

ALLEGHENY COLLEGE
COMPUTER AND INFORMATION SCIENCE DEPARTMENT

Senior Thesis

gAImble

by

William Wolff

ALLEGHENY COLLEGE

DEPARTMENT OF COMPUTER AND
INFORMATION SCIENCE

Project Supervisor: **Douglas Luman**
Co-Supervisor: **Morgan Green**

Abstract

Ever since the rise in popularity with large language models such as GPT, Gemini, and Claude. We now are seeing an advance towards a new age of technology. With what we know now about large language models, there are still further questions on how advanced are large language models. This test is to see how well can large language models reason in a risk/reward scenario with the scenario chosen being blackjack. The average player wins 42% of the time. Using the basic blackjack strategy as a baseline evaluation, we look to see if 3 large language models (GPT, Gemini, and Claude) can reason and be either more than 2% better or worse than the average player. To see if the reasoning is on par with reasoning, each large language cannot make an illegal move more than 3% of the time. Furthermore, this project look to see if models can reason more than 20% of the total runs that have occurred when ran.

Table of contents

1	Introduction	5
1.1	Motivation	8
1.2	Current State of the Art	9
1.3	Goals of the Project	11
1.4	Ethical Implications	13
2	Related work	17
3	Method of approach	23
3.1	Models	23
3.2	Blackjack	23
3.3	Code	26
4	Experiments	31
4.1	Blackjack	31
4.2	Table of Similar hands	33
4.3	CSV Evaluation	34
4.3.1	Comparison with Book Model	34
4.3.2	Analysis of Similar Hands	35
4.3.3	Win Percentage	36
4.4	Reasoning Results and Illegal Plays	36
4.5	Threats to Validity	40
5	Conclusion	42
5.1	Summary of Results	42
5.2	Future Work	44
5.3	Future Ethical Implications and Recommendations	47

List of Figures

1	Process of Experiment	25
2	Example of Outputs for Each LLM	33
3	Comparison of Data Results for Each LLM	34

List of Tables

1 Introduction

Large language models (LLM) in the modern time have been a major advancement in the computing world as well as the real world. LLM, such as GPT[1], Claude[2], and Gemini[3], have gained mass popularity as being an aid to helping many with writing papers as well as code[24]. LLM now seemingly have the ability to reason and give explanation behind the thought process of their explanation. This leads in to the growth of all artificial intelligence (AI) in the world. Though they give a response most of the time they are automated responses in a format that is default to prompt when asked to give a response. The ability of LLM to reason is mostly due to prompting techniques such as few-shot prompting and zero-shot prompting[14]. This work is to test if large language models can reason in a reward/penalty environment, see if any of the LLM being used adapt strategies to maximize long-term profit, and as well to test the ability to take risks based on incomplete information.

LLM in the modern era represent a transformative advancement in the fields of computing and artificial intelligence, with significant implications for the real world. These models, GPT[1], Claude[2], and Gemini[3], have garnered widespread acclaim for their ability to assist with diverse tasks such as drafting essays, writing code, and generating creative content[7]. Their growing capabilities extend beyond simple text generation; they now seemingly possess the ability to reason, articulate complex explanations, and simulate thought processes. However, while their responses often appear thoughtful and deliberate, they are largely shaped by underlying algorithms and training paradigms, with techniques like zero-shot and few-shot prompting playing a critical role in steering their outputs[14].

This study seeks to move beyond static evaluation and explore the dynamic reasoning capabilities of LLM in scenarios that involve risk, reward, and penalty[7]. By testing these models in environments that require decision-making under uncertainty, such as games or simulations, we aim to evaluate their ability to adapt strategies to maximize long-term profit[12]. Specifically, this work will examine whether large language models can demonstrate risk assessment, strategic planning, and the capacity to learn or adjust behaviors in the pursuit of an optimal outcome[12]. Understanding their ability to reason in such contexts not only provides insight into their cognitive-like capabilities but also serves as a litmus test for their potential applications in fields such as finance, game theory, and autonomous decision-making systems. This exploration bridges the gap between theoretical capabilities and practical implications, offering a deeper perspective on how these models perform in scenarios demanding more than static responses.

The ultimate goal of this project is to investigate whether LLM show signs of evolution in reasoning skills and decision-making processes[20]. By placing these models in a dynamic, reward-oriented environment, we aim to determine whether they can adapt their strategies over time, learn from feedback, and make more sophisticated decisions based on incomplete or uncertain information. This exploration is important for understanding whether large language

models simply execute pre-trained patterns or whether they can demonstrate emergent behavior that resembles adaptive reasoning[17]. Insights from this study will help assess the development trajectory of large language models, highlighting their potential for continuous improvement and their ability to address increasingly complex real-world challenges. Furthermore, it will inform future efforts to design LLM that are not only smarter, but also more attuned to human reasoning and problem-solving skills.

By analyzing how large language models respond to repeated challenges and changing scenarios, this project seeks to discover whether their reasoning evolves in ways that demonstrate genuine adaptability or remains limited to pre-programmed heuristics[17]. If large language models can demonstrate the ability to refine strategies, balance short-term risks with long-term rewards, and adjust behavior based on cumulative results, this could signal important advances in their cognitive capabilities. The findings will not only validate the growing sophistication of these models, but also open the door to new applications in areas such as strategic planning, autonomous decision-making and human-computer collaboration[17]. Furthermore, the project aims to build a framework for evaluating the development of LLM reasoning, providing researchers and developers with benchmarks to measure progress over time. Ultimately, these findings can guide the ethical development of large language models, ensuring that, as they develop, large language models do so in a way that is safe, transparent and beneficial to society.

A key component of this study is to determine whether large language models can exhibit learning behavior over time. While traditional models process each question independently, the ability to retain and apply past experiences could represent a breakthrough in LLM reasoning[19]. By tracking patterns in decision-making, we aim to observe whether repeating games results in measurable improvements, similar to how human players refine their strategies through experience. Furthermore, this study seeks to evaluate whether different large language models take different approaches when faced with identical risk-reward scenarios. By comparing their responses, we can assess whether there are significant differences in their thought processes, strategy adaptations, or risk tolerance levels. This comparison will be critical to identifying strengths and weaknesses among leading models and guiding future development efforts in LLM reasoning[15]. Additionally, this study will explore whether large language models can recognize and correct past mistakes, thereby refining their strategies over multiple rounds of decision-making. This self-improvement aspect is critical to the development of artificial intelligence, as it mirrors the human ability to learn from mistakes and adjust behavior accordingly. By assessing whether LLM can move beyond static decision trees towards more detailed and evolving strategies, we can better understand the extent of its adaptation in dynamic environments[10].

Reasoning in LLMs is a complex, evolutionary process involving pattern recognition, probabilistic decision making, and the application of acquired knowledge to new situations[22]. Unlike human reasoning, which integrates intuition, experience, and abstract thought, LLMs operate by recognizing

statistical relationships in large data sets. This means that while they can mimic known strategies, their understanding is fundamentally different from human cognition. In structured decision-making tasks such as blackjack, LLM reasoners are tested on their ability to assess risk, evaluate probabilities, and follow strategic principles based on established models. However, because LLMs lack true cognitive understanding or awareness, their reasoning is limited by the quality and breadth of the data on which they are trained.

One of the main challenges of reasoning in LLMs is ensuring consistency in decision-making. Because these models often rely on probabilistic outputs, they may not always make the same decision when faced with identical inputs. This can be particularly problematic in environments where consistency is important, such as finance, legal reasoning, or medical diagnosis. In a game of blackjack, for example, an LLM may sometimes decide to hit on 12 against a dealer’s 4, and at other times choose to stand, depending on slight variations in how the dice are processed. While this variability can sometimes lead to creative problem solving, it also raises concerns about reliability and predictability, especially in high-stakes applications. Reasoning in LLMs can be affected by biases present in the training data. If an LLM is trained primarily on suboptimal blackjack strategies, it may introduce flawed decision-making patterns without realizing its inefficiency. Similarly, in broader applications, if an LLM is trained on a biased or incomplete dataset, its reasoning may reflect those biases, leading to skewed or unfair outcomes. This highlights the importance of not only selecting high-quality training data, but also implementing a process of continuous evaluation and refinement to ensure that LLMs make rational, unbiased, and ethically sound decisions[6].

Despite these challenges, reasoning in LLMs has the potential to improve through advances in reinforcement learning and contextual understanding. By integrating real-time feedback mechanisms, LLMs can learn from their mistakes, refining their decision-making processes over time[22]. Additionally, increasing the interpretability of reasoning in LLMs—allowing users to understand why an LLM made a particular decision—can increase trust and transparency. As LLMs continue to evolve, honing their reasoning abilities will be essential to ensuring they can function effectively in complex real-world scenarios, from strategic games like blackjack to critical decision-making areas like healthcare, law, and finance.

Another important factor under investigation is whether large-scale language models exhibit a tendency towards risk-averse or risk-seeking behavior in extended games. By analyzing their decision-making patterns, we can determine whether LLMs prefer conservative strategies that minimize losses or whether they take calculated risks in pursuit of larger gains. Understanding these trends will provide insight into how models prioritize short-term versus long-term gains and whether they can effectively balance the two in complex scenarios[25]. Ultimately, this study will contribute to a broader discussion about decision-making in LLMs in real-world applications beyond games. The ability to process incomplete information, consider probabilities, and optimize strategies is relevant to fields such as finance, healthcare, and autonomous systems. By investigating

the performance of LLMs in controlled risk-reward environments, we can gain valuable insights into their potential for practical, high-stakes decision-making across a variety of domains.

1.1 Motivation

One of the most well-known risk/reward games in America is blackjack. Statistics have shown that in 2018, 46% of Americans from the ages of 18-29 play blackjack [13]. Blackjack being made to be as the theoretically best odds for a player to beat the dealer is the reason why it is the most played game in casinos and as well online casinos. As well there are many methods that claim to be the most effective to exercise your winnings, which defines more for large language models to try and play with in order to learn more about having the better strategy[12]. Blackjack is an excellent reasoning task for LLM because it encapsulates a variety of logical, probabilistic, and strategic thinking challenges. These aspects make it ideal for testing and showcasing LLM capabilities in reasoning, decision-making, and conversational interaction[17]. Another key aspect that is key is testing the memory capacity of the large language models and see if past wins, losses, or pushes affect how the next hand will be played[12].

The motivation behind this project stems from the multifaceted nature of blackjack as a reasoning task[12]. It combines probabilistic thinking, strategic adaptation, and decision-making under uncertainty, all of which are key areas where large language models are believed to excel. Additionally, blackjack has inspired countless strategies aimed at maximizing winnings, from basic strategy charts to advanced techniques like card counting[14]. This rich landscape of potential approaches provides an ideal testing ground for LLM to simulate reasoning, assess risks, and refine strategies. By engaging with blackjack, we can explore whether these models can learn to adapt to new information, apply optimal strategies, and balance short-term gains with long-term rewards[12].

Blackjack allows us to investigate memory capabilities in LLM —an often overlooked aspect of their reasoning processes[6]. By tracking whether past hands successes or failures influence the models future decisions, we can better understand their ability to simulate learning and behavioral adjustment. This study is not just about measuring performance in a game; it’s about probing the deeper question of whether large language models can function effectively in environments that require dynamic, contextual reasoning and adaptability. Ultimately, this project seeks to shed light on the potential of large language models to operate in real-world scenarios where uncertainty, strategy, and memory interplay to determine success[7].

This research extends beyond blackjack itself, as the insights gained may have implications for broader applications of LLMs. Many real-world problems involve decision-making under uncertainty, from financial forecasting to medical diagnostics and autonomous systems. By analyzing how LLMs navigate the risk-reward dynamics of blackjack, we can infer their potential effectiveness in high-stakes environments where selecting optimal strategies is critical [25]. This exploration helps bridge the gap between the theoretical capabilities of LLMs

and their practical applications in decision science. Furthermore, understanding how different LLMs approach the same game can provide valuable insights into their unique reasoning structures. Some models may exhibit more conservative strategies, prioritizing risk mitigation, while others may adopt an aggressive approach in pursuit of greater rewards. These differences in behavior may indicate how different models weight probabilities, process past experiences, and prioritize different decision-making interests[1]. By categorizing and comparing these trends, we can better understand the diversity in LLM reasoning systems and refine them for more complex problem-solving tasks.

In addition to individual decision-making, the study also investigated whether LLMs could collaborate or adapt their strategies when receiving external advice or rule modifications. In real-world applications, LLMs frequently interact with humans and other agents, requiring the ability to adapt in dynamic environments [2]. If a model could incorporate new strategies based on external feedback or information, it could represent a step toward more interactive and responsive LLMs, capable of not only learning but also integrating human guidance into their decision-making processes. Ultimately, the project is not just about testing LLMs in a gaming environment, but also assessing their cognitive flexibility. If LLMs can refine their strategies over time, learning from experience, and adjusting their approach based on situational factors, it could signal a shift toward more autonomous and reasoning LLM decisions. Blackjack serves as an ideal microcosm for this broader investigation, offering a structured yet dynamic environment in which adaptability and learning can be rigorously tested [3].

1.2 Current State of the Art

Large language models should have the ability to provide the user with a quality response without requiring any user adjustment to the prompts or refine repeatedly if the initial response is not adequate. As well, there is chain-of-thought prompting which allows the large language model to give connecting steps to allow the model to think step by step[23]. Most existing work related to reasoning with large language models imply that large language models struggle in direct reasoning scenarios. Though if we were to consider taking an approach towards a risk reward scenario, we will see an emergence of a chain-of-thought prompting[24]. The most used large language models frequently have updates to their systems in order to be the most efficient in their jobs. These updates led to several changes in the response types to the most effective when understanding a response from an inputted answer[25].

Reasoning tasks for a large language model involve leveraging the model’s capabilities to analyze and generate responses based on complex inputs from a user[24]. These tasks enable the model to solve problems as well as explain concepts and draw inferences of what is the most ideal explanation [7]. Reasoning tasks for a large involve leveraging the model’s capabilities to process, analyze, and generate responses based on complex inputs. These tasks are designed to mimic human-like thinking, enabling the model to solve problems, explain con-

cepts, and draw inferences. Typically, large language models handle reasoning tasks that deal with deductive reasoning, inductive reasoning, abductive reasoning, analogical reasoning, and several other reasoning decisions[17]. Some challenges with large language models handling vague or incomplete inputs are trying to avoid skewed or ethically questionable outputs. As well to manage tasks that involve multi-step reasoning or long-term dependencies.

To enhance reasoning capabilities, the adoption of chain-of-thought prompting has proven to be a breakthrough strategy. This technique encourages models to articulate intermediate steps in reasoning processes, mimicking human-like logical progression[17]. By iteratively refining their approach to a problem, large language models (LLMs) can mitigate the risks associated with generating incomplete or inaccurate outputs. This process is especially useful in areas like mathematical problem-solving, ethical reasoning, or nuanced decision-making, where the correct solution often lies in the journey rather than just the endpoint[6]. Advancements in fine-tuning and system updates have focused on aligning LLMs with user needs while minimizing biases and errors. Regular updates incorporate the latest research and user feedback to create a more robust system, capable of handling the ever-growing complexity of tasks[6]. By integrating principles of explainability and transparency, these updates aim to make LLMs not only more efficient but also more trustworthy.

A significant part of improving reasoning lies in context management. For instance, LLMs must balance processing immediate user inputs with broader contextual understanding, which requires retaining relevant information across interactions[7]. Techniques such as attention mechanisms and memory modules have shown promise in addressing this need. Moreover, as tasks grow in complexity, the ability to evaluate risk-reward scenarios dynamically becomes crucial. This capability enables LLMs to weigh potential outcomes and prioritize solutions that maximize user satisfaction or align with ethical considerations.

Ultimately, the interplay between structured reasoning strategies like chain-of-thought prompting and technological advancements in system updates defines the trajectory of LLMs[14]. By addressing current challenges, including vagueness and multi-step reasoning dependencies, these systems edge closer to achieving human-like cognitive performance in reasoning and decision-making tasks[24]. As research into LLM reasoning capabilities progresses, it becomes increasingly evident that models must go beyond pattern recognition and surface-level responses to engage in deeper, more structured thinking. The ability to effectively handle ambiguity, process complex logical relationships, and adapt to evolving inputs is a critical factor in determining whether LLMs can truly function as advanced reasoning agents[1]. One of the most promising avenues for this evolution is the integration of reinforcement learning techniques alongside chain-of-thought prompting. By allowing models to “learn” from previous interactions and adjust their responses accordingly, reinforcement learning introduces an element of iterative improvement that mirrors human learning processes[12].

A key challenge in reasoning tasks is the ability to recognize and reconcile conflicting information. In real-world applications, LLMs often encounter incon-

sistencies in data, requiring them to resolve contradictions and determine the most reliable conclusions [4]. By leveraging techniques such as self-consistency decoding — where multiple independent reasoning paths are generated and compared — models can improve their decision-making accuracy [22]. This method reduces the susceptibility to misleading or biased inputs and encourages more balanced and informed outputs. The scalability of LLM reasoning also plays a key role in assessing their viability in the real world. While smaller, task-specific models may excel in niche areas, general-purpose LLMs must manage large amounts of knowledge while remaining efficient at the reasoning task [25]. This balance involves not only optimizing response accuracy but also ensuring that the model remains responsive and interpretable. Researchers have explored modular LLM architectures, in which specialized submodels address different aspects of reasoning (e.g., probability estimation, ethical considerations, and strategic planning) and collaborate to produce cohesive and insightful conclusions [24].

Another important consideration is the ethical dimension of reasoning in LLMs. As LLMs become more autonomous in their decision-making, ensuring that their reasoning processes align with human values and ethical principles is critical [2]. Bias mitigation strategies, fairness assessments, and transparency measures must continue to be refined to avoid unintended consequences in LLM-based reasoning systems. The future of reasoning in LLMs depends on finding a balance between autonomy and accountability, ensuring that models produce conclusions that are reliable, explainable, and ethically sound [7]. Research will play a critical role in advancing thinking in LLMs. By combining insights from cognitive science, neuroscience, and behavioral economics, LLM researchers can develop more sophisticated reasoning paradigms that replicate human cognitive strengths while mitigating common pitfalls [14]. As LLMs mature, their potential extends far beyond traditional Natural Language Processing applications — offering transformative possibilities in areas such as scientific discovery, legal analysis, and complex decision support systems [4]. The ability to effectively process complex reasoning tasks will define the next generation of artificial intelligence, pushing the boundaries of what an LLM can achieve in solving real-world problems.

1.3 Goals of the Project

As we read into the more studies of large language models, many studies have tested around the central idea of reasoning tasks and test these models[17]. But in this study we come to try and test as well if the reasoning task can be explained by trying to test and see if the memory is being used over the course of one to two months. Blackjack is the best suited reasoning task for testing large language models because it combines probabilistic reasoning, decision making, and understanding of rules within a dynamic context[12]. The game involves calculating probabilities based on visible cards, anticipating outcomes, and balancing risk versus reward elements of strategic reasoning. Additionally, it requires the model to follow structured rules, adapt its decisions to varying

scenarios, and potentially consider human psychology depending on previous outcomes[12]. This makes blackjack a test for evaluating an LLM’s ability to apply logical reasoning, process numerical relationships, and simulate decision-making under uncertainty which are skills for complex problem solving.

Expanding blackjack selection as a reasoning task, the game’s multifaceted complexity reflects real-world decision-making scenarios. The ability to consider probabilities while adhering to a strict set of rules tests a model’s ability to balance logical inference with strict structural guidelines[6]. The interplay between adaptability and compliance makes blackjack especially suitable for testing large language models[17]. For example, the decision to hit, hold, double, or split depends not only on short-term probabilities but also on long-term outcomes – which requires models to simulate forward-thinking strategies and compare potential paths[12]. This element of dynamic decision making is critical to evaluating how well a large language model can reason through sequential tasks that require short-term calculations and long-term planning.

Another dimension of blackjack as a test of reasoning lies in its essentially incomplete information. The cards held by the dealer introduce an element of uncertainty, forcing the model to make predictions based on partial data. This aspect evaluates the model’s ability to resolve ambiguities and use probabilistic reasoning[7]. Scenarios like these mimic real-world challenges, where complete information is often not available and decisions must be based on estimates. By analyzing how large language model processes and synthesizes these uncertainties, researchers can gain insights into the model’s potential application in areas such as risk assessment, financial forecasting, and strategic planning[12].

Additionally, extending testing over one to two months provides a unique opportunity to explore the model’s memory capabilities. By introducing a longitudinal element – where the results and decisions of previous games can inform future actions – researchers can examine how effectively large language models retain, retrieve, and apply knowledge learned over time[24]. Memory, in this context, becomes an important component in reasoning, allowing the model to refine its strategy and adapt to the patterns identified in previous rounds. This approach aligns with a broader goal in AI development: creating systems that not only perform tasks in isolation, but also develop understanding and behavior through experience.

The iterative nature of blackjack also makes it an excellent platform for evaluating the ability of large language models to generalize past experience[9]. By observing how a large language model’s strategy evolves over multiple sessions, we can assess whether the model is capable of detecting patterns, adapting to different strategies used by virtual retailers, and improving its results over time. The ability to generalize and adapt is at the heart of intelligent reasoning because it indicates the ability to move beyond static, rule-based processing toward dynamic, context-sensitive problem solving[17]. Assessing generalization is critical for applications in areas such as adaptive learning, autonomous systems, and decision-making tools, where the ability to learn and improve in a variety of environments is critical.

In addition to testing probabilistic reasoning and adaptability, blackjack

offers a unique opportunity to evaluate how effectively large language models can simulate human-like reasoning processes[9]. Decision-making in the real world often involves balancing objective probabilities with subjective judgments influenced by prior outcomes, emotions, or biases. Although large language models lack human emotions, their ability to mimic different decision-making patterns in blackjack – such as avoiding overly risky plays after consecutive losses – can provide insight into how close the models are to human behavior[9]. This aspect of testing can be useful for applications where large language models need to interact with humans, such as virtual assistants, customer service, or collaborative problem-solving scenarios[9].

1.4 Ethical Implications

Exploring large language models in risk-reward tasks such as blackjack introduces several ethical considerations, reflecting the broader responsibility to evaluate the social impact of large language models[20]. Although this project primarily focuses on testing reasoning skills, it is important to address potential risks and ensure results are in line with ethical standards. The use of large language models often involves processing data provided by users, raising questions about the privacy of the information. In scenarios where models are trained or tested using user interactions, there is a risk of accidentally collecting sensitive information[17].

A major concern with large language models is ensuring the accuracy and reliability of the information they produce. In risk-reward tasks, inaccurate reasoning or unreliable predictions can result in suboptimal decisions, thus interfering with the assessment of the model’s capabilities[20]. To address this, the project will include a rigorous validation process to ensure that model decisions are aligned with established probabilistic reasoning strategies and principles. Any limitations in reasoning will be clearly documented to avoid misinterpretations of the model’s capabilities and also examined to see if the response is an attempt at the large language model trying to reason or not.

Although this project is designed for academic and research purposes, developing reasoning capabilities in large language models can be misused, such as automating gaming strategies for financial exploitation[12]. Unintended consequences may also include an increase in addictive behavior if the model’s insights are inappropriately applied. To minimize this risk, this investigation will emphasize transparency and include a disclaimer that the results are for educational purposes only. Efforts will also be made to avoid creating systems that could directly facilitate gaming activities[20]. Furthermore, data from this experiment will not be alarmed on an economical gain or loss but rather on a win/loss data to see the results.

Large language models often reflect biases present in their training data, which can manifest themselves in reasoning tasks. For example, biases in the way a model assesses risk or interprets results can lead to unfair or flawed strategies[17]. To address this, testing will be performed using a variety of scenarios to identify and document any distortions in the model’s reasoning

process[17]. These findings will inform future improvements, ensuring that the strategies resulting from this model are impartial and fair. The implementation of a model capable of considering risk and reward scenarios may, indirectly, pose risks to third parties, such as business actors, the business world or regulators. For example, a model that optimizes blackjack strategy may influence casino operations or player behavior in undesirable ways[12]. To mitigate these risks, this project will focus only on theoretical testing and not practical application and will include clear limitations on how results are disseminated or applied.

To address these ethical concerns, the project incorporates several mitigation measures, including the use of synthetic data, transparent documentation of model limitations, and rigorous bias testing [20]. Furthermore, the project will emphasize the educational and exploratory nature of the work, preventing misuse, and encouraging the responsible development and implementation of comprehensive language models. By anticipating potential harms and implementing precautions, the research aims to contribute to a broader understanding of comprehensive language models while maintaining ethical standards. In addition to the immediate ethical concerns, another key consideration is the long-term societal impact of developing LLM reasoning in risk-reward scenarios. As these models become more capable of making probabilistic decisions, there is a growing need to ensure that they are applied in ways that benefit society rather than exacerbate existing inequalities [8]. For example, if LLMs develop sophisticated strategies for blackjack and similar decision-based games, their capabilities could extend to financial markets, insurance modeling, and other high-risk sectors where ethical considerations play a critical role. Ensuring that these models are designed with safeguards to prevent abuse in high-risk sectors is an ongoing challenge that researchers and policymakers must address [12]. Another ethical dimension to consider is the role of human oversight in LLM-guided reasoning. While this study focused on blackjack as a controlled environment to test LLM decision-making, the principles explored here can be extended to LLM applications in legal reasoning, medical diagnosis, and automated decision systems [25]. In these contexts, transparency and explainability are paramount. The ability of a model to articulate its reasoning in a comprehensible manner ensures that users can critically evaluate its conclusions, rather than blindly trusting automated results [20]. Future research should prioritize not only improving the accuracy of decision-making in LLMs, but also making their reasoning processes more interpretable to human users.

Furthermore, the potential psychological impact of gambling strategies in LLMs should be carefully examined. While this research is purely exploratory in nature, the knowledge gained could be used to improve LLM-based gaming tools, potentially leading to improvements in problematic gaming behavior [13]. Ethical LLM research should be aware of unintended consequences and take steps to mitigate harm, such as promoting responsible use of LLMs and collaborating with regulatory bodies to establish guidelines for LLM application in gaming and risk assessment [12]. This study will maintain its cautious approach, ensuring that its findings are framed in a way that does not encourage exploratory applications of LLM-based decision making. Bias mitigation efforts

should also be an ongoing priority. LLMs learn from large datasets that may contain inherent biases, and these biases can influence risk assessment strategies in subtle but impactful ways [17]. If LLMs disproportionately favor aggressive betting strategies or miscalculate risk due to biases in historical training data, this could lead to erroneous conclusions. To address this, researchers will conduct extensive bias testing across a range of game conditions and analyze whether the model’s decision-making is consistent with established, unbiased probabilistic strategies [24]. Additionally, results will be reported transparently, detailing any biases found and outlining steps for future improvements in fairness and accuracy.

The ethical considerations surrounding data privacy and security cannot be overlooked. While this study does not rely on personal data, the broader implications of LLMs in decision-making scenarios must be considered. LLMs interacting with users in real-world risk-reward contexts could inadvertently collect sensitive behavioral data, raising concerns about data protection and informed consent[17]. To prevent future ethical dilemmas, it is essential to advocate for robust privacy policies, ensuring that LLMs used in decision-making tasks prioritize user anonymity and data security. This research aims to responsibly explore the capabilities of LLMs in risk-reward scenarios while addressing ethical concerns related to bias, transparency, potential misuse, and societal impact. By taking a proactive approach to identifying and mitigating risks, this study contributes to the responsible advancement of LLMs reasoning while emphasizing the importance of ethical safeguards in future LLM-driven decision-making systems[20].

Moreover, the psychological and societal effects of LLM-driven decision-making extend beyond gambling. The ability of LLMs to analyze vast amounts of data and generate optimized strategies has implications for various risk-reward scenarios, including stock trading, loan approvals, and even legal decision-making. If left unchecked, LLM models trained on flawed or biased datasets could inadvertently reinforce harmful patterns, disproportionately impacting vulnerable populations[10]. For instance, an LLM that learns risk assessment strategies based solely on historical market trends may encourage speculative trading behaviors, leading to financial instability for inexperienced investors. Therefore, ethical LLM research must incorporate interdisciplinary perspectives, including behavioral psychology, financial ethics, and regulatory compliance, to ensure that LLM-driven decision-making aligns with human well-being and fairness[16].

As LLMs become more integrated into everyday decision-making, transparency of the strategies they generate should be a top priority [19]. Users who interact with LLMs—whether in gambling, financial investment, or other strategic domains—must have a clear understanding of how these systems work and the logic behind their recommendations. Developing an explainable LLM framework will be critical to addressing this issue. By making LLM decision-making processes more interpretable and accessible, researchers can help prevent over-reliance on LLM generated strategies while promoting critical thinking and informed decision-making among users. This study will contribute to this effort by

comprehensively documenting and publishing its findings, ensuring that the role of LLMs in strategic decision-making remains transparent and accountable[10].

Furthermore, interdisciplinary collaboration is essential to mitigate the potential risks of LLM-based gambling tools and other decision support systems. Regulatory bodies, LLM researchers, behavioral scientists, and policymakers should work together to establish clear ethical guidelines and industry standards. These guidelines should not only address the fairness and accuracy of LLMs, but also implement safeguards against potential abuses such as reinforcement of addiction, unfair advantages in gaming, or manipulation of user behavior based on data. LLMs should be designed to empower users with knowledge and support responsible decision-making, rather than serve as opaque tools that can be exploited to cause financial or psychological harm[12].

Moving forward, responsible LLM development must prioritize continuous assessment and improvement. Bias audits, impartiality assessments, and user impact studies should become standard practice in LLM research, ensuring that LLM is developed in ways that benefit society rather than exacerbate existing inequities or ethical issues[19]. LLM-driven decision-making has the potential to revolutionize many industries, but its implementation must be done with care and integrity. By integrating a robust ethical framework into LLM research and development, we can ensure that LLM technologies positively impact human decision-making, rather than creating unintended risks and vulnerabilities.

2 Related work

Early systems relied on predefined rules to simulate reasoning processes but lacked adaptability to complex, dynamic environments. The advent of machine learning introduced statistical reasoning methods, enabling systems to infer patterns from data rather than rely solely on fixed rules. More recently, large language models like GPT-4, Claude, and Gemini have demonstrated emergent reasoning capabilities, thanks to their training on vast datasets and sophisticated architectures. Researchers test these large language models by using arithmetic reasoning tasks and seeing if the large language models can learn to solve the task by learning robust generalizable algorithms, or do they are able to memorize training data from previous results [17].

Large language models rely heavily on prompting techniques to guide their responses and reasoning processes, with methods like zero-shot prompting, few-shot prompting, and chain-of-thought prompting playing critical roles [23]. Zero-shot prompting involves providing minimal input to generate a response, leveraging the model’s general training to infer context. Few-shot prompting enhances this by including examples in the prompt to steer the model toward desired patterns or formats [24], improving the specificity and relevance of its output. Chain-of-thought prompting, a more recent advancement, allows large language models to break down complex problems into intermediate steps, mimicking human step-by-step reasoning [23]. This method has shown significant promise in tasks requiring multi-step logic, arithmetic, or strategic planning, as it enables the model to connect ideas and justify its decisions more effectively. These prompting techniques are pivotal in helping large language models simulate reasoning processes, making them essential tools for exploring the boundaries of AI cognition.

Despite their effectiveness, prompting techniques also reveal certain limitations of large language models, particularly when applied to reasoning tasks. Many existing studies highlight that while chain-of-thought prompting can improve performance, it often depends on the quality of the prompt itself, requiring careful crafting by users [24]. This dependency underscores a broader issue: the inability of large language models to consistently generate coherent reasoning without external guidance. Furthermore, vague or incomplete prompts can lead to unreliable outputs, challenging the model’s ability to handle uncertainty autonomously. The integration of risk-reward scenarios, such as blackjack, offers a unique opportunity to test whether chain-of-thought prompting and other methods can enable large language models to develop adaptive strategies, make probabilistic decisions, and balance short-term and long-term goals. By assessing how large language models respond to these challenges, this research aims to uncover the extent to which prompting frameworks can enhance their reasoning capabilities and adaptability in complex, dynamic environments [22].

Large language models rely heavily on nudge techniques to guide their responses and reasoning processes, with methods such as zero shot and chain-of-thought playing an important role. The zero-shot request involves providing minimal information to generate a response, leveraging general model training

to infer context. Multi-step prompts improve this by including examples in the prompt to guide the model toward the desired pattern or format, thereby increasing the specificity and relevance of your output. Encouraging chain thinking [17], a more recent advancement, allows large language models to break down complex problems into intermediate steps, mimicking human-like step-by-step reasoning. This method shows promise in tasks that require logic, arithmetic, or multi-step strategic planning, as it allows the model to connect ideas and justify its decisions more effectively.

Despite their effectiveness, stimulus techniques also reveal certain limitations of LLM, especially when applied to reasoning tasks. Many studies highlight that although CoT commands can improve performance, this often depends on the quality of the command itself, requiring caution on the part of the user [24]. This dependency highlights a broader problem: the inability of the LLM to consistently produce coherent reasoning without external guidance. Additionally, unclear or incomplete queries can produce unreliable results, challenging the model’s ability to handle uncertainty independently. The integration of risk-reward scenarios such as blackjack provides a unique opportunity to test whether CoT stimuli and other methods can enable LLMs to develop adaptive strategies, make probabilistic decisions, and balance short- and long-term objectives. By assessing how LLMs respond to these challenges, this research aims to reveal the extent to which an encouraging framework can improve reasoning skills and adaptability in complex and dynamic environments [18].

Exploring nudge techniques such as chain-of-thought in reasoning tasks reveals the potential of large language models to engage in more structured and logical problem solving. The chain-of-thought incentive encourages the model to outline intermediate steps before arriving at a final answer, thus allowing the model to simulate deliberative reasoning [17]. This approach is especially useful in tasks such as mathematical problem solving, logical deduction, and even strategic games, where multi-step reasoning is important. By breaking down complex problems into smaller, more manageable components, chain-of-thought boost not only improves accuracy but also provides transparency into the model’s decision-making process. This transparency is especially important for applications where users need to trust the reasoning behind the model output, such as in educational or analytical environments [18].

The ability of LLMs to justify their reasoning in a human-understandable manner will be critical to their widespread acceptance in high-stakes decision-making environments. Incorporating counterfactual reasoning, in which the LLM explains not only why it reached a particular decision but also what would have happened if different inputs had been provided, could significantly increase user confidence[11]. This approach would enable stakeholders—whether in the legal, financial, or medical fields—to better understand the causal relationships underlying the LLMs predictions and recommendations. Future research should explore methods for integrating these reasoning capabilities into LLMs, ensuring that their decision-making processes remain auditable and interpretable by non-expert users.

Another important area for improvement is the dynamic adaptation of LLM

reasoning in real-time decision making. Current models typically rely on static datasets and pre-defined heuristics, which may not always be in line with evolving real-world conditions[21]. Integrating a continuous learning framework, where LLMs refine their reasoning strategies based on real-time feedback and performance assessments, can fill this gap. For example, in financial markets or medical diagnostics, an LLM must be able to adapt its decision making based on new economic trends or the latest clinical research. Implementing a safe and controlled reinforcement learning mechanism, where the LLM system improves its reasoning through a structured trial-and-error process while respecting ethical constraints, can improve its adaptability to the real world without compromising reliability or fairness[10].

Interdisciplinary collaboration will be critical to advancing reasoning methodologies in LLM. Insights from cognitive science, behavioral economics, and decision theory can inform the development of LLM systems that better reflect human reasoning processes[15]. By incorporating psychological models of judgment and decision-making, LLM researchers can refine LLM to more accurately mimic human critical thinking, including aspects such as uncertainty management and ethical considerations. A multidisciplinary approach will also help address key challenges in LLM reasoning, such as mitigating confirmation bias, overfitting to training data, or misinterpreting ambiguous information.

Moving forward, an important consideration for the development of reasoning in LLMs is the establishment of standard parameters for assessment. Unlike traditional machine learning tasks, where accuracy metrics are well-established, reasoning is inherently more complex to measure[6]. The creation of a robust assessment framework that evaluates LLM reasoning ability across multiple dimensions—such as logical coherence, factual accuracy, impartiality, and adaptability—will be critical to ensuring meaningful progress. Future research should prioritize the development of these parameters to facilitate transparent comparisons across different LLMs and reasoning methodologies. Improving reasoning capabilities in LLM is not only about increasing computing power, but also about aligning decision-making in LLM with human values, ethical standards, and real-world complexities[8]. By combining advances in LLM interpretability, dynamic adaptation, interdisciplinary insights, and rigorous assessment standards, the future of LLM-based reasoning can lead to more reliable, transparent, and socially responsible LLM systems.

However, reliance on carefully designed tracks poses significant challenges. Large language models often struggle when faced with ambiguous or poorly constructed instructions, which can produce illogical, inconsistent, or irrelevant results. This dependency highlights the importance of user experience in effectively using this model, creating potential barriers for non-technical users. Furthermore, although encouraging chain-of-thought can improve reasoning [24], it does not inherently address context retention or memory issues in interactions. In a risk-reward scenario like blackjack, for example, the model must not only process each hand, but also adjust its strategy based on cumulative results – a challenge that requires the integration of dynamic memory and learning mechanisms, which currently it's out of reach for most people. pushing technique.

Therefore, the development and refinement of driving strategies must overcome these limitations to unlock the full potential of large language model in reasoning tasks. Future advances could include automated fast generation to reduce user dependencies, improved memory capabilities to handle long-term dependencies, and adaptive frameworks that allow models to learn from feedback in a single session. Testing these approaches in a structured environment, such as a risk-reward scenario, offers valuable insights into the intersection between motivation, reasoning, and adaptability [6]. By pushing the limits of what large language model can achieve with minimal human intervention, this research contributes to a deeper understanding of how these models can simulate complex reasoning processes and adapt to dynamic real-world tasks.

Expanding the role of drivers, their evolution highlights the interaction between user input and model capabilities. Techniques such as encouraging chain-of-thought show that large language model can simulate reasoning when equipped with a structured framework, effectively demonstrating latent problem-solving skills [17]. However, this structure relies heavily on user-supplied external scaffolding. This dependence raises questions about the extent to which large language model or models truly “reasons” rather than simply reproducing patterns learned during training. In risk-reward scenarios like blackjack, the success of a chain-of-thought request depends on your ability to guide the model to deal with uncertainty, adapting to changing probabilities and making decisions based on incomplete information. Testing how well these techniques support reasoning in such contexts helps to elucidate the strengths and limitations inherent in nudge-based approaches.

Furthermore, the ethical implications of nudges must be considered, as they shape how large language models interact with users and the types of results they produce. Poorly crafted requests or an opaque thought process can result in unintended consequences, such as the production of biased, misleading, or harmful content. In scenarios that involve decision-making under conditions of uncertainty, this risk is increased, as faulty reasoning can propagate errors or reinforce existing prejudices. To mitigate these risks, it is necessary not only to improve warning methods, but also to develop users’ literacy in preparing effective warnings. By combining advances in model design with a user-centered approach, future driving strategies can ensure that LLM remains a reliable, fair and effective tool for thinking in structured and unstructured environments[4]. Through this research, we aim to contribute to the refinement of this strategy, paving the way for a more robust and adaptable application of LLM technology.

As large language model capabilities progress, the application serves as an important bridge between inherent potential and practical application in solving complex problems. Iterative improvements to the boost strategy not only improve model performance but also expand the range of tasks these systems can perform. In a dynamic, probabilistic environment like blackjack, nudging techniques like chain-of-thought provide a unique lens to evaluate how well large language models can replicate human-like adaptive reasoning. However, it also highlights the importance of developing strong cues to ambiguity and being able to generate reliable reasoning in a variety of scenarios [23]. This research

not only seeks to test the limits of current incentives, but also aims to provide knowledge that informs the next generation of tools and methodologies, encouraging the development of large language models that are not only smarter, but also more accessible and ethically sound.

Expanding the scope of reasoning strategies in large language models (LLMs) requires a deeper exploration of their limitations and potential improvements. While prompting techniques such as chain-of-thought have proven effective in enhancing structured reasoning, their success remains contingent on the user’s ability to construct meaningful prompts[23]. This reliance introduces a bottleneck in usability, particularly for non-expert users who may struggle to craft precise instructions that guide the model toward optimal reasoning. Future advancements in automated prompting mechanisms may address this issue, allowing LLMs to refine their reasoning processes autonomously without heavy user intervention. One possible approach involves dynamic prompt engineering, where models iteratively refine their responses based on feedback or self-evaluation, ensuring more robust and contextually appropriate reasoning[24]. Additionally, the exploration of reinforcement learning techniques in conjunction with prompting strategies presents an opportunity to enhance LLM decision-making capabilities. Current models largely operate in a static, one-off response paradigm, but integrating reinforcement learning from human feedback (RLHF) or self-play mechanisms could allow LLMs to refine their decision-making over iterative interactions[5]. In the context of blackjack or other risk-reward scenarios, this would mean the model could adapt its strategies over time, improving its ability to evaluate probabilistic outcomes and optimize for long-term rewards. Such an approach would shift LLMs from passive pattern recognizers to more dynamic agents capable of learning from experience, further blurring the line between artificial and human-like reasoning.

Moreover, LLMs’ ability to generalize learned reasoning strategies across different domains remains an open question. While chain-of-thought prompting aids in structured problem-solving, its applicability across diverse reasoning tasks—ranging from arithmetic to strategic gameplay—needs further empirical validation[14]. The degree to which LLMs can transfer reasoning skills learned in one domain (e.g., probabilistic decision-making in blackjack) to another (e.g., financial risk assessment) will determine their long-term value as adaptive reasoning systems. To explore this, comparative studies across multiple domains should be conducted, assessing whether prompting frameworks allow LLMs to develop transferrable reasoning skills or if they remain domain-specific in their problem-solving abilities. Another critical aspect to consider is the balance between deterministic and probabilistic reasoning in LLMs. Traditional logic-based AI systems excel in deterministic environments where rules are explicitly defined, while probabilistic models leverage uncertainty to make informed predictions[12]. LLMs, positioned at the intersection of these paradigms, must effectively navigate scenarios that require both structured logical deduction and probabilistic inference. In blackjack, for instance, a model must decide whether to hit or stand based on both deterministic rules (e.g., card values) and prob-

abilistic assessments (e.g., likelihood of drawing a favorable card). Evaluating how well LLMs can integrate these reasoning styles will be instrumental in understanding their potential for real-world decision-making applications.

In addition to improving reasoning methodologies, ensuring the interpretability of LLM-generated reasoning is paramount. One of the persistent criticisms of LLMs is their “black-box” nature, where even seemingly logical responses lack explicit justification[4]. The integration of explainability frameworks, such as self-explanatory prompting or structured decision trees, can help address this issue by forcing the model to articulate its reasoning in a clear and step-by-step manner. For applications where trust and transparency are crucial—such as legal decision-making, medical diagnostics, or financial risk assessments—this capability will be essential. Looking ahead, advancements in memory-augmented architectures may further enhance LLMs’ ability to handle reasoning tasks that require long-term dependencies[13]. Current LLMs struggle with maintaining context across extended interactions, which limits their effectiveness in scenarios that require cumulative decision-making. By integrating memory mechanisms—such as external memory stores or transformer-based recurrence—future iterations of LLMs could maintain strategic consistency over multiple steps, enabling more coherent and informed decision-making in risk-reward scenarios.

Ethical considerations must continue to guide the development of LLM reasoning capabilities. While enhancing AI-driven reasoning holds promise for various applications, safeguards must be in place to prevent misuse in high-risk domains such as automated gambling strategies, financial market manipulation, or decision-making systems with legal and ethical implications[7]. Responsible AI development should prioritize transparency, user education, and regulatory compliance to mitigate the risks associated with AI-powered reasoning systems. By addressing these challenges and opportunities, this research contributes to a broader understanding of how LLMs can simulate structured reasoning, adapt to dynamic environments, and provide reliable decision-making support across a range of applications. As AI technology evolves, continued experimentation with prompting techniques, reinforcement learning, probabilistic modeling, and explainability frameworks will be key to unlocking the full potential of LLM reasoning in complex, real-world scenarios.

3 Method of approach

3.1 Models

To ensure the success of the experiment, it is important to use the most advanced and appropriate large Language Model (LLM) available in production. The selection process involves careful consideration of various models based on performance, popularity, and ability to handle complex reasoning tasks effectively. At the end of September, the most suitable LLMs identified and used for this experiment were OpenAI o1 [1], Gemini 1.5 Pro [3], and Claude 3.5 Sonnet [2]. These models were selected for their broad applicability in real-world applications, their proven capabilities in a variety of reasoning tasks, and their overall robustness in producing high-quality results.

When looking at all the models of choice, OpenAI o1 is known for its contextual understanding and strong adaptability [1]. This model has been widely recognized for excelling in structured reasoning, creative problem solving, and language understanding. With Gemini 1.5 Pro, it is powered by Google DeepMind and is highly effective in multimodal tasks while demonstrating high accuracy in logical reasoning and analytical problem solving [3]. As well, Claude 3.5 Sonnet is designed for reliable and secure AI interactions. It is strong in nuanced reasoning, maintains long and coherent discussions, and handles complex questions efficiently [2]. The experiment aims to optimize the quality of the results while ensuring that the selected model can meet the stringent demands of the risk reward reasoning tasks. Its implementation provides a strong foundation for achieving experimental goals and maximizing overall efficiency.

While running this experiment, other models such as Deep Seek were released but with the process beginning we made the decision to not use it as it as well not gained popularity in America as it is a Japanese product. Furthermore with the models, to avoid servers crashing and being overloaded. It was necessary to run at night, specifically starting at 8 p.m. because Claude and Gemini raised the issues of having been overloaded.

3.2 Blackjack

Blackjack, also known as 21, is a popular card game played between one or more players and a dealer. The objective of the game is to get a card that is closer to 21 than the dealer’s card without going over (busting). Each player is dealt two cards, usually face up, while the dealer is dealt one card face up and one card face down (the “hole” card). Cards 2 through 10 are worth their face value, face cards (Jack, Queen, King) are worth 10, and Aces can be worth either 1 or 11, depending on which value is most favorable to the player’s hand. Players can choose to “hit” (take another card), “stand” (keep the current hand), “double” or “double down” (double the bet and receive a final card), or “split” (if they have a pair, they can split it into two separate hands). Play continues until all players have acted, after which the dealer reveals his hole cards and plays according to the established rules—usually hitting until he reaches 17 or more.

Winning at blackjack involves having a higher card total than the dealer without going over 21, or having the dealer bust. A player who is dealt an Ace and a 10-value card in his or her first two cards has “blackjack,” which is usually an automatic win with a higher payout (usually 3:2). If the player and the dealer have cards of the same value, it is a “tie” or a “push” and the player’s bet is returned. Basic strategy in blackjack is based on probability and helps the player decide the statistically optimal move for each possible combination of cards and dealer. Because these games combine elements of luck and decision-making, they are often studied in the context of risk management, game theory, and artificial intelligence [15].

Blackjack is considered an efficient reasoning task because it involves a clear and well-defined set of rules and probabilities that allow strategic decision-making within a limited and manageable state space. Players must constantly evaluate their hands against the dealer’s upcard, estimating the probability of drawing a favorable card versus the risk of bust[8]. It requires basic probabilistic reasoning and decision-making under uncertainty, but the number of possible actions—hit, stand, double, split—is small and limited by the structure of the game. Furthermore, card values are simple and outcomes are immediate, allowing for rapid feedback and learning. As a result, blackjack is an ideal setting for testing and modeling reasoning strategies, both for the study of human cognition and artificial intelligence, because it balances complexity and tractability without overtaxing computational resources.

The idea of using blackjack as the reasoning task for the experiment over others is to see if the reason is based around the memory of previous wins and losses. Other real world games such as chess and other games also involve reasoning, but often rely more on long-term planning, pattern recognition, and structured decision making than on immediate memory of past outcomes. Blackjack offers a unique environment where reasoning can be tested with respect to short-term memory, risk assessment, and probabilistic thinking. Because this game involves decisions based on recent card history, past wins and losses, and calculated risks, it serves as an ideal experiment to determine whether reasoning is influenced by memories of past experiences and not pure logical deduction. Additionally, blackjack allows controlled variation in decision-making scenarios, making it easier to isolate memory-based reasoning from other cognitive factors, such as strategic depth or opponent behavior, which are more prominent in games such as chess or poker[6]. Blackjack supports strategy optimization through a variety of familiar tools, such as “basic strategy graphs,” which encode statistically optimal decisions for each card combination. This graph shows that while the game involves uncertainty, it can still be governed through systematic reasoning. Most importantly, feedback is immediate—players can quickly see the results of each action—which allows for rapid learning and adaptation.

For LLM and cognitive science, this makes blackjack a useful testing ground for studying how agents learn, plan, and make decisions efficiently in uncertain environments without requiring large amounts of computation or memory. Blackjack requires players to make optimal decisions based on incomplete information using logical, probabilistic, and often heuristic-based reasoning. The

game presents a dynamic environment where the player must interpret the total number of cards they currently hold, the dealer’s visible card, and the distribution of the remaining cards in the deck (especially in card counting scenarios). However, the decision space is still relatively small compared to more complex games such as poker or chess, so the problem is computationally efficient but intellectually interesting. Players must consider the expected value of possible actions—such as hitting or standing—by assessing the probability of drawing a card that will improve or worsen their hand. This creates a clear connection between perception, calculation, and action[10].

Key tools in use of project: - SQLite - LiteLLM - OpenAI o1 - Claude 3.5 Sonnet - Gemini 1.5 pro - JSON

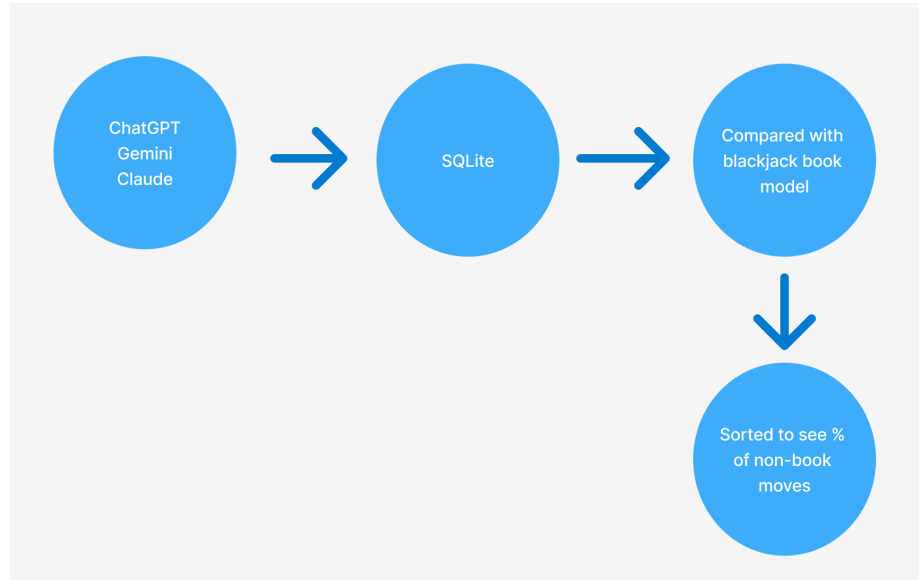


Figure 1: Process of Experiment

Blackjack plays an important role in systems design and experimental research, especially in areas such as artificial intelligence, cognitive science, reinforcement learning, and decision theory. As a finite but nontrivial environment with a mix of stochastic outcomes and strategic decision-making, blackjack serves as an ideal testing ground for designing, validating, and comparing reasoning systems. In systems design, blackjack allows developers to model environments where agents must act under uncertainty and partial information. The structure of the game—sequential decisions, clear rules, and measurable outcomes—makes it easy to simulate, monitor, and tune. This helps test components such as decision policies, value estimation functions, or learning algorithms under controlled conditions.

In experimental settings, blackjack is often used to study how agents (humans and machines) learn from experience, deal with risk, or balance explo-

ration and exploitation [21]. For example, in reinforcement learning research, blackjack is a canonical problem used to demonstrate basic concepts such as policy iteration. The simplicity of the game allows for rapid iteration and analysis of agent behavior, learning rate, and convergence. Similarly, in cognitive psychology, blackjack is used to investigate how humans make decisions under pressure, how they develop or abandon optimal strategies, and how biases or heuristics influence their play. Blackjack provides a balanced experimental environment that is complex enough to be meaningful, yet simple enough to allow for repeatability, interpretability, and scalability—essential qualities for systems design and scientific experimentation[21].

At first for preliminary test, the method to create this project was with a physical six shoe of cards. Now that has changed to an automated game that runs through with an LLM making decisions. An automated game is efficient because it structures the gameplay optimally while ensuring minimal computational overhead. The `play_round` function is designed to handle the entire round of the game using a well-organized sequence of moves. This eliminates the need for redundant queries to retrieve the same data multiple times, thereby reducing database access times and increasing efficiency. Function calls also ensure that we extract only the necessary data, avoiding unnecessary iterations across the data set. Another fundamental aspect of efficiency in this approach is the modularity of functions [6]. Separating `play_player`, `play_dealer`, and `play_round` keeps game logic structured and reusable. Instead of combining all the logic into one function, which can make debugging and maintenance difficult, the gameplay is divided into different, manageable functions. This modular design makes it easier to test individual components, increases maintainability and reduces the chance of errors. Additionally, functions such as `hand_value` and `deal_card` ensure that redundant calculations are avoided as they are optimized to provide results in minimal operations.

3.3 Code

Prompts play a critical role in the success of this project, as they directly influence how well the large language model (LLM) interprets the game scenario, understands the rules, and produces the right decisions. The structure, clarity and specificity of commands determine the quality of the response, ensuring that the AI can accurately assess the state of the game and make strategic choices based on probability, risk assessment and optimal blackjack strategy. Well-crafted prompts not only provide LLM with important game details – such as table rules, possible actions, and players’ current hands – but also guide the model toward logical, legally sound decisions. This ensures that the AI stays within the bounds of the game, preventing it from making impossible moves, misunderstanding card values, or failing to recognize important strategic considerations such as insurance, doubling down, or surrender. Additionally, well-structured commands standardize responses across AI models, allowing for more consistent and reliable assessment of reasoning patterns [16].

On the other hand, poorly designed alerts can create ambiguity, leading

to inconsistent and illogical decision making. If underlying rules or strategic choices are missing, the model may choose suboptimal options, take illegal actions, or fail to provide proper reasoning. Incomplete commands can also cause variations in model behavior, making it difficult to compare different LLMs objectively. By repeating and refining commands, we not only improve the accuracy of responses, but also the depth of reasoning displayed by the AI[10]. Strong command prompts the model to clearly articulate its decision-making process, providing information about whether its reasoning is based on memory, probability calculations, or fundamental blackjack strategy. This iterative approach helps create a robust and measurable framework for evaluating AI performance on reasoning-based tasks, ultimately improving the reliability and effectiveness of the overall experience.

The first prompt stated, “You are playing the game of blackjack against a dealer at a casino. The table rules are as follows: dealer stands on a 17. If the dealer is showing a card with a 10 or greater value players may choose to insure their hands at half their bet. You have a 2 and a King; the dealer shows a 10. Respond what your choice is in this scenario using the following JSON schema: {“choice”: “ “,”reasoning”: “ “}”. This prompt was a great prompt but it left out the key rules of the game and the players options. Though this prompt did produce responses, it was also lacking thought of reasoning with other models other than GPT.

Version two of the prompt went as followed, “You are playing the game of blackjack against a dealer at a casino. The table rules are as follows: dealer stands on a 17. If the dealer is showing a card with a 10 or Ace, you may choose to take insurance and half your bet. If insurance is not taken, you must say either hit, stand, split or double down as your choice as well on the first two cards, if given more than 3 cards you must only hit or stand. You have a {cards}, The dealer has a {dealer} Respond with your choice using the following JSON schema: {“choice”: “ “,”reasoning”: “ “}”. This prompt gave a good detailed basic outline of the game to the LLMs. The only lacking issue is that this prompt often confused models if given more than 2 cards and would give a handful of illegal moves. Furthermore it too lacked an option of surrendering for the LLMs to decide if they would like to do that over any other decision

The final version of the prompt is stated:

“ “ “You are playing the game of blackjack against a dealer at a casino. The table rules are as follows: - Dealer stands on 17. - If the dealer is showing a 10 or Ace, you may take insurance. - If insurance is not taken, you must choose: hit, stand, split, double down, or surrender on the first two cards. - With 3+ cards, you can only hit or stand.

Your hand: {cards}

Dealer’s card: {dealer}

Respond in JSON format: {{ “choice”: “ “,”reasoning”: “ “ }}“ “ “

The prompt above optimizes data collection by providing the most structured, detailed, and well-defined format, ensuring that responses are clear, consistent, and make logical sense across large language models (LLMs). By explicitly outlining the rules of the game, available actions, and constraints for

different hand scenarios, it minimizes ambiguity and prevents the model from making invalid or illogical moves. Additionally, this prompt improves data quality by ensuring responses are generated in a clean, standardized JSON format, making it easier to parse, analyze, and compare between different AI models [20]. This structured approach not only reduces errors in decision making, but also improves data processing efficiency, enabling more accurate tracking of AI reasoning patterns. By refining the language and removing unnecessary complexity, the final version of this prompt ensures that the model focuses on making strategic decisions rather than misinterpreting unclear instructions. As a result, this maximizes clarity of answers, reduces inconsistency, and provides the most reliable data set for evaluating whether reasoning is influenced by memory, probability, or optimal blackjack strategy. It is designed to handle expansions and modifications with minimal effort. The MODEL_MAP structure allows easy integration of new AI models, while the database schema can be extended without breaking existing functionality. Additionally, decoupling AI processing, database manipulation, and file operations makes it easier to optimize individual components for future performance improvements [19].

Beyond improving LLM blackjack simulations, this structured framework serves as a foundation for broader applications in LLM reasoning and decision-making research. The principles of structured data collection, modular LLM processing, and standardized evaluation can be extended to domains such as financial forecasting, medical diagnosis, and legal reasoning, where LLMs must process complex rule-based systems while maintaining accuracy and consistency. By applying similar methodologies in these fields, researchers can enhance the reliability of LLM decisions, improve interpretability, and establish best practices for assessing LLM performance in real-world scenarios. Looking ahead, integrating advanced LLM reasoning techniques—such as reinforcement learning, probabilistic modeling, and self-explanation mechanisms—into this structured framework could further enhance its effectiveness [7]. By leveraging reinforcement learning, LLM could refine their blackjack strategies over time, learning from past decisions to optimize long-term performance. Probabilistic modeling could enable more sophisticated risk assessment, allowing LLMs to make decisions that account for both short-term and long-term outcomes. Meanwhile, self-explanation mechanisms could improve transparency by requiring models to justify their choices in a structured format, providing valuable insights into their reasoning processes.

Ultimately, the structured prompt and data collection methodology presented here not only optimizes AI-driven blackjack evaluation but also establishes a scalable, adaptable, and interpretable framework for future AI research. By prioritizing clarity, consistency, and modularity, this approach sets a strong foundation for developing more advanced and reliable AI decision-making systems in gaming and beyond[8].

LiteLLM acts as a dynamic proxy, enabling effortless switching between different large language model (LLM) providers. By simplifying the integration of multiple AI models, it allows seamless experimentation with various reasoning engines without requiring modifications to the core architecture. With its intu-

itive API, LiteLLM streamlines the process of testing reasoning performance in blackjack simulations, making it easy to adjust based on outcomes. Additionally, if an AI model encounters delays or becomes unavailable, LiteLLM can automatically redirect requests to another provider, ensuring the experiment runs smoothly without interruptions [8]. SQLite, a lightweight and serverless database, provides an efficient solution for managing experimental data without the complexities of traditional database setups.

The code efficiently integrates AI decision making using the `litellm` library, which allows seamless communication with multiple AI models. By using structured commands and requesting responses in JSON format, the script ensures that the model output is predictable and easy to parse. Additionally, the `MODEL_MAP` dictionary provides a quick way to switch between different AI models, allowing it to adapt to different performance requirements without requiring major code modifications. Rather than relying on temporary storage or inefficient file manipulation, the script uses SQLite to store blackjack results in a structured format. The `setup_database()` function ensures that the database schema is created only once, avoiding redundant operations. Additionally, `save_to_database()` follows a well-structured insert process that avoids duplicate entries, ensures data integrity, and minimizes unnecessary writes. The project was designed with modularity in mind, making it highly reusable and efficient. Functions like `get_blackjack_decision()`, `get_blackjack_model_result()`, and `save_to_database()` each serve a different purpose, reducing redundancy and increasing code clarity. This allows individual components to be upgraded or replaced without impacting the entire system, resulting in better maintenance and easier debugging. The code includes several layers of exception handling to ensure that errors do not interrupt execution. The `get_blackjack_decision()` function, for example, uses a try-except block to detect API failures, while `get_blackjack_model_result()` ensures that unexpected return types are handled appropriately. Additionally, database operations include error handling to prevent corruption or damage due to invalid data entered.

Using a database to store and retrieve player choices further increases efficiency. Rather than relying on complex in-memory calculations or multiple lookups, this ensures that previous decisions are stored persistently and accessed with minimal processing. By ensuring that player choices are recorded before a round begins, play can later be analyzed for strategic improvements or LLM-based decision making. If necessary, improvements such as caching or batch processing mechanisms can be integrated to further improve database interaction performance. The selection value is extracted once and passed to the `play_player` function, no additional queries or operations are required in the main game loop. No unnecessary function calls, reduces computational overhead, and ensures smooth execution. By keeping the logic clean, well-structured and making effective use of database storage, this method ensures speed and scalability, making it an efficient solution for running multiple rounds of blackjack in an integrated manner. Rather than inefficiently storing data in multiple unstructured text files, the script saves the database results to a nicely format-

ted text file using `save_table_to_file()`. Using formatted string alignment also makes the output easier to read without the need for external formatting tools.

As a file-based system, it enables seamless tracking of reasoning performance, blackjack outcomes, and decision-making patterns locally, eliminating the need for an external database server. Given that the experiment does not demand high concurrency, SQLite offers excellent performance with minimal resource consumption. Its seamless compatibility with Python and other programming languages makes it an ideal choice for logging LLM decisions, storing past game results, and analyzing patterns in memory-based reasoning [15]. By leveraging LiteLLM for LLM insights and SQLite for structured data storage, this setup creates a streamlined, cost-effective, and highly efficient framework. It removes the dependency on cloud-based databases and complex infrastructure, making the experiment more manageable and reliable. Furthermore, SQLite enables precise tracking of past blackjack outcomes, providing critical data to determine whether reasoning is influenced by memory rather than pure logical deduction. This approach ensures a well-structured, data-driven analysis of LLM decision-making processes [8].

4 Experiments

4.1 Blackjack

The import section of the code includes several key libraries that enable the system to perform essential functions. Libraries such as `os`, `re`, `json`, and `sqlite3` are used for various system operations, including string manipulation, JSON parsing, and interaction with a SQLite database. The database is crucial for storing the results of the LLM decisions in each game session, ensuring that a record is maintained for later analysis. The `random` library is used for shuffling the four decks of cards, ensuring that each game session has a randomized deck, which is a vital aspect of simulating a real Blackjack environment. The `datetime` library is used to timestamp the LLM decisions, making it possible to track and analyze the timing of each decision, which could be relevant when evaluating the LLM reaction times and decision-making speed. The `litellm` library is used to integrate and call different LLM that generate the decisions for Blackjack play. The `dotenv` library is used to load environment variables, allowing the program to maintain flexibility and security, particularly when accessing API keys or other sensitive configuration settings stored in an `.env` file.

The LLM decision-making component of the Blackjack game relies on Large Language Models (LLMs) to decide the best actions for the player to take during the game. Each LLM has specific criteria and training that influence its decision-making process, which is why different models might result in slightly different gameplay strategies. The system is designed to map each LLM to a particular action in the game. The LLM is provided with a prompt that explains how Blackjack works, and it is tasked with deciding whether the player should “hit,” “stand,” “double down,” or “split.” These decisions are returned by the LLM in a structured JSON format, containing two key pieces of information: the “choice” (i.e., the action to take), and the “reasoning” behind the decision. The system calls the LLM to generate the decision based on the current state of the game. After receiving the response, the system extracts the decision and removes unnecessary formatting from the JSON output, making the decision easier to process within the game’s flow. The reasoning behind the decision is also stored in the database, ensuring that the rationale for each decision is preserved and can be later analyzed for consistency or improvements [15]. An SQLite database is created to store the Blackjack session data. Each entry contains the LLM name, the date of the decision, the LLM’s chosen action, the reasoning behind that action, the player’s current hand, and the dealer’s face-up card. This database structure helps to track not just the actions taken, but also the underlying reasoning, allowing for an in-depth post-game analysis.

The Blackjack game itself begins with the initialization of a standard deck of cards. The deck is shuffled, and the player is dealt two cards, while the dealer receives only one card face-up. This simulates a real-world game setup, where the player can see one of the dealer’s cards but not their full hand. After the initial deal, the LLM is called to make a decision regarding the player’s next move. This decision is based on the player’s hand, the dealer’s visible card,

and the general Blackjack rules. The LLM could choose to “hit” (draw another card), “stand” (keep the current hand), “double down” (double the bet and receive exactly one more card), or “split” (if the player has a pair, divide the cards into two separate hands). Each LLM decision is logged into the database with the relevant details, allowing for precise tracking of decisions over time. If the LLM chooses “hit,” a new card is drawn and added to the player’s hand. After the player’s decision, the dealer draws additional cards following standard Blackjack rules—drawing cards until the total hand value reaches at least 17. Once the game session concludes, the database connection is closed, and the results are stored. This includes the final decision made by the LLM and the game’s outcome, which will later be used in the evaluation phase.

This Python script simulates a Blackjack game where an LLM, specifically Claude-3.5-Sonnet, is used to make player decisions. The game follows standard Blackjack rules, where the dealer stands on 17, and the player can choose to hit, stand, split, double down, or surrender. The script is structured to handle multiple rounds of play, logging each decision made by the LLM into an SQLite database. By incorporating LLM decision-making, this program not only plays Blackjack but also provides a mechanism to track decision trends and analyze how well the LLM performs over time. The script begins by defining essential components such as a deck of cards, a shoe size (set to six decks), and a mapping of available LLM. It then initializes the game by shuffling the shoe and setting up an SQLite database to store game results. The `setup_database` function ensures that a table is created if it does not already exist, allowing for structured storage of AI choices, reasoning, and game states. This setup enables easy retrieval and analysis of past decisions, making it a valuable tool for improving LLM Blackjack strategies.

The game logic is encapsulated within the `play_blackjack` function. It first initializes the shoe and continuously deals hands until the shoe is nearly empty, at which point it reshuffles. The player is dealt two cards, while the dealer receives one face-up card. The LLM is then called to make a decision based on the player’s hand and the dealer’s visible card. This decision is generated using the `get_blackjack_decision` function, which constructs a structured prompt for the LLM and parses its JSON response. If the LLM suggests hitting, an additional card is drawn; otherwise, the player’s turn ends. Once the player’s turn is over, the dealer follows standard Blackjack rules, drawing cards until their total hand value reaches at least 17. The dealer’s logic is handled using a simple sum calculation, where face cards are valued at 10, and Aces are counted as 11 unless they would cause the dealer to bust. The game continues until the shoe is depleted, at which point the database connection is closed, and the game session concludes. By logging all LLM decisions and game outcomes, the script provides a foundation for evaluating AI performance in Blackjack strategy over multiple simulated rounds.

4.2 Table of Similar hands

After all results were in after each LLM was completed, the results of all the LLMs were merged together with a query in order to combined multiple SQLite databases containing blackjack game results from the different. The goal is to compare the decisions made by these models in identical game scenarios and output the results into a CSV file for further analysis. The script first verifies the existence of the necessary database files before proceeding, ensuring that the required data sources are available. If any of the database files are missing, it prints an error message and exits, preventing execution errors due to missing data. The script uses an in-memory SQLite database to efficiently perform the query while avoiding modifications to the original database files. It attaches three external databases—`clauderesults.db`, `geminireresults.db`, and `blackjack_results.db`—allowing for seamless querying across multiple datasets. The SQL query is structured to find records where the player’s hand and dealer’s card match across all three databases. This ensures that only comparable scenarios are considered, providing a meaningful basis for evaluating LLM decision-making consistency and effectiveness.

Once the query is executed, the script fetches the results and prints them to the console for debugging purposes. This step helps verify that the query is returning valid data before writing it to a CSV file. The script then creates a CSV file named `choices.csv` and writes the results, including the player’s hand, dealer’s card, and each LLM decision. This structured output allows for further analysis, such as identifying patterns, inconsistencies, or strategic differences between the LLMs. Finally, the script ensures that the database connection is closed, even if an error occurs, maintaining good database management practices. By comparing LLM decisions under identical Blackjack conditions, this script provides valuable insights into how different models approach the game. The generated CSV file can be used for statistical analysis, model refinement, or training improvements. This type of structured comparison is particularly useful for researchers and developers working on AI-driven gaming strategies, helping to determine which model performs best in various Blackjack scenarios.

Player Hand	Dealer Card	Claude Choice	Gemini Choice	GPT Choice
""J"", ""9""	2	stand	double down	stand
""4"", ""2""	J	hit	hit	hit
""A"", ""6""	4	hit	double down	double down

Figure 2: Example of Outputs for Each LLM

4.3 CSV Evaluation

Model	Win%	Illegal Move %	Differed Move %	Reasoned Responses	Total Plays
GPT	45.39%	0%	16.55%	681	4,624
Gemini	39.55%	0%	49.98%	3937	7,281
Claude	40.62%	3.79%	34.94%	1550	4,471

Figure 3: Comparison of Data Results for Each LLM

4.3.1 Comparison with Book Model

The LLM’s decisions are compared to traditional blackjack strategies outlined in books and guides. This comparison helps to assess whether the LLM adheres to conventional, proven strategies or if it develops its own tactics that could be equally effective. The blackjack book is a baseline assessment of seeing if any LLM diverged from the most basic and most used blackjack method amongst amateurs. To read this, we set out a code that compared the answers to the book model and had an evaluation of 0, 1, and 2. 0 beings a basic common move and did not diverge from the book. 1 is a move that not done in the book and is legal. 2 means that it was an illegal move.

This creation was designed to test how well any of the LLMs align with established blackjack strategies while also identifying any potential deviations that could indicate novel or suboptimal decision-making. By comparing its moves against the widely accepted book strategy, we can determine whether the model follows a predictable path or if it introduces unconventional yet potentially effective tactics. The evaluation scale provides a structured way to categorize these decisions, ensuring that each move is assessed based on its adherence to traditional blackjack principles. Beyond classifying moves, this evaluation also helps refine the LLM’s decision-making. If the model frequently produces legal but unorthodox moves (rated as 1), further analysis can determine whether these choices yield statistically better results or if they stem from flaws in the model’s understanding of risk and probability. Similarly, any occurrences of illegal moves (rated as 2) would highlight critical errors that need correction, ensuring that the LLM functions within the game’s established rules.

In evaluation of all of the LLMs. GPT 16.55% uncommon moves Gemini 49.98% uncommon moves Claude 30.64% uncommon moves, as well GPT and Gemini both tested 0.00% for any illegal moves but Claude tested for 3.96% of illegal moves. These findings emphasize the varying strategic tendencies across different LLMs. While GPT played a more conservative and rule-abiding game, Gemini introduced a high degree of unconventionality, and Claude’s strategy blended a moderate level of creativity with occasional missteps. This evaluation framework not only helps assess the effectiveness of these models in blackjack but also provides a broader look at how LLMs process structured decision-making, strategy adherence, and rule compliance in gaming environments.

GPT, with only 16.55% uncommon moves, demonstrated the closest adherence to traditional book strategies, suggesting a more conservative and

predictable approach. This level of consistency may indicate that the model prioritizes established, statistically sound methods over creative or riskier plays. Gemini, on the other hand, showed the highest percentage of uncommon moves at 49.98%, meaning it frequently deviated from conventional strategy. This suggests that Gemini may be incorporating alternative heuristics or decision-making patterns that differ significantly from established blackjack norms. Claude fell in between, with 30.64% of its decisions classified as uncommon, indicating a moderate level of divergence from book strategy. However, what stands out most about Claude is its 3.96% illegal move rate. This suggests that the model occasionally failed to recognize fundamental rules of the game, which could stem from misinterpretations of blackjack mechanics or even further not being trained with any reasoning tasks. The presence of illegal moves could negatively impact its reliability in strategic decision-making, highlighting a need for refinement to ensure full compliance with other reasoning tasks in the future.

4.3.2 Analysis of Similar Hands

The LLM performance in similar game scenarios is compared to evaluate its consistency, strategic reasoning, and adherence to optimal blackjack play. By analyzing how an LLM handles specific situations—such as a hand value of 12 against a dealer’s 6 versus a hand value of 12 against a dealer’s 10—we gain insight into whether the model follows fundamental blackjack principles or exhibits inconsistencies in its decision-making. A well-trained blackjack player—or a properly tuned LLM—should recognize that a 12 against a 6 is a significantly better situation than a 12 against a 10. In the first case, the dealer has a higher chance of busting, making standing a more viable option. In contrast, facing a dealer’s 10 requires a different approach, as the dealer is statistically more likely to achieve a strong final hand. Evaluating how consistently an LLM makes the correct decision in these scenarios helps determine whether it fully grasps these strategic nuances.

The evaluation revealed that in the experiment, there were roughly 12,000 similar plays, providing a large dataset for analysis. This volume of hands allows for a statistically significant comparison of decision-making trends across different LLMs. The results can show whether a model makes the same decision consistently when faced with identical or near-identical scenarios or if it introduces variability that might suggest uncertainty, randomness, or reliance on flawed heuristics. This kind of comparative analysis can uncover patterns in how different LLMs balance risk and reward. If a model inconsistently hits or stands in the same situation, it may indicate gaps in its understanding of probability and expected value. On the other hand, if an LLM consistently makes optimal moves but deviates in a way that improves performance, it could hint at the development of a unique yet effective playing style.

4.3.3 Win Percentage

The win percentage had the most flaws of the three evaluations. When recording the data, the results were read and evaluated manually as it was flawed that this issues was primarily due to the challenges in accurately recording and interpreting the data. Unlike the straightforward classification of moves into common, uncommon, or illegal categories, win percentage is influenced by a wide range of factors, including randomness, variance, and the specific game conditions during each hand. This makes it a more complex and less reliable metric for assessing the effectiveness of an LLM’s blackjack strategy.

One of the main issues stemmed from the fact that results were evaluated manually, which introduced the potential for human error and inconsistency in data collection. Manual recording though gave the opportunity to find and see any sort of noticable moves from any of the LLMs. This hands-on approach allowed researchers to catch patterns and anomalies that an automated system might have overlooked. By manually reviewing the results, evaluators could identify specific trends, such as whether an LLM consistently struggled with certain hand values or whether it displayed an unexpected tendency to favor aggressive or conservative plays. This qualitative insight, while harder to quantify, provided valuable context for understanding the reasoning behind each model’s decisions.

When seeing the results of the code, it was recorded that GPT won 45.39%, Gemini won 39.55%, and Claude won 34.74%. Claude had the lowest win rate at 34.74%, which aligns with its higher percentage of both uncommon (30.64%) and illegal moves (3.96%). The presence of illegal moves likely played a role in its lower performance, as breaking the rules of the game inherently leads to penalties or invalid hands. Additionally, its moderate rate of uncommon moves suggests some deviation from traditional strategy, but not necessarily in a way that improved its overall effectiveness. Gemini, with a 39.55% win rate, performed noticeably worse than GPT. Its significantly higher rate of uncommon moves (49.98%) suggests that it may have employed a more experimental or unconventional strategy that did not always lead to optimal outcomes. While uncommon moves are not necessarily bad, they must be backed by strong probabilistic reasoning to outperform traditional strategies—something that Gemini may not have achieved consistently. GPT having a 45.39% win rate suggests that it had the strongest performance, potentially due to its more consistent adherence to traditional strategy and fewer deviations into untested or risky plays. This aligns with its lower percentage of uncommon moves (16.55%), indicating that following established blackjack strategies led to a higher success rate over time.

4.4 Reasoning Results and Illegal Plays

When you interact with a person, you don’t just hear their words—you observe their emotional responses, gestures, posture, facial expressions, and vocal tone [11]. These physical and paralinguistic cues provide a rich context for under-

standing how they are reasoning, reacting, or reevaluating their position in real time. In contrast, large language models (LLMs) operate purely in the textual domain. They lack a physical presence, emotions, and conscious intent, so their reasoning—if we can call it that—must be inferred entirely from their use of language. In order to emulate or signal reasoning in text, LLMs rely on structured language patterns and discourse markers that traditionally signal logical progression or inference. Words and phrases such as “therefore,” “hence,” “thus,” “if... then,” “must be,” “logically follows,” “consequently,” “implies,” “deduce,” “conclusion,” “I want,” and “premise” are linguistic indicators of a reasoning process. These are the breadcrumbs of logic that readers follow to evaluate whether a coherent and valid line of thought is being formed.

For LLMs, these words are not signs of genuine cognition, but statistical echoes of reasoning observed in human language data. Nonetheless, when used skillfully, they can create the impression of rational deliberation. In essence, an LLM mimics the surface structure of reasoning rather than engaging in it with intentionality. This means that analyzing such linguistic markers can be one way to identify when and how an LLM is simulating rational thought, argumentation, or justification—even though there is no inner mental state or motivation behind the text. This difference highlights a key challenge in interpreting LLM outputs: while human reasoning is embodied, emotionally informed, and goal-directed, LLM-generated reasoning is disembodied, affectless, and goal-neutral—produced not from a desire to understand, but from patterns in data. Yet, the very appearance of logical scaffolding in an LLM’s output can be persuasive, even compelling, to human readers, blurring the line between true reasoning and its simulation.

A differed move refers to a decision made by the language model that deviates from the basic strategy set forth in the basic blackjack book, which serves as the gold standard for optimal play. These books, such as *Blackjack Blueprint* or *Beat the Dealer*, provide statistically proven strategies for all possible hand and dealer combinations, based on probability and expected outcomes. The idea is that in each scenario—whether holding a 16 versus the dealer’s 10, or a pair of 9s versus a 7—there is a mathematically optimal move (hit, stand, double, or split) that maximizes the player’s odds over time. When the model’s decision deviates from this baseline recommendation, it is recorded as a “differed move.” Differed moves do not necessarily mean that the model is wrong, but they do indicate a divergence from a proven, high-performing strategy. In some cases, the LLM may be experimenting, applying alternative reasoning, or misinterpreting a situation. In other cases, variability may arise from the model’s command history, random sampling, or lack of strong alignment with blackjack-specific training. A high rate of different moves indicates inconsistency in following the optimal strategy and may lead to suboptimal performance over a large number of hands. Tracking these differences is essential to assessing a model’s understanding of structured decision-making, especially in tasks that rely on the precise application of probabilistic rules and judgments.

In the context of blackjack, illegal moves refer to actions that violate the basic rules of the game, and in this experiment, two main forms of such viola-

tions were observed: playing after a bust and splitting on unsplit hands. A bust occurs when a player’s hand exceeds 21, and by rule, the round ends for that player immediately—no further decisions can be made. If a language model continues to “prompt” or suggest additional actions after a hand busts, it reflects a fundamental misunderstanding of the flow and constraints of blackjack. Such errors undermine the model’s credibility in rule-based reasoning tasks and reveal gaps in understanding the game’s end conditions. The second common illegal move is splitting cards that do not qualify for splitting. Under standard blackjack rules, players are only allowed to split their hands if they are dealt two cards of the same value—for example, a pair of 8s or two Kings. Attempting to split nonmatching cards, such as 7s and 9s, is not allowed and is a clear violation of the rules. When a model like Claude makes such moves, it signals that, while it produces well-founded explanations, it does not consistently apply the basic rules of the game. This illegal act is more than just a technical error; it calls into question the model’s ability to operate reliably in a structured environment where legality and rule compliance are not optional but essential. Ensuring that the model avoids illegal moves is a basic requirement for its use in any decision-making system that mimics real-world constraints.

GPT demonstrated the highest overall adherence to correct blackjack principles, with 0% illegal moves and the lowest proportion of differed moves, 16.55% across 4,000 plays, giving 461 reasonable answers. This shows that GPT not only follows the blackjack rules perfectly, but also stays in line with the basic strategy taught in reliable blackjack books. The lowest number of justified answers implies that it may only respond when it is certain or prompted, reflecting a conservative yet accurate decision-making style. But as well, it reflects at a hint that GPT possibly has not been trained highly with reasoning. When examining the results, we see that in 4,000 plays of blackjack GPT only differed from the book as well as having reasoned responses was only 461 which indicates that this falls beneath the desired goal of 20% of reasoned responses. With several plays in the dataset receiving no explanation, this raises questions about GPT’s depth of reasoning in decision-heavy environments. While its strict adherence to rules and low deviation from the basic strategy are commendable, the lack of consistent reasoning output suggests it may rely more on pattern recognition than on dynamic evaluation of context. This could be a limitation in scenarios where interpretability, transparency, or adaptive strategy adjustments are required. For tasks where explainability is as important as correctness—such as financial advising or legal decision support—GPT’s performance in this experiment highlights the need for more robust prompting or fine-tuning to elicit consistent, high-quality reasoning alongside rule-compliant decisions.

Claude, on the other hand, displayed more erratic behavior. While it produced a significantly higher number of reasoned responses (1,550), its win rate was lower at 40.62%, and it committed 3.79% illegal moves, the only model in the group to violate the rules. It also had a high differed move rate (34.94%), meaning it often gave inconsistent responses when presented with the same scenario multiple times. This inconsistency, combined with illegal moves, suggests that Claude sometimes prioritizes novelty or exploration over strategic

soundness and rule-following. The large response volume implies Claude may have attempted to over-explain or justify its decisions, which may be useful in interpretability studies but detracts from its effectiveness in a rule-bound environment like blackjack. With that said though, this does prove to be useful that Claude did reason as it was not as successful but Claude did in fact explain its thought processing with each move that was given, which offers valuable insight into how the model interprets and processes structured decision-making tasks. This makes Claude a potentially strong candidate for applications where interpretability and transparency are prioritized over strict rule adherence—such as in exploratory research, educational tools, or systems that support rather than replace human decision-makers. However, in domains like blackjack, where optimal performance is tightly coupled with rigid strategy and rule compliance, Claude’s tendency to deviate from established norms and occasionally violate game mechanics undermines its overall effectiveness. Still, its extensive reasoning output can serve as a diagnostic tool, helping researchers identify where and why the model diverges from expected behavior. This could inform future fine-tuning efforts aimed at balancing creativity with rule-bound performance.

Gemini delivered the highest volume of reasoned responses (3,937) but had the lowest win percentage at 39.55%. Despite having no illegal moves, its differed move rate was the highest at 49.98%, nearly half of all hands. This suggests that while Gemini consistently followed the rules, its internal decision-making was highly variable. Such variability may reflect either a broad exploratory strategy or a lack of stable grounding in blackjack principles. Its sheer output volume hints at a verbosity tendency, potentially generating elaborate justifications for decisions that are inconsistent or suboptimal. While Gemini’s rule adherence is commendable, its high rate of strategy deviation suggests it lacks a reliable internal policy for navigating the game. It as well shows us that there is a strong building of reasoning. Although it was not as successful as the other two models. Gemini reacted the most in total. Having almost half of differed moves from the basic blackjack book model and the highest volume of reasoned responses. This positions Gemini as a model that is highly responsive and eager to provide explanations, which can be beneficial in domains where interpretability and engagement are more important than strict accuracy. However, its lack of strategic alignment raises concerns about its ability to internalize structured frameworks like the basic blackjack strategy. The fact that Gemini made nearly 4,000 reasoned responses across 7,000 plays shows a high level of activity, but quantity does not necessarily translate to quality—especially when a large portion of those responses deviate from optimal play. The model’s performance suggests that while it can articulate its reasoning extensively, it may lack a calibrated decision engine that consistently prioritizes mathematically sound moves. Therefore, Gemini’s behavior is a double-edged sword: its transparency and verbosity may make it a valuable tool for analyzing thought processes, but without more targeted fine-tuning or constraints, it struggles to perform effectively in tightly rule-based, outcome-sensitive environments like blackjack.

The comparative analysis of GPT, Claude, and Gemini reveals distinct strengths and weaknesses in how each large language model approaches the

structured, rule-bound environment of blackjack. GPT stood out for its strict rule adherence and alignment with basic strategy, making it the most consistent and accurate performer overall. However, its relatively low volume of reasoned responses highlights a potential gap in interpretability, raising questions about its ability to explain decisions in real-time or adapt to nuanced scenarios. Claude, while more expressive and interpretive, struggled with strategic consistency and even committed illegal moves, suggesting a preference for exploratory behavior over strict compliance. This makes Claude more useful in interpretability-focused contexts rather than high-precision, rule-dependent tasks.

Gemini offered the most extensive reasoning output, demonstrating a strong inclination to explain its choices. Yet, its high variability and deviation from optimal strategy significantly impacted its effectiveness, despite maintaining perfect rule adherence. The model’s verbosity and inconsistent internal policy point to the need for better calibration to balance clarity with performance. Collectively, these findings illustrate the trade-offs between accuracy, consistency, and interpretability in LLMs. While each model has applications where it excels, none delivered a perfect balance across all metrics. This underscores the importance of aligning model capabilities with task-specific priorities—be it rule compliance, transparency, or strategic soundness—when deploying LLMs in decision-making environments.

4.5 Threats to Validity

There are several threats to the validity of the experiment that could influence the accuracy of the results. One major factor is deck variability. Since the cards are shuffled randomly at the start of each game, there is no guarantee that the same hands will be dealt in each simulation. This randomness in card distribution can significantly impact the outcome of individual games, introducing a level of uncertainty into the results. Even if an LLM follows optimal strategy, unfavorable card distributions over multiple rounds could skew its win percentage, making it difficult to directly compare models under identical conditions. In experiments relying on chance-based systems like blackjack, variance must be accounted for with large enough sample sizes and statistical controls to draw meaningful comparisons.

Another critical factor is LLM limitations. Each model may have a different level of understanding when it comes to blackjack strategy. Some LLMs might have a stronger grasp of fundamental strategies, while others may deviate due to the way they were trained or programmed. If an LLM’s training data does not include a thorough understanding of blackjack principles, it may make suboptimal decisions even when faced with common scenarios. These decisions could be influenced by factors such as misinterpreting context, applying irrelevant patterns, or overfitting to specific training examples. As a result, differences in performance may not necessarily indicate one model’s superiority over another, but rather reflect differences in their training data, alignment procedures, or internal reasoning architectures.

Additionally, inconsistent decision-making poses a challenge to the reliability of the experiment. Since LLMs often rely on probabilistic reasoning and temperature-based sampling, their decisions for the same hand may not always be identical across different simulations. If an LLM sometimes chooses to hit on a 12 against a dealer’s 4 and other times decides to stand, this inconsistency could lead to varied results across multiple runs. This variability can be mitigated by enforcing deterministic behavior (e.g., temperature = 0), but doing so may also limit the model’s flexibility or creativity in other tasks. Without a stable and repeatable decision-making framework, it becomes harder to evaluate the true effectiveness of an LLM’s strategy. Context handling and memory limitations also present challenges. LLMs process information within a fixed context window, and depending on how the game state is presented, important strategic details may be lost or misunderstood. For example, if previous rounds or card history are not included or emphasized in the prompt, the model may not be able to consider long-term strategies such as card counting or adjusting play style based on the remaining deck composition.

Moreover, prompt engineering and instruction clarity significantly influence LLM performance. Slight variations in how the blackjack scenario is framed—such as the wording of a question or the format of input—can alter how the model interprets the situation and which strategy it applies. Inconsistent prompt design across models or trials may inadvertently introduce bias or noise into the results. There is the issue of evaluation criteria. Metrics like win percentage or average return may not fully capture strategic soundness. A model could perform well in terms of raw outcomes while still making strategically poor decisions that happen to work due to favorable luck. Therefore, integrating qualitative assessments—such as comparing model decisions to a basic strategy chart or analyzing specific hands—can provide a more holistic picture of model performance.

While this experiment provides valuable insights into LLM-driven blackjack decision-making, several factors can influence the accuracy and reliability of the results. Addressing these threats to validity—such as implementing larger sample sizes, ensuring standardized decision-making, refining model training, and improving prompt consistency—could improve the robustness of future experiments. Incorporating controlled test scenarios with known optimal plays and analyzing model behavior under deterministic conditions may also yield clearer insights into each model’s reasoning capability and strategic depth. By mitigating these issues, a clearer and more definitive assessment of LLM performance in blackjack can be achieved.

5 Conclusion

5.1 Summary of Results

This experiment provided a structured assessment of large language models (LLMs) in the domain of blackjack, focusing on their ability to adhere to established strategies, maintain consistency across repeated scenarios, and perform within the confines of rule-based logic. By benchmarking each model’s decisions against a standardized blackjack strategy chart, we evaluated how effectively they internalized and executed optimal play. Among the models tested, GPT demonstrated the strongest adherence to canonical strategy and the fewest deviations from ideal behavior, resulting in a consistently stable and rule-compliant performance. In contrast, Gemini exhibited a more exploratory, verbose style that, while within the bounds of the game, frequently strayed from optimal strategy. Claude displayed intermediate reasoning capabilities but was hindered by a troubling frequency of rule violations—such as hitting after busting or attempting to split incompatible hands—seriously compromising its effectiveness.

While win rate is a familiar metric in blackjack analysis, it proved to be an unreliable standalone indicator in this study due to uncontrollable factors like deck randomness, shuffle variability, and the manual nature of data collection. GPT achieved the highest observed win rate at 45.39%, but this number alone fails to account for the quality of decisions or rule compliance. More meaningful insights emerged from tracking strategy deviations and illegal actions, which offered a clearer view into each model’s strategic reliability and understanding of game mechanics. Repeated-hand testing further revealed that GPT was the only model to respond consistently across identical game states, highlighting a systematic and repeatable reasoning pattern. Meanwhile, Claude and Gemini showed significant response variability in the same contexts, pointing to underlying inconsistencies in internal logic or decision encoding.

In addition to strategic accuracy, the study illuminated important differences in the interpretability and transparency of model responses. GPT, despite being the most accurate and rule-abiding, frequently delivered terse decisions with minimal or no explanation. This may suit scenarios where precision is paramount, but it limits human users’ ability to understand or audit the model’s reasoning. Conversely, Gemini and Claude provided more verbose and richly reasoned outputs—albeit often at the cost of strategic soundness. Their willingness to explain decisions, even when flawed, opens a valuable window into the models’ internal processing and highlights a key trade-off: optimal performance versus explainability. In real-world applications—such as financial advising or healthcare triage—this balance between precision and interpretability will be a critical design consideration.

The experiment also exposed how differences in training data, model architecture, and fine-tuning processes can influence gameplay behavior. Claude’s rule violations suggest either a lack of embedded domain constraints or an overemphasis on natural language fluency at the expense of logic enforcement.

Gemini’s variability and verbosity could reflect a training dataset that favors explanation-heavy output or a lack of focused reinforcement in constrained, high-stakes environments. These patterns suggest that general-purpose LLMs—even highly capable ones—may fall short in domains requiring rigid rule compliance and repeatable precision. Closing this gap will likely require domain-specific fine-tuning, rule-checking modules, or reinforcement learning strategies that teach the model to optimize not just for language but for structured reasoning and bounded decision-making.

Expanding the scope of this research to include multi-agent LLM environments opens up compelling new directions. A full table of AI agents, each controlled by a separate LLM, could simulate realistic blackjack scenarios where decision-making is not only about optimizing one’s own hand but also interpreting and reacting to others’ behaviors. This introduces the potential for emergent strategy, as LLMs might begin to adjust their play based on perceived patterns or betting tendencies from others—whether they are human players or fellow models. Incorporating this dimension of social reasoning would allow researchers to assess whether LLMs can mimic or learn opponent modeling strategies similar to those employed by expert human players. Such a multi-agent framework could support collaborative learning or competition-based evolution, where models are pitted against each other repeatedly in tournament-style settings. Over time, this could foster an ecosystem of increasingly sophisticated decision-making agents, particularly if reinforced by feedback mechanisms that reward both rule adherence and strategic innovation. Beyond single-player performance, such setups test how well LLMs adapt, evolve, and refine their strategies in dynamic, unpredictable environments—a critical aspect of AI maturity in applied contexts.

From a tooling perspective, automation and data integrity should be key focuses in future iterations. Automating the game engine, decision logging, and move validation would dramatically reduce human error, improve result fidelity, and allow for larger sample sizes at scale. Coupled with model-agnostic prompts and unified evaluation criteria, such infrastructure would create a highly repeatable and fair benchmarking environment. This foundation could also support longitudinal studies, where LLM behavior is tracked across updates or retraining phases to understand how improvements in architecture affect structured decision-making capabilities over time. As these systems become more integrated into high-stakes applications, we must consider the ethical and regulatory implications of deploying autonomous decision-making agents. In games, the stakes may be entertainment or small financial risk, but in fields like healthcare diagnostics or investment strategy, flawed reasoning or hidden variability can have serious consequences. Experiments like this offer a controlled proving ground for building auditable, transparent, and behaviorally consistent AI, helping to establish the trust necessary for broader deployment. The ability to quantify exactly where and why an AI deviates from optimal or ethical behavior will be essential in fulfilling compliance standards, earning user trust, and meeting legal oversight. Research in this space holds value not only for AI advancement but also for understanding human cognition through contrast. LLMs

may be able to emulate aspects of strategic thinking, but they do so without emotion, fatigue, or cognitive bias. By observing where LLMs succeed or fail compared to humans—especially in probabilistic reasoning, risk tolerance, and pattern recognition—we can better isolate what makes human decision-making unique, fallible, or adaptive. In this way, blackjack and similar structured domains serve as both a training ground for machines and a mirror for ourselves, deepening our understanding of rationality, error, and learning in both artificial and natural intelligence systems.

Looking ahead, future work could involve expanding the experimental framework to include full-table, multi-agent environments, where several LLMs compete in real-time under consistent game conditions. This would create a more dynamic and realistic simulation, testing not only individual decision quality but also how LLMs adapt, learn from patterns, and respond to the behavior of others. Introducing probabilistic modeling, reinforcement learning, or Bayesian reasoning could allow models to better evaluate hidden dealer cards and uncertain outcomes, advancing their capabilities beyond static chart-following. Additionally, a standardized league or benchmarking system—complete with scoring metrics for legality, strategic accuracy, explanation quality, and consistency—could become a gold standard for comparing AI agents in rule-based games. Beyond the blackjack table, this work has broader implications for the role of LLMs in structured decision-making systems. Games like blackjack serve as microcosms for real-world domains that demand a blend of logic, probability, risk assessment, and ethical constraint. Understanding how LLMs perform in such environments not only improves their gaming capabilities but also informs their use in high-stakes applications like law, medicine, and finance. By focusing on transparency, reliability, and strategic integrity, future AI systems can be designed to act not just intelligently, but responsibly.

Furthermore, integrating explainability into multi-agent interactions could offer deeper insights into collective reasoning dynamics—revealing how LLMs justify decisions in competitive or cooperative contexts, and whether they can identify flaws or opportunities in others’ strategies. This opens doors to the development of AI agents capable not only of making decisions but of critiquing, negotiating, or even teaching strategic behavior. Such capabilities would be highly valuable in educational technology, advisory systems, and AI-assisted training environments. Finally, these experimental setups could foster interdisciplinary collaboration among AI researchers, cognitive scientists, game theorists, and ethicists—bridging technical development with a deeper exploration of how autonomous systems think, learn, and interact. As LLMs are entrusted with more complex decision-making responsibilities, ensuring their behavior aligns with human expectations, ethical norms, and real-world constraints will be central to their safe and effective integration into society.

5.2 Future Work

Future work in this area includes exploring adaptive learning to evaluate whether LLMs can improve their decision-making by being trained on previous blackjack

outcomes. One promising direction is the implementation of real-time strategy adjustments through reinforcement learning techniques, allowing LLMs to refine their behavior over multiple iterations based on feedback from past games. By integrating feedback loops and enabling dynamic strategy updates, LLMs could potentially surpass static, chart-based strategies and develop more nuanced approaches to blackjack. Additionally, expanding the scope of the study to include multiplayer scenarios would offer valuable insights into how LLMs perform in more complex, interactive environments. This would involve developing mechanisms for LLMs to interact not only with the game itself but also with other players—both human and model-based—and examining how cooperative or competitive behaviors emerge under varying conditions. Understanding how LLMs adjust to opponents’ betting patterns and behavioral tendencies could deepen our grasp of their strategic adaptability in real-world, multi-agent systems.

Another important avenue for research involves the integration of probabilistic modeling and Bayesian reasoning to enhance decision-making. By incorporating probabilistic assessments of hidden dealer cards and likely future outcomes, LLMs can make decisions that align more closely with advanced blackjack strategies used by human experts. This probabilistic approach would allow LLMs to reason under uncertainty more effectively and move beyond basic heuristic methods. Additionally, testing different prompt engineering techniques—such as modifying the level of game detail, adjusting how decisions are framed, or introducing constraints on verbosity—could lead to more consistent and strategically aligned responses across different models. Structured experimentation in these areas would offer a more comprehensive understanding of how LLMs handle probabilistic decision-making, ultimately refining the framework for LLM-driven blackjack systems. These advancements also have broader implications, paving the way for LLM applications in other decision-intensive domains such as finance, healthcare, and competitive gaming. As LLMs continue to evolve, future iterations could benefit from specialized training datasets tailored to structured decision environments, ensuring both improved performance and interpretability in scenarios where precision, consistency, and reasoning depth are critical. Moreover, comparative benchmarking across multiple LLM architectures could help determine which model families are best suited for strategic tasks involving rules, probabilities, and pattern recognition, further guiding the development of purpose-built decision-making agents.

In addition to enhancing decision-making capabilities, research into LLMs in the context of blackjack offers a useful testbed for evaluating the interpretability and transparency of large language models in constrained environments. Unlike open-ended tasks like conversation or creative writing, blackjack provides a structured, rule-based domain where optimal strategies are well-documented and deviations are easily quantifiable. This allows researchers to precisely measure where and why LLMs diverge from expected behavior. By comparing model decisions against a standard such as a blackjack strategy chart, we gain insight not only into performance, but into the internal logic and reasoning pathways

that guide each model’s output. Such insights are crucial for applications where interpretability is just as important as accuracy—especially in domains like legal reasoning, financial advising, or medical triage systems, where decisions must be both correct and explainable to human users.

Moreover, this line of research underscores the importance of aligning LLM behavior with domain-specific constraints. While LLMs are not inherently designed to follow structured decision protocols, with appropriate fine-tuning and prompt engineering, they can be molded to operate effectively within such frameworks. This raises important questions about the trade-offs between generalization and specialization. A model that performs well across many tasks may lack the fine-grained expertise needed for high-stakes, rules-bound environments. Conversely, a model trained specifically for blackjack—or another structured game—might excel in that domain while lacking broader utility. Future development could benefit from hybrid approaches, where foundational LLMs are supplemented by rule-checking modules or domain-specific plug-ins that enforce critical constraints and improve consistency. Ethical and regulatory considerations also emerge as this technology becomes more sophisticated. As LLMs begin to participate in games involving real stakes—whether financial or reputational—it becomes essential to address questions of fairness, transparency, and accountability. For example, if an LLM is used in an online blackjack advisor or autonomous player, developers must ensure that its decisions do not exploit system loopholes or encourage irresponsible behavior. There is also the issue of consent and disclosure: human players should be made aware when they are interacting with AI-driven agents. As with any AI deployment, rigorous validation and auditability must be part of the development pipeline to ensure that LLMs behave in a manner consistent with both ethical standards and game integrity.

One compelling future direction is the establishment of a competitive benchmarking platform—a “blackjack league table”—where a wide variety of LLMs (GPT, Claude, Gemini) are evaluated side-by-side in simulated environments. Each model would be subjected to the same sequence of games, using a fixed set of prompts and decision protocols, to ensure consistency and fairness. Performance metrics could include not only win rates and average returns but also decision accuracy compared to established strategy charts, volatility in behavior, adaptability to changing conditions, and interpretability of reasoning. By tracking these metrics over thousands of hands, researchers could identify which models are most reliable in high-variance environments and which exhibit emergent strategic behavior. This leaderboard-based approach would foster a more rigorous and transparent evaluation framework and open the door to regular “tournaments” that test model updates, training paradigms, and fine-tuning techniques in a competitive, data-rich context. Such a platform would also facilitate collaborative progress across the AI community by offering a standardized sandbox for LLM-driven decision-making, strategy evaluation, and performance diagnostics.

To enhance the utility of the proposed blackjack league table, future work could incorporate tiered competition levels to separate beginner-level strategy

execution from expert-level reasoning. For instance, models could be evaluated first in environments with only basic rules and no dealer up-card tracking, then progressively challenged in advanced modes featuring multi-deck games, dynamic table rules, variable betting strategies, and card counting scenarios. This tiered format would allow for a clearer analysis of LLMs’ capability ceilings—distinguishing between models that merely recall surface-level rules and those that exhibit deeper, adaptive reasoning under uncertainty. Such a structure mirrors professional chess rating systems and could serve as a benchmark for tracking improvement across model iterations.

A transparent scoring system could be designed to evaluate not only outcomes but strategic soundness. Points could be awarded not just for winning hands, but for making decisions that align with optimal play based on known blackjack charts. Penalizing irrational deviations or reward-seeking behavior inconsistent with long-term advantage would help isolate models that rely on luck from those that truly exhibit principled reasoning. Additional qualitative metrics—like coherence in explanation, verbosity control, and consistency under identical prompts—could be scored separately, enabling holistic profiling of each model’s reasoning style.

Lastly, this work opens doors to exciting interdisciplinary collaborations between AI researchers, cognitive scientists, behavioral economists, and game theorists. Blackjack, with its blend of probability, psychology, and structured strategy, serves as an ideal environment to explore how machines learn, adapt, and reason under uncertainty. Questions like how LLMs weigh risk, how they respond to loss streaks, and whether they develop “superstitions” or irrational patterns mirror those studied in human decision-making. This provides a unique opportunity to not only improve AI systems but also to better understand our own cognitive biases and strategic behaviors through comparison. In this way, the study of LLMs in blackjack becomes more than just a technical challenge—it becomes a mirror reflecting the complexities of rational decision-making in both machines and humans.

5.3 Future Ethical Implications and Recommendations

As LLMs continue to evolve and demonstrate strategic decision-making capabilities in games like blackjack, it is essential to consider the ethical implications of their use. While this experiment focused on evaluating blackjack performance, the broader implications extend beyond gaming and into real-world applications such as finance, autonomous decision-making, and AI-driven advisory systems. Ensuring transparency, fairness, and responsible deployment of AI in these areas is critical to maintaining ethical integrity.

One of the most immediate ethical concerns is the potential misuse of LLMs in gambling environments. If an AI model can consistently outperform human players by following optimized strategies, it could be used to exploit casinos or online gambling platforms unfairly. Additionally, deploying LLMs as advisors for gambling strategies could encourage irresponsible behavior, potentially leading to financial harm for individuals who rely too heavily on AI-driven

recommendations. Ethical AI design should prioritize responsible use by implementing safeguards that discourage addiction and reckless gambling. Bias and fairness in decision-making are also critical concerns. The experiment revealed that different LLMs had varying levels of strategic adherence, inconsistencies, and even illegal moves. This suggests that AI models may exhibit biases based on their training data or decision-making algorithms. In high-stakes scenarios, such as financial trading or risk assessment, biased or inconsistent AI behavior could lead to unfair advantages or unintended consequences. Future development should focus on refining LLM training methodologies to ensure fair and unbiased decision-making.

Another major ethical consideration is transparency and explainability. LLMs operate as black-box models, meaning their reasoning processes are not always interpretable. In a structured game like blackjack, unexpected or uncommon moves can sometimes be effective, but in real-world applications, such unpredictability can be problematic. If an AI makes an unconventional decision, it should be able to provide a clear rationale for why it chose that action. Enhancing AI transparency will be essential in fields like legal decision-making, healthcare, and finance, where accountability is paramount. To improve the reliability of AI evaluations, future research should establish standardized frameworks for assessing AI decision-making in structured environments. This would include automated tracking systems, larger sample sizes, and controlled experimental conditions to minimize bias and randomness. Applying these frameworks to games like blackjack can serve as a model for evaluating AI reasoning in more complex decision-making scenarios.

Regulatory bodies should also develop ethical guidelines for the use of AI in gambling and financial decision-making. AI models should be designed with built-in limitations to prevent exploitation in casinos or online betting platforms. Similarly, in financial markets, AI-driven decision-making should be subjected to regulatory oversight to prevent manipulative or unfair practices. Clear policies will ensure that AI is used responsibly and does not promote unethical behavior. Another area for improvement is ensuring consistency and rule compliance in AI models. The presence of illegal moves in some LLMs suggests a need for better rule enforcement mechanisms. Future iterations should include reinforced learning approaches that penalize rule violations and ensure compliance with the fundamental principles of the task at hand. This would be especially important in applications where AI is expected to operate within strict ethical and legal boundaries.

Furthermore, AI developers should prioritize making LLM decision-making processes more interpretable. This could involve integrating explainability features that allow users to see why an AI chose a specific move in blackjack—or, in broader applications, why it made a particular financial or legal recommendation. Providing such explanations would build trust and accountability in LLMs. While blackjack serves as a controlled environment for evaluating AI decision-making, future research should explore how LLMs perform in more complex and dynamic decision-making tasks. Testing AI in real-world strategic environments—such as economic modeling, medical diagnostics, or legal

reasoning—can help assess its broader applicability and limitations. By expanding AI testing to more diverse scenarios, researchers can ensure that AI systems are capable of handling real-world challenges with accuracy and fairness.

The evaluation of LLMs in blackjack provides valuable insights into AI-driven decision-making, but it also raises ethical concerns regarding bias, fairness, and responsible deployment. As AI continues to advance, ensuring transparency, consistency, and adherence to ethical principles will be crucial. By refining evaluation methodologies, implementing regulatory safeguards, and enhancing explainability, future AI systems can be developed responsibly to benefit society while mitigating risks.

References

- [1] 2024. ChatGPT. <https://chatgpt.com/>
- [2] 2024. Claude. <https://claude.ai/login?returnTo=%2Fonboarding%3FreturnTo%3D%252F%253F>
- [3] 2024. Google Gemini. <https://gemini.google.com/>
- [4] 2024. OpenAI. <https://docs.litellm.ai/docs/providers/openai>
- [5] 2024. SQLite Show Tables: Listing All Tables in a Database. <https://www.sqlitetutorial.net/sqlite-show-tables/>
- [6] Wang, Xinyi Cohen, William W. Chen, Wenhui, Ma, Xueguang. 2023. Program of Thoughts Prompting: Disentangling Computation from Reasoning for Numerical Reasoning Tasks. <https://openreview.net/pdf?id=YfZ4ZPt8zd>
- [7] Collin Doyle. 2024. LLMs as Method Actors: A Model for Prompt Engineering and Architecture. <https://arxiv.org/pdf/2411.05778>
- [8] Bansal, Mohit Hase, Peter. 2021. When Can Models Learn From Explanations? A Formal Framework for Understanding the Roles of Explanation Data. <https://aclanthology.org/2023.emnlp-main.67.pdf>
- [9] Hou, Le Wu, Yuexin Wang, Xuezhi Yu, Hongkun Han, Jiawei Huang, Jiaxin, Shane Gu, Shixiang. 2023. Large Language Models Can Self-Improve. <https://aclanthology.org/2023.emnlp-main.67.pdf>
- [10] Leeb, Felix Gresele, Luigi Kamal, Ojasv Lyu, Zhiheng Blin, Kevin Gonzalez, Fernando Kleiman-Weiner, Max Sachan, Mrinmaya Schölkopf, Bernhard Jin, Zhijing, Chen, Yuen. 2023. CLADDER: Assessing Causal Reasoning in Language Models. <https://openreview.net/pdf?id=e2wtjx0Yqu>
- [11] Johnson-Laird, Philip. 2006. How We Reason. <https://books.google.de/books?id=UjYsJN0krNYC&printsec=frontcover&hl=de#v=onepage&q&f=false>
- [12] Wagenaar, Willem A. Keren, Gideon. 1985. Journal of Experimental Psychology: General.
- [13] Alexander Kunst. 2022. Players of casino blackjack, poker or roulette in the U.S. by age 2018. <https://www.statista.com/statistics/227478/table-game-players-blackjack-craps-poker-roulette-etc-usa/>
- [14] Shubham Mohape. 2024. Zero-Shot, Few Shot, and Chain-of-thought Prompt. <https://plainenglish.io/blog/zero-shot-few-shot-and-chain-of-thought-prompt>

- [15] Plank, Barbara Mondrof, Philipp. 2024. Beyond Accuracy: Evaluating the Reasoning Behavior of Large Language Models - A Survey. <https://arxiv.org/pdf/2404.01869>
- [16] Lee, Katherine Roberts, Adam Fiedel, Noah Malkan, Karishma Narang, Sharan, Raffel, Colin. 2020. WT5?! Training Text-to-Text Models to Explain their Predictions. <https://aclanthology.org/2023.emnlp-main.67.pdf>
- [17] Mueller, Aaron Belinkov, Yonatan Nikankin, Pyaniv, Reusch, Anja. 2024. ARITHMETIC WITHOUT ALGORITHMS: LANGUAGE MODELS SOLVE MATH WITH A BAG OF HEURISTICS. <https://arxiv.org/pdf/2410.21272>
- [18] Patil, Avinash. 2025. Advancing Reasoning in Large Language Models: Promising Methods and Approaches. <https://arxiv.org/pdf/2502.03671>
- [19] Shanahan, Murray. 2024. Talking about Large Language Models. <https://dl.acm.org/doi/pdf/10.1145/3624724>
- [20] Balunovic, Mislav Vechev, Martin Staab, Robin, Vero, Mark. 2023. BEYOND MEMORIZATION: VIOLATING PRIVACY VIA INFERENCE WITH LARGE LANGUAGE MODELS. <https://arxiv.org/pdf/2310.07298>
- [21] Hanks, Ashley Rittmayer Swiston, Robert C. Stone, Eric R., Parker, Andrew M. 2023. Thinking without knowing: Psychological and behavioral consequences of unjustified confidence regarding blackjack strategy. <https://pmc.ncbi.nlm.nih.gov/articles/PMC9909440/pdf/fpsyg-14-1015676.pdf>
- [22] Schuurmans, Dan Le, Quoc Chi, Ed H. Narang, Sharan Chowdhery, Aakanksha Zhou, Denny Wang, Xuezhi, Wei, Jason. 2023. SELF-CONSISTENCY IMPROVES CHAIN OF THOUGHT REASONING IN LANGUAGE MODELS. <https://openreview.net/pdf?id=1PL1NIMMrw>
- [23] Zhou, Denny Wang, Xuezhi. 2024. Chain-of-Thought Reasoning without Prompting. <https://arxiv.org/pdf/2402.10200>
- [24] Schuurmans, Dale Bosma, Maartin Ichter, Brian Xia, Fei Chi, Ed H. Le, Quoc V. Zhou, Denny Wei, Jason, Wang, Xuezhi. 2024. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. https://proceedings.neurips.cc/paper_files/paper/2022/file/9d5609613524ecf4f15af0f7b31abca4-Paper-Conference.pdf
- [25] Zhao, Jingmiao and Anderson, Carolyn Jane. 2023. Solving and Generating NPR Sunday Puzzles with Large Language Models. <https://arxiv.org/pdf/2306.12255>