CEO-Poker: **C**ontrolled **E**nsembled **O**pponent Modeling for Imperfect Information Game

by

Yuanheng Li

An honors thesis submitted in partial fulfillment

of the requirements for the degree of

Bachelor of Science

Business and Economics Honors Program

NYU Shanghai

May 2026

Professor Marti G. Subrahmanyam
Professor Christina Wang
Professor Wendy Jin

Faculty Advisers

Professor Emiliano Catonini

Thesis Adviser

Contents

Abstract

This thesis studies the trade-off between equilibrium robustness and exploitative profitability in Heads-Up No-Limit Texas Hold'em. In two-player zero-sum imperfect-information games, equilibrium-oriented strategies provide worst-case safety, but they need not maximize payoff against weak or systematically biased opponents. I propose CEO-Poker, a modular decision framework that combines opponent modeling, explicit equity estimation, a GTO-oriented policy, an exploitative policy, and a meta-controller that adaptively interpolates between safe and exploitative play. Empirically, CEO-Poker does not defeat Slumbot, a strong near-equilibrium benchmark, but it produces more stable trajectories than a baseline LLM agent and remains closer to a near-break-even regime. More importantly, when facing clearly non-equilibrium opponents, CEO-Poker-Equity substantially outperforms Slumbot in cumulative profit, showing stronger ability to monetize structured opponent mistakes. The results suggest that practical poker AI should not rely on pure equilibrium play or pure exploitation alone, but on controlled deviation from robust play, guided by numerical equity evaluation and opponent-aware adaptation.

Acknowledgments

I would like to express my deepest gratitude to my thesis adviser, Professor Emiliano Catonini, for his guidance, patience, and thoughtful feedback throughout the development of this thesis. His comments helped me clarify the theoretical framing of the project, refine the game-theoretic notation, and better connect the empirical results to the central question of equilibrium robustness and exploitative play. I am also grateful to Professor Marti G. Subrahmanyam, Professor Christina Wang, and Professor Wendy Jin for their support and guidance through the Business and Economics Honors Program. I would also like to thank Xinyi Yang for her consistent help in organizing the thesis process and communicating important program arrangements. Finally, I am thankful to everyone who offered advice, feedback, or encouragement during this project. I also owe a small but sincere debt to everyone who has ever played Texas Hold'em with me; knowingly or not, they all contributed to my understanding of imperfect information, risk, and questionable calls. This thesis benefited greatly from all of their support.

1 Introduction

Artificial intelligence has achieved remarkable success in perfect-information games such as chess and Go [Silver et al., 2016]. By contrast, poker remains a central benchmark for strategic reasoning under imperfect information, where agents must act without direct access to the opponent's private cards and must therefore reason over beliefs, incentives, and uncertainty. In this setting, equilibrium-based methods have produced highly competitive systems, yet they are not always optimized for practical performance against weak or systematically biased opponents. The central motivation of this thesis is that robustness and practical profitability are not identical notions. A strategy that is hard to exploit in theory may still leave money on the table when facing suboptimal opponents. This observation motivates the design of a poker agent that can remain grounded in equilibrium reasoning while adapting to exploitable behavioral patterns.

Existing poker agents can be broadly divided into three categories. The first category consists of rule-based and probability-based agents [Billings et al., 1998], which are computationally simple but strategically limited. The second category consists of equilibrium-oriented agents [Wang et al., 2022, Yuan et al., 2021a, Yuan et al., 2021b] based on Counterfactual Regret Minimization (CFR) or related reinforcement learning methods, which are theoretically principled but often insufficiently adaptive. The third category consists of large-language-model-based reasoning agents [Zhuang et al., 2025, Huang et al., 2024], which can display flexible contextual behavior but generally lack formal guarantees and may fail to compute precise probabilistic quantities such as hand equity. The research problem addressed in this thesis is therefore the following: how can one design a poker system that preserves the robustness of equilibrium-oriented play while extracting additional value from opponents who deviate from equilibrium?

2 Background

2.1 Heads-Up No-Limit Texas Hold'em

This thesis studies Heads-Up No-Limit Texas Hold'em (HUNL), a two-player zero-sum poker game played with a standard deck of 52 cards. At the beginning of each hand, the two players post mandatory blinds and each receives two private hole cards. Five community cards are then revealed in stages: three cards on the flop, one card on the turn, and one card on the river. Between these card-revelation stages, players participate in betting rounds in which they may fold, check, call, bet, or raise, depending on the current betting context. If neither player folds before the final betting round is completed, the hand proceeds to showdown, where the stronger five-card poker hand wins the pot. The hand rankings are shown in Table 10.

Several poker terms are used repeatedly throughout the thesis. The *stack* denotes the number of chips currently held by a player. The *pot* denotes the total number of chips contested in the current hand. The *board* refers to the set of public community cards that have already been revealed. A *showdown* occurs when the remaining players reveal their private cards after the final betting round. More behavior-oriented terms are also useful. A player is called *tight* if they enter relatively few hands, and *loose* if they enter many hands. A player is called *passive* if they tend to call rather than raise, and *aggressive* if they apply pressure through betting and raising. In principle, no-limit poker admits a continuous betting space because a player may choose any wager up to the size of the remaining stack. This makes the full game extremely large. In practice, one commonly introduces a discretized action space so that the game can be simulated, learned, and evaluated within a manageable computational budget. The experiments in this thesis follow this practice and use a discretized betting structure as Table 11 aligned

with the action abstraction used by *Slumbot*, a publicly available heads-up no-limit poker agent widely used as a benchmark for evaluating poker-playing systems [Jackson, 2017].

2.2 Game Representation and Notation

To describe the decision problem, we fix notation for the remainder of the thesis. Let h denote a history, that is, the ordered sequence of card realizations and betting actions that have occurred so far. Let z denote a terminal history. In poker, randomness arises from card dealing, so the game is naturally modeled as a finite extensive-form game with chance moves,

$$G = (N, H, P, A, \mathcal{I}, u)$$

where $N = \{1, 2, c\}$ consists of the two strategic players and a pseudo-player c, called chance. The chance player represents the randomness of card dealing and determines the private hole cards and subsequent public cards according to the deck distribution. H is the set of feasible histories, P is the player function that specifies who acts after each nonterminal history, A is the action correspondence, $\mathcal{I}$ is the collection of information sets, and $u_i(z)$ denotes player i's payoff at terminal history z. Because poker includes stochastic card dealing and sequential actions, outcomes depend on chance realizations and players' decisions. As a result, payoffs are defined over terminal histories z, which encode both the realized cards and the sequence of actions. Expected utility is then obtained by taking expectations over terminal histories induced by a strategy profile. At any decision point, player i does not observe the full underlying state of the game, because the opponent's private cards remain hidden. Instead, player i acts on the basis of an information set $I \in \mathcal{I}$, which consists of all histories that are indistinguishable from player i's perspective. Strategic decisions are therefore made over information sets rather than fully observed world states. When a policy is written in implemented state form, we use $\pi(a \mid s_i)$

to denote the probability of taking action a given player i's decision state s_i, which includes the public game situation together with player i's own private information. When a policy is written in the language of extensive-form games, we use $\sigma(a \mid I)$ to denote the probability of taking action a at information set I. The notation π is used mainly for implemented agents and learned decision modules and σ is used when the discussion is explicitly game-theoretic. Since the present setting is heads-up poker, the game is zero-sum, so the two players' terminal utilities sum to zero. This zero-sum structure makes the notions of equilibrium, exploitability, and best response especially central to both the theoretical and empirical analyses developed later.

2.3 Hand Equity and Expected Value

A central poker-specific quantity is *hand equity*, which measures the probability that a player's hand wins at showdown conditional on the currently available information. Let x denote the player's private cards and let b denote the current board. We write

$$E(x, b) = \Pr(\text{win at showdown} \mid x, b)$$

where the probability is taken with respect to the unknown opponent cards and any future unrevealed community cards. Depending on the context, ties may either be counted separately or incorporated by assigning half credit. Because exact equity computation is generally expensive in realistic game settings, a common practical approximation is Monte Carlo simulation [Ishikota, 2016]. If N simulated completions of the remaining cards are generated and $X_t \in \{0, \frac{1}{2}, 1\}$ records the outcome of simulation t, then the empirical equity estimate is

$$\widehat{E}(x, b) = \frac{1}{N} \sum_{t=1}^{N} X_t$$

2.4 Nash Equilibrium, Best Response, and Robustness

Let $\sigma = (\sigma_i, \sigma_{-i})$ denote a strategy profile, where σ_i is the strategy of player i and σ_{-i} is the opponent's strategy. Since poker includes stochastic card dealing, payoffs must be evaluated in expectation over both strategic randomization and chance outcomes. Accordingly, let

$$U_i(\sigma) = \mathbb{E}_{z\sim(\sigma,\pi_c)}[u_i(z)]$$

denote the expected utility of player i, where π_c is the fixed chance policy induced by the deck distribution. A Nash equilibrium is a strategy profile σ^* such that no player can improve their expected payoff by unilateral deviation. Formally, for each player i,

$$U_i(\sigma_i^*, \sigma_{-i}^*) \geq U_i(\sigma_i, \sigma_{-i}^*), \qquad \forall \sigma_i$$

In a two-player zero-sum game, an equilibrium strategy provides a natural benchmark for strategic robustness. This robustness interpretation follows from the minimax theorem, and a more formal discussion is provided in Section 4. Informally, if a player adopts an equilibrium strategy, the opponent cannot systematically exploit them beyond the value of the game. However, robustness does not imply payoff maximization against every specific opponent. When an opponent deviates from equilibrium in a predictable way, a best-response-style adjustment may achieve higher expected returns than strict equilibrium play. This distinction between equilibrium safety and exploitative opportunity motivates the hybrid framework developed in this thesis.

3 Literature Review

3.1 Counterfactual Regret Minimization

Libratus [Brown and Sandholm, 2018] and DeepStack [Meng et al., 2024] claimed victory over professional human players in two-player No-Limit Texas Hold'em settings. The key to these poker AIs was Counterfactual Regret Minimization (CFR), a family of iterative algorithms that use regret minimization to approximate an ϵ-Nash equilibrium in two-player zero-sum games. For player i and strategy σ, the Counterfactual Value (CFV) $v_i^\sigma(I)$ is defined as:

$$v_i^\sigma(I) = \sum_{h \in I} \left(\pi_{-i}^\sigma(h) \sum_{z \in \mathcal{Z}} (\pi^\sigma(h, z) u_i(z)) \right)$$

where I denotes the information set, $\pi_{-i}^\sigma(h)$ the reach probability of opponents at history h, and $u_i(z)$ the payoff of player i at terminal history z. CFV defines a utility over information sets that enables regret minimization at each I. The CFV of an action a is:

$$v_i^\sigma(I, a) = \sum_{h \in I} \left(\pi_{-i}^\sigma(h) \sum_{z \in \mathcal{Z}} (\pi^\sigma(h \cdot a, z) u_i(z)) \right)$$

The counterfactual regret of action a in I after T iterations is:

$$R_i^T(I, a) = \sum_{t=1}^{T} \left(v_i^{\sigma^t}(I, a) - v_i^{\sigma^t}(I) \right)$$

and the cumulative regret for player i is:

$$R_i^T(a|I_i) = \max_{\sigma_i' \in \Sigma_i} \sum_{t=1}^{T} \left(u_i(\sigma_i', \sigma_{-i}^t) - u_i(\sigma_i^t, \sigma_{-i}^t) \right)$$

At each iteration, strategies are updated according to the accumulated positive regrets using regret matching $\sigma_i^{T+1}(a|I) \propto R_i^{T,+}(I,a)$, and the average strategy is computed as:

$$\bar{\sigma}_i^T(a|I_i) = \frac{\sum_{t=1}^T \pi_i^{\sigma^t}(I_i)\sigma_i^t(a|I_i)}{\sum_{t=1}^T \pi_i^{\sigma^t}(I_i)}$$

CFR [Zinkevich et al., 2007a] repeatedly updates strategies using these counterfactual values. Each iteration computes new CFVs to minimize local regret independently at every information set. Current improvements to CFR include Monte Carlo sampling variants MCCFR [Lanctot et al., 2009] and advanced regret-matching algorithms such as CFR+ [Tammelin, 2014], which significantly accelerate convergence.

3.2 Opponent Modeling and Hybrid Decision Systems

While CFR-based agents converge toward Nash equilibrium strategies that are theoretically unexploitable, they typically do not adapt to opponent-specific behavior. Consequently, equilibrium agents may fail to exploit systematic deviations from optimal play. This limitation motivates opponent modeling techniques that infer behavioral patterns from observed actions and adjust strategies. Classical poker analysis often describes opponents using two behavioral dimensions: *tight versus loose* and *passive versus aggressive*, producing four common player archetypes which capture differences in hand selection and betting behavior [Billings et al., 1998]. Several studies attempt to integrate opponent information into equilibrium-style algorithms. For example, Opponent-Restricted CFR [Yuan et al., 2021b] conditions regret updates on opponent-type information, enabling the agent to exploit predictable behavioral patterns while preserving the structure of regret minimization. Other work [Frosterud and Sandholm, 2025] trains specialized agents corresponding to different opponent archetypes and switches among them based on observed play. A natural extension of this idea

is to combine multiple strategic components within a unified decision system. In a general ensemble formulation,

$$\pi_{\text{ens}}(a \mid s) = \sum_{m=1}^{M} w_m \pi_m(a \mid s), \qquad \sum_{m=1}^{M} w_m = 1, \quad w_m \geq 0$$

Although such an ensemble corresponds to a mixed strategy in game-theoretic terms, combining heterogeneous policies can improve practical robustness and allow adaptation to opponent behaviors. Ensemble-based poker systems have shown promising empirical results. The EnsembleCard framework [Yuan et al., 2022], for example, integrates a near-equilibrium CFR component, an exploitative policy, and heuristic modules within a unified architecture, combining their action distributions through weighted aggregation. These results suggest that hybrid systems integrating equilibrium reasoning with opponent-aware adaptation provide a practical direction for strong poker agents, a direction followed by the framework proposed in this thesis.

4 Theoretical Foundation: Equilibrium and Exploitation

4.1 Equilibrium as a Maximin Benchmark

In a two-player zero-sum game, the strategic role of a Nash equilibrium is best understood through the minimax theorem. Let $U_i(\sigma_i, \sigma_{-i})$ denote player i's expected utility under (σ_i, σ_{-i}). Since the game is zero-sum, $U_i(\sigma_i, \sigma_{-i}) = -U_{-i}(\sigma_i, \sigma_{-i})$. The value of the game is

$$v = \max_{\sigma_i} \min_{\sigma_{-i}} U_i(\sigma_i, \sigma_{-i}) = \min_{\sigma_{-i}} \max_{\sigma_i} U_i(\sigma_i, \sigma_{-i}),$$

and any equilibrium profile $\sigma^* = (\sigma_i^*, \sigma_{-i}^*)$ attains this value [von Neumann, 1928]. Thus,

$$\begin{cases} U_i(\sigma_i^*, \sigma_{-i}) \geq v, & \forall \sigma_{-i} \\ U_i(\sigma_i, \sigma_{-i}^*) \leq v, & \forall \sigma_i \end{cases}$$

Hence an equilibrium strategy is *maximin-optimal*: it protects the player against the worst-case opponent and guarantees at least the value of the game. This is the precise sense in which equilibrium play is robust. It is not merely a heuristic notion of "solid play" but a formal worst-case guarantee implied by zero-sum structure. In extensive-form imperfect-information games such as poker, the same logic applies to behavioral strategies over information sets. In particular, Counterfactual Regret Minimization (CFR) provides a route to approximate equilibrium: if regret is driven to zero, the average strategy converges to an ϵ-Nash equilibrium [Zinkevich et al., 2007b]. Therefore, when this thesis refers to a GTO-oriented component, it means not exact equilibrium computation in the full no-limit game, but an approximation to the equilibrium benchmark that inherits this robustness interpretation.

4.2 Why Equilibrium Is Not Payoff-Maximizing

The maximin property should not be confused with opponent-specific payoff maximization. Suppose now that the opponent does *not* play an equilibrium strategy, but instead follows some fixed strategy τ_{-i}. The optimization problem faced by player i becomes

$$\max_{\sigma_i} U_i(\sigma_i, \tau_{-i})$$

whose solution set is the set of best responses

$$BR_i(\tau_{-i}) = \arg\max_{\sigma_i} U_i(\sigma_i, \tau_{-i})$$

By definition, for any equilibrium strategy σ_i^*,

$$U_i(\beta_i, \tau_{-i}) \geq U_i(\sigma_i^*, \tau_{-i}) \qquad \forall \beta_i \in BR_i(\tau_{-i})$$

Moreover, the inequality is strict whenever $\sigma_i^* \notin BR_i(\tau_{-i})$. Therefore, equilibrium play is generally *not* pointwise optimal against a specific non-equilibrium opponent. This distinction is fundamental. Equilibrium solves the robust optimization problem

$$\max_{\sigma_i} \min_{\sigma_{-i}} U_i(\sigma_i, \sigma_{-i})$$

where the player guards against arbitrary adversarial behavior. By contrast, exploitation solves the conditional optimization problem

$$\max_{\sigma_i} U_i(\sigma_i, \tau_{-i})$$

where the opponent is treated as a particular object to be exploited rather than as an unrestricted adversary. These are different objective functions, and in general admit different optimizers. For this reason, it is more accurate to say that equilibrium play is *worst-case optimal*, not universally payoff-maximizing. This point has been stated explicitly in the computational game-theory literature. Ponsen, de Jong, and Lanctot observe that a Nash equilibrium is *not* a best response against players that are not themselves playing a Nash equilibrium [Ponsen et al., 2014]. In the same spirit, Johanson and Bowling define exploitability relative to the game value and distinguish

it from exploitation against a particular modeled opponent [Johanson and Bowling, 2009]. Thus, from both the game-theoretic and algorithmic perspectives, there is no contradiction in saying that equilibrium is the correct benchmark for safety while a non-equilibrium response may be strictly superior for profit extraction against a weak opponent.

4.3 Exploitability, Brittleness, and the Need for Robust Exploitation

The previous argument might suggest that one should always replace equilibrium play with a pure best response once an opponent appears weak. However, this conclusion is too aggressive when the opponent model is noisy or incomplete. Let $\hat{\tau}_{-i}$ denote an estimate of the opponent's strategy. A pure best response $BR_i(\hat{\tau}_{-i})$ maximizes payoff against the estimated model, but can perform poorly when the estimate is wrong. In this sense, best responses are often brittle.

The literature on robust counter-strategies formalizes precisely this problem. Johanson, Zinkevich, and Bowling show that robust counter-strategies should balance exploitation of suspected tendencies with protection against model error [Johanson et al., 2008]. Johanson and Bowling further argue that when an accurate model of the opponent is available, a best response is appropriate; when no useful model is available, Nash equilibrium is a sensible default; and when the model is imperfect, a compromise strategy is preferable, accepting some reduction in worst-case safety in exchange for higher utility if the model is approximately correct [Johanson and Bowling, 2009]. This is exactly the strategic setting of practical poker: opponents are often exploitable, but the exploitation signal is noisy, partial, and state-dependent.

Formally, if we define the exploitability of a strategy σ_i by

$$\text{Exploitability}(\sigma_i) = v - \min_{\sigma_{-i}} U_i(\sigma_i, \sigma_{-i})$$

then any equilibrium strategy has zero exploitability, while a pure best response to a misspecified

opponent model generally does not. The strategic problem is therefore not simply "equilibrium or best response," but rather how far one should deviate from equilibrium as a function of estimated opponent weakness and uncertainty.

4.4 Implication for the CEO-Poker Framework

The preceding analysis yields the theoretical logic of the proposed system. A GTO-oriented component is justified because it anchors play near the maximin benchmark and limits downside against strong or poorly understood opponents. An exploitative component is justified because, once the opponent exhibits systematic deviations from equilibrium, the conditional objective shifts from worst-case protection to opponent-specific payoff maximization. These two objectives coincide only in special cases; in general, they pull in different directions.

Accordingly, the central design problem is one of *adaptive interpolation*. When the opponent appears close to equilibrium, or when the evidence about their weakness is unreliable, the system should remain close to the equilibrium-oriented policy. When the opponent exhibits a stable and predictable deviation, the system should tilt away from equilibrium toward an exploitative response. The meta-controller introduced in Section 5 operationalizes exactly this trade-off by letting the weight on the exploitative policy increase with the estimated exploit tendency and the confidence of that estimate. In this sense, the CEO-Poker architecture is not an ad hoc ensemble. It is a practical implementation of a theoretically grounded principle: equilibrium for safety, deviation for exploitation, and interpolation under uncertainty.

5 Methodology

CEO-Poker is designed as a modular decision system composed of five components: an opponent analyzer, an equity calculator, a GTO-oriented player, an exploit player, and a meta

controller. The system aims to maintain equilibrium-style robustness when the opponent appears strong or uncertain, while shifting toward exploitative play when the opponent appears weak or predictable. Figure 1 illustrates the overall architecture.

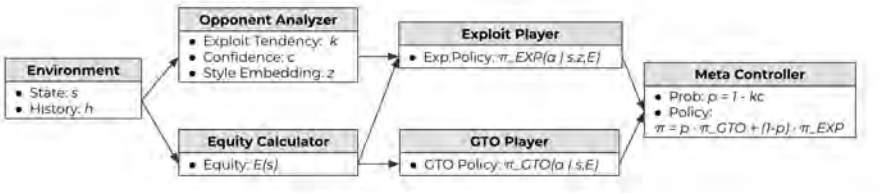

Figure 1: Overview of the poker agent decision system.

Environment and State Representation Let s denote the current game state and h denote the full action history available to the system. The state s includes public cards, private cards, pot size, stack sizes, betting context, legal actions, and player position. The history h consists of the ordered sequence of observed actions and public card revelations accumulated over play. Since the raw history can be high-dimensional and difficult to use directly, later modules extract lower-dimensional summaries from h to z for downstream decision-making.

Opponent Analyzer The opponent analyzer takes the full history h as input and produces

$$k \in [0, 1], \qquad c \in [0, 1], \qquad z \in \mathbb{R}^d$$

where k represents the estimated exploit tendency of the opponent, c is a confidence score for this estimate, and z is a style embedding that summarizes the opponent's behavioral characteristics. Intuitively, k measures the degree to which the opponent appears to deviate from equilibrium play, while c reflects the reliability of this estimate. The vector z serves as a compressed representation of the strategic information contained in h.

Equity Calculator The equity calculator estimates showdown equity using Monte Carlo simulation. Given hero hole cards x and board state b, the estimated equity is

$$\widehat{E}(x, b) = \frac{1}{N} \sum_{t=1}^{N} X_t$$

where $X_t \in \{0, \frac{1}{2}, 1\}$ records loss, tie, or win under random completion of the game. This estimate is used as a signal for both equilibrium-oriented and exploitative decision modules.

GTO-Oriented Player The GTO-oriented player outputs a policy

$$\pi_{\text{GTO}}(a \mid s, E)$$

whose objective is to approximate equilibrium-style play while remaining tractable. Depending on the setting, this component may be implemented using rule-based equilibrium heuristics, reinforcement learning agents, or prompt-based approximations.

Exploit Player The exploit player produces a policy

$$\pi_{\text{EXP}}(a \mid s, z, E)$$

which conditions on the game state, the estimated equity, and opponent-specific style information z. Its objective is to increase expected payoff against exploitable opponents by deviating from generic equilibrium strategies in targeted ways.

Meta Controller The meta controller combines the two policies through an adaptive weighting mechanism. In the specification used in this thesis,

$$p = 1 - kc$$

$$\pi(a \mid s) = p\,\pi_{\mathrm{GTO}}(a \mid s, E) + (1 - p)\,\pi_{\mathrm{EXP}}(a \mid s, z, E)$$

When both the estimated exploit tendency k and the confidence score c are large, the system shifts toward the exploitative policy. Otherwise, it remains closer to the equilibrium-oriented component, thereby maintaining robustness against strong or uncertain opponents.

6 Experimental Design

6.1 Engineering Infrastructure

The experimental system combines four components. First, Slumbot serves as the primary online benchmark because it provides a strong public heads-up no-limit Texas Hold'em agent and a fixed external evaluation environment [Jackson, 2017]. Second, RLCard serves as the reinforcement-learning interface for DQN and NFSP-style agents, but with a custom Slumbot-compatible no-limit Hold'em wrapper rather than the default environment [Zha et al., 2020]. Third, PyPokerEngine is used for local two-player simulations and for baseline-versus-baseline and LLM-versus-baseline matches [Ishikota, 2016]. Fourth, custom wrappers handle action translation, legality correction, equity estimation, logging, aggregation, and metric computation. The online and local pipelines share the same policy interface whenever possible. Each policy receives private hole cards, public board cards, a Slumbot-style action string, and seat position. Policies using equity information also receive Monte Carlo equity estimates, pot size, call cost,

pot odds, and legal raise bounds. The equity simulator uses 3,000 rollouts per estimate in the two-player setting. For reinforcement-learning agents, the custom RLCard environment exposes a vector observation containing private cards, public cards, street identity, player position, pot and stack features, call amount, legal raise bounds, and a bounded recent-action history. The same environment also exposes an exact information-state key for tabular and CFR-style methods. This allows comparison of value-based learning, fictitious self-play, regret minimization, and LLM-based policies under a unified rule system.

6.2 Baseline Agents

CEO-Poker is compared against several groups of baselines [1].

1. The first group consists of simple rule-based agents: Check-Fold, All-In, Always Call, Always Raise, and Random. These agents are intentionally weak or highly biased, but they serve as important diagnostic baselines. They reveal whether an evaluated agent can exploit obvious mistakes, survive extreme aggression, and avoid losing to trivial policies.

2. The second group consists of probability-based agents. The Equity Threshold agent estimates hand equity and chooses actions according to fixed equity cutoffs. The Equity-EV agent additionally compares equity to pot odds and call cost, approximating a simple expected-value decision rule. These baselines test whether CEO-Poker improves over direct card-strength heuristics rather than merely reproducing equity-threshold play.

3. The third group consists of reinforcement-learning and regret-minimization baselines. DQN provides a value-based deep RL baseline [Mnih et al., 2015]; NFSP provides a self-play method for imperfect-information games [Heinrich and Silver, 2016]; MCCFR

[1] The exact implemented decision rules for the non-learning baseline agents are provided in Appendix B.

and CFR+ provide regret-minimization baselines grounded in counterfactual regret minimization [Zinkevich et al., 2007a, Lanctot et al., 2009, Tammelin, 2014]; and Deep CFR provides a neural approximation of CFR-style regret learning [Brown et al., 2019]. These agents are trained in the custom Slumbot-compatible RLCard environment, then evaluated against the same local baselines and, when available, against online Slumbot.

4. The fourth group consists of LLM-based agents [2]. The single LLM baseline directly maps a poker state prompt to a legal Slumbot-style action. The exploitative single LLM variant, LLM-EO, augments the prompt with opponent statistics and equity features. CEO-Poker further extends this idea with a multi-agent exploitative organization in which opponent modeling and action recommendation are coordinated before producing the final move. This comparison isolates whether CEO-Poker's organizational structure improves over a single prompted LLM policy.

6.3 Evaluation Metrics

6.3.1 Profitability Metrics

Let w_t denote the hero's net chip winnings in hand t, and let T denote the number of hands. Final cumulative winnings are defined as

$$W_T = \sum_{t=1}^{T} w_t$$

Win rate is the fraction of hands with positive net winnings:

$$\text{WinRate} = \frac{1}{T} \sum_{t=1}^{T} \mathbf{1}[w_t > 0]$$

[2]The action-token format and the prompt template used for the LLM-based agents are provided in Appendix C.

Maximum drawdown measures the largest peak-to-trough loss in the cumulative winnings curve:

$$\text{MDD} = \max_{1 \le t \le T} \left(\max_{1 \le s \le t} W_s - W_t \right)$$

6.3.2 Strategic Behavior Metrics

VPIP measures how often the agent voluntarily puts chips into the pot:

$$\text{VPIP} = \frac{1}{T} \sum_{t=1}^{T} v_t$$

where $v_t = 1$ if the hero voluntarily calls or bets/raises in hand t, and $v_t = 0$ otherwise. PFR measures how often the hero raises preflop:

$$\text{PFR} = \frac{1}{T} \sum_{t=1}^{T} r_t$$

where $r_t = 1$ if the hero makes a preflop raise. Aggression factor is defined as

$$\text{AF} = \frac{\#\text{bets} + \#\text{raises}}{\#\text{calls}}$$

Showdown rate measures how often a hand reaches showdown:

$$\text{ShowdownRate} = \frac{1}{T} \sum_{t=1}^{T} s_t$$

where $s_t = 1$ if hand t reaches showdown, and $s_t = 0$ otherwise. This metric complements the previous three factors by showing whether an agent tends to end hands before showdown through folding or pressure, or instead continues to later streets and reveals cards more frequently.

6.4 Experimental Protocol

Unless otherwise specified, each local evaluation consists of 1,000 hands per opponent. Local matches alternate seating across hands to reduce positional bias. CEO-Poker and related LLM baselines are evaluated against the full local baseline suite, including All-In, Always Call, Always Raise, Check-Fold, Random, Equity, and Equity-EV. In addition to local evaluation, the main agents are tested against Slumbot over 1,000 hands. The local baseline suite is used for controlled diagnosis, while Slumbot serves as the strongest external benchmark. The reinforcement-learning baselines are trained in self-play in the custom Slumbot-compatible RLCard environment. DQN and NFSP are trained for 50,000 episodes. MCCFR and CFR+ are trained for 100 iterations with 20 traversals per iteration. Deep CFR is trained for 50 iterations with the same traversal budget. These settings reflect local computational constraints rather than an attempt to solve the full game. For LLM-based policies, the main experiments use Qwen 3.5 Plus with deterministic decoding. The model output is restricted to Slumbot-style legal actions before being passed to the environment. Equity-enabled variants additionally receive Monte Carlo equity, pot odds, call cost, and legal raise bounds, while exploitative variants also receive summarized opponent behavior from prior hands. All methods are evaluated under the same logging and metric pipeline. Because poker outcomes are high-variance, the analysis reports not only profit-based outcomes but also strategic-behavior and decision-quality metrics.

7 Results

7.1 Baseline Performance Against Slumbot

The first experiment evaluates simple rule-based agents, probability-based agents, and a single-agent LLM baseline against Slumbot. The purpose is to establish benchmark difficulty before introducing exploitative or multiple LLM reasoning. Table 1 shows that all baseline agents lose money over 1,000 hands. Even policies that win many hands, such as All-In and Always Raise, suffer large cumulative losses because their strategies are highly exploitable. The results show two important patterns. First, raw hand win rate is not sufficient to measure poker strength. All-In wins 95.8% of hands but loses 367,200 chips, because many wins are small while losing hands are extremely costly. Second, simple aggression is not equivalent to good exploitation. Always Raise has high VPIP and PFR but performs worst overall. These results justify using Slumbot as a strong external benchmark and motivate richer agents that can condition decisions on equity, opponent behavior, and action legality. The cumulative trajectories in Figure 2 further illustrate the differences in variance and drawdown across strategies.

Table 1: Baseline performance against Slumbot over 1,000 hands.

Agent	Final Winnings	Win Rate	VPIP	PFR
Check-Fold	−80,450	0.108	0.000	0.000
Equity	−100,950	0.372	0.493	0.219
Equity-EV	−111,050	0.807	0.900	0.900
LLM	−131,850	0.247	0.359	0.028
Always Call	−169,700	0.490	0.952	0.000
Random	−306,500	0.444	0.613	0.324
All-In	−367,200	0.958	1.000	1.000
Always Raise	−540,200	0.753	1.000	0.942

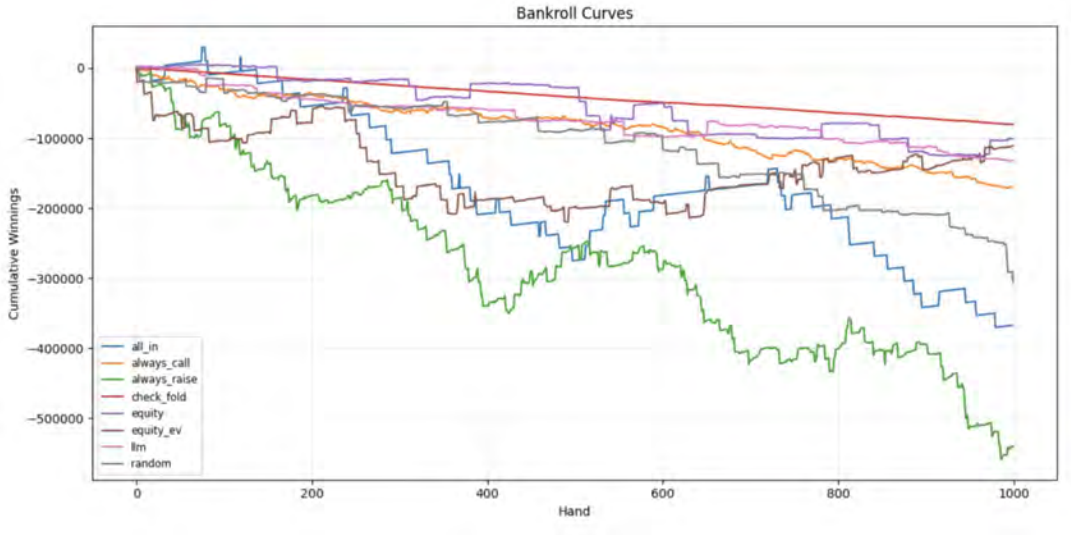

Figure 2: Cumulative bankroll trajectories of baseline agents against Slumbot over 1,000 hands.

7.2 Training Behavior of Equilibrium-Oriented RL Agents

The second set of experiments studies reinforcement-learning and regret-minimization agents trained in the custom Slumbot-compatible RLCard environment. These agents are not claimed to be fully game-theoretic optimal agents. Rather, they represent equilibrium-oriented learning attempts under the computational budget available for this thesis. Table 2 reports their direct performance against Slumbot. Every RL-based agent remains negative against Slumbot, indicating that local self-play and limited regret-minimization training do not produce a robust strategy under this abstraction and training budget.

Table 2: RL-based agents against Slumbot over 1,000 hands.

Agent	Training Scale	Final Winnings	Win Rate	Max Drawdown
DQN	50,000 episodes	−120,450	0.390	180,000
NFSP	50,000 episodes	−370,250	0.454	381,500
MCCFR	100 iterations	−284,650	0.573	307,750
CFR+	50–100 iterations	−297,900	0.559	332,050
Deep CFR	50 iterations	−339,000	0.477	401,850

The local post-training evaluations reveal why these agents remain fragile. DQN performs well against Random but loses heavily to Equity and Equity-EV. NFSP also beats Random and

Check-Fold but collapses against Always Raise, Equity, and Equity-EV. MCCFR and CFR+ show similar vulnerability to extreme aggression, especially Always Raise and Equity-EV. Deep CFR is more stable against All-In and Random, but still loses substantially to stronger probability-based opponents. Overall, these results suggest that limited local equilibrium-style training does not automatically yield practical robustness in no-limit poker.

7.3 Single LLM Baseline Results

The single LLM baseline performs substantially better against weak local opponents than against Slumbot. In local PyPokerEngine matches, the LLM wins against All-In, Always Call, Always Raise, Check-Fold, and Random, but loses heavily against Equity and Equity-EV as shown in Table 3. This pattern suggests that the LLM can exploit obvious strategic mistakes but struggles when the opponent follows consistent probability-based decision rules.

Table 3: Single-agent LLM performance against local baselines over 1,000 hands.

Opponent	Final Winnings	Win Rate
All-In	646,550	0.183
Always Call	702,650	0.574
Always Raise	275,950	0.231
Check-Fold	477,100	0.763
Equity	−930,450	0.647
Equity-EV	−282,600	0.465
Random	604,500	0.592

The negative result against Equity is very informative. Although the LLM wins 64.7% of hands, it loses 930,450 chips overall. This shows that poker performance depends on pot size and risk control more than hand win frequency. The single LLM has useful contextual reasoning, but without explicit numerical support it often mismanages bet sizing and downside risk.

7.4 Opponent-Aware Single LLM Results

LLM-EO augments the ordinary LLM prompt with historical opponent statistics. This improves performance against several opponents, especially All-In and Always Raise. LLM-EO wins 1,431,800 chips against All-In and 1,682,800 chips against Always Raise, much higher than the single LLM baseline. However, opponent-aware prompting is not uniformly beneficial. Compared with the single LLM baseline, LLM-EO earns less against Always Call and remains negative against Equity and Equity-EV. This indicates that historical statistics help identify exploitable tendencies, but a single LLM may overreact to those statistics or fail to balance exploitation with baseline play.

Table 4: Opponent-aware LLM-EO performance against local baselines.

Opponent	Final Winnings	Win Rate
All-In	1,431,800	0.636
Always Call	263,250	0.592
Always Raise	1,682,800	0.381
Check-Fold	539,700	0.885
Equity	−194,400	0.684
Equity-EV	−344,600	0.437
Random	751,850	0.615

7.5 Exploitative Multiple-LLM Results

CEO-Poker extends the exploitative idea by coordinating multiple LLM-based decision components through opponent modeling and meta-level control. The system separates the decision process into an opponent-modeling component, a safer non-exploitative action proposal, an exploitative action proposal, and a meta-controller. The opponent-modeling component outputs an exploit tendency k and confidence c, and the controller uses their product to arbitrate between the exploitative and safer recommendations. Table 5 reports the local results without explicit

equity augmentation. Under this setting, CEO-Poker achieves positive final winnings against every local baseline, although the gains against Equity and Equity-EV are close to break-even. Relative to the ordinary LLM baseline, it improves performance against All-In, Always Call, Always Raise, Equity, Equity-EV, and Random, while performing worse against Check-Fold. This suggests that coordinating multiple LLM-based components improves robustness across opponent types even without explicit numerical support, although the resulting gains remain uneven. Relative to LLM-EO, CEO-Poker is more balanced but not always more profitable. It performs much better against Always Call, but worse against All-In and Always Raise in the no-equity setting. This remains an important result: coordinating multiple LLM-based components is useful, but the gains against more disciplined opponents remain modest without accurate numerical evaluation of equity and pot odds.

Table 5: CEO-Poker performance against local baselines without explicit equity augmentation.

Opponent	Final Winnings	Win Rate
All-In	1,383,050	0.492
Always Call	800,000	0.617
Always Raise	881,650	0.303
Check-Fold	189,100	0.754
Equity	7,450	0.660
Equity-EV	6,550	0.477
Random	730,300	0.637

7.6 Ablation on the Equity Calculator

The equity calculator produces large aggregate improvements in local baseline experiments. Table 6 compares total final winnings across the seven local opponents before and after adding explicit equity information. The gain remains large for all three LLM-family agents. The result supports the hypothesis that LLM-based poker agents do not reliably compute equity internally. Once equity, pot odds, call cost, and legal raise bounds are supplied explicitly, the agents

become much better at converting contextual reasoning into profitable decisions. CEO-Poker with equity produces the highest total local profit among the tested LLM-family agents.

Table 6: Effect of explicit equity information across seven local baseline opponents.

Agent	Without Equity	With Equity	Improvement
LLM	1,493,700	8,717,849	+7,224,149
LLM-EO	4,130,400	11,348,300	+7,217,900
CEO-Poker	3,998,100	11,426,600	+7,428,500

7.7 Opponent-Aware Conditioning and Multiple-LLM Control

The opponent-aware results should be interpreted as a structural comparison rather than a pure ablation. The current implementation supports opponent statistics, an opponent-modeling output k, and a confidence output c. It does not contain a separate implemented variable z. Therefore, the empirical question is whether moving from ordinary prompting to opponent-aware prompting and then to multiple-LLM control improves performance. Table 7 summarizes this comparison using total local winnings with equity information enabled. Both LLM-EO and CEO-Poker outperform the ordinary LLM baseline. CEO-Poker is slightly stronger overall than LLM-EO, mainly because it improves against Always Call, Check-Fold, Equity, and Random.

Table 7: Structure comparison with equity information enabled.

Agent	Total Local Winnings	Best Opponent Result	Worst Opponent Result
LLM	8,717,849	2,465,550	21,050
LLM-EO	11,348,300	3,336,050	198,250
CEO-Poker	11,426,600	3,327,100	311,150

The worst-opponent result is especially important. With equity enabled, CEO-Poker remains positive against every local baseline, including Equity and Equity-EV. This reinforces the broader pattern that opponent-aware control and explicit equity information are complementary: equity improves numerical decision quality, while opponent-aware control helps shape exploitation.

7.8 CEO-Poker Versus Slumbot

The final comparison evaluates CEO-Poker and related LLM-based agents directly against Slumbot. Instead of reporting only final cumulative winnings, Figure 3 shows the full bankroll trajectories over 1,000 hands. All agents remain slightly negative over the full horizon, indicating that none fully defeats Slumbot under the current setup. However, the trajectories reveal an important structural difference relative to earlier baselines. The original LLM exhibits a clear downward trend, reflecting systematic loss accumulation. In contrast, several improved variants, including LLM-EO, CEO-Poker, and LLM-Equity, display more stable dynamics, with frequent recoveries and less persistent decline. This pattern suggests that the improved agents are no longer being consistently exploited in the same way as the baseline LLM. While they do not achieve sustained profitability, their performance is closer to a near-break-even regime, where gains and losses fluctuate rather than drift monotonically downward. At the same time, the results highlight a key limitation. Methods that significantly improve performance against weak and rule-based opponents do not automatically translate into superior performance against a strong near-equilibrium benchmark. In particular, aggressive exploitation or heavy reliance on numerical signals may introduce volatility that Slumbot can still punish.

7.9 CEO-Poker-Equity and Slumbot Against Imperfect Opponents

The direct match against Slumbot shows that CEO-Poker-Equity does not yet surpass a strong near-equilibrium benchmark. However, this comparison alone does not fully capture the intended objective of the proposed framework. The central motivation of CEO-Poker is not merely to approximate equilibrium play, but to convert persistent and observable opponent mistakes into additional profit. To evaluate this aspect directly, Table 8 compares Slumbot and CEO-Poker-

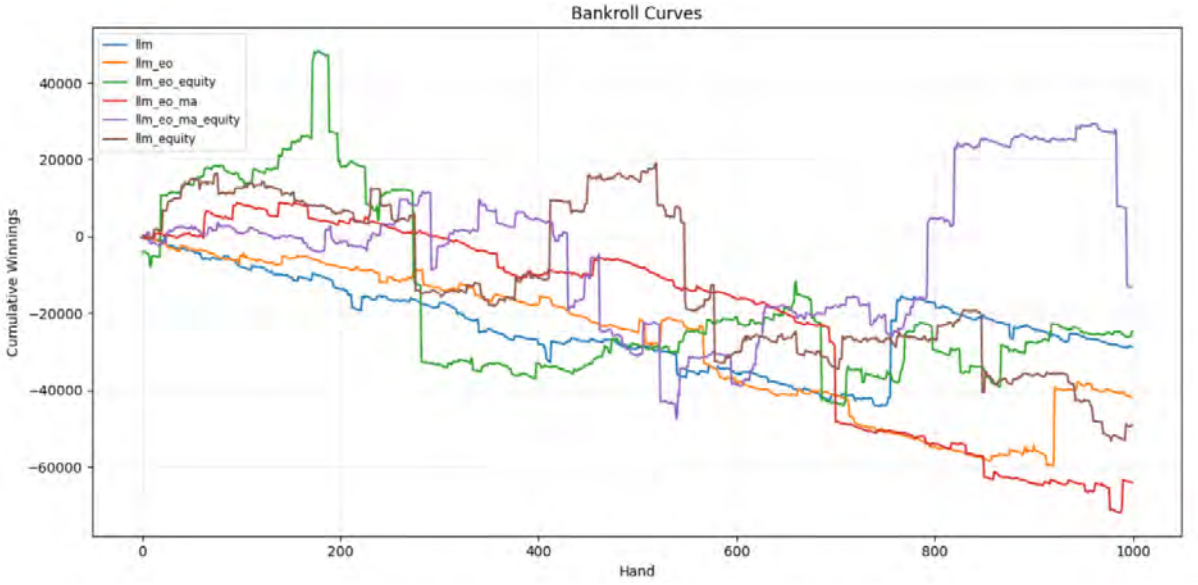

Figure 3: Bankroll curves of LLM-family agents against Slumbot over 1,000 hands.

Equity when each faces four deliberately distinctive non-equilibrium opponents: Check-Fold, Always Call, Always Raise, and All-In. These baselines represent qualitatively different deviations from strong play: Check-Fold is excessively passive, Always Call over-defends and under-raises, Always Raise over-applies aggression, and All-In collapses the betting game into an extreme one-shot commitment rule. Together, they form a compact stress test for exploitative capability. Table 8 shows that CEO-Poker-Equity earns substantially higher cumulative profit than Slumbot against all four imperfect opponents. Against Check-Fold, CEO-Poker-Equity wins nearly twenty times as many chips as Slumbot. Against Always Call, the difference becomes even more pronounced: CEO-Poker-Equity earns over 3.3 million chips, compared to roughly 170 thousand for Slumbot. Large gains also appear against Always Raise and All-In, although the profit multiples are smaller than those against the two passive baselines. These results support the central claim of the thesis: once the opponent deviates from strong play in a structured and persistent manner, CEO-Poker-Equity is able to extract substantially more value than a policy designed primarily for robustness. An especially informative observation is that higher profitability does not necessarily correspond to a higher hand-level win rate. Against

Always Call, Slumbot achieves a slightly higher win rate (0.510 versus 0.488), yet its cumulative profit is dramatically lower. This indicates that the advantage of CEO-Poker-Equity arises not from winning more hands, but from winning larger pots and controlling losses more effectively. In poker terms, the system monetizes opponent mistakes more efficiently.

Table 8: Common outcome metrics for Slumbot and CEO-Poker-Equity against four distinctive non-equilibrium opponents.

Agent	Metric	Check-Fold	Always Call	Always Raise	All-In
Slumbot	Final Winnings	80,450	169,700	540,200	367,200
	Mean/Hand	80.45	169.70	540.20	367.20
	Win Rate	0.892	0.510	0.247	0.042
	Max DD	300	12,600	103,300	131,900
CEO-Poker-Equity	Final Winnings	1,577,450	3,327,100	2,758,550	1,089,200
	Mean/Hand	1,577.45	3,327.10	2,758.55	1,089.20
	Win Rate	0.903	0.488	0.308	0.376
	Max DD	1,200	214,950	116,650	138,750

Table 9 provides additional insight into the behavioral mechanisms underlying these gains. CEO-Poker-Equity exhibits clear adaptation across opponent types. Against passive opponents such as Check-Fold and Always Call, it maintains high participation and relatively strong preflop aggression, consistent with an attempt to enlarge pots when resistance is weak. Against highly aggressive opponents such as Always Raise and All-In, it reduces both VPIP and PFR, indicating a more selective response that avoids unnecessary variance while exploiting over-commitment. By contrast, Slumbot's behavior remains comparatively stable across opponent types, reflecting a strategy optimized for robustness rather than targeted exploitation. While this leads to strong performance against near-equilibrium opponents, it limits the extent to which Slumbot can capitalize on systematic mistakes. The contrast suggests that CEO-Poker-Equity is not simply more aggressive overall, but more responsive in how aggression is deployed. Taken together, these results refine the main empirical conclusion of the thesis. CEO-Poker-Equity does not yet outperform Slumbot in direct head-to-head play, but it demonstrates a distinct advantage in

imperfect-opponent settings. When opponents deviate from equilibrium behavior, the system generates substantially higher profit and exhibits more targeted exploitative dynamics. This highlights the fundamental trade-off between equilibrium robustness and exploitative efficiency, and positions CEO-Poker-Equity as a complementary approach rather than a direct replacement for equilibrium-based strategies.

Table 9: Behavioral comparison of Slumbot and CEO-Poker-Equity against four distinctive non-equilibrium opponents.

Agent	Metric	Check-Fold	Always Call	Always Raise	All-In
Slumbot	VPIP	0.565	0.552	0.301	0.040
	PFR	0.347	0.248	0.155	0.021
	AF	1.792	1.041	0.665	0.491
	Showdown Rate	0.138	0.772	0.247	0.059
CEO-Poker-Equity	VPIP	0.813	0.924	0.610	0.573
	PFR	0.420	0.422	0.331	0.257
	AF	1.941	2.569	1.337	0.633
	Showdown Rate	0.211	0.843	0.447	0.329

8 Discussion

8.1 Interpretation Through Game Theory

The central theoretical claim of this thesis is that equilibrium-oriented play and exploitative play solve different optimization problems. An equilibrium strategy provides a worst-case guarantee, whereas an exploitative strategy maximizes payoff against a specific opponent. The empirical results are consistent with this distinction. Against weak and highly biased opponents such as All-In, Always Raise and Check-Fold, exploitative variants generate large positive winnings. Against Slumbot, however, all agents remain negative, and more aggressive strategies do not consistently improve outcomes. This contrast is expected. When the opponent is strongly exploitable, deviation from equilibrium increases payoff. When the opponent is near

equilibrium, the scope for profitable deviation becomes limited, and overconfident exploitation is punished. CEO-Poker should therefore be understood not as a replacement for equilibrium solving, but as a practical system for environments in which opponents are rarely fully optimal.

8.2 Why Equity Matters

The clearest empirical finding is the importance of explicit equity information. Across all LLM-based agents, adding equity produces large and consistent improvements against local baselines. This indicates that LLMs do not reliably perform the numerical evaluation required in poker, where decisions depend on comparing hand strength, pot odds, and risk. Equity and language-based reasoning play complementary roles. The LLM captures contextual patterns and opponent behavior, while the equity module provides disciplined numerical grounding. The strongest local performance arises when both are present. However, the Slumbot results show that equity alone is not sufficient for strong play against near-equilibrium opponents. While it improves exploitation, it may also introduce strategic overconfidence in marginal situations where conservative baseline play would be safer.

8.3 Why Opponent Modeling Matters

Opponent-aware conditioning is also important for improvement. Relative to the baseline LLM, incorporating opponent statistics improves performance against several biased strategies, indicating that even coarse behavioral summaries contain usable signal. In practice, repeated actions reveal stable tendencies that can be exploited through targeted deviations. However, opponent modeling alone is not sufficient. Without equity support, it can lead to overreaction or unstable behavior against more disciplined opponents. The variables k and c used in CEO-Poker are therefore best interpreted as heuristic indicators of opponent weakness and confidence

rather than formal measures of exploitability. The results suggest that opponent modeling is most effective when combined with numerical evaluation and controlled policy adjustment.

8.4 Robustness Versus Exploitation

A consistent pattern in the results is the trade-off between exploitative upside and robustness. The opponent-aware LLM variant achieves strong gains against highly biased opponents but performs unevenly across settings. CEO-Poker, by contrast, is more balanced: it sacrifices some extreme gains while improving performance against more ambiguous opponents such as Equity and Equity-EV. This behavior reflects the role of the meta-controller. Rather than always pursuing the strongest apparent exploit, the system adjusts its strategy based on estimated opponent weakness and confidence. The Slumbot experiments illustrate why this balance is necessary. Methods that perform best against weak opponents do not necessarily perform best against stronger ones. Exploitation is therefore beneficial only when the signal is reliable; otherwise, equilibrium-style caution becomes valuable again.

8.5 Limitations

Several limitations should be acknowledged. First, the current GTO-oriented component is not a true equilibrium solver. The RL and CFR-style baselines trained under the available computational budget remain far from equilibrium-quality play, so the "safe" component should be interpreted as relatively conservative rather than formally optimal. Second, the opponent model is approximate. The system uses summarized behavioral features and heuristic signals, but it does not infer a full opponent strategy in the extensive-form sense. This limits both theoretical precision and interpretability. Third, the experiments rely on action abstraction and finite-sample evaluation. The discretized betting structure simplifies computation but alters

the full game, and 1,000-hand matches still contain significant variance. Finally, the LLM components remain difficult to analyze mechanistically. While they often produce effective decisions, their internal reasoning is not directly observable, making it difficult to separate robust strategy from prompt sensitivity.

8.6 Future Work

Several directions follow from these limitations. First, strengthening the equilibrium-oriented backbone with more advanced CFR or RL methods would improve the safety side of the system. Second, replacing the heuristic opponent analyzer with a learned or Bayesian model could provide better-calibrated estimates of opponent behavior and uncertainty. Third, richer representations of opponent dynamics may capture more nuanced exploitability patterns, including temporal and context-dependent behavior. Fourth, the meta-controller could be extended beyond the current linear rule to a learned or optimization-based policy arbitration mechanism. Finally, future evaluation should include longer matches and stronger benchmarks to better distinguish variance from structural performance differences and to assess whether exploitative gains generalize beyond the current experimental setting.

9 Conclusion

This thesis studies how to balance equilibrium-style robustness with exploitative profitability in heads-up no-limit Texas Hold'em. Equilibrium-oriented play provides worst-case safety, but it is not necessarily payoff-maximizing against weak or systematically biased opponents. CEO-Poker addresses this gap through a modular decision framework that combines opponent analysis, explicit equity estimation, a GTO-oriented component, an exploitative component, and a meta-controller for adaptive policy selection. The empirical results support three main

findings. First, explicit equity information is crucial for LLM-based poker agents. Across local baseline experiments, adding equity, pot odds, call cost, and legal raise bounds produces large improvements, showing that contextual reasoning alone is insufficient in a game that requires disciplined probabilistic evaluation. Second, opponent-aware conditioning improves the ability to exploit structured behavioral mistakes. Historical opponent statistics and meta-level control help the agent move beyond generic play and adjust its decisions to different opponent types. Third, the strongest results arise when numerical equity evaluation and opponent-aware adaptation are combined. CEO-Poker-Equity remains profitable against every local baseline and achieves the highest total local winnings among the tested LLM-family agents. The comparison with Slumbot further clarifies the contribution of the framework. CEO-Poker-Equity does not yet surpass Slumbot in direct head-to-head play, which suggests that it should not be viewed as a replacement for strong equilibrium-based systems. However, against deliberately imperfect opponents, CEO-Poker-Equity substantially outperforms Slumbot in cumulative profit. This indicates that the proposed framework is better at converting persistent and observable opponent mistakes into value. The central contribution of this thesis is therefore not a complete solution to heads-up no-limit poker, but a theoretically grounded and empirically validated step toward adaptive decision-making in imperfect-information games. More broadly, the results suggest that practical intelligence in strategic environments requires both protection against exploitation and the ability to recognize when profitable deviation from robust play is justified.

References

[Billings et al., 1998] Billings, D., Papp, D., Schaeffer, J., and Szafron, D. (1998). Opponent modeling in poker. In *Proceedings of the 15th National Conference on Artificial Intelligence (AAAI-98)*, pages 493–498. AAAI Press.

[Brown et al., 2019] Brown, N., Lerer, A., Gross, S., and Sandholm, T. (2019). Deep counterfactual regret minimization. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 793–802. PMLR.

[Brown and Sandholm, 2018] Brown, N. and Sandholm, T. (2018). Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. *Science*, 359(6374):418–424.

[Frosterud and Sandholm, 2025] Frosterud, A. and Sandholm, S. (2025). Ai and game theory: Strategic decision making in imperfect information environments – evaluating ai-driven poker bots against common playstyles. Stockholm, Sweden. Degree Project in Technology, First Cycle, 15 credits.

[Heinrich and Silver, 2016] Heinrich, J. and Silver, D. (2016). Deep reinforcement learning from self-play in imperfect-information games. *arXiv preprint arXiv:1603.01121*.

[Huang et al., 2024] Huang, C., Cao, Y., Wen, Y., Zhou, T., and Zhang, Y. (2024). Pokergpt: An end-to-end lightweight solver for multi-player texas hold'em via large language model. *arXiv preprint arXiv:2401.06781*.

[Ishikota, 2016] Ishikota, I. (2016). Pypokerengine: A toolkit for building poker ai. `https://github.com/ishikota/PyPokerEngine`.

[Jackson, 2017] Jackson, E. (2017). Slumbot 2017. `https://www.slumbot.com/`.

[Johanson and Bowling, 2009] Johanson, M. and Bowling, M. (2009). Data biased robust counter strategies. In *Proceedings of the Twelfth International Conference on Artificial Intelligence and Statistics*, volume 5 of *Proceedings of Machine Learning Research*, pages 264–271.

[Johanson et al., 2008] Johanson, M., Zinkevich, M., and Bowling, M. (2008). Computing robust counter-strategies. In *Advances in Neural Information Processing Systems 20*, pages 721–728.

[Lanctot et al., 2009] Lanctot, M., Waugh, K., Zinkevich, M., and Bowling, M. (2009). Monte carlo sampling for regret minimization in extensive games. In *Advances in Neural Information Processing Systems (NeurIPS 2009)*, pages 1078–1086.

[Meng et al., 2024] Meng, L., Yang, J., Tian, R., Dai, X., Wu, Z., Gao, J., and Jiang, Y.-G. (2024). Deepstack: Deeply stacking visual tokens is surprisingly simple and effective for lmms.

[Mnih et al., 2015] Mnih, V., Kavukcuoglu, K., Silver, D., et al. (2015). Human-level control through deep reinforcement learning. *Nature*, 518(7540):529–533.

[Ponsen et al., 2014] Ponsen, M., de Jong, S., and Lanctot, M. (2014). Computing approximate nash equilibria and robust best-responses using sampling. *arXiv preprint arXiv:1401.4591*.

[Silver et al., 2016] Silver, D. et al. (2016). Mastering the game of go with deep neural networks and tree search. *Nature*, 529(7587):484–489.

[Tammelin, 2014] Tammelin, O. (2014). Solving large imperfect information games using cfr+. In *Proceedings of the 22nd International Conference on Automated Planning and Scheduling (ICAPS)*, pages 602–610.

[von Neumann, 1928] von Neumann, J. (1928). Zur theorie der gesellschaftsspiele. *Mathematische Annalen*, 100(1):295–320.

[Wang et al., 2022] Wang, K., Bai, D., and Zhou, Q. (2022). Deepholdem: An efficient end-to-end texas hold'em artificial intelligence fusion of algorithmic game theory and game information. *Unknown Journal/Conference*. Preliminary version — full publication details to be confirmed.

[Yuan et al., 2021a] Yuan, W., Hu, Z., Luo, J., . . . , and Chen, J. (2021a). Imperfect information game in multiplayer no-limit texas hold'em based on mean approximation and deep cfvnet.

[Yuan et al., 2021b] Yuan, W., Hu, Z., Wei, T., and Luo, J. (2021b). Opponent-restricted response solving on texas hold'em poker. In *Proceedings of the 2021 China Automation Congress (CAC 2021)*. IEEE.

[Yuan et al., 2022] Yuan, W., Wei, T., Luo, J., Lu, L., Zhang, W., and Chen, J. (2022). Ensemblecard: A strategy ensemble bot for two-player no-limit texas hold'em poker. Preprint version – full publication details to be verified.

[Zha et al., 2020] Zha, D., Lai, K.-H., Huang, S., Cao, Y., Reddy, K., Vargas, J., Nguyen, A., Wei, R., Guo, J., and Hu, X. (2020). Rlcard: A platform for reinforcement learning in card games. In *Proceedings of the Twenty-Ninth International Joint Conference on Artificial Intelligence (IJCAI)*, pages 5264–5266.

[Zhuang et al., 2025] Zhuang, R., Gupta, A., Yang, R., Rahane, A., Li, Z., and Anumanchipalli, G. (2025). Pokerbench: Training large language models to become professional poker players. *arXiv preprint arXiv:2501.08328*.

[Zinkevich et al., 2007a] Zinkevich, M., Johanson, M., Bowling, M., and Piccione, C. (2007a). Regret minimization in games with incomplete information.

[Zinkevich et al., 2007b] Zinkevich, M., Johanson, M., Bowling, M., and Piccione, C. (2007b). Regret minimization in games with incomplete information. In *Advances in Neural Information Processing Systems 20*, pages 1729–1736.

A Poker Rules and Bet-Size Discretization Rules

Table 10: Poker Hand Rankings from Highest to Lowest

Hand Ranking	**Description**
Royal Flush	A, K, Q, J, 10 of the same suit
Straight Flush	Five consecutive cards of the same suit
Four of a Kind	Four cards of the same rank
Full House	Three of a kind + a pair
Flush	Five cards of the same suit (not consecutive)
Straight	Five consecutive cards of different suits
Three of a Kind	Three cards of the same rank
Two Pair	Two different pairs
One Pair	Two cards of the same rank
High Card	None of the above; highest card plays

Table 11: Slumbot Bet-Size Discretization Rules

Action Type	**Allowed Sizes**
Initial bets	0.25, 0.5, 0.75, 1.0, 1.5, 2.0, 4.0, 8.0, 15.0, 25.0, 50.0
Raises	0.5, 1.0, 2.0, 4.0, 8.0, 15.0, 25.0, 50.0
Three-bets	0.5, 1.0, 2.0
Four-bets and beyond	Pot-sized bet only
All-in	Always permitted

B Baseline Agent Rules

Table 12 summarizes the implemented decision rules for the non-learning baselines used in the experiments. In all cases, the returned action is subsequently interpreted under the same Slumbot betting semantics used throughout the evaluation code, where `k` denotes check, `c` denotes call, `f` denotes fold, and `bXXX` denotes a bet or raise to street-level commitment `XXX`.

Table 12: Implemented baseline-agent decision rules.

Agent	Decision Rule
Check-Fold	Check when no bet is outstanding. If facing only the forced small-blind completion (50 chips), call; otherwise fold.
All-In	Always return `b20000`, i.e., shove the full fixed stack of 20,000 chips.
Always Call	Call whenever facing a bet; otherwise check.
Always Raise	Attempt a raise on every decision. Candidate raise-to amounts are generated from the Slumbot bet-size abstraction for the current raise depth and then legalized against the minimum-raise and stack constraints; if no legal raise remains, fall back to call when facing a bet and to check otherwise.
Random	If no bet is outstanding, sample uniformly from {`k`, raise}; if facing a bet, sample uniformly from {`c`, `f`, raise}. For the raise branch, shuffle the abstraction-derived raise targets and execute the first target that survives legalization; otherwise fall back to check or call.
Equity Threshold	Estimate heads-up showdown equity by Monte Carlo simulation (`nb_simulation`= 5000) and compare it with a fixed threshold of 0.5. If equity is below threshold, check when free and fold when facing a bet. If equity is at least threshold and no bet is outstanding, check. If equity is at least threshold and facing a bet, call with probability 0.6; otherwise attempt the largest legal abstraction-derived raise, with fallback to call if no legal raise is available. This rule does not use pot odds or call-cost calculations.
Equity-EV	Estimate heads-up showdown equity by Monte Carlo simulation (`nb_simulation`= 5000), enumerate legal fold/check-call/raise actions, assign each action a heuristic expected value, and choose the action with maximal value. Fold is assigned value 0. If no bet is outstanding, check is valued as eq $\times$ total_to. If facing a bet, call uses call cost `last_bet_size` and is valued as eq $\times$ total_to $-$ (1 $-$ eq) $\times$ call_cost. Each candidate raise is first legalized; for a legal raise-to amount `final_rt`, the additional cost is `final_rt-total_to` and the raise value is eq $\times$ (total_to + cost) $-$ (1 $-$ eq) $\times$ cost. Pot odds are not used in this baseline.

C LLM Prompt

```
You are a poker bot playing heads-up No-Limit Texas Hold'em.

CHOOSE EXACTLY ONE ACTION TOKEN:
"k" = check
"c" = call
"f" = fold
"bXXX" = bet or raise to EXACT chips committed on this street XXX.

RULES:
- Output ONLY the action token, no explanation.
- If no bet exists (last_bettor == -1), allowed = "k" or "bXXX".
- If facing a bet, allowed = "c", "f", or "bXXX".
- Raise amount XXX is a street-level number, not total invested across all streets.
- Current street bet-to is <street_last_bet_to>;
  total invested so far is <total_last_bet_to>.

GAME STATE:
Hole cards: <hole_cards>
Board cards: <board>
Action history: "<action_str>"
Last bet size: <last_bet_size>
Total bet-to: <total_last_bet_to>
Position: <client_pos>

Return ONLY:  k  /  c  /  f  /  bXXX
```