUATION

Short runs of strong production carry limited information about an athlete's true rate. Early-season batting averages, for example, predict the rest of the season better once they are shrunk toward the group average (Efron & Morris, 1975), and NHL playoff goaltenders who have been saving well show no tendency to keep doing so (Ding et al., 2025). MEAN REVERSION makes this belief precise. It ranks athletes on their 60-day rate and uses the 7-day window only to detect changes in role, such as an injury, a benching, or a promotion to a top line. The prompt states the correlations we measured between each lookback window and the following week's fantasy points (0.31 for 7 days, 0.43 for 30 days, and 0.47 for 60 days), so the agent can check the claim against the data.

Prompt for MEAN REVERSION

YOU BELIEVE MOST OF A HOT STREAK IS NOISE. Five good games is a small sample, and the player who produced them is usually about as good as they were a month ago.
This is measurable on the data you can query, and it is worth knowing precisely. Taking every skater in a season and asking which lookback best predicts the FOLLOWING week: a seven-day rate correlates +0.31 with what happens next, a thirty-day rate +0.43, and a sixty-day rate +0.47. The longer the window, the better the forecast. The short window is the one that feels most informative and is the least.
So when you build a ranking, reach back sixty days rather than seven, and treat a player's recent surge as weak evidence that they have changed. Use the short window for what it is actually good for—whether someone is hurt, benched, or newly on a top line—and the long window for how good they are.
Where a decision is close and the two windows disagree, side with the long one. How you spend and how you bet are yours to judge.

Figure 17: Prompt for the MEAN REVERSION strategy.

C.2.2 SCHEDULE TIEBREAK (ST), VALUATION

Fantasy hockey players track each team's schedule and favor athletes whose games add points without forcing another starter to the bench (Allen, 2022). SCHEDULE TIEBREAK uses schedule only to break ties between athletes of similar quality. It counts *usable starts* rather than raw games: an extra game is worth nothing if the athlete would sit on the bench behind a stronger teammate that week.

Prompt for SCHEDULE TIEBREAK

PLAYER QUALITY COMES FIRST; USABLE GAMES BREAK CLOSE TIES.
Do not sacrifice meaningful player quality simply to chase schedule advantages. Instead, when two players project similarly, prefer the player whose games are more useful during the period you will own them.
Consider the total games during the block; whether those games occur when your other strong players are less likely to occupy the same starting slots; whether roster competition could leave otherwise valuable games on your bench; and whether the schedule creates more opportunities for the player to enter your actual starting line-up.
Think in terms of USABLE STARTS rather than raw games scheduled. A four-game player is not automatically better than a three-game player if the extra game is unlikely to enter your line-up, and schedule should not rescue a substantially inferior player.
Your edge is using schedule as a disciplined tiebreaker among genuinely comparable players.

Figure 18: Prompt for the SCHEDULE TIEBREAK strategy.

C.2.3 ROOM READER (RR), MARKET

Prices in an auction draft depend on how the room spends. Field data from fantasy basketball auction drafts show that managers tend to overbid for the first two or three players they win and then underbid for the next few (Boudreau & Shunda, 2016). ROOM READER is built to exploit such patterns. It enters the draft with valuations but no fixed list of targets. If rivals overpay early, it saves money for later. If rivals are passive and good athletes go cheaply, it buys them at once.

Prompt for ROOM READER

LET THE ROOM TELL YOU WHERE THE BARGAINS ARE.
Enter the draft with valuations, but do not enter with a rigid list of players you must own.
Watch how aggressively the room is spending relative to the quality of the players being acquired.
If opponents are paying unusually high prices early, become more selective and preserve enough purchasing power to attack better values later. If the room is unexpectedly passive and high-quality players are available below what you believe they are worth, BUY THEM NOW rather than saving money for an imagined future bargain.
Judge every acquisition by the alternative opportunities the current room is creating.
Do not bid merely because another player appears 'too cheap'. You are not responsible for making the market fair. Only continue bidding when you are genuinely willing to own the player at that price.
Your strategy must remain fluid: exploit the pricing mistakes the room actually makes rather than the mistakes you expected it to make.

Figure 19: Prompt for the ROOM READER strategy.

C.2.4 BARGAIN HUNTER (BH), MARKET

BARGAIN HUNTER is a fixed-rule strategy that never pays more than an athlete's list price. Rules that cap each bid at a set multiple of an athlete's fair price are among the simplest bidding strategies analyzed for fantasy auction drafts (Anagnostopoulos et al., 2016). BARGAIN HUNTER relies on the observation that many seats claim good athletes for the minimum price and give them up when challenged.

Prompt for BARGAIN HUNTER

BUY THE ROOM'S BARGAINS, AND NEVER PAY ABOVE MARKET.
Rank players by their sixty-day rate, their role and their team's schedule. Claim at the minimum legal price.
Before your own claim each round, look at the claims already standing. Among them find the player with the highest list price whose standing claim is below half of that list price, and contest him at half of his list price (list 30.00 $\rightarrow$ 15.00). That bid takes him from any seat not committed to him, and still buys him below market. If your contest wins, he is your pick for the round; if it loses, claim from your own board at the minimum legal price.
When a rival contests one of your claims, match while the challenge is at or below the player's list price, and decline above it. You do not pay above market for anyone, ever.
Money unspent at the end of the block is destroyed; it does not become betting money and does not carry forward. From the midpoint of the draft onward, if you are still holding more than half your budget, stop claiming at the floor and bid list price for the best player on your board.
Your edge is that the room claims good players for a dollar and lets them go for half of list, so you buy at that price all block and finish with a full roster bought entirely below market.

Figure 20: Prompt for the BARGAIN HUNTER strategy.

C.2.5 LOYALIST (LO), CONTEST

LOYALIST is a commitment strategy. It never challenges, and it defends its own claims up to $1.5\times$ list. This cap matches the fair-price bidding rule of Anagnostopoulos et al. (2016), which bids up to $3/2$ of an athlete's fair price and, in their simplified model of a fantasy auction draft, guarantees a manager at least a third of an equal share of total team value, whatever rivals bid. The prompt adds a reason based on reputation: a seat that folds once invites challenges every round.

Prompt for LOYALIST

YOU DO NOT GET OUTBID ON YOUR OWN PLAYERS.
Rank players by their sixty-day rate, role and schedule, and ignore seven-day streaks in either direction. Every round, claim the best available player who fits a slot you still need, at the minimum legal price; goalies are ranked on the same table as everyone else, so take one when he is the best value on your board.
Never contest another seat's claim; a contest spends your pick of the round on someone else's choice.
When a rival contests one of your claims, MATCH it, at any price up to 1.5 times the list price of that player (list 20.00 $\rightarrow$ match anything up to 30.00), provided that after paying you can still fill every remaining slot at the minimum legal price. Do not reason about whether this particular challenge is worth matching: the player was your pick because he was the best available, letting him go means the next best, and a rival who

Prompt for LOYALIST (continued)

sees you fold will contest you every round. Decline only when the challenge is above 1.5 times list or would leave you unable to complete the roster.
Your edge is that your roster is exactly the roster you drafted, and nobody takes your best players for a dollar.

Figure 21: Prompt for the LOYALIST strategy.

C.2.6 DECOY (DC), CONTEST

When bidders have limited budgets, a bidder can gain by bidding on one item to raise the price a rival pays, leaving the rival less to spend on later items (Benoît & Krishna, 2001), and laboratory experiments confirm that budget constraints change how people bid in sequential auctions (Pitchik & Schotter, 1988). DECOY adapts this idea to our claim and challenge auction, where each rival has not only a limited budget but also a single challenge per round. It keeps a *bait list* of athletes with the most fantasy points over the last seven days who are not in its own top fifteen, since these are the athletes a rival chasing recent form will contest.

Prompt for DECOY

SELL THE FLAVOUR OF THE WEEK TO THE ROOM.
Rank players by their sixty-day rate, their role and their team's schedule. A seven-day surge is weak evidence about how good a player is. Keep a second, separate list, the BAIT LIST: the ten skaters and three goalies with the most fantasy points over the last seven days under this run's scoring, minus anyone your own ranking places in your top fifteen. Goalies on a short run of starts belong on it. These are the players the room is about to overpay for.
IN THE DRAFT. Bait in a round means this: your FIRST claim of the round is the top player on the bait list, at the minimum legal price. If anyone contests it, at any price, decline and let them have him, then claim the player you actually want at the minimum legal price. If nobody contests it, you own him; accept that and move on. Bait in rounds 1 and 2 no matter what. In rounds 3 and later, bait only if a rival who has contested one of your claims earlier in this draft has not yet used his turn this round; otherwise claim the player you actually want at the minimum legal price and nothing else.
Never contest another seat's claim. When a rival contests one of your real claims, match it if the challenge is at most one fifth of the list price, and decline and pick again otherwise; you do not pay for players.
Your edge is that a seat which bids on form spends its steal and its pick of the round on the player you least wanted to keep, while your real target goes through for a dollar.

Figure 22: Prompt for the DECOY strategy.

C.2.7 TAX COLLECTOR (TC), CONTEST

TAX COLLECTOR applies the same budget-draining logic (Benoît & Krishna, 2001) to the seats that refuse to let go of their claims. It observes that most seats give up a challenged athlete cheaply, while a seat that keeps matching can be made to pay more each time.

Prompt for TAX COLLECTOR

TAKE HOT PLAYERS FOR A DOLLAR, AND BILL THE SEAT THAT WILL NOT LET GO.
Rank skaters by their sixty-day rate, role and schedule, and claim them at the minimum legal price. Rank goalies by fantasy points over the last seven days and take one, the hottest available, at the minimum legal price, in the first round where he is the best value on your board. Own one goalie only.
IN THE DRAFT, BEFORE YOUR OWN CLAIM. Look at the claims already standing. Among them, find the players who are in the top ten skaters or top three goalies by fantasy points over the last seven days. If there are any, contest the highest-ranked of them. Your bid depends on the claimant's record in this draft: if he has never matched a contest, bid the minimum legal raise; if he has matched once, bid the list price; if he has matched twice or more, bid 1.5 times the list price (list 20.00 → 30.00). If no standing claim is on a hot player but some seat has matched a contest earlier in this draft and has a claim standing, contest his best standing claim on the same ladder. Compute the bid and use it every round; a seat that keeps matching is paying more than the player is worth each time, and that is the point. If your contest wins, that player is your pick for the round. If nobody has a claim standing, claim from your own board at the minimum legal price.
When a rival contests one of your claims, match it if the challenge is at most one fifth of the list price, and decline and pick again otherwise.

Prompt for TAX COLLECTOR (continued)

Your edge is that most seats let a contested player go for a dollar, and the one seat that does not can be made to pay for every star it keeps.

Figure 23: Prompt for the TAX COLLECTOR strategy.

C.3 SCORING RUBRICS

Each run settles fantasy points with one scoring rubric, selected by `scoring_file`. A player's points for a week are the sum over games of each statistic times its weight. Agents never see the config file. They see the normalized rubric below, filled out to the full category board, as a JSON object: under `scoring_this_block` in STATE OF AFFAIRS, once per draft block, and under `scoring` in the weekly line-up and betting messages.

Balanced Rubric

Prompt for Balanced Scoring Rubric

```
{
  "goals": 8.0,
  "assists": 5.0,
  "points": 0.0,
  "plusMinus": 0.0,
  "pim": 0.0,
  "powerPlayPoints": 0.0,
  "powerPlayGoals": 0.0,
  "powerPlayAssists": 0.0,
  "shortHandedGoals": 0.0,
  "shortHandedAssists": 0.0,
  "shortHandedPoints": 0.0,
  "gameWinningGoals": 0.0,
  "overtimeGoals": 0.0,
  "shots": 1.5,
  "hits": 0.0,
  "blocks": 1.8,
  "takeaways": 0.0,
  "giveaways": 0.0,
  "faceoffsWon": 0.0,
  "faceoffsLost": 0.0,
  "shifts": 0.0,
  "timeOnIce": 0.0,
  "wins": 6.0,
  "losses": 0.0,
  "otLosses": 2.0,
  "saves": 0.75,
  "shotsAgainst": 0.0,
  "goalsAgainst": -3.5,
  "shutouts": 4.0,
  "goalieMinutes": 0.0
}
```

Figure 24: Scoring rubric object for `balanced`, as shown to agents.

Forward Heavy Rubric

Prompt for Forward Heavy Scoring Rubric

```
{
  "goals": 9.5,
  "assists": 5.5,
  "points": 0.0,
  "plusMinus": 0.0,
  "pim": 0.0,
```

Prompt for Forward Heavy Scoring Rubric (continued)


```
  "powerPlayPoints": 0.0,
  "powerPlayGoals": 0.0,
  "powerPlayAssists": 0.0,
  "shortHandedGoals": 0.0,
  "shortHandedAssists": 0.0,
  "shortHandedPoints": 0.0,
  "gameWinningGoals": 0.0,
  "overtimeGoals": 0.0,
  "shots": 1.5,
  "hits": 0.0,
  "blocks": 1.0,
  "takeaways": 0.0,
  "giveaways": 0.0,
  "faceoffsWon": 0.0,
  "faceoffsLost": 0.0,
  "shifts": 0.0,
  "timeOnIce": 0.0,
  "wins": 6.0,
  "losses": 0.0,
  "otLosses": 2.0,
  "saves": 0.7,
  "shotsAgainst": 0.0,
  "goalsAgainst": -3.5,
  "shutouts": 4.0,
  "goalieMinutes": 0.0
}
```


Figure 25: Scoring rubric object for `forward_heavy`, as shown to agents.

Defence Heavy Rubric

Prompt for Defence Heavy Scoring Rubric


```
{
  "goals": 8.0,
  "assists": 4.5,
  "points": 0.0,
  "plusMinus": 1.0,
  "pim": 0.0,
  "powerPlayPoints": 0.0,
  "powerPlayGoals": 0.0,
  "powerPlayAssists": 0.0,
  "shortHandedGoals": 0.0,
  "shortHandedAssists": 0.0,
  "shortHandedPoints": 0.0,
  "gameWinningGoals": 0.0,
  "overtimeGoals": 0.0,
  "shots": 1.3,
  "hits": 0.9,
  "blocks": 2.6,
  "takeaways": 0.0,
  "giveaways": 0.0,
  "faceoffsWon": 0.0,
  "faceoffsLost": 0.0,
  "shifts": 0.0,
  "timeOnIce": 0.0,
  "wins": 8.0,
  "losses": 0.0,
  "otLosses": 2.5,
  "saves": 0.9,
  "shotsAgainst": 0.0,
  "goalsAgainst": -4.0,
```

Prompt for Defence Heavy Scoring Rubric (continued)

```
  "shutouts": 5.5,
  "goalieMinutes": 0.0
}
```

Figure 26: Scoring rubric object for `defence_heavy`, as shown to agents.

Goalie Heavy Rubric

Prompt for Goalie Heavy Scoring Rubric

```
{
  "goals": 8.0,
  "assists": 5.0,
  "points": 0.0,
  "plusMinus": 0.0,
  "pim": 0.0,
  "powerPlayPoints": 0.0,
  "powerPlayGoals": 0.0,
  "powerPlayAssists": 0.0,
  "shortHandedGoals": 0.0,
  "shortHandedAssists": 0.0,
  "shortHandedPoints": 0.0,
  "gameWinningGoals": 0.0,
  "overtimeGoals": 0.0,
  "shots": 1.5,
  "hits": 0.0,
  "blocks": 1.8,
  "takeaways": 0.0,
  "giveaways": 0.0,
  "faceoffsWon": 0.0,
  "faceoffsLost": 0.0,
  "shifts": 0.0,
  "timeOnIce": 0.0,
  "wins": 12.0,
  "losses": 0.0,
  "otLosses": 4.0,
  "saves": 1.5,
  "shotsAgainst": 0.0,
  "goalsAgainst": -7.0,
  "shutouts": 8.0,
  "goalieMinutes": 0.0
}
```

Figure 27: Scoring rubric object for `goalie_heavy`, as shown to agents.

D DATA

D.1 SOURCES

The data are collected from the following sources:

- **NHL game API** (`api-web.nhle.com`): boxscores, play-by-play, schedules, standings, rosters, and NHL Edge player tracking.
- **NHL stats API** (`api.nhle.com/stats`): shift charts and per-game reports, including ice-time splits, power play, penalty kill, shot attempts, 5v5 rates, faceoffs, on-ice goals, goalie advanced statistics, and team power play and penalty kill statistics.
- **Player news**: 126,967 Daily Faceoff notes from 2011 to 2026, obtained from the site's archive and supplemented with Wayback Machine snapshots.

Coverage spans the 2010–11 through 2025–26 seasons and includes both regular-season and playoff games. NHL Edge tracking is available from the 2021–22 season onward. Preseason and All-Star games are excluded.

Fields removed. Some fields are dropped because they could identify a player or team, or because they consist primarily of display content that adds storage size without providing additional information. These include:

- names, jersey numbers, birth date and place, nationality, draft details, and awards;
- team names and logos, venues, headshots, links, broadcast data, and betting data;
- season and career totals that extend beyond the relevant cutoff.

Endpoints consisting primarily of such content, including player biographies and landing pages, draft records, and venue information, are not collected.
The resulting database contains $20,547$ games, 739,588 skater-game lines, 6.5 million play-by-play events, and 15.6 million shifts.

Agent-visible views	**Information exposed**	**Availability at cutoff t**
`games`, `skater_games`, `goalie_games`, `events`, `shifts`, `advanced_*`	Game results, box-score and advanced statistics, play-by-play events, and shifts	Games played on or before t
`standings`	Team records and standings	Most recent weekly snapshot on or before t
`rosters`	Player–team assignments	Latest assignment observed on or before t
`edge_*`	Season-level NHL Edge tracking statistics for skaters, goalies, and shots	Completed seasons only
`notes`	Anonymized player news	Notes published on or before t
`fixtures`	Game dates and matchups; no scores or outcomes	Full current-season schedule
`players`	Position, height, weight, and handedness	Not date-filtered

Table 6: Information exposed to agents through the database at weekly cutoff t.

D.2 Processing, filtering and optimization

Filtering.

- Only regular-season and playoff games are kept.
- Rows in the stats reports that cover more than one game are dropped. The remaining rows are joined to games by player and date.
- Shift rows without a game, player, period, or shift number are dropped.
- Notes are deduplicated and timestamped by publication time. Notes without a date are dropped. Note names are linked to NHL IDs through rosters and NHL player search.
- The roster view keeps one team per player per season: the latest team the player appeared for up to the cutoff. Without this, a traded player's former teams would be counted too.

Derived statistics. Categories missing from boxscores (special-teams splits, faceoffs, game-winning goals, and penalties drawn) are derived from play-by-play. Derived power-play goals match the boxscore in 399 of 400 sampled games, and the boxscore is used wherever the two differ. Rate and percentage fields are not stored, since they can be computed from the counts.

Optimization.

- **Collection.** Stats reports are requested once per game date rather than once per game (about 38k requests instead of 226k). Every response is cached, and the database is built offline from the cache.
- **Storage.** The main tables are indexed on date, player, and game, and notes are indexed on publication time, player, and category.
- **Queries.** Views are rebuilt only when the cutoff changes. A repeated identical query is flagged and returns nothing new.

- **Fan-out guard.** A query that aggregates historical rows together with future fixtures in the same `SELECT` is rejected. In such a query, every history row repeats once per fixture, which inflates `COUNT` and `SUM`. The error message tells the agent to aggregate each side separately first.

D.3 Anonymization

Players are represented only as `P####`, and teams are represented only as `T####`. Names are never stored, preventing models from directly using pretrained knowledge about player or team identities. Notes are scrubbed of names before storage. Team and player names are replaced with tokens, while any other capitalized name is replaced with `[PERSON]`. URLs and social-media handles are removed. Trophy names, weekdays, and months are protected from these replacements. After scrubbing, 8,034 notes contain `[PERSON]`.
Real dates and season identifiers remain visible.

D.4 Temporal Filtering and Leakage Controls

Views are rebuilt at each weekly cutoff t, and each week is scored using games from $t+1$ through $t+7$. An automated audit checks, at sampled cutoffs, that no dated view returns rows after t and that the schedule contains no results.
These controls address leakage through the database only. They do not prevent parametric leakage, whereby a model may draw on information already present in its pretrained knowledge about real NHL seasons (Section D.7).

D.5 Player Eligibility, Prices, and Scoring

Player pool. The eligible pool consists of players, including goalies, who recorded at least one game with at least 60 seconds of playing time during the 30 days preceding t.

Base value. Each player's base value is

$$v = 0.65r + 0.35s, \tag{1}$$

where r is fantasy points accumulated during the 60 days preceding t, divided by the maximum value in the eligible pool, and s is the corresponding value for the previous regular season.
Players without a previous season are priced using r alone. Fantasy points used for pricing are calculated using a fixed reference scoring board, regardless of the scoring board used by the particular experimental run.

Price. The player price is

$$\text{Price} = \max\left(1,\ c \cdot v^{1.3} \cdot u\right), \tag{2}$$

where

$$u \sim \text{Uniform}(0.8, 1.2) \tag{3}$$

is generated by hashing the experimental seed and player ID. The constant c is chosen so that the most expensive legal squad costs three times the draft budget of 200.

D.6 Experimental Windows and Inclusion

Runs use consecutive weekly cutoffs beginning in late October or November and have durations of 8, 12, or 24 weeks. The seasons included are 2013–14 through 2018–19 and 2022–23 through 2025–26. The 2019–20 through 2021–22 seasons are excluded because of pandemic-related disruptions. Table 7 lists the exact experimental windows.
A run is included only if every planned week settles successfully.

D.7 Leakage Audits

Several audits were conducted to assess potential leakage:

- **Identifier scan.** Tool outputs contained no names, raw identifiers, venues, or birthplaces.
- **Reasoning and tool-use audit.** Reasoning and tool-use artifacts were audited to verify that players did not rely on real player identities.

Table 7: Windows of rated runs.

Cutoffs (first → last)	Weeks	Runs
2013-10-28 → 2014-01-13	12	1
2013-11-11 → 2013-12-30	8	2
2014-10-27 → 2014-12-15	8	2
2014-10-27 → 2015-01-12	12	9
2015-10-26 → 2015-12-14	8	3
2015-10-26 → 2015-12-28	10	1
2015-10-26 → 2016-01-11	12	3
2016-10-31 → 2016-12-19	8	2
2016-10-31 → 2017-01-16	12	4
2016-10-31 → 2017-04-10	24	1
2017-10-30 → 2018-01-15	12	1
2018-10-09 → 2018-12-25	12	1
2018-10-29 → 2019-01-14	12	1
2018-10-09 → 2019-03-19	24	2
2022-10-24 → 2023-01-09	12	1
2023-10-23 → 2024-01-08	12	1
2024-10-28 → 2025-01-13	12	1
2025-10-27 → 2026-01-12	12	1

D.8 Windows of Rated Runs

E Generalizability

E.1 Game Configuration Knobs

Each run is defined by one YAML configuration file. Unknown keys are rejected.

Knob	Default	Options	Effect
Run setup			
`weeks`	required	list of dates	Weekly cutoff dates; one scoring week follows each.
`agents`	required	model specs, `stats`, `oracle`, `projection`	Seats in the game.
`seed`	none	integer	Seeds price noise, draft order, and tie-breaks.
`temperature`	0.0	≥ 0	LLM sampling temperature.
Roster			
`roster_slots`	F, F, D, G, FLEX	list of slot labels	Positions started each week. `SK` takes any skater; `FLEX` takes any player.
`squad_size`	10	$\geq$ number of slots	Players owned. Extra players form a bench, and the agent picks starters each week.
`position_taxonomy`	F / D / G	mapping	Which slots each position may fill.
Draft (`draft.*`)			
`mode`	static	static, auction	Static: turn order at list price. Auction: randomized order per round, claims at any price, rivals may steal a standing claim.
`budget`	200	> 0	Draft money per agent per block; unspent money is lost.
`redraft_every`	4	≥ 1	Weeks per draft block.
`order_schedule`	random (seeded)	agent orders per block	Fixes draft order per block.

Knob	Default	Options	Effect
`min_increment`	1	> 0	Auction: smallest raise over a standing claim.
`defend`	true	true, false	Auction: a claimant may match a steal and keep the player.
`dispossessed`	repick	repick, forfeit	Auction: a claimant who loses a player picks again for free, or receives the cheapest legal player.
Prices (`draft.pricing.*`)			
`window_days`	60	≥ 7	Recent-form window for value.
`season_weight`	0.35	$[0, 1]$	Share of value from the previous season.
`noise`	0.20	$[0, 1)$	Per-player price jitter, uniform $\pm$ this share.
`alpha`	1.3	> 0	Convexity: price $\propto$ value$^{\alpha}$.
`scarcity`	3.0	> 0	The most expensive legal squad costs this $\times$ budget.
`min_price`	1	> 0	Price floor.
`basis`	reference	reference, current	Value computed on a fixed reference board, or on the run's own scoring.
`visibility`	summary	none, summary, full	Market information shown: none, price percentiles per position, or every player's price.
Betting			
`bankroll`	100	> 0	Starting betting money.
`min_bet`	5	≥ 5	Minimum weekly stake.
`payout`	parimutuel	parimutuel, vs_field	Parimutuel: the pot is split in proportion to stake $\times$ fantasy points. `vs_field`: each agent receives stake $\times$ (its points / field average).
`bet_timing`	before	before, after	Bet before or after rivals' lineups are revealed.
`slot_bonus`	0	≥ 0	Winner-take-all bonus per starting slot.
Scoring			
`scoring` / `scoring_file`	reference board	category weights, or balanced, forward_heavy, defence_heavy, goalie_heavy	Fantasy points per category; only categories available from boxscores are allowed.
`scoring_schedule`	none	list of boards	Different scoring board per week.
Information			
`visibility`	full	private, standings, full	What agents see about rivals each week: own result only, plus bankrolls, or plus rosters, bets, and points.
`context_tokens`	model window	integer	Optional cap on context length.
Memory (`memory.*`)			
`policy`	full	full, scratchpad	Carried across blocks: all prior questions and decisions, or only the agent's notes.
`max_chars`	60,000	integer	Cap on carried memory.
`max_blocks`	8	integer	How many prior blocks memory reaches back.
Research tools (`tools.*`)			

Knob	Default	Options	Effect
surface	db	db, api	SQL over the database, or the HTTP endpoint catalogue.
notes	true	true, false	Access to player news.
predraft_budget	15	≥ 0	Research calls before each draft.
budget	2	≥ 0	Research calls per draft pick.
budget_mode	per_pick	per_pick, pooled	Budget per pick, or pooled across the draft.
weekly_budget	5	≥ 0	Research calls per scoring week (lineup and bet).
scratchpad	true	true, false	Private notes tool.
scratchpad_max_chars	4,000	integer	Scratchpad size; oldest notes drop first.
Treatments			
strategies	none	strategy names per seat	Adds a strategy prompt to a seat.
expose_strategies	none	true, false, seat list	Tells the chosen seats their rivals' strategies.
expose_strategies_detail	headline	headline, full	How much of the rival strategy is revealed.
opponent_tracking	false	true, false	Adds a prompt to track opponents.
Opponent rotation (rotation.*)			
enabled	false	true, false	Draws a subset of the agent pool for each block.
arena_size	none	≥ 2	Seats per block.
schedule	random	participants per block	Fixes who plays each block.
baselines_per_arena	none	0 to arena size	Seats reserved for non-LLM baselines.
anchors	0	0 to arena size	Agents seated in every block.
matchmaking_seed	none	integer	Seed for the draws.
bankroll_reset_each_block	false	must be true when rotating	Resets betting money each block.
suggest_opponent_memory	false	true, false	Prompts agents to remember opponents across blocks.

F CODE RELEASE

The code for all experiments in this paper is available at https://anonymous.4open.science/r/interactive_sports-5B4C.