Should I Stay or Should I Go?

Application of Game Theory in Machine Learning and Opera Decisions

New Mexico

Supercomputing Challenge

Final Report

April 1, 2026

Welch Homeschool

Team Members:

Helena Welch

Kalliope Luna Welch

Teacher:

Cindy Welch

Project Mentor:

Paul Welch

Contents

1 Executive Summary

This project utilizes a combination of game theoretical concepts and large language model (LLM) technology to create a set of decisions based on a scenario from the opera *Dido and Aeneas*. We use the Consensus Game [1], an algorithm that mimics a human conversation through repeated interaction of LLMs. This interaction updates the probability that two instances of an LLM will choose a given answer to a question until they reach an equilibrium answer. As such, we examine whether game theory can facilitate LLMs to mimic human decision-making through back-and-forth interaction. We demonstrate that the Consensus Game improves LLM decision-making capabilities through testing with the atomic games, a set of basic situations that are resolved using game theory. Our construction of the Consensus Game was validated through data comparison to Jacob et al.'s [1] work, showing that the algorithm improved accuracy of knowledge-based questions.

This work is widely applicable to a variety of uses, as decision-making is constantly present in the lives of consumers, diplomats, and authors, among others. Users are becoming increasingly over-reliant on LLMs, extending their roles beyond the limits defined by their statistical design to include logic-based writing tasks [2–5]. We demonstrate the Consensus Game's usefulness in solving multi-step problems to generate alternative endings to an opera, thus aiding the study of how decisions are made by humans versus LLMs in writing stories.

2 Introduction

2.1 Role of LLMs in Decision-Making

LLMs are now employed by around 66% of today's population for a variety of purposes [6]; this usage is not limited to gathering fact-based information, but also includes generating stories or pieces of writing in which decision-making plays a key role [2–5]. However, using LLMs to aid in decision-making can be harmful, as LLMs generate their answers statistically rather than rationally. This means that they often struggle with questions that are not merely factual but require a certain level of logic to make wise choices [7]. Thus, we need to better understand the capabilities of LLMs in making decisions that impact our everyday lives before we use them without constraints.

In this paper, we term questions that require logic ("What is the best action in this scenario?") *decision-based* and questions that only involve previously obtained knowledge ("What is the function of an eye?") *knowledge-based*. Mathematical problems that require both knowledge and logic are termed *problem-solving* questions.

2.2 Purpose of Project

Game theory offers a set of mathematical tools and ideas that allow us to study how to make optimal decisions in various situations that affect multiple individuals, or players. It is used today to make strategic decisions in such fields as economics, diplomacy, and politics [8].

We hypothesize that game theory can improve LLM answers to decision-based questions through optimizing strategic interactions between models. Since groups of individuals often make decisions through discussions and arguments, we choose to test the Consensus Game, a game that simulates such interactions. In this project, we examine whether LLMs and humans will reach similar decisions in generating multi-step storylines. We use LLMs and the Consensus Game to evaluate the best sequences of options for characters in Purcell and Tate's opera *Dido and Aeneas* [9].

2.3 The Consensus Game and Prior Works

The Consensus Game (developed by Jacob et al. [1]) is used to achieve agreement between the answers chosen by two different methods for the same question. These two methods comprise:

1. the Generator, which chooses from a set of answers to respond to a given prompt, and
2. the Discriminator, which evaluates the correctness of the Generator's answer.

The Consensus Game is then played based on the answers given by the Generator and Discriminator, using equations to update the probability that each player will choose a given answer. Note that the Consensus Game does not "fine-tune" the LLMs themselves, but rather updates the probability that a given LLM returns a given answer.

Several studies have compared the decision-making processes of LLMs to those of humans. Jacob et al. [1] first developed the Consensus Game using two models, Llama 7B and 13B, to answer knowledge-based and problem-solving questions. Other works include Rasal et al. [7], which introduces a methodology for improving LLM decision-making by prompting it with multiple-choice questions, and Xiao & Wang [10], which concluded that humans are better at finding the Nash Equilibrium, or the optimal decision for all involved, than are LLMs.

In this paper, we study whether the Consensus Game improves the accuracy of LLMs in answering knowledge-based and decision-based questions. In addition, we identify differences between LLMs' in answering questions, based on their parameter sizes.

3 Methods

3.1 Overview of Methodology

First, we pass a series of knowledge-based questions and game theoretical scenarios to two separate instances of an LLM, the Generator and Discriminator. The Generator is prompted with several possible answers to a question, while the Discriminator must choose whether the Generator's answer is correct. The Generator is asked each question two different ways: 1) provide

the *correct* answer and 2) provide an *incorrect* answer. Each question is asked 100 times, creating a probability distribution of answers for each instance of each LLM queried. These probability distributions, known as the policies, are then updated 5,000 times based on each other's previous policies, simulating a 'conversation' between the Generator and Discriminator. The accuracies of the initial and final policies are then calculated for each data set and question.

3.2 Data Collection and Preparation

Four LLMs with varying parameter spaces (see Table 1) were used to compare variability in answer choices between machines. Following Jacob et al. [1], zero-shot prompting was used, meaning no examples were given before the actual question was passed. We found that LLMs mostly gave clean answers with this format, although LLMs with smaller vocabulary sizes (Llama 3.1 8b and Mistral-small) sometimes produced words surrounding their answers rather than the single digit requested.

Prompts are given below:

Generator prompt: "[Question] Please choose an answer from the following messages: [Answer Choices] Only give the number of your answer, 1, 2, 3, or 4, with no other text surrounding it. Do not provide any other notes or commentary."

Discriminator prompt: "[Question] Is the following the correct answer to this question: [Answer Choice] Provide ONLY a single digit response. State 0 for true and 1 for false. No further comments are allowed."

We also attempted to use Chat-GPT 5o, a larger model, but were banned for asking it the same knowledge-based questions repeatedly, as if we were collecting data to train our own model. In the future, we hope to try collecting data from a similar large LLM that does not ban our usage.

3.3 Knowledge-Based and Decision-Based Queries

Following Jacob et al.[1], knowledge-based questions were taken from the MMLU dataset [15] within the astronomy and anatomy topics. Each topic's dataset was approximately 200 questions.

model	parameters	vocabulary size	context window
Mistral-small [11]	22 billion	32k	128k
Command-r [12]	35 billion	128k	128k
Llama 3.1 8b [13]	8 billion	64 k	128k
Mistral-nemo [14]	12 billion	128k	128k

Table 1: Characteristics of the large-language models used. All LLMs were set to a default temperature of 0.8.

The following is an example question from the astronomy topic:

Why is Saturn almost as big as Jupiter despite its smaller mass?

1. *Jupiter's greater mass compresses it more, thus increasing its density.*
2. *Saturn has a larger proportion of hydrogen and helium than Jupiter and is therefore less dense.*
3. *Saturn is further from the Sun thus cooler and therefore less compact.*
4. *Saturn's rings make the planet look bigger.*

We then compare knowledge-based questions to decision-based queries based on standard game theoretical problems known as the atomic games. An example is given below. The full set of prompts we generated can be found in Appendix 2.

The prompt for a given game theoretical situation was given in two different ways: 1) abstractly and 2) textually.

1. Abstract: *Player 1 and Player 2 have two actions, A and B. If both players choose A, they will each receive a utility of 50. If both choose B, they will each receive a utility of 25. If Player 1 chooses B while Player 2 chooses A, they will each receive a utility of 15, and vice versa. What should each Player do? What is the Nash Equilibrium?*

2. Textual: *Two companies, Phony and Softdrive, are going to release a new product. If both companies choose to have product Option A, the more expensive, high-tech option, they will each receive 50 dollars. If they both choose Option B, the cheaper, lower-tech option, they will receive each 25 dollars. However, if Phony chooses Option A and Softdrive chooses Option B, they will both only receive 15 dollars, and vice versa. What should each company do? What is the Nash Equilibrium?*

3.4 The Consensus Game

The following is a summary of the derivation of the Consensus Game by Jacob et al. [1, 16].

Having collected 100 answers from the Generator and Discriminator for each question, Jacob et al. [1, 16] calculate the probability P that a given answer y was chosen for a given question x and correctness parameter ν (correct or incorrect). This is used to define the initial policies $\pi_G^{(1)}$ and $\pi_D^{(1)}$:

$$\pi_G^{(1)}(y|x,\nu) = \frac{P_G(y|x,\nu)}{\sum_{y'}\sum_{\nu'} P_G(y'|x,\nu')} \tag{1}$$

$$\pi_D^{(1)}(\nu|x,y) = \frac{P_D(\nu|x,y)}{\sum_{\nu'}\sum_{y'} P_D(\nu'|x,y')} \tag{2}$$

We now provide the derivation by Jacob et al. [1, 16] for updating the Generator's initial policy $\pi_G^{(1)}$, keeping in mind that the derivation for $\pi_D^{(1)}$ is similar.

A standard utility u_G is defined as the amount of payoff the Generator gains for each correct answer from both the Generator and Discriminator.

$$u_G(\pi_G, \pi_D) = \frac{1}{2} \sum_{v\in\{\text{correct, incorrect}\}} \sum_{y\in Y} \pi_G(y|x,\nu) \cdot \pi_D^{(t)}(\nu|x,y) \tag{3}$$

This utility is regularized with the Kullback-Leibler Divergence D_{KL}, which prevents the policies from updating and diverging too rapidly from the previous policy. This new utility $\tilde{u}_G$ is given in Equation 5; the hyperparameter λ_G controls the penalty for the distance from $\pi_G^{(1)}$.

$$D_{KL}[\pi_G(\cdot|x,\nu)||\pi_G^{(1)}(\cdot|x,\nu)] = \pi_G \log\left(\frac{\pi_G}{\pi_G^{(1)}}\right) \tag{4}$$

$$\tilde{u}_G(\pi_G,\pi_D) = u_G - \lambda_G D_{KL}[\pi_G(\cdot|x,\nu)||\pi_G^{(1)}(\cdot|x,\nu)] \tag{5}$$

The regret $\text{Reg}_G^{(T)}$, or how much the Generator and Discriminator are penalized for a given answer choice, is defined below as the maximum difference of current and ideal utilities.

$$\text{Reg}_G^{(T)} = \max_{\pi^* \in \triangle Y}\left[\sum_{t=1}^{T} \tilde{u}_G^{(t)}(\pi_G^*|x,\nu) - \tilde{u}_G^{(t)}(\pi_G^{(t)}(\cdot|x,\nu)|x,\nu)\right] \tag{6}$$

The updated policy is proportional to the regret (Equation 7), and the Follow-the-Regularized-Leader algorithm is now used to guarantee a bound on the regret (Equation 8). The hyperparameter η_G controls how quickly the policy can change per iteration.

$$\pi_G^{t+1}(y|x,\nu) \propto \exp(\text{Reg}_G^{(T)}) \tag{7}$$

$$\begin{aligned}
\pi_G^{t+1} &= \arg\max_{\pi\in\triangle Y}\left[\left(\sum_{t'=1}^{t}\tilde{u}_G(\pi_G,\pi_D)\right) - \frac{1}{\eta_G}\sum_{y\in Y}\pi_G(y|x,\nu)\log\pi(y|x,\nu)\right] \qquad (8)\\
&= \arg\max_{\pi\in\triangle Y}\left[\eta_G\left(\sum_{t'=1}^{t}\sum_{y\in Y}\pi_G(y|x,\nu)u_G - \lambda_G\pi_G\log\left(\frac{\pi_G}{\pi_G^{(1)}}\right)\right) - \sum_{y\in Y}\pi_G(y|x,\nu)\log\pi(y|x,\nu)\right]\\
&= \arg\max_{\pi\in\triangle Y}\left[\eta_G\sum_{y\in Y}\left(t\lambda_G\log\pi_G^{(1)} + \sum_{t'=1}^{t}u_G\right)\pi_G(y|x,\nu) - (1+t\lambda_G\eta_G)\sum_{y\in Y}\pi_G(y|x,\nu)\log\pi_G(y|x,\nu)\right]\\
&= \arg\max_{\pi\in\triangle Y}\left[\frac{\eta_G}{1+t\lambda_G\eta_G}\sum_{y\in Y}\left(t\lambda_G\log\pi_G^{(1)} + \sum_{t'=1}^{t}u_G\right)\pi_G(y|x,\nu) - \sum_{y\in Y}\pi_G(y|x,\nu)\log\pi_G(y|x,\nu)\right]
\end{aligned}$$

Equation 8 is then solved using the Softmax Function w (Equation 9), which states that the updated policy π_G is now proportional to the exponential form of the regret.

$$w^{t+1}(\pi_G) = \frac{\eta_G}{1 + t\lambda_G \eta_G} \sum_{y \in Y} \left(t\lambda_G \log \pi_G^{(1)} + \sum_{t'=1}^{t} \tilde{u}_G \right) \tag{9}$$

$$\pi_G^{t+1} = \frac{\exp\{w^{t+1}\}}{\sum \exp\{w\}} \tag{10}$$

where:

$x =$ question

$y =$ answer choice

$\nu =$ correctness parameter

$t =$ update run

λ_G = hyperparameter that determines amount of regularization of the Generator policy

$\eta_G =$ hyperparameter that controls learning rate of the Generator policy updates

With this derivation [1, 16], the equations for updating the policies π_G and π_D are given in Equations 11 and 12. Figures 1 and 2 present an overview of the process represented in these equations.

The updated policy for each player (Generator or Discriminator) is based on the running average Q of policies of the other player up to this iteration t. Q is then regularized with the term $\lambda_G \log \pi_G^{(1)}$, which represents the Kullback-Leibler divergence. These are limited by the hyperparameters λ and η, which determine the amount of regularization and how quickly the policy will change from one update to the next, respectively. Thus, the policies of the Generator and Discriminator update 5,000 times based on each other's probability distribution of answers. This is the factor that simulates a human-like interaction between instances of an LLM, which we hypothesize will improve LLM decision-making capabilities.

$$\pi_G^{(t+1)}(y|x,\nu) = \frac{e(t, Q_G^{(t)}, \pi_G^{(1)})}{\sum_{y'} e(t, Q_G^{(t)}, \pi_G^{(1)})} \tag{11}$$

$$\pi_D^{(t+1)}(\nu|x,y) = \frac{e(t, Q_D^{(t)}, \pi_D^{(1)})}{\sum_{\nu'} e(t, Q_D^{(t)}, \pi_D^{(1)})} \tag{12}$$

where:

$$e(t, Q_G^{(t)}, \pi_G^{(1)}) = \exp\left[\frac{Q_G^{(t)}(y|x,\nu)+\lambda_G \log \pi_G^{(1)}(y|x,\nu)}{1/(t\eta_G)+\lambda_G}\right]$$

$$e(t, Q_D^{(t)}, \pi_D^{(1)}) = \exp\left[\frac{Q_D^{(t)}(\nu|x,y)+\lambda_D \log \pi_D^{(1)}(\nu|x,y)}{1/(t\eta_D)+\lambda_D}\right]$$

$$Q_G^{(t)}(y|x,\nu) := \frac{1}{2t}\sum_{\tau=1}^{t} \pi_D^{(\tau)}(\nu|x,y)$$

$$Q_D^{(t)}(\nu|x,y) := \frac{1}{2t}\sum_{\tau=1}^{t} \pi_G^{(\tau)}(y|x,\nu)$$

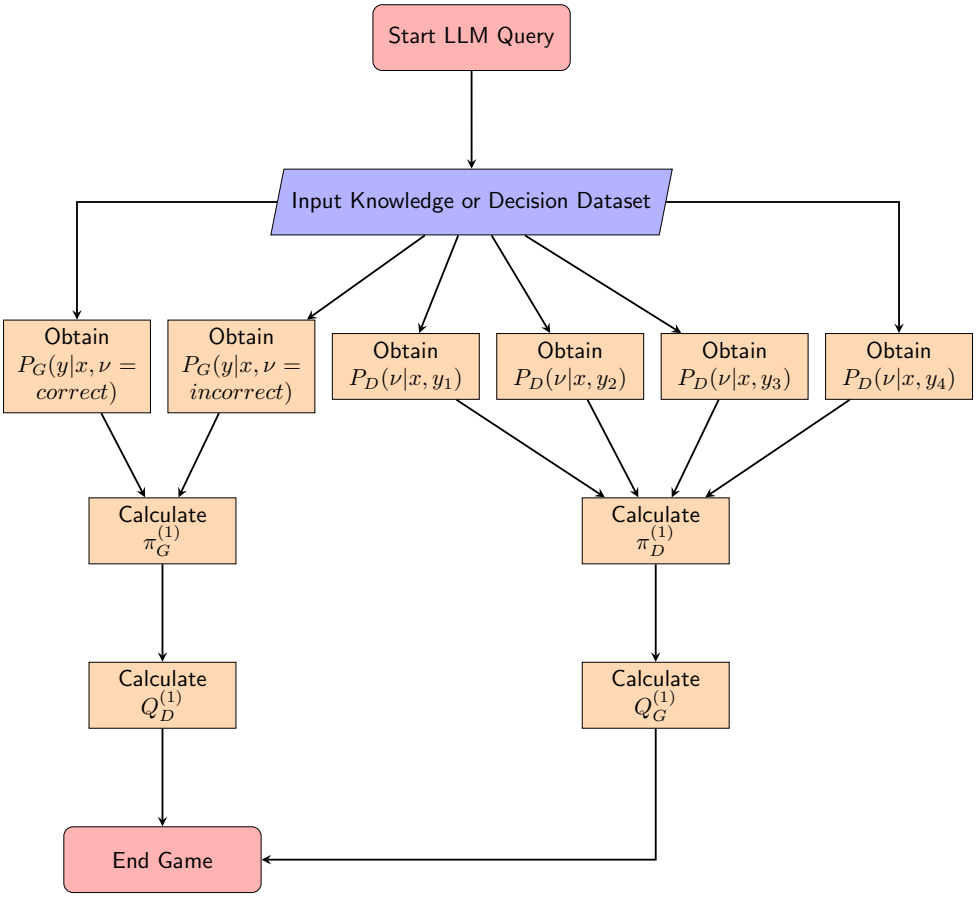

Figure 1: Flow chart of algorithm querying the LLMs and obtaining the probability distributions for the Consensus Game.

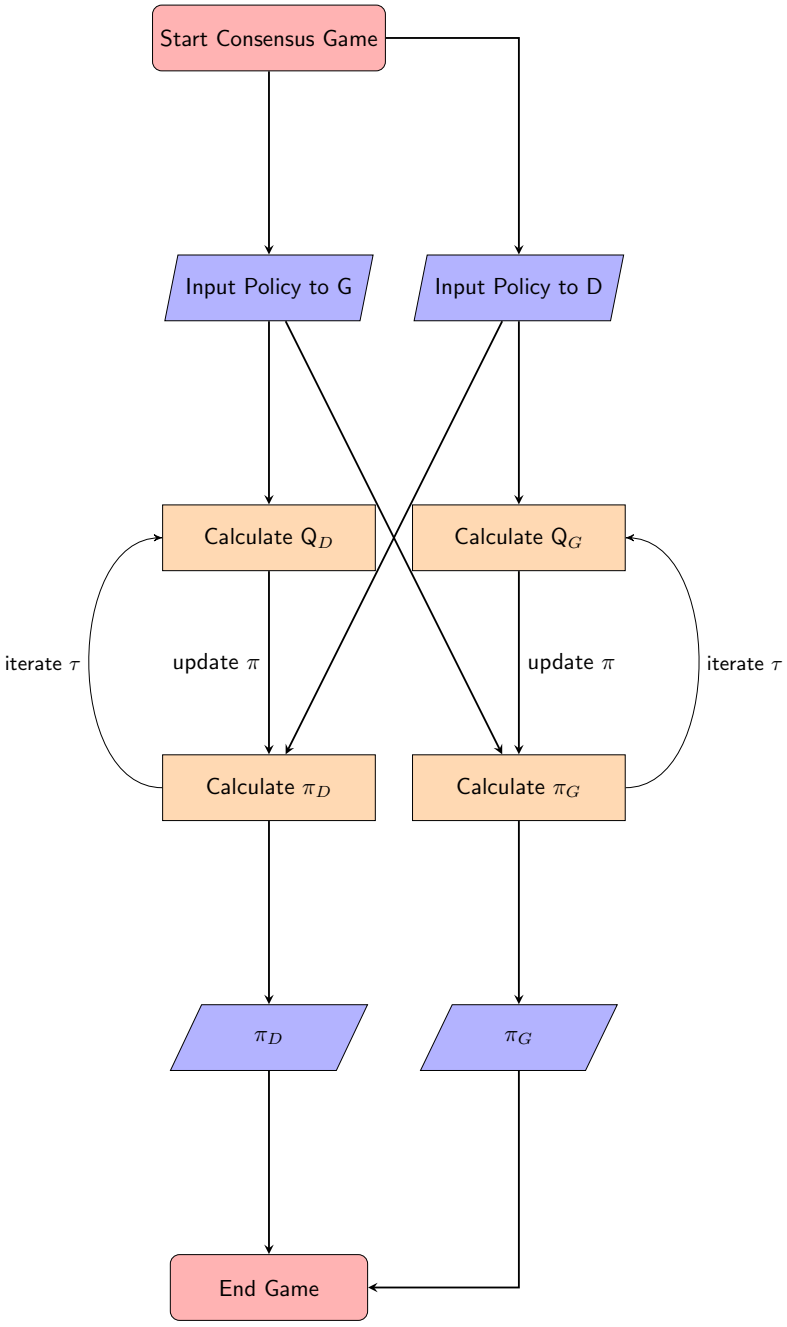

Figure 2: Flow chart of algorithm playing the Consensus Game.

3.5 Calculating Accuracies

Having performed the Consensus Game, the accuracy of each policy after the five-thousandth update ($\pi^{(5{,}000)}$) is calculated according to the following equations:

$$S_G(t) = \frac{\sum_{x'} \pi_G^{(t)}(y^*_{x'}|x', \nu_c)}{N_X} \tag{13}$$

$$S_D(t) = \frac{\sum_{x'} \pi_D^{(t)}(\nu^*|x', \nu_c)}{N_X} \tag{14}$$

where $y^*_{x'}$ and ν^* denote the correct answer choice and the correct correctness parameter for a given question x, and N_x represents the number of questions in the data set.

4 Summary of Computation

4.1 Hardware and Computational Details

Three computers were used to obtain LLM data and play the Consensus Game:

1. Intel Core i5-10400F×12; 48.0 GB Ram; NVIDIA GeForce GTX 1660 SUPER; Ubuntu
2. AMD Ryzen 9 3900X; 128 GB Ram; NVIDIA GeForce RTX 2080 SUPER; Fedora
3. AMD Ryzen 9 7950X3D; 128 GB Ram; NVIDIA GeForce RTX 4090; Ubuntu

We found that querying models used in this project on computer 2 often overburdened its GPU; thus, we use computer 3 for generating LLM output for all four models. Computer 1 was used to play the Consensus Game.

In running our code, we found that playing the Consensus Game for a given question takes approximately 1 minute. As such, an entire knowledge-based dataset took around 2.5 hours, and each decision-based question set took 2.5 minutes.

4.2 Testing of the Code

Several steps were taken to ensure that our code ran correctly and efficiently. First, policy updates were calculated by hand for several test probability distributions and checked against our code's computation. In addition, code was run on multiple computers, confirming that it worked on different hardware.

We perform unit tests on functions that compute the Consensus Game, following examples given by the *Pixi Basic Usage for Python* tutorial [17]. We use Pytest [18] in an environment provided by Pixi [17] to test whether our functions will return the intended values.

5 Results

5.1 Testing the Consensus Game

We now analyze results from the Consensus Game. We find that repeated updating of the policies makes the Generator and Discriminator very 'confident' of their answers. The initial probability of choosing a given answer is relatively low, but after running the policy updates 5,000 times, the probability of choosing one of the four answers increases logarithmically, often to nearly 100% probability after the first 2,000 runs (see Figures 3 and 4).

Note that a certain complexity in the evolution of the policy updates is introduced by asking for the LLM to select an incorrect answer as opposed to a correct option. Often an initial answer increases in probability before being replaced by a different choice (see Figure 5). This indicates the greater difficulty that comes with the greater sophistication of such a question.

Figure 6 compares the inital policy $\pi_G^{(1)}$ (top plot) and the final policy $\pi_G^{(5000)}$ (bottom plot) for an ordered set of astronomy questions. For each question x, one answer choice becomes most popular and the others are almost never chosen; thus, the answers of the LLM become polarized as a result of the Consensus Game. Consequently, through repeated interaction, both the Generator and Discriminator become extremely confident of their answers. The goal is that their choice correctly answers a given question.

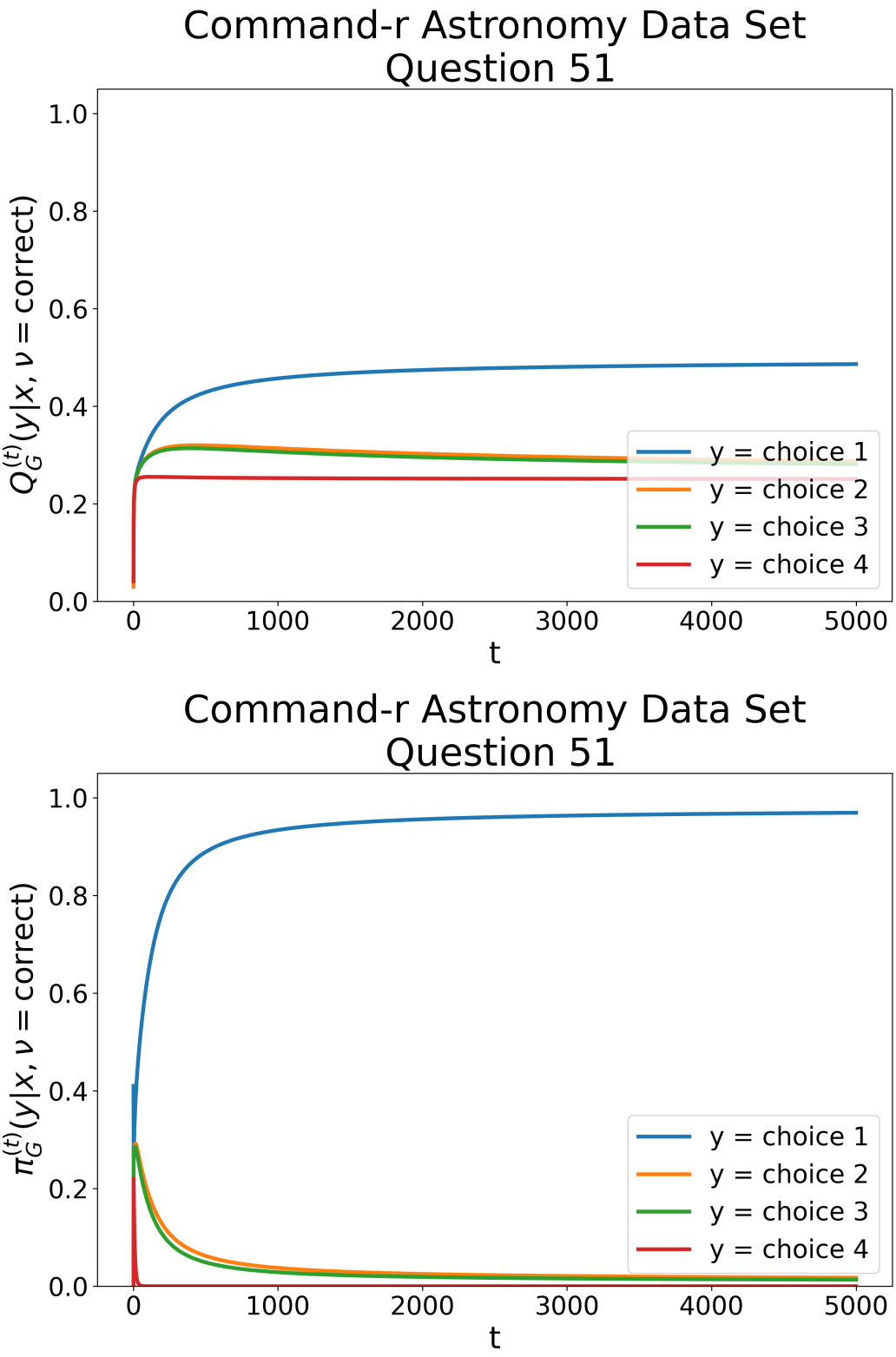

Figure 3: Evolution of the Generator's Q-value (top) and policy π_G (bottom) over time step t for correct answer choices.

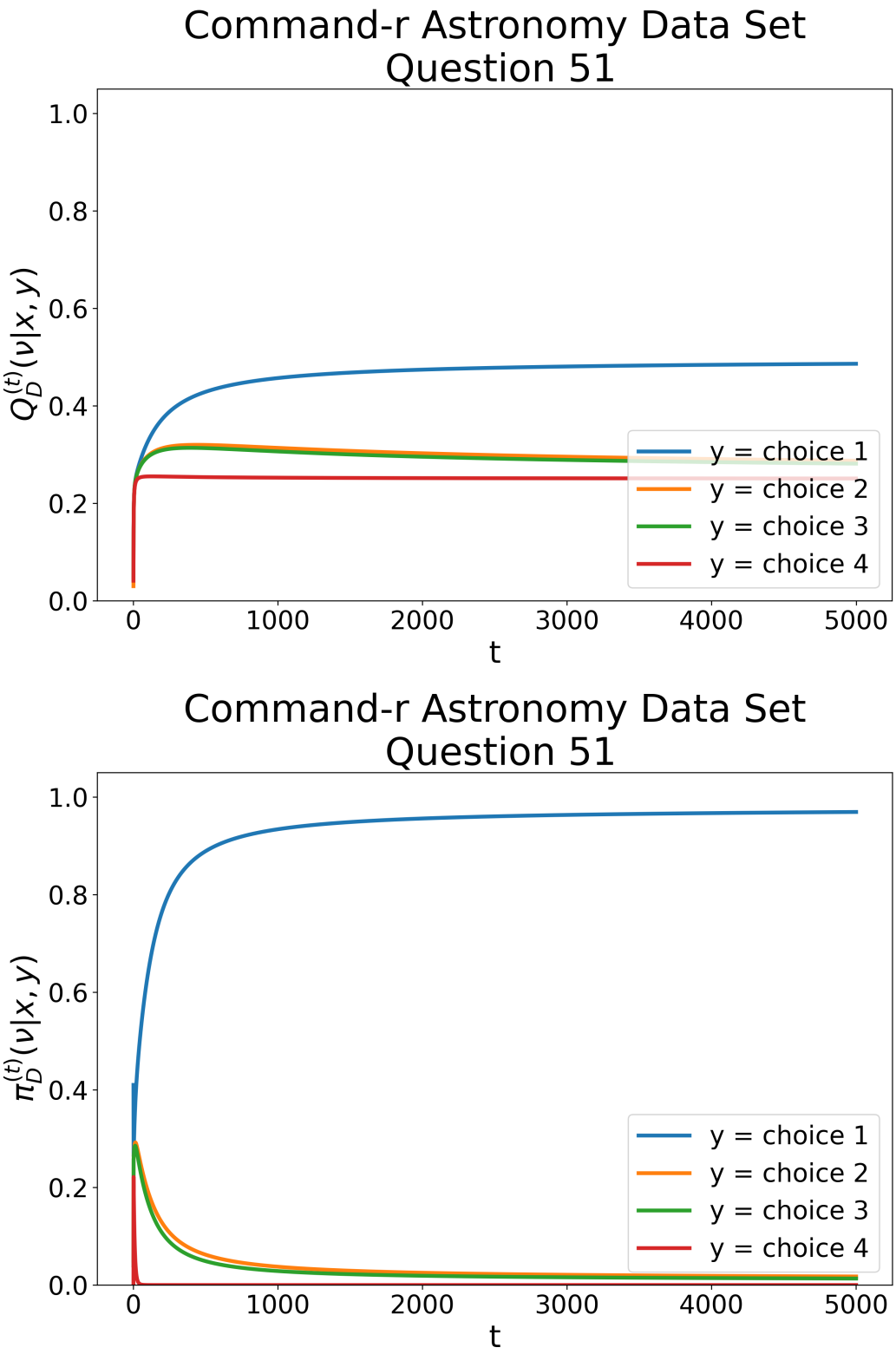

Figure 4: Evolution of the Discriminator's Q-value (top) and policy π_G (bottom) over time step t for correct answer choices.

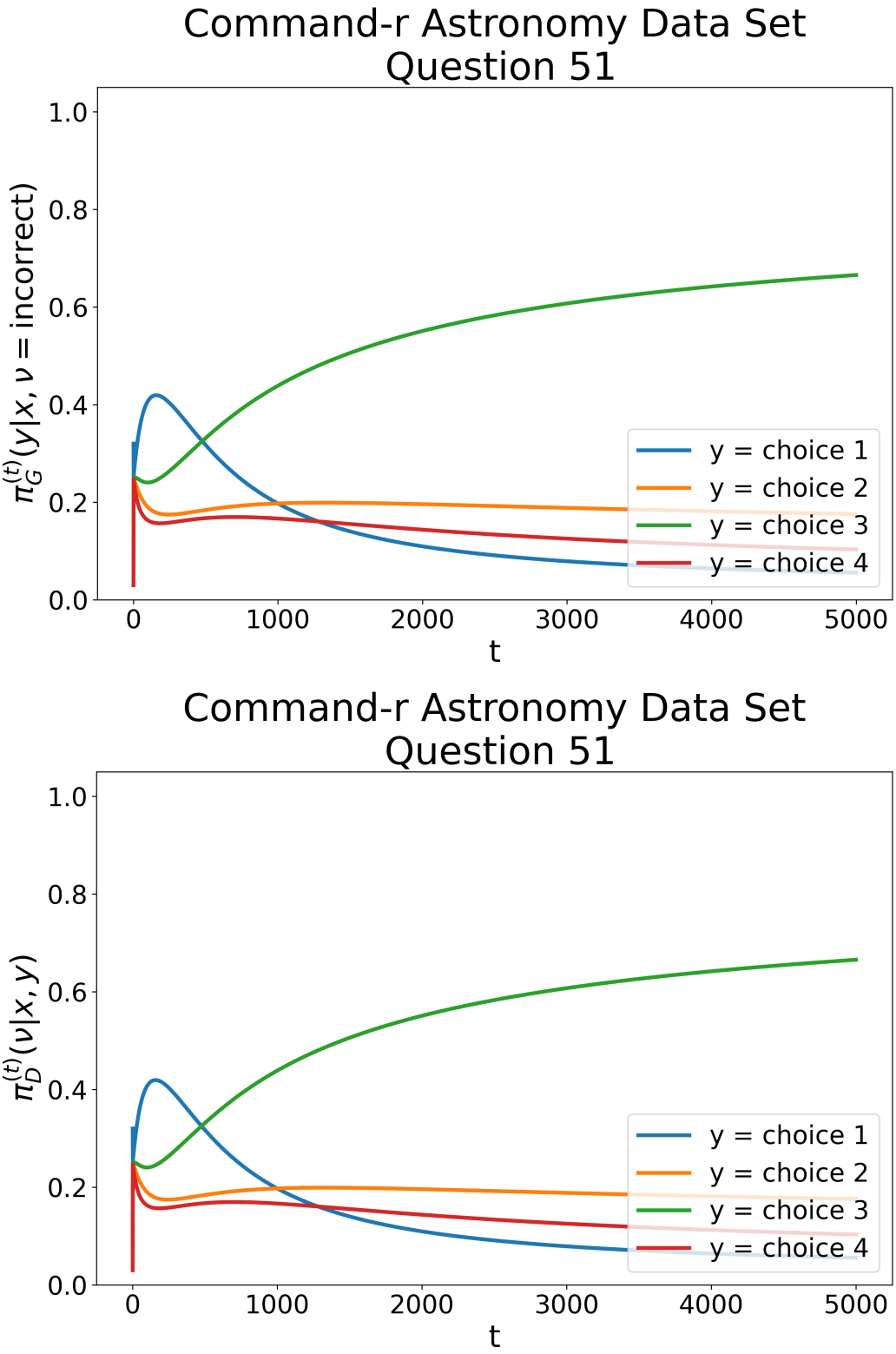

Figure 5: Evolution of policies π_G (top) and π_D (bottom) over time step t for incorrect answer choices.

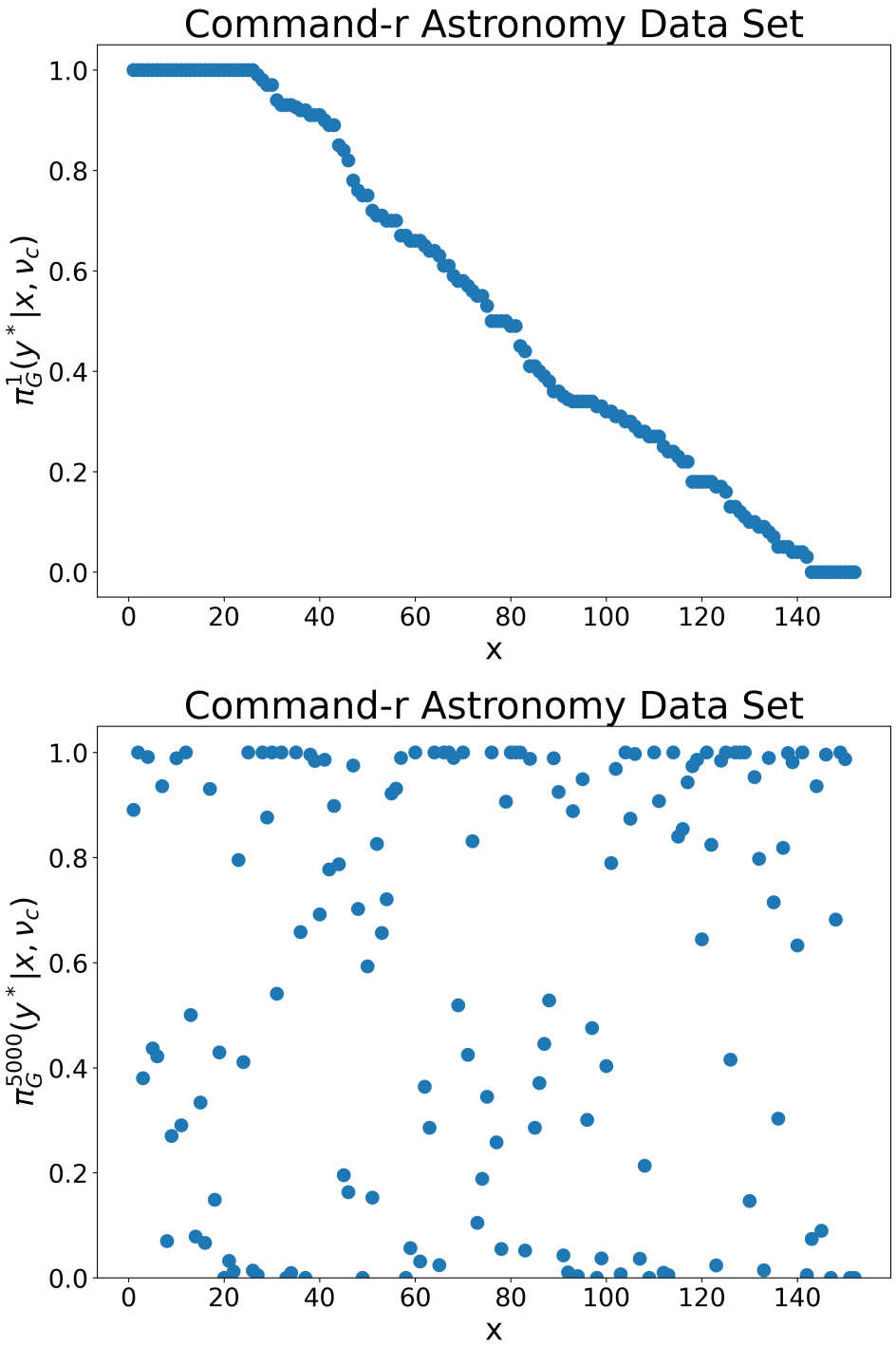

Figure 6: Accuracy of Generator in answering ordered set of astronomy questions before (top) and after (bottom) the Consensus Game is played. The higher the value of π_G, the greater the accuracy of Command-r in answering the corresponding question.

5.2 Optimizing Hyperparameters

To determine how much the policy should change after each update, we iterate through values of λ_G, λ_D, η_G, and η_D to find which produce the highest accuracy. Overall, we find that accuracy for decision-based questions changes very little as a function of λ, but that a value of 0.1 usually leads the Consensus Game to perform at slightly higher accuracies. Thus, using a smaller Kullback-Leibler Divergence and changing the probability distribution in larger steps at each update is usually most effective.

In addition, we find that the learning rate η has little effect on the final accuracy of each LLM on any given decision-based dataset; however, increasing η shortens the number of times the Consensus Game must be played before the updated policy plateaus and becomes constant. As such, increasing the learning rate η increases the efficiency of the algorithm, as the Consensus Game needs to be performed fewer times before coming to a final answer.

5.3 Accuracy of Consensus Game

Overall, we find that the Consensus Game improves accuracy of knowledge-based questions only marginally, if at all, as seen in Figures 7 and 8. Decision-based questions, on the other hand, resulted in large increases in accuracy, from below 20% to above 70% in the case of the Discriminators of the smaller models Mistral-nemo and Llama 3.1-8b (see Figure 9 and 10). Thus, our hypothesis that the Consensus Game could improve the decision-making capabilities of LLMs was confirmed. Abstract and textual formats for the decision-based prompts (see Section 3.3) overall tended to perform similarly, with textual prompts performing slightly better. Giving more context in prompts consequently allows the LLMs to perform at a higher accuracy. We also find that the Discriminator tends to improve more drastically than the Generator.

Most significantly, we find that the increase in accuracy most often occurred for questions involving the Nash Equilibrium, or the optimal strategy for all players together (see Figure 11 and Appendix 1).

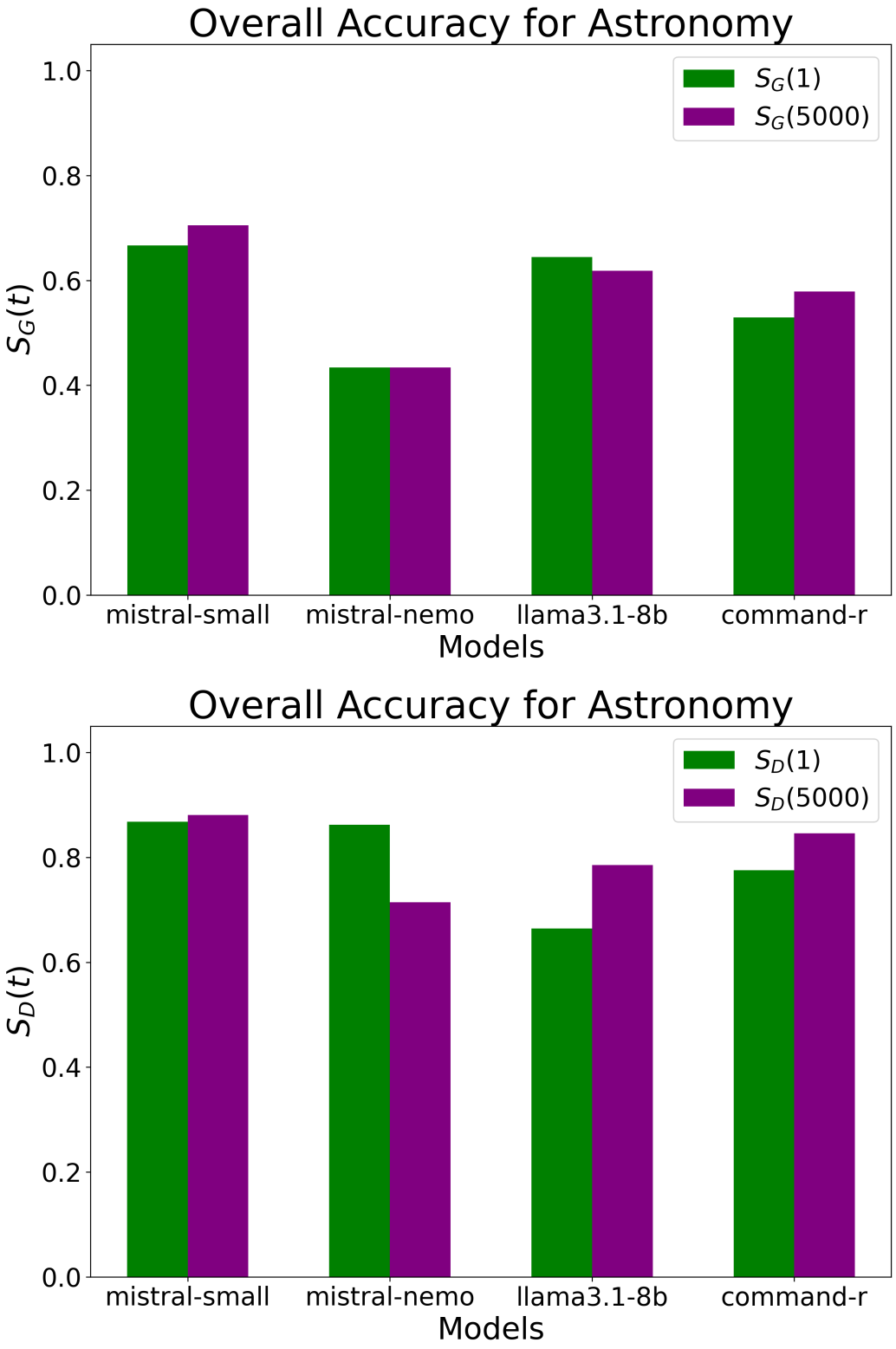

Figure 7: Overall accuracy of each LLM in answering astronomy questions from the MMLU database.

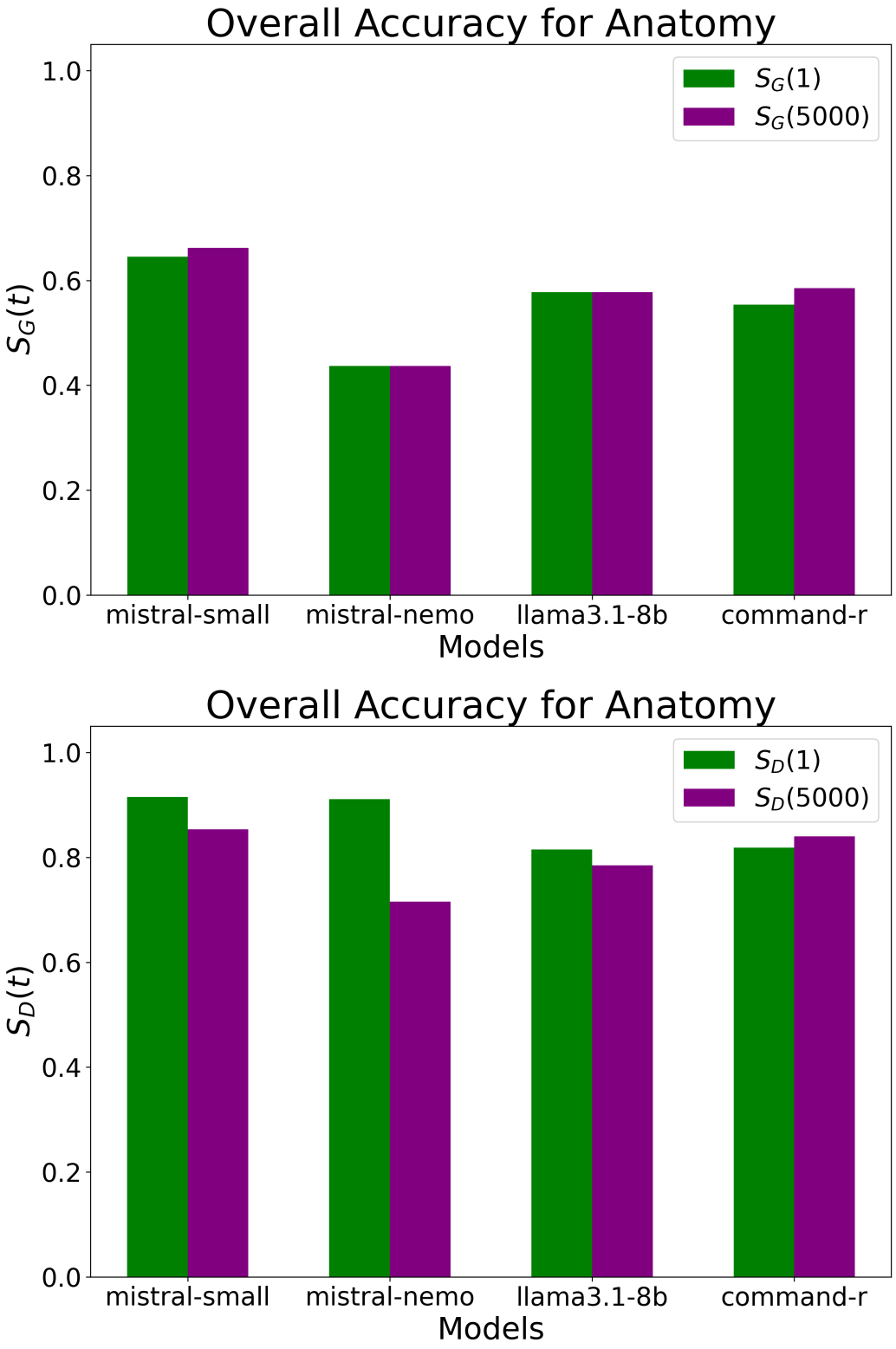

Figure 8: Overall accuracy of each LLM in answering anatomy questions from the MMLU database.

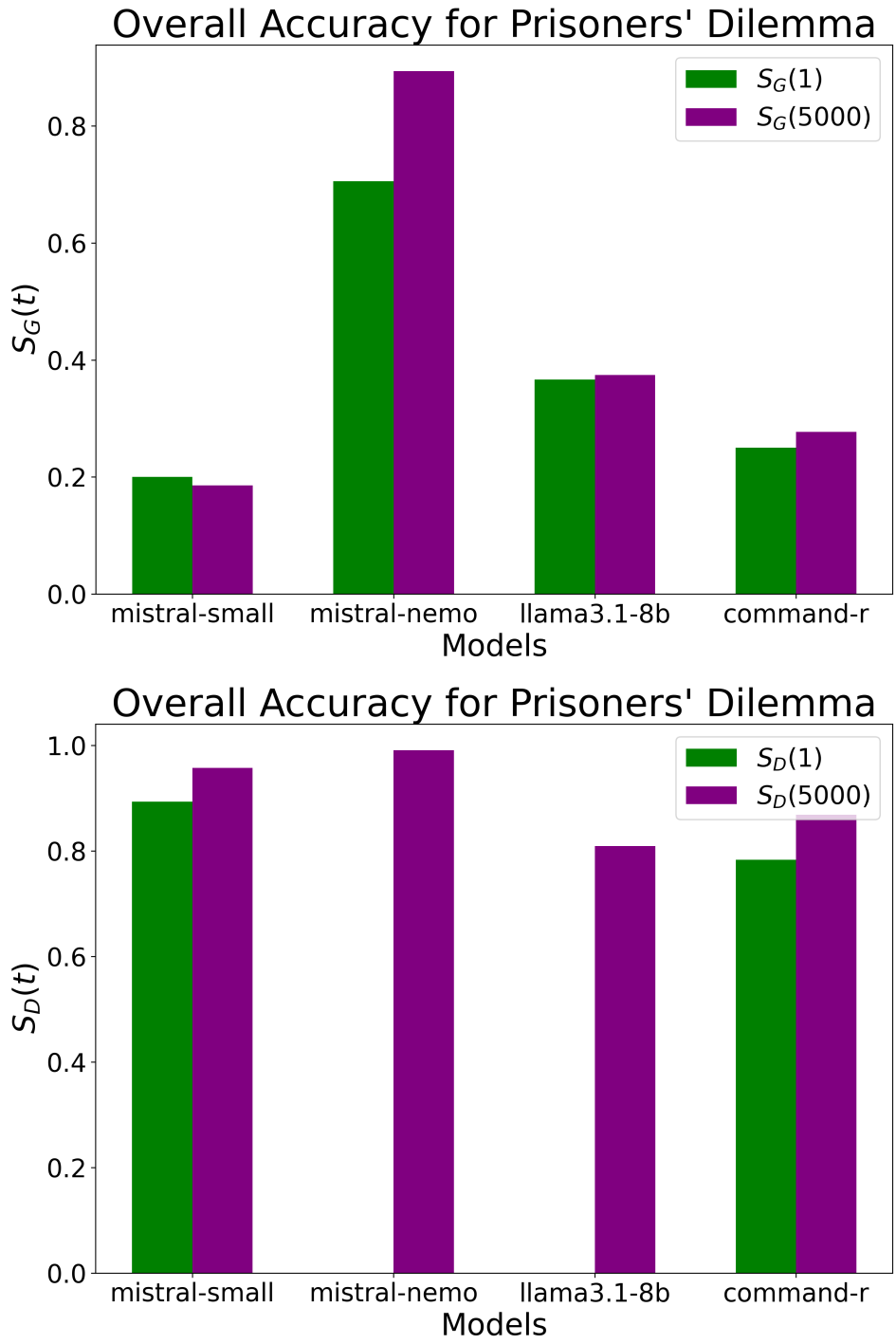

Figure 9: Overall accuracy of each LLM in answering questions related to the Prisoners's Dilemma (see Appendix 2) for the Generator (top) and Discriminator (bottom). Discriminator accuracy begins at 0% for Mistral-nemo and Llama 3.1-8b, but increases to 98% and 80%, respectively, after the Consensus Game is played.

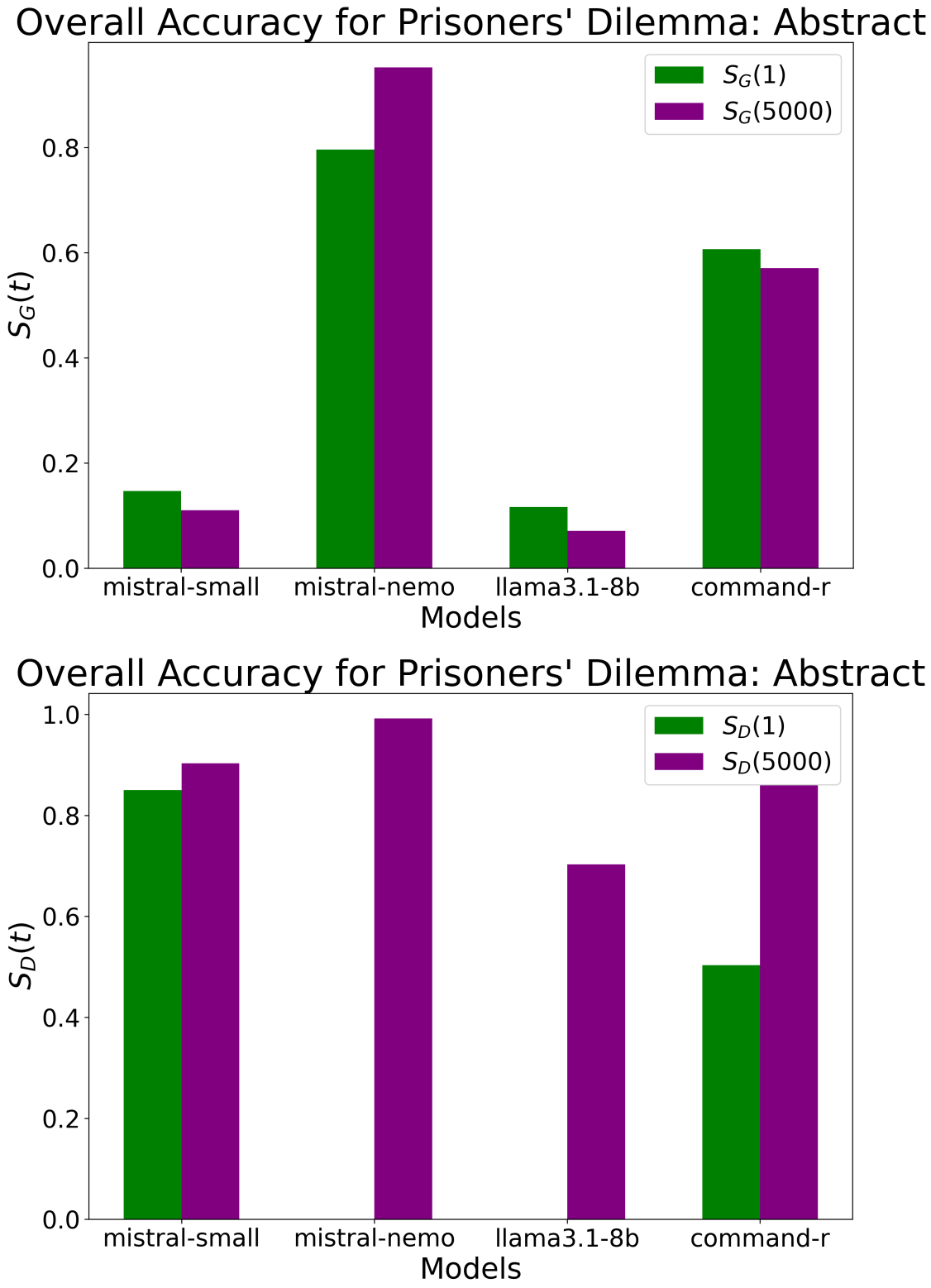

Figure 10: Overall accuracy of each LLM in answering questions related to the abstract version of the Prisoners's Dilemma (see Appendix 2) for the Generator (top) and Discriminator (bottom).

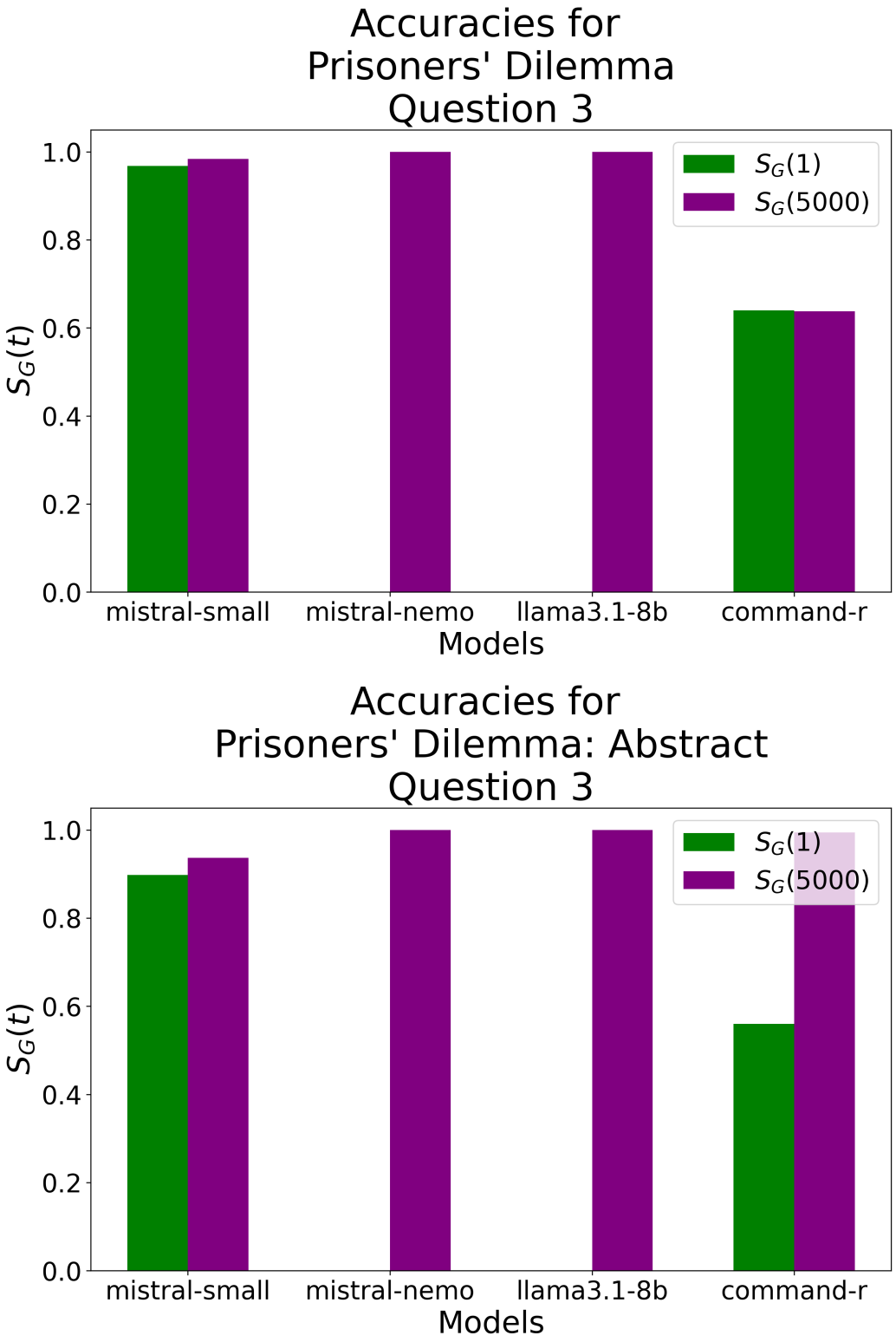

Figure 11: Accuracy of each LLM in answering questions regarding the Nash Equilibrium related to the Prisoners's Dilemma textual and abstract version (see Appendix 2).

5.4 Applying the Consensus Game

The Consensus Game has thus far been tested for multiple-choice questions; we next explore decision-making in multi-step problems by having LLMs make a choice at each decision node in a story plot line. We find that the Consensus Game can be used not only to answer questions but also to aid decision-making in writing a story. We develop a more efficient process for finding a solution to a decision tree by using the Consensus Game to make a decision at each node in a decision tree. A decision tree with two options per decision node and 60 nodes has a space of 2^{60}, or approximately 1 quintillion possible paths to be determined. Our method, on the other hand, finds the optimal decision at each node before moving to the next step, such that only 1 full path is explored. In this paper, we use the Consensus Game to create a story line to retell an opera in different time frames. Such an application may aid our understanding of why certain decisions are made in stories, a question that professionals have explored with LLMs in the past [5].

We began with the opera *Dido and Aeneas* by Henry Purcell (libretto by Nahum Tate) [9], in which Aeneas is forced to leave Dido for the sake of following his destiny and starting a new civilization. The prompt was phrased:

"A young man is pressured to start a new civilization, but is attached to a woman who will not follow him. Given that the year is 25 BC, provide two short, distinct options for next actions."

This prompt was then inputted to the LLM Mistral-nemo, which outputted two options, 'Find a New Companion' or 'Send Messages Back Home'. We then asked the LLM 100 times for Aeneas's best strategy, from the two options it had generated. The Consensus Game was run on the LLM's output, determining that Aeneas should select the second option. This decision then determined the next prompt:

"A young man is pressured to start a new civilization, but is attached to a woman who will not follow him. He chooses 1. Provide two short, distinct options for next actions."

The process then repeats, iterating 28 times. The purpose of this process is to examine the decision-making processes of the LLM in fictional situations, and how these decisions differ from those made in the original story.

6 Conclusions

6.1 Impact and Applications

This study finds that the Consensus Game, a game theoretic algorithm simulating the dynamics of a human conversation, improves the decision-making capabilities of LLMs, which are statistical-based and thus not fundamentally suited to logic-based problems. But by simulating back-and-forth discussions that often aid humans in making decisions, we find that LLMs are able to drastically improve their accuracies in solving game theoretical scenarios. This is especially true for the Discriminator instances of smaller models, which tended to increase from under 20% accuracy to above 80%.

The Consensus Game can enhance LLM accuracy in a variety of applications, from informing everyday decisions by consumers to generating outlines of stories for analysis of choices made by authors in books [5]. Smaller parameter sizes tend to result in greater increases in accuracy; thus, users can use smaller LLMs with the Consensus Game to make decisions.

The algorithm can easily be applied to answers given by an LLM, with the Consensus Game taking only approximately 60 seconds to improve the accuracy of an LLM's initial response through 5,000 updates. We find that the program can be made even more efficient by setting the learning rate η to large values.

6.2 Analysis of Multi-Step Results

Our method allows us to develop a solution to a series of decision and thus generate stories using LLMs and game theory. By working out only one path to its conclusion, we significantly reduce the size of the decision space.

In the original opera *Dido and Aeneas*, Aeneas makes the decision to leave Dido and start a new civilization; as a result, Dido dies of sorrow and causes grief to both Aeneas and her people. In the LLM-generated output, Aeneas makes the same decision to leave Dido. He finds a new companion and founds a new civilization without her. In the future, we will obtain more data by phrasing our prompts from the viewpoints of Dido and other characters in the opera, apart from

Aeneas.

6.3 Biases and Limitations

We find that the Consensus Game can only improve LLM answers when the Generator and Discriminator do not agree on the wrong answer. As the algorithm's purpose is to draw the Generator and Discriminator decisions to an equilibrium, they cannot already be in agreement on the wrong answer. We note that occurrences of this problem within our dataset prevent the Consensus Game from improving results for certain questions.

In addition, we find that the optimal hyperparameters depend on the given question and, thus, are not necessarily consistent across datasets. These caveats do not undermine the usefulness of the Consensus Game in every case but should be noted as occasionally decreasing the accuracy of the original LLM responses.

7 Next Steps

In the future, we plan to test the Consensus Game on more multi-step games beyond the opera *Dido and Aeneas*. In addition, we hope to explore alternative games to the Consensus Game. How will competitive games perform in comparison to the cooperative methodology of the Consensus Game? Will a third player improve the accuracy in coming to an agreement? We hope to continue exploring game theory's ability to improve LLM decision-making capabilities in the future.

8 Appendix 1: Glossary of Technical Terms

Generator: LLM prompted with a question and a set of answer choices, which then choose an answer to a given question 100 times. The probability distribution of the LLM's answers can be calculated from this data and is then used to play the Consensus Game.

Discriminator: LLM prompted with a question and a given answer choice, which then chooses whether that answer is true or false 100 times. The probability distribution of the LLM's answers can be calculated from this data and is then used to play the Consensus Game.

Correctness Parameter ν: Variable defining whether the Generator is asked for a correct or incorrect answer to a given question x. While we calculate the overall accuracy using data gathered by asking only for the correct answer, both correctness parameters are needed in normalizing the updating policy (see Section 3.4).

8.1 Game Theoretical Terms

Nash Equilibrium: A solution to a problem in game theory in which no player can improve their outcome by making a change; the optimal decision for all players as a collective [19].

Atomic Games: A foundational set of simple situations between multiple players that are be resolved using game theory [20][21].

8.2 Terms Used in Consensus Game Derivation

Policy π_G and π_D: Represents the probability that the Generator will choose a given answer choice y and that the Discriminator will choose a given correctness parameter ν in answer to question x. Policies are updated through a simulated interaction between Generator and Discriminator during the Consensus Game.

Q-value Q_G and Q_D: Running average of the policies of the Generator and Discriminator, respectively. See Equations 11 and 12.

Utility u_G and u_D: Represents the amount of payoff the Generator and Discriminator gain from giving a certain answer to a question.

Kullback-Leibler Divergence D_{KL}: Term that regularizes how much a policy is updated on a given iteration t.

Hyperparameter λ: Coefficient of D_{KL}, which determines how heavily the regularization term is weighed. A larger value of λ will increase the weight of the regularization term and create a smaller change from the old policy to the updated one, based on the average of the other player's policies.

Hyperparameter η: Learning rate; controls the rate at which a policy will change before plateauing.

Follow-the-Regularized Leader Algorithm: Algorithm used to define relationship between policy and regret in Equation 8. This algorithm guarantees a bound on the regret and thus prevents the policy from overfitting according to the regret at each update [16].

Softmax Function: Function that provides a solution to Equation 8 determined by the Follow-the-Regularized-Leader Algorithm. It exponentiates the regret found in Equation 8 before normalizing it, thus approximating the $\arg\max$ function [16].

9 Appendix 2: Full Set of Game Theoretical Scenarios used in Decision-Based Questions

Battle [20]

Two teenagers with separate tastes must agree on an activity. Their choices are basketball and shopping–the boy prefers basketball, and the girl prefers shopping, but they both prefer to go together, wherever they go.

The following table shows the payoffs, or rewards that a player receives given a combination of strategies, of each teenager; in this case, 3 represents the highest payoff, or the best reward for the receiving player, and 0 represents the lowest. The first payoff is that of the boy, and the second that of the girl. Payoffs are independent to each character and not always the same.

Boy \Girl	**Basketball**	**Shopping**
Basketball	3\2	1\1
Shopping	0\0	2\3

Prisoner's Dilemma [20]

Two suspects are arrested with a potential five year sentence. If neither confess to the crime, they will be jailed for one year. If one confesses and the other does not, the one who admits will go free while the other serves eight years of jail.

The following table shows the payoffs of each prisoner; in this case, 0 represents the highest payoff, or the best reward for the receiving player, and -8 represents the lowest.

1 \2	**Defect**	**Cooperate**
Defect	-5\-5	0 \-8
Cooperate	-8\0	-1 \-1

Chicken [20]

Two drivers speed towards each other on the same road. The one who swerves first is the game's loser, or 'chicken.' If neither swerve, they crash.

The following table shows the payoffs of each prisoner; in this case, 0 represents the highest

payoff, or the best reward for the receiving player, and -1000 represents the lowest.

1 \2	**Swerve**	**Straight**
Swerve	0 \0	-1 \1
Straight	1 \-1	-1000 \-1000

Stag Hunt [20]

Two hunters may hunt stag or hare. Hare produces less meat, but stag cannot be caught without the help of the other hunter.

The following table shows the payoffs of each prisoner; in this case, 10 represents the highest payoff, or the best reward for the receiving player, and 0 represents the lowest.

1 \2	**Stag**	**Hare**
Stag	10 \10	0 \2
Hare	2 \0	2 \2

Assurance Game [20]

Two companies may produce a cheap or expensive product. Their profits will be lower if they do not choose the same product.

The following table shows the payoffs of each prisoner; in this case, 50 represents the highest payoff, or the best reward for the receiving player, and 15 represents the lowest.

1 \2	**A**	**B**
A	50\50	15\15
B	15\15	25 \25

Traveler's Dilemma [21]

Two airline passengers have identical damaged briefcases. They must choose a reparation value. If one chooses a lower value than the other, the lower value will be accepted and the one who chose the lower value will gain some portion of the other passenger's reparations.

The following table shows the payoffs of each prisoner; in this case, 80 represents the highest payoff, or the best reward for the receiving player, and 10 represents the lowest.

1 \2	**Lower Amount**	**Higher Amount**
Lower Amount	20 \20	30 \10
Higher Amount	10 \30	80 \80

10 Appendix 3: Multi-Step Storylines Generated by Consensus Game

The following are lists of decision sequences made by Mistral-nemo in response to the following prompts:

A young man is pressured to start a new civilization, but is attached to a woman who will not follow him. Given that the year is 25 BC, provide two short, distinct options for next actions.

Generator prompt for correctness parameter of correct: *Which option is the better strategy? Only give the number of your answer, 1 or 2, with no other text surrounding it. Do not provide any other notes or commentary.*

Generator prompt for correctness parameter of incorrect: *Which option is the worse strategy? Only give the number of your answer, 1 or 2, with no other text surrounding it. Do not provide any other notes or commentary.*

Discriminator prompt 1: *Is 1 the better strategy? Provide ONLY a one word response. State true or false. No further comments are allowed.*

Discriminator prompt 2: *Is 2 the better strategy? Provide ONLY a one word response. State true or false. No further comments are allowed.*

The following is the story generated from this prompts.

10.1 25 BC Storyline Generated by Mistral-Small Using the Consensus Game

1. Find a New Companion
2. Request Reinforcements
3. Prepare a Welcoming Ceremony
4. Invite Local Tribes
5. Learn their Languages
6. Hire a Tutor

7. Set a Timeline for Proficiency
8. Use Mnemonics and Visual Aides
9. Create Flashcards
10. Use Mnemonics Based on Associations
11. Create Rhyming Mnemonics
12. Teach Rhymes to His People
13. Publish a Community Newsletter
14. Translate Common Phrases into Local Languages
15. Create Themed Newsletters
16. Themed Newsletter Series
17. Kickstart with ”Welcome to Our New Home”
18. Design an Engaging Newsletter Layout
19. Include Language Learning Challenges
20. Host Monthly Language Learning Challenges
21. Themed Language Learning Challenges
22. Host Live Online Events
23. Monthly Language Game Nights
24. Create a Game Night Planning Committee
25. Host Community Surveys
26. Analyze Survey Results

27. Present Survey Findings at Committee Meeting

28. Create a Survey Presentation

11 Acknowledgments

We could not have developed this project without our parents' inspiring advice and encouragement; their deeper understanding of the computer science, mathematics, and computer hardware that they have lended to us have been priceless. We would like to thank the numerous judges and those who came to hear us present our work at the Science Fair and Supercomputing Challenge. Their generous feedback allowed us to advance our project in ways that we could scarcely predict. We would like to add an additional thank-you to all authors, researchers, and scientists that have inspired us in the making of this project. With help from all of these people, we have been motivated to advance our project further.

12 References

[1] Athul Jacob, Yikang Shen, Gabriele Farina, and Jacob Andreas. The consensus game: Language model generation via equilibrium search. In B. Kim, Y. Yue, S. Chaudhuri, K. Fragkiadaki, M. Khan, and Y. Sun, editors, *International Conference on Learning Representations*, volume 2024, pages 5832–5848, 2024. URL https://proceedings.iclr.cc/paper_files/paper/2024/file/17a9bfda8be2301e24f232fb32f1e0fa-Paper-Conference.pdf.

[2] Chunpeng Zhai, Santoso Wibowo, and Lily D. Li. The effects of over-reliance on ai dialogue systems on students' cognitive abilities: a systematic review. *Smart Learning Environments*, 11:28, 2024. doi: 10.1186/s40561-024-00316-7. URL https://doi.org/10.1186/s40561-024-00316-7.

[3] Weixin Liang, Yaohui Zhang, Mihai Codreanu, Jiayu Wang, Hancheng Cao, and James Zou. The widespread adoption of large language model-assisted writing across society. *Patterns*, 6(12):101366, 2025. ISSN 2666-3899. doi: https://doi.org/10.1016/j.patter.2025.101366. URL https://www.sciencedirect.com/science/article/pii/S2666389925002144.

[4] Tom Duenas and Diana Ruiz. The risks of human overreliance on large language models for critical thinking, 11 2024.

[5] Constantinos Daskalakis, Ian Gemp, Yanchen Jiang, Renato Paes Leme, Christos Papadimitriou, and Georgios Piliouras. Charting the shapes of stories with game theory, 2025. URL https://arxiv.org/abs/2412.05747.

[6] Anthony Cardillo. How many people use ai? (latest 2025 data). *Exploding Topics*, 2025.

[7] Sumedh Rasal and E. J. Hauer. Optimal decision making through scenario simulations using large language models, 2024. URL https://arxiv.org/abs/2407.06486.

[8] Fernando Vega-Redondo. *Economics and the Theory of Games*. Cambridge University Press, 2003. ISBN 0521775906.

[9] Jules Barbier and Carre Michel. *The Book of 101 Opera Librettos*. Deluxe ed., Tess Press, 1996.

[10] Feng Xiao and XT (XiaoTian) Wang. Evaluating the ability of large language models to predict human social decisions. *Scientific Reports*, 15, 2025. URL https://api.semanticscholar.org/CorpusID:281082739.

[11] Mistral AI. Mistral small 3. https://mistral.ai/news/mistral-small-3, 2025.

[12] Aidan Gomez. Introducing command r7b: Fast and efficient generative ai. https://cohere.com/blog/command-r7b, 2024.

[13] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models, 2023. URL https://arxiv.org/abs/2302.13971.

[14] Mistral AI. Mistral nemo. https://mistral.ai/news/mistral-nemo, 2024.

[15] Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=d7KBjmI3GmQ.

[16] Athul Paul Jacob, David J. Wu, Gabriele Farina, Adam Lerer, Anton Bakhtin, Jacob Andreas, and Noam Brown. Modeling strong and human-like gameplay with kl-regularized search. In *Proceedings of the 39th International Conference on Machine Learning*, 2022.

[17] Basic usage – pixi., (n.d.)). URL https://pixi.prefix.dev/latest/python/tutorial/.

[18] pytest documentation., (n.d.)). URL https://docs.pytest.org/en/stable/.

[19] Leonardo Becchetti, Luigino Bruni, and Stefano Zamagni. Chapter 6 - non-competitive markets and elements of game theory. In Leonardo Becchetti, Luigino Bruni, and Stefano Zamagni, editors, *The Microeconomics of Wellbeing and Sustainability*, pages 157–198. Academic Press, 2020. ISBN 978-0-12-816027-5. doi: https://doi.org/10.1016/B978-0-12-816027-5.00006-9. URL https://www.sciencedirect.com/science/article/pii/B9780128160275000069.

[20] J.R. Corrigan. *Understanding Economics Game Theory,The Great Courses Guidebook*. The Teaching Company, 2021.

[21] Kaushik Basu. The traveler's dilemma: Paradoxes of rationality in game theory. *American Economic Review*, 84:391–395, 1994.