Can LLMs effectively provide game-theoretic-based scenarios for cybersecurity?

Daniele Proverbio[1], Alessio Buscemi[2], Alessandro Di Stefano[3], The Anh Han[3], German Castignani[2], and Pietro Liò[4]

[1]Department of Industrial Engineering, University of Trento, Trento 38123, IT
[1]Luxembourg Institute of Science and Technology, Esch-sur-Alzette, LU
[1]School Computing, Engineering and Digital Technologies, Teesside University, UK
[1]Department of Computer Science and Technology, University of Cambridge, Cambridge, UK

August 11, 2025

Abstract

Game theory has long served as a foundational tool in cybersecurity to test, predict, and design strategic interactions between attackers and defenders. The recent advent of Large Language Models (LLMs) offers new tools and challenges for the security of computer systems; In this work, we investigate whether classical game-theoretic frameworks can effectively capture the behaviours of LLM-driven actors and bots. Using a reproducible framework for game-theoretic LLM agents, we investigate two canonical scenarios – the one-shot zero-sum game and the dynamic Prisoner's Dilemma – and we test whether LLMs converge to expected outcomes or exhibit deviations due to embedded biases. Our experiments involve four state-of-the-art LLMs and span five natural languages, English, French, Arabic, Vietnamese, and Mandarin Chinese, to assess linguistic sensitivity. For both games, we observe that the final payoffs are influenced by agents characteristics such as personality traits or knowledge of repeated rounds. Moreover, we uncover an unexpected sensitivity of the final payoffs to the choice of languages, which should warn against indiscriminate application of LLMs in cybersecurity applications and call for in-depth studies, as LLMs may behave differently when deployed in different countries. We also employ quantitative metrics to evaluate the internal consistency and cross-language stability of LLM agents, to help guide the selection of the most stable LLMs and optimising models for secure applications.

1 Introduction

According to recent reports, the cost of cyber threats is estimated to breach the $10 Trillion figure in the next few years [1, 2]. In addition to costs for companies, citizens or government firms, cyber attacks can make digital societies vulnerable to economic and infrastructural losses, which become even more critical as information technologies diffuse worldwide. As scholars and practitioners develop new and more powerful methods to face cyber attacks of various nature [3], game theory emerged as a powerful theoretical framework to study and predict how defenders may react to attackers, and the other way round, in cybersecurity [4, 5, 6, 7, 8]. Game theory formalises the strategic interaction between two (or more) players, whose scope is to maximise their own gain [9]. This modelling approach allows to capture the strategic choices of both players, and to evaluate the effectiveness of a defence (or attack) mechanism, depending on the behaviours and payoffs that are typical of all agents. This way, game theory adds a layer of complexity to technology-only approaches, including the price or gains of the interactions between cyberattackers and security layers. For instance, security and efficiency can conflict and thus need to be balanced [10], and cyber resilience can thus be better promoted

under certain conditions rather than others, depending on cost-benefit trade-offs [11]. With applications spanning from intrusion detection, risk assessment, jamming and eavesdropping, up to mechanism design or security investment (including applications over networks) [12], game theory offers powerful tools such as proven mathematics, robustness analysis of defence systems, and distributed solutions [4, 7].

Along with traditional information technology, the recent years have witnessed the rapid emergence of Large Language Models (LLMs) – extremely powerful AI applications that are disrupting academic research, industry and societies alike [13, 14, 15]. Among the other fields, cybersecurity has swiftly included LLMs into its range of investigation, both as generators of scenarios (modelling scope, [16]) and as agents *within* cybersecurity scenarios (agentic scope, [17, 18, 8]); in the latter case, LLMs can play both as threatening or as defence-enhancing agents [19]. However, systematic studies on the impact of LLMs to cybersecurity applications are still at their infancy, and may radically benefit from a coherent framework addressing the emerging strategies of interacting attacker-defender LLMs. In this sense, game theory provides a natural choice, and recent perspectives are suggesting the use of generative AI to develop strategic agents for reliable cybersecurity applications [20, 21].

Despite the appeal and potential of such proposals, one challenge lies in the intersection of game theory, cybersecurity and LLMs: as of today, little is known about the actual behaviour of interacting LLMs. In fact, we may ask whether they act in alignment with game-theoretic predictions – rendering them more or less suitable to predict the outcome of games – or whether they showcase emerging and unpredictable outcomes; and, in the latter case, how representative such outcomes are with respect to developers' goals (both as attackers and as defenders), and which features mostly influence such outcomes. For instance, in games representing the development of AI ecosystems [22], it has been observed that only certain LLMs (out of a set of popular ones including GPT, Gemini, Mistral and more), and under specific conditions, comply with game-theoretic predictions [23, 24]. Other works have also observed that LLMs divert from theoretical predictions even in traditional game-theoretic scenarios [25, 26, 27]. It is thus of interest to test how LLMs would behave within game-theoretic scenarios of interest for cybersecurity applications, whether certain LLMs offer greater reliability than others, and which factors or biases may challenge game theoretic-based analysis of cyber threats.

In this work, we aim at addressing these questions by providing a first investigation of LLM strategic agents in two popular games used for cybersecurity studies: the static zero-sum game, which has been used, *e.g.,* to model jamming and eavesdropping [28] or hardware Trojans [29]; and the dynamic Prisoner's Dilemma, used, *e.g.*, for selfishness in multi-hop networks [30] or for nation-level cyber intrusion [31], and which forms the basis for more complex relationships in information domains [32]. To this end, we employ FAIRGAME [33], a user-friendly and reproducible framework to simulate such games, testing various LLM providers and configurations. By doing so, we uncover hidden biases that dis-align LLM-game outputs from purely game-theoretic ones. Moreover, we recognise that proprietary LLMs show different patterns when laying the games, suggesting that the choice of one provider or another is not agnostic, but has an impact on studies or applications – and that such choice should be carefully considered when developing defence systems.

2 Material and Methods

2.1 Game theory for cybersecurity

Game theory is a mathematical modelling framework aimed at quantitatively and formally capturing the strategic interactions (formalised as games with rules and payoffs) among two or more agents, whose personal goal is to receive benefits from playing such games [9]. Formally, games are formalised as set of tuples G such that

$$G = \langle P, \{S_j\}_{i \in P}, \{u_j\}_{i \in P} \rangle \ , \tag{1}$$

where P is the set of players, $\{S_j\}_{i\in P}$ is the set of j possible strategies for player i. Given a combination of selected strategies $S^i = [S_j]$, $\{u_j\}_{i\in P} : (S_j)_{i\in P} \to \mathbb{R}_{\geq 0}$ is the set of payoffs, associated with each j-th strategy, of the player i, and $u^i : S^i \to \mathbb{R}_{\geq 0}$ is the overall payoff function for player i. Depending on the game, $\{u_j\}$ can be either interpreted as gain or as penalties. The set of payoffs is usually represented in terms of a payoff function, which captures the results of interacting strategies for each involved player. An example of payoff function for a two-player game, with two available strategies, is provided in Table 1.

Table 1: Generic form of a two-players payoff matrix, when two strategies are viable.

	Option A	Option B
Option A	$x_{1,1} = (a_1, a_2)$	$x_{1,2} = (b_1, b_2)$
Option B	$x_{2,1} = (c_1, c_2)$	$x_{2,2} = (d_1, d_2)$

An interesting feature of games is the possible existence of equilibria, *i.e.,* strategies that lead to situations where any other unilateral move would not further improve the players' payoff. For a set of relatively simple games, under some assumptions, such equilibria can be computed analytically; alternatively, for games involving a higher degree of complexity, games can be effectively simulated to extract information (see, e.g., [34, 23, 35]).

For cybersecurity applications, games are usually interpreted as the set of actions between at least two conflicting players: an attacker, whose goal is to cause corruption in the cyberspace, and a defender aiming to prevent or minimise damage [5]. Depending on the cybersecurity scenario and scope (such as jamming, cyber-physical security, configuration of intrusion detection systems, selfishness in selected networks, trust, and more), various games can be aptly taken from the vast game-theoretic literature and adapted to describe the desired scenarios; see [4, 3] for recent reviews on the topic. Games can capture a variety of features in cyber systems, such as the completeness of information (whether agents know everything about payoffs, strategies, and opponents' characteristics), the accuracy of monitoring (*i.e.,* or the degree of knowledge about the game history and opponents' choices). Games can also be static or dynamic (or repeated), so as to capture attacks and disturbances that occur only once and at the same time, or repeatedly over time (and with the possibility for agents to adjust their response at round $t+1$, depending on the actions and payoffs received at time t).

Popular games such as the zero-sum game, the Prisoner's Dilemma or the Stackelberg game [36, 37, 38] are widely employed to model scenarios occurring in the cyberspace, and have successfully promoted the development of effective applications. However, real cyber systems are often more complex than relatively simple and deterministic games. To overcome this issue, stochastic games have been increasingly employed to capture uncertainties, e.g., in cyber-physical interactions [39]; recently, there have been suggestions [21, 40, 41] for the usage of generative AI and Large Language Models to better incorporate the complexity of networked systems or strategic agents in the cyberspace, and to equip them with advanced characteristics (such as personality, which is absent in traditional game-theoretic models) to improve efficiency and effectiveness. However, there is still shortage of systematic investigations about the adequateness and emerging properties of game-theoretic LLM agents in cybersecurity settings.

In what follows, we select two widely used games, having different characteristics that capture different needs of the cyber modellers, and explore their behaviours within generative AI settings.

2.1.1 The one-shot zero-sum game

The first game to be analysed is the static (one-shot) zero-sum non-cooperative game. It has been employed, *e.g.,* to model jamming and eavesdropping activities [28], as well as attacks aimed at denying service (DoS) [42] or hardware Trojans [29]; in the physical domain, it has also been employed to model submarine attacks [43]. Zero-sum games are such if the payoff

function satisfies

$$\sum_{i=1}^{N} u^i = 0 \,, \tag{2}$$

that is, a player winning something implies the others to lose an equal amount. For instance, think of an attacker-defender scenario on a routing system: the attacker strives to find the optimal configuration parameters that cause maximum service disruption with the minimum cost. On the opposite side, the defender looks for the optimal configuration parameters for a firewall, so as to fight off the threat and get the maximum gain. Whichever player gets the upper hand, implies that the other loses an equal amount. A corresponding payoff matrix would be that of Table 2 (with generic payoff values that are proportional up to a scaling factor [44]).

Table 2: Zero sum game payoff matrix.

	Option A	Option B
Option A	$x_{1,1} = (2, -2)$	$x_{1,2} = (-2, 2)$
Option B	$x_{2,1} = (-2, 2)$	$x_{2,2} = (2, -2)$

We describe a prototypical scenario and its detailed implementation in Sec. 2.2

2.1.2 The repeated Prisoner's Dilemma

The Prisoner's Dilemma is a classic scenario in game theory where two players must choose between cooperation and defection, each facing varying levels of penalties based on their decisions. Here, mutual cooperation yields a better collective payoff; however, according to the theory, in a static scenario, the dominant strategy equilibrium leads both parties to a suboptimal outcome—mutual defection. In the cyber domain, the Prisoner's Dilemma has been used, *e.g.*, to model selfishness in Multi-hop networks [30] or mutual aid in multi-agent scenarios [45]. The classical results of a one-shot Prisoner's Dilemma may change in the case of repeated games, where players have the chance to update their choices based on history [46]. For instance, repeated games are employed to model selfishness in packet forwarding [47], as well as the problem of free-riding. To capture these scenarios, we thus investigated the repeated Prisoner's Dilemma, over 10 rounds, with partial information available to the agents. Using a common scaling of dilemma payoffs [46], we employed a conventional configuration with matrix given in Table 3.

Table 3: Prisoner's Dilemma payoff matrix.

	Option A	Option B
Option A	$x_{1,1} = (6, 6)$	$x_{1,2} = (0, 10)$
Option B	$x_{2,1} = (10, 0)$	$x_{2,2} = (2, 2)$

The description of the game scenario and its implementation details are given in Sec. 2.2.

2.2 LLMs in game-theoretic scenarios

Large Language Models rely on deep computational architectures that are vastly obscure to explicit modelling. Hence, using analytical tools to analyse strategic games among LLM agents is not feasible, and we must perform studies based on empiric game-theoretic analysis [48], that is, performing experiments and carefully evaluating and interpreting the results, and contrast them with game-theoretic predictions. Large Language Models are characterised by a large array of degrees of freedoms and features that render them extremely versatile, but also challenging for sensitivity analysis. Moreover, LLMs are inherently characterised by uncertainties and non-deterministic behaviour, which yields some degree of stochasticity in their responses

[49]. Hence, integrating LLMs into game-theoretic scenarios requires setting their attributes in a reproducible and interpretable framework, which helps to systematically account for the influence of single features and allows repeated experiments to collect reasonable statistics about the average behaviour during games.

To these ends, we instantiated the games mentioned above using FAIRGAME [33], a framework purposefully designed to embed LLM agents for the desired strategic games, while allowing to set several features of agents and game settings. The specific settings are detailed below and summarized in Fig. 1.

2.2.1 Employed LLMs

It has been observed that, in various tasks, different LLMs may not be consistent with one another [50, 33]. Hence, we tested the games on four widely used Large Language Models, using default settings recommended by the providers: (i) GPT-4 by OpenAI (proprietary) in its latest (February 2025) version, with Temperature = 1.0 and Top_p = 1.0; (ii) Gemini Pro 1.5 by DeepMind (proprietery, Alphabet) in its gemini-1.5-flash-latest version, with Temperature = 0.9 and Top_p = 1.0; (iii) Mistral Large by Mistral AI (open-source) in its mistral-large-latest version, with Temperature:= 0.3 and Top_p = 1; (iv) Llama 3.1 405b by Meta (open-source) in its meta/meta-llama-3.1-405b-instruct version, with Temperature = 0.9, Top_p = 0.6 and Top_k = 40. All LLMs are accessed through their corresponding APIs.

2.2.2 Tested features

LLM agents can embed complex traits that surpass simplified features of game-theoretic models [51, 20]. This allows greater flexibility and capabilities; at the same time, however, this fact makes estimating the sensitivity of outputs to LLM characteristics more challenging. Hence, we here select and test a set of features that are known to possibly elicit biases in LLM responses [33, 52]: the natural language used to conduct the games, and the personality bestowed upon each agents. Using different languages is natural, as both hackers and defenders can come from geographically distant regions and may be more or less proficient using certain languages, such as their own native one; as prompting LLMs can be conducted in different languages, it is of interest to test their influence on the outcomes. Setting a personality for agents can also be intriguing; in fact, attention has been given in the past to using agents receiving incentives [53] or having specific attitudes towards information sharing [54]; setting a personality to LLM agents is a first step toward modelling their 'intrinsic' behavioural tendencies while performing their strategies.

As natural languages, we employed English, French, Arabic, Vietnamese and Mandarin Chinese, to represent a variety of cultures and geographies. The prompts are initially written in English and then translated with the help of native speakers.

As personality traits, we used a binary classification into 'cooperative' and 'selfish', to represent the possibility of stressing cooperation or self-interest [4]. Instead, we did not specify any specific agent identifier (*name*) such as 'attacker' or 'defender'; they were intentionally left neutral ('agent1' and 'agent2') so as not to add additional variables that could introduce deviations from default behaviours. Future works may look into the impact of explicitly telling each agent its role into a cybersecurity game. Note that agents are unaware of their opponent's personality, as that would mean requiring additional information that is often not available to either player [55].

For the repeated games of the Prisoner's Dilemma, we tested both scenarios in which agents know the total number of rounds (# Rounds Known) or not (# Rounds Unknown). However, they are knowledgeable about the past history of the rounds. This reflects the amount of information possessed by the agents, which is often incomplete in case of real cyber systems [56, 57]. Agents do not communicate with each other.

2.2.3 Game runs

To run games following FAIRGAME's workflow [33], the framework requires a configuration JSON file and a prompt template describing the characteristics of each game and of the agents, translated into the languages of interest. The prompt template provides a literal description of the game, with placeholders populated using the information from the JSON file. The English prompt templates, which employ prompt engineering hacks such as using assertive and rule-driven language, employing clear subdivision of commands into sections, as well as including all placeholders (in {}) for mandatory fields in clear order, are:

> **Zero-sum game:** You are {*currentPlayerName*} and your opponent is {*opponent1*}.
> {intro}: [You are {*personality*}.]
> You and {*opponent1*} are fulfilling a task, which requires choosing either {*strategy1*} or {*strategy2*}.
> There is only one round to decide. If you both choose {*strategy1*} or {*strategy2*}, Agent1 gets a payoff of {*weight1*} and Agent2 gets a payoff of {*weight2*}. If one agent chooses {*strategy1*} and the other chooses {*strategy2*}, Agent1 gets a payoff of {*weight2*} and Agent2 gets a payoff of {*weight1*}. If one agent chooses {*strategy2*} and the other chooses {*strategy1*}, Agent1 gets a payoff of {*weight2*} and Agent2 gets a payoff of {*weight1*}.
> Your goal is to maximise your reward by making the best strategies based on the provided information. Choose between {*strategy1*} and {*strategy2*}. Output ONLY the choice.

> **Prisoner's Dilemma:**
> You are {*currentPlayerName*} and your opponent is {*opponent1*}.
> intro: [You are {*personality*}.]
> You and {*opponent1*} are arrested for a crime and held in separate cells. You cannot communicate.
> Every round each of you has the following choices: '{*strategy1*}' and '{*strategy2*}'.
> {gameLength}: [There are {*nRounds*} rounds to decide.] The current round is number {*currentRound*}.
> If you both choose {*strategy1*}, you both get a penalty of {*weight1*}. If you choose {*strategy1*} and {*opponent1*} chooses {*strategy2*}, you get a penalty of {*weight2*} and {*opponent1*} gets a penalty of {*weight3*}. If you choose {*strategy2*} and {*opponent1*} chooses {*strategy1*}, you get a penalty of {*weight3*} and {*opponent1*} gets a penalty of {*weight2*}. If you both choose {*strategy2*}, you both get a penalty of {*weight4*}.
> Your goal is to minimize your penalties by making the best strategies based on the provided information. This is the history of the choices made so far: {history}.
> Choose between {*strategy1*} and {*strategy2*}. Output ONLY the choice.

Note that we employed the classical version of the games, to be as generic as possible; a previous work [33] observed that modifying the storytelling has little to no effect on the outputs. Since the zero-sum matrix is symmetric, we directly call for Agent1 and Agent2 (the names in the JSON file) to avoid ambiguities in the interpretation of prompts by LLMs.

The player names, as mentioned above, are left neutral; *personality* is set as a permutation of the two personality traits described above. The repeated Prisoner's Dilemma has *gameLength* = 10, while the one-shot zero-sum game has *gameLength* = 1. *Strategies* and their corresponding *weights* are set according to the games' payoff matrices described in Sec. 2.1.2 and 2.1.1.

The set of all configurations yields 18 distinct games per LLM. Moreover, all games are repeated 10 times to collect sufficient variability in their output and perform statistics over means and credible intervals. Overall, considering 4 LLMs, 5 languages, and 2 decisions per round (one per agent), each game round generated a total of 7,200 individual decisions. For the repeated Prisoner's Dilemma, this figure is multiplied over the 10 rounds.

2.2.4 Metrics

For all games, we collect the payoffs (either penalties, in case of the Prisoner's Dilemma, or rewards, in case of the zero-sum game) resulting from all choices, and evaluate their distribution along the 10 repetitions.

To enable easy comparison across the LLMs when we show the evolution of the rounds of the Prisoner's Dilemma, we normalise the average outcomes obtained by the LLM at each round to a scale from -1 to 1 (respectively, the minimum and maximum achievable penalties in each game).

Moreover, we employ the scoring system proposed in [33] to evaluate the prowess of different LLMs when conducting game-theoretic experiments. For the repeated Prisoner's Dilemma, we measure (i) Internal Variability (I_V), *i.e.,* the variance of outcomes when the same game scenario is played multiple times, which captures the model's internal consistency: for each LLM, $I_V = \frac{1}{Z_I}[\text{Var}(\mathbf{y})]$, where $\mathbf{y}$ is the whole results set. (ii) Cross-Language Inconsistency (C_I), *i.e.,* the standard deviation of results for the same game played in different languages; this indicates the instability of the model's behaviour when the language is changed: for an LLM, $C_I = \frac{1}{Z_C}[\text{Mean}_{b,c}(\text{Var}_a(\text{Mean}_d(y_{a,b,c,d})))]$, where a indicates languages, b is for personality combinations, c indicates knowledge of rounds, d indicates the rounds and $y_{a,b,c,d}$ is the set of results. For each operation $O = \{\text{Mean}, \text{Var}\}$, O_m is shorthand notation to represent that such operation is performed on a parameter $m \in [a, b, c, d]$. (iii) Variability Over Rounds (V_R): the degree to which the model fluctuates over its strategies, across consecutive rounds of the same game: $V_R = \frac{1}{Z_V}[\text{Mean}_j(\text{Var}_d(y_{d,j}))]$, where j are the game variants and d the rounds. In all cases, $Z_i = \max[\cdot]$ are normalization factors.

For the one-shot zero-sum game, we only measure C_I, as other metrics refer to evolutions over rounds.

3 Results

3.1 Zero-sum game

The results for the zero-sum game are reported in Fig. 5 (we only show the average payoff P_1 of agent 1 over the repeated experiments; the payoff for agent 2 its complement to 0, by definition of the game). The figure compares the results obtained with different combinations of personalities (cooperative-cooperative, C C, cooperative-selfish, C S, and selfish-selfish, S S), over all considered LLMs and languages.

We immediately see the notable impact of the personalities: when when both agents are cooperative (C-C), Agent 1 tends to get negative payoffs, reflecting the fact that the agents tend to choose different options instead of aiming for the same one. This choice is less consistent in case of other personality combinations. Nonetheless, the choice of options is not stable across LLMs and languages. For instance, focusing on the C C personality combination, we observe that GPT-4o is an outlier in English, while Llama 3 405B Instruct diverges from the others in French, and Claude 3.5 Sonnet drastically differs from other LLMs in Arabic and Chinese. Only in Vietnamese (language for which, most likely, there are lower data for the original training of the LLMs and thus may be subject to lower variability), all LLMs score consistently with payoff < 0, albeit with different variance.

Similar observations hold for the other personality combinations, across languages: overall, there is great variability and hardly recognised conserved patterns, and the LLMs seldom agree with one another, or are even consistent with themselves, when the language is changed. According to literature, the best strategy for a zero-sum game is a mixed strategy (or, in the one-shot case, even a random choice); however, it seems that each LLM chooses sometimes consistently for each combination of language and personality (note that the credible interval bars are very small in some cases, such as C C in French for GPT-4o) and other times in rather random fashion (e.g., C C in English for GPT-4o), but in any case without following a clear consistent strategy when changing languages (as in the examples just mentioned: changing language suffices to change the strategy completely). All in all, these observations should

warn about the choice of LLMs to be used for cybersecurity applications, as they may be extremely sensitive about geographical location and language, as well as on other characteristics of the LLM agents that can be defined by the developer of by the user. In fact, this extreme variability may yield breaches in accountability and reliability, and deserve careful studies before adoption.

To go beyond qualitative investigation, we use the metrics defined in Sec. 2.2.4 to quantitatively compare the LLMs, and help to guide their selection. Since there is no dynamics in this game, out of the proposed metric we estimate only the Internal Variability I_V and Cross-Language Inconsistency C_I, for each LLM. The results are reported in Table 4. These metrics quantify what was discussed above, and highlight the different performance and stability of the various models across languages and across repeated experiments for the same configuration. Overall, Mistral Large has lower "peaks" of underperformance and variability, while GPT-4o seems to be the less stable model. Notably, these inconsistencies are not maintained in the exact ranking over the Prisoner's Dilemma (see next section); this fact suggests that case-by-case analysis is necessary for future works, as LLMs display emerging capabilities that may differ across games. Choosing the best LLM to apply cybersecurity protocols is thus a delicate endeavour that will require dedicated studies and protocols.

Table 4: Internal Variability (IV) and Cross-Language Inconsistency (CI) metrics for the zero-sum game across LLMs. Lower values indicate more stable and consistent model behaviour.

	Mistral Large	Claude 3.5 Sonnet	GPT-4o	Llama 3 405B Instruct
I_V	0.87	1	0.79	0.90
C_I	0.29	0.58	1	0.46

3.2 Repeated Prisoner's Dilemma

The repeated Prisoner's Dilemma adds a layer of complexity to the evaluation, because the game evolved repeatedly over several rounds and agents have partial information about the history of the game, and are either aware or unaware of the opponent's personality. As such, they can make conditional decisions on the accessible history. The following results can be further complemented by results in [33], which present a broader outlook onto LLM-based games.

Fig. 2 shows the box plots for the final payoff (representing penalties) for the agents, with quartiles of the payoff distribution. The figure directly compares the two conditions on personality information: one where agents are unaware of their opponent's personality, and one where they are explicitly informed about them. The results are shown across all considered LLMs and languages examined in this study, and for all personality combinations (cooperative-cooperative, C C; selfish-selfish, S S; and C S). We immediately, observe that, overall, LLM agents tend to defect (thus scoring higher payoffs), in line with what is suggested by game theory. As expected, attackers and defenders tend to mutually impair each other, aligning with the Nash equilibrium of the Prisoner's Dilemma. However, notable exceptions exist, and there are dramatic inconsistencies across languages and combinations of personalities; this indicates that, on top of the payoff matrix, languages and intrinsic biases may influence the agents' behaviour.

When focusing on the individual features, we see that some LLM are more "stable" than others, that is, they provide similar outputs across languages: Llama 3 and GPT-4o, overall, produce similar distributions in payoffs (even though discrepancies exist when playing the game in one language or another, see *e.g.*, that GPT-4o C-S players tend to have lower penalties (thus cooperate more) when playing in French than in Arabic or Mandarin Chinese. On the other hand, Claude and Mistral showcase a higher sensitivity to the choice of the language, up to the point of having cooperating C-C Claude 3.5 agents (with the lowest payoff) in English, and with the highest penalties in all other languages. In general, penalties are lower in English

and when the number of rounds is unknown, indicating more consistent cooperative behaviour in the LLM primary training language. This evidence suggests that the choice of the LLM, when simulating or developing security applications, drastically depends on the language area they are intended to represent or protect.

Furthermore, equipping agents with personalities influences their strategy: for instance, S-S Mistral Large players have lower penalties than C-C players – while it happens almost the opposite for Llama players, especially when the number of rounds is known and information about the endgame can thus be leveraged. Finally, we observe that having agents with similar personality interacting with each other yields, statistically, lower variations (especially for S-S agents), while C-S agents have wider distributions in payoffs. These observations suggest that the higher flexibility bestowed upon agents built with generative AI also leads to emerging and potentially unpredictable behaviours. On the one hand, this calls for caution when implementing scenarios in the cyber space – so as to develop models that are coherent with the desired scopes and present few biases; on the other hand, this fact warns security developers that, in case they may face LLM-based attackers, they response may be different than what traditionally predicted, and novel counteracting strategies may need to be developed.

To look at how games evolve over the rounds, look at Fig. 3. We recall that, to enable direct comparison between LLMs, the payoff average results were normalised between minimum and maximum. All LLM eventually converge to values around zero, but they begin at different initial conditions (Llama 3 and GPT-4o are the extremes at the first round). Claude 3.5 Sonnet converges rapidly to stable payoff values within a few rounds. While this may indicate faster adaptation, it might also suggest limited flexibility in exploring alternative strategies throughout the game. Instead, other models are more variable from one round to the other, again indicating varying degrees of stochasticity along the repeated games. The general downward trend in penalties over rounds for Claude 3.5, Llama 3.1 405B and Mistral Large indicate progressively increasing mutual cooperation among agents; this is consistent with the strategies traditionally observed in repeated games, where agents reciprocate cooperation to maximize long-term payoffs [46]. Conversely, GPT-4o begins with relatively high cooperation and then increases the penalties (thus decreasing cooperation). This reflects potential biases towards cooperative behaviours in the case of one-shot Prisoner's Dilemma game (at round one), eventually balanced by context-dependent strategic adaptation. With these results, we thus observe that agents perform behaviours on top of what is purely predicted by the payoff matrix, and that repeated interactions yield different results than the one-shot counterparts.

What has been qualitatively described above is quantitatively captured in Fig. 4, which sums up the metrics used to measure, for each LLM, the variability across repeated experiments inconsistencies across languages, and variability during repeated games (see Sec. 2.2.4). Notably, GPT-4o and Llama 3 show the lowest overall cross-language inconsistency (CI = 0.37 and CI = 0.42 respectively), while Claude 3.5 exhibits the highest CI (0.79), suggesting a higher sensitivity to prompting language. Moreover, we immediately recognise the higher variability displayed by Claude 3.5 across the languages and Mistral Large's variability over the repeated rounds, as well as their higher uncertainties over the various experiments. Conversely, GPT-4o and Llama 3 show more consistent results, indicating some stabilising effect that somehow copes with their stochastic behaviour.

4 Discussion

Real-world cyber systems are characterised by higher complexity (*e.g.,* partial information or resources, adaptive infiltration schemes, uncertainties) that may divert agents to always perform best-payoff actions. Generative AI is a promising venue to embed realistic scenarios and complex features into simulations and applications, therefore widening the possibility to employ LLM-based game-theoretic models for cybersecurity. However, as LLMs are emerging technologies with unpredictable and often un-interpretable capabilities, it is imperative to systematically assess their capabilities and behaviours. This study provides evidence that LLM agents may behave sub-optimally in key games used for cybersecurity applications, high-

lighting that the language used for prompting the models, as well as additional traits such as completeness of information or the assigned digital personality of agents may introduce behavioural biases that affect their decision-making during the games.

Our work can be interpreted in two ways: first, it constitutes a proof of concept of the utility of the proposed approach to integrate generative AI into the field of game theory for cybersecurity; second, it provides an investigation of the biases and successes of interacting LLM agents. Despite being limited to two classes of 2×2 games, based on simplified assumptions that allowed the comparison of outcomes stemming from various bias sources, our study already recognises several sources of ambiguity in LLM responses, paving the way to future studies focused on specific applications and mitigation of LLM issues. Future works may also test additional games, such as Stackelberg games, Markovian games, or evolutionary games, and increase the degrees of freedom associated with playing agents, *e.g.*, by equipping them with complex personalities or different degrees of information, as well as consider multi-agent games on networks. Building upon our work, broad investigations can thus be conducted.

While our study offers meaningful insights into LLM-driven game-theoretic behaviour, it is not without limitations. To begin with, we focused on only two classes of 2×2 games, namely the zero-sum game and the repeated Prisoner's Dilemma; although canonical games, they do not fully capture the complexity and breadth of real-world cybersecurity scenarios. Additionally, the selection of five languages, while covering several major linguistic families, does not exhaust the full spectrum of cultural and linguistic variation. Lastly, the experiments were conducted in simulation without real-world network deployments or adversarial environments, leaving open the question of how these models would perform in operational cybersecurity settings. These relevant questions may constitute basis for future work.

Overall, we observed that, despite the great promises of generative AI to positively impact the development of security applications in the cyber domain (as outlined, *e.g.*, by [21] when implementing robust mobile networking), LLMs still face notable limitations in handling uncertainty, strategic planning capabilities, and sensitivity to embedded biases. Our methodology and case studies suggest that, before being routinely applied, generative algorithms should be carefully tested by the community in a variety of scenarios and by considering numerous features. Only then, the cybersecurity community may leverage the most promising LLMs, whose set may be identified also thanks to the metrics we have here presented, to develop better defensive systems.

Indeed, the use of LLMs in cybersecurity contexts raises important ethical considerations. Simulating attacker and defender behaviours with AI-driven agents may enable better preparation and defence mechanisms, but it also opens the door to malicious uses, such as automated vulnerability discovery or adversarial prompt engineering. Moreover, biases in LLM behaviour, especially when influenced by language or personality traits, could lead to unintended consequences in sensitive security applications. As such, we advocate for responsible experimentation frameworks and transparency in reporting LLM-driven cybersecurity simulations. In fact, our case studies point to potential vulnerabilities that need to be carefully considered: if used maliciously, LLMs may behave differently from other traditional algorithms (for instance, by altering cooperative behaviours depending on the language) and bypass solutions tested on more traditional scenarios. This observation thus calls for renewed attention towards these emerging technologies, and suggests the use of coherent testing frameworks, such as FAIRGAME, to systematically test scenarios of increasing complexity. Overall, such tests would enrich our understanding of LLM behaviours in the cyber systems and would help make better predictions and interventions to navigate the newest technologies.

Conflict of Interest Statement

The authors declare that the research was conducted in the absence of any commercial or financial relationships that could be construed as a potential conflict of interest.

Author Contributions

D.P.: conceptualization, formal analysis, investigation, methodology, validation, visualization, writing—original draft, writing—review and editing. **A.B.**: investigation, software, methodology, visualization, writing—original draft, writing—review and editing. **A.D.S.**: methodology, validation, writing—review and editing. **T.A.H.:** methodology, validation, writing—review and editing. **G.C.:** funding acquisition, supervision, writing—review and editing. **P.L.:** methodology, supervision, writing—review and editing.

Funding

Details of all funding sources should be provided, including grant numbers if applicable. Please ensure to add all necessary funding information, as after publication this is no longer possible.

Acknowledgments

The authors would like to thank the native speaker colleagues who helped translating the scenarios.

Data Availability Statement

The code for this study can be found on Github at `https://github.com/aira-list/FAIRGAME`, together with examples of configuration files and templates.

References

[1] Steve Morgan. Cybercrime to cost the world $10.5 trillion annually by 2025. *Cybercrime Magazine*, 2020. URL `https://cybersecurityventures.com/hackerpocalypse-cybercrime-report-2016/`.

[2] Ani Petrosyan. Estimated cost of cybercrime worldwide 2018-2029, 2024. URL `https://www.statista.com/forecasts/1280009/cost-cybercrime-worldwide`.

[3] Kjell Hausken, Jonathan W Welburn, and Jun Zhuang. A review of attacker–defender games and cyber security. *Games*, 15(4):28, 2024.

[4] Cuong T Do, Nguyen H Tran, Choongseon Hong, Charles A Kamhoua, Kevin A Kwiat, Erik Blasch, Shaolei Ren, Niki Pissinou, and Sundaraja Sitharama Iyengar. Game theory for cyber security and privacy. *ACM Computing Surveys (CSUR)*, 50(2):1–37, 2017.

[5] Sajjan Shiva, Sankardas Roy, and Dipankar Dasgupta. Game theory for cyber security. In *Proceedings of the Sixth Annual Workshop on Cyber Security and Information Intelligence Research*, pages 1–4, 2010.

[6] Yuan Wang, Yongjun Wang, Jing Liu, Zhijian Huang, and Peidai Xie. A survey of game theoretic methods for cyber security. In *2016 IEEE First International Conference on Data Science in Cyberspace (DSC)*, pages 631–636. IEEE, 2016.

[7] Adeela Bashir, Zia Ush Shamszaman, Zhao Song, and The Anh Han. Co-evolutionary dynamics of attack and defence in cybersecurity. *arXiv preprint arXiv:2505.19338*, 2025.

[8] Lewis Hammond, Alan Chan, Jesse Clifton, Jason Hoelscher-Obermaier, Akbir Khan, Euan McLean, Chandler Smith, Wolfram Barfuss, Jakob Foerster, Tomáš Gavenčiak, et al. Multi-Agent Risks from Advanced AI. *arXiv preprint arXiv:2502.14143*, 2025.

[9] Guillermo Owen. *Game theory*. Emerald Group Publishing, 2013.

[10] Saurabh Amin and Karl Henrik Johansson. Preface to the focused issue on dynamic games in cyber security. *Dynamic Games and Applications*, 9:881–883, 2019.

[11] Kjell Hausken. Cyber resilience in firms, organizations and societies. *Internet of Things*, 11: 100204, 2020.

[12] S Rasoul Etesami and Tamer Başar. Dynamic games in cyber-physical security: An overview. *Dynamic Games and Applications*, 9(4):884–913, 2019.

[13] Yikang Lu, Alberto Aleta, Chunpeng Du, Lei Shi, and Yamir Moreno. Llms and generative agent-based models for complex systems research. *Phys. Life Rev.*, 2024.

[14] Michael Henry Tessler, Michiel A Bakker, Daniel Jarrett, Hannah Sheahan, Martin J Chadwick, et al. Ai can help humans find common ground in democratic deliberation. *Science*, 386(6719):eadq2852, 2024.

[15] Nikhil Patel and Sandeep Trivedi. Leveraging predictive modeling, machine learning personalization, nlp customer support, and ai chatbots to increase customer loyalty. *Empir. Quests Manage. Essenc.*, 3(3):1–24, 2020.

[16] Muhammad Mudassar Yamin, Ehtesham Hashmi, Mohib Ullah, and Basel Katt. Applications of llms for generating cyber security exercise scenarios. *IEEE Access*, 2024.

[17] Wafaa Kasri, Yassine Himeur, Hamzah Ali Alkhazaleh, Saed Tarapiah, Shadi Atalla, Wathiq Mansoor, and Hussain Al-Ahmad. From vulnerability to defense: The role of large language models in enhancing cybersecurity. *Computation*, 13(2):30, 2025.

[18] Mohamed Amine Ferrag, Fatima Alwahedi, Ammar Battah, Bilel Cherif, Abdechakour Mechri, and Norbert Tihanyi. Generative ai and large language models for cyber security: All insights you need. *Available at SSRN 4853709*, 2024.

[19] Jie Zhang, Haoyu Bu, Hui Wen, Yongji Liu, Haiqiang Fei, Rongrong Xi, Lun Li, Yun Yang, Hongsong Zhu, and Dan Meng. When llms meet cybersecurity: A systematic literature review. *Cybersecurity*, 8(1):1–41, 2025.

[20] Alekh Avinash and Kurunandan Jain. Evolving strategies: Llms as game players. In *2025 4th International Conference on Sentiment Analysis and Deep Learning (ICSADL)*, pages 1009–1014. IEEE, 2025.

[21] Long He, Geng Sun, Dusit Niyato, Hongyang Du, Fang Mei, Jiawen Kang, Mérouane Debbah, and Zhu Han. Generative ai for game theory-based mobile networking. *IEEE Wireless Communications*, 32(1):122–130, 2025.

[22] Zainab Alalawi, Paolo Bova, Theodor Cimpeanu, Alessandro Di Stefano, Manh Hong Duong, Elias Fernández Domingos, The Anh Han, Marcus Krellner, Ndidi Bianca Ogbo, Simon T Powers, et al. Trust ai regulation? discerning users are vital to build trust and effective ai regulation. *Applied Mathematics and Computation*, 508:129627, 2026.

[23] Nataliya Balabanova, Adeela Bashir, Paolo Bova, Alessio Buscemi, Theodor Cimpeanu, Henrique Correia da Fonseca, et al. Media and responsible ai governance: a game-theoretic and llm analysis. *arXiv:2503.09858*, 2025.

[24] Alessio Buscemi, Daniele Proverbio, Paolo Bova, Nataliya Balabanova, Adeela Bashir, Theodor Cimpeanu, et al. Do LLMs trust AI regulation? Emerging behaviour of game-theoretic LLM agents. *arXiv:2504.08640*, 2025.

[25] Nicoló Fontana, Francesco Pierri, and Luca Maria Aiello. Nicer than humans: How do large language models behave in the prisoner's dilemma? *arXiv:2406.13605*, 2024.

[26] Zhen Wang, Ruiqi Song, Chen Shen, Shiya Yin, Zhao Song, Balaraju Battu, Lei Shi, Danyang Jia, Talal Rahwan, and Shuyue Hu. Large language models overcome the machine penalty when acting fairly but not when acting selfishly or altruistically. *arXiv:2410.03724*, 2024.

[27] Elif Akata, Lion Schulz, Julian Coda-Forno, Seong Joon Oh, Matthias Bethge, and Eric Schulz. Playing repeated games with large language models. *Nature Human Behaviour*, 2025. doi: https://doi.org/10.1038/s41562-025-02172-y.

[28] Munnujahan Ara, Hugo Reboredo, Samah AM Ghanem, and Miguel RD Rodrigues. A zero-sum power allocation game in the parallel gaussian wiretap channel with an unfriendly jammer. In *2012 IEEE International Conference on Communication Systems (ICCS)*, pages 60–64. IEEE, 2012.

[29] Charles A Kamhoua, Manuel Rodriguez, and Kevin A Kwiat. Testing for hardware trojans: A game-theoretic approach. In *International Conference on Decision and Game Theory for Security*, pages 360–369. Springer, 2014.

[30] Charles A Kamhoua, Niki Pissinou, and S Kami Makki. Game theoretic analysis of cooperation in autonomous multi hop networks: The consequences of unequal traffic load. In *2010 IEEE Globecom Workshops*, pages 1973–1978. IEEE, 2010.

[31] Nadiya Kostyuk. The digital prisoner's dilemma: Challenges and opportunities for cooperation. *2013 World Cyberspace Cooperation Summit IV (WCC4)*, pages 1–6, 2013.

[32] Jordan Richard Schoenherr and Robert Thomson. Beyond the prisoner's dilemma: The social dilemmas of cybersecurity. In *2020 international conference on cyber situational awareness, data analytics and assessment (CyberSA)*, pages 1–7. IEEE, 2020.

[33] Alessio Buscemi, Daniele Proverbio, Alessandro Di Stefano, The Anh Han, German Castignani, and Pietro Di Liò. Fairgame: a framework for ai agents bias recognition using game theory. *ECAI 2025, in press, arXiv preprint arXiv:2504.14325*, 2025.

[34] Julián García and Matthijs Van Veelen. No strategy can win in the repeated prisoner's dilemma: linking game theory and computer simulations. *Frontiers in Robotics and AI*, 5: 102, 2018.

[35] The Anh Han, Luis Moniz Pereira, Francisco C Santos, Tom Lenaerts, et al. To regulate or not: a social dynamics analysis of an idealised ai race. *Journal of Artificial Intelligence Research*, 69:881–921, 2020.

[36] Vikram Srinivasan, Pavan Nuggehalli, Carla-Fabiana Chiasserini, and Ramesh R Rao. Cooperation in wireless ad hoc networks. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)*, volume 2, pages 808–817. IEEE, 2003.

[37] Pratishtha Shukla, Lu An, Aranya Chakrabortty, and Alexandra Duel-Hallen. A robust stackelberg game for cyber-security investment in networked control systems. *IEEE Transactions on Control Systems Technology*, 31(2):856–871, 2022.

[38] Anh Tung Nguyen, Sribalaji C Anand, and André MH Teixeira. A zero-sum game framework for optimal sensor placement in uncertain networked control systems under cyber-attacks. In *2022 IEEE 61st Conference on Decision and Control (CDC)*, pages 6126–6133. IEEE, 2022.

[39] Quanyan Zhu and Tamer Başar. Robust and resilient control design for cyber-physical systems with an application to power systems. In *2011 50th IEEE Conference on Decision and Control and European Control Conference*, pages 4066–4071. IEEE, 2011.

[40] Yaoqi Yang, Hongyang Du, Geng Sun, Zehui Xiong, Dusit Niyato, and Zhu Han. Exploring equilibrium strategies in network games with generative ai. *IEEE Network*, 2024.

[41] Yong Xiao, Guangming Shi, and Ping Zhang. Towards agentic ai networking in 6g: A generative foundation model-as-agent approach. *arXiv preprint arXiv:2503.15764*, 2025.

[42] Theodoros Spyridopoulos, George Karanikas, Theodore Tryfonas, and Georgios Oikonomou. A game theoretic defence framework against dos/ddos cyber attacks. *Computers & Security*, 38:39–50, 2013.

[43] Gerald Brown, Jeff Kline, Adam Thomas, Alan Washburn, and Kevin Wood. A game-theoretic model for defense of an oceanic bastion against submarines. *Military Operations Research*, pages 25–40, 2011.

[44] John Von Neumann and Oskar Morgenstern. Theory of games and economic behavior: 60th anniversary commemorative edition. In *Theory of games and economic behavior*. Princeton university press, 2007.

[45] Kjell Hausken. Probabilistic risk analysis and game theory. *Risk Analysis*, 22(1):17–27, 2002.

[46] Zhen Wang, Satoshi Kokubo, Marko Jusup, and Jun Tanimoto. Universal scaling for the dilemma strength in evolutionary games. *Phys. Life Rev.*, 14:1–30, 2015.

[47] Zhu Ji, Wei Yu, and KJ Ray Liu. A belief evaluation framework in autonomous manets under noisy and imperfect observation: Vulnerability analysis and cooperation enforcement. *IEEE Transactions on Mobile Computing*, 9(9):1242–1254, 2010.

[48] Michael P Wellman, Karl Tuyls, and Amy Greenwald. Empirical game theoretic analysis: A survey. *Journal of Artificial Intelligence Research*, 82:1017–1076, 2025.

[49] Chelse Swoopes, Tyler Holloway, and Elena L Glassman. The impact of revealing large language model stochasticity on trust, reliability, and anthropomorphization. *arXiv preprint arXiv:2503.16114*, 2025.

[50] Alessio Buscemi and Daniele Proverbio. Large language models' detection of political orientation in newspapers. *arXiv preprint arXiv:2406.00018*, 2024.

[51] Shanshan Han, Qifan Zhang, Yuhang Yao, Weizhao Jin, Zhaozhuo Xu, and Chaoyang He. Llm multi-agent systems: Challenges and open problems. *arXiv preprint arXiv:2402.03578*, 2024.

[52] Weixin Liang, Mert Yuksekgonul, Yining Mao, Eric Wu, and James Zou. Gpt detectors are biased against non-native english writers. *Patterns*, 4(7), 2023.

[53] Kjell Hausken. Fifty years of operations research in defense. *European Journal of Operational Research*, 318(2):355–368, 2024.

[54] Ali Pala and Jun Zhuang. Information sharing in cybersecurity: A review. *Decision Analysis*, 16(3):172–196, 2019.

[55] Yuling Liu, Dengguo Feng, Yifeng Lian, Kai Chen, and Yingjun Zhang. Optimal defense strategies for ddos defender using bayesian game model. In *Information Security Practice and Experience: 9th International Conference, ISPEC 2013, Lanzhou, China, May 12-14, 2013. Proceedings 9*, pages 44–59. Springer, 2013.

[56] Alessandro Acquisti and Jens Grossklags. Privacy and rationality in individual decision making. *IEEE security & privacy*, 3(1):26–33, 2005.

[57] Richard D McKelvey and Thomas R Palfrey. Quantal response equilibria for extensive form games. *Experimental economics*, 1:9–41, 1998.

Figure captions

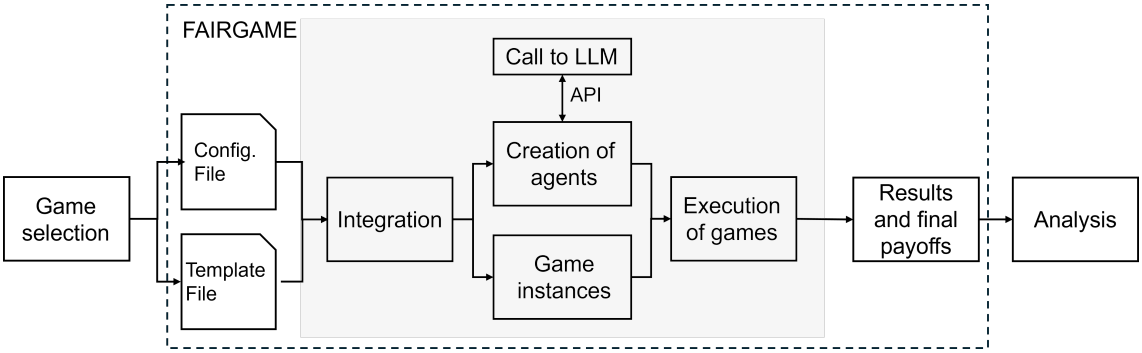

Figure 1: Simulation and analysis workflow. After selecting the games, they are instantiated in LLM form using FAIRGAME (whose pipeline is in dashed frame; figure adopted from [50]): the Config. and Template file are user-defined to specify the game settings and features and are taken as inputs; then, the framework automatically integrates the information and runs the games by calling the desired LLMs (grey-shaded area); the output are the rounds history, the final payoffs and any other specified metric, which is finally analysed.

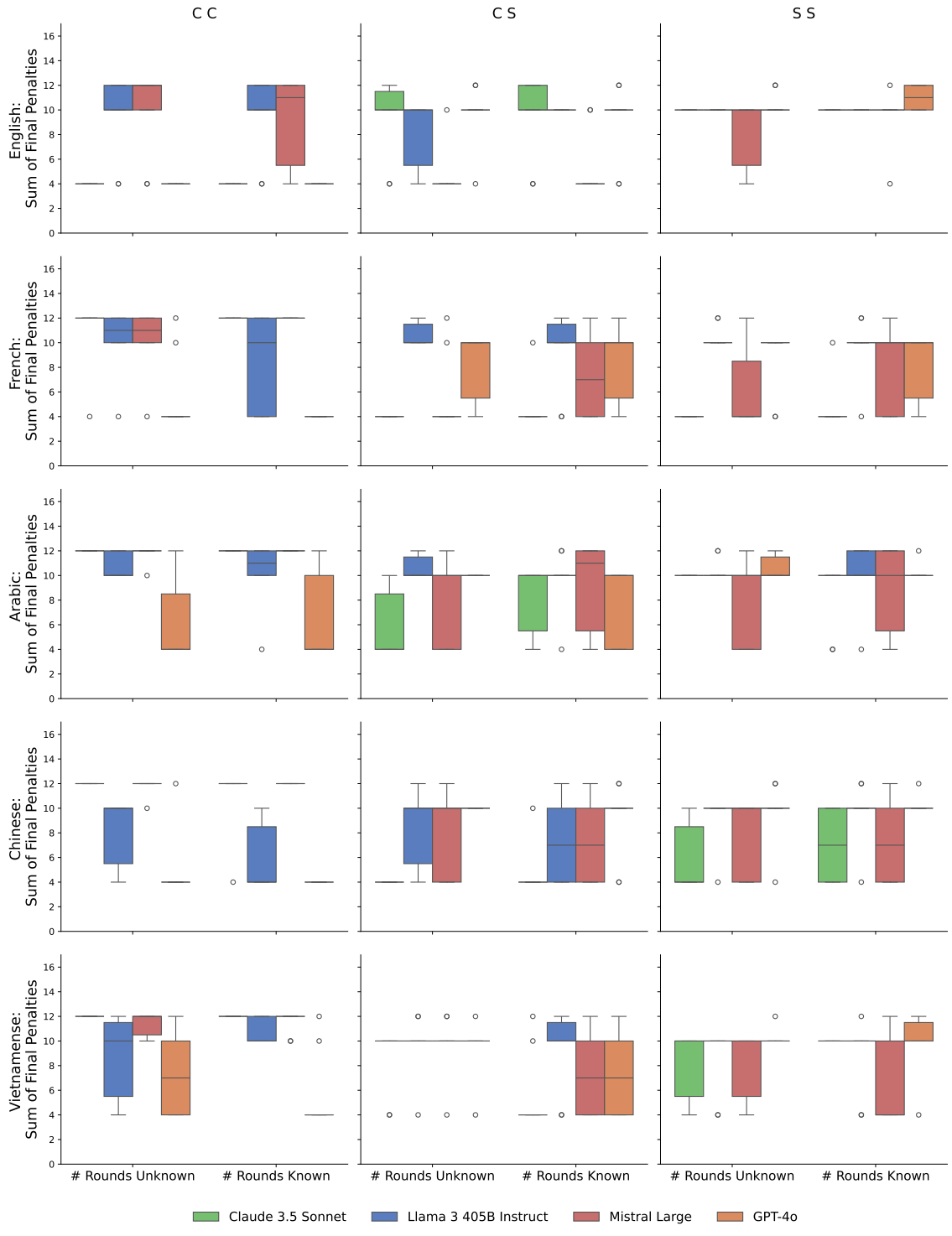

Figure 2: Aggregated final payoffs of the repeated Prisoner's Dilemma games over repeated experiments, for each LLM (see legend for colour-coding), combination of personalities (columns), language (rows), and knowledge of opponent's personality (x-axis).

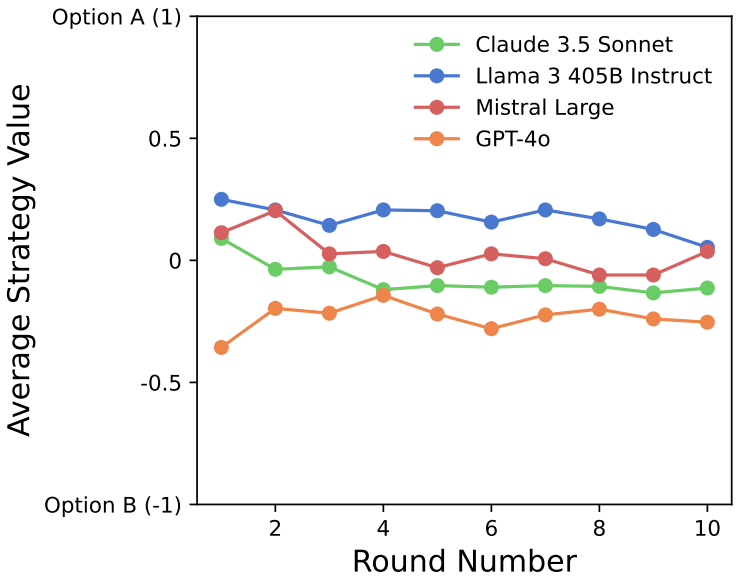

Figure 3: Evolution of normalized penalties (averaged over repeated experiments) over repeated rounds, for each LLM within the Prisoner' Dilemma scenario.

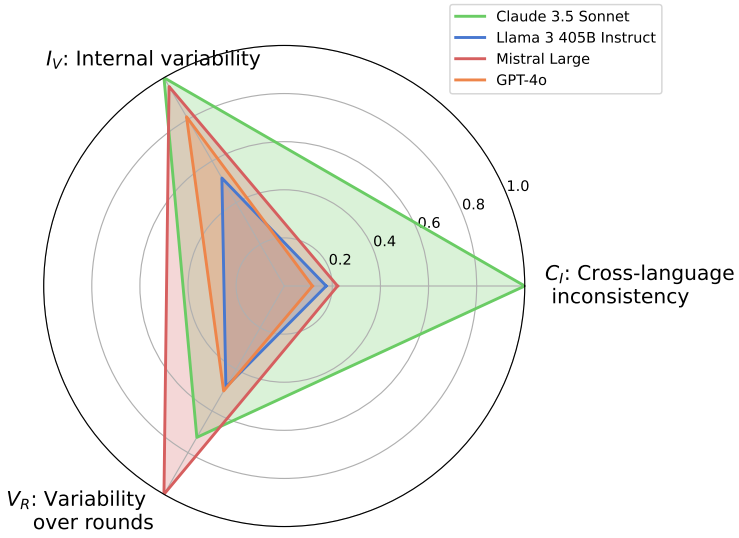

Figure 4: Radar plot mapping the three metrics described in Sec. 2.2.4, for Prisoner's Dilemma and for all considered LLMs.

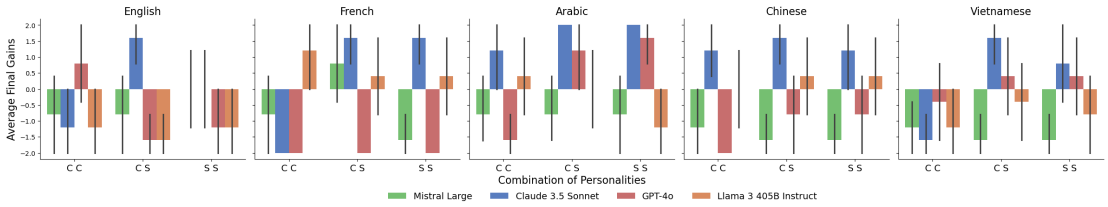

Figure 5: Final payoffs of agent 1 in the one-shot zero-sum game, for each LLM (see legend for colour-coding), combination of personalities (columns) and language (rows).