Multicultural Spyfall: Assessing LLMs through Dynamic Multilingual Social Deduction Game

Anonymous ACL submission

Abstract

The rapid advancement of Large Language Models (LLMs) has necessitated more robust evaluation methods that go beyond static benchmarks, which are increasingly prone to data saturation and leakage. In this paper, we propose a dynamic benchmarking framework for evaluating multilingual and multicultural capabilities through the social deduction game Spyfall. In our setup, models must engage in strategic dialogue to either identify a secret agent or avoid detection, utilizing culturally relevant locations or local foods. Our results show that our game-based rankings are consistent with Chatbot Arena rankings on the overlapping models. However, we find a significant performance gap in non-English contexts: models are generally less proficient when handling locally specific entities and often struggle with rule-following or strategic integrity in non-English languages. We demonstrate that this game-based approach provides a scalable, leakage-resistant, and culturally nuanced alternative to traditional NLP benchmarks.

1 Introduction

The rapid advancement of Large Language Models (LLMs) and their expanding multilingual capabilities have made robust evaluation increasingly critical. While numerous multilingual benchmarks have been developed (Wu et al., 2025; Hu et al., 2020), many are static, making them susceptible to data saturation and potential "leakage" into training sets over time. To address these limitations, researchers have explored dynamic benchmarking, utilizing text-based games (Hu et al.; Song et al., 2025; Ma et al., 2025; Kim et al., 2025), human preference evaluations (Chiang et al., 2024; Kim et al., 2025), or debates (Moniri et al., 2025). However, a significant gap remains in combining these dynamic approaches to specifically evaluate multilingual and multicultural nuances.

Figure 1: In this example, the Spy fails to identify the target entity, Jam Gadang. Due to insufficient context and careful questioning from the other players, the Spy guesses Monas (a landmark in a different part of Indonesia) and loses the game.

To bridge this gap, we propose a dynamic benchmarking framework for multilingual and multicultural understanding through strategic gameplay. Specifically, we adapt the social deduction game Spyfall[1]. In a standard game of Spyfall, all players except one (the "Spy") are given a specific location; the Spy's goal is to deduce that location through conversation, while the other players attempt to identify the Spy by asking subtle, context-specific questions.

Our framework utilizes this concept by playing the game in diverse languages and replacing generic locations with culturally relevant settings. This requires models to not only understand the language but also possess a deep cultural understanding, in our case, location and food as a proxy for culture, following (Adilazuarda et al., 2024).

Our findings indicate that our benchmark rankings follow Chatbot Arena's ordering of the models

[1] https://hobbyworldint.com/portfolio-item/spyfall/

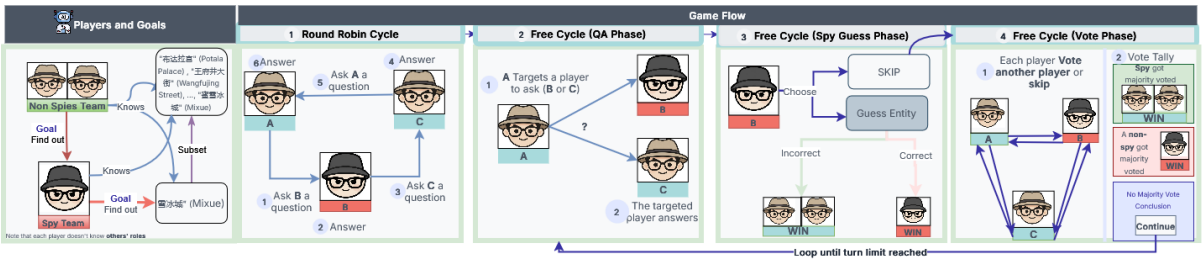

Figure 2: The flow of the game of Cultured Turn-Based Spyfall

we evaluate, though our rating spread is narrower. Furthermore, our analysis reveals that model proficiency significantly declines in non-English contexts and when handling locally specific cultural entities. We observed that while top-tier models remain competitive, weaker models often fail to follow game logic or inadvertently leak the secret location to the Spy when operating in non-English languages. We also find that guessing food is more challenging than guessing location.

By utilizing social deduction games to evaluate multilingual and multicultural capabilities, our benchmark addresses several flaws in current evaluation methodologies. First, it is inherently resistant to saturation; because models compete against one another in a dynamic environment, the 'performance ceiling' evolves as models improve. Second, it is robust against data leakage, as the interactive nature of the game cannot be captured by static training sets. Finally, this framework is highly scalable and extensible; localized versions for new languages or cultures can be implemented simply by updating the underlying entity pools and cultural settings, thus eliminating the need for the labor-intensive manual annotation typically required for new benchmarks.

In summary, our contributions are as follows:

- We introduce a dynamic benchmarking framework by extending the social deduction game Spyfall to evaluate the multilingual and multicultural reasoning capabilities of LLMs.
- We demonstrate that our game-based rankings consistent with the LMSYS Chatbot Arena on the six models that we evaluate, while showing capability differences not visible in static benchmark.
- We conduct a comprehensive analysis of language proficiency, cultural knowledge, and strategic reasoning, identifying performance degradation in non-English settings and locally specific cultural contexts.

2 Cultural Turn-Based Spyfall

In this paper, we adapt the board game Spyfall into a turn-based format to accommodate LLM benchmarking. Moreover, we expand the game to support multiple languages and extend the set of entities from generic locations to locally specific locations and foods.

2.1 Game Mechanics

The game is played by 3 to 8 players and features two primary roles: **one spy** and the remaining players as **non-spies**. Each player is given a list of 30 possible entities. At the start of the game, non-spy players are informed of one of the 30 entities, while the spy does not receive any location information. No player knows the roles of the others. The non-spies aim to identify the spy, whereas the spy attempts to deduce the entity without being detected.

The gameplay proceeds through three phases: a question-and-answer phase ($\mathcal{P}_{\text{QA}}$), in which players interrogate one another with questions relevant to the given location; a voting phase ($\mathcal{P}_{\text{V}}$), where any player can accuse others (including the spy voting against non-spies); and a spy-guessing phase ($\mathcal{P}_{\text{SG}}$), in which the spy attempts to identify the revealed location. The game's design subtly engages all participants. Specifically, the spy must avoid exposing themselves during questioning, while non-spies must respond without revealing obvious hints about the location and simultaneously deduce who the spy is.

Our version differs from the original game. In the original one, gameplay occurs in real time, allowing players to initiate votes or accusations at any moment within a time limit. Since we use LLMs as players, real-time play would disadvantage slower models and exacerbate latency issues. We therefore adapt the game to a turn-based for-

mat. As a result, the time limit is replaced with a turn limit, which we set to 2 × the number of players in the game. The overall flow of the game is shown in Figure 2, and an example of the running game is presented in Figure 1.

2.2 Game Entities $\mathcal{E}$

In contrast to the original game, we do not limit guessing to places but also include food names as objects to be guessed. We therefore refer to the list of guessable objects as entities, denoted by $\mathcal{E}$. Each game selects one entity from 30 choices. All 30 entities are displayed to each player, with one target entity, $\mathcal{E}_t$, that is not known to the spy.

2.3 Game Phases $\mathcal{P}$

We design the game flow to be turn-based, where each turn consists of and begins with the following phases: $\mathcal{P}_{QA}$, followed by $\mathcal{P}_{SG}$, and finally $\mathcal{P}_V$. We place $\mathcal{P}_{SG}$ before $\mathcal{P}_V$ because we prioritize observing the spy ' s ability to deduce knowledge, particularly local knowledge.

Question-and-Answer Phase $\mathcal{P}_{QA}$ In this phase, the current player asks another player a question, and the targeted player must answer it. Only one question may be asked per turn. After the turn ends, the answerer becomes the questioner in the next turn.

Spy Guess Phase $\mathcal{P}_{SG}$ During this phase, the spy may choose to either guess the target entity $\mathcal{E}_t$ or skip the phase. If the spy chooses to guess $\mathcal{E}_t$ and the guess is correct, the spy wins the game; otherwise, the non-spies win.

Voting Phase $\mathcal{P}_V$ In the voting phase, each player votes for another player, accusing them of being the spy. Players may vote for a specific player or choose to skip voting. If a player receives more than half of the votes (e.g., 3 out of 5 players), the game ends, and the accused player's role is revealed. If the accused player is the spy, the non-spies win; otherwise, the spy wins.

The end game conditions are presented in Appendix A

2.4 Game Cycles

In $\mathcal{P}_{QA}$, following the original game rules, the questioner selects a target player, and the target becomes the questioner in the next turn. This allows the same pair of players to target each other in subsequent turns. As a result, other players may have no actions during this phase.

To address this issue, we introduce two cycles that structure the turns described above: the Round Robin Cycle and the Free Cycle. In the Round Robin Cycle, during $\mathcal{P}_{QA}$, each player must target the next player in a fixed order, ensuring that every player participates at least once as both a questioner and an answerer. During this cycle, the game skips $\mathcal{P}_{SG}$ and $\mathcal{P}_V$, as the primary objective is to ensure that all players actively participate in the game.

After all players complete $\mathcal{P}_{QA}$ in the Round Robin Cycle, the game proceeds to the Free Cycle. In this cycle, the default rules apply, including $\mathcal{P}_{QA}$, $\mathcal{P}_{SG}$, and $\mathcal{P}_V$.

2.5 Game Players

In this setup, both the spy and non-spy roles are played by Large Language Models (LLMs). To illustrate, consider a five-player game consisting of four non-spies and one spy. If the non-spies are assigned to Model A and the spy to Model B, each non-spy instance is treated independently despite sharing the same underlying model. We evaluate models across all possible role permutations to ensure a comprehensive all-to-all comparison. Instruction and prompt that each model receives can be seen in Appendix C.

3 Experiment Setup

Game Configuration The games are played with five players, four of whom are non-spies, each played by an LLM independently, and the spy is played by another LLM. The turn limit is five turns in the Round-robin Cycle and five turns in the Free Cycle, totaling 10 turns. The players' order is shuffled to remove positional bias in the analysis. Each game is played once.

Models We use six models: `gemini-2.5-pro` (Comanici et al., 2025), `gemini-2.5-flash` (Comanici et al., 2025), `qwen3-30b-a3b` (Yang et al., 2025), `gemma-3-12b-it` (Team et al., 2025), `qwen3-8b` (Yang et al., 2025), and `llama-3.1-8b-instruct` (Grattafiori et al., 2024)[2]. We choose LLMs that have multilingual capabilities and also based on a variety of capabilities, from strong to weak LLMs, so that the

[2]In subsequent sections, we may shorten their names accordingly due to space constraints.

Model	Overall	Generic Location				Local Location			Local Food		
		GEN	GID	GARZ	GZH	LID	LARZ	LZH	FID	FARZ	FZH
1 Gemini-P	**1136**$_{\pm 11}$	**1086**$_{\pm 33}$	**1116**$_{\pm 35}$	**1163**$_{\pm 40}$	**1174**$_{\pm 39}$	**1154**$_{\pm 38}$	**1171**$_{\pm 41}$	**1161**$_{\pm 42}$	1075$_{\pm 36}$	**1157**$_{\pm 37}$	1136$_{\pm 38}$
2 Gemini-F	1107$_{\pm 11}$	1057$_{\pm 34}$	1071$_{\pm 34}$	1130$_{\pm 38}$	1103$_{\pm 39}$	1086$_{\pm 34}$	1161$_{\pm 36}$	1139$_{\pm 39}$	**1098**$_{\pm 35}$	1112$_{\pm 38}$	**1150**$_{\pm 38}$
3 Qwen30B-T	1016$_{\pm 12}$	1045$_{\pm 33}$	982$_{\pm 35}$	1032$_{\pm 37}$	1030$_{\pm 38}$	996$_{\pm 35}$	1071$_{\pm 36}$	985$_{\pm 38}$	1002$_{\pm 36}$	1032$_{\pm 37}$	1005$_{\pm 34}$
4 Gemma12B	1003$_{\pm 11}$	995$_{\pm 32}$	1031$_{\pm 34}$	1050$_{\pm 35}$	997$_{\pm 36}$	975$_{\pm 34}$	1000$_{\pm 38}$	994$_{\pm 36}$	1023$_{\pm 36}$	1001$_{\pm 36}$	981$_{\pm 36}$
5 Qwen8B	967$_{\pm 11}$	948$_{\pm 35}$	954$_{\pm 33}$	964$_{\pm 37}$	970$_{\pm 34}$	939$_{\pm 33}$	959$_{\pm 39}$	959$_{\pm 35}$	994$_{\pm 33}$	1034$_{\pm 37}$	962$_{\pm 35}$
6 Llama8B	771$_{\pm 13}$	869$_{\pm 37}$	846$_{\pm 39}$	660$_{\pm 56}$	727$_{\pm 44}$	850$_{\pm 37}$	638$_{\pm 55}$	762$_{\pm 42}$	807$_{\pm 41}$	664$_{\pm 57}$	766$_{\pm 43}$

Table 1: Bradley-Terry ratings across scenarios with 95% bootstrap confidence interval with 1000 resamples. **Gold**, Silver, Bronze, and 4th indicate top 4 per column.

ranking leaderboard's pattern can be analyzed. We use `openrouter.ai` to run the inferences.

Scenarios and Languages We define three scenarios: Generic (G), where $\mathcal{E}$ consists of the original generic places from the original game, most of which are also used in (Kim et al., 2025), using the English language and translated to local languages; Local Locations (L), where $\mathcal{E}$ consists of local locations residing in a region that uses the target language; and Local Food (F), where $\mathcal{E}$ consists of food originating from the respective region that uses the target language. We focus on Simplified Chinese (`zh`), Egyptian Arabic (`arz`), and Indonesian (`id`) with their respective countries: China, Egypt, and Indonesia. In the game, each player is shown 30 entities. The list of entities for each scenario can be seen in Appendix B.

Evaluation Metrics To compute the rank of each model, we use the Bradley-Terry Model (Bradley and Terry, 1952), following Chatbot Arena's metric (Chiang et al., 2024), due to its stability in computing the ratings. Additionally, we compare the win rate of each model, where the win rate of a model M is $\frac{\text{\#Wins}}{\text{\#Games Played}_M} \times 100$.

We also calculate the **leakage rate**, which we define as the percentage of games in which a non-spy player reveals the target entity $\mathcal{E}_t$ during $\mathcal{P}_{QA}$. A leakage is considered to have occurred if any non-spy player provides information that explicitly discloses the identity of $\mathcal{E}_t$ to other players, including the spy. The leakage rate of a model M is computed as Leakage Rate$_M = \frac{\text{\#Games with Leakage}_M}{\text{\#Non-spy Games}_M} \times 100\%$. In analyzing spy behavior, non-spy behavior, and entity analysis, we use Shannon entropy (Shannon, 1948) to calculate how spread out the votes or entity guesses from the players are.

4 Overall Game Analysis

We compare three different settings: overall performance across all languages and scenarios, and the different roles that the model played, with a total of 9,000 matches[3].

TB-Spyfall rankings are consistent with Chatbot Arena on overlapping models We compare the ranking between our benchmark and Chatbot Arena, accessed on January 1st, 2026. On the five overlapping models[4], our ranking matches Chatbot Arena's ordering. Specifically, the rating spread differs, the gap between Gemini Pro and Gemma 12B is 106 points in our benchmark (1136 vs 1030) compared to 62 points in Chatbot Arena (1402 vs 1340), and the gap between top and bottom models is much larger in ours. We attribute these differences to several factors. Firstly, our setup samples fewer matches per model (600 vs 82,000 for Geminin 2.5 Pro in the Arena). Additionally, Spyfall tests strategic concealment, theory-of-mind reasoning over multi-turn interaction, and culturally specific knowledge, which even the best-performing models struggle with.

Non-English languages reveal wider performance gaps between models and cause noticeable degradation in weaker models Table 1 shows that the ratings of G_{ID}, G_{ZH}, and G_{EG} have wider gaps compared to G_{EN}. In G_{EN}, the second-best model, Gemini Flash, and Qwen30B-Thinking have a close gap (12%), which widens in other scenarios. Additionally, the third and fourth ranks fluctuate between Qwen30B-Thinking and Gemma12B. On the other hand, Qwen3-8B and Llama-3.1-8B are consistently in fifth and sixth place with a significant gap, with `arz` being the worst-performing language for Llama-3.1-8B.

[3]Further Statistical robustness observation can be seen in Appendix D

[4]We omit Qwen3-8B as it is missing from the leaderboard.

Local and food scenarios affect performance rankings differently As shown in Table 1, the rankings in location and food scenarios differ. For instance, in most local location scenarios, Gemini Pro consistently outperforms Gemini Flash, while in food scenarios, specifically F_{ID} and F_{EG}, Gemini Flash outperforms Gemini Pro. Additionally, similar to generic locations, the third and fourth places fluctuate between Qwen30B-Thinking and Gemma12B in both local location and food scenarios. Finally, Llama-3.1-8B consistently ranks lowest in all scenarios, with the gap widening in Egyptian Arabic scenarios.

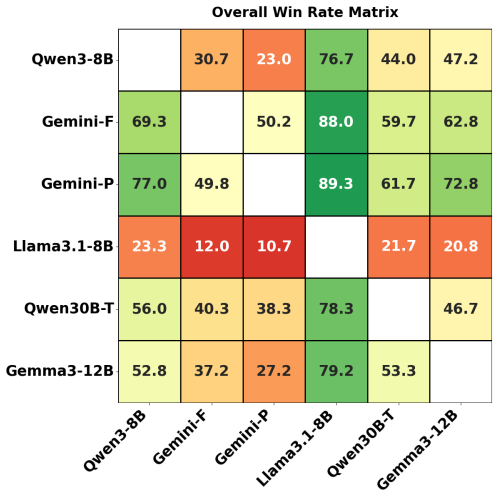

Figure 3: Win Rate Comparisons across Models and Scenarios. Each cell corresponds to win rate of Model in the y-axis againts Model in the x-axis across 600 matches.

Some models perform better against specific opponents From the match matrices in Figure 3, all models have a win rate of more than 75% against Llama-3.1-8B, demonstrating an easy matchup that explains the low rating of Llama-3.1-8B across all scenarios. Gemini Pro and Gemini Flash have a close matchup (approximately 50% win rate) against each other, indicating a competitive pairing. Despite this closeness, when models other than Gemini Flash face Gemini Pro, their win rates are lower than when facing Gemini Flash, demonstrating Gemini Pro's stronger performance against other models compared to Gemini Flash. Interestingly, Qwen30B-Thinking has less than a 47% win rate against Gemma3-12B despite having a higher rating, indicating an unfavorable matchup for Qwen30B-Thinking against Gemma3-12B. However, Gemma3-12B has a lower win rate against Gemini Pro and Gemini Flash compared to Qwen30B-Thinking.

Model	en	id	zh	arz
Gemini-F	0.0	0.0	0.0	0.0
Gemini-P	0.0	0.0	0.0	0.0
Gemma3-12B	0.0	1.6	4.9	1.6
Llama3.1-8B	17.3	37.1	29.6	13.6
Qwen30B-T	0.7	0.4	0.0	0.0
Qwen3-8B	0.0	1.8	8.7	2.4

Table 2: Non-Spy Information Leakage Rates (%) by Language.

5 What Skills Can TB-Spyfall Show?

Table 1 reports ratings computed from win and loss outcomes, which capture relative performance between models. If a scenario makes all models perform worse by a similar amount, their pairwise win rates against each other barely change, and the rating gaps remain closely stable across scenarios and languages (See Appendix E for detailed observation). Table 1 is therefore useful for comparing models overall, but it does not reveal the skills that the models have. To succeed in this game, we argue that a model at least needs these capabilities: language proficiency (producing fluent, rule-compliant output in the target language), cultural knowledge (recognizing and reasoning about region-specific entities), and strategic reasoning (modeling other players' beliefs to conceal or detect roles). In this section, we observe the models by categorizing the analysis along these three axes to elucidate each model's skills.

5.1 Language Proficiency

Llama3.1-8B Has High Leakage Rates Up to 37%, While Gemini Models Have None In this experiment, despite being informed about the game rules, some models leak the $\mathcal{E}_t$ directly, which may allow the spy to guess it easily. Table 2 shows the leakage rate across models and languages. Llama3.1-8B has the highest leakage rate compared to other models, particularly in `id`, reaching 37%. In contrast, Gemini Flash and Gemini Pro have 0% leakage across all languages. Interestingly, Qwen3-8B has a moderate leakage rate in `zh` (~9%), despite being heavily trained on `zh` data. This suggests that language capability does not directly correlate with the leakage rate.

Models Follow Target Languages but Struggle With a Dialect Table 3 shows the language

Model	id	zh	arz
Gemini-F	id 98.5 ms 1.0	zh 99.9 wuu 0.1	arz 70.89 apd 10.45
Gemini-P	id 98.7 ms 1.2	zh 99.9 ja 0.1	arz 80.47 apd 7.56
Gemma3	id 98.8 ms 0.9	zh 99.5 en 0.2	msa 59.44 acx 19.89
Llama3.1	id 96.6 ms 2.3	zh 89.2 en 8.0	msa 29.76 acx 28.99
Qwen-30B	id 98.9 ms 0.7	zh 95.4 en 4.2	msa 49.51 acx 25.93
Qwen3-8B	id 98.6 ms 0.6	zh 99.9 en 0.1	msa 50.3 acx 34.70

Table 3: Language detection of outputs. Top-1 (first) / Top-2 (second).

used by each model across all scenarios (Setup in Appendix F). Overall, most models are consistent in using the target language, with more than 95% usage (except for the Llama3.1-8B model) with the exception with Arabic dialect. These results show that Qwen-8B, Qwen-30B, Llama-8B, and Gemma-12B predominantly generate Modern Standard Arabic (msa) rather than the target arz, with arz usage below 13% across these models. Even Gemini Pro and Gemini Flash, which show substantially higher arz usage (~70-80%), occasionally produce Sudanese Arabic (apd), a geographically close and linguistically overlapping dialect (Keleg et al., 2025). Supported by our manual check by sampling some of the games, Most models struggle to consistently adhere to arz dialect despite explicit instructions, instead defaulting to msa which follows behavior observed in Robinson et al., 2025.

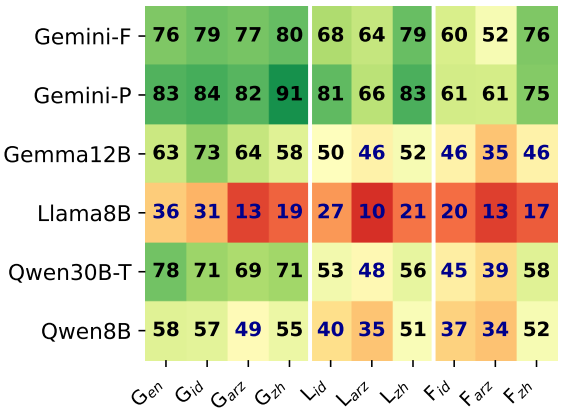

Figure 4: Win rates of Spy (%) across model and scenario.

Some models have an advantage in certain languages where they gain higher spy win rates while others decline in Generic scenario As can be seen in Figure 4, compared to G_{en}, Gemini family models improve their spy win rate by up to 8% in G_{id} and G_{zh}, while Qwen30B-Thinking and Qwen3-8B show moderate drops (up to 7% and 3%, respectively) in spy win rate, followed by a huge drop for Llama3.1-8B in G_{zh}(~17%). Meanwhile, in G_{arz}, the Qwen and Llama models significantly tend to drop the most in spy win rate, showing that arz is more challenging for these models, while others have slight changes.

Qwen3-8B and Qwen30B-Thinking have the highest spy win rates in F and L scenarios in zh language compared to other languages In Figure 4, For F scenarios, both models achieve their highest win rates in zh (Qwen30B-T: 58%, Qwen8B: 52%) compared to id (45%/37%) and arz (39%/34%), with gaps exceeding 18 percentage points, showing that these models have better local knowledge of Chinese food and locations than those of other languages. This is expected, as these models are heavily trained on zh data.

5.2 Cultural Knowledge

Each language scenario differs in difficulty for guessing Egyptian Arabic (arz) consistently proves most difficult for spies to guess correctly across all scenarios. In generic locations, spy guess accuracy is: en (~68%), id (~67.5%), zh (~64.7%), and arz (59.8%). This gap widens dramatically in local contexts, where for local food, accuracy drops to: zh (52.0%), id (41.7%), and arz (24.5%), representing around 2× difference between easiest and hardest. Local locations show a similar pattern: zh (55.7%), id (51.5%), arz (34.3%).

Overall, location and food scenarios in spy role have lower win rates compared to generic location In Figure 4, in most models, the spy win rate in local location and food scenarios drops significantly compared to the generic location scenario, indicating that predicting local knowledge is more challenging for the spy role. Similar to the generic location scenario, the arz language has the lowest spy win rate compared to other languages. Both location and food scenarios have similar spy win rates across models.

Lang	$\mathcal{E}_t$	H	Acc	Guess Distribution (Top-5)
id	Loloh Cemcem	3.21	9%	*Es Lidah Buaya* (4); *Martabak* (4); *Bandrek* (3); *Beras Kencur* (3); ✓*Loloh Cemcem* (2)
arz	ترمس (Turmus)	3.43	5%	حمص الشام (Hummus al-Sham) (3); كحك العيد (Kahk el-Eid) (3); حواوشي (Hawawshi) (2); سحلب (Sahlab) (2); كشري (Koshary) (2)
zh	馕 (Naan)	2.49	29%	烧饼 *(Shaobing)* (5); ✓馕 *(Naan)* (5); 元宵 *(Yuanxiao)* (2); 肉夹馍 *(Roujiamo)* (2); 手抓饼 *(Shouzhuabing)* (1)

H = Entropy (bits); Acc = Accuracy (%); ✓= Correct guess

Table 4: Detailed Guess Distribution for High-Entropy Locations and low accuracy $\mathcal{E}_t$

	G		L		F	
	Acc	H	Acc	H	Acc	H
en	67.3	1.49	–	–	–	–
id	67.1	1.49	49.6	1.81	40.9	2.35
zh	64.2	1.55	54.5	1.50	51.6	2.14
arz	59.5	1.69	33.8	2.10	24.4	2.80

Table 5: Overall Accuracy (Acc) and Entropy (H) by Dataset

Local Scenarios and Food Are More Challenging than Generic Locations We evaluate the difficulty of each scenario by analyzing the spy players' ability to guess $\mathcal{E}_t$, measured through overall accuracy and vote entropy across locations (Table 5). Overall, generic scenarios exhibit higher guess accuracy and lower vote entropy compared to local locations and food scenarios, indicating that generic locations are easier for the spy to deduce. This performance gap likely stems from the fact that generic locations are more globally recognized by models, whereas local entities require specialized regional knowledge. Notably, in both local location and food scenarios, the `arz` language exhibits the lowest overall entity accuracy and the highest vote entropy compared to other languages.

Lower-Accuracy Entities Share Characteristics with Distractors We sampled local food entities with the lowest accuracy to identify common patterns of confusion, as shown in Table 4. In `id`, the entity *Loloh Cemcem* refers to a traditional drink; it was frequently confused with similar beverages such as *Es Lidah Buaya*, *Bandrek*, and *Beras Kencur*. In `zh`, 馕 (Naan) refers to a staple flatbread that shares significant characteristics with 烧饼 (Shaobing), 肉夹馍 (Roujiamo), and 手抓饼 (Shouzhua Bing). In `arz`, ترمس (Turmus, boiled lupin beans) is a staple street snack in Egypt. While it shares characteristics with other snacks, these foods are distinct; however, the models fail to take this into consideration.

Some Errors are due to Model's Lack of Cultural Knowledge As shown in Table 4, some confusion occurs between entirely unrelated entities. For example, the Indonesian traditional healthy drink was quite often confused with *Martabak*, a pancake-like street food. Upon further investigation, we note that some weaker non-spies did not know the drink and answered incorrectly, as if it were a fried food. The spy, thinking it was a dessert, also guessed incorrectly. Stronger models such as Gemini tend to discuss the factual information regarding the drink correctly.

5.3 Strategic Reasoning

Category	Count	%
Spy Guess Wrong	2,917	32.41
Spy Guess Correct	3,257	36.19
Vote Majority to Spy	716	7.96
Vote Majority to Non-Spy	363	4.03
Spy Surrender	458	5.09
Non-Spy Surrender	1,289	14.32

Table 6: Overall Game Outcome Statistics

Game Endings Are Dominated by Spy Guess Actions Table 6 shows the overall game outcome statistics across all models and scenarios. The most frequent game ending is when the spy guesses the location, comprising 36.19% correct guesses and 32.41% wrong guesses, summing to 68.6%. This shows that the spy is more likely to win through guessing rather than being voted out. Additionally, surrender actions are also significant (≈19.41%), dominated by the Llama3.1-8B model (70.52% of all surrenders). Finally, the voting phase only contributes to 12% of all game endings, showing that the models are less likely to vote out the spy.

Why Does a Spy Player Get Voted Out? Through a qualitative analysis of 20 randomly sampled matches where Gemini Pro or Flash were eliminated, we identified three primary reasons for their

Model	Guess ×	Got Voted	Quit
Llama8B	82.4	5.4	12.2
Qwen8B	81.1	7.1	11.8
Qwen30B-T	80.8	6.6	12.6
Gemma12B	78.0	8.8	13.2
Gemini-P	63.6	25.9	10.5
Gemini-F	60.8	29.4	9.7
Average	**74.5**	**13.9**	**11.7**

Table 7: Non-Spy Victory Distribution by Model (when model is non-spy). Column denotes spy last action. Percentages show the win rate win methods of Non-spies

failure despite their underlying strength. First and most prevalent, spies often gave **generic or incorrect answers** while non-spies provided specific signals about the target entity ($\mathcal{E}_t$). For instance, in an L_{ID} match where $\mathcal{E}_t$ is "Binus," the spy responded vaguely about costs ("it depends on perspective..." translated from `id`), failing to recognize it as a famously expensive private university, which effectively revealed their identity. Second, models occasionally asked **"obvious fishing questions"** to gather information. In one L_{zh} match involving "Peking University," the spy asked about the "main users" of the place, even though the academic context had already been explicitly established by non-spies. Finally, matches in L_{arz} and L_{zh} showed spies being **voted out for targeting innocent non-spies**, triggering retaliatory votes that eliminated the spy regardless of their previous answer quality.

Majority wins of non-spies are due to spy guessing wrongly Table 7 shows that, overall, most non-spy wins are due to the spy guessing wrongly (74.5% across models), while comparable percentages are due to votes caught and spies quitting. There are some differences in the Gemini families, where both have close to 60% of non-spy wins due to the spy guessing wrongly and around 25% due to votes caught, which is significantly higher compared to other models (around 20%). This shows that the Gemini families have better spy detection capabilities compared to other models.

More analysis on probing this section can be seen further in Appendix G

6 Related Works

Dynamic and Game-Based LLM Evaluation Recent research has increasingly leveraged strategic games to evaluate LLMs capabilities. Some benchmarks assess LLMs capability on games, such as social deduction games. For instance, AvalonBench (Light et al., 2023) evaluates deception and negotiation skills, while several studies utilize Werewolf to assess emergent strategic behaviors (Bailis et al., 2024; Agarwal et al., 2025; Song et al., 2025). Additionally, AmongAgents (Chi et al., 2024) adapts the game Among Us into a fully text-based format. Similarly, (Kim et al., 2025) investigates Spyfall, though their evaluation is limited to generic settings in English.

Beyond social deduction, debate-based evaluation (Moniri et al., 2025) and interactive fiction (Hausknecht et al., 2020; Côté et al., 2019) probe oversight and reasoning capabilities. Strategic benchmarks like GTBench (Duan et al., 2024), and GameBench (Costarelli et al., 2024) reveal that LLMs excel in probabilistic scenarios but struggle with complete-information games. Some works also collect aggregated multi-agent benchmarks like AgentBench (Liu et al., 2025).

Multilingual and Multicultural Benchmark Culture is often operationalized through proxies representing specific concepts (Adilazuarda et al., 2024). Several works have constructed static multicultural benchmarks within native language contexts such as CulturalBench (Chiu et al., 2025), BLEnD (Myung et al., 2025), GlobalPIQA (Chang et al., 2025), ArabCulture (Sadallah et al., 2025), or COPAL-ID (Wibowo et al., 2024). These works mostly show that models are struggling with non-western contexts. However, these benchmarks typically rely on static formats that are susceptible to data contamination and performance saturation, limitations that we address in this paper.

7 Conclusion

We introduce a dynamic benchmarking framework based on Spyfall to evaluate LLMs on their multilingual and multicultural reasoning. Our results show that while top models perform well, there is still a noticeable drop in capability when models are tested in non-English languages or on specific cultural topics. This framework offers a reliable and scalable evaluation tool for how well models truly understand the diverse local cultural context.

Future work will explore the application of dynamic benchmarking to interactive tasks like debate to further assess multilingual and multicultural nuances, as our framework is significantly more scalable and offers greater resilience to data leakage and saturation.

Limitations

While this study focuses on three countries and their primary languages, it establishes a robust framework for evaluating regional linguistic nuances. Although the current dataset includes 30 local landmarks and 30 traditional foods per country, the benchmark is designed to be highly scalable. Its dynamic architecture allows for the seamless integration of additional cultural entities and languages in future iterations.

Finally, employing four identical model instances for the non-spy team may introduce behavioral coupling. However, the modular nature of our environment fully supports the future exploration of heterogeneous, many-to-many model interactions.

Ethical Considerations

Cultural Representation and Bias: While we aim to evaluate multicultural understanding, we acknowledge that our selection of 30 food items and 30 landmarks per country is non-exhaustive. These entities represent a subset of cultural identities and may reflect certain regional biases within the chosen countries. We have made efforts to include widely recognized cultural markers to ensure fair testing.

Intentional Deception: The social deduction framework involves models playing the role of a "Spy," which requires the use of strategic concealment or misdirection. We emphasize that this is used strictly as a proxy for reasoning and communication capability within a game context and should not be applied to encourage harmful deceptive behavior in real-world applications.

LLMs Usage: We use LLMs such as Gemini and Writefull (Overleaf) to correct the grammar in our writing.

References

Muhammad Farid Adilazuarda, Sagnik Mukherjee, Pradhyumna Lavania, Siddhant Shivdutt Singh, Alham Fikri Aji, Jacki O'Neill, Ashutosh Modi, and Monojit Choudhury. 2024. Towards measuring and modeling "culture" in LLMs: A survey. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15763–15784, Miami, Florida, USA. Association for Computational Linguistics.

Mrinal Agarwal, Saad Rana, Theo Sundoro, Hermela Berhe, Spencer Kim, Vasu Sharma, Sean O'Brien, and Kevin Zhu. 2025. Wolf: Werewolf-based observations for llm deception and falsehoods. *Preprint*, arXiv:2512.09187.

Suma Bailis, Jane Friedhoff, and Feiyang Chen. 2024. Werewolf arena: A case study in llm evaluation via social deduction. *Preprint*, arXiv:2407.13943.

Ralph Allan Bradley and Milton E Terry. 1952. Rank analysis of incomplete block designs: I. the method of paired comparisons. *Biometrika*, 39(3/4):324–345.

Tyler A Chang, Catherine Arnett, Abdelrahman Eldesokey, Abdelrahman Sadallah, Abeer Kashar, Abolade Daud, Abosede Grace Olanihun, Adamu Labaran Mohammed, Adeyemi Praise, Adhikarinayum Meerajita Sharma, and 1 others. 2025. Global piqa: Evaluating physical commonsense reasoning across 100+ languages and cultures. *arXiv preprint arXiv:2510.24081*.

Yizhou Chi, Lingjun Mao, and Zineng Tang. 2024. Amongagents: Evaluating large language models in the interactive text-based social deduction game. *Preprint*, arXiv:2407.16521.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios Nikolas Angelopoulos, Tianle Li, Dacheng Li, Hao Zhang, Banghua Zhu, Michael Jordan, Joseph E. Gonzalez, and Ion Stoica. 2024. Chatbot arena: An open platform for evaluating llms by human preference. *Preprint*, arXiv:2403.04132.

Yu Ying Chiu, Liwei Jiang, Bill Yuchen Lin, Chan Young Park, Shuyue Stella Li, Sahithya Ravi, Mehar Bhatia, Maria Antoniak, Yulia Tsvetkov, Vered Shwartz, and Yejin Choi. 2025. CulturalBench: A robust, diverse and challenging benchmark for measuring LMs' cultural knowledge through human-AI red-teaming. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25663–25701, Vienna, Austria. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, Luke Marris, Sam Petulla, Colin Gaffney, Asaf Aharoni, Nathan Lintz, Tiago Cardal Pais, Henrik Jacobsson, Idan Szpektor, Nan-Jiang Jiang, and 3416 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *Preprint*, arXiv:2507.06261.

Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. 2024. Gamebench: Evaluating strategic reasoning abilities of llm agents. *arXiv preprint arXiv:2406.06613*.

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Ruo Yu Tao, Matthew Hausknecht, Layla El Asri,

Mahmoud Adada, Wendy Tay, and Adam Trischler. 2019. Textworld: A learning environment for text-based games. *Preprint*, arXiv:1806.11532.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. 2024. Gtbench: Uncovering the strategic reasoning limitations of llms via game-theoretic evaluations. *Preprint*, arXiv:2402.12348.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. Interactive fiction games: A colossal adventure. *Preprint*, arXiv:1909.05398.

Junjie Hu, Sebastian Ruder, Aditya Siddhant, Graham Neubig, Orhan Firat, and Melvin Johnson. 2020. Xtreme: A massively multilingual multi-task benchmark for evaluating cross-lingual generalisation. In *International conference on machine learning*, pages 4411–4421. PMLR.

Lanxiang Hu, Qiyu Li, Anze Xie, Nan Jiang, Ion Stoica, Haojian Jin, and Hao Zhang. Gamearena: Evaluating llm reasoning through live computer games. In *The Thirteenth International Conference on Learning Representations*.

Go Inoue, Bashar Alhafni, Nurpeiis Baimukan, Houda Bouamor, and Nizar Habash. 2021. The interplay of variant, size, and task type in Arabic pre-trained language models. In *Proceedings of the Sixth Arabic Natural Language Processing Workshop*, pages 92–104, Kyiv, Ukraine (Virtual). Association for Computational Linguistics.

Armand Joulin, Edouard Grave, Piotr Bojanowski, and Tomas Mikolov. 2017. Bag of tricks for efficient text classification. In *Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistics: Volume 2, Short Papers*, pages 427–431, Valencia, Spain. Association for Computational Linguistics.

Amr Keleg, Sharon Goldwater, and Walid Magdy. 2025. Revisiting common assumptions about Arabic dialects in NLP. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3309–3327, Vienna, Austria. Association for Computational Linguistics.

Byungjun Kim, Dayeon Seo, Minju Kim, and Bugeun Kim. 2025. Fine-grained and thematic evaluation of llms in social deduction game. *IEEE Access*.

Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. 2023. Avalonbench: Evaluating llms playing the game of avalon. *Preprint*, arXiv:2310.05036.

Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, Shudan Zhang, Xiang Deng, Aohan Zeng, Zhengxiao Du, Chenhui Zhang, Sheng Shen, Tianjun Zhang, Yu Su, Huan Sun, and 3 others. 2025. Agentbench: Evaluating llms as agents. *Preprint*, arXiv:2308.03688.

Xinbei Ma, Ruotian Ma, Xingyu Chen, Zhengliang Shi, Mengru Wang, Jen tse Huang, Qu Yang, Wenxuan Wang, Fanghua Ye, Qingxuan Jiang, Mengfei Zhou, Zhuosheng Zhang, Rui Wang, Hai Zhao, Zhaopeng Tu, Xiaolong Li, and Linus. 2025. The hunger game debate: On the emergence of over-competition in multi-agent systems. *Preprint*, arXiv:2509.26126.

Behrad Moniri, Hamed Hassani, and Edgar Dobriban. 2025. Evaluating the performance of large language models via debates. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pages 2040–2075, Albuquerque, New Mexico. Association for Computational Linguistics.

Junho Myung, Nayeon Lee, Yi Zhou, Jiho Jin, Rifki Afina Putri, Dimosthenis Antypas, Hsuvas Borkakoty, Eunsu Kim, Carla Perez-Almendros, Abinew Ali Ayele, Víctor Gutiérrez-Basulto, Yazmín Ibáñez-García, Hwaran Lee, Shamsuddeen Hassan Muhammad, Kiwoong Park, Anar Sabuhi Rzayev, Nina White, Seid Muhie Yimam, Mohammad Taher Pilehvar, and 3 others. 2025. Blend: A benchmark for llms on everyday knowledge in diverse cultures and languages. *Preprint*, arXiv:2406.09948.

Nathaniel Romney Robinson, Shahd Abdelmoneim, Kelly Marchisio, and Sebastian Ruder. 2025. ALQASIDA: Analyzing LLM quality and accuracy systematically in dialectal Arabic. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 22048–22065, Vienna, Austria. Association for Computational Linguistics.

Abdelrahman Sadallah, Junior Cedric Tonga, Khalid Almubarak, Saeed Almheiri, Farah Atif, Chatrine Qwaider, Karima Kadaoui, Sara Shatnawi, Yaser Alesh, and Fajri Koto. 2025. Commonsense reasoning in Arab culture. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7695–7710, Vienna, Austria. Association for Computational Linguistics.

Claude E. Shannon. 1948. A mathematical theory of communication. *The Bell System Technical Journal*, 27(3):379–423.

Zirui Song, Yuan Huang, Junchang Liu, Haozhe Luo, Chenxi Wang, Lang Gao, Zixiang Xu, Mingfei Han, Xiaojun Chang, and Xiuying Chen. 2025. Beyond survival: Evaluating llms in social deduction games with human-aligned strategies. *Preprint*, arXiv:2510.11389.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, and 197 others. 2025. Gemma 3 technical report. *Preprint*, arXiv:2503.19786.

Haryo Wibowo, Erland Fuadi, Made Nityasya, Radityo Eko Prasojo, and Alham Aji. 2024. COPAL-ID: Indonesian language reasoning with local culture and nuances. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 1404–1422, Mexico City, Mexico. Association for Computational Linguistics.

Minghao Wu, Weixuan Wang, Sinuo Liu, Huifeng Yin, Xintong Wang, Yu Zhao, Chenyang Lyu, Longyue Wang, Weihua Luo, and Kaifu Zhang. 2025. The bitter lesson learned from 2,000+ multilingual benchmarks. *Preprint*, arXiv:2504.15521.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, Chujie Zheng, Dayiheng Liu, Fan Zhou, Fei Huang, Feng Hu, Hao Ge, Haoran Wei, Huan Lin, Jialong Tang, and 41 others. 2025. Qwen3 technical report. *Preprint*, arXiv:2505.09388.

A Game Endings

The winning conditions for each role are as follows:

Spy's win conditions The spy wins if they can deduce $\mathcal{E}t$ in $\mathcal{P}_{SG}$, a non-spy is majority voted in $\mathcal{P}V$, the turn limit is exceeded, or one of the non-spies quits the game (e.g., a formatting issue in the LLM's output).

Non-spies' win conditions The non-spy team wins if the spy guesses $\mathcal{E}t$ incorrectly, the spy gets majority voted in $\mathcal{P}_V$, or the spy quits the game (e.g., a formatting issue in the LLM's output).

B Selected Entities

Here are entities that we selected with care for the model for each scenario:

Generic Entities (EN): Airplane, Amusement Park, Bank, Beach, Carnival, Casino, Circus Tent, Corporate Party, Crusader Army, Day Spa, Embassy, Hospital, Hotel, Military Base, Movie Studio, Nightclub, Ocean Liner, Passenger Train, Police Station, Pirate Ship, Polar Station, Restaurant, School, Service Station, Space Station, Submarine, Supermarket, Theater, University, Zoo.

Generic Entities (ID): Pesawat Terbang, Taman Hiburan, Bank, Pantai, Karnaval, Kasino, Tenda Sirkus, Pesta Perusahaan, Pasukan Perang Salib, Spa, Kedutaan Besar, Rumah Sakit, Hotel, Pangkalan Militer, Studio Film, Klub Malam, Kapal Pesiar, Kereta Penumpang, Kantor Polisi, Kapal Bajak Laut, Stasiun Kutub, Restoran, Sekolah, Bengkel, Stasiun Luar Angkasa, Kapal Selam, Supermarket, Teater, Universitas, Kebun Binatang.

Generic Entities (ZH): 飞机, 游乐园, 银行, 海滩, 嘉年华, 赌场, 马戏团帐篷, 公司派对, 十字军, 水疗中心, 大使馆, 医院, 酒店, 军事基地, 电影制片厂, 夜总会, 远洋客轮, 客运火车, 警察局, 海盗船, 极地站, 餐厅, 学校, 加油站, 空间站, 潜水艇, 超市, 剧院, 大学, 动物园.

Generic Entities (ARZ): طيارة, ملاهي, بنك, جيش, شركة حفلة, سيرك, كازينو, كرنفال, شاطئ, الصليبيين, نهاري سبا, سفارة, مستشفى, فندق, عسكرية قاعدة, افلام تصوير ستوديو, كلوب نايت, ركاب سفينة, ركاب قطر, قسم شرطة, قراصنة سفينة, قطبية محطة, مطعم, مدرسة, بنزين محطة, فضاء, محطة, غواصة, ماركت سوبر, مسرح, جامعة, الحيوان, حديقة.

ARZ Food: محشي حمام, العيد كحك, كراويه, حصى حلبة, كنافة, شكشوكة, ممبار, بيض بالبسطرمة,

ملوخية, ينسون, حواوشي, كوارع شوربة, مسقعة, حلة, بالفراخ بطاطس صينية, محشي, الشام حمص, ترمس, كشري, عصفور لسان شوربة, فور بيتي, كباب, مدمس فول, زينب صوابع, مشوية بطاطا, سحلب, ورنجة فسيخ, بلبن رز, لحمة فتة, سوبيا, كركديه.

ID Food: Nastar, Nasi Tumpeng, Roti Buaya, Nasi Uduk, Lontong Sayur, Rendang, Ayam Taliwang, Babi Guling, Gado Gado, Tempe Mendoan, Capcai, Martabak, Lapis Legit, Bika Ambon, Cimol, Sate, Pempek, Bakso, Coto Makasar, Rawon, Seblak, Cakalang fufu rica-rica, Tuak, Cap Tikus, Cendol, Soda Gembira, Es Lidah Buaya, Beras Kencur, Bandrek, Loloh Cemcem.

ZH Food: 馒头, 花卷, 发糕, 窝头, 烧饼, 馕, 元宵, 粽子, 年糕, 鱼香肉丝, 回锅肉, 地三鲜, 水煮鱼, 辣子鸡, 白切鸡, 烧鹅, 羊肉泡馍, 葱爆羊肉, 东坡肉, 煎饼果子, 肉夹馍, 臭豆腐, 凉皮, 手抓饼, 串串香, 茶叶蛋, 肠粉, 双皮奶, 芝麻糊, 切糕.

ARZ Places: الكبير, بالقاهرة الامريكية الجامعة, مول سيتي فيستيفال, الاهرامات, المصري المتحف, علي, حجوجة, فؤاد بور, ستانلي كوبري, تسيباس, كايرو, النيل, المعلقة الكنيسة, الازهر جامع, محمد مسجد, للحضارة القومي, ستيفانو سان, مصر سكي, جامعة, النيل قصر, القاهرة برج, شمس عين جامعة, المتحف, الملوك, سيلانترو, نصر مدينة, العبد, الحسين, كوبري, طارق ابو, جاد, بارك دريم, ستارز سيتي, حلوان, وادي, الدهان كبابجي, اسكندرية مكتية, كشري.

ID Places: Trans Studio, Malioboro, Ragunan, Tanah Abang, Menteng, J.C.O, Pizza Domino, Fort Rotterdam, Ciater, Jam Gadang, Nusa Penida, Bromo, Dufan, Gedung Sate, ITB, MAN Insan Cendekia, SMAN 8 Jakarta, Binus, Tuku, Sushi Tei, Restoran Sederhana, Warteg, Taman Nasional Sebangau, Pura Agung Besakih, Monas, BSD City, Blok M, Istiqlal, Borobudur, Kota Tua.

ZH Places: 瑞幸咖啡, 喜茶, 蜜雪冰城, 海底捞火锅, 西贝莜面村, 稻香村, 必胜客, 麦当劳, 王府井大街, 三里屯太古里, 南京路步行街, 故宫博物院, 秦始皇兵马俑, 上海迪士尼乐园, 天安门广场, 东方明珠电视塔, 布达拉宫, 卢沟桥, 乐山大佛, 都江堰, 北京大学, 清华大学, 复旦大学, 中国科学院, 上海交大, 少林寺, 雍和宫, 上海外滩, 陆家嘴, 中关村.

C Prompt Templates

During each game phase, every model receives a prompt containing the rules, game history, a phase description, and response instructions. Responses must adhere to a strict JSON format; any output that fails to do so is deemed invalid. In such instances, the team responsible for the invalid move immediately forfeits the game quit (e.g., an invalid move by the spy results in a victory for the non-spies). We argue that a model's inability to follow structural instructions is a clear indicator of limited capability, making a loss a fair and representative outcome. The full prompts are provided in Appendix .

This is the rule prompt that is always inserted in the LLM's prompt, where the languages are: "Indonesian", "Simplified Chinese", "Egyptian Arabic", and "English" (depends on the scenario run). Each phase and role will have their respective prompt. Here are the respective prompts:

```
# Base context used for all prompts
base_context: |
  **Game Context:** You are playing "Spyfall:
       Structured Edition" with other players
      .

  **How Spyfall Structured Edition Works:**
  - One player is secretly assigned as "the
      Spy" at the start
  - All other players are "Villagers/Non-
      Spies" who know the secret entity
  - The Spy does not know the entity and must
       figure it out
  - Villagers must identify who the Spy is
      through questioning and voting

  **What is the Entity?**
  The entity can be anything: a location (
      Beach, Hospital), a movie (Star Wars,
      Titanic),
  a famous person (Einstein, Beyoncé), an
      object (Smartphone, Pizza), a concept (
      Birthday Party, Job Interview), etc.
  Villagers know the entity; the Spy must
      deduce it from questions and answers.

  **Structured Game Flow (Predictable Phases)
      :**

  1. Round Robin Phase (Phase 1A):
     - Each player asks the next player one
         question sequentially
     - Goes around the full circle once
     - Everyone participates, building
         baseline information

  2. Free Question Cycles (n cycles of Phase
      1B-2-3):
     Each cycle has three phases:

     a) Phase 1B - Free Question:
        - Last answerer asks anyone one
            question
        - Provides targeted interrogation
            opportunities

     b) Phase 2 - Spy Guess Decision:
        - Spy can guess the entity or skip
        - If correct guess → Spy wins
        - If incorrect guess → Villagers win
        - Spy gets n chances throughout the
```

```
            game

    c) Phase 3 - Accusation Vote:
       - All players vote for someone (or
           skip)
       - Need majority (>50%) to eliminate
           someone
       - If eliminated player is Spy → 
           Villagers win
       - If eliminated player is Villager →
           Spy wins
       - No majority → game continues to next
            cycle

  3. Final Round (after n cycles):
     - Final Spy guess opportunity
     - Final accusation vote
     - If no majority in final vote → Spy wins
         (survived!)

  Note: n equals the number of players in the
       game.

  **Win Conditions:**
  - Villagers win: Spy guesses wrong OR
      majority votes eliminate Spy
  - Spy wins: Spy guesses correctly OR
      survives all votes OR Villagers
      eliminate innocent player

  **Strategic Differences from Classic:**
  - Voting happens in cycles, not randomly
  - Spy has multiple guess opportunities (
      strategic timing!)
  - Vote history is public (builds narrative
      across rounds)
  - Structured phases allow better planning

  **Your Task:** Based on your role (Spy or
      Villager), play strategically:

  *As a Villager:*
  Share entity knowledge subtly and build
      your case across voting rounds. Avoid
      being obvious! Be extremely subtle in
      giving hints. Think carefully about how
       you answer and ask questions
      strategically.

  Important guidelines:
  - Do not ask or answer questions too
      directly
  - Review the entity list carefully to avoid
       making it easy for the Spy to guess
  - Avoid narrowing it down to one specific
      place or category
  - Do not provide direct hints to the entity
       (e.g., avoid saying "crew" or "vessel"
       when the entity is Pirate Ship, or "
      extreme" when it's Polar Bear)
  - Use subtlety and misdirection to make the
       Spy think multiple entities are
      possible
  - Poor strategy example: Mentioning "
      patient" when the entity is Hospital,
      or "temporary stay" when the entity is
      Hotel
  - Be very subtle so the Spy still needs to
      guess among many possibilities

  *As a Spy:*
  - Blend in, gather clues across multiple
      rounds, and time your guess
      strategically.
  - Avoid being obvious and don't reveal that
       you're the Spy.
  - Answer strategically using misdirection
      and subtlety.
  - Do not guess the entity if you cannot
      narrow it down to one location yet in
      Spy Guess Decision phase.

  Important guidelines:
  - Blend in and avoid obvious behavior
  - Use misdirection and subtlety in your
      questions and answers
  - Poor strategy example: Asking "Is it a
      place where people go to relax?" when
      the entity is Beach
  - Be very subtle in your approach

  **Critical Reminders:**
  - Check the entity guesses list that has
      been provided to you
  - Use it to guide your questioning and
      answering strategy (for both Spy and
      Villagers)
  - Consider conversation history, voting
      patterns, and strategic phase timing
  - Ensure you adhere to the response format
  - You must speak in {language} language at
      any cost!

# Prompts for question generation
question_generation:
  spy: |
    {base_context}

    You are the Spy and need to ask a question
         to another player.

    **Response Format Requirements:**
    You must respond with only valid JSON
        wrapped in triple pipes |||...|||
    Think through your approach before
        outputting JSON. No text should
        appear after the JSON.
    Think step by step!

    Example:
    Let's think step by step...
    |||
    {
      "question": "YOUR QUESTION HERE",
      "targeted_player": "TARGET_PLAYER_NAME"
    }
    |||

    Required fields:
    - question: string (the actual question to
         ask)
    - targeted_player: string (the player
        being targeted by the question. Think
         strategically about who to ask)

  non_spy: |
    {base_context}

    You are a Villager and need to ask a
        question.
```

```
    **Response Format Requirements:**
    You must respond with only valid JSON
        wrapped in triple pipes |||...|||
    Think through your approach before
        outputting JSON. No text should
        appear after the JSON.
    Think step by step!

    Example:
    Let's think step by step...
    |||
    {
      "question": "YOUR QUESTION HERE",
      "targeted_player": "TARGET_PLAYER_NAME"
    }
    |||

    Required fields:
    - question: string (the actual question to
         ask)
    - targeted_player: string (the player
        being targeted by the question. Think
         strategically about who to ask)

# Prompts for answer generation
answer_generation:
  spy: |
    {base_context}

    You are the Spy answering a question.

    **Response Format Requirements:**
    You must respond with only valid JSON
        wrapped in triple pipes |||...|||
    Think through your approach before
        outputting JSON. No text should
        appear after the JSON.
    Think step by step!

    Example:
    Let's think step by step...
    |||
    {
      "answer": "It can get a bit
           uncomfortable, so I usually prepare
           accordingly."
    }
    |||

    Required fields:
    - answer: string (your actual answer)

    Important: Do not make it obvious that you
         are the Spy!

  non_spy: |
    {base_context}

    You are a Villager answering a question.

    **Response Format Requirements:**
    You must respond with only valid JSON
        wrapped in triple pipes |||...|||
    Think through your approach before
        outputting JSON. No text should
        appear after the JSON.
    Think step by step!

    Example:
    Let's think step by step...
    |||
    {
      "answer": "YOUR ANSWER HERE"
    }
    |||

    Required fields:
    - answer: string (your actual answer)

# Prompts for entity guessing (spy only)
entity_guess:
  spy: |
    {base_context}

    You are the Spy in Phase 2 (Spy Guess
        Decision).

    Important: You do not have to rush if you
        are unsure! Skip if you haven't
        pinpointed a single location yet.

    Your task: Decide whether to guess the
        entity now or skip.

    **Response Format Requirements:**
    You must respond with only valid JSON
        wrapped in triple pipes |||...|||
    Think through your approach before
        outputting JSON. No text should
        appear after the JSON.
    Think step by step!

    Example (making a guess):
    Let's think step by step...
    |||
    {
      "best_guess": "Beach",
      "should_guess": true,
      "confidence": 0.85
    }
    |||

    Example (skipping):
    Let's think step by step...
    |||
    {
      "best_guess": null,
      "should_guess": false,
      "confidence": 0.3
    }
    |||

    Required fields:
    - best_guess: string or null (your guess
        from the entity list if should_guess=
        true, else null)
    - should_guess: boolean (true to guess now
        , false to skip)
    - confidence: number 0.0-1.0 (how
        confident you are in your guess)

# Prompts for vote_initiation (deciding who
    to vote for in Structured Edition)
vote_initiation:
  spy: |
    {base_context}

    You are the Spy deciding who to vote for
        in the Accusation Vote phase.
```

```
    You can skip voting if you are unsure!

    **Response Format Requirements:**
    You must respond with only valid JSON
        wrapped in triple pipes |||...|||
    Think through your approach before
        outputting JSON. No text should
        appear after the JSON.
    Think step by step!

    Example:
    Let's think step by step...
    |||
    {
      "target_player_name": "Charlie",
      "should_vote": true,
      "confidence": 0.75
    }
    |||

    Example to skip voting:
    Let's think step by step...
    |||
    {
      "target_player_name": null,
      "should_vote": false,
      "confidence": 0.6
    }
    |||

    Required fields:
    - target_player_name: string or null (
        player name, or null to skip)
    - should_vote: boolean (true to vote for
        target, false to skip)
    - confidence: number 0.0-1.0 (confidence
        in your voting decision)

  non_spy: |
    {base_context}

    You are a Villager deciding who to vote
        for in the Accusation Vote phase (
        Phase 3).
    You can skip voting if you are unsure!

    **Response Format Requirements:**
    You must respond with only valid JSON
        wrapped in triple pipes |||...|||
    Think through your approach before
        outputting JSON. No text should
        appear after the JSON.
    Think step by step!

    Example:
    Let's think step by step...
    |||
    {
      "target_player_name": "Charlie",
      "should_vote": true,
      "confidence": 0.88
    }
    |||

    To skip voting:
    Let's think step by step...
    |||
    {
      "target_player_name": null,
      "should_vote": false,
      "confidence": 0.45
    }
    |||

    Required fields:
    - target_player_name: string or null (
        player name, or null to skip)
    - should_vote: boolean (true to vote for
        target, false to skip)
    - confidence: number 0.0-1.0 (confidence
        in your voting decision)
```

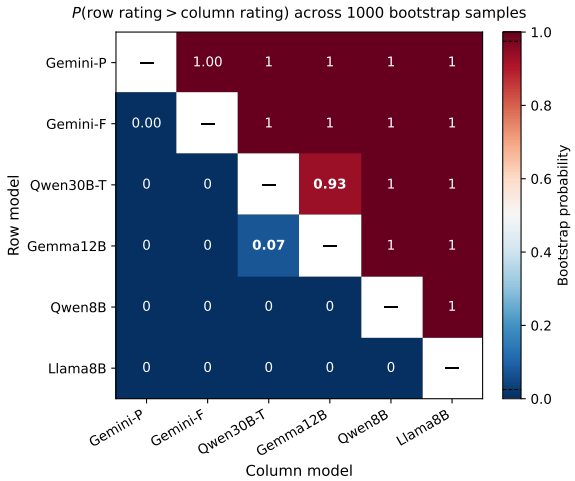

Figure 5: Pairwise probability that the row model has a higher BT rating than the column model, computed from 1000 bootstrap resamples on overall matches. Values ≥0.975 (or ≤0.025) indicate statistically significant rank ordering at the 5% level.

D Statistical Robustness in Overall Game Rating

In Table 1, to assess rank stability under sampling variability, we compute 95% confidence intervals with 1000-replicate match-level bootstrap following the work of Chiang et al., 2024. The top two (Gemini-P, Gemini-F) and bottom one (Llama-8B) maintain non-overlapping CIs against all other models across every scenario, indicating robust positioning. However, the mid-tier models ordering between Qwen30B-T (1016) and Gemma12B (1003) is not statistically distinguishable in the overall ranking (CIs [1005, 1028] and [992, 1014] overlap; bootstrap P(Qwen30B-T>Gemma12B)=0.93P(Qwen30B-T > Gemma12B) = 0.93 (Figure 5), Qwen30B-T, and these two models flip ranks in 4 of 10 scenarios, which shows that these models are competitive. Within-scenario CIs (half-width ≈35) are approximately 3× wider than overall (≈11), reflecting the 10× smaller sample size.

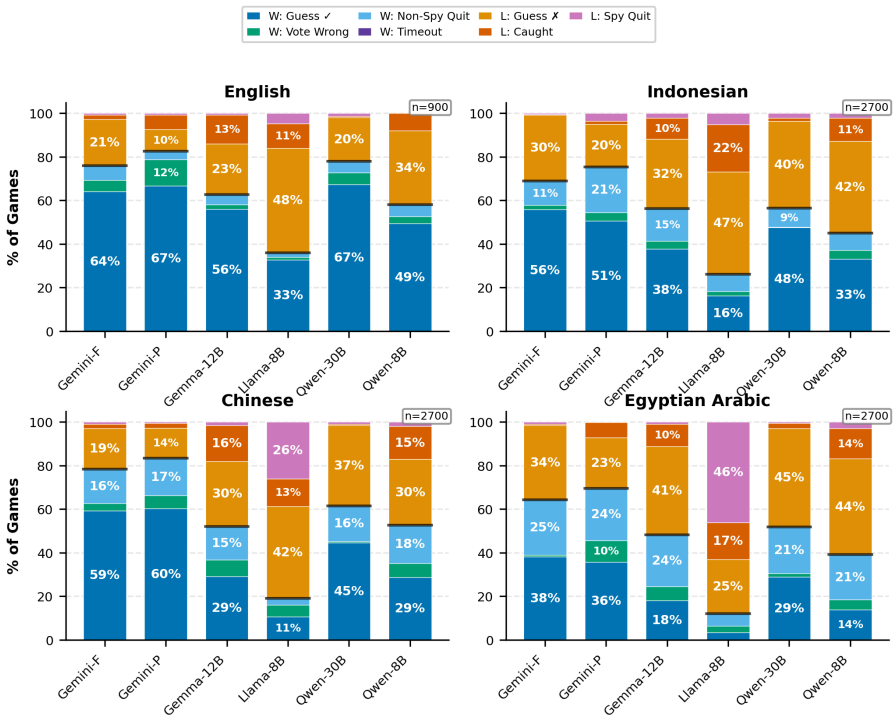

Figure 6: Game End Statistics by Language

E Why do models maintain similar rating gaps despite these spy guess accuracy drop?

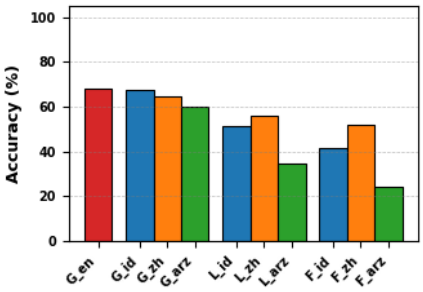

Figure 7: Spy Guess Accuracy Rate by Scenario

First, we direct you to observe the following:

In Figure 7, in `id`, Generic, Local Location, and Local Food win rates are ~65% ~→ 50%~ → 40%.

Focusing on `id` in Table 5 shows that Entity-level uncertainty increases dramatically compared to the generic scenario: Generic entities show 67.1% accuracy with an entropy of 1.49, while Local Food drops to 40.9% accuracy with an entropy of up to 2.80. Local Locations fall to 49.6% accuracy with an entropy that reaches 2.10, indicating much higher uncertainty.

These findings show that entity knowledge differentially affects model performance, but the rating shown in Table 1 is maintained. We dive in to answer this occurrence.

Complementing our analysis in Section Z (**Lower-Accuracy Entities Share Characteristics with Distractors**), we analyzed the win rates mechanism by observing `arz` (the most challenging scenarios for a spy to guess). Below, we present the analysis of spy wins via correct guesses and non-spy wins via incorrect spy guesses.

By observing Table 8 and Table 9, The rating stability persists because when models play as spies, their insufficient cultural knowledge hurts them, but when they play as non-spies, these occurrences help them by causing opponent spies to fail. For instance:

1. In Gemini Pro, spy wins drop from Generic 49.3% → Food 29.7%, while Non-spy wins increase (Generic 34% → Food 60%)

2. In Qwen30B-T, spy wins drop from Generic 50.0% → Food 15.3%, while Non-spy wins increase (Generic 28% → Food 54.7%)

This reveals that non-spies don't necessarily perform better strategically in local contexts, they simply benefit from opponent spy failures on unfamiliar entities.

Model	Generic Spy Win	Local Food Spy Win	Local Loc. Spy Win
Gemini-P	49.33% (n=74)	24.67% (n=37)	33.33% (n=50)
Gemini-F	47.33% (n=71)	28.67% (n=43)	38.67% (n=58)
Qwen30B-T	50.00% (n=75)	15.33% (n=23)	21.33% (n=32)
Gemma3-12B	33.33% (n=50)	10.67% (n=16)	10.00% (n=15)
Qwen3-8B	22.00% (n=33)	8.67% (n=13)	10.67% (n=16)
Llama-3.1-8B	7.33% (n=11)	1.33% (n=2)	1.33% (n=2)

Table 8: Spy win rates with guessed location across Generic, Local Food, and Local Location settings.

Model	Generic Non-Spy Win	Local Food Non-Spy Win	Local Loc. Non-Spy Win
Gemini-P	34.00% (n=51)	60.00% (n=90)	50.67% (n=76)
Gemini-F	23.33% (n=35)	54.00% (n=81)	44.00% (n=66)
Qwen30B-T	28.00% (n=42)	54.67% (n=82)	50.00% (n=75)
Gemma3-12B	30.00% (n=45)	43.33% (n=65)	35.33% (n=53)
Qwen3-8B	20.67% (n=31)	59.33% (n=89)	38.00% (n=57)
Llama-3.1-8B	4.67% (n=7)	4.67% (n=7)	2.67% (n=4)

Table 9: Non-Spy win rates (spy guessed wrong) across Generic, Local Food, and Local Location settings.

Another interesting thing that we observe is that spy guess accuracy does impact models' rating placements. However, examining the above Generic scenarios in `arz` reveals that Qwen30B-T and Gemini Pro achieve nearly equal spy guess rates (50%), yet Gemini Pro maintains a higher overall rating (1163 vs 1032). This gap persists for several additional reasons, such as: Gemini Pro demonstrates better spy detection capabilities when playing as a non-spy (Table 7 and Figure 10) and employs vote deception as a spy that misleads as a non-spy.

F Language Identification Setup

We identified the language using fastText (Joulin et al., 2017). Then, we process rows detected as Arabic with Arabic dialect identification trained on the MADAR Corpus (Inoue et al., 2021), which has demonstrated better performance on fine-grained Arabic dialect distinctions.

G Additional Behavior Analysis

In Overall, Gemini Pro followed by Gemma3-12B has the highest percentage of Incorrect Spy Voting by non-spies As shown in Table 10, Gemini Pro has the highest percentage of incorrect spy voting by non-spies (8.5%), followed by Gemma3-12B (7.8%), showing that these models are more subtle in covering the location and mislead the non-spies to vote wrongly. Meanwhile, Llama3.1-8B has the lowest incorrect spy voting by non-spies (2.6%), showing that non-spies can easily identify the spy in this model.

Model	id	zh	arz
Gemini-F	2.0	3.3	0.7
Gemini-P	3.8	6.0	9.8
Gemma-12B	3.6	7.6	6.4
Llama-8B	2.0	5.3	3.1
Qwen-30B	0.2	0.4	1.6
Qwen-8B	4.0	6.4	4.7

Table 10: Vote wrong rate (%) by model across languages when these models play as a spy. This metric shows how often the opposite non-spies players incorrectly voted out a teammate.

The Game Dominantly Ends in one turn in Free Cycle Overall, the game is ended on either in P_{SG} or P_V on the first turn of the Free Cycle, with mean of 6.03, standard deviation of 0.67 and median of 6.00. We attribute this due to the Round Robin Cycle which allows Spy and Non-Spy gauge the information better. However, there are a significant amount of wrong spy guess (≈32.41%) shows that the spy players are confident despite guessing wrongly and the prompt that tells them to assure to guess if they are sure.

Spy Guess action dominates the game ending of a spy We then analyze the spy guess rate without leakage, where the spy must deduce the entity based on the QA history only. As shown in Figure 7, the spy wins are mostly due to spy guess $\mathcal{E}_t$ correctly, whereas spy also loses due to spy guessing wrongly. Despite having the prompt that tell the spy to avoid guessing unless it is confident to

guess, the spy tends to guess early which results also in high wrong guess (attributed more than 15% of total games for all models except for Gemini Pro in English and Chinese data). Additionally, Qwen 30B has the highest spy guess ratio (correct or wrong) demonstrate its behavior in this game to often do this action.

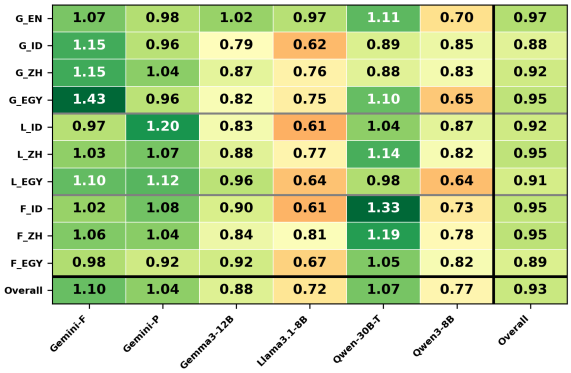

Figure 8: Spy Vote Dispersion Sccore by Model and Scenario. Higher is better.

Non-spy Vote Entropy In order to measure how united the non-spies are, we check the entropy of votes across non-spy players, where these are played by the same language model can be seen in Figure 9.

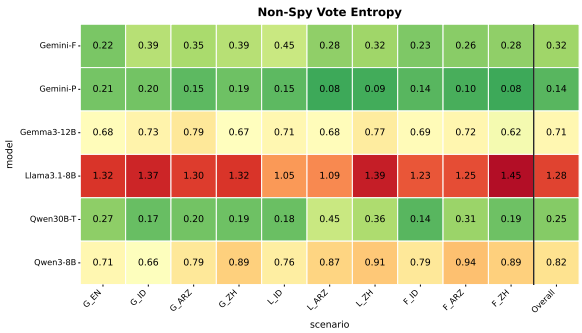

Figure 9: Non-spy Vote Entropy Heatmap

Spy Win Rate's rank follows overall rating Figure 4 shows the model wise win rate when playing as a spy across scenarios, where it is shown that the overall spy win rate follows the trend as the overall rating in Table 12.

Rank orders correspond to model size The ranking results suggest that, based on the size and capability of the models, closed-source models outperform open-source models, and larger models outperform smaller ones (Table 12). Additionally, within comparably sized models, we can see that Qwen3-8B outperforms Llama-3.1-8B by a large margin in our benchmark, which follows a similar trend observed in other benchmarks; for instance, in the Qwen 3 report (Yang et al., 2025). It is worth noting that the win rate also follows the Elo rating ranking.

Spy Vote Dispersion Varies Depends on The Model Capacity in In calculating the voting session when the spy is present, we introduce **Vote Dispersion**, which can be calculated using $H \times (1 - V_S)$, where H is the Shannon entropy value and V_S denotes the percentage of votes that the spy receives (1 means everyone votes for the spy) in a single voting session. This equation penalizes cases where entropy is low but the votes target the spy. A higher value is better for the spy player. The visualization (Figure 8) reveals the vote dispersion, where Gemini-F (1.095), Qwen-30B-T (1.070), and Gemini-P (1.037) achieve the highest dispersion scores, indicating superior ability to manipulate voting patterns and evade detection when acting as the spy. In contrast, smaller models like Llama3.1-8B (0.721) and Qwen3-8B (0.770) struggle significantly, being more easily identified and voted out. Scenario differences are relatively minor (0.878 to 0.974), suggesting that spy evasion success depends more on model capability than on language or local domain. The overall average of 0.929 indicates moderate voting chaos across all conditions.

Spy Players does not always guess correctly in match with leakage As shown in Table 11, despite the leakage happening, the spy guess rate with leakage does not guarantee to be 100% correct, though, overall, it is still considered high (more than 80%). Counting models that have more than 20 matches with leakage, ZH has overall lower spy guess accuracy compared to EN and ID languages which have comparable guess rate, shown by juxtaposing Gemma and Qwen models. An example of this case, given Gemini Flash as Spy Model in L_{ID}, where $\mathcal{E}_t$ is "Bika Ambon" (A cake from Indonesia), in one of the match, one of the non-spy leaks the entity by answering "Can we meet Bika Ambon in festival places in Java?" (Translated from `id`). The spy guesses wrongly by answering "Nastar" instead of "Bika Ambon" (Also a cake from Indonesia).

Non-spy Voting Accuracy As the non-spy objective is to find the spy, we measure the performance

Table 11: Spy guess accuracy with non-spy leakage by language.

Model	en	id	zh	arz
Gemini-F	100.0 (31)	94.1(17)	92.9(14)	87.5(8)
Gemini-P	95.2 (21)	100.0(23)	100.0(16)	100.0(6)
Gemma-12B	93.8 (16)	94.4(18)	85.7(14)	100.0(4)
Llama-8B	66.7 (6)	100.0(5)	83.3(6)	–
Qwen-30B	88.5 (26)	95.8(24)	86.7(15)	83.3(6)
Qwen-8B	88.0 (25)	96.3(27)	80.0(10)	0.0(1)

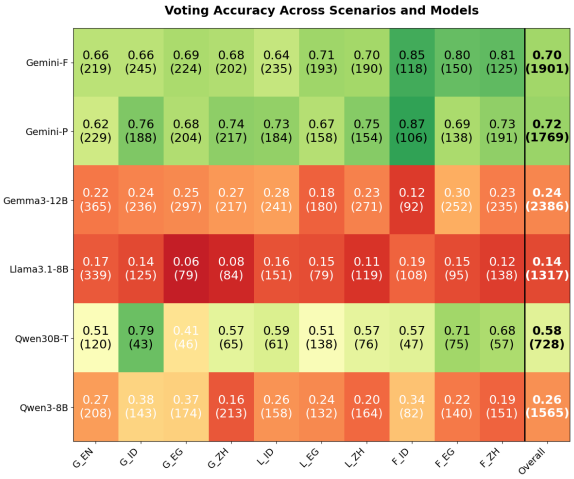

Figure 10: Non-spy Detective Rate

in which the non-spy successfully identify the spy using Voting Accuracy Metric. As the number of players in the game is five, the random baseline is 25%. Figure 10 shows the voting accuracy of each model across scenarios. Gemma3-12B, Qwen3-8B, and Llama3.1-8B have voting accuracy lower than random baseline in overall where Llama3.1-8B has the lowest voting accuracy (14%), meanwhile, Qwen3-30B Thinking has a moderate voting accuracy and Gemini families has comparable and higher voting accuracy (around 70%).

Qwen30B-Thinking has the least voting frequencies This is demonstrated in Figure 10, where in the calculation SKIP vote is excluded. the skip rate of Qwen30B-Thinking is 70.42% where others are around (22%-34%). This shows that Qwen30B-Thinking is more conservative in voting, which may lead to a higher voting accuracy.

Game Endings Are Dominated by Spy Guess Actions Table 6 shows the overall game outcome statistics across all models and scenarios. The most frequent game ending is when the spy guesses the location, comprising 36.19% correct guesses and 32.41% wrong guesses, summing to 68.6%. This shows that the spy is more likely to win

Rank	Model	Rating	Win Rate (%)
1	Gemini-P	1135.8	70.13
2	Gemini-F	1107.4	66.00
3	Qwen30B-T	1015.7	51.93
4	Gemma12B	1003.0	49.93
5	Qwen8B	966.8	44.30
6	Llama8B	771.3	17.70

Table 12: Bradley-Terry ratings and overall win rates for each model.

through guessing rather than being voted out. Additionally, surrender actions are also significant (≈19.41%), dominated by the Llama3.1-8B model (70.52% of all surrenders). Finally, the voting phase only contributes to 12% of all game endings, showing that the models are less likely to vote out the spy.

Egyptian Arabic Local Entities are the hardest to be guessed by the spy As shown in Table 5, in both local location and food scenarios, Egyptian Arabic language has the lowest overall entity accuracy and the highest vote entropy compared to other languages, showing that these scenarios are more challenging for the spy to deduce the entity. Additionally, Indonesian local location and food scenarios have a moderate difficulty, while Chinese local location and food scenarios are the easiest for the spy to deduce the entity.

Spy game ending As shown in Figure 6, in all languages, most models tend to win through correct spy guessing. Generally, the incorrect spy guess in `id`, `zh`, and `arz`, is higher than `en`. Additionally, in `zh` and `arz` languages, all models tend to win more through non-spy voting majority compared to `en` and `id`.