Learning Strategic Poker Decision-Making with Large Language Models

Lucas de Oliveira Diniz

Universidade Federal de Uberlândia
Faculdade de Computação
Programa de Pós-Graduação em Ciência da Computação

Uberlândia
2026

Lucas de Oliveira Diniz

Learning Strategic Poker Decision-Making with Large Language Models

Dissertation submitted to the Graduate Program in Computer Science of the Faculty of Computing at the Federal University of Uberlândia as partial fulfillment of the requirements for the degree of Master in Computer Science.

Concentration area: Computer Science

Advisor: Prof. Dr. Murillo Guimarães Carneiro

Uberlândia
2026

Ficha Catalográfica Online do Sistema de Bibliotecas da UFU
com dados informados pelo(a) próprio(a) autor(a).

D585 Diniz, Lucas de Oliveira, 1995-
2026 Learning Strategic Poker Decision-Making with Large Language Models [recurso eletrônico] / Lucas de Oliveira Diniz. - 2026.

Orientador: Murillo Guimarães Carneiro.
Dissertação (Mestrado) - Universidade Federal de Uberlândia, Pós-graduação em Ciência da Computação.
Modo de acesso: Internet.
DOI http://doi.org/10.14393/ufu.di.2026.510
Inclui bibliografia.
Inclui ilustrações.

1. Computação. I. Carneiro, Murillo Guimarães,1988-, (Orient.). II. Universidade Federal de Uberlândia. Pós-graduação em Ciência da Computação. III. Título.

CDU: 681.3

Bibliotecários responsáveis pela estrutura de acordo com o AACR2:
Gizele Cristine Nunes do Couto - CRB6/2091
Nelson Marcos Ferreira - CRB6/3074

UNIVERSIDADE FEDERAL DE UBERLÂNDIA
Coordenação do Programa de Pós-Graduação em Computação
Av. João Naves de Ávila, 2121, Bloco 1A, Sala 243 - Bairro Santa Mônica, Uberlândia-MG, CEP 38400-902
Telefone: (34) 3239-4470 - www.ppgco.facom.ufu.br - cpgfacom@ufu.br

ATA DE DEFESA - PÓS-GRADUAÇÃO

Programa de Pós-Graduação em:	Ciência da Computação				
Defesa de:	Dissertação, 16/2026, PPGCO				
Data:	08 de Julho de 2026	Hora de início:	10:00	Hora de encerramento:	12:45
Matrícula do Discente:	12422CCP004				
Nome do Discente:	Lucas de Oliveira Diniz				
Título do Trabalho:	Learning Strategic Poker Decision-Making with Large Language Model				
Área de concentração:	Ciência da Computação				
Linha de pesquisa:	Inteligência Artificial				
Projeto de Pesquisa de vinculação:	----------				

Reuniu-se por videoconferência, a Banca Examinadora, designada pelo Colegiado do Programa de Pós-graduação em Ciência da Computação, assim composta: Professores Doutores: Rita Maria da Silva Julia - FACOM/UFU, Lucas Nascimento Ferreira - DCC/ICEx/UFMG e Murillo Guimarães Carneiro - FACOM/UFU, orientador do(a) candidato(a).

Os examinadores participaram desde as seguintes localidades: Lucas Nascimento Ferreira - Belo Horizonte/MG. Os outros membros da banca e o aluno(a) participaram da cidade de Uberlândia.

Iniciando os trabalhos o(a) presidente da mesa, Prof. Dr. Murillo Guimarães Carneiro, apresentou a Comissão Examinadora e o(a) candidato(a), agradeceu a presença do público, e concedeu ao(à) Discente a palavra para a exposição do seu trabalho.

A seguir o senhor presidente concedeu a palavra, pela ordem sucessivamente, aos examinadores, que passaram a arguir o(a) candidato(a). Ultimada a arguição, que se desenvolveu dentro dos termos regimentais, a Banca, em sessão secreta, atribuiu o resultado final, considerando o candidato(a):

Aprovado

Esta defesa faz parte dos requisitos necessários à obtenção do título de Mestre.

O competente diploma será expedido após cumprimento dos demais requisitos, conforme as normas do Programa, a legislação pertinente e a regulamentação interna da UFU.

Nada mais havendo a tratar foram encerrados os trabalhos. Foi lavrada a presente ata que após lida e achada conforme foi assinada pela Banca Examinadora.

Documento assinado eletronicamente por **Lucas Nascimento Ferreira**, **Usuário Externo**, em 13/07/2026, às 10:27, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Rita Maria da Silva Julia**, **Professor(a) do Magistério Superior**, em 14/07/2026, às 22:11, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

Documento assinado eletronicamente por **Murillo Guimarães Carneiro**, **Professor(a) do Magistério Superior**, em 03/08/2026, às 20:52, conforme horário oficial de Brasília, com fundamento no art. 6º, § 1º, do Decreto nº 8.539, de 8 de outubro de 2015.

A autenticidade deste documento pode ser conferida no site https://www.sei.ufu.br/sei/controlador_externo.php?acao=documento_conferir&id_orgao_acesso_externo=0, informando o código verificador **7455781** e o código CRC **02984B77**.

To those who cultivate the virtue of curiosity

Acknowledgements

I am grateful to my advisor, Prof. Murillo Carneiro, for embracing and supporting the development of a research direction that is unconventional, yet deeply connected to my academic and professional interests. His openness in engaging with a project at the intersection of artificial intelligence, games, and poker was fundamental to the completion of this dissertation. This work also reflects the importance of pursuing research questions that emerge from one's own trajectory, experience, and intellectual curiosity.

Special thanks are due to Richard Zhuang, author of PokerBench, for making the dataset available and for his availability and generosity in helping me better understand aspects of its construction. His clarifications contributed to the development of this research.

I also thank my friend Guilherme for his support, technical discussions, and programming-related assistance throughout this work.

The financial support provided by the Coordenação de Aperfeiçoamento de Pessoal de Nível Superior – Brasil (CAPES) during my master's studies is gratefully acknowledged.

Finally, I thank the members of the examination committee for their availability, time, and careful evaluation of this dissertation.

"The most exciting phrase in science is not 'Eureka!' but 'That's funny!'"

Isaac Asimov

Resumo

A tomada de decisão no pôquer No-Limit Texas Hold'em envolve a integração de informações categóricas, numéricas e sequenciais sob incerteza. Essas informações podem ser representadas como texto estruturado, permitindo avaliar Large Language Models (LLMs) como preditores de decisões estratégicas. Esta dissertação examina essa possibilidade em No-Limit Texas Hold'em 6-max, modalidade com informação imperfeita na qual cada decisão envolve uma ação e, em casos agressivos, um tamanho de aposta ou aumento. A tarefa é formulada como predição supervisionada sobre estados textualizados, separando os componentes categórico e numérico. Desenvolve-se um pipeline híbrido que seleciona ações por log-probabilidades e obtém tamanhos por geração determinística e parsing numérico, permitindo analisar separadamente a escolha da ação e sua parametrização numérica. A abordagem utiliza exemplos do PokerBench, avaliados em preflop e postflop. Seis LLMs abertos são comparados em few-shot prompting, e o modelo de melhor desempenho, Qwen3-14B, é especializado por fine-tuning supervisionado com LoRA e QLoRA em cada fase. Além da Acurácia de Ação, desenvolve-se uma formulação contínua da Action-and-Sizing Accuracy (Ac-s), que preserva a prioridade da ação e atribui crédito proporcional ao tamanho previsto quando a ação agressiva está correta. Few-shot prompting fornece uma linha de base, mas fica abaixo da adaptação supervisionada. Após o fine-tuning, o Qwen3-14B alcança 93,3% de acurácia de ação no preflop e 91,8% no postflop, com ganhos de 17,2 e 40,3 pontos percentuais, respectivamente. As análises mostram melhorias consistentes entre ações e contextos, enquanto os diagnósticos condicionais indicam elevada concordância proporcional dos sizings em ambos os estágios após o acerto da ação agressiva. Assim, a dissertação contribui com uma formulação operacional, uma métrica contínua, um pipeline reproduzível e uma análise de LLMs abertos para decisão estratégica estruturada em um domínio de informação imperfeita.

Palavras-chave: Modelos de Linguagem. Pôquer. No-Limit Texas Hold'em. Fine-Tuning Supervisionado. Decisão Estratégica.

Abstract

Decision-making in the poker variant No-Limit Texas Hold'em involves integrating categorical, numerical, and sequential information in an imperfect-information environment. These elements can be represented as structured text, enabling the investigation of Large Language Models (LLMs) as predictors of strategic decisions. This dissertation examines that possibility in 6-max No-Limit Texas Hold'em, an imperfect-information setting in which each decision involves an action and, in aggressive cases, a bet or raise sizing. The task is formulated as supervised prediction over textualized states, separating its categorical and numerical components. A hybrid pipeline selects actions through log-probability scoring and obtains sizings through deterministic generation followed by numerical parsing, enabling separate analysis of action selection and numerical parameterization. PokerBench examples are evaluated in preflop and postflop settings. Six open LLMs are compared under few-shot prompting, and the best-performing model, Qwen3-14B, is specialized through supervised fine-tuning with LoRA and QLoRA for each game stage. In addition to Action Accuracy, a continuous formulation of Action-and-Sizing Accuracy (Ac-s) is developed to preserve action-first semantics while assigning proportional credit to sizing predictions when the aggressive action is correct. Few-shot prompting provides an informative baseline but remains below supervised adaptation. After fine-tuning, Qwen3-14B achieves 93.3% Action Accuracy preflop and 91.8% postflop, corresponding to gains of 17.2 and 40.3 percentage points, respectively. Analyses show consistent improvements across actions and contexts, while conditional diagnostics indicate high proportional sizing agreement in both stages after the aggressive action is correctly identified. The dissertation therefore contributes an operational formulation, a continuous metric, a reproducible evaluation pipeline, and an empirical analysis of open LLMs for structured strategic decision-making in an imperfect-information domain.

Keywords: Language Models. Poker. No-Limit Texas Hold'em. Supervised Fine-Tuning. Strategic Decision-Making.

List of Figures

List of Tables

List of Acronyms

Ac	Action Accuracy
Ac-s	Action-and-Sizing Accuracy
AIVAT	Agent Independent Variance Assessment Tool
BB	Big Blind
CFR	Counterfactual Regret Minimization
CI	Confidence Interval
EV	Expected Value
GTO	Game Theory Optimal
LLM	Large Language Model
LoRA	Low-Rank Adaptation
MLP	Multilayer Perceptron
NF4	NormalFloat 4-bit
NLHE	No-Limit Texas Hold'em
OR	Odds Ratio
QLoRA	Quantized Low-Rank Adaptation
SB	Small Blind
SFT	Supervised Fine-Tuning

Contents

Chapter 1

Introduction

Poker is a competitive card game in which players make repeated decisions under uncertainty. Although the cards distributed in each hand involve chance, long-term performance also depends on strategic choices such as whether to continue in the hand, how much to bet, and how to respond to the actions of other players (Potter van Loon et al., 2015). The game is practiced in both live and online environments and is internationally recognized as a mind sport (IMSA, 2024).

Its relevance to the study of strategic interaction extends back to early game-theoretic research. Simplified poker games were used to examine mixed strategies, hidden information, and equilibrium behavior (KUHN, 1950). With the development of artificial intelligence, increasingly complex poker variants became benchmarks for studying how computational agents can act when relevant information is unavailable and opponents may adapt their strategies (BILLINGS et al., 2003; SANDHOLM, 2015).

Among the different poker variants, Texas Hold'em is a community-card game in which each player receives two private cards and may combine them with five community cards to form the best possible five-card hand (CBGC, 2016). In the no-limit betting structure, a player may bet or raise any permitted amount up to the chips available in their stack (JOHANSON, 2013). No-Limit Texas Hold'em (NLHE) is particularly challenging because aggressive decisions involve both the choice of an action and a numerical bet or raise size. Different sizes can alter the risks, incentives, and future decision paths associated with the same categorical action (BILLINGS et al., 2003; JOHANSON, 2013).

Rather than developing a complete autonomous poker-playing agent, this dissertation investigates whether open Large Language Models (LLMs) can predict reference decisions from structured textual descriptions of individual game states. Experiments compare few-shot prompting with supervised fine-tuning using preflop and postflop examples derived from PokerBench (ZHUANG et al., 2025). Reported results concern agreement with benchmark evaluation decisions rather than win rate, exploitability, or performance in complete poker matches.

1.1 Context and Motivation

From a computational perspective, poker differs from perfect-information games because players must decide without observing all relevant information. During a hand, each player sees their own private cards and the public cards on the table, but not the private cards held by the opponents. Decisions must therefore consider not only the visible cards, but also table position, stack and pot sizes, previous actions, and the set of available betting options (OSBORNE; RUBINSTEIN, 1994; SANDHOLM, 2015; RUSSELL; NORVIG, 2020).

Betting history is particularly important because earlier actions change both the immediate state of the hand and the interpretation of an opponent's possible holdings. A raise, call, or check may provide indirect information about the strategies and private cards consistent with the observed sequence. Consequently, two situations with similar visible cards may require different decisions when their positions, stack sizes, or previous actions differ (BILLINGS et al., 2003; SANDHOLM, 2015).

Game complexity also depends strongly on the betting rules. In heads-up games, which involve two players, limit Texas Hold'em restricts bet sizes and the number of raises, reducing the number of possible betting sequences. Even under these restrictions, the heads-up limit configuration adopted by the Annual Computer Poker Competition (ACPC), an established competition for evaluating poker-playing programs, contains approximately 3.16×10^{17} game states and 3.19×10^{14} information sets (BARD et al., 2013; JOHANSON, 2013). An information set groups game states that cannot be distinguished by the acting player because some information, such as the opponents' private cards, remains hidden (OSBORNE; RUBINSTEIN, 1994; SANDHOLM, 2015).

No-limit betting produces a substantially larger decision space because the actions available at one point depend on previous bets and on the chips remaining in each player's stack. Stack depth is commonly expressed in big blinds. A big blind is the larger of two mandatory chip contributions posted before the private cards are dealt and also serves as a standard unit for comparing stack, pot, and bet sizes (CBGC, 2016; POKERSTARS, 2026). For an ACPC heads-up no-limit configuration with stacks of 200 big blinds, Johanson reported approximately 6.31×10^{164} game states and 2.76×10^{160} canonical information sets (JOHANSON, 2013). Although these figures depend on the stack depth, chip granularity, and betting rules of the analyzed configuration, they illustrate how numerical bet sizing expands the game tree.

Specialized poker systems have addressed the scale of the game through abstraction, regret minimization, search, neural value estimation, and subgame-solving methods (BILLINGS et al., 2003; ZINKEVICH et al., 2007). Successive systems demonstrated the effectiveness of these approaches in increasingly complex settings: Cepheus nearly solved heads-up limit Texas Hold'em (BOWLING et al., 2015); DeepStack and Libratus achieved expert or superhuman performance in heads-up no-limit poker (MORAVČÍK et al., 2017;

BROWN; SANDHOLM, 2017a); and Pluribus extended superhuman performance to multiplayer no-limit poker (BROWN; SANDHOLM, 2019). ReBeL later combined self-play reinforcement learning with equilibrium-oriented reasoning (BROWN et al., 2020).

Despite their methodological differences, these systems rely on specialized computational machinery designed to construct, approximate, or refine strategies over a sequential game. Commercial study tools expose related strategic information through action frequencies, expected values, range matrices, and alternative bet sizes (GTO Wizard, 2025; PioSOLVER, 2026). Such representations are useful for advanced analysis, but their interpretation requires familiarity with ranges, mixed strategies, betting trees, and solver-specific interfaces.

Language models introduce a distinct experimental perspective. Positions, cards, stack and pot quantities, and action histories can be serialized as structured text and processed through the standard input interface of an LLM. Rather than computing an equilibrium or searching the game tree at query time, a trained model can map a textual decision state directly to a reference-like output. Such prediction may support rapid individual queries, but it does not reproduce the strategy construction, search, or re-solving capabilities of a poker solver.

Recent studies have evaluated LLMs in poker through general-purpose prompting, domain-specific training, external-tool integration, and agent-oriented frameworks (GUPTA, 2023; ZHUANG et al., 2025; HUANG et al., 2024; LIN et al., 2026; MAUGIN; CAZENAVE, 2025; LI et al., 2026; PROVOST et al., 2026). Their settings differ in game format, state representation, model access, training procedure, tool dependence, and evaluation objective. Chapter 3 examines these approaches in detail.

Important methodological questions nevertheless remain. In protocols that represent an aggressive no-limit decision through an action and a numerical size, aggregate action accuracy alone cannot reveal errors in sizing calibration, individual action classes, table positions, postflop streets, or sensitivity to the textual representation.

Against this background, the dissertation conducts a controlled comparison of open LLMs under few-shot prompting and evaluates supervised adaptation of the strongest model. Preflop and postflop behavior, action errors, numerical sizing, and robustness are examined to provide a more detailed account than aggregate accuracy alone.

Potential human-facing applications provide a secondary motivation. Language-based models could support interactive study interfaces or help present strategic information produced by external poker tools in a more accessible form. Usability, explanation quality, and learning outcomes are not evaluated here; the empirical contribution remains restricted to reference-decision prediction and diagnostic analysis.

1.2 Problem Statement and Hypothesis

A central problem investigated in this dissertation is the extent to which open LLMs can predict reference actions from structured textual descriptions of individual 6-max No-Limit Texas Hold'em states. More specifically, the study examines whether supervised adaptation to poker-specific examples improves action prediction relative to few-shot prompting in both preflop and postflop settings.

Performance is measured through agreement with benchmark labels at isolated decision points. Such agreement does not imply that an LLM computes an equilibrium, reproduces the full strategy that generated the labels, or performs the search and re-solving procedures used by specialized poker systems (BOWLING et al., 2015; MORAVČÍK et al., 2017; BROWN; SANDHOLM, 2017a; BROWN; SANDHOLM, 2019). Instead, the LLM is evaluated as a text-based predictor of individual reference decisions, not as a replacement for a solver.

Supervised fine-tuning on poker-specific examples is hypothesized to increase agreement with reference actions relative to few-shot prompting in both settings. Each target contains a categorical action and, for bets or raises, a numerical size expressed in big blinds. Action prediction constitutes the primary task, while sizing is examined as a complementary empirical dimension to determine whether action-level gains are accompanied by improved proportional agreement with the reference sizings.

1.3 Objectives

The general objective of this dissertation is to evaluate the performance of open Large Language Models as predictors of reference actions in 6-max No-Limit Texas Hold'em, with emphasis on the effect of supervised domain adaptation relative to few-shot prompting.

More specifically, the dissertation aims:

- ❑ To systematically evaluate representative open LLMs under few-shot prompting and quantify the effect of supervised fine-tuning on the selected model across preflop and postflop decision settings.
- ❑ To characterize action-prediction performance and error patterns across action classes and relevant poker contexts, including table positions and postflop streets.
- ❑ To determine whether improvements in categorical action prediction are accompanied by improved proportional agreement between predicted and reference bet and raise sizes.

- To assess the robustness of the main empirical findings under selected variations in prompt formulation, evaluation configuration, action-extraction procedure, decoding settings, evaluation-set size, and training-data availability.

1.4 Contributions

Main contributions of this dissertation are summarized as follows:

- **Systematic investigation of supervised adaptation for strategic poker decision prediction.** A controlled study compares six open-weight LLMs under repeated few-shot prompting and subsequently adapts the strongest model to both preflop and postflop decision distributions. Paired statistical tests, confusion matrices, and breakdowns across action classes, table positions, and postflop streets determine not only whether supervised fine-tuning improves aggregate performance, but also where those gains occur and which decision patterns remain difficult. Such experimental coverage provides a detailed account of supervised adaptation in a 6-max No-Limit Texas Hold'em setting that combines categorical actions, numerical sizings, hidden information, and sequential context.

- **Hybrid inference and evaluation framework for action-and-sizing decisions.** Available categorical actions are selected through next-token log-probability scoring, while bet and raise sizings are obtained through deterministic generation and numerical parsing. Separating both components prevents formatting and parsing failures from being mistaken for categorical decision errors, restricts action predictions to the available choices, and enables reproducible analysis of action selection and numerical sizing within the same decision pipeline.

- **Continuous joint metric for categorical correctness and proportional sizing agreement.** A continuous formulation of Action-and-Sizing Accuracy (Ac-s) is developed to evaluate the complete poker decision. Correct aggressive actions receive a score determined by the proportion between predicted and reference sizings, with progressively stronger penalties as their proportional distance increases. Equivalent proportional deviations receive equal penalties regardless of the numerical scale or whether the prediction overestimates or underestimates the reference. A conditional relative sizing score complements Ac-s by measuring numerical agreement only after the aggressive action has been correctly selected, allowing joint decision performance and pure sizing quality to be interpreted separately.

Collectively, the proposed framework connects systematic supervised adaptation, controlled categorical inference, and proportional numerical evaluation in a unified methodology for studying open LLMs as predictors of equilibrium-oriented poker decisions. Result-

ing evidence identifies both the degree to which reference strategies can be approximated from textual states and the remaining boundaries of such approximation, without equating prediction agreement with poker solving or complete strategic play.

1.4.1 Resulting Publications

Research conducted for this dissertation produced publications and manuscripts corresponding to successive stages of the investigation.

An initial few-shot evaluation was published in the paper *Evaluation of LLMs for Effective Recommendation of Poker Strategies*, presented at the XXII Encontro Nacional de Inteligência Artificial e Computacional (ENIAC 2025) (DINIZ; CARNEIRO, 2025). That study compared open LLMs without supervised adaptation and established the preliminary empirical basis for the dissertation.

A subsequent paper, entitled *Supervised Fine-Tuning of Large Language Models for Strategic Decision-Making in Poker*, was accepted for presentation and publication at the 36th Brazilian Conference on Intelligent Systems (BRACIS 2026). Its contribution extends the initial study by incorporating supervised adaptation and comparing fine-tuned performance with few-shot prompting.

A broader manuscript, entitled *PokerLLM: Learning Strategic Poker Play with Large Language Models*, was submitted to *IEEE Transactions on Games*. Its scope consolidates the complete experimental framework developed in the dissertation, including preflop and postflop fine-tuning, action-and-sizing evaluation, statistical validation, contextual analyses, robustness tests, and ablation experiments.

Together, the works trace the progression from few-shot evaluation to supervised domain adaptation and to the broader diagnostic framework consolidated in the dissertation.

1.5 Scope and Delimitations

Interpretation of the reported results is limited to agreement with benchmark labels for isolated 6-max No-Limit Texas Hold'em states under the adopted PokerBench-derived distribution (ZHUANG et al., 2025). Although preflop and postflop examples are included, complete hand trajectories and repeated interaction are not modeled. Consequently, the experiments do not measure general poker-playing ability, long-term win rate, bankroll outcomes, opponent adaptation, or cumulative errors during sequential play.

Equilibrium computation also lies outside the experimental setting. During inference, the evaluated LLMs do not search a game tree, query an external solver, perform real-time re-solving, or construct a complete strategy. Solver-related information is present only through the reference labels used for training and evaluation. Agreement with those labels therefore does not establish that a model reproduces the underlying solver policy,

computes an equilibrium, or possesses the strategic and computational capabilities of a specialized poker solver.

Categorical action agreement constitutes the primary evaluation target. For bets and raises, sizing results measure numerical proximity to the available reference. Such results do not estimate expected value, quantify exploitability, evaluate mixed-strategy frequencies, or determine whether alternative sizes could also be strategically viable.

Only open LLMs are included, supporting local execution, reproducibility, and control over adaptation and evaluation. Closed proprietary models, multimodal inputs, online search, and external tools during inference are excluded. Alternative learning and planning paradigms—including self-play reinforcement learning, Monte Carlo Tree Search, online planning, opponent modeling, and specialized regression models for sizing—are also outside the scope.

Natural-language explanations are not part of the evaluated output. Available examples provide action and sizing labels but no reference rationales against which generated explanations could be validated. Explanation quality, pedagogical value, user interaction, and learning outcomes therefore remain outside the empirical scope.

1.6 Structure of the Dissertation

Remaining chapters are organized as follows.

- **Chapter 2 – Background.** Presents the rules and decision structure of No-Limit Texas Hold'em, the 6-max format, preflop and postflop stages, imperfect-information games, game-theoretic concepts, and the foundations of Large Language Models, prompting, and supervised fine-tuning.
- **Chapter 3 – Related Work.** Reviews specialized poker artificial intelligence, imperfect-information game solving, data-driven poker analysis, and recent applications of Large Language Models to poker and strategic decision-making.
- **Chapter 4 – Problem Formulation.** Defines the observable poker state, action space, and numerical sizing component, and formalizes the preflop and postflop prediction tasks.
- **Chapter 5 – Materials and Methods.** Describes the datasets, prompt representation, evaluated models, few-shot configuration, supervised fine-tuning procedures, evaluation pipeline, performance metrics, statistical tests, ablations, and reproducibility measures.
- **Chapter 6 – Results and Discussion.** Reports and interprets few-shot and fine-tuned performance, action and sizing results, error patterns, statistical comparisons, contextual analyses, robustness tests, and ablation findings.

- **Chapter 7 – Conclusion.** Consolidates the main findings, revisits the hypothesis and objectives, discusses the limitations of the evidence, and presents directions for future research.

Chapter 2

Background

Understanding the research problem, experimental design, and interpretation of the results requires two complementary foundations. The first is the structure of No-Limit Texas Hold'em as a decision environment: private information, public information, available actions, sequential betting, and strategic uncertainty. The second is the use of Large Language Models as text-based prediction systems that can process structured representations of such decision states and be adapted through prompting or supervised fine-tuning.

Poker concepts are presented together with their interpretation, so that terms such as preflop, postflop, position, street, stack, action, and sizing can later be used without ambiguity. The chapter also introduces the game-theoretic intuition behind equilibrium-oriented strategies and the main adaptation methods used in the experimental methodology.

2.1 No-Limit Texas Hold'em as a Decision Environment

Poker is a family of card games in which players compete for a pot formed by the chips committed during a hand. A player may win the pot either by holding the strongest eligible hand at showdown, when the remaining players compare their cards, or by causing all opponents to fold, meaning that they abandon the hand before this comparison occurs (CBGC, 2016; POKERSTARS, 2026). Betting therefore determines both how many chips are placed at risk and whether opponents continue in the hand.

No-Limit Texas Hold'em (NLHE) is a community-card poker variant in which each player receives two private cards, known as hole cards, and may combine them with five public community cards to form the best possible five-card hand (CBGC, 2016). Chips committed during the hand accumulate in the pot, which is awarded according to the outcome of the hand.

Before the private cards are dealt, two players post mandatory chip contributions

called the small blind and the big blind. A big blind is the larger of these two contributions and serves as a standard unit for expressing stack, pot, bet, and raise sizes (CBGC, 2016; POKERSTARS, 2026). A stack denotes the chips currently available to a player, whereas the pot contains the chips already committed during the hand.

Four betting rounds, commonly called streets, organize the progression of a hand. Preflop occurs before any community cards are revealed. Three community cards are then revealed on the flop, followed by one additional card on the turn and another on the river. Flop, turn, and river are collectively referred to as postflop. Betting may occur during each street until all active players have matched the current bet, only one player remains, or the hand reaches showdown (CBGC, 2016).

No-limit rules permit a player to bet or raise any available amount up to the chips remaining in their stack (JOHANSON, 2013). Numerical flexibility substantially expands the number of possible betting sequences because different sizes change the pot, the amount required for opponents to continue, and the decisions that may become available later in the hand (BILLINGS et al., 2003; JOHANSON, 2013).

Six-max NLHE is played with up to six seats. Standard positions are under the gun (UTG), hijack (HJ), cutoff (CO), button (BTN), small blind (SB), and big blind (BB). Position determines the order of action. Players acting later in a betting round observe more of their opponents' decisions before selecting their own action, making position an important component of the decision context (BILLINGS et al., 2003).

Available information combines private, public, and contextual elements. Hole cards are private to the acting player, while community cards, pot size, stack quantities, and previous actions are observable. Opponents' hole cards remain hidden. Earlier actions are therefore part of the observable state: they change pot and stack quantities and provide evidence about the hidden holdings and strategies consistent with the current hand. The adopted representation includes this within-hand history, but not persistent opponent profiles across multiple hands (OSBORNE; RUBINSTEIN, 1994; SANDHOLM, 2015).

Available actions depend on the current betting situation. Folding abandons the hand, calling matches an outstanding bet, and checking passes the action without committing additional chips when no bet is pending. Betting introduces chips when no previous bet exists on the current street, whereas raising increases an existing bet. Not every action is available in every state: for example, a player cannot check while facing an unmatched bet and cannot call when no bet is pending (CBGC, 2016; POKERSTARS, 2026).

Preflop and postflop states contain different amounts and types of observable information. Preflop decisions depend on hole cards, positions, stacks, blinds, and the preceding actions before community cards are revealed. Postflop decisions additionally incorporate the public board, current street, pot size, and betting history accumulated over earlier stages. Such differences motivate their treatment as separate prediction settings in the experimental design (ZHUANG et al., 2025).

Figure 1 summarizes the information associated with a simplified postflop decision. Private cards belong to the acting player, while the board, pot, stacks, positions, and previous actions form the observable context from which a decision must be predicted.

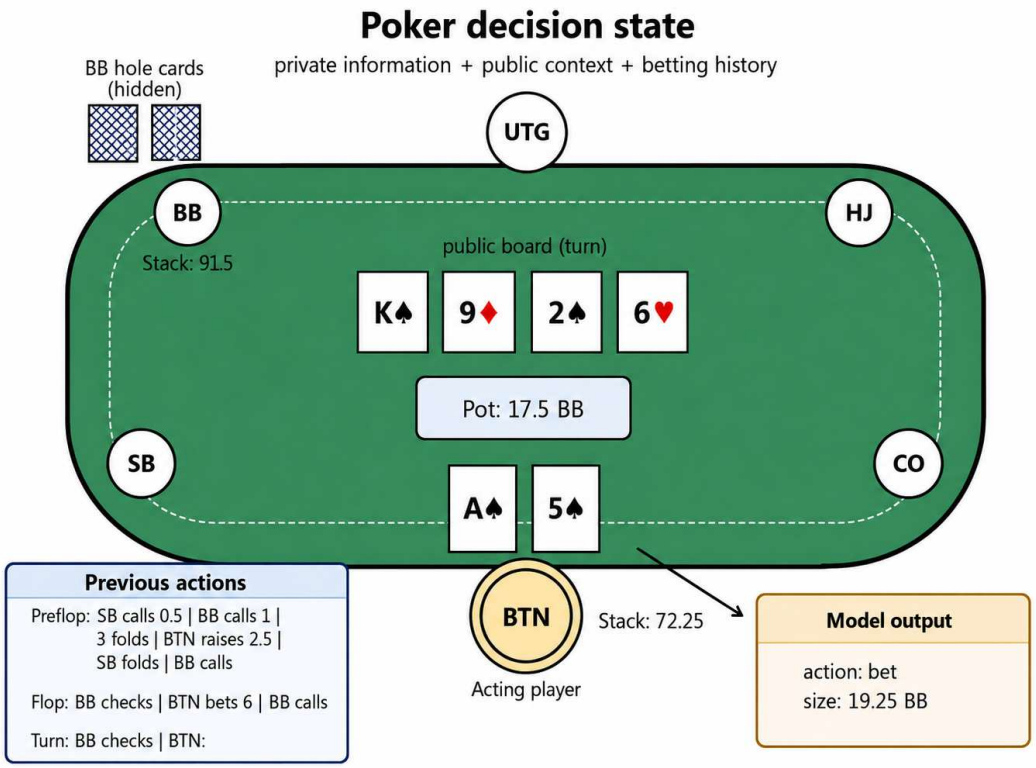

Figure 1 – Simplified 6-max No-Limit Texas Hold'em decision state with private cards, public context, actions and model output.

Formal definitions of the observable state, action space, and prediction target are presented in Sections 4.1 and 4.2.

2.2 Imperfect Information and Game Theory Optimal Strategies

Poker is an imperfect-information game because no player observes the complete state of a hand. Each player knows their own hole cards and the public information revealed so far, but opponents' private cards remain hidden. Decisions must therefore account for multiple hidden states that may be compatible with the same observable board, stack configuration, and betting history (OSBORNE; RUBINSTEIN, 1994; SANDHOLM, 2015).

An information set groups game states that are indistinguishable to the acting player. For example, different opponent hole cards may produce the same observable information at a particular decision point. A strategy specifies how a player acts at each information set, while a mixed strategy assigns probabilities to multiple possible actions instead

of always selecting a single deterministic response (OSBORNE; RUBINSTEIN, 1994; SHOHAM; LEYTON-BROWN, 2008).

Strategic randomization helps prevent a player's behavior from becoming fully predictable and reduces opportunities for an opponent to exploit deterministic patterns. In poker, the same class of hands may therefore be associated with different actions at different frequencies, such as betting in some instances and checking in others (BILLINGS et al., 2003; SANDHOLM, 2015).

Game theory provides a formal framework for analyzing these strategic interactions (NEUMANN; MORGENSTERN, 1944; NASH, 1951; OSBORNE; RUBINSTEIN, 1994). Let S_i denote the strategy set of player i, and let $s_i \in S_i$ represent one possible strategy. A strategy profile is written as (s_i, s_{-i}), where s_{-i} contains the strategies of the other players. A Nash equilibrium is a profile $s^* = (s_i^*, s_{-i}^*)$ in which no player can improve their expected outcome by changing strategy unilaterally:

$$u_i(s_i^*, s_{-i}^*) \geq u_i(s_i, s_{-i}^*) \quad \forall s_i \in S_i, \ \forall i. \tag{1}$$

In this dissertation, Game Theory Optimal (GTO) refers to equilibrium-oriented poker strategies that approximate a Nash-equilibrium solution for a defined game configuration (BILLINGS et al., 2003; ZHUANG et al., 2025). Rather than assigning one fixed action to every private hand, such strategies may distribute folds, calls, checks, bets, or raises across different frequencies and numerical sizes (JOHANSON, 2013).

Poker solvers are specialized computational systems used to approximate strategies for defined poker configurations. Their outputs may organize private hands into ranges and associate them with action frequencies and candidate bet sizes. Consequently, the resulting output may describe a distribution over multiple hands and actions rather than a single universal recommendation (BILLINGS et al., 2003; JOHANSON, 2013; MORAVČÍK et al., 2017; BROWN; SANDHOLM, 2017a).

Figure 2 provides an AI-generated schematic illustration of this type of output. Colored portions of the hand matrix represent hypothetical action frequencies, while the enlarged examples show that one private hand may be assigned to more than one action. Displayed hands, colors, and percentages are illustrative only and do not constitute experimental data or outputs reproduced from a commercial solver.

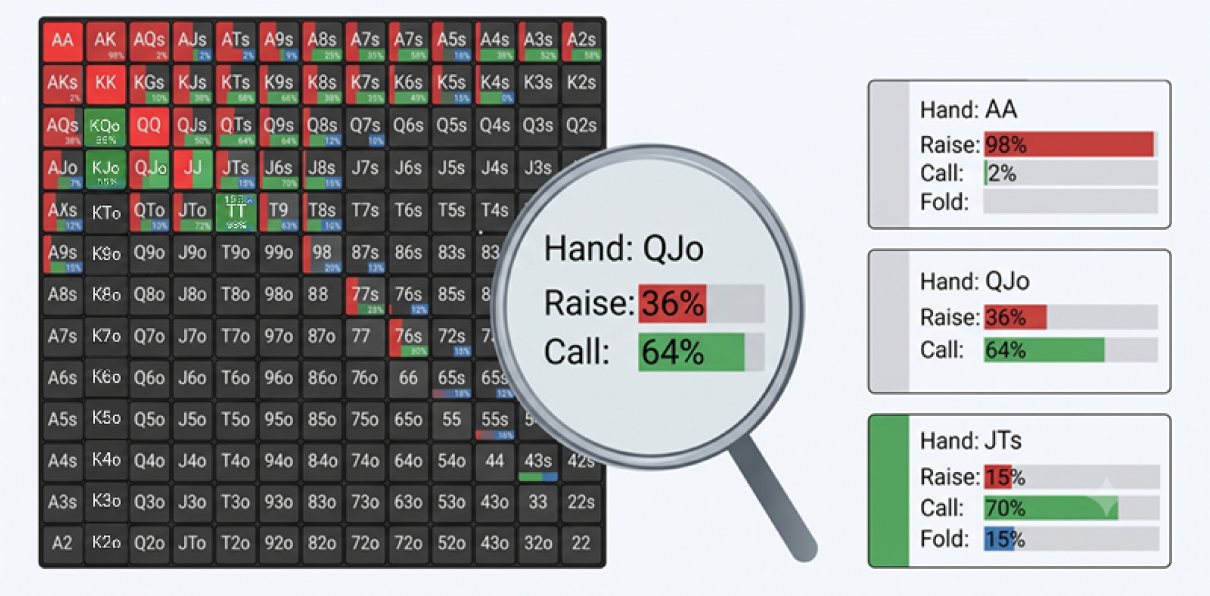

Figure 2 – AI-generated illustration of a solver-style poker output with hypothetical mixed action frequencies.

Mixed action distributions such as those illustrated in the figure are not preserved in full by the prediction task adopted in this dissertation. During the construction of PokerBench, the underlying poker space was reduced through filtering and scenario selection. Board situations were organized into representative textures, and hole-card situations were selected when one action had a dominant probability greater than 50% (ZHUANG et al., 2025). Each resulting example therefore contains a single target output rather than the complete mixed action distribution.

Agreement with a PokerBench-derived label consequently measures whether the model predicts the selected dominant reference decision for an individual state. It does not measure reproduction of the full mixed strategy or the frequencies assigned to alternative actions. Chapter 4 formally defines the observable state, action space, numerical sizing component, and prediction target.

2.3 Decision Outputs and State Representation

A poker decision in NLHE contains a categorical component and, for bets and raises, a numerical component. The categorical component identifies one of the actions defined in Section 2.1, while the numerical component specifies the size of an aggressive action.

In the formulation adopted here, action prediction is treated as a constrained classification problem because evaluation is restricted to the actions that are available in the current state. Poker rules determine which actions are available, while PokerBench

represents the target decision through an action label (CBGC, 2016; ZHUANG et al., 2025).

A numerical component is required when the action is aggressive, namely a bet or raise. In no-limit poker, the numerical size of an aggressive action is strategically meaningful. A small bet and a large bet may share the same categorical label but create different risks, incentives, pot sizes, and subsequent decision conditions (BILLINGS et al., 2003; JOHANSON, 2013). Bet sizing can therefore be interpreted as numerical action-parameter prediction: the model must choose an action category and, when required, predict its associated numerical value.

Throughout this dissertation, sizes are expressed in big blinds (BB). A big blind is the larger mandatory blind posted before the hand begins and provides a standard unit for representing poker quantities. Expressing stacks, pots, bets, and raises in big blinds makes decision states comparable across games played at different monetary stakes (CBGC, 2016; POKERSTARS, 2026).

A decision state is represented as a structured object containing the information available to the acting player:

$$s = (h, b, p, r, q, \alpha), \tag{2}$$

where h denotes the acting player's private hole cards, b denotes the public board cards when available, p denotes the player's position, r denotes the current betting round or street, q denotes numerical quantities such as stack and pot size, and α denotes the sequence of previous actions.

Under the formulation adopted in this dissertation, the structured state s is serialized into text before being supplied to the LLM. The model must then predict a decision represented as

$$d = (a, z), \tag{3}$$

where a is the categorical action and z is a numerical size when the action requires one. The global action set can be written as

$$\mathcal{A} = \{\texttt{fold}, \texttt{call}, \texttt{check}, \texttt{bet}, \texttt{raise}\}. \tag{4}$$

For a particular state s, evaluation considers only the available subset $\mathcal{A}(s) \subseteq \mathcal{A}$. Such a distinction is necessary because the rules and current betting context constrain which actions can be selected.

Separating action selection from numerical sizing is one of the motivations for the evaluation methodology. A model may correctly predict a raise while producing a size far from the reference decision. Conversely, a plausible numerical value does not compensate for an incorrect categorical action. Evaluating the two components individually helps identify whether an error arises from action selection, numerical sizing, or both.

2.4 Large Language Models and Structured Text Representations

Large Language Models are neural sequence models trained on large text corpora for probabilistic language modeling and text generation. Most modern LLMs are based on the Transformer architecture, which uses attention mechanisms to model relationships between tokens at different positions in a sequence (VASWANI et al., 2017; BOMMASANI et al., 2021; RUSSELL; NORVIG, 2020).

A token is a discrete unit produced by the model's tokenizer and may correspond to a complete word, part of a word, punctuation, or another textual symbol. LLMs process and predict sequences of such units rather than complete sentences as indivisible objects (VASWANI et al., 2017; BOMMASANI et al., 2021).

Large-scale pretrained language models are often discussed as foundation models because their general-purpose representations can be adapted to multiple downstream tasks (BOMMASANI et al., 2021). Empirical scaling studies suggest that model size, training-data scale, and computational resources can substantially affect language-model performance, although scale alone does not guarantee reliable behavior in specialized decision domains (KAPLAN et al., 2020; BROWN et al., 2020). Accordingly, this dissertation evaluates the models directly on the specialized poker decision task rather than inferring domain performance from general language-model benchmarks.

One of the most common language-modeling approaches is autoregressive prediction, in which the probability of each token is conditioned on the preceding sequence. Given $x_{1:T} = (x_1, x_2, \ldots, x_T)$, the joint probability is factorized as

$$P_\theta(x_{1:T}) = \prod_{t=1}^{T} P_\theta(x_t \mid x_{<t}), \tag{5}$$

where $x_{<t}$ denotes the tokens preceding position t, and θ represents the model parameters. During training, the model learns to assign higher probability to the observed next token. A common objective is the negative log-likelihood, equivalently expressed as cross-entropy loss:

$$\mathcal{L}(\theta) = -\sum_{t=1}^{T} \log P_\theta(x_t \mid x_{<t}). \tag{6}$$

During pretraining, the model learns statistical regularities of language and structured text from large-scale data. Instruction tuning subsequently adapts pretrained models to follow task descriptions and produce responses in expected formats (WEI et al., 2022; OUYANG et al., 2022). Such processes allow LLMs to perform multiple tasks through natural-language instructions.

For the purposes of this dissertation, the most relevant property of LLMs is their ability to process structured textual representations. A poker decision state can be converted

into a prompt containing categorical information, numerical quantities, and sequential actions. Instead of receiving a manually engineered numerical feature vector, the model processes a textual serialization of the available state information.

Such a conversion is referred to in this dissertation as textualization. It provides a common input format for heterogeneous elements such as positions, cards, stack sizes, pot quantities, and betting histories. Model performance then depends partly on whether the relevant strategic variables can be recovered from that representation and mapped to the appropriate action and, when applicable, numerical size.

Within the poker setting, a prompt is not merely a natural-language explanation intended for human readability. It constitutes the model's representation of a partially observable decision state. Consequently, changes in wording, ordering, or formatting may influence how the model interprets the state, a possibility examined through the robustness analyses reported later in the dissertation.

Publicly available model weights support local inference, controlled prompting, and parameter-efficient adaptation. Accordingly, the experiments consider instruction-tuned models from the Llama, Gemma, Mistral, and Qwen families (GRATTAFIORI et al., 2024; Gemma Team, 2024; JIANG et al., 2023; YANG et al., 2025).

2.5 Prompting and Supervised Fine-Tuning

LLMs can be adapted to a task in different ways. Prompting is one approach: the model receives an instruction describing the task and produces an answer without updating its parameters. In zero-shot prompting, no labeled examples are provided in the prompt. In few-shot prompting, the prompt includes a small number of input-output examples before the query, demonstrating the expected format and behavior (BROWN et al., 2020; LIU et al., 2023; ZHAO et al., 2021; LU et al., 2022).

Few-shot prompting requires no parameter updates and can be applied directly to an instruction-tuned model. Its performance, however, may vary with instruction wording, demonstration selection, example ordering, and output format (ZHAO et al., 2021; LU et al., 2022; LIU et al., 2023). In a specialized domain such as poker, demonstrations may clarify the expected input-output format, although they provide much less task-specific supervision than a labeled fine-tuning dataset.

Supervised fine-tuning (SFT) provides a more direct adaptation mechanism. In SFT, the model is trained on labeled examples from the target domain, updating its parameters so that it becomes more likely to produce the desired outputs. For poker decision prediction, this means training the model on examples that pair textual decision states with reference actions and, when applicable, numerical sizes. Instruction tuning and supervised fine-tuning have been widely used to adapt pretrained LLMs to follow task instructions and produce target responses more reliably (SANH et al., 2022; WEI et al.,

2022; OUYANG et al., 2022).

If the training dataset contains input-output pairs (x_i, y_i), where x_i is the prompt and y_i is the target output, SFT minimizes the negative log-likelihood of the target tokens conditioned on the input:

$$\mathcal{L}_{\mathrm{SFT}}(\theta) = -\sum_{i=1}^{N}\sum_{t=1}^{|y_i|}\log P_\theta(y_{i,t} \mid x_i, y_{i,<t}). \tag{7}$$

In instruction-style fine-tuning, the loss is typically computed over the target response tokens rather than over the entire prompt. Thus, the model is optimized to generate the desired decision conditioned on the textual representation of the poker state. In this setting, fine-tuning is not used to teach the rules of poker from scratch, but to adapt a general-purpose instruction-following model to the specific distribution of poker decision examples.

The comparison between the two regimes evaluates how much performance can be elicited through in-context examples and how much additional improvement results from supervised domain adaptation.

2.6 Parameter-Efficient Fine-Tuning with LoRA and QLoRA

Fine-tuning large language models can be computationally expensive because it may require updating billions of parameters. Parameter-efficient fine-tuning methods reduce this cost by updating only a small number of additional parameters while keeping most of the original model fixed (LESTER et al., 2021; HU et al., 2022; DETTMERS et al., 2023). Low-Rank Adaptation (LoRA) is one such method. Instead of directly updating a pretrained weight matrix W, LoRA represents the update as a low-rank decomposition (HU et al., 2022). If $W \in \mathbb{R}^{d\times k}$ is a frozen pretrained weight matrix, the adapted weight can be written as

$$W' = W + \Delta W, \tag{8}$$

where

$$\Delta W = BA, \tag{9}$$

with $B \in \mathbb{R}^{d\times r}$ and $A \in \mathbb{R}^{r\times k}$, where $r \ll \min(d, k)$. The matrices A and B are trainable, while W remains frozen. This greatly reduces the number of trainable parameters.

LoRA is commonly applied to attention and projection matrices of Transformer models, under the assumption that the task-specific weight update can be represented through a low-rank decomposition (HU et al., 2022).

QLoRA extends this idea by combining LoRA with quantization. Quantization represents model weights using fewer bits, reducing memory consumption during training. In QLoRA, the base model is loaded in a quantized format, while low-rank adapters are trained on top of it. Such a configuration enables supervised fine-tuning of larger models on more accessible hardware (DETTMERS et al., 2023).

For the present research, LoRA and QLoRA are relevant because they make supervised adaptation feasible without requiring full fine-tuning of all model parameters. As a result, the evaluation can compare prompting with domain-specific adaptation while maintaining a practical computational setup.

Together, these foundations connect the domain-specific formulation of poker decisions with the computational methodology used in the dissertation. Poker provides the structured imperfect-information decision states, while LLMs provide the text-based prediction mechanism. The next chapter reviews related work on poker AI, data-driven strategic modeling, and LLM-based decision prediction.

Chapter 3

Related Work

Research on computational poker spans distinct problem formulations, game settings, state representations, and evaluation protocols. Specialized poker agents operate over explicit game structures, numerical abstractions, action-probability distributions, search trees, regret values, and neural estimates of future utility (BILLINGS et al., 2003; ZINKEVICH et al., 2007; MORAVČÍK et al., 2017; BROWN et al., 2019). Data-driven approaches instead learn predictive relationships from hand histories encoded through numerical or categorical features (TEÓFILO, 2010; CARNEIRO; LISBOA, 2018; DINIZ, 2022). More recent studies investigate Large Language Models (LLMs) that receive textual or semi-structured poker states and produce actions, numerical sizes, explanations, or agent decisions (GUPTA, 2023; HUANG et al., 2024; ZHUANG et al., 2025; LIN et al., 2026; MAUGIN; CAZENAVE, 2025).

Meaningful comparison across such research lines requires attention to the underlying game setting. Cepheus addresses heads-up limit Texas Hold'em; DeepStack and Libratus address heads-up no-limit Texas Hold'em; Pluribus addresses multiplayer no-limit Texas Hold'em; and ReBeL includes experiments in Leduc poker and larger imperfect-information games (BOWLING et al., 2015; MORAVČÍK et al., 2017; BROWN; SANDHOLM, 2017a; BROWN; SANDHOLM, 2019; BROWN et al., 2020). Variations in card structure, number of players, betting rules, available action sizes, and computational complexity prevent results from being interpreted independently of the game in which they were obtained.

Evaluation protocols differ as well. Specialized systems may be assessed through equilibrium approximation, exploitability, or performance over complete matches; supervised predictors may reproduce observed human actions or player characteristics; and language-model benchmarks may measure agreement with reference decisions for isolated states (BOWLING et al., 2015; BROWN; SANDHOLM, 2017a; BROWN; SANDHOLM, 2019; CARNEIRO; LISBOA, 2018; DINIZ, 2022; ZHUANG et al., 2025). Results from these protocols are informative within their respective objectives but are not directly interchangeable.

Organization of the chapter follows the representation and decision mechanism adopted by each research line. Specialized poker-solving systems and their non-textual representations are reviewed first. Feature-based and data-driven methods are then examined as alternative formulations of poker prediction. Subsequent sections discuss LLM-based decision systems, including models adapted directly to poker data, systems augmented with external solvers, and benchmarks based on isolated decisions or complete play. A final synthesis positions the present dissertation according to game setting, state representation, adaptation method, solver dependence, prediction target, and evaluation protocol.

3.1 Specialized Poker Solving and Non-Textual Representations

Specialized poker-solving methods represent the game through explicit computational structures rather than natural-language descriptions. Common representations include extensive-form game trees, information sets, abstractions of private and public cards, restricted betting actions, regret tables, probability distributions over actions, and numerical estimates of future utility. Optimization, self-play, search, subgame solving, and equilibrium-approximation procedures operate directly on such structures (BILLINGS et al., 2003; ZINKEVICH et al., 2007; JOHANSON, 2013; SANDHOLM, 2015).

Game abstraction reduces the computational demands of poker variants containing very large numbers of private-card combinations, public boards, betting histories, and information sets. Card abstraction groups hands or public states considered strategically similar, whereas action abstraction restricts the bet and raise sizes represented in the game tree. Computation consequently becomes more tractable, although the resulting strategy is calculated for a simplified representation of the original game (BILLINGS et al., 2003; JOHANSON, 2013; SANDHOLM, 2015).

Counterfactual Regret Minimization (CFR) approximates equilibrium strategies by repeatedly traversing an extensive-form game and updating counterfactual regret values for the actions available at each information set. Accumulated positive regrets determine mixed action probabilities, while the strategies produced over successive iterations are averaged. Unlike supervised prediction, CFR does not require labeled state-action examples because strategic information is generated through repeated interaction with the represented game (ZINKEVICH et al., 2007).

Cepheus applied large-scale regret minimization and abstraction to heads-up limit Texas Hold'em, a two-player variant with a fixed betting structure. Its strategy was reported as essentially solving that game according to the remaining degree of exploitability. Outputs corresponded to probability distributions over available actions at information sets, rather than textual recommendations generated independently for isolated prompts (BOWLING et al., 2015).

DeepStack addressed heads-up no-limit Texas Hold'em, where variable bet and raise sizes produce a larger action space. Continual re-solving constructed local strategies from the current public state, depth-limited lookahead restricted future game-tree expansion, and a neural value function estimated outcomes beyond the search horizon. Numerical card ranges and game quantities therefore supported a specialized search procedure rather than a text model that directly generated an action (MORAVČÍK et al., 2017).

Libratus also targeted heads-up no-limit Texas Hold'em. Its architecture combined a precomputed blueprint strategy, nested subgame solving, and mechanisms for identifying and repairing weaknesses in the initial strategy. Evaluation occurred through complete matches against expert human players, rather than through classification of independent decision states (BROWN; SANDHOLM, 2017a; BROWN; SANDHOLM, 2017b).

Pluribus extended superhuman match performance to multiplayer no-limit Texas Hold'em, including six-player games. A blueprint strategy was generated through self-play and combined with a limited search procedure during play. Although its number of players is closer to the 6-max setting adopted here, Pluribus acted over complete matches and did not receive textualized states or predict one benchmark label for each independent example (BROWN; SANDHOLM, 2019).

ReBeL combined reinforcement learning and search through public-belief-state representations. Such states combine publicly observed information with probability distributions over hidden private states, enabling value learning and search under imperfect information. Experiments included Leduc poker, a simplified research game with a reduced deck and betting structure, alongside larger game settings. Evidence obtained in Leduc is useful for evaluating the algorithm under controlled conditions but is not directly transferable to six-player no-limit Texas Hold'em (BROWN et al., 2020).

Deep CFR incorporated neural function approximation into regret minimization by training networks to estimate counterfactual regret and average-strategy quantities. Numerical encodings of game information served as inputs, replacing some of the large tabular structures required by conventional CFR. Despite its use of neural networks, Deep CFR remains distinct from an LLM predictor because the networks approximate internal quantities required by a solving algorithm rather than process natural-language states and generate action-and-sizing responses (BROWN et al., 2019).

Evidence from Cepheus, DeepStack, Libratus, and Pluribus concerns solution quality or performance over complete matches in their respective poker settings (BOWLING et al., 2015; MORAVČÍK et al., 2017; BROWN; SANDHOLM, 2017a; BROWN; SANDHOLM, 2019). Direct numerical comparison with the present dissertation would be inappropriate because those systems use explicit game models and were not evaluated on the PokerBench-derived isolated action-and-sizing task (ZHUANG et al., 2025).

Solver-based computation may assume different architectural roles. In DeepStack and Libratus, solving machinery forms part of the decision process itself (MORAVČÍK et al.,

2017; BROWN; SANDHOLM, 2017a). ToolPoker instead uses external poker solvers to support action selection and strategic explanations during inference (LIN et al., 2026). Solver-augmented LLM systems are examined in Section 3.4.

3.2 Feature-Based and Data-Driven Poker Decision Prediction

Feature-based approaches formulate poker as a supervised prediction problem using structured numerical and categorical variables extracted from hand histories. Instead of solving an explicit game tree or processing a textual prompt, such methods represent poker situations through fixed-length vectors and train classifiers to predict observed actions or player characteristics (TEÓFILO, 2010; CARNEIRO; LISBOA, 2018; DINIZ, 2022).

Carneiro and de Lisboa predicted player actions in Zoom Texas Hold'em from engineered features describing hand quality, position, aggressiveness, and the current game situation. Among the evaluated classifiers, nonlinear models such as a Multilayer Perceptron were used to capture patterns in observed human behavior (CARNEIRO; LISBOA, 2018). Unlike the present dissertation, their study relied on numerical feature vectors and human action labels rather than textualized states and PokerBench-derived reference decisions.

Diniz analyzed online poker through statistical and machine-learning methods applied to aggregated behavioral features extracted from hand histories. Tree-based classifiers, including Random Forest, were used to distinguish winning and losing players (DINIZ, 2022). Although the target was player-level classification rather than state-level action and sizing prediction, the study provides another example of a non-textual representation for poker modeling.

Taken together, these studies constitute non-textual supervised formulations of poker modeling based on structured features rather than natural-language representations (TEÓFILO, 2010; CARNEIRO; LISBOA, 2018; DINIZ, 2022). Their reported results are not directly comparable with those of the present dissertation because the input representations, prediction targets, datasets, and evaluation units differ.

3.3 Large Language Models for Structured and Strategic Decision-Making

Large Language Models have been applied beyond conventional language-generation tasks to settings involving reasoning, planning, tool use, and interaction with structured environments (BROWN et al., 2020; MIALON et al., 2023; GALLOTTA et al., 2024; HU et al., 2024). Such applications rely on the ability of Transformer-based models to

process instructions and contextual examples as token sequences (VASWANI et al., 2017; BROWN et al., 2020).

Structured decision problems, however, are not naturally represented as unrestricted prose. Their states may combine categorical variables, numerical quantities, ordered events, constraints, and partially observable information. LLM-based approaches commonly convert those elements into textual or semi-structured sequences that preserve relevant state variables while remaining compatible with the model's language interface (MIALON et al., 2023). In this dissertation, such a conversion is referred to as textualization.

Domain-specific applications also require more than fluent text generation. Financial LLMs, for example, combine specialized data curation, task-specific evaluation, and parameter-efficient adaptation to process numerical and contextual information (YANG et al., 2023; LI et al., 2023). Although financial prediction and poker decision-making have different objectives, both illustrate that general language ability does not by itself establish reliable performance in a specialized decision domain.

Games provide controlled environments for studying the connection between language models and decision-making. LLMs have been investigated as agents, planners, non-player characters, evaluators, and interfaces to external tools (GALLOTTA et al., 2024; HU et al., 2024). Game states represented through text can be mapped to actions directly, but complete game-playing systems may also require memory, search, environment interaction, or specialized computational tools.

Tool-augmented architectures are especially relevant when a decision depends on calculations or search procedures that are difficult to reproduce through language modeling alone. In such systems, an LLM may interpret a request, select or query an external component, and express the resulting recommendation, while the tool performs the specialized computation (MIALON et al., 2023). Applied to poker, that component may be a solver that provides an equilibrium-oriented action, value estimate, or strategic analysis.

Game-theoretic settings introduce additional challenges because a decision may depend on hidden information, mixed strategies, incentives, and possible responses from other agents. Research examining LLMs in strategic interactions reports both emerging capabilities and persistent limitations in game-theoretic reasoning (SUN et al., 2025). Consequently, fluent explanations or familiarity with poker terminology cannot be treated as sufficient evidence of decision quality.

Evaluation must instead examine the actions produced for controlled states and distinguish among different system roles. An LLM may act as a direct predictor trained from labeled decisions, an autonomous agent interacting over complete games, or an interface to an external solver. Such roles involve different sources of strategic information and require different evaluation protocols. Section 3.4 reviews how existing poker studies instantiate these alternatives.

3.4 Large Language Models in Poker

Initial research on LLMs in poker focused on determining whether general-purpose models already contained useful strategic knowledge from pretraining. Gupta evaluated ChatGPT and GPT-4 in raise-first-in preflop decisions for nine-player No-Limit Texas Hold'em. Prompts described the player's position and private cards, and the resulting decision matrices were compared with reference preflop charts. No poker-specific parameter adaptation was performed. Both models displayed sensitivity to hand strength and position, but their action distributions remained inconsistent with the adopted game-theoretic references (GUPTA, 2023).

Such an evaluation established that fluent poker knowledge and appropriate terminology do not necessarily produce reliable decisions. Its scope was nevertheless restricted to the first voluntary preflop action, without community cards, longer action histories, model adaptation, or explicit numerical sizing. Later work therefore shifted from testing general-purpose models directly toward constructing poker-specific training and evaluation pipelines.

PokerGPT represented an early attempt to adapt a lightweight LLM for multiplayer Texas Hold'em. Historical game records were filtered, transformed into textual instructions, and used to fine-tune the model through a pipeline described by the authors as involving reinforcement learning from human feedback. Data preparation emphasized actions from players with high observed win rates, making the learned target a representation of successful human behavior rather than an equilibrium strategy computed for every state (HUANG et al., 2024).

Evaluation of PokerGPT emphasized agent-level properties such as win rate, model size, training time, and response speed. Such an approach differs from Gupta's prompt-only analysis because poker-specific parameters are learned from real game records and the model is intended to operate in multiplayer play. However, performance is assessed primarily through gameplay rather than through a fixed collection of isolated reference decisions, which makes detailed diagnosis across action classes and numerical sizes more difficult (HUANG et al., 2024).

PokerBench introduced a different formulation by treating poker decision-making as a standardized instruction-response benchmark. Its evaluation set contains 11,000 6-max No-Limit Texas Hold'em states, divided into 1,000 preflop and 10,000 postflop examples. Preflop references were collected from equilibrium-oriented strategy data, while postflop states were solved computationally. Each state is serialized as text and paired with one reference action and, when applicable, a numerical size (ZHUANG et al., 2025).

Construction of PokerBench reduces the underlying strategic distribution to a supervised prediction target. Board situations were selected to cover representative textures, and hole-card situations were filtered so that one action had probability greater than 50%. Evaluation then uses Action Accuracy for the categorical decision and Exact Match

for the complete action-and-sizing response (ZHUANG et al., 2025). Such a protocol provides faster and more reproducible model comparison than repeated full-game simulations, although it evaluates agreement with selected individual references rather than reproduction of complete mixed strategies.

PokerBench compared closed and open LLMs under few-shot prompting and also fine-tuned Llama and Gemma models on accompanying poker examples. Fine-tuning produced substantial improvements, and additional heads-up simulations showed that checkpoints with higher benchmark scores defeated lower-scoring checkpoints. The authors also reported that straightforward supervised fine-tuning remained insufficient for learning a complete optimal playing strategy (ZHUANG et al., 2025). Consequently, PokerBench established both a reusable evaluation protocol and a direct motivation for studying adaptation methods and diagnostic behavior more systematically.

SpinGPT expanded LLM adaptation to Spin & Go, a three-player tournament format with different stack dynamics and strategic conditions from fixed-stack 6-max cash-game states. Training followed two stages: supervised fine-tuning on 320,000 decisions from high-stakes experts and reinforcement learning on 270,000 solver-generated hands. Reported evaluation included tolerant agreement with solver actions and agent play against an existing poker bot (MAUGIN; CAZENAVE, 2025).

Results reported for SpinGPT include 78% tolerant action agreement with the solver, followed by a heads-up gameplay experiment using an additional deep-stack heuristic (MAUGIN; CAZENAVE, 2025). Its contribution therefore goes beyond isolated supervised prediction by combining human demonstrations, solver-generated trajectories, and reinforcement learning. Direct comparison with this dissertation remains limited because Spin & Go is a tournament format, training targets combine expert and solver sources, and the reported metrics do not follow the same 6-max preflop/postflop protocol.

ToolPoker represents a distinct solver-augmented direction. Lin et al. first evaluated LLMs on two-player imperfect-information games of increasing complexity, including Kuhn poker, Leduc Hold'em, and Limit Texas Hold'em. Their analysis examined both gameplay and generated reasoning, identifying recurrent reliance on shallow heuristics, factual errors, and cases in which the selected action contradicted an apparently correct explanation (LIN et al., 2026).

Behavior cloning and step-level reinforcement learning improved the form of the generated reasoning but did not independently recover reliable game-theoretic behavior. ToolPoker therefore teaches the LLM to query external poker solvers during its reasoning process. Solver responses provide actions and supporting quantities, while the LLM integrates those outputs into a structured recommendation and explanation (LIN et al., 2026). Here, solving is not merely the source of offline training labels: it remains part of the inference-time decision architecture.

Such a configuration directly illustrates an LLM acting as an interface to a specialized

decision module. ToolPoker targets both action quality and professional-style reasoning, whereas the present dissertation evaluates whether a standalone adapted model can predict reference decisions without querying a solver during inference. Differences in poker variants are also substantial because ToolPoker emphasizes two-player simplified and limit-betting environments rather than 6-max No-Limit Texas Hold'em (LIN et al., 2026).

GTO Wizard Benchmark moves evaluation away from handpicked isolated examples and toward complete heads-up No-Limit Texas Hold'em matches. Agents interact through a standardized interface and play against a strong reference agent, while the AIVAT variance reduction procedure improves the statistical efficiency of the resulting win-rate estimates (PROVOST et al., 2026). Zero-shot evaluations of several LLMs indicate progress in strategic reasoning but also a substantial remaining gap relative to the reference agent.

Its evaluation objective differs from PokerBench and from the present dissertation. PokerBench measures agreement with reference labels for selected 6-max states, whereas GTO Wizard Benchmark measures cumulative agent performance over sequential heads-up hands. Match-based evaluation captures error propagation and long-term interaction but does not isolate action and sizing errors as directly as a labeled state-level benchmark (ZHUANG et al., 2025; PROVOST et al., 2026).

PokerSkill explores another alternative by combining an LLM with a human-designed library of poker skills rather than model fine-tuning or inference-time solver queries. A deterministic context engine analyzes the current state, retrieves relevant fragments from a layered rule library, and uses them to constrain the actions considered by the language model. Evaluation against strong poker agents showed substantial reductions in losses relative to default-prompt LLM baselines (LI et al., 2026).

Unlike ToolPoker, PokerSkill does not obtain strategic calculations from an external solver during inference. Unlike PokerBench and SpinGPT, it also avoids poker-specific parameter training. Its performance instead depends on explicit expert knowledge encoded in the skill library and on a context engine that selects the relevant rules (LI et al., 2026). Consequently, PokerSkill represents a grounded rule-based architecture rather than either a standalone learned policy or a solver-backed assistant.

Progress across these studies reveals a sequence of increasingly specialized approaches. Prompt-only evaluations test poker knowledge already present in general-purpose models; fine-tuned systems adapt model parameters from human or solver-related data; benchmarks standardize either isolated decisions or complete matches; and augmented architectures supply external solvers or expert-designed skills during inference (GUPTA, 2023; HUANG et al., 2024; ZHUANG et al., 2025; MAUGIN; CAZENAVE, 2025; LIN et al., 2026; PROVOST et al., 2026; LI et al., 2026).

Reported results cannot be ranked through a single numerical metric because the reviewed studies address different poker variants, system roles, and evaluation units. Re-

stricted preflop analysis, multiplayer Texas Hold'em, 6-max NLHE states, three-player Spin & Go, simplified two-player games, Limit Hold'em, and complete heads-up no-limit matches involve different action spaces and strategic horizons. Table 1 therefore compares their game settings, methods, and evaluation emphases rather than presenting their results as scores on a common task.

Table 1 – Methodological comparison of closely related LLM-based poker studies.

Work	**Game setting**	**Method**	**Evaluation**
Gupta (2023)	9-player NLHE; preflop	Prompting of closed LLMs against reference charts	Isolated preflop action agreement
PokerGPT (2024)	Multiplayer Texas Hold'em	Fine-tuning on textual game records	Gameplay performance and computational efficiency
SpinGPT (2025)	3-player Spin & Go	SFT on expert decisions followed by reinforcement learning	Decision agreement and gameplay performance
ToolPoker (2026)	Heads-up Kuhn, Leduc, and Limit Hold'em	Behavior cloning and reinforcement learning with solver queries	Actions, gameplay, and explanations
GTO Wizard Benchmark (2026)	Heads-up No-Limit Texas Hold'em	Zero-shot evaluation of LLM agents against GTO Wizard AI	Standardized complete-match evaluation
PokerSkill (2026)	Heads-up No-Limit Texas Hold'em	Retrieval from an expert-designed poker skill library	Match performance against GTO Wizard AI
PokerBench (2025)	6-max NLHE; preflop and postflop	Few-shot prompting and SFT on textual decision states	Action accuracy, action–sizing agreement, and model matches
This dissertation	**6-max NLHE; preflop and postflop**	**Few-shot prompting and SFT on PokerBench-derived states**	**Action and sizing; error, context, and robustness analyses**

Table 1 shows that the reviewed studies assign different roles to language models and evaluate them at different levels. Some systems are assessed through complete gameplay, whereas PokerBench and the present dissertation evaluate predictions for isolated 6-max NLHE states.

PokerBench is the closest methodological reference because it also applies few-shot prompting and supervised fine-tuning to textualized preflop and postflop states. It evaluates categorical actions and complete action–sizing responses and further relates benchmark performance to matches between selected model checkpoints (ZHUANG et al., 2025). The present dissertation adopts this general formulation while placing greater emphasis on sizing-specific errors, contextual breakdowns, and sensitivity to selected experimental variations. Its distinction therefore lies in the granularity of the diagnostic analysis rather than in introducing the textual prediction task.

3.5 Positioning of This Work

Existing poker research spans specialized solvers, feature-based predictors, standalone LLMs, fine-tuned poker agents, solver-augmented assistants, expert-grounded systems, and complete-match benchmarks. Such approaches differ in what information is sup-

plied to the model, where strategic knowledge originates, whether specialized computation remains available during inference, and whether performance is measured for isolated decisions or sequential play (MORAVČÍK et al., 2017; BROWN; SANDHOLM, 2017a; CARNEIRO; LISBOA, 2018; ZHUANG et al., 2025; LIN et al., 2026; LI et al., 2026; PROVOST et al., 2026).

Within this landscape, the present dissertation is positioned as a controlled diagnostic study of open LLMs for isolated 6-max No-Limit Texas Hold'em decision prediction. Evaluated models receive textualized states and predict PokerBench-derived reference outputs without querying a solver during inference. Accordingly, the study does not attempt to reproduce the search, equilibrium computation, opponent adaptation, or sequential decision process of full poker agents.

Mapping textual poker states to reference decisions is not unique to this dissertation. PokerBench previously introduced that formulation and evaluated prompting and fine-tuning on preflop and postflop examples (ZHUANG et al., 2025). The present work builds on that foundation through a controlled comparison of representative open models from multiple families, repeated few-shot evaluations, supervised adaptation of the selected model, and a more granular analysis of prediction errors.

A further distinction lies in the diagnostic treatment of prediction quality. Categorical action agreement remains the primary outcome, while numerical sizing is evaluated through continuous joint and conditional proportional measures. Results are additionally analyzed across action classes, table positions, postflop streets, and selected robustness conditions. Such analyses are intended to characterize where agreement with the benchmark improves and where errors remain concentrated, rather than to infer complete poker-playing competence.

Natural-language explanation is deliberately excluded from the empirical claims. ToolPoker evaluates reasoning traces and combines the LLM with external solver outputs, whereas PokerSkill grounds decisions through an expert-designed skill library (LIN et al., 2026; LI et al., 2026). The dataset adopted here contains action and sizing references but no validated rationales. Explanation quality, pedagogical usefulness, and solver-assisted consultation therefore remain outside the evaluation protocol.

Overall, the contribution lies neither in introducing the first textual poker benchmark nor in claiming that LLMs replace specialized solving systems. Its scope is a systematic analysis of how open LLMs predict PokerBench-derived reference decisions, how supervised domain adaptation changes that behavior, and whether gains in action agreement are accompanied by adequate numerical sizing and consistent performance across selected experimental variations.

Chapter 4

Problem Formulation

Poker decision recommendation is formalized as a supervised prediction problem over structured textual inputs. The objective is to map an observable poker state, represented in text, to a strategic decision composed of a categorical action and, when applicable, a numerical size. Rather than modeling complete matches or full game-tree search, the formulation focuses on individual decision points extracted from No-Limit Texas Hold'em hands.

The following sections define the observable state, the output representation, the distinction between preflop and postflop prediction tasks, and the supervised objective used to adapt language models to this domain.

4.1 Observable Poker State

Let s denote the observable state associated with the acting player at a decision point. Since poker is an imperfect-information game, s does not represent the full underlying game state. It contains only the information available to the acting player at that point in the hand.

In the present formulation, s includes the private, public, contextual, and sequential components represented in the adopted decision dataset. A compact representation is

$$s = (h, b, p, r, q, \alpha), \tag{10}$$

where h denotes the acting player's private hole cards, b denotes the public board cards when available, p denotes the acting player's position, r denotes the current round or street, q denotes numerical quantities such as stack sizes, pot size, and amounts committed during the betting sequence, and α denotes the sequence of previous actions.

Rather than receiving s as a fixed feature vector, the model receives a textualized version of the state. A textualization function $\tau(\cdot)$ maps the structured state to a prompt-like representation:

$$x = \tau(s). \tag{11}$$

The prediction task is therefore defined over structured text rather than over raw logs or manually engineered tabular feature vectors.

4.2 Decision Output Representation

Each observable state is associated with a poker decision. Let d denote the target decision for state s. The output is represented as

$$d = (a, z), \tag{12}$$

where a is the categorical action and z is the numerical size associated with the action when such a size is required.

The general action vocabulary is denoted by

$$\mathcal{A} = \{\texttt{fold}, \texttt{call}, \texttt{check}, \texttt{bet}, \texttt{raise}\}. \tag{13}$$

For a given state s, prediction is constrained to the subset of actions available in that state, denoted by $\mathcal{A}(s) \subseteq \mathcal{A}$.

The subset of aggressive actions is defined as

$$\mathcal{A}_{\text{agg}} = \{\texttt{bet}, \texttt{raise}\}. \tag{14}$$

If $a \in \mathcal{A}_{agg}$, then $z \in \mathbb{R}_{>0}$ represents the numerical size of the bet or raise, expressed in big blinds. If $a \notin \mathcal{A}_{agg}$, no numerical size is required.

This decomposition reflects two different prediction challenges. The action component is discrete and constrained by the state, whereas the sizing component requires numerical calibration for aggressive decisions.

4.3 Preflop and Postflop Prediction Tasks

The supervised prediction problem is evaluated separately in preflop and postflop settings. This separation reflects not only a poker distinction, but also a difference in contextual complexity and input structure.

In the preflop setting, the input state is defined before any community cards have been revealed. The textual representation typically includes the acting player's hole cards, position, blind structure, stack-related quantities, and the preflop action history. The corresponding prediction function can be written as

$$f_\theta^{\text{pre}}(x) \to \hat{d}^{\text{pre}} = (\hat{a}^{\text{pre}}, \hat{z}^{\text{pre}}). \tag{15}$$

In the postflop setting, the input additionally contains public board cards, the current street, pot-related quantities, and a richer action history. The corresponding prediction function is

$$f_\theta^{\text{post}}(x) \to \hat{d}^{\text{post}} = (\hat{a}^{\text{post}}, \hat{z}^{\text{post}}). \tag{16}$$

Separate evaluation is used because the two stages correspond to different decision distributions. Within the adopted representation, preflop inputs are generally shorter and more regular, whereas postflop inputs contain public boards, additional streets, and more heterogeneous action histories.

4.4 Prediction and Adaptation Objectives

After defining the observable state and the decision output, the learning problem can be expressed as supervised prediction from textual poker states to reference decisions. Each training example pairs a text-encoded decision state with the corresponding target action and, when applicable, its numerical size. The supervised dataset is represented as

$$\mathcal{D} = \{(x_i, y_i)\}_{i=1}^{N}, \tag{17}$$

where $x_i = \tau(s_i)$ is the textualized state and y_i is the target output sequence encoding the structured decision $d_i = (a_i, z_i)$.

In the few-shot setting, model parameters remain fixed, and prediction is conditioned on an instruction together with a small number of in-context examples. No optimization is performed over $\mathcal{D}$.

In the supervised fine-tuning setting, trainable model parameters are adapted to the poker decision distribution. For an autoregressive language model, the objective is to maximize the conditional probability of the target output sequence given the input prompt. Equivalently, training minimizes the negative log-likelihood of the target tokens:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{i=1}^{N} \sum_{t=1}^{|y_i|} \log P_\theta(y_{i,t} \mid x_i, y_{i,<t}). \tag{18}$$

Here, $y_{i,t}$ denotes the t-th token of the target output, and $y_{i,<t}$ denotes the preceding target tokens. In instruction-style fine-tuning, optimization is typically applied to the target response tokens conditioned on the prompt.

Although the learning objective is autoregressive text generation, the predicted output is interpreted as a structured decision. The empirical evaluation therefore separates categorical action prediction from numerical sizing prediction. Such a separation is necessary because the two components exhibit different error patterns and levels of difficulty.

This formulation establishes the abstract task studied in the dissertation: predicting a structured poker decision from a textualized observable state. Chapter 5 describes the

dataset, prompt construction, adaptation regimes, inference pipeline, and metrics used to operationalize this task.

Chapter 5

Materials and Methods

Following the problem formulation introduced in Chapter 4, the methodological framework defines how open Large Language Models are evaluated as text-based decision predictors for 6-max No-Limit Texas Hold'em. Each example pairs an observable poker state, represented as text, with a reference decision containing a categorical action and, when applicable, a numerical bet size.

Two central distinctions organize the methodology. The first concerns adaptation regimes: few-shot prompting evaluates instruction-tuned models with in-context examples and no parameter updates, whereas supervised fine-tuning adapts the selected model to poker-specific decision data. The second concerns decision components: categorical action selection and numerical sizing are evaluated separately because they correspond to different prediction challenges. By separating these dimensions, the study measures agreement with reference decisions while diagnosing whether errors arise from categorical action selection, numerical sizing, or their interaction.

5.1 Methodological Overview

Poker decision recommendation is operationalized as a structured prediction task over textualized decision states. Rather than relying on unconstrained free-form generation for the entire decision, the evaluation protocol separates the categorical and numerical components of no-limit poker actions. Candidate actions are scored through log-probabilities over available actions, while numerical sizes are generated and parsed only when aggressive actions are involved. This hybrid design provides a controlled basis for comparing prompting and fine-tuning while preserving the action-and-sizing structure of the task.

The experimental procedure is organized into five stages. First, PokerBench-derived decision examples are represented as structured instruction-output pairs. Second, representative open instruction-tuned LLMs are assessed under controlled few-shot prompting protocols. Third, the strongest few-shot model is specialized through supervised fine-tuning using stage-specific training data. Fourth, predictions are obtained through the

hybrid action-and-sizing pipeline. Finally, model behavior is analyzed through aggregate metrics, statistical validation, confusion matrices, contextual breakdowns, sizing diagnostics, and complementary robustness and ablation experiments.

Figure 3 summarizes the proposed methodology. Starting from textual poker decision states, the pipeline supports two adaptation regimes: few-shot prompting with fixed model parameters and supervised fine-tuning of the selected model. Predictions are then evaluated through the hybrid action-and-sizing protocol and analyzed using both aggregate metrics and diagnostic analyses.

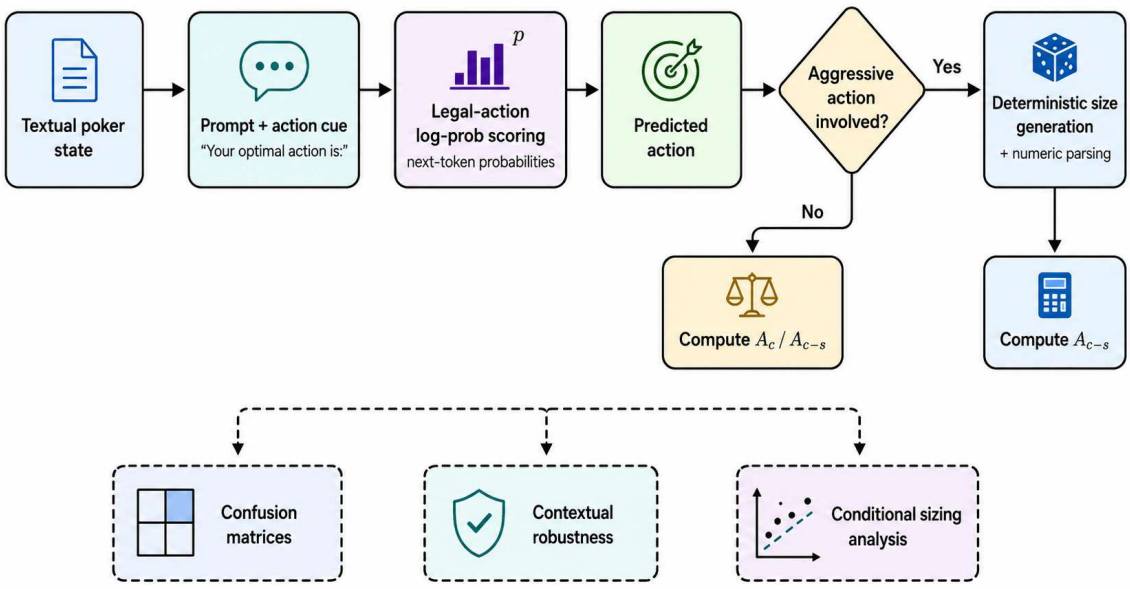

Figure 3 – Overview of the experimental pipeline for few-shot prompting, supervised fine-tuning, and hybrid action-and-sizing evaluation.

Preflop and postflop decisions are treated separately because they instantiate different levels of contextual complexity within the same prediction framework. Preflop states are more regular and contain no public board cards, whereas postflop states include board information, street information, pot-related quantities, and richer action histories. This stage-specific design allows the methodology to test whether model behavior changes as the decision state becomes more context-dependent.

The chapter is organized according to the main methodological components of the study. The dataset and textual representation are presented first, followed by the evaluated models and the few-shot and supervised fine-tuning protocols. Subsequent sections describe the hybrid action-and-sizing pipeline, evaluation metrics, statistical validation, and reproducibility procedures.

5.2 Dataset

Decision examples are derived from PokerBench, a benchmark designed for evaluating language models in No-Limit Texas Hold'em decision-making (ZHUANG et al., 2025). PokerBench provides separate preflop and postflop datasets, with examples represented as instruction-output pairs. The instruction describes the decision state in structured natural language, while the output specifies the reference strategic decision.

Preflop data contains approximately 63,000 training examples and 1,000 evaluation examples. Postflop data contains approximately 500,000 training examples and 10,000 evaluation examples. The original PokerBench train–test partitions are preserved throughout the experiments.

Each example corresponds to an individual decision point. Depending on the stage of the hand, the instruction may include blind structure, player positions, the acting player's hole cards, stack sizes, pot size, public board cards, current street, and previous actions. Outputs contain the reference decision. Non-aggressive outputs contain only an action label, such as `fold`, `call`, or `check`. Aggressive outputs contain an action label followed by a numerical size, such as `bet 10.0` or `raise 12.5`.

All decision instances follow the benchmark's fixed-stack configuration, in which players begin with stacks of 100 big blinds. Accordingly, valid bet and raise sizings are positive values no greater than 100 BB.

Reference labels are treated as equilibrium-oriented strategic decisions. In the PokerBench construction, preflop decisions are based on GTO Wizard strategies, while postflop decisions are generated from WASM-Postflop solver outputs (GTO Wizard, 2025; WASM Postflop, 2023; ZHUANG et al., 2025). Thus, the objective is not to imitate arbitrary human behavior, but to evaluate whether LLMs can predict PokerBench-derived, equilibrium-oriented reference decisions from textual descriptions of poker states.

PokerBench also applies balancing and filtering choices to make the evaluation more informative. In particular, fold-heavy regions are downsampled so that a trivial strategy, such as folding most hands, does not obtain artificially high accuracy (ZHUANG et al., 2025). Such balancing is important because many weak poker hands are correctly folded, and an unbalanced benchmark could overestimate the performance of models that overpredict passive actions.

The benchmark does not enumerate the complete 6-max NLHE state space. Instead, its filtering, balancing, and scenario-selection procedures provide a controlled collection of decision states for comparative evaluation.

Figure 4 summarizes the action distributions of the PokerBench-derived training and benchmark sets used in this study. Training sets are not uniformly distributed, especially in the preflop setting, where fold and call are much more frequent than raise and check. By contrast, benchmark subsets are substantially more balanced across action classes than the corresponding training sets. Separating these distributions is relevant because

the training distribution influences what the model learns, whereas the benchmark distribution provides a more controlled basis for comparing model behavior across actions and game stages.

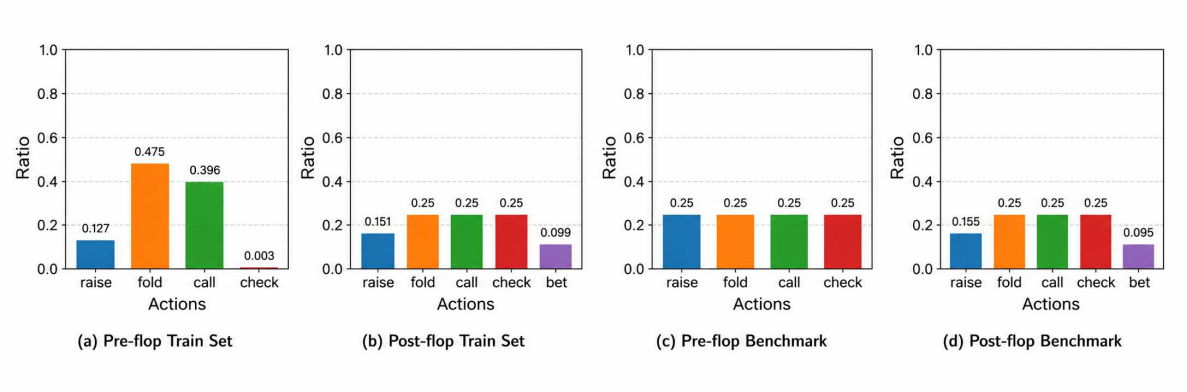

Figure 4 – Action-class distributions in the PokerBench-derived training and test sets for preflop and postflop decisions. Adapted from (ZHUANG et al., 2025).

PokerBench also contains postflop examples with varied public board contexts. Such variation matters because postflop decision-making depends not only on private cards and previous actions, but also on the interaction between the acting player's hand and the public board. Therefore, the postflop setting provides a broader test of contextual generalization than repeated evaluation on a narrow set of fixed board situations.

5.3 Textual Representation and Prompt Format

All decision states are represented as text. Such a design allows LLMs to process poker situations without requiring a specialized poker engine or explicit game-tree search at inference time. The structured textual representation functions as the model input and contains the observable information available to the acting player.

A typical prompt includes the acting player's position, private cards, stack-related quantities, pot size, previous actions, and, for postflop examples, the public board and current street. Retaining the within-hand action history is important because earlier actions alter numerical quantities and constrain the hidden holdings and strategies compatible with the current observation. Numerical quantities such as stacks, pots, bets, and raises are normalized in big blinds. Normalizing values in this way reduces dependence on the monetary stakes of a specific table and makes examples comparable across different blind levels.

Figure 5 makes the textual representation more concrete by illustrating how a poker decision state is converted into an instruction-output pair. Unlike the schematic representation introduced in Chapter 2, the goal here is not to show the visual structure of the table, but rather the natural-language encoding presented to the language model. The

example highlights how the main components of the decision state — such as position, private cards, public board cards, pot and stack information, and betting history — are serialized into an instruction, while the target output contains the recommended action and, when applicable, its numerical size. The same representation is used in both few-shot prompting and supervised fine-tuning.

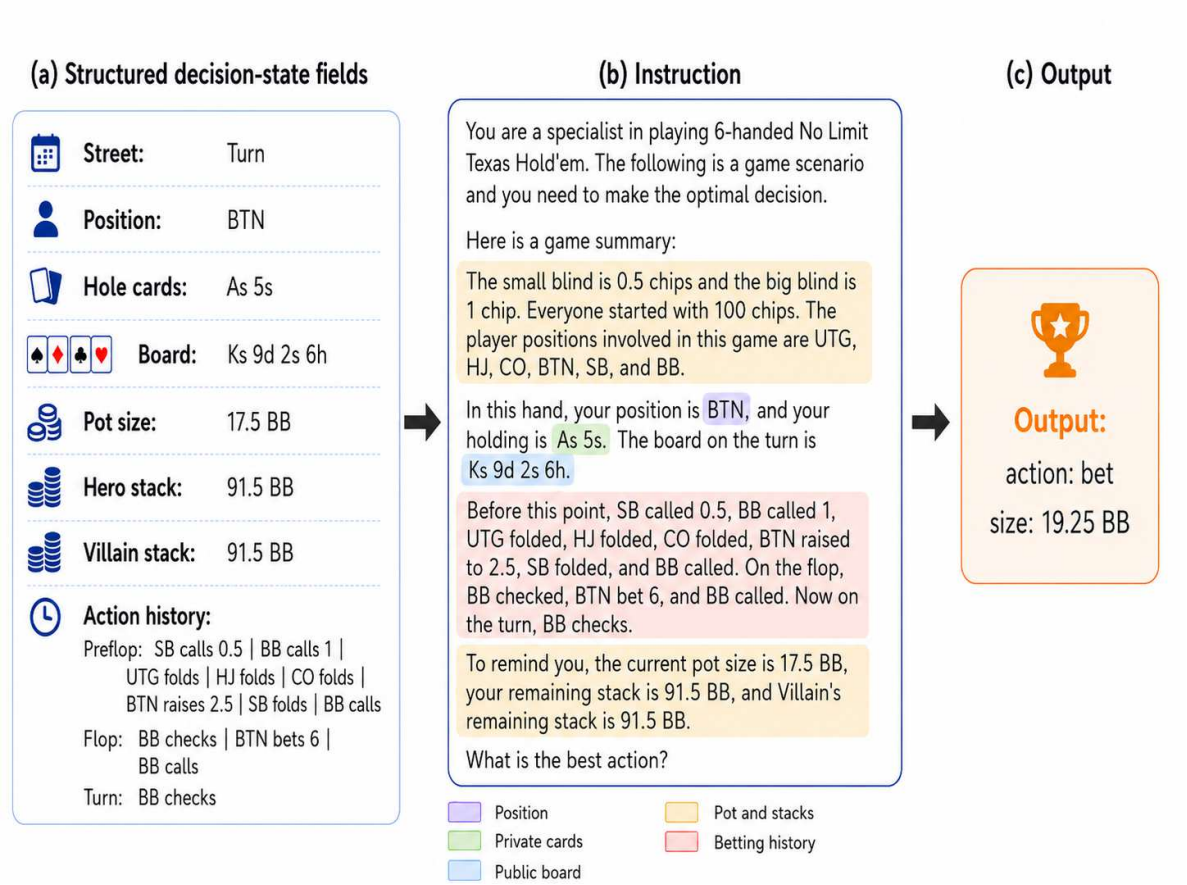

Figure 5 – Example of the process used to convert structured poker decision-state fields into a textual model instruction and a structured action-and-size output.

By expressing the decision state in this form, the problem becomes compatible with the standard input-output interface of Large Language Models.

Expected outputs are intentionally concise. The model is instructed to produce only the recommended decision, without explanation. Restricting the response format reduces ambiguity during evaluation and makes it possible to parse the output into an action component and, when applicable, a numerical sizing component. Examples of valid outputs include:

- ❑ `fold`
- ❑ `call`
- ❑ `check`

❑ `bet 7.5`

❑ `raise 12.0`

Such an output format reflects the structure of no-limit poker decisions. Some decisions require only a categorical action, while aggressive decisions also require a numerical size. For this reason, the evaluation methodology distinguishes Action Accuracy from Action-and-Sizing Accuracy.

With the data source and textual input-output format defined, the methodology next specifies the model families evaluated under this framework.

5.4 Evaluated Models

Building on the open LLM families introduced in Chapter 2, the few-shot evaluation includes six instruction-tuned models selected to provide diversity in model family, parameter scale, and feasibility for local inference. The evaluated models are Llama 3.2-3B, Gemma 2-9B, Qwen3-4B, Mistral-7B, Llama 3-8B, and Qwen3-14B (GRATTAFIORI et al., 2024; Gemma Team, 2024; YANG et al., 2025; JIANG et al., 2023).

The few-shot stage evaluates how instruction-tuned models behave when poker decision prediction is performed only from in-context examples, without parameter updates. This stage provides a controlled baseline for assessing whether general instruction-following ability is sufficient to recover strategic poker decisions from textualized states.

The model selected for supervised fine-tuning was the one that obtained the highest few-shot Action Accuracy in both decision stages. Under this predefined criterion, Qwen3-14B was selected for the subsequent preflop and postflop experiments. Separate preflop and postflop adaptations were used because the two stages differ in input structure, action distribution, and contextual complexity.

5.5 Few-Shot Prompting Protocol

Under few-shot prompting, the model receives a small number of labeled examples before the query example. These demonstrations show the expected mapping between a poker situation and the corresponding decision output. Model parameters are not updated; performance instead depends on the pretrained model, the instruction format, and the selected in-context examples.

The final protocol uses one demonstration for each available action class. Preflop prompts therefore contain four demonstrations, representing `fold`, `call`, `check`, and `raise`, whereas postflop prompts contain five, additionally representing `bet`. This construction provides balanced coverage of the output classes while keeping the prompt compact and comparable across evaluated models.

Preliminary pilot evaluations considered larger numbers of demonstrations per action class. These configurations did not produce consistent improvements and substantially increased prompt length and inference cost. The final protocol therefore retained one demonstration for each available action class. These pilot observations were used only to define the experimental configuration and are not reported as a separate comparative result.

Few-shot performance can be sensitive to the examples selected for the prompt. To account for this variability, evaluation is repeated with different random seeds for in-context example sampling from the training set. Preflop few-shot experiments are evaluated across ten seeds, while postflop few-shot experiments are evaluated across five seeds. Reported few-shot results are therefore presented as means and standard deviations across prompt seeds.

Variation across prompt seeds measures sensitivity to demonstration selection and is not interpreted as a direct test of model overfitting.

Such a protocol separates prompt variability from supervised adaptation. Few-shot results measure the ability of instruction-tuned models to use a small number of examples at inference time, whereas fine-tuned results measure the effect of adapting model parameters to the poker decision distribution.

5.6 Supervised Fine-Tuning Protocol

Supervised fine-tuning adapts Qwen3-14B to the poker decision format. Unlike few-shot prompting, supervised adaptation updates trainable parameters using labeled examples from the target domain. The objective is to learn the mapping between structured poker states and reference decisions, reducing dependence on in-context demonstrations at inference time.

Parameter-efficient adaptation is performed with LoRA and QLoRA (HU et al., 2022; DETTMERS et al., 2023). LoRA introduces low-rank trainable matrices into selected model layers, allowing domain adaptation without updating all original parameters. QLoRA further reduces memory requirements by combining low-rank adaptation with quantized model weights. Together, these techniques make fine-tuning Qwen3-14B feasible under the available GPU memory.

Separate adapters are trained for preflop and postflop examples. The preflop adapter is trained on the preflop training partition and evaluated on the corresponding test set, whereas the postflop adapter uses the respective postflop partitions. This separation avoids mixing decision distributions with substantially different contextual structures and supports stage-specific analysis.

Training targets preserve the complete decision response. Non-aggressive examples contain only the categorical action label, whereas aggressive examples contain the action

followed by its numerical sizing, such as `raise 14.0`. Aggressive targets therefore provide direct supervision for both action selection and numerical sizing.

The main hyperparameters are summarized in Table 2. Learning rate, dropout, warmup ratio, and number of epochs differ between preflop and postflop adapters because the datasets have different sizes and input distributions.

Table 2 – Supervised fine-tuning hyperparameters for the preflop and postflop Qwen3-14B LoRA/QLoRA adapters.

Hyperparameter	**Preflop**	**Postflop**
Quantization	4-bit NF4	4-bit NF4
Compute dtype	bfloat16	bfloat16
LoRA rank	16	16
LoRA alpha	32	32
LoRA dropout	0.05	0.10
Learning rate	1×10^{-4}	2×10^{-4}
Scheduler	cosine	cosine
Warmup ratio	0.03	0.05
Per-device batch size	2	4
Gradient accumulation	8	4
Effective batch size	16	16
Maximum sequence length	1024	1024
Epochs	2	3

A limited number of epochs and LoRA dropout are used as regularization measures. Unlike few-shot prompting, fine-tuned inference does not involve random demonstration sampling. Each final checkpoint is evaluated once under fixed deterministic settings. Variability across independent fine-tuning runs is not estimated.

Once prompting and fine-tuning regimes have been specified, the methodology turns to how model outputs are transformed into comparable categorical and numerical predictions.

5.7 Hybrid Evaluation Pipeline

The hybrid evaluation pipeline was designed to reflect the structure of no-limit poker decisions. Each decision may contain both a categorical action and a numerical size, but these two components create different evaluation challenges. Free-form generation can introduce formatting variation, invalid outputs, or parsing errors, especially when the goal is to evaluate action classification. For this reason, action prediction and sizing prediction are evaluated through separate but connected procedures.

For action prediction, the model is not evaluated through unconstrained free-form generation. Instead, the prompt ends with the action cue, and the available candidate

actions are scored using log-probabilities. The predicted action is the available action with the highest score:

$$\hat{a} = \arg\max_{a \in \mathcal{A}(s)} \log P_\theta(a \mid x), \tag{19}$$

where x is the textual representation of the decision state and $\mathcal{A}(s)$ is the set of actions available in state s. In implementation, the candidate action tokens are scored at the decision point after the prompt cue. Scoring candidate actions in this way reduces dependence on decoding randomness and avoids treating invalid or verbose generated text as action predictions.

Sizing is evaluated separately. When the reference action or the predicted action is aggressive, that is, `bet` or `raise`, a deterministic generation step is performed. Triggering generation in either case ensures that a numerical output is available for diagnostic inspection whenever sizing may be relevant. Numerical success is nevertheless credited only under the conditions defined by the evaluation metrics. The first numerical value parsed from the generated output is interpreted as the predicted size in big blinds. Deterministic decoding is used to reduce variability and make sizing evaluation reproducible.

The hybrid pipeline limits computational cost relative to generating a complete response for every test example. Action prediction is performed through a single scoring step over candidate actions, avoiding full-sequence generation for every example. Full deterministic generation is only used when sizing must be evaluated, that is, when the reference or predicted action is aggressive. This restriction is important for scaling evaluation to thousands of preflop and postflop examples under limited GPU memory.

Structurally, the hybrid design mirrors the task itself. Choosing the correct action and choosing the correct size are related but distinct problems. A model may correctly recognize that an aggressive action is appropriate while still choosing an inaccurate size. Conversely, a model may produce a plausible numerical value while selecting the wrong action category. Separating these components allows the analysis to identify whether performance gains come mainly from improved action selection, improved sizing, or both.

Given the hybrid prediction procedure, the evaluation metrics must account for both the categorical action and the numerical size associated with aggressive decisions.

5.8 Evaluation Metrics

Action Accuracy (Ac) is the primary metric, measuring the proportion of examples for which the predicted categorical action matches the reference action. Let a_i be the reference action for example i and $\hat{a}_i$ be the predicted action. For a test set of N examples, Action Accuracy is defined as

$$Ac = \frac{1}{N} \sum_{i=1}^{N} \mathbb{I}[\hat{a}_i = a_i]. \tag{20}$$

Accuracy is a standard measure for classification tasks, but it may need to be complemented by task-specific diagnostics when outputs contain heterogeneous components (SOKOLOVA; LAPALME, 2009).

As a methodological contribution, we develop a continuous formulation of Action-and-Sizing Accuracy (Ac-s) to jointly evaluate categorical action selection and, when required, numerical sizing. Correct non-aggressive actions receive a per-example score of one, whereas incorrect categorical actions receive zero.

For a correctly predicted bet or raise with valid positive sizings, proportional numerical agreement is measured by

$$r_i = \exp\left(-\left|\log\left(\frac{\hat{z}_i}{z_i}\right)\right|\right), \tag{21}$$

where z_i denotes the reference sizing and $\hat{z}_i$ denotes the predicted sizing for example i. The same score can be written more intuitively as

$$r_i = \frac{\min(z_i, \hat{z}_i)}{\max(z_i, \hat{z}_i)}. \tag{22}$$

In other words, the smaller sizing is divided by the larger one. The resulting score lies in $(0, 1]$: an exact prediction receives 1, and the score gradually decreases as the predicted sizing moves proportionally farther from the reference in big blinds.

For example, predicting 24 BB for a 20 BB reference receives the same score as predicting 6 BB for a 5 BB reference. Both predictions are 20% larger than their respective references and yield $r_i = 5/6 \approx 0.833$, even though their absolute errors are 4 BB and 1 BB.

Proportional evaluation is suitable for no-limit poker because the relevance of an absolute sizing error depends on the magnitude of the reference action. Ratio-based comparison is also symmetric: predicting twice the reference sizing and predicting half of it produce the same score.

The per-example joint score is defined as

$$q_i = \begin{cases} 0, & \hat{a}_i \neq a_i, \\ 1, & \hat{a}_i = a_i \text{ and } a_i \notin \{\texttt{bet}, \texttt{raise}\}, \\ r_i, & \hat{a}_i = a_i,\ a_i \in \{\texttt{bet}, \texttt{raise}\},\ \text{and both sizings are valid}, \\ 0, & \hat{a}_i = a_i,\ a_i \in \{\texttt{bet}, \texttt{raise}\},\ \text{and the predicted sizing is invalid}. \end{cases} \tag{23}$$

A predicted sizing is valid when it is finite, positive, and no greater than 100 BB. Action-and-Sizing Accuracy is obtained by averaging the per-example scores:

$$Ac\text{-}s = \frac{1}{N}\sum_{i=1}^{N} q_i. \tag{24}$$

Ac-s preserves full credit for correct decisions that do not require sizing, while correct aggressive decisions receive continuous credit according to their proportional agreement with the reference sizing. The metric therefore summarizes categorical and numerical agreement within a single score over the complete evaluation set.

Numerical sizing is additionally examined through a conditional relative sizing score. This diagnostic uses the same proportional component r_i employed by Ac-s, but is restricted to aggressive reference decisions for which the predicted action is correct and both the reference and predicted sizings are valid.

Let $\mathcal{C}$ denote the set of eligible examples. The conditional score is defined as

$$R_{\text{cond}} = \frac{1}{|\mathcal{C}|}\sum_{i\in\mathcal{C}} r_i. \tag{25}$$

While Ac-s evaluates the complete decision pipeline over all test instances, the conditional score isolates sizing quality after the categorical action has been correctly identified. Reporting the number of eligible examples together with the conditional score makes its denominator explicit and supports direct comparison between preflop and postflop sizing performance.

Having defined the prediction and scoring protocol, the analysis next evaluates whether the observed fine-tuning gains are statistically supported.

5.9 Statistical Validation

Statistical validation assesses whether the gains from supervised fine-tuning are robust rather than merely numerical fluctuations. Qwen3-14B is the focus of this comparison because it is the model selected for supervised fine-tuning after the few-shot evaluation. Bootstrap resampling is used to estimate uncertainty in the improvement in Action Accuracy (EFRON, 1979; EFRON; TIBSHIRANI, 1993). McNemar's test is used for paired comparison between few-shot and fine-tuned predictions, since both methods are evaluated on the same test examples (MCNEMAR, 1947; AGRESTI, 2013).

Odds ratios and confidence intervals are also reported. Together, bootstrap confidence intervals, McNemar's test, and odds ratios provide complementary evidence about the magnitude and reliability of the fine-tuning gains.

5.10 Reproducibility Procedures

Reproducibility procedures close the methodological description by specifying how the experimental protocol can be reconstructed and inspected beyond aggregate scores. The

original train-test separation is preserved for both preflop and postflop data. Few-shot experiments use fixed random seeds for in-context example sampling, allowing the same prompt configurations to be reconstructed. Fine-tuned models are evaluated through fixed inference settings, and deterministic decoding is used for sizing generation.

During evaluation, the pipeline records aggregate metrics, action predictions, sizing predictions, parsing outcomes, confusion matrices, and contextual breakdowns whenever applicable. Runtime statistics, including the number of log-probability calls, generation calls, and average processing time per sample, are also recorded whenever available. Recorded outputs allow results to be inspected beyond aggregate accuracy and support later analyses of class-level errors, position-level robustness, street-level robustness, sizing behavior, and computational cost.

Hybrid action-and-sizing evaluation also contributes to reproducibility by reducing dependence on free-form generation for action classification. Candidate actions are scored directly through next-token log-probabilities, while numerical sizes are generated only when needed. As a result, the protocol provides a more stable basis for comparing few-shot prompting and supervised fine-tuning in poker decision prediction.

The experiments were organized through versioned scripts for data preprocessing, prompt construction, model evaluation, metric computation, and result analysis. These scripts preserve the main experimental decisions used in the dissertation, including the original train–test splits, fixed few-shot seeds, adopted inference settings, and the hybrid action-and-sizing evaluation protocol. The preflop and postflop fine-tuning scripts, together with the hybrid evaluator, are publicly available at <https://github.com/lucasodnz/PokerLLM-MSc-Dissertation>. Full PokerBench datasets, trained adapters, model weights, and large prediction files are not redistributed.

Chapter 6

Results and Discussion

Empirical analysis covers preflop and postflop decision prediction in 6-max No-Limit Texas Hold'em. Following the protocol defined in Chapter 5, six open instruction-tuned LLMs are first compared under repeated few-shot prompting. Qwen3-14B, the strongest model in both stages, is subsequently adapted through supervised fine-tuning and examined through aggregate performance, action-level errors, statistical validation, contextual breakdowns, sizing behavior, and complementary robustness analyses.

Interpretation distinguishes categorical action selection from the numerical sizing required by bets and raises. Action Accuracy (Ac) measures agreement with the reference action. Continuous Action-and-Sizing Accuracy (Ac-s) evaluates the complete decision by combining action correctness with proportional sizing agreement. Conditional sizing analysis then isolates numerical quality among valid aggressive decisions for which the reference action was correctly predicted.

Results are organized around four questions: how open LLMs differ under few-shot prompting; how much supervised adaptation improves the strongest candidate; whether such gains extend across actions and strategic contexts; and how accurately numerical sizings are recovered. Additional analyses in Appendix C examine sensitivity to evaluation-set size, prompt formulation, action-prediction method, sizing instructions, and decoding temperature.

6.1 Main Performance Across Few-Shot Models

Table 3 compares few-shot and fine-tuned performance on preflop and postflop decisions. Few-shot values report means and standard deviations across ten preflop seeds and five postflop seeds. Fine-tuned results correspond to one fixed evaluation run for each stage.

Qwen3-14B achieves the strongest few-shot Action Accuracy in both tasks. Preflop performance reaches 76.1%, followed by Llama 3-8B at 66.0% and Qwen3-4B at 60.6%.

Continuous Ac-s follows the same overall ranking, with Qwen3-14B reaching 71.4%. Such results motivate its selection for supervised adaptation.

Table 3 – Few-shot and fine-tuned performance of open LLMs on preflop and postflop action-and-sizing prediction.

Setting	Model	Version	Preflop			Postflop		
			Ac (%)	Ac-s (%)	Runs	Ac (%)	Ac-s (%)	Runs
Few-shot	Llama	3.2-3B	28.6 ± 4.6	28.5 ± 4.6	10	36.2 ± 2.1	34.9 ± 2.6	5
Few-shot	Gemma	2-9B	53.8 ± 5.1	52.9 ± 5.4	10	42.0 ± 0.9	39.1 ± 0.9	5
Few-shot	Qwen	3-4B	60.6 ± 7.2	58.7 ± 6.3	10	36.1 ± 3.2	35.5 ± 3.4	5
Few-shot	Mistral	7B	52.3 ± 4.3	52.0 ± 4.5	10	48.4 ± 1.4	47.2 ± 1.7	5
Few-shot	Llama	3-8B	66.0 ± 2.4	64.1 ± 2.1	10	51.1 ± 1.8	49.3 ± 2.5	5
Few-shot	Qwen	3-14B	76.1 ± 2.6	71.4 ± 2.6	10	51.5 ± 2.4	50.3 ± 2.5	5
Fine-tuned	**Qwen**	**3-14B**	**93.3**	**93.2**	**1**	**91.8**	**91.6**	**1**

Model performance varies substantially across families, and parameter count alone does not explain the observed ranking. Architectures also differ in tokenizer behavior, pretraining data, instruction tuning, and numerical processing. Qwen3-14B is therefore selected because it is empirically strongest under the adopted protocol, rather than because the comparison establishes a general scaling relationship.

Few-shot prompting provides meaningful preflop performance but remains less reliable in the richer postflop setting. Public board cards, pot-related quantities, multiple betting streets, and longer action histories expand the contextual space after the flop. Repeated in-context examples alone consequently recover only part of the reference decision mapping, motivating direct supervised adaptation.

Available actions are scored through next-token log-probabilities, while numerical sizings are generated deterministically when required. Accordingly, Table 3 distinguishes categorical agreement from continuous joint action-and-sizing performance. Close Ac and Ac-s values after fine-tuning indicate that the improvement extends to both components of the predicted decision.

6.2 Effect of Supervised Fine-Tuning

Supervised adaptation substantially improves Qwen3-14B in both decision stages. Preflop Action Accuracy increases from 76.1% under few-shot prompting to 93.3% after fine-tuning, corresponding to a gain of 17.2 percentage points. Continuous Ac-s rises from 71.4% to 93.2%, showing that the improvement extends beyond categorical action selection to the complete action-and-sizing decision.

Near-equality between preflop Ac and Ac-s indicates that correctly predicted aggressive actions are generally accompanied by sizings with high proportional agreement. Section 6.6 examines this numerical component independently of action-selection errors.

Postflop fine-tuning raises Action Accuracy from 51.5% under few-shot prompting to 91.8%, corresponding to a gain of 40.3 percentage points. Continuous Ac-s increases from 50.3% to 91.6%. Proximity between both fine-tuned metrics indicates that correctly predicted postflop bets and raises are also generally accompanied by proportionally accurate sizings.

Gains in both stages show that supervised adaptation recovers domain-specific decision patterns that are not consistently elicited through a small number of in-context examples. Preflop and postflop remain distinct categorical prediction settings: the former contains more regular action structures, whereas the latter combines public cards, multiple streets, pot-related quantities, and longer betting histories. Once an aggressive action is correctly selected, however, both stage-specific adapters recover its numerical parameterization with high proportional agreement.

6.3 Action-Level Error Analysis

Aggregate metrics do not show whether errors are concentrated in specific action classes. Figure 6 presents row-normalized confusion matrices for Qwen3-14B under few-shot prompting and supervised fine-tuning in both decision stages.

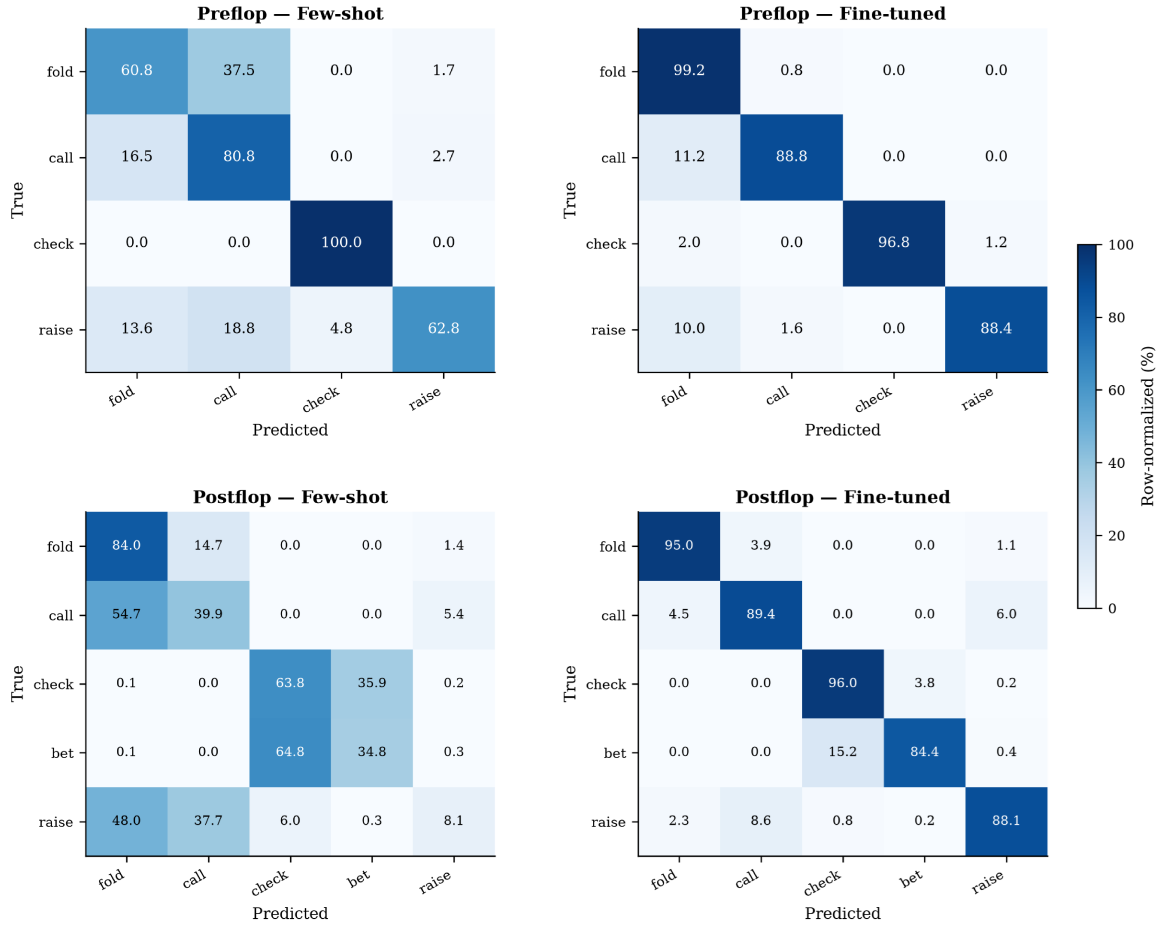

Figure 6 – Row-normalized confusion matrices for Qwen3-14B under few-shot prompting and supervised fine-tuning in preflop and postflop decision prediction.

Few-shot preflop predictions show substantial confusion between aggressive and passive alternatives. Only 60.8% of reference folds and 62.8% of reference raises are recovered correctly. Folds are frequently predicted as calls, while raises are often mapped to folds or calls. Calls are more stable, reaching 80.8% recall, and checks are recovered perfectly within the restricted subset of states in which that action is available.

Supervised adaptation makes the preflop matrix considerably more diagonal. Recall reaches 99.2% for fold, 88.8% for call, 96.8% for check, and 88.4% for raise. Remaining errors are concentrated mainly in calls predicted as folds and raises predicted as folds. Improvement therefore extends across all action classes rather than being driven by a single dominant category.

Postflop few-shot predictions exhibit a stronger passive bias. More than half of reference calls are predicted as folds, while most bets are predicted as checks. Reference raises are also commonly mapped to folds or calls. Such behavior indicates difficulty in recovering aggressive decisions from states containing public cards, multiple streets, pot-related quantities, and longer betting histories.

Fine-tuning sharply reduces these postflop confusions. Recall reaches 95.0% for fold,

89.4% for call, 96.0% for check, 84.4% for bet, and 88.1% for raise. Bet–check confusion remains the main residual pattern, with 15.2% of reference bets predicted as checks. Even so, the postflop matrix becomes strongly diagonal, confirming that supervised adaptation reduces both passive bias and aggressive-action confusion.

6.4 Statistical Validation of Fine-Tuning Gains

Table 4 examines whether the improvements obtained through supervised fine-tuning are supported by paired sample-level evidence. For each decision stage, changes in Action Accuracy are summarized by their magnitude, confidence interval, McNemar statistic, significance level, and odds ratio. Together, these measures assess both the size and the consistency of the observed gains.

Table 4 – Statistical validation of fine-tuning gains in Action Accuracy.

Stage	Few-shot Ac (%)	SFT Ac (%)	ΔAc (pp)	95% CI	McNemar χ^2	p	OR
Preflop	76.1 ± 2.6	93.3	+17.2	[15.4, 19.0]	117.2	< 0.001	5.9 [4.5–8.6]
Postflop	51.5 ± 2.4	91.8	+40.3	[39.4, 41.2]	3463.0	< 0.001	14.9 [13.3–16.1]

For preflop decisions, supervised fine-tuning improves Action Accuracy by 17.2 percentage points on average. The 95% confidence interval across the ten run-level differences, [15.4, 19.0], remains far from zero. McNemar statistics also indicate strongly asymmetric paired outcomes, with a median value of $\chi^2 = 117.2$ and $p < 0.001$. The median odds ratio is 5.9, ranging from 4.5 to 8.6 across runs, showing that fine-tuning consistently corrects more few-shot errors than it introduces.

Postflop improvement is larger in absolute terms. Action Accuracy increases by 40.3 percentage points, with a 95% confidence interval of [39.4, 41.2]. McNemar's test yields $\chi^2 = 3463.0$ and $p < 0.001$, while the odds ratio reaches 14.9, with a 95% confidence interval of [13.3, 16.1]. Such results are consistent with the pronounced reduction in passive predictions and aggressive-action confusion observed in the postflop matrices.

Statistical evidence therefore supports substantial fine-tuning gains in both stages. Preflop improvement remains consistent across different few-shot demonstration samples, while the larger postflop effect reflects the correction of systematic action-prediction errors in the more contextually diverse decision setting.

6.5 Contextual Performance Across Positions and Postflop Streets

Aggregate results may conceal whether performance depends on table position or postflop street. Figure 7 therefore reports Action Accuracy across preflop positions, postflop

positions, and postflop streets for Qwen3-14B under few-shot prompting and supervised fine-tuning.

Figure 7 – Action Accuracy across preflop positions, postflop positions, and postflop streets for Qwen3-14B under few-shot prompting and supervised fine-tuning.

Preflop few-shot performance varies across positions, ranging from 60.8% to 84.6%. Supervised fine-tuning both raises accuracy and reduces this variation, with position-level results between 90.3% and 94.4%. Improvement is therefore not confined to a particular seat or decision context, but extends across the full preflop positional structure.

Postflop few-shot accuracy ranges from 45.6% to 56.1% across positions. After fine-tuning, the corresponding values increase to between 88.9% and 92.7%. Gains remain substantial throughout the table, indicating that supervised adaptation is not dependent on a single positional subset.

Performance also remains strong across postflop streets. Fine-tuned accuracy reaches high values on the flop, turn, and river, despite the increasing amount of public information and betting history represented in later-street states. Residual variation across streets is modest relative to the overall improvement over few-shot prompting.

Contextual breakdowns consequently reinforce the aggregate findings. Supervised fine-tuning improves Action Accuracy across positions in both stages and maintains strong postflop performance as the decision sequence progresses from flop to river.

6.6 Conditional Sizing Diagnostics

Continuous Ac-s evaluates the complete decision over all test instances, combining categorical correctness with proportional sizing agreement whenever an aggressive action is correctly predicted. Conditional sizing analysis isolates the numerical component by considering only eligible examples in which the reference action is a bet or raise, the predicted action is correct, and both reference and predicted sizings are valid.

Table 5 – Conditional sizing performance of the fine-tuned Qwen3-14B models.

Stage	Conditional sizing score	Eligible n
Preflop	0.994	221
Postflop	0.992	2,168

Conditional sizing performance is high in both decision stages. Qwen3-14B reaches a score of 0.994 preflop and 0.992 postflop, indicating that correctly predicted aggressive actions are generally accompanied by numerical sizings very close to their reference values in proportional terms.

Comparable scores across both stages show that supervised fine-tuning learns not only the categorical decision but also its numerical parameterization. Postflop contains a broader combination of public cards, streets, pot and stack quantities, previous bet sizes, and betting histories. Despite this additional contextual variation, sizing quality after correct aggressive action selection remains nearly as high as in preflop.

Such results complement the global Ac-s values reported in Table 3. Global Ac-s measures performance over the complete evaluation set, including categorical action errors, whereas the conditional score reported here isolates sizing quality after the aggressive action has already been identified.

6.7 Qualitative Analysis of Non-Exact Sizing Predictions

Conditional sizing scores summarize proportional agreement but do not show how individual numerical deviations are treated by the continuous metric. Table 6 presents four representative preflop cases in which the aggressive action is correctly predicted but the numerical sizing does not exactly match the reference.

Table 6 – Representative non-exact preflop sizing predictions.

Hand	Pos.	Context	Ref.	Pred.	Similarity
1	BB	SB calls	3.0	5.0	0.600
2	BTN	HJ raises, CO calls	9.5	15.0	0.633
3	UTG	UTG raises, SB raises	19.0	20.0	0.950
4	HJ	HJ raises, CO and SB call, BB raises	29.2	29.25	0.998

Hand 1 represents a blind-versus-blind context in which the small blind calls and the big blind raises. Training examples with the same ordered action sequence contain reference sizings between 3 and 5 BB. Predicting 5.0 instead of the 3.0 reference therefore suggests that the model recovered a sizing pattern observed in a related context, although

it did not preserve the exact reference value. Its similarity score of 0.600 reflects the substantial proportional difference between both sizes.

Hand 2 contains the largest numerical deviation among the selected examples. After an HJ raise and a CO call, the BTN makes an aggressive raise in a multi-player context, commonly referred to as a squeeze. Predicting 15.0 instead of 9.5 produces a similarity of 0.633. Larger squeeze sizings are more commonly associated with out-of-position responses from the blinds, suggesting that the model may have captured the broader multi-player raising pattern without fully preserving its positional adjustment.

Hands 3 and 4 show substantially closer numerical agreement. In Hand 3, a prediction of 20.0 for a 19.0 reference receives a similarity of 0.950, indicating only a small proportional deviation. Hand 4 reaches 0.998 because 29.25 differs only slightly from the 29.2 reference. Such a result is consistent with rounding-level numerical variation rather than a meaningful change in the predicted sizing pattern.

Taken together, the examples illustrate distinct forms of non-exact agreement. Some predictions appear to reproduce sizing patterns observed in related contexts, whereas others suggest incomplete positional adjustment, small numerical displacement, or rounding. Exact matching would classify all four cases as failures, while proportional similarity preserves their markedly different degrees of agreement.

6.8 Additional Robustness and Ablation Analyses

Reproducing an LLM-based evaluation raises several practical questions beyond aggregate model performance. Results may depend on the number of evaluated examples, the wording used to formulate the task, the procedure adopted to select categorical actions, the instruction used to request numerical sizing, or the decoding configuration.

Complementary experiments examine these factors in preflop and postflop decisions. Evaluation-size sensitivity is measured by subsampling the available prediction pairs. Prompt reformulations test alternative task descriptions, while action-prediction comparisons examine log-probability scoring and generation-based procedures. Additional analyses vary the sizing instruction and decoding temperature after the aggressive action has been selected. Complete configurations and numerical results are reported in Appendix C.

6.8.1 Evaluation-Size Sensitivity

Evaluation-size sensitivity measures how strongly Ac and Ac-s vary when only a fraction of the available test instances is used. Estimates are recomputed from stored prediction pairs rather than through additional inference, isolating sample-size variation from model generation.

Across the evaluated fractions, results remain close to the complete-test-set values in both decision stages. Variation is slightly more visible preflop because its complete test set contains 1,000 examples, compared with 10,000 postflop examples. Nevertheless, reduced subsets preserve the main interpretation of the results, supporting the adequacy of both hold-out evaluation sets.

6.8.2 Action-Prompt Reformulation

Action-prompt reformulation examines whether categorical prediction remains stable when the poker state and available actions are preserved but the task framing or answer cue is changed. Tested variants include concise requests, solver-oriented wording, removal of the specialist persona, and other modifications to the expected response structure.

Results in both stages show that limited changes to the adopted formulation produce smaller effects than broader changes to task framing. Preflop removal of the persona reduces Ac by 2.5 percentage points, while a shorter single-cue construction differs by 1.0 point. Postflop removal of the persona changes Ac by only 0.04 points. Concise and solver-oriented formulations produce larger reductions in both stages.

Such behavior indicates that performance is associated more strongly with the poker-specific task formulation and response structure than with the persona sentence alone. Faithful reproduction should therefore preserve the adopted action instruction rather than treating semantically related prompts as interchangeable.

6.8.3 Action-Prediction Method Comparison

Categorical action selection is compared across next-token log-probability scoring and generation-based alternatives. Preflop conditions include free-form generation, action-list prompting, multiple-choice formulation, and a shorter completion cue. Postflop generation-based methods are compared with log-probability scoring on a matched subset of 2,000 instances.

Preflop log-probability scoring obtains the highest Ac among the evaluated methods. Free-form generation remains close, whereas explicit action-list and multiple-choice formulations produce larger reductions. Postflop full generation differs from log-probability scoring by only 0.05 percentage points on the matched subset, and action-only generation obtains the same Ac.

Log-probability scoring remains preferable for the main evaluation because it is deterministic, directly compares the available actions, and avoids dependence on generated formatting or action parsing.

6.8.4 Sizing-Prompt Sensitivity

Sizing-prompt sensitivity isolates numerical generation after the aggressive action has already been identified. Alternative instructions request the same bet or raise sizing while preserving the poker state, selected action, decoding procedure, numerical parser, and sizing-validity rules.

Preflop Ac-s varies by at most 0.16 percentage points across the seven formulations relative to the baseline of that ablation run. In the separate postflop ablation, the action-specific alternative decreases Ac-s by 0.86 percentage points relative to its corresponding baseline. Sizing performance therefore remains close across the evaluated formulations, although the postflop variation is larger than any individual preflop change.

Because action predictions are fixed before sizing generation, these results measure sensitivity of the numerical component rather than robustness of the complete action-and-sizing pipeline. Absolute Ac-s values from the ablation runs are not compared directly with the primary evaluation because their sizing prompts and generation executions differ.

6.8.5 Temperature Sensitivity

Temperature sensitivity examines whether stochastic decoding changes numerical sizing after categorical action selection. Sizing generation is evaluated using greedy decoding and temperatures of 0.2, 0.7, and 1.0, while categorical actions remain fixed.

Continuous Ac-s remains close to the greedy-decoding baseline in both stages, with maximum absolute variations of 0.45 percentage points in preflop and 0.02 percentage points in postflop. Conditional sizing, however, reveals different behavior across stages. Postflop variation remains negligible, reaching at most 0.04 percentage points, whereas preflop conditional sizing varies by up to 5.45 percentage points. Sampled preflop decoding also increases the number of invalid numerical outputs under correctly predicted aggressive actions.

Greedy generation is retained because it removes sampling variability, reduces invalid numerical outputs, and facilitates exact reproduction. Postflop sizing is strongly robust to the evaluated temperatures, while preflop conditional sizing requires greater caution under stochastic decoding.

6.8.6 Implications for Reproducibility

Taken together, the complementary analyses identify which components require particular care during reproduction. Evaluation-set subsampling produces limited variation in both stages. Aggregate Ac-s remains comparatively stable across the evaluated sizing-prompt and temperature conditions. Conditional analysis nevertheless shows greater preflop sensitivity to stochastic decoding, whereas postflop sizing remains highly stable.

Practical reproduction should therefore preserve the benchmark representation, adopted action prompt, available-action set, and categorical scoring procedure. Deterministic sizing generation remains preferable when exact comparability is required.

6.9 General Discussion

Results consistently show that supervised fine-tuning provides substantially stronger agreement with reference poker decisions than repeated few-shot prompting. Improvement appears in aggregate Action Accuracy, in the continuous Ac-s metric, across individual action classes, and throughout positional and street-level breakdowns. Such convergence across complementary analyses indicates that the gains are not driven by a single action, position, or restricted subset of decision states.

Few-shot prompting nevertheless establishes a meaningful baseline, particularly in preflop. Qwen3-14B reaches the strongest performance among the evaluated open models in both stages, while smaller models display substantial variation that cannot be explained by parameter count alone. Differences in tokenization, instruction tuning, numerical processing, and pretraining likely contribute to the observed ranking. Model selection in this study is therefore empirical and specific to the adopted representation and evaluation protocol rather than evidence of a universal scaling relationship.

Supervised adaptation produces different but complementary gains in the two decision stages. Preflop improvement raises Action Accuracy from 76.1% to 93.3% and continuous Ac-s from 71.4% to 93.2%. Postflop improvement is even larger, raising Action Accuracy from 51.5% to 91.8% and continuous Ac-s from 50.3% to 91.6%. Public cards, street information, pot-related quantities, and longer betting histories create a broader contextual space, but the fine-tuned model maintains strong performance across positions and streets.

Action selection and numerical sizing should still be interpreted as distinct components of the complete decision. Ac measures whether the reference categorical action is recovered, whereas Ac-s evaluates whether correctly predicted aggressive actions are also accompanied by proportionally similar sizings. Conditional sizing analysis removes the action-selection component and examines numerical agreement only after an aggressive action has been correctly identified. Joint use of these metrics prevents strong aggregate performance from concealing whether residual errors arise from categorical selection or numerical parameterization.

Conditional sizing performance remains high in both decision stages, reaching 0.994 preflop and 0.992 postflop. Once the aggressive action is correctly identified, both stage-specific adapters therefore recover the associated numerical sizing with strong proportional agreement.

Preflop sizing is strongly associated with position and the ordered preceding action

sequence, producing a comparatively regular prediction structure. Postflop states contain a broader combination of public boards, streets, pot and stack quantities, previous bet sizes, and betting histories. Despite this additional contextual variation, the postflop conditional score remains close to the preflop result. Such similarity suggests that the remaining postflop performance gap is associated primarily with categorical action selection rather than with sizing calibration after the correct aggressive action has been chosen.

Qualitative examples further illustrate why binary exact matching provides an incomplete description of numerical performance. Predictions with small rounding-level differences and predictions with substantial proportional deviations would otherwise receive the same failure label. Continuous similarity preserves their different degrees of agreement and remains comparable across reference sizings of different magnitudes.

Complementary analyses also clarify which experimental choices matter for reproduction. Within the evaluated configurations, test-set subsampling and sizing-generation variations produce limited changes relative to some action-prompt reformulations. Next-token log-probability scoring attains the highest Action Accuracy among the tested action-prediction methods and avoids formatting-dependent categorical errors. Aggregate numerical performance remains comparatively stable across the tested prompt formulations and decoding temperatures, although preflop conditional sizing shows greater sensitivity under stochastic decoding. Reproduction should therefore preserve the benchmark representation, available-action set, adopted action prompt, and categorical scoring procedure, while deterministic sizing generation remains preferable for exact comparability.

Broader implications extend beyond poker-specific terminology. Inputs in this study combine categorical, numerical, sequential, and hidden-information context, while outputs combine a categorical choice with an optional continuous parameter. Strong performance under this structure suggests that supervised open LLMs can approximate complex reference mappings expressed through text, particularly when inference separates categorical scoring from numerical generation.

Interpretation remains restricted to agreement with PokerBench-derived decisions for isolated states. Experiments do not measure expected value, exploitability, adaptation to specific opponents, consistency across complete hands, or performance in repeated play. High benchmark agreement therefore supports the use of fine-tuned LLMs as predictors of equilibrium-oriented reference decisions, but does not establish poker solving, complete strategic competence, or safe use as standalone decision advisers.

Chapter 7

Conclusion

Open Large Language Models were investigated as text-based predictors of equilibrium-oriented decisions in 6-max No-Limit Texas Hold'em. Poker states were represented through structured textual instructions, and each target contained a categorical action and, when required, a numerical bet or raise sizing. Rather than developing a complete poker-playing agent, the study examined how accurately open LLMs can approximate reference decisions under controlled few-shot and supervised fine-tuning settings.

7.1 Main Findings

Few-shot prompting provides a meaningful baseline, but supervised fine-tuning produces substantially stronger agreement with the reference decisions. Qwen3-14B achieved the highest few-shot performance among the six evaluated models and was therefore selected for adaptation. Preflop Action Accuracy increased from 76.1% to 93.3%, while continuous Action-and-Sizing Accuracy increased from 71.4% to 93.2%. Postflop Action Accuracy increased from 51.5% to 91.8%, while continuous Ac-s increased from 50.3% to 91.6%.

Action-level, positional, street-level, and paired statistical analyses show that the gains extend beyond aggregate averages. Fine-tuning reduces systematic action confusions and improves performance across the evaluated decision contexts. Complementary experiments also clarify the effect of evaluation-set size, prompt formulation, action-prediction method, sizing instruction, and decoding temperature, providing practical guidance for reproducing the adopted pipeline.

Numerical sizing requires an evaluation distinct from categorical action selection. Continuous Ac-s measures the complete decision by assigning proportional credit to correct aggressive actions, while the conditional sizing score isolates numerical agreement after the action has been correctly identified. Preflop and postflop fine-tuning reach conditional sizing scores of 0.994 and 0.992, respectively, over 221 and 2,168 eligible decisions. Compara-

ble scores indicate that, once an aggressive action is correctly selected, both stage-specific adapters recover its numerical parameterization with high proportional agreement.

Postflop remains the more challenging categorical prediction setting because its states combine public cards, multiple streets, pot and stack quantities, previous bet sizes, and longer betting histories. Despite this broader contextual variation, the small difference between postflop Ac and Ac-s and the high conditional sizing score suggest that the remaining performance gap is associated primarily with action selection rather than with sizing calibration after the correct aggressive action has been chosen.

Methodologically, the dissertation contributes a hybrid evaluator that combines next-token log-probability scoring over available actions with deterministic generation of numerical sizings. Joint use of Ac, continuous Ac-s, and the conditional sizing score provides a reproducible evaluation of both categorical and numerical components of poker decisions.

Overall, supervised open LLMs can achieve strong agreement with PokerBench-derived references when supplied with domain-specific training and a controlled inference procedure. Such agreement supports their use as predictors of structured poker decisions, but does not establish poker solving or complete strategic competence.

7.2 Limitations and Future Work

Interpretation remains restricted to isolated benchmark states with fixed 100-BB stacks and one reference decision per example. Experiments do not measure expected value, exploitability, opponent adaptation, consistency over complete hands, or performance in repeated play. Mixed strategies may also assign probability to multiple actions or sizings, whereas the adopted evaluation compares each prediction with one annotated reference. Natural-language explanations were not evaluated because validated strategic rationales are unavailable in the dataset.

Future work should extend the analysis to variable stack depths, additional game formats, complete-hand trajectories, multiple training seeds, alternative textual representations, and evaluation procedures that account for mixed strategies and multiple acceptable sizings. Expected value and exploitability would provide more direct strategic criteria than reference agreement alone.

Combining LLMs with reinforcement learning, self-play, Monte Carlo Tree Search, or Counterfactual Regret Minimization may support systems that integrate learned textual representations with explicit strategic optimization (SUTTON; BARTO, 2018; SILVER et al., 2016; BROWN et al., 2020). Integration with poker solvers, equity calculators, retrieval systems, and game-tree search could also position the LLM as an interface or orchestration component while specialized tools remain responsible for strategic computation.

Beyond poker, future research could investigate whether the proposed separation between categorical prediction and conditional numerical estimation transfers to other structured domains in which outputs combine a discrete decision with an associated continuous quantity.

Bibliography

AGRESTI, A. **Categorical Data Analysis**. 3. ed. [S.l.]: Wiley, 2013.

BARD, N. et al. The annual computer poker competition. **AI Magazine**, v. 34, n. 2, 2013. Disponível em: <https://doi.org/10.1609/aimag.v34i2.2474>.

BILLINGS, D. et al. Approximating game-theoretic optimal strategies for full-scale poker. In: **Proceedings of the 18th International Joint Conference on Artificial Intelligence**. [S.l.: s.n.], 2003.

BOMMASANI, R. et al. On the opportunities and risks of foundation models. **arXiv preprint arXiv:2108.07258**, 2021.

BOWLING, M. et al. Heads-up limit hold'em poker is solved. **Science**, v. 347, n. 6218, p. 145–149, 2015. Disponível em: <https://doi.org/10.1126/science.1259433>.

BROWN, N. et al. ReBeL: Self-play reinforcement learning for imperfect-information games. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2020. v. 33, p. 3649–3661.

BROWN, N. et al. Deep counterfactual regret minimization. In: **Proceedings of the 36th International Conference on Machine Learning**. [S.l.: s.n.], 2019. p. 793–802.

BROWN, N.; SANDHOLM, T. Libratus: The superhuman AI for no-limit poker. In: **Proceedings of the 26th International Joint Conference on Artificial Intelligence**. [s.n.], 2017. p. 5226–5228. Disponível em: <https://doi.org/10.24963/ijcai.2017/772>.

BROWN, N.; SANDHOLM, T. Safe and nested subgame solving for imperfect-information games. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017. v. 30.

BROWN, N.; SANDHOLM, T. Superhuman AI for multiplayer poker. **Science**, v. 365, n. 6456, p. 885–890, 2019. Disponível em: <https://doi.org/10.1126/science.aay2400>.

BROWN, T. B. et al. Language models are few-shot learners. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2020. v. 33, p. 1877–1901.

CARNEIRO, M. G.; LISBOA, G. A. de. What's the next move? learning player strategies in zoom poker games. In: **2018 IEEE Congress on Evolutionary**

Computation (CEC). IEEE, 2018. p. 1–8. Disponível em: <https://doi.org/10.1109/CEC.2018.8477652>.

CBGC. **Texas Hold'em Rules**. 2016. <https://oag.ca.gov/sites/all/files/agweb/pdfs/gambling/BGC_texas.pdf>. California Bureau of Gambling Control. Official game rules. Accessed: 2026-07-21.

DETTMERS, T. et al. QLoRA: Efficient finetuning of quantized llms. In: **Advances in Neural Information Processing Systems**. [s.n.], 2023. v. 36. Disponível em: <https://doi.org/10.52202/075280-0441>.

DINIZ, L. d. O. **Estudo do jogo de pôquer utilizando métodos estatísticos e aprendizado de máquina**. 2022. Trabalho de Conclusão de Curso, Universidade Federal de Uberlândia. Accessed: 2026-05-18. Disponível em: <https://repositorio.ufu.br/handle/123456789/36217>.

DINIZ, L. d. O.; CARNEIRO, M. G. Evaluation of LLMs for effective recommendation of poker strategies. In: **Anais do XXII Encontro Nacional de Inteligência Artificial e Computacional**. Sociedade Brasileira de Computação, 2025. p. 379–390. Disponível em: <https://doi.org/10.5753/eniac.2025.12461>.

EFRON, B. Bootstrap methods: Another look at the jackknife. **The Annals of Statistics**, v. 7, n. 1, p. 1–26, 1979. Disponível em: <https://doi.org/10.1214/aos/1176344552>.

EFRON, B.; TIBSHIRANI, R. J. **An Introduction to the Bootstrap**. Chapman and Hall/CRC, 1993. Disponível em: <https://doi.org/10.1007/978-1-4899-4541-9>.

GALLOTTA, R. et al. **Large Language Models and Games: A Survey and Roadmap**. 2024. Disponível em: <https://doi.org/10.1109/TG.2024.3461510>.

Gemma Team. **Gemma 2: Improving Open Language Models at a Practical Size**. 2024.

GRATTAFIORI, A. et al. **The Llama 3 Herd of Models**. 2024.

GTO Wizard. **GTO Wizard**. 2025. <https://gtowizard.com/>. Poker solver and training platform. Accessed: 2026-05-18.

GUPTA, A. Are ChatGPT and gpt-4 good poker players? a pre-flop analysis. **arXiv preprint arXiv:2308.12466**, 2023.

HU, E. J. et al. LoRA: Low-rank adaptation of large language models. In: **International Conference on Learning Representations**. [S.l.: s.n.], 2022.

HU, S. et al. **A Survey on Large Language Model-Based Game Agents**. 2024.

HUANG, C. et al. **PokerGPT: An End-to-End Lightweight Solver for Multi-Player Texas Hold'em via Large Language Model**. 2024.

IMSA. **World Poker Federation Becomes the Ninth Affiliate Member of the International Mind Sports Association**. 2024. <https://imsa.sport/2024/11/22/world-poker-federation-becomes-the-ninth-affiliate-member-of-the-international-mind-sports-associa>. Accessed: 2026-07-21.

JIANG, A. Q. et al. **Mistral 7B**. 2023.

JOHANSON, M. **Measuring the Size of Large No-Limit Poker Games**. [S.l.], 2013. Disponível em: <https://poker.cs.ualberta.ca/publications/2013-techreport-nl-size.pdf>.

KAPLAN, J. et al. Scaling laws for neural language models. **arXiv preprint arXiv:2001.08361**, 2020.

KUHN, H. W. A simplified two-person poker. In: KUHN, H. W.; TUCKER, A. W. (Ed.). **Contributions to the Theory of Games**. Princeton University Press, 1950, (Annals of Mathematics Studies, 24). p. 97–103. Disponível em: <https://doi.org/10.1515/9781400881727-010>.

LESTER, B.; AL-RFOU, R.; CONSTANT, N. The power of scale for parameter-efficient prompt tuning. In: **Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing**. [s.n.], 2021. p. 3045–3059. Disponível em: <https://doi.org/10.18653/v1/2021.emnlp-main.243>.

LI, B.; WANG, B.; HUANG, L. **PokerSkill: LLMs Can Play Expert-Level Poker without Training or Solvers**. 2026. Disponível em: <https://doi.org/10.48550/arXiv.2605.30094>.

LI, Y. et al. **Large Language Models in Finance: A Survey**. 2023. Disponível em: <https://doi.org/10.20944/preprints202409.1085.v2>.

LIN, M. et al. **How Far Are LLMs from Professional Poker Players? Revisiting Game-Theoretic Reasoning with Agentic Tool Use**. 2026.

LIU, P. et al. Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing. **ACM Computing Surveys**, v. 55, n. 9, p. 1–35, 2023. Disponível em: <https://doi.org/10.1145/3560815>.

LU, Y. et al. Fantastically ordered prompts and where to find them: Overcoming few-shot prompt order sensitivity. In: **Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics**. [s.n.], 2022. p. 8086–8098. Disponível em: <https://doi.org/10.18653/v1/2022.acl-long.556>.

MAUGIN, N.; CAZENAVE, T. **SpinGPT: A Large-Language-Model Approach to Playing Poker Correctly**. 2025. Disponível em: <https://doi.org/10.1007/978-3-032-23657-9_8>.

MCNEMAR, Q. Note on the sampling error of the difference between correlated proportions or percentages. **Psychometrika**, v. 12, n. 2, p. 153–157, 1947. Disponível em: <https://doi.org/10.1007/BF02295996>.

MIALON, G. et al. Augmented language models: A survey. **Transactions on Machine Learning Research**, 2023.

MORAVČÍK, M. et al. DeepStack: Expert-level artificial intelligence in heads-up no-limit poker. **Science**, v. 356, n. 6337, p. 508–513, 2017. Disponível em: <https://doi.org/10.1126/science.aam6960>.

NASH, J. F. Non-cooperative games. **Annals of Mathematics**, v. 54, n. 2, p. 286–295, 1951. Disponível em: <https://doi.org/10.2307/1969529>.

NEUMANN, J. von; MORGENSTERN, O. **Theory of Games and Economic Behavior**. [S.l.]: Princeton University Press, 1944.

OSBORNE, M. J.; RUBINSTEIN, A. **A Course in Game Theory**. [S.l.]: MIT Press, 1994.

OUYANG, L. et al. Training language models to follow instructions with human feedback. In: **Advances in Neural Information Processing Systems**. [s.n.], 2022. v. 35, p. 27730–27744. Disponível em: <https://doi.org/10.52202/068431-2011>.

PioSOLVER. **PioSOLVER: A GTO Solver for Hold'em Poker**. 2026. <https://piosolver.com/>. Poker solver and training software. Accessed: 2026-07-21.

POKERSTARS. **Texas Hold'em Rules**. 2026. <https://www.pokerstars.com/poker/learn/lesson/texas-holdem-rules/>. Game rules and terminology. Accessed: 2026-07-21.

Potter van Loon, R. J. D. et al. Beyond chance? the persistence of performance in online poker. **PLOS ONE**, v. 10, n. 3, p. e0115479, 2015. Disponível em: <https://doi.org/10.1371/journal.pone.0115479>.

PROVOST, M.-A. et al. **GTO Wizard Benchmark**. 2026. Disponível em: <https://doi.org/10.48550/arXiv.2603.23660>.

RUSSELL, S.; NORVIG, P. **Artificial Intelligence: A Modern Approach**. 4. ed. [S.l.]: Pearson, 2020.

SANDHOLM, T. Solving imperfect-information games. **Communications of the ACM**, v. 58, n. 10, p. 78–87, 2015. Disponível em: <https://doi.org/10.1145/2699390>.

SANH, V. et al. Multitask prompted training enables zero-shot task generalization. In: **International Conference on Learning Representations**. [S.l.: s.n.], 2022.

SHOHAM, Y.; LEYTON-BROWN, K. **Multiagent Systems: Algorithmic, Game-Theoretic, and Logical Foundations**. Cambridge University Press, 2008. Disponível em: <https://doi.org/10.1017/CBO9780511811654>.

SILVER, D. et al. Mastering the game of go with deep neural networks and tree search. **Nature**, v. 529, n. 7587, p. 484–489, 2016. Disponível em: <https://doi.org/10.1038/nature16961>.

SOKOLOVA, M.; LAPALME, G. A systematic analysis of performance measures for classification tasks. **Information Processing & Management**, v. 45, n. 4, p. 427–437, 2009. Disponível em: <https://doi.org/10.1016/j.ipm.2009.03.002>.

SUN, H. et al. Game theory meets large language models: A systematic survey. In: **Proceedings of the Thirty-Fourth International Joint Conference on Artificial Intelligence**. [s.n.], 2025. p. 10669–10677. Disponível em: <https://doi.org/10.24963/ijcai.2025/1184>.

SUTTON, R. S.; BARTO, A. G. **Reinforcement Learning: An Introduction**. 2. ed. [S.l.]: MIT Press, 2018.

TEÓFILO, L. F. G. **Building a Poker Playing Agent Based on Game Logs Using Supervised Learning**. Dissertação (Master's thesis) — Faculdade de Engenharia da Universidade do Porto, 2010.

VASWANI, A. et al. Attention is all you need. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2017. v. 30.

WASM Postflop. **WASM Postflop: Free GTO Solver for Web Browser**. 2023. <https://github.com/b-inary/wasm-postflop>. Open-source Texas Hold'em postflop solver. Accessed: 2026-05-18.

WEI, J. et al. Finetuned language models are zero-shot learners. **arXiv preprint arXiv:2109.01652**, 2022.

YANG, A. et al. **Qwen3 Technical Report**. 2025.

YANG, H.; LIU, X.-Y.; WANG, C. D. **FinGPT: Open-Source Financial Large Language Models**. 2023. Disponível em: <https://doi.org/10.2139/ssrn.4489826>.

ZHAO, Z. et al. Calibrate before use: Improving few-shot performance of language models. In: **Proceedings of the 38th International Conference on Machine Learning**. [S.l.]: PMLR, 2021. v. 139, p. 12697–12706.

ZHUANG, R. et al. PokerBench: Training large language models to become professional poker players. In: **Proceedings of the AAAI Conference on Artificial Intelligence**. [s.n.], 2025. v. 39, n. 24, p. 26175–26182. Disponível em: <https://doi.org/10.1609/aaai.v39i24.34814>.

ZINKEVICH, M. et al. Regret minimization in games with incomplete information. In: **Advances in Neural Information Processing Systems**. [S.l.: s.n.], 2007. v. 20.

Appendix

APPENDIX A

Illustration of Poker Hand Stages

This appendix provides a compact visual illustration of the flow of a Texas Hold'em hand. The screenshots are included only as visual aids to support the reader's understanding of poker terminology and game structure; they are not part of the experimental protocol.

Figure 8 illustrates a preflop decision state. At this stage, no community cards have been revealed yet, and each active player holds two private cards, also called hole cards. The table layout shows the relative seating positions of the players, their chip stacks, the pot size, and the current action sequence. The red arrow highlights the action log, indicating the sequence of decisions that has occurred up to the current point in the hand.

After the preflop round, community cards are revealed in stages: the flop consists of the first three community cards, the turn adds a fourth community card, and the river adds the fifth and final community card. These public cards are shared by all remaining players and progressively change the strategic context of the hand. Figure 9 illustrates the final stage, the showdown, in which the remaining players reveal their private cards and the winner is determined.

Figure 8 – Illustrative preflop decision state. The red arrow highlights the action log, showing the sequence of actions before the current decision.

Figure 9 – Illustrative showdown state. At showdown, the remaining players reveal their private cards and the winner is determined using the complete board of five community cards.

APPENDIX **B**

Example Textual Prompts

This appendix provides examples of the textual prompts used to represent poker decision states in the experiments. The prompts follow the PokerBench-style format adopted throughout the dissertation: a natural-language game summary is provided, followed by a direct answer cue. The model is expected to output only the recommended action, optionally followed by a numerical size when the action is aggressive.

In the supervised fine-tuning setting, the prompt is used as the input and the reference label is used as the supervised target. In the few-shot setting, labeled examples are concatenated before the target instance, but the target decision follows the same textual representation. For clarity, the examples below separate the prompt from the reference output.

B.1 Preflop Prompt Example

Prompt:

```
You are a specialist in playing 6-handed No Limit Texas Holdem.  The following will be a
game scenario and you need to make the optimal decision.

Here is a game summary:

The small blind is 0.5 chips and the big blind is 1 chips.  Everyone started with 100
chips.
The player positions involved in this game are UTG, HJ, CO, BTN, SB, BB.
In this hand, your position is BB, and your holding is [Ace of Club and Six of Spade].
Before the flop, SB call.  Assume that all other players that is not mentioned folded.

Now it is your turn to make a move.
To remind you, the current pot size is 1.0 chips, and your holding is [Ace of Club and Six
of Spade].

Decide on an action based on the strength of your hand on this board, your position, and
actions before you.  Do not explain your answer.
Your optimal action is:
```

Reference output: `check`

B.2 Postflop Prompt Example

Prompt:

```
You are a specialist in playing 6-handed No Limit Texas Holdem.  The following will be a
game scenario and you need to make the optimal decision.

Here is a game summary:

The small blind is 0.5 chips and the big blind is 1 chips.  Everyone started with 100
chips.
The player positions involved in this game are UTG, HJ, CO, BTN, SB, BB.
In this hand, your position is HJ, and your holding is [Six of Spade and Six of Club].
Before the flop, HJ raise 2.0 chips, and BB call.  Assume that all other players that is
not mentioned folded.
The flop comes Ten Of Spade, King Of Club, and Ace Of Heart, then BB check.

Now it is your turn to make a move.
To remind you, the current pot size is 4.0 chips, and your holding is [Six of Spade and Six
of Club].

Decide on an action based on the strength of your hand on this board, your position, and
actions before you.  Do not explain your answer.
Your optimal action is:
```

Reference output: `bet 3`

APPENDIX C

Additional Robustness and Ablation Analyses

Complete configurations and numerical results for the complementary analyses summarized in Section 6.8 are reported here. The experiments examine evaluation-set size, action-prompt formulation, action-prediction method, sizing-prompt wording, and decoding temperature.

Results are presented for preflop and postflop whenever the corresponding experiment was conducted in both stages. Each analysis isolates one component of the evaluation pipeline while preserving the remaining model, data, and inference settings.

C.1 Evaluation-Size Sensitivity

Evaluation-size sensitivity was assessed by recomputing the metrics on random subsamples of the available prediction pairs. No additional model inference was required. Five random subsamples were drawn for each reduced fraction, while the complete test sets correspond to the full evaluation results.

Table 7 – Sensitivity to the number of evaluated test instances.

Stage	Test fraction	N	Ac (%)	Ac-s (%)
Preflop	25%	250	92.9 ± 1.0	92.7 ± 1.1
Preflop	50%	500	93.6 ± 0.7	93.5 ± 0.7
Preflop	100%	1,000	93.3	93.2
Postflop	25%	2,500	91.60 ± 0.44	91.45 ± 0.58
Postflop	50%	5,000	91.77 ± 0.33	91.60 ± 0.21
Postflop	100%	10,000	91.78	91.61

Estimates remain close to the complete-test-set values across all evaluated fractions. Preflop variation is slightly more visible because its complete test set contains 1,000 in-

stances, whereas postflop evaluation uses 10,000. At 50% of the postflop test set, Ac and Ac-s differ from their complete-set values by less than 0.1 percentage points. Reduced subsets therefore preserve the main interpretation of both metrics in each stage, supporting the adequacy of the adopted hold-out evaluation sets.

C.2 Action-Prompt Reformulation

Action-prompt reformulation evaluates whether categorical predictions change when the poker state and available actions remain fixed but the task instruction is expressed differently. All variants use next-token log-probability scoring, isolating prompt construction from the action-selection method and numerical sizing generation.

P0 corresponds to the adopted instruction. P1 shortens the request while preserving the poker-decision objective. P2 introduces solver-oriented framing. P3 presents the state through a neutral input–prediction formulation. P4 removes the explicit specialist persona while preserving most of the decision wording. A single-cue construction provides an additional preflop comparison using a shorter completion cue.

Table 8 – Action Accuracy under alternative action-prompt formulations.

Stage	Variant	Ac (%)	ΔAc (pp)	Flipped (%)
Preflop	P0: adopted prompt	93.3	—	0.0
Preflop	P1: concise	82.6	−10.7	13.8
Preflop	P2: solver framing	84.5	−8.8	11.1
Preflop	P3: neutral input–prediction framing	34.0	−59.3	60.2
Preflop	P4: no-persona	90.8	−2.5	3.1
Preflop	Single-cue construction	92.3	−1.0	2.4
Postflop	P0: adopted prompt	92.21	—	–
Postflop	P1: concise	87.63	−4.58	–
Postflop	P2: solver framing	86.36	−5.85	–
Postflop	P4: no-persona	92.17	−0.04	–

Preflop results vary markedly across the tested formulations. Removing the persona while preserving most of the instruction reduces Ac by 2.5 percentage points, and the single-cue construction differs by 1.0 point from the adopted prompt. More extensive changes produce larger reductions, reaching 59.3 points under the neutral input–prediction formulation.

Postflop results follow a related pattern. Concise and solver-oriented reformulations reduce Ac by 4.58 and 5.85 percentage points, respectively, whereas removing the specialist persona changes performance by only 0.04 points. Prompt sensitivity is therefore associated more strongly with task framing and decision wording than with the persona sentence alone.

Observed differences indicate that categorical performance is not invariant to alternative task formulations. Reproduction should preserve the adopted action prompt, particularly its poker-specific framing and expected response structure.

C.3 Action-Prediction Method Comparison

Action-prediction method comparison examines whether categorical performance depends on how the model output is obtained. M0 applies next-token log-probability scoring directly over the available actions. M1 generates the complete decision in free form. M2 requests an action through a generation-based action-only formulation. Additional preflop conditions use an explicit action-list prompt, a multiple-choice formulation, and a shorter single-cue generation format.

Generation-based postflop conditions are evaluated on a 2,000-instance subset because full response generation and parsing require greater inference cost than scoring candidate actions. The M0 subset result provides the matched comparison anchor, while the complete M0 result contextualizes performance on the full 10,000-instance test set.

Table 9 – Action Accuracy under alternative action-prediction methods.

Stage	Method	N	Ac (%)	ΔAc (pp)
Preflop	M0: log-probability scoring	1,000	93.3	0.0
Preflop	M1: free-form generation	1,000	92.5	−0.8
Preflop	M2: constrained action-list prompt	1,000	88.4	−4.9
Preflop	M3: multiple-choice formulation	1,000	84.5	−8.8
Preflop	Single-cue generation	1,000	92.3	−1.0
Postflop	M0: log-probability scoring, full set	10,000	92.21	0.00
Postflop	M0: log-probability scoring, subset	2,000	91.60	−0.61
Postflop	M1: full generation	2,000	91.65	−0.56
Postflop	M2: action-only generation	2,000	91.60	−0.61

Preflop log-probability scoring obtains the highest Action Accuracy among the evaluated methods. Free-form generation remains close, decreasing Ac by only 0.8 percentage points. Larger reductions occur when the response space is reformulated as an explicit action-list or multiple-choice task.

Postflop generation-based methods remain close to log-probability scoring on the matched 2,000-instance subset. Full generation differs by only 0.05 percentage points, while action-only generation obtains the same Ac as the subset M0 condition. The 0.61-point difference between the complete and subset M0 results reflects the evaluated sample rather than a change in action-prediction method.

Log-probability scoring remains preferable for the main evaluation because it is deterministic, directly compares the available actions, and avoids dependence on generated formatting or action parsing.

C.4 Sizing-Prompt Sensitivity

Sizing-prompt sensitivity examines whether numerical predictions change when the selected aggressive action is kept fixed but the instruction used to request its sizing is reformulated. Action predictions remain unchanged across conditions within each stage, so only sizing generation contributes to differences in Ac-s. Results are interpreted relative to the baseline prompt of each ablation run rather than to the primary evaluation run.

Table 10 summarizes the preflop templates. S0 corresponds to the baseline instruction adopted within the preflop sensitivity experiment. Remaining variants modify the amount of detail, whether the selected action is explicitly named, the response cue, or the inclusion of numerical examples. Postflop sensitivity was evaluated in a separate ablation run using its own baseline and alternative sizing instructions.

Table 10 – Sizing-prompt variants used in the preflop sensitivity analysis.

Template	Instruction
S0	`Output only the numeric size in chips. Size:`
S1	`Return only the size in chips as a number (e.g., 10 or 12.5). No words. Size:`
S2	`Given the selected action, output the corresponding bet or raise size in chips. Return a single number only. Size:`
S3	`Numeric size in chips:`
S4	`Size:`
S5	`The selected action is {action}. Provide the strategic size for this action as a single numeric chip amount. Do not include words. Size:`
S6	`Write only the numeric amount for the selected {action}. For example: 10 or 12.5. Amount:`

Although the templates differ in wording and level of detail, all conditions preserve the same poker state, selected aggressive action, decoding procedure, numerical parser, and sizing-validity rules. Differences in Table 11 therefore isolate the effect of the sizing instruction on numerical generation.

Table 11 – Sensitivity of Ac-s to alternative sizing-prompt formulations.

Stage	Template	ΔAc-s (pp)
Preflop	S0	0.00
Preflop	S1	−0.13
Preflop	S2	−0.16
Preflop	S3	−0.14
Preflop	S4	−0.10
Preflop	S5	−0.07
Preflop	S6	−0.12
Postflop	S0	0.00
Postflop	S6	−0.86

Preflop Ac-s varies within a narrow interval across all seven templates, with a maximum reduction of only 0.16 percentage points relative to its ablation baseline. Postflop Ac-s decreases by 0.86 points under S6 relative to the corresponding postflop baseline. Across the evaluated variants, sizing performance remains close to the baseline instruction within each ablation run, although the postflop alternative produces a larger change than any individual preflop reformulation.

Because action predictions are fixed before sizing generation, these results measure sensitivity of the numerical component rather than robustness of the complete action-and-sizing pipeline. Absolute Ac-s values from these ablation runs are not compared directly with the primary evaluation because their sizing prompts and generation executions differ.

C.5 Temperature Sensitivity

Temperature sensitivity examines whether stochastic decoding changes numerical sizing after the aggressive action has already been selected. Categorical actions are obtained through deterministic log-probability scoring and therefore remain fixed across all conditions.

Sizing generation was evaluated using greedy decoding and temperatures of 0.2, 0.7, and 1.0. Continuous Ac-s remained close to the greedy-decoding baseline in both stages. Maximum absolute variation reached 0.45 percentage points in preflop and 0.02 percentage points in postflop.

Conditional sizing results reveal a different pattern across stages. Postflop variation remained negligible, with a maximum absolute change of 0.04 percentage points in the conditional sizing score. Preflop conditional performance varied by as much as 5.45 percentage points, while the mean number of invalid sizings under correctly predicted aggressive actions increased from 12 under greedy decoding to approximately 36 under

sampled decoding. Thus, preflop aggregate Ac-s remains stable, but conditional sizing quality is more sensitive to stochastic generation.

Greedy decoding is retained in the main evaluation because it removes sampling variability, reduces invalid numerical outputs, and facilitates exact reproduction. Observed results support strong temperature robustness in postflop sizing, while preflop findings recommend greater caution when interpreting conditional sizing under stochastic decoding.