
STRATEGIC LLM DECODING THROUGH BAYESIAN GAMES

Weitong Zhang

Imperial College London

weitong.zhang@ic.ac.uk

Chengqi Zang

University of Tokyo

chengqi.zang@u-tokyo.ac.jp

Bernhard Kainz

Imperial College London

Friedrich-Alexander University Erlangen-Nürnberg

bernhard.kainz@ic.ac.uk

ABSTRACT

Large Language Models (LLMs) often produce outputs that – though plausible – can lack consistency and reliability, particularly in ambiguous or complex scenarios. Challenges arise from ensuring that outputs align with both factual correctness and human intent. This is problematic in existing approaches that trade improved consistency for lower accuracy. To mitigate these challenges, we propose a novel game-theoretic approach to enhance consistency and reliability during the decoding stage of LLM output generation. Our method models the decoding process as a multistage Bayesian Decoding Game. The strategic decoding process dynamically converges to a consensus on the most reliable outputs without human feedback or additional training. Remarkably, our game design allows smaller models to outperform much larger models through game mechanisms (*e.g.* 78.1 LLaMA13B vs 76.6 PaLM540B), as well as integrating various LLM strategies and models, demonstrating the potential of game-theoretic tools to improve the truthfulness and reliability of LLMs.

1 INTRODUCTION

Large Language Models (LLMs) have demonstrated extraordinary capabilities in tasks such as factual question answering, fact-checking, and open-ended text generation (Brown et al., 2020; Radford et al., 2021). Yet, like Damocles’ sword, this remarkable progress comes with a hidden price – as these generative models increase in complexity and scale, resulting in outputs that, while plausible, may be factually incorrect or subtly misleading (McKenzie et al., 2023). This dilemma – whether manifesting as an inevitable artifact of the model’s optimization process or as unintended hallucinations (Banerjee et al., 2024; Bai et al., 2024) – poses a fundamental challenge, often outpacing the ability of human judgment to accurately assess the fidelity and truthfulness of the generated content (Leike et al., 2018).

Traditional approaches attempt to optimize model outputs through human feedback (*e.g.*, RLHF (Christiano et al., 2017; 2018; Saunders et al., 2022; Markov et al., 2023)). However, human feedback, inherently constrained by limitations in interpretability (Singh et al., 2024) and the challenge of reliably assessing the complex logical structures (Hendrik Kirchner et al., 2024) of AI-generated content, cannot cope with increasingly complex reasoning (Casper et al., 2023; Leike et al., 2018). In light of these challenges, the reliability of LLMs in collaborative and high-stakes decision-making remains deeply uncertain and we pose the question:

How can we enable LLMs to systematically verify their outputs through strategic multi-agent interactions, surpassing both single-agent reasoning and human evaluation limitations

To answer this question, we explore a game-theoretic approach by introducing a *Verifier*, serving as a proxy for human judgment to systematically assess generators as outlined in Fig. 1. The motivation for this approach is threefold: (1) LLMs are increasingly employed to assist in evaluating

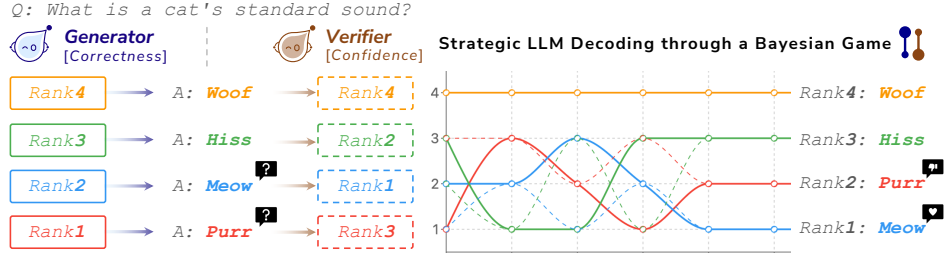


Figure 1: Illustration of strategic decoding through a Bayesian game (BDG) for cat sound decoding (Q.) with candidates (A.), with initial rankings (left) and convergence dynamics (right) between Generator (solid) and Verifier (dashed). The ambiguity in answers is resolved through game-theoretic (not ad-hoc interaction, *i.e.*, debate) collaboration (consensus) between arbitrary LLMs based on correctness and confidence (not black-box generation).

their own outputs, offering a more scalable alternative to solely relying on human feedback (Bai et al., 2023; Saunders et al., 2022; Markov et al., 2023; Mu et al., 2024); (2) the flexibility to adjust game-theoretic objectives – such as utilities and policies between the generator and verifier – allows us to analyze latent decoding consistency and legibility as a function (Jacob et al., 2024; Hendrik Kirchner et al., 2024); and (3) in scenarios where human guidance is constrained, structured AI interactions can effectively elicit and refine latent knowledge, thereby enhancing model reliability, and generation consistency (Christiano et al., 2021; Turpin et al., 2024).

Focus of the paper. Realistically, neither models nor humans can be expected to be perfectly correct or reliable. Thus, our work focuses on achieving consistency through systematic verification and reliability through strategic interaction between LLMs calibrated on correctness and confidence measures. We design a multi-step **Bayesian Decoding Game** with complex action spaces that enable generators and verifiers to iteratively refine their strategies. Through the proposed no-regret optimization, our framework drives agents toward an equilibrium that ensures both consistency and reliability. Our framework addresses two types of outputs that are challenging for existing methods:

1. **Equilibrium-based Consistencies:** Outputs where strategic interactions converge to equilibrium states that systematically validate correctness and reliability.
2. **Game-emergent Inconsistencies:** Subtle flaws game-theoretically exposed through strategic agent interactions, surpassing human detection capabilities.

We formulate this verification process as a multi-step **Bayesian Decoding Game** with complex action spaces (Fig. 2). In this game-theoretic framework, generators and verifiers engage in strategic interactions: generators sample outputs based on latent model knowledge, while verifiers assess these outputs. To enhance the efficiency and reliability of this process, we improve upon traditional no-regret optimization through Markovian strategy updates and σ_i -separation constraints, enabling faster convergence to optimal equilibrium while maintaining clear separation between correct and incorrect outputs.

2 A BAYESIAN DECODING GAME (BDG)

2.1 PRELIMINARIES: MODELING LLM DECODING AS A SIGNALING GAME

To begin with, we define LLM decoding as a signaling game. The simplest form of a signaling game (Gibbons et al., 1992) can be described as follows: the generator receives a signal (**Correct** or **Incorrect**) and then takes a strategy (choose an answer implied by the signal from the candidate answer set) to transmit the signal information to the verifier. The verifier has to make a judgment (**Correct** or **Incorrect**) of the signal based on the strategy of the generator. If the judgment matches the signal, both the generator and verifier receive utility 1, and otherwise 0; in LLM decoding, the signaling game has been used to fine-tune the LLMs to output the best possible answer(s) under equilibrium. Equilibrium Consensus Game (ECG) proposed by (Jacob et al., 2024) is a pioneer-

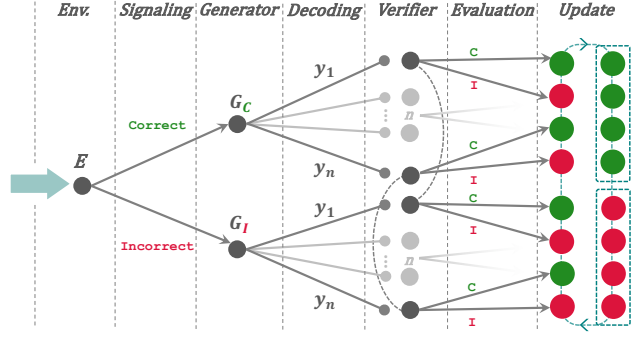


Figure 2: **Overview of the Bayesian Decoding Game under a signaling game structure.** Environment (Env) sends private signals to generator ($G_{C/I}$), who generates candidates ($y_1, \dots, y_n$) for information transmission. The verifier makes judgments (C/I) without observing original signals, enabling strategic decoding through Markovian Strategy Update (dashed) until equilibrium.

ing work on this problem, but like all existing consensus game frameworks, it fails to address a fundamental challenge: the **Collusion in a Nash Equilibrium**.

Theorem 1. *More than one (mixed) strategy ¹ Nash Equilibrium exists for this game.*

Definition 1. (Bonjour et al., 2022) *Collusion in a competitive multi-agent game occurs when two or more agents cooperate covertly to the disadvantage of others.*

Collusion in a Nash Equilibrium (NE). Thm. 1. is both a guarantee and a curse; the existence of an equilibrium ensures convergence, but the presence of multiple equilibria raises the risk of undesirable outcomes under collusion, where low-quality output may incorrectly align with successful verification. The proof and explanation are in Appx. D.1.

Example. In a signaling game, given the query “What is the capital of Switzerland?”, one **Collusive** Nash Equilibrium can be given by a **Correct** signal, generator chooses “Zurich”, verifier judges {**Correct**} signal, generator chooses “Bern”, verifier judges {**Incorrect**} which means that the verifier makes judgments only conditioning on the generator’s choice pattern rather than factual correctness. Under this equilibrium, the more plausible but incorrect answer (Zurich) is validated while the correct answer (Bern) is rejected.

Algorithmic collusion has been studied quite extensively in literature including (Xu & Zhao, 2024; Koirala & Laine, 2024; Sadoune et al., 2024). However, only requiring the game to converge to a Nash Equilibrium is not enough to avoid collusion, therefore BDG introduces a **Separating Equilibrium** for improved consistency.

2.2 AN OPTIMAL EQUILIBRIUM FOR DECODING GAME

Collusion Avoidance with Separating Equilibrium. To ensure that both the generator distinguishes between the correct and incorrect signal and the verifier verifies answers correctly, we designed the BDG and convergence algorithm to constrain the equilibrium to be Separating Equilibrium (SE).

Definition 2. (Separating Equilibrium (Black et al., 2012)) *A Separating Equilibrium (SE) is a type of Perfect Bayesian Equilibrium (PBE) Appx. C where agents with different types (signal) choose different strategies.*

Definition 3. (Decoding Game) *The Decoding Game is an alternative version of the signaling game in §2.1, and its payoff is determined by the preference ordering of each player, $O_i \in S_y$, $i \in \{G, V\}$,*

¹mixed strategies refer to a probability distribution over all strategies rather than committing to one strategy

where $|\mathcal{Y}|$ is the cardinality of the candidate set $\mathcal{Y}$ and $S_{\mathcal{Y}}$ is the set of all permutations of elements in $\mathcal{Y}$. We define the utility of the decoding game as

$$u_G(O_G, O_V) = u_V(O_G, O_V) = \mathbb{1}_{(O_G=O_V)} \quad (1)$$

such that $\mathbb{1}_{(O_G=O_V)}(\cdot, \cdot)$ is the indicator function at $O_G = O_V$. O_i is the preference relation indicated by players' strategy, $s_G(y \mid x, \text{correct}, b_S)$, $s_V(\text{correct} \mid x, y, b_V)$:

$$\begin{aligned} s_G(y_i \mid x, \text{correct}, b_S) \geq s_G(y_j \mid x, \text{correct}, b_S) &\iff y_i \succsim_G y_j \\ s_V(\text{correct} \mid x, y_i, b_V) \geq s_V(\text{correct} \mid x, y_j, b_V) &\iff y_i \succsim_V y_j \end{aligned} \quad (2)$$

$b_G = b_G(y \mid x, \text{correct})$ is the generator's belief of the probability of y being judged correctly by the verifier, and $b_V = b_V(\text{correct} \mid x, y)$ is the verifier's belief of the probability of y being associated with the correct environment signal received by the generator. s_G, s_V are the strategies for the generator and verifier, respectively.² With the preference relation, we determine O_G, O_V , and the equilibrium is reached when preference relations align.

To avoid collusion, we need to enforce an σ_i -**Separated** condition for the equilibrium, which is defined as

Definition 4. (σ_i -Separated Equilibrium (σ_i -SE)) For both the generator and verifier, given constants σ_G, σ_V , the generator's equilibrium strategy is said to be σ_G -separated, $i \in \{G, V\}$ if and only if $\min_{y_i \in \mathcal{Y}} \|s_G(y_i \mid x, \text{correct}, b_S) - s_G(y_i \mid x, \text{incorrect}, b_S)\| > \sigma_G$, whereas for the verifier, we have $\min_{y_i \in \mathcal{Y}} s_V(\text{correct} \mid x, y_i, b_S) - s_V(\text{incorrect} \mid x, y_i, b_S) > \sigma_V$

Example. The σ_G -separated constraint enforces that the generator's strategies for different signals must maintain an $L1$ distance of at least σ_G , meaning its output distributions are distinctly different when receiving correct versus incorrect signals. Similarly, σ_V -separation ensures that the verifier's judgment probabilities maintain a clear quantitative distinction of at least σ_V between different outputs.

2.3 BDG OPTIMIZATION: NO-REGRET OPTIMIZATION FOR EQUILIBRIUM

No-Regret Optimization. Based on the Decoding Game in §2.2, we propose two strategy update schedules to numerically achieve optimal convergence of σ_i -SE in Defi. 3., 4.. The multiplicity of SE leads to convergence to suboptimal outcomes, necessitating the definition of an initial strategy for each player. This "true" prior is denoted as $s_V^{(1)}(\cdot \mid x, y)$ and $s_G^{(1)}(\cdot \mid x, v)$ following (Jacob et al., 2024).

Through repeated interactions and iterative policy refinement, no-regret learning approximates equilibria in large games. Our cumulative regret is defined as:

$$\text{Reg}_i^{(T)} := \frac{1}{T} \left(\sum_{t=1}^T u_i(s_i^*, s_D^{(t)}; b_i) - u_i(s_i^{(t)}, s_D^{(t)}; b_i) \right), \quad (3)$$

where s_i^* is the optimal hindsight strategy that maximizes this value. Rather than computing regret at each iteration, s_i^* is selected based on the time-averaged strategies.

In sequential games with private information and discrete choices, global regret minimization is achieved by minimizing regret locally within each information set, given the finite nature of these sets. For example, to minimize overall regret, the generator must minimize regret by selecting an optimal mixed strategy s_G , conditioned on the signal correctness received from the environment. The verifier follows a similar procedure, updating its strategy with respect to each $y_i \in \mathcal{Y}$.

For this problem, the payoff is maximized when the generator and verifier align their strategies and minimize their confidence difference. Thus, the strategy update should be directed towards alignment with the opponent's strategy based on the adaptability³ of players (Roughgarden, 2010).

²The generator's strategy is a probability distribution over all candidates given the signal and the prompt, and the verifier's strategy is a probability distribution of correct and incorrectness given the prompt and the chosen candidate; there is a difference between **belief** and **strategy**: the **belief** is the player's belief in the opponent's strategy.

³An adaptive player is a function that inputs (1) the opponent i , (2) time t , (3) mixed strategies $s^1, \dots, s^t$ produced by i , and (4) past strategies $a^1, \dots, a^{t-1}$, and outputs a coupled strategy and belief.

Markovian Strategy Update. To maximize the utility given by Eq. 1 the players update their strategy based on the belief. Each player’s belief $b_{i,t}$ at time t of the opponent’s strategy is given by the opponent’s strategy in period $t - 1$. We hence propose a Markovian strategy update schedule:

$$b_G^{(t+1)}(y | x, v) = s_V^{(t)}(v | x, y), \quad b_V^{(t+1)}(v | x, y) = s_G^{(t)}(y | x, v) \quad (4)$$

$$s_G^{(t+1)}(y | x, v) \propto \exp \left\{ \frac{\frac{1}{2} b_G^{(t+1)}(y | x, v) + \lambda_G \log s_G^{(t)}(y | x, v, b_G^{(t)})}{1/(\eta_G t) + \lambda_G} \right\} \quad (5)$$

$$s_V^{(t+1)}(v | x, y) \propto \exp \left\{ \frac{\frac{1}{2} b_V^{(t+1)}(v | x, y) + \lambda_V \log s_V^{(t)}(v | x, y, b_V^{(t)})}{1/(\eta_V t) + \lambda_V} \right\} \quad (6)$$

Initial policies are $s_V^{(1)}(\cdot | x, y)$, $s_G^{(1)}(\cdot | x, v)$, where $\eta_i, \lambda_i, i \in \{G, V\}, \delta$ are the learning rate and stiffness hyperparameter and consistency bound. The two strategy update schedules we propose show satisfactory convergence properties, and the stopping criteria are given by:

1. Preference Alignment: $O_G = O_V$.
2. Consistency: $\|s_G(y | x, \text{correct}, b_S) - a_{NV}(\text{correct} | x, y, b_V)\| < \delta$.
3. Collusion Avoidance: satisfy σ_i -SE in Defi. 4.

Theorem 2. *A Markovian update schedule for a Decoding Game converges to an optimal σ_i -Separated Equilibrium.*

The proof can be found in Appx. D.2. Under BDG’s utility and the design of the no-regret algorithm, our method reaches σ_i -SE 30 times faster than the current state-of-the-art (Jacob et al., 2024) based on Average Recall update with an accurate correctness alignment between the generator and verifier. Table 1 and Appx. F illustrate the difference in game design between BDG and ECG.

2.4 BDG ANALYSIS: PROPERTIES AND BEHAVIOR

Equilibrium Properties. At σ_i -SE with signal distribution $\mathbf{P}(\text{correct}, \text{incorrect}) = (0.5, 0.5)$, we analyze the separation characteristics induced by our no-regret optimization. According to the environment, we label the $\frac{n}{2}$ most preferred candidates as correct, and the rest as incorrect. We denote the candidate in each group as $y_{i,C}, y_{i,I}$, respectively. For candidate set $\mathcal{Y}$ with $|\mathcal{Y}| = n$ where $n \bmod 2 = 0$, we characterize degree of separation between correct- and incorrect-ness by the separation score:

$$|s_V^{(t)}(\text{correct} | x, y_{\frac{n}{2}}) - s_V^{(t)}(\text{correct} | x, y_{\frac{n}{2}+1})|$$

This measure at $t = 1$ quantifies the verifier’s initial separation score between the least correct and least incorrect candidates under the prior. When this value is small (when the verifier is ambiguous about the correctness classification), the ambiguity is revealed and sorted through the preference fluctuation during the strategic interaction with the Markovian update, in contrast to the Average Recall update shown in Fig. 3 b (right corner). Our separating constraint enforces that the equilibrium separation score is bounded below by the same parameter as in Defi. 4. under a **rational** assumption, which is described by the proposition below

Proposition 1 *Under any signal distribution environment such that $\mathbf{P}(\text{correct}, \text{incorrect}) = (p, 1 - p)$ s.t. $p < 1$ and the rationality condition that **the equilibrium confidence scores is greater than $\frac{1}{2}$ for correct candidates and less than $\frac{1}{2}$ the incorrect candidates**, the separation score is also bounded below by the same parameter in Defi.4.*

$$s_V^*(\text{correct} | x, y_k) - s_V^*(\text{correct} | x, y_{k+1}) \geq \sigma_V \quad (7)$$

if and only if the σ_i -separated condition is enforced. k is the least correct candidate and $k + 1$ is the least incorrect candidate in equilibrium, determined by the candidate set cardinality and signal distribution.

Prop. 1 exemplifies how σ_i -separated condition (Defi.4.) ensure that, in a decision-making environment, correct and incorrect candidates can be properly segregated. Especially, our Markovian updates maintain this separation while ensuring convergence, as demonstrated below. However,

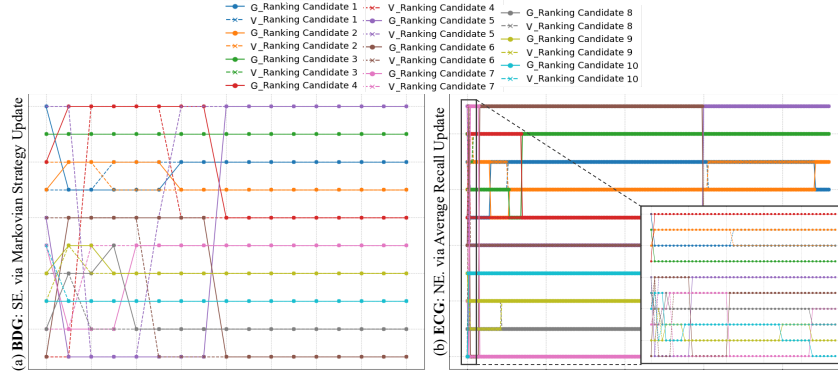


Figure 3: **Convergence dynamics comparison between Separating Equilibrium (SE) and Nash Equilibrium (NE).** We track generator (G, solid lines) and verifier (V, dashed lines) rankings for 10 answer candidates. (a) BDG’s Markovian update achieves rapid convergence to SE within 100 iterations, with clear separation in rankings and consistent alignment between G and V. (b) ECG’s Average-recall update (Jacob et al., 2024) converges to NE but exhibits persistent oscillations and ranking ambiguity.

there is no such guarantee based on the Average Recall update in Table. 1 and Nash equilibrium of Thm. 1., more details can be found in Appx. D.3. A comparison can be found in Fig. 3.

Reliable Behavior. The σ_i -SE in BDG prevents collusion through strategic separation which ensures reliable behaviors of LLMs and agents. In equilibrium, we examine both correctness alignment between the generator and verifier and collusion prevention:

Intuition. At equilibrium, reliable behavior emerges from two mechanisms: the strategic separation enforces a strict preference ordering that prevents collusion, while the reliability measure ensures this preference translates to an optimal balance between strategic consistency and behavioral reliability.

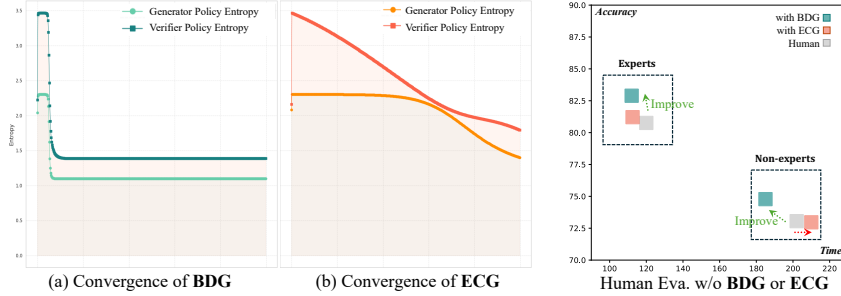


Figure 4: **Left: Policy entropy dynamics of BDG vs ECG.** (a) BDG exhibits initial exploration followed by rapid stabilization, demonstrating efficient convergence to separating equilibrium. (b) ECG shows continuous entropy decrease without stabilization, reflecting unstable agent interactions seen in Fig. 3. **Right: Performance Comparison.** Experts, Non-Experts, and Game-Theoretic Strategies (BDG and ECG) on time, accuracy. The evaluation is based on a user study (n=183) where participants classified LLM-generated math solutions under three conditions (baseline, BDG-guided, ECG-guided), with expertise levels determined by a 150s temporal threshold.

We analyze policy entropy dynamics between BDG and ECG to understand the equilibrium behaviorally. We evaluate convergence through policy entropy $H(\pi) = -\sum \pi(*) \log \pi(*)$ which measures agent strategy uncertainty. This metric captures both convergence efficiency (entropy reduction rate) and equilibrium stability (final entropy level) for generator and verifier policies. Fig. 4 shows how BDG achieves reliable separation: the entropy trajectories show rapid stabilization after initial exploration, validating our game-theoretic framework and theoretical guarantees.

3 EXPERIMENTS

Focus and Setting. We aim to answer the following questions: (1) What design choices enable decoding games to improve language generation performance? (2) To what extent does our BDG improve consistency? (3) To what extent does the BDG improve factual validity and reliability?

BDG focuses on improving the consistency and reliability of LLMs. However, consistency and reliability manifest themselves in various forms across different domains and dimensions, including correctness, truthfulness, factuality, valid reasoning, value alignment, among others. We first assess efficiency and reliability through a multidimensional comparison with another game-theoretic method (Jacob et al., 2024) and several variants. Then, we evaluate performance on a diverse set of LLMs used for real-world tasks: MMLU (Hendrycks et al., 2020b), ARC-Easy (E.), -Challenge (C.) (Clark et al., 2018), RACE-High (H.) (Lai et al., 2017). It is important to note that BDG is a game-theoretic decoding strategy and not a deliberation/training-based method like a prover-verifier-game (PVG) (Hendrik Kirchner et al., 2024), or contrastive-objective based generation (Li et al., 2022). Nevertheless, we demonstrate effectiveness through benchmarks in reasoning task: GSM8K (Cobbe et al., 2021), medical tasks: PubMedQA (Jin et al., 2019), MMLU-Medical (M.), and ethical scenarios, including justice, virtue, deontology and utilitarianism in Ethics (Hendrycks et al., 2020a), that BDG yields reliable improvements and demonstrates synergistic potential across various scenarios.

Action Space in the Game. To define the action space in BDG, the generator selects from a finite set of candidates $\mathcal{Y}$. For multiple-choice tasks, $\mathcal{Y}$ directly corresponds to the given options. For open-ended generative tasks, we construct $\mathcal{Y}$ by sampling candidates from the LLM’s distribution $P_{LLM}(y \mid q, \text{correct})$ using nucleus (Holtzman et al., 2019) and top-k (Fan et al., 2018) sampling methods. This standardized action space allows BDG and benchmarks to be applied consistently across different types of tasks while maintaining tractable strategy spaces.

Baselines and Models. For fair comparisons, following the setting and scores (Jacob et al., 2024), we use LLaMA models (Touvron et al., 2023) (7B, 13B parameters) with 16-bit inference across all experiments unless otherwise specified. On multiple-choice datasets, we employ: *Generative Ranking (G)*: Ranks candidates by $P_{LLM}(y \mid x, \text{correct})$ following (Brown et al., 2020; Touvron et al., 2023); *Discriminative Ranking (D)*: Re-weights query-candidate pairs using $\pi_D^{(1)}(\text{correct} \mid x, y)$ based on (Jacob et al., 2024); *Self-Contrastive Decoding (SCD)*: Utilizes $\pi_G^{(1)}$ for reweighting candidates (Jacob et al., 2024; Li et al., 2022); *Equilibrium Consensus Game (ECG)*: Average Recall update with Nash equilibrium discriminator (x, y) by $\pi_D^*(\text{correct} \mid x, y)$ (Jacob et al., 2024); BDG update query-candidate pairs based on Markovian Strategy with SE discriminator (x, y) by $\pi_D^*(\text{correct} \mid x, y)$.

Prompting. Unless otherwise specified, the condition for the P_{LLM} corresponds to the standard zero-shot prompt (Jacob et al., 2024; Hendrycks et al., 2020b). Furthermore, we combine chain-of-thought (CoT) (Wei et al., 2022), and few-shots setting (Wei et al., 2022) as orthogonal analysis.

3.1 GAME-THEORETIC DESIGN

Searching & Convergence Behavior. We have compared searching behaviors of **BDG** with the most closely related method, the **ECG** (Jacob et al., 2024), in the multiple-choice question answering (MCQA) task (Clark et al., 2018). Fig. 3 and 4 provide a visual case study. BDG demonstrates con-

Table 1: Comparison between ECG and BDG.

Criteria	ECG: Equilibrium Consensus Game	BDG: Bayesian Decoding Game	Thm.
Strategy	ER-update $x_{i,t+1} = x_{i,t} + \frac{1}{2t} \sum_0^t x_{-i,t}$	last-round belief update $b_{i,t} = a_{-i,t-1}$	2
Convergence	NE	SE	3
Update	Average Recall	Markovian	3
Complexity	$\mathcal{O}(n^2)$	$\mathcal{O}(n \log n)$	N/A

Table 2: Comparison of inconsistency (InC.%) and improvements (Imp.%) between Accuracies of G, ECG, and BDG.

Domain	Model	InC.%	G	ECG	Imp.%	BDG	Imp.%
MMLU	LLaMA-7B	69.0% ↓	30.4	39.9	31.3% ↑	40.5	33.2% ↑
	LLaMA-13B	60.6% ↓	41.7	45.1	8.1% ↑	46.9	12.5% ↑
ARC-E.	LLaMA-7B	56.1% ↓	68.2	71.5	4.8% ↑	75.3	10.4% ↑
	LLaMA-13B	46.1% ↓	71.2	76.4	7.3% ↑	78.1	9.7% ↑
ARC-C.	LLaMA-7B	65.9% ↓	47.3	58.3	23.2% ↑	59.6	26.0% ↑
	LLaMA-13B	59.1% ↓	51.9	61.4	18.3% ↑	62.2	19.8% ↑
RACE-H.	LLaMA-7B	62.0% ↓	46.4	56.4	21.5% ↑	57.7	24.4% ↑
	LLaMA-13B	58.8% ↓	47.9	62.8	31.1% ↑	60.3	25.9% ↑
Average		59.7% ↓	50.6	59.0	18.2% ↑	60.1	20.2% ↑

Table 3: Model consistency across different domains.

Domain	Model	G	MI	SCD	D	ECG	BDG
MMLU	LLaMA-7B	30.4	33.1	30.5	40.4	39.9	40.5
	LLaMA-13B	41.7	41.8	41.7	41.9	<u>45.1</u>	46.9
ARC-E.	LLaMA-7B	68.2	68.8	69.5	52.5	<u>71.5</u>	75.3
	LLaMA-13B	71.2	71.5	73.0	65.0	<u>76.4</u>	78.1
ARC-C.	LLaMA-7B	47.3	47.4	56.5	42.7	<u>58.3</u>	59.6
	LLaMA-13B	51.9	52.1	59.3	48.5	<u>61.4</u>	62.2
RACE-H.	LLaMA-7B	46.4	46.3	53.1	46.0	<u>56.4</u>	57.7
	LLaMA-13B	47.9	48.4	58.9	55.1	62.8	<u>60.3</u>

sistent and reliable convergence. Conversely, the ECG exhibits prolonged and inconsistent searching behavior. Despite continuous shifts in candidate selections, ECG fails to achieve stable convergence with persistent disagreement between the generator and verifier. Tab. 1 highlights the improved convergence properties of the BDG over the ECG.

Game-emergent Inconsistencies. We quantified the degree of inconsistency during the decoding stage by analyzing the disagreement percentage between Generative (G) and Discriminative Ranking (D) following (Jacob et al., 2024). The game-theoretic ECG and BDG reveal inherent model inconsistencies with a 59.7% disagreement rate between them. In Tab. 2, G and D often yield conflicting results, indicating significant inherent inconsistencies during the decoding stage of generative models. These discrepancies can be effectively mitigated by our approach, specifically during the decoding process, without the need for additional training. Tab. 2 shows that BDG consistently outperforms both G and ECG, particularly in cases with higher disagreement rates. We achieve superior consistency with higher correctness with fewer updates in each case Fig. 3.

Human vs. Game-Theoretic Detection We conducted a user study (n=183) evaluating mathematical assessment under three conditions: unassisted baseline, BDG-guided verification, and ECG-guided verification. Performance metrics included solution accuracy and completion time, with participants stratified into expert/non-expert groups based on empirically determined temporal threshold (150s).

Fig. 4 reveals significant performance disparities between experts and non-experts, quantitatively illustrating human evaluation limitations as generation complexity increases. Game-theoretic approaches, particularly **BDG**, enhance decoding effectively – without further free – maintaining accuracy while closely aligning with human intent. BDG consistently improves accuracy across non-experts and experts levels and significantly reduces sample identification time, outperforming unassisted baseline and ECG across multiple dimensions. This also suggest its effectiveness in bridging the expertise gap. Additional results about this finding are in provided in Appx. I.

3.2 CONSISTENCY BENCHMARKING: ACROSS DOMAINS WITH SMALLER MODELS

With “relatively easy” reasoning and comprehension tasks, we show superior performance compared to baselines and other game-theoretic methods in Tab. 3 due to the efficient alignment of consistency.

Table 4: The orthogonal enhancements with few shots.

Domain		Model	BDG	
			zero-shot	few-shot
Medical	PubMedQA	LLaMA-7B	71.45	71.89
		LLaMA-13B	74.00	74.47
	MMLU-M.	LLaMA-7B	51.35	52.90
		LLaMA-13B	56.01	58.85
Ethics	Justice	LLaMA-13B	52.27	53.15
	Virtue	LLaMA-13B	33.10	33.82
	Deontology	LLaMA-13B	52.41	53.01
	Utilitarianism	LLaMA-13B	65.35	66.75

Table 5: The reliability across different domains with CoT.

Domain	Model	Decoding Methods				Game-theoretic	
		Greedy	MI	SCD	D	ECG	BDG
GSM8K	LLaMA-7B	10.8	14.7	13.4	15.0	15.1	15.8
	LLaMA-13B	14.9	22.5	23.1	22.5	23.0	22.7
TruthfulQA	LLaMA-7B	33.41	34.79	34.91	34.17	34.27	35.07
	LLaMA-13B	33.05	36.30	34.61	39.05	38.63	40.01

In a broader comparison, our zero-shot LLaMA-13B (78.1, ARC-E.) outperforms larger models, PaLM-540B model (76.6) (Chowdhery et al., 2023).

With more challenging reasoning and multitask understanding tasks, such as ARC-C, RACE-H, and MMLU, we achieve the best equilibrium decoding with fewer rounds and higher accuracy. Our LLaMA-13B (46.9, MMLU; 57.7, RACE-H.) outperforms zero-shot GPT-3-175B (37.7, MMLU) (Hendrycks et al., 2020b), LLaMA-65B (51.6, RACE-H.) (Touvron et al., 2023), and PaLM-540B (49.1, RACE-H.) (Hendrycks et al., 2020b).

3.3 ORTHOGONAL ENHANCEMENTS FOR ROBUST DECODING

Datasets in Tab. 4, 5 involve challenging scenarios to test models’ reasoning abilities. We use these benchmarks to study whether we can combine our approach with various orthogonal strategies. Based on game theory, BDG does not conflict with the computationally intensive game mechanism during training, nor does it conflict with CoT and few-shot variations. BDG shows enhanced performance in more challenging scenarios in Table 5, establishing a highly novel direction in decoding research. Furthermore, it achieves broader accuracy and robustness across datasets, underscoring its adaptability and trustworthiness.

4 DISCUSSION

Game Design over ECG and PVG. BDG and ECG share the common goal of aligning generative models with human intentions to improve output reliability, yet they differ significantly in their game design, achieving substantial gains with reduced computational overhead. While ECG utilizes moving-average updates to foster consensus, often leading to unstable and fluctuating equilibria, BDG employs a structured Bayesian framework that drives interactions toward an optimal equilibrium with greater stability. In contrast, Prover-Verifier Games (PVGs) (Hendrik Kirchner et al., 2024), which contribute to ChatGPT o1 (OpenAI, 2024), use a RL-based alignment and focus on adversarial training phases featured by RL and competitive dynamics. This requires intensive training and causes potential deviations from cooperative strategies. Appx. E and F explore the distinct phases and transitions between these frameworks, highlighting BDG’s scalability and its departure from the training-intensive PVG.

Robustness and Integrative Potential. BDG achieves consistent performance improvements across diverse domains, maintaining effectiveness even with lower-quality initial LLM outputs. The

framework readily integrates with existing techniques such as self-consistency and chain-of-thought prompting, while offering fast equilibrium convergence and reliable verification.

Balancing Correctness and Reliability. Reliability (Rastogi et al., 2023) tries to give an account of the prover model’s failure modes and sense-making, whether the reasoning is correct or not. The resulting decoding can be arbitrarily complex (Nanda et al., 2023). In contrast, correctness allows to verify if a given solution is correct, ignoring how the generator reasoned it to be reliable (consistent with the environment). Consequently, reliability requires model outputs that are coherent and consistent to human understanding (Mökander et al., 2023). We show that it is possible to have both, without sacrificing correctness for reliability (Hendrik Kirchner et al., 2024), and especially in high-stakes settings reliability is as important as correctness (Casper et al., 2024).

Limitation. One potential limitation arises from the explicit specification of correctness consistency branches during the game process, as this alignment is primarily intended to match human intent with model outputs, similar to game-based approaches (Jacob et al., 2024; Hendrik Kirchner et al., 2024). Adding multi-metrics and multiple agents to achieve game-based deliberation is possible.

5 RELATED WORK

Multi-Agent Debate Systems. Previous work has explored mechanisms where multiple language model instances “debate” to refine and converge to a final answer (Du et al., 2023; Chen et al., 2023; Khan et al., 2024; Kori et al., 2022). It is possible to categorize our method as a major variant of this multi-agent debate in which the interaction occurs within a game-theoretic framework, rather than directly within the language models’ outputs. This structured signaling game enables BDG to enhance the correctness and reliability of outputs without relying on human feedback, by dynamically optimizing the generation and verification processes. Additionally, this approach can resolve ambiguity, confusion, and low accuracy caused by inconsistencies, but not by poor reasoning.

Signaling Game. Conventional signaling game settings have been successfully deployed for Poker (Brown & Sandholm, 2018; 2019), Stratego (Perolat et al., 2022), Diplomacy (FAIR; Bakhtin et al., 2022; Jacob et al., 2022), and LLM tasks (Hendrik Kirchner et al., 2024; Chen et al., 2023). Building on these insights, we propose a novel signaling game framework between a generator and verifier for systematic LLM output verification.

Decoding Strategies. Top-k sampling (Fan et al., 2018), nucleus sampling (Holtzman et al., 2019), and typical sampling (Meister et al., 2023) focus on generating high-confidence text but do not address the correctness of the outputs. Candidates were generated using these methods. Equilibrium-ranking (Jacob et al., 2024) applies an average-moving strategy to the initial distribution. In contrast, BDG integrates a multistage signaling game that inherently balances correctness and consistency during the generation process. BDG can be seamlessly combined with these strategies to enhance the reliability and reliability of generated text.

Ranking Techniques. Ranking is a widely used approach to select the correct output from a set of candidates generated by language models. (Thoppilan et al., 2022) use additional human annotations to train a ranking model for response filtering. (Hendrik Kirchner et al., 2024) trains different provers and verifiers for increasing output legibility. Although our work also utilizes existing language models as discriminators, *BDG* eliminates the need for additional training and does not impose specific assumptions on the structure of either the generator or discriminator.

6 CONCLUSION

BDG is a game-theoretic framework that enhances both the consistency and reliability of LLMs. By framing the decoding process as a multistage signaling game between a generator and verifier, BDG efficiently aligns model outputs with human intent while mitigating the trade-off between correctness and reliability. Our approach achieves superior performance across benchmarks, often surpassing larger models, and demonstrates its adaptability when combined with existing techniques like chain-of-thought prompting. BDG ensures reliable and robust LLM outputs, offering a scalable, training-free solution to the challenges of ambiguity and inconsistency in generative models.

ACKNOWLEDGMENT

This work was supported by the JADS programme and UK Research and Innovation [UKRI Centre for Doctoral Training in AI for Healthcare grant number EP/S023283/1]. HPC resources were provided by the Erlangen National High Performance Computing Center (NHR@FAU) of the Friedrich-Alexander-Universität Erlangen-Nürnberg (FAU) under the NHR project b143dc and b180dc. NHR funding is provided by federal and Bavarian state authorities. NHR@FAU hardware is partially funded by the German Research Foundation (DFG) – 440719683. Support was also received by the ERC - projects MIA-NORMAL 101083647 as well as DFG 513220538, 512819079.

IMPACT STATEMENT

With the improvement of generation quality, one can imagine more potent disinformation (e.g., automatic generation of fake news) that may be hard to distinguish from human-authored content. It might be worthwhile to augment current decoding techniques so that the generated outputs will also be watermarked without compromising their quality.

REFERENCES

- Yuntao Bai et al. Rlaif: Reinforcement learning from ai feedback. [arXiv preprint arXiv:2304.03442](#), 2023.
- Zechen Bai, Pichao Wang, Tianjun Xiao, Tong He, Zongbo Han, Zheng Zhang, and Mike Zheng Shou. Hallucination of multimodal large language models: A survey. [arXiv preprint arXiv:2404.18930](#), 2024.
- Anton Bakhtin, David J Wu, Adam Lerer, Jonathan Gray, Athul Paul Jacob, Gabriele Farina, Alexander H Miller, and Noam Brown. Mastering the game of no-press diplomacy via human-regularized reinforcement learning and planning. [arXiv preprint arXiv:2210.05492](#), 2022.
- Sourav Banerjee, Ayushi Agarwal, and Saloni Singla. Llms will always hallucinate, and we need to live with this. [arXiv preprint arXiv:2409.05746v1](#), 2024.
- John Black, Nigar Hashimzade, and Gareth Myles. [A dictionary of economics](#). Oxford University Press, USA, 2012.
- Trevor Bonjour, Vaneet Aggarwal, and Bharat Bhargava. Information theoretic approach to detect collusion in multi-agent games. In [Uncertainty in Artificial Intelligence](#), pp. 223–232. PMLR, 2022.
- Noam Brown and Tuomas Sandholm. Superhuman ai for heads-up no-limit poker: Libratus beats top professionals. [Science](#), 359(6374):418–424, 2018.
- Noam Brown and Tuomas Sandholm. Superhuman ai for multiplayer poker. [Science](#), 365(6456):885–890, 2019.
- Tom B Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. In [Advances in Neural Information Processing Systems \(NeurIPS\)](#), pp. 1877–1901, 2020.
- Stephen Casper, Xander Davies, Claudia Shi, Thomas Krendl Gilbert, Jérémy Scheurer, Javier Rando, Rachel Freedman, Tomasz Korbak, David Lindner, Pedro Freire, et al. Open problems and fundamental limitations of reinforcement learning from human feedback. [arXiv preprint arXiv:2307.15217](#), 2023.
- Stephen Casper, Carson Ezell, Charlotte Siegmann, Noam Kolt, Taylor Lynn Curtis, Benjamin Bucknall, Andreas Haupt, Kevin Wei, Jérémy Scheurer, Marius Hobbhahn, et al. Black-box access is insufficient for rigorous ai audits. In [The 2024 ACM Conference on Fairness, Accountability, and Transparency](#), pp. 2254–2272, 2024.

-
- Justin Chih-Yao Chen, Swarnadeep Saha, and Mohit Bansal. Reconcile: Round-table conference improves reasoning via consensus among diverse llms. [arXiv preprint arXiv:2309.13007](#), 2023.
- In-Koo Cho and David M Kreps. Signaling games and stable equilibria. [The Quarterly Journal of Economics](#), 102(2):179–221, 1987.
- Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. [Journal of Machine Learning Research](#), 24(240): 1–113, 2023.
- Paul Christiano, Buck Shlegeris, and Dario Amodei. Supervising strong learners by amplifying weak experts. [arXiv preprint arXiv:1810.08575](#), 2018.
- Paul Christiano, Ajeya Cotra, and Mark Xu. Eliciting latent knowledge: How to tell if your eyes deceive you. [Google Docs](#), December, 2021.
- Paul F Christiano, Jan Leike, Tom Brown, Miljan Martic, Shane Legg, and Dario Amodei. Deep reinforcement learning from human preferences. [Advances in neural information processing systems](#), 30, 2017.
- Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick, and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning challenge. [arXiv preprint arXiv:1803.05457](#), 2018.
- Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. [arXiv preprint arXiv:2110.14168](#), 2021.
- Yilun Du, Shuang Li, Antonio Torralba, Joshua B Tenenbaum, and Igor Mordatch. Improving factuality and reasoning in language models through multiagent debate. [arXiv preprint arXiv:2305.14325](#), 2023.
- Meta Fundamental AI Research Diplomacy Team (FAIR)[†], Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, et al. Human-level play in the game of diplomacy by combining language models with strategic reasoning. [Science](#), 378(6624):1067–1074, 2022.
- Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. [arXiv preprint arXiv:1805.04833](#), 2018.
- Drew Fudenberg. [Game Theory](#). MIT press, 1991.
- Tianyu Gao, Xingcheng Yao, and Danqi Chen. Simcse: Simple contrastive learning of sentence embeddings. [arXiv preprint arXiv:2104.08821](#), 2021.
- Robert Gibbons et al. A primer in game theory. 1992.
- Jan Hendrik Kirchner, Yining Chen, Harri Edwards, Jan Leike, Nat McAleese, and Yuri Burda. Prover-verifier games improve legibility of llm outputs. [arXiv e-prints](#), pp. arXiv–2407, 2024.
- Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. [arXiv preprint arXiv:2008.02275](#), 2020a.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. [arXiv preprint arXiv:2009.03300](#), 2020b.
- Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. [arXiv preprint arXiv:1904.09751](#), 2019.
- Athul Paul Jacob, David J Wu, Gabriele Farina, Adam Lerer, Hengyuan Hu, Anton Bakhtin, Jacob Andreas, and Noam Brown. Modeling strong and human-like gameplay with kl-regularized search. In [International Conference on Machine Learning](#), pp. 9695–9728. PMLR, 2022.

-
- Athul Paul Jacob, Yikang Shen, Gabriele Farina, and Jacob Andreas. The consensus game: Language model generation via equilibrium search. In The Twelfth International Conference on Learning Representations, 2024.
- Qiao Jin, Bhuwan Dhingra, Zhengping Liu, William W Cohen, and Xinghua Lu. Pubmedqa: A dataset for biomedical research question answering. arXiv preprint arXiv:1909.06146, 2019.
- Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. Debating with more persuasive llms leads to more truthful answers. arXiv preprint arXiv:2402.06782, 2024.
- Pravesh Koirala and Forrest Laine. Algorithmic collusion in a two-sided market: A rideshare example. arXiv preprint arXiv:2405.02835, 2024.
- Avinash Kori, Ben Glocker, and Francesca Toni. Explaining image classification with visual debates. arXiv preprint arXiv:2210.09015, 2022.
- Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. Race: Large-scale reading comprehension dataset from examinations. arXiv preprint arXiv:1704.04683, 2017.
- Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. Scalable agent alignment via reward modeling: a research direction. arXiv preprint arXiv:1811.07871, 2018.
- Jiwei Li and Dan Jurafsky. Mutual information and diverse decoding improve neural machine translation. arXiv preprint arXiv:1601.00372, 2016.
- Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. arXiv preprint arXiv:2210.15097, 2022.
- Todor Markov, Chong Zhang, Sandhini Agarwal, Florentine Eloundou Nekoul, Theodore Lee, Steven Adler, Angela Jiang, and Lilian Weng. A holistic approach to undesired content detection in the real world. In Proceedings of the AAAI Conference on Artificial Intelligence, volume 37, pp. 15009–15018, 2023.
- Ian R McKenzie, Alexander Lyzhov, Michael Pieler, Alicia Parrish, Aaron Mueller, Ameya Prabhu, Euan McLean, Aaron Kirtland, Alexis Ross, Alisa Liu, et al. Inverse scaling: When bigger isn’t better. arXiv preprint arXiv:2306.09479, 2023.
- Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally typical sampling. Transactions of the Association for Computational Linguistics, 11:102–121, 2023.
- Jakob Mökander, Jonas Schuett, Hannah Rose Kirk, and Luciano Floridi. Auditing large language models: a three-layered approach. AI and Ethics, pp. 1–31, 2023.
- Tong Mu, Alec Helyar, Johannes Heidecke, Joshua Achiam, Andrea Vallone, Ian Kivlichan, Molly Lin, Alex Beutel, John Schulman, and Lilian Weng. Rule based rewards for language model safety, 2024.
- Neel Nanda, Lawrence Chan, Tom Lieberum, Jess Smith, and Jacob Steinhardt. Progress measures for grokking via mechanistic interpretability. arXiv preprint arXiv:2301.05217, 2023.
- OpenAI. ChatGPT-4: Version Opus-1, 2024. URL <https://openai.com/gpt-4>.
- Julien Perolat, Bart De Vylder, Daniel Hennes, Eugene Tarassov, Florian Strub, Vincent de Boer, Paul Muller, Jerome T Connor, Neil Burch, Thomas Anthony, et al. Mastering the game of stratego with model-free multiagent reinforcement learning. Science, 378(6623):990–996, 2022.
- Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. Advances in Neural Information Processing Systems, 34:4816–4828, 2021.

-
- Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. arXiv preprint arXiv:2103.00020, 2021.
- Charvi Rastogi, Marco Tulio Ribeiro, Nicholas King, Harsha Nori, and Saleema Amershi. Supporting human-ai collaboration in auditing llms with llms. In Proceedings of the 2023 AAI/ACM Conference on AI, Ethics, and Society, pp. 913–926, 2023.
- Tim Roughgarden. Algorithmic game theory. Communications of the ACM, 53(7):78–86, 2010.
- Igor Sadoune, Marcelin Joanis, and Andrea Lodi. Algorithmic collusion and the minimum price markov game. arXiv preprint arXiv:2407.03521, 2024.
- William Saunders, Catherine Yeh, Jeff Wu, Steven Bills, Long Ouyang, Jonathan Ward, and Jan Leike. Self-critiquing models for assisting human evaluators. arXiv preprint arXiv:2206.05802, 2022.
- Chandan Singh, Jeevana Priya Inala, Michel Galley, Rich Caruana, and Jianfeng Gao. Rethinking interpretability in the era of large language models. arXiv preprint arXiv:2402.01761, 2024.
- Yixuan Su, Tian Lan, Yan Wang, Dani Yogatama, Lingpeng Kong, and Nigel Collier. A contrastive framework for neural text generation. Advances in Neural Information Processing Systems, 35: 21548–21561, 2022.
- Romal Thoppilan, Daniel De Freitas, Jamie Hall, Noam Shazeer, Apoorv Kulshreshtha, Heng-Tze Cheng, Alicia Jin, Taylor Bos, Leslie Baker, Yu Du, et al. Lamda: Language models for dialog applications. arXiv preprint arXiv:2201.08239, 2022.
- Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. arXiv preprint arXiv:2302.13971, 2023.
- Miles Turpin, Julian Michael, Ethan Perez, and Samuel Bowman. Language models don’t always say what they think: unfaithful explanations in chain-of-thought prompting. Advances in Neural Information Processing Systems, 36, 2024.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. Chain-of-thought prompting elicits reasoning in large language models. Advances in neural information processing systems, 35:24824–24837, 2022.
- Zhang Xu and Wei Zhao. On mechanism underlying algorithmic collusion. arXiv preprint arXiv:2409.01147, 2024.

Appendix Contents

- A.** Reproducibility Statement
- B.** Potential Ethics Risks and Societal Impact
- C.** Game-theoretic Formulation Supplementary
- D.** Proofs of Theorems
 - B.1** Proof of Thm. 1
 - B.2** Proof of Thm. 2
 - B.3** Proof of Prop. 1
- E.** From Bayesian Decoding Game (BDG) to Prover-Verifier Game (PVG)
- F.** From Bayesian Decoding Game (BDG) to Equilibrium Consensus Game (ECG)
- G.** Experiment Details
- H.** Searching & Convergence Behavior Supplementary
- I.** Human Evaluation

A REPRODUCIBILITY STATEMENT

We conducted our evaluations using widely recognized benchmarks such as ARC-Easy, ARC-Challenge, MMLU, and RACE. The experiments were performed using the open-source LLaMA 7B and 13B models. Key aspects of the game, including update policies and initial strategies, are thoroughly detailed in both the main text and appendix to facilitate accurate replication of the results. All experiments were conducted on NVIDIA A6000 and A100 GPUs, with runtimes ranging from 0.5 to 6 hours depending on the model size, task, and experimental settings. Further details on the game-theoretic mechanisms and specific design choices can be found in the methods section and the appendix.

B POTENTIAL ETHICS RISKS AND SOCIETAL IMPACT

Bayesian Decoding Game (BDG) is a novel game-theoretic framework that significantly enhances both the consistency and reliability of large language model outputs. By framing the decoding process as a multistage signaling game between a generator and verifier, BDG efficiently aligns model outputs with human intent while mitigating the trade-off between correctness and reliability. BDG ensures reliable and robust LLM outputs, offering a scalable, training-free solution to the challenges of ambiguity and inconsistency in generative models.

With the improvement of generation quality, one can imagine more potent disinformation (e.g., automatic generation of fake news) that may be hard to distinguish from human-authored content. It might be worthwhile to augment current decoding techniques so that the generated outputs will also be watermarked without compromising their quality. More potential ethics risks and societal impact can be seen from Fig. 5.

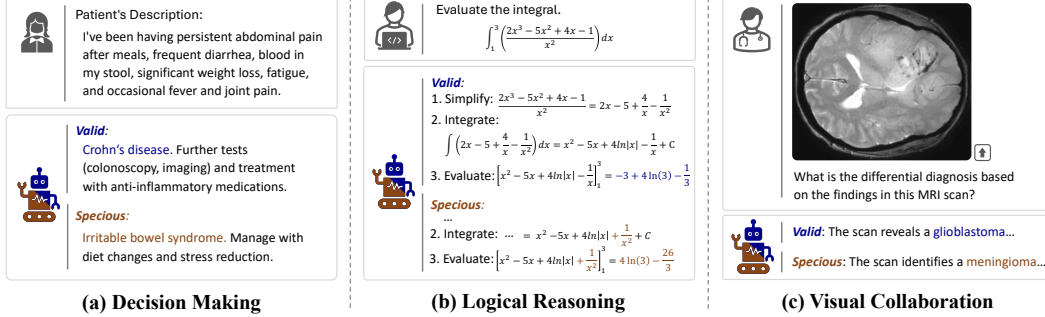


Figure 5: **Distinguishing different type of LLM outputs, particularly when human evaluation may overlook plausible errors.** The three panels demonstrate how models can generate both accurate and reliable, and plausible but misleading responses.

C GAME-THEORETIC FORMULATION SUPPLEMENTARY

A generative language model (LM) maps input x to output y according to some distribution $P_{\text{LM}}(y | x)$. Here, we do not impose restrictions on the form of input or output, as illustrated in Fig. 1, 2, 5. Instead, we address a multi-faceted problem involving a question x and a set of answer candidates $\mathcal{Y}$, generated by pre-trained language models on specific tasks. In the first stage, using this candidate set, we leverage generative LMs in two distinct ways:

Generatively, by supplying as input

1. a prompt x ,
2. the set of candidates $\mathcal{Y}$, and

3. a natural language prompt indicating that a correct or incorrect answer is desired. The LM may be thought of as modeling a distribution $P_{\text{LM}}(y \mid x, \text{incorrect})$, where the token incorrect denotes the fact that the model was prompted to generate an incorrect answer.

Verifiably, by supplying as input

1. the same x and
2. a possible candidate answer $y \in \mathcal{Y}$, together with
3. a prompt indicating that a correctness assessment $v \in \{\text{correct}, \text{incorrect}\}$ is sought. In this case, the language model acts as a models a distribution $P_{\text{LM}}(v \mid x, y)$ where $v \in \{\text{correct}, \text{incorrect}\}$.

The essence of a signaling game (Gibbons et al., 1992) is that one player (the generator) takes an strategy, the signal, to convey information to another player (the verifier); in the simplest setup, the final payoff depends on whether the verifier correctly judges the generator’s type based on the generator’s signal. Based on this intuition from game theory, (Jacob et al., 2024) design a Equilibrium Consensus Game (ECG), without a formal definition of the game. Thus, we firstly provide a comprehensive game-theoretic formulation for generative model decoding, and propose improvements to address limitations.

Formally, the signaling game’s components can be defined as follows:

1. *Players*: Generator and Verifier;
2. *Choice sets*: Generator’s choice set is $y \in \mathcal{C}_G = \mathcal{Y}$, with prompt p randomly drawn from $\{\text{Correct}, \text{Incorrect}\}$, and the Verifier’s choice set is $v \in \mathcal{C}_V = \{\text{Correct}, \text{Incorrect}\}$, based on the generator’s choice $y \in \mathcal{Y}$;
3. *Payoff Function*: $u_G = u_V = \mathbb{1}_{p=v}(p, v)$, where $\mathbb{1}$ equals 1 if the correctness prompt x matches the verification result, and 0 otherwise.

We are now ready to state the fundamental concept of this signaling game, a Perfect Bayesian Nash Equilibrium (PBNE) (Cho & Kreps, 1987). We use the short form Perfect Bayesian Equilibrium (PBE) with the auxiliary definitions Defi. 5. and 6. for PBE Definition.

Definition (Perfect Bayesian Equilibrium (Fudenberg, 1991)) A Perfect Bayesian Nash Equilibrium (PBE) is a pair (s, b) of strategy profile and a set of beliefs such that

1. s is **sequentially rational** given beliefs b , and
2. b is **consistent** with s .

Example 1. For generative model decoding, the generator’s belief is given by its perceived probability distribution, $\mathbb{P}(\{\text{correct}, \text{incorrect}\}) = (p_i, 1 - p_i)$, for each $y_i \in \mathcal{Y}$ of the verifier’s judgment, and with its belief and type, the generator chooses a mixed strategy that maximizes its utility, i.e., if the generator’s type is **correct**, then its optimal mixed strategy would be allocating positive possibility only on y_i such that $p_i > 1 - p_i$ and zero possibility to other y_i .

Definition 5. (Sequential Rationality)

A player is said to be sequentially rational iff, at each information set he is to move, he maximizes his expected utility given his beliefs at the information set (and given that he is at the information set) - even if this information set is precluded by his own strategy.

Definition 6. (Consistency on Path)

Given any (possibly mixed) strategy profile s , an information set is said to be on the path of play if and only if the information set is reached with positive probability according to s . Given any strategy profile s and any information set I on the path of play of s , the beliefs of a player at I are said to be consistent with s if and only if his beliefs are derived using the Bayes rule and s .

D PROOFS OF THEOREMS

D.1 PROOF OF THEOREM 1

Theorem 1. More than one (mixed) strategy Nash Equilibrium exists for this game.

Proof of Theorem 1.:

Suppose that the candidate set has 2 options (can be extended to any cardinality $|\mathcal{Y}|$), y_1, y_2 , one equilibrium can be described as: If the environment sends correct/incorrect, the generator generates the probability distribution $(1, 0)/(0, 1)$ for (y_1, y_2) given his belief that verifier probabilistic judgment, $\{\text{correct, incorrect}\}$, for y_1, y_2 is $(1, 0), (0, 1)$.

For the verifier, he believes that if the environment chooses correct/incorrect, then he believes that of generator's probabilistic generation for (y_1, y_2) are $(1, 0), (0, 1)$, therefore the verifier's best response is given by $(\text{correct, incorrect}) = (1, 0)$ if sees y_1 , $(\text{correct, incorrect}) = (0, 1)$ if sees y_2 . The (strategy and belief) for the generator and verifier above constitute one PBE for our game. For another equilibrium, we can revert every 0s and 1s in the above strategy profile, for all the strategies and the beliefs.

D.2 PROOF OF THEOREM 2

Theorem 2. The Markovian update schedule for our Decoding Game will converge to an equilibrium.

Proof of Theorem 2.:

We will show that the Markovian update schedule is in fact no-regret (thus guarantees CCE-convergence) for correct generator, and when generator receives incorrect signal, she will automatically perform the reversed strategy; then, if the Markovian update schedule converges to CCE for the incorrect signal, it automatically satisfies that the Markovian schedule will converge to a Bayes-CCE of our Decoding Game.

Definition 7. A randomized strategy profile $\mathbf{s} \in \Delta(\Sigma)$ is a coarse-correlated Bayesian equilibrium if for every $a'_i \in A_i$ and for every $v_i \in \mathcal{V}_i$:

$$\mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{v}} [U_i(\mathbf{s}(\mathbf{v}); \mathbf{v}_i) \mid \mathbf{v}_i = v_i] \geq \mathbb{E}_{\mathbf{s}} \mathbb{E}_{\mathbf{v}} [U_i(a'_i, \mathbf{s}_{-i}(\mathbf{v}_{-i}); \mathbf{v}_i) \mid \mathbf{v}_i = v_i]$$

We will first prove that the Markovian update schedule is asymptotically no-regret. For the generator, suppose that at time t , the chosen strategy is $a_G^{(t)}(y \mid x, \text{correct})$, and the optimal hindsight strategy that maximize U is given by $a_{NV}^{(t)}(\text{correct} \mid x, y) \forall y$, which is the normalized verifier's strategy on each candidate $y \in \mathcal{Y}$, and our update schedule

$$s_G^{(t+1)}(y \mid x, v) \propto \exp \left\{ \frac{\frac{1}{2} b_G^{(t+1)}(y \mid x, v) + \lambda_G \log s_G^{(t)}(y \mid x, v, b_G^{(t)})}{1/(\eta_G t) + \lambda_G} \right\}$$

such that

$$b_G^{(t+1)}(y \mid x, v) = s_V^{(t)}(v \mid x, y)$$

therefore, the regret at time t is given by

$$\|a_G^{(t)}(\mathbf{y} \mid x, \text{correct}) - a_{NV}^{(t)}(\text{correct} \mid x, \mathbf{y})\|$$

and in time $t + 1$, we have that the generator is **at least** $\frac{1}{2}\eta_G$ closer to the verifier's strategy, and the verifier is also **at least** $\frac{1}{2}\eta_V$ closer to the generator's strategy. Thus we have that

$$\begin{aligned} & \|a_G^{(t+1)}(\mathbf{y} \mid x, \text{correct}) - a_{NV}^{(t+1)}(\text{correct} \mid x, \mathbf{y})\| \leq \\ & \left(1 - \frac{1}{2}(\eta_G + \eta_V)\right) \|a_G^{(t)}(\mathbf{y} \mid x, \text{correct}) - a_{NV}^{(t)}(\text{correct} \mid x, \mathbf{y})\| \end{aligned}$$

and by construction we have $1 - \frac{1}{2}(\eta_G + \eta_V) < 1$, then asymptotically we can obtain that

$$\lim_{t \rightarrow \infty} \|a_G^{(t)}(\mathbf{y} \mid x, \text{correct}) - a_{NV}^{(t)}(\text{correct} \mid x, \mathbf{y})\| \rightarrow 0$$

thus the cumulative average regret also goes to 0 asymptotically. Therefore, the Markovian update schedule is no-regret and thus will converge to a Bayes-CCE of this game; moreover, with our stopping criteria satisfying Defi.4., the σ_i -separated condition is automatically satisfied with σ_i of our choice.

D.3 PROOF OF PROPOSITION 1

Proposition 1 Under any signal distribution environment such that $\mathbf{P}(\text{correct}, \text{incorrect}) = (p, 1 - p)$ s.t. $p < 1$, if the equilibrium confidence scores is conditioned that the correct candidates is greater than $\frac{1}{2}$ and the incorrect candidates is less than $\frac{1}{2}$, then the separation score is also bounded below by the same parameter in Defi.4.

$$s_V^*(\text{correct} \mid x, y_k) - s_V^*(\text{correct} \mid x, y_{k+1}) \geq \sigma_V \quad (8)$$

if and only if the σ_i -separated condition is satisfied. where k is the least correct candidate and $k + 1$ is the least incorrect candidate in equilibrium, determined by the candidate set cardinality and signal distribution.

Proof of Proposition 1: we first show that the σ_i -separatedness implies separation score bound.

for any given environment such that the signal distribution is given by $\mathbf{P}(\text{correct}, \text{incorrect}) = \mathbf{P}(p, 1 - p)$ this proof applies, for simplicity, we will provide the proof only for $\mathbf{P}(p, 1 - p) = (0.5, 0.5)$, the only difference will be the index of the least correct and least incorrect candidate. According to Defi. 4. the σ_i -separated condition, in equilibrium, the inequality below is satisfied

$$|s_V^*(\text{correct} \mid x, y_{\frac{n}{2}}) - s_V^*(\text{incorrect} \mid x, y_{\frac{n}{2}})| > \sigma_V$$

and

$$s_V^*(\text{correct} \mid x, y_{\frac{n}{2}}) = 1 - s_V^*(\text{incorrect} \mid x, y_{\frac{n}{2}})$$

thus the inequality becomes

$$|2 \cdot s_V^*(\text{correct} \mid x, y_{\frac{n}{2}}) - 1| > \sigma_V$$

we condition that **the equilibrium confidence for correct candidate being greater than $\frac{1}{2}$ and the incorrect candidate being less than $\frac{1}{2}$** , thus we can remove the absolute value and get $2 \cdot s_V^*(\text{correct} \mid x, y_{\frac{n}{2}}) - 1 > 0$, moreover, for $s_V^*(\text{correct} \mid x, y_{\frac{n}{2}+1})$, we have

$$1 - 2 \cdot s_V^*(\text{correct} \mid x, y_{\frac{n}{2}+1}) > \sigma_V$$

adding the two inequalities together, divided by 2, we get

$$s_V^*(\text{correct} \mid x, y_{\frac{n}{2}}) - s_V^*(\text{correct} \mid x, y_{\frac{n}{2}+1}) > \sigma_V$$

Then, we prove that reverse by contradiction. if the σ_i -separated condition is not satisfied as a stopping criterion, then the separation score can be bounded above by some constant less than σ_V .

We first assume that for all candidates, the correct and incorrect confidence score is bounded above uniformly by some σ' such that $\sigma' < \sigma_V$, which is given by

$$|s_V^*(\text{correct} \mid x, y_i) - s_V^*(\text{incorrect} \mid x, y_i)| < \sigma'$$

with the boldfaced rationality condition, we have that

$$2 \cdot s_V^*(\text{correct} \mid x, y_{\frac{n}{2}}) - 1 < \sigma' \quad 1 - 2 \cdot s_V^*(\text{correct} \mid x, y_{\frac{n}{2}+1}) < \sigma'$$

adding the two equalities together we have that

$$\begin{aligned} 2 \cdot s_V^*(\text{correct} \mid x, y_{\frac{n}{2}}) - 2 \cdot s_V^*(\text{correct} \mid x, y_{\frac{n}{2}+1}) &< 2\sigma' \\ s_V^*(\text{correct} \mid x, y_{\frac{n}{2}}) - s_V^*(\text{correct} \mid x, y_{\frac{n}{2}+1}) &< \sigma' < \sigma_V \end{aligned}$$

which showcases that without enforcing Defi.4., under the boldfaced rationality condition, the separation could be bounded above by some constant less than σ_V .

E FROM **TRAINING-FREE** BAYESIAN DECODING GAME (BDG) TO **RL-BASED** PROVER-VERIFIER GAME(PVG)

Prover-Verifier Game (PVG) (Hendrik Kirchner et al., 2024), structured as zero-sum games, encounter substantial challenges that undermine their efficacy in ensuring reliable outputs. The adversarial nature of zero-sum games inherently prioritizes winning over mutual consistency, which leads to strategic behavior focused on exploiting the opposing agent rather than achieving genuine correctness e.g., model collapse. This often results in provers generating outputs that are optimized to mislead the verifier rather than to align with factual truth, thus producing equilibria that favor strategic manipulation over accurate assessment. Such dynamics complicate the training process, requiring extensive tuning and computational resources without guaranteeing robust, interpretable results. Furthermore, the reliance on reinforcement learning in these systems falls short of effectively replacing human feedback, as the trained verifier cannot fully replicate the nuanced judgment required to evaluate complex or ambiguous output. These limitations fall into the misalignment between training objectives and practical needs, where models become adept at adversarial optimization but lack the reliability and consistency necessary for real-world application. However, Bayesian Decoding Game (BDG) can bring the advantages of the game and bridge the purposes with proper implementations, which bypass the extensive training and adversarial pitfalls by directly modeling output verification through probabilistic reasoning, enhancing the interpretability and reliability of the generated content without the dependencies on zero-sum competition.

Here, we consider the connections between the PVG and BDG, and before the comparison, we give a brief introduction of PVG.

We consider a scenario of problems with ground-truth answers $(x, y) \in \mathcal{X} \times \mathcal{Y}$, where x denotes a problem statement and $y = y(x)$ denotes the mapping from the prompt x to ground-truth answer or set of answers. PVG has set the prover’s type to be drawn from a known distribution, which can be either honest or sneaky, to produce an answer z condition on the prompt x and type; such that the honest prover is told to produce correct answers and make sure the verifier recognize the correctness, whereas the sneaky prover is told to trick the verifier by providing wrong answers that are not easy to be detected; then, after receiving solution, the verifier verifies the answer based solely on the answer and prompt, and the output $v(x, z) \in \{0, 1\}$ estimates the correctness of the solution’s answer. The output of the verifier is either 1(the statement is correct) or 0(the statement is incorrect).

More precisely, The game follows the procedure below:

1. The type of prover t is randomly drawn from a known distribution $P(t)$, such that $t \in \{h, s\}$, where h, s dictates honest and sneaky prover’s type.
2. Given a problem statement, the prover generates proofs by maximizing the payoff function(utility) of each round myopically, such that

$$z = \operatorname{argmax} r(v(x, z), \mathbb{1}_{t=h} \mathbb{1}_{\text{correct}}(x, z) + \mathbb{1}_{t=s}(1 - \mathbb{1}_{\text{correct}}(x, z)))$$

the function $r(v, q)$ satisfies that $r(1, 1) > r(0, 1) > r(1, 0) > r(0, 0)$, which corresponds to the fact that both prover prioritizing generating proofs that match their types (correct answer for honest and incorrect answer for sneaky), $\forall v \quad r(v, 1) > r(v, 0)$ and then both want their answers to be accepted as correct by the verifier $\forall q \quad r(1, q) > r(0, q)$.

3. the verifier provides verification $v \in \{0, 1\}$ based solely on the proof x given that the utility is 1 if $v(x, z) = \mathbb{1}(x, z)$ and 0 otherwise
Theoretically, this is a simple Stackleberg game such that an equilibrium exists and before diving into the equilibrium strategy profile, an important definition from (Hendrik Kirchner et al., 2024) needs to be mentioned.

Definition 8. (Checkability Hendrik Kirchner et al. (2024)) Prover π from model family $\mathcal{H}$ is verifiable to verifier v on the data distribution P_{XY} if and only if the prover always generates correct and convincing solutions, and no sneaky prover from the same model family $\mathcal{H}$ can fool

the verifier into accepting incorrect solutions. In mathematical notations, the two properties are given by:

Completeness: $\forall x \in \mathcal{X}, \mathbb{1}_{\text{correct}}(x, \pi(x)) = v(x, \pi(x)) = 1.$

Soundness: $\forall \pi' \in \mathcal{H}, \forall x \in \mathcal{X}, \mathbb{1}_{\text{correct}}(x, \pi'(x)) = 0 \implies v(x, \pi'(x)) = 0.$

With the definition in hand, the equilibrium strategy profile is given by (*the honest prover always provides checkable and correct proof, the sneaky prover always provides noncheckable and incorrect proof, the verifier can always verify the correctness of the given proof*). For the neural networks to approximate the equilibrium strategies, (Hendrik Kirchner et al., 2024) utilized a reinforcement learning-based algorithm to train the prover and the verifier.

RL-based PVG Hendrik Kirchner et al. (2024) can fit in part into the framework of our training-free BDG framework. As for the game-theoretic setting, PVG is a zero sum verifier-lead Stackleberg game, the strategy update schedule must be modified to fit the utility defined in Hendrik Kirchner et al. (2024). Moreover, the verifier’s strategy update cannot be achieved training-free as her utility only depends on the ground truth right/wrong of the candidate and thus needs to be trained; but on the prover side, both honest and sneaky prover can update strategies pain-free from the verifier’s trained strategies.

Firstly, we define the strategy for verifier and prover in the same way as in BDG, such that given the environment signal, the prover generates a probability distribution for a set of answers, and the verifier always generates a probability distribution of {correct, incorrect} for each of the answers. Also, we make the same assumption that each player can observe the opponent’s full strategy profile rather than the realized strategy; then, we are ready to highlight the difference in schedule update under the Markovian schedule, the condition where $v = \text{correct}$, we will abbreviate that as $\text{correct} = \mathbf{C}$, stays the same, such that because they want to align their strategies with the verifier

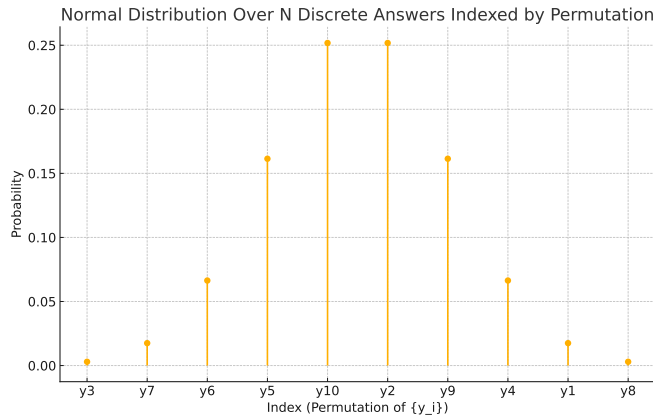
$$b_P^{(t+1)}(y | x, \mathbf{H}) = s_V^{(t)}(\mathbf{C} | x, y)$$

$$a_P^{(t+1)}(y | x, \mathbf{H}) \propto \exp \left\{ \frac{\frac{1}{2} b_P^{(t+1)}(y | x, \mathbf{C}) + \lambda_P \log a_P^{(t)}(y | x, \mathbf{C}, b_P^{(t)})}{1/(\eta_P t) + \lambda_P} \right\}$$

However, for the sneaky prover, her utility is maximized when the verifier mistakens the correctness of the problem. Therefore, the optimal update schedule for the sneaky prover is given updating toward a normal distribution over the preference generated by the probability distribution of verifier’s strategy. The reason for this update is because, near the correct/incorrectness boundary is where the verifier tends to make mistakes, such that

$$a_P^{(t+1)}(y | x, \mathbf{S}) \propto \exp \left\{ \frac{\frac{1}{2} \mathcal{N}(y | a_V) + \lambda_P \log a_P^{(t)}(y | x, \mathbf{I}, b_P^{(t)})}{1/(\eta_P t) + \lambda_P} \right\}$$

For example, if there are 10 answer candidates, the verifier’s preference from her strategy is given by $y_3 \succ y_7 \succ y_6 \succ y_5 \succ y_{10} \succ y_2 \succ y_9 \succ y_4 \succ y_1 \succ y_8$, then $\mathcal{N}(y | a_V)$ is given by



F FROM MEMORYLESS BAYESIAN DECODING GAME (BDG) TO MOVING-AVERAGE EQUILIBRIUM CONSENSUS GAME (ECG)

The moving average update schedule proposed by Jacob et al. (2024) requires both the generator and the verifier to keep track of the average strategy of the opponent in addition to the strategy in the last round, while our Markovian framework allows the players to be memoryless. To better compare ECG with our update schedule, we provide a general, unifying framework called the History window schedule, where the player's belief is given by the average of past history strategies for the period n , and at the same time, this schedule retains a large part the initial policy for each round with a stiffness parameter $\lambda_i, i \in \{G, V\}$. The belief is given by

$$\begin{aligned} b_G^{(t+1)}(y | x, v) &= \frac{1}{n} \sum_{\tau=t-n+1}^t s_V^{(\tau)}(v | x, y) \\ b_V^{(t+1)}(v | x, y) &= \frac{1}{n} \sum_{\tau=t-n+1}^t s_G^{(\tau)}(y | x, v) \end{aligned} \tag{9}$$

Thus the strategy update is given by

$$\begin{aligned} s_G^{(t+1)}(y | x, v) &\propto \exp \left\{ \frac{\frac{1}{2} b_G^{(t+1)}(y | x, v) + \lambda_G \log s_G^{(1)}(y | x, v)}{1/(\eta_G t) + \lambda_G} \right\} \\ s_V^{(t+1)}(v | x, y) &\propto \exp \left\{ \frac{\frac{1}{2} b_V^{(t+1)}(v | x, y) + \lambda_V \log s_V^{(1)}(v | x, y)}{1/(\eta_V t) + \lambda_V} \right\} \end{aligned}$$

As it can be noted in 9, if we take $n = t$, the update schedule coincides with ECG which requires the memory of the moving-average of full history, rather if we take $n = 1$, the update schedule becomes fully memoryless and requires no memory of any past events other than the last period's opponent strategy.

G EXPERIMENT DETAILS

Baselines and Models. For the fair comparison following (Jacob et al., 2024), we use the same public 7B and 13B parameter models from the LLaMA family (Touvron et al., 2023) and perform 16-bit inference for all our experiments. Since we have a multi-round optimization game and in order to distinguish consensus/ zero-sum games, we define ours as a verifier rather than a discriminator. Across the experiments, all the approaches and orthogonal techniques involved:

- **Generative Ranking (G):** The baseline (Brown et al., 2020; Touvron et al., 2023) ranks every candidate y by $P_{\text{LLM}}(y \mid x, \text{correct})$ and picks the top candidate. This is the standard approach used in past work. Due to implementation differences and non-public resources, we report the existing scores in (Jacob et al., 2024).
- **Discriminative Ranking (D):** Following (Jacob et al., 2024), this approach reweighs every query-candidate pair (x, y) by $\pi_D^{(1)}(\text{correct} \mid x, y)$. Typically, this would surpass the performance of ordinary individuals, who might neglect to notice the ambiguity errors. And outstrip the generators that might trust the unreliable decoding.
- **Mutual Information Ranking (MI):** The mutual-information based baseline reweighs every candidate y by $P_{\text{LM}}(y \mid x, \text{correct}) \cdot P_{\text{LM}}(\text{correct} \mid x, y)$ (Li & Jurafsky, 2016).
- **Self-Contrastive Decoding (SCD):** The contrastive-based method (Jacob et al., 2024; Li et al., 2022) utilizes the contrastive-based generator $\pi_G^{(1)}$ to reweight every candidate y by $\pi_G^{(1)}(\text{correct} \mid x, y)$. This method achieves a contrasting effect by comparing negative samples instead of employing a verifier (in BDG)/ discriminator (in ECG).
- **Equilibrium Consensus Discriminator (ECG):** This approach is based on discriminator π_D^* (Jacob et al., 2024). It reweighs every query-candidate pair (x, y) by $\pi_D^*(\text{correct} \mid x, y)$. This method, involving comprehensive policies and updates, serves as our main benchmark.
- **Bayesian Decoding Game (BDG):** This approach utilizes our Bayesian Decoding Game-based discriminator π_D^* . This approach reweighs every query-candidate pair (x, y) by $\pi_D^*(\text{correct} \mid x, y)$.

Orthogonal Techniques. Furthermore, BDG can combine chain-of-thought (CoT) (Wei et al., 2022) and few-shots setting (Wei et al., 2022) as orthogonal extra gains.

- **Chain-of-Thought (CoT):** CoT (Wei et al., 2022) prompting enables language models to generate intermediate reasoning steps, improving performance on complex tasks. By providing exemplars of reasoning chains, the model is guided to produce more coherent and accurate responses.
- **Few-Shot:** Few-shot setting (Wei et al., 2022) involves providing the model with a small number of example input-output pairs within the prompt. This technique helps the model adapt to the task at hand without additional fine-tuning, improving its ability to generalize from limited data.

Hyperparameters. We set η_D , λ_D and η_G , λ_G with 0.1 compared to ECG. Experiments are run 5000 times with early stopping based on equilibrium convergence. BDG can usually converge by 500 iterations or less. The hyperparameters can be larger according to the tasks and initial model ability.

Extra Metrics. Following (Li et al., 2022), we have

- *Diversity.* This metric aggregates n-gram repetition rates:

$$\text{DIV} = \prod_{n=2}^4 \frac{\text{unique n-grams}(x_{\text{cont}})}{\text{total n-grams}(x_{\text{cont}})}.$$

Models that score low for diversity are prone to repetition, while models that score high for diversity are lexically diverse.

- *MAUVE.* MAUVE (Pillutla et al., 2021) measures the similarity between generated text and gold reference text.

- *Coherence*. (Su et al., 2022) approximates coherence by cosine similarity between the sentence embeddings of prompt x_{pre} and generated continuation x_{cont} :

$$\text{COH}(x_{\text{cont}}, x_{\text{pre}}) = \frac{\text{EMB}(x_{\text{pre}}) \cdot \text{EMB}(x_{\text{cont}})}{\|\text{EMB}(x_{\text{pre}})\| \cdot \|\text{EMB}(x_{\text{cont}})\|},$$

where $\text{EMB}(x)$ represents the pre-trained SimCSE embedding (Gao et al., 2021).

- *Human Evaluation*. To further evaluate the quality of the generated text, we consider two critical aspects: *correctness* and confidence in *reliability*. More details can be found in the next section.

H SEARCHING & CONVERGENCE BEHAVIOR SUPPLEMENTARY

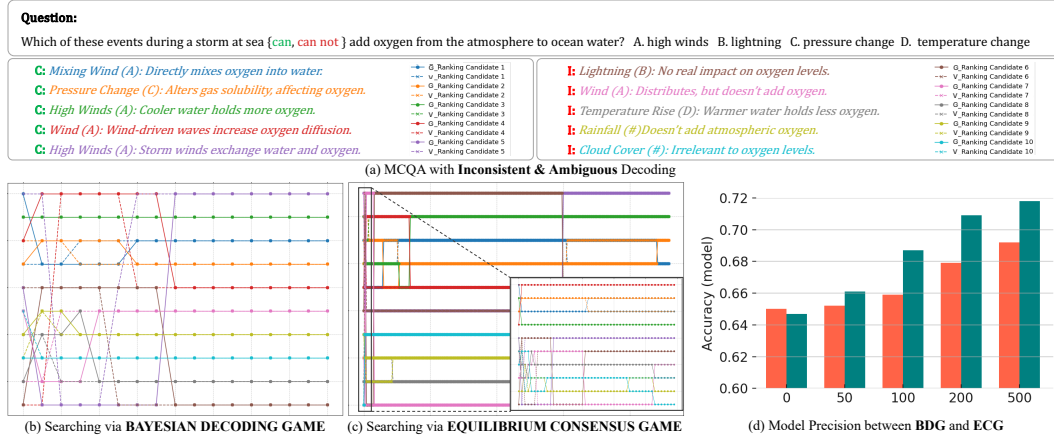


Figure 6: **BDG's game design quickly reaches equilibrium and consensus between the generator and discriminator, typically within 100 epochs.** In contrast, **ECG** requires significantly more epochs (3000 in this case) and exhibits continuous fluctuations (as shown in the lower right) before achieving consensus. (Zoom in for details.)

We first compare searching behaviors of **BDG** with the most closely related method, the **ECG** (Jacob et al., 2024), in the multiple-choice question answering (MCQA) task (Clark et al., 2018). Fig.6 provides a visual case study. BDG demonstrates a swift and consistent convergence in (b).

Conversely, the ECG, shown in (c), exhibits prolonged and inconsistent searching behavior. Despite continuous shifts in candidate selections, ECG fails to achieve stable convergence with persistent disagreement between the generator and verifier. (d) and Tab.1 highlights the enhanced and fast convergence properties of the BDG over the ECG.

I HUMAN EVALUATION

Setting. In this experiment, participants were tasked with evaluating the correctness of ten answers to a high-school level multiple-choice mathematics problem generated by a Large Language Model (LLM). Participants were instructed to classify each answer as correct, incorrect, or ambiguous. The experiment was conducted in two stages:

In the first stage, participants were given two minutes to classify as many answers as possible, and their results were recorded. In the second stage, participants were allowed to allocate their time freely to complete the remaining classifications, and they were asked to record the time upon completion of their classifications. Below is the questionnaire we utilized for the experiment.

Each participant was randomly assigned three distinct problems, and the corresponding solutions were classified under three conditions: without any hints, with a BDG hint, and with an ECG hint. The hints provided were rankings of the answers generated by the respective models (BDG and ECG). The assignment of different problems across the three conditions was designed to prevent memorization and to control for potential confounding effects related to the content of the specific problem. Problems were drawn from a pool of questions with similar difficulty levels, allowing for consistent observation of treatment effects across varying problem sets.

Samples. To better illustrate the experiment setting details, we provided the questionnaire interface, the instructions, and two cases set below.

Please read the instructions

#Question	Type	2 Minutes (Mandatory)				Unlimited Time			
		Correct	Incorrect	Ambiguous	Actual Time (Seconds)	Correct	Incorrect	Ambiguous	Actual Time (Seconds)
#1	w/o	Sol No.{	Sol No.(s)	Sol No.(s)	Seconds	Sol No.{	Sol No.(s)	Sol No.(s)	Seconds
#2	BDG	Sol No.{	Sol No.(s)	Sol No.(s)	Seconds	Sol No.{	Sol No.(s)	Sol No.(s)	Seconds
#3	ECG	Sol No.{	Sol No.(s)	Sol No.(s)	Seconds	Sol No.{	Sol No.(s)	Sol No.(s)	Seconds

Submit Questionnaire

Instructions For task 1, you will judge the **correctness** of each solution under a mandatory 2-minute limit, and for the ones that you are unsure, label them as ambiguous and record the result on the answer sheet provided. If you did not finish in 2 minutes, then record your incomplete result at the 2-minute cutoff and continue your task until finished, also remember to record your completion time.

For task 2, 3, we will provide a reference (ranking of the solutions to assist and accelerate your correctness judgement, however, always stick to your own judgement if there exists a divergence. All other procedures are the same as task 1.

Question #

Random selected On a Cartesian coordinate plane, points (1, 2) and (7, 4) are opposite vertices of a square. What is the area of the square?

Choices A. 20 B. 34 C. 40 D. $\sqrt{40}$

Task 1 *Candidates*

0. "To find the area of the square, we multiply the length of the diagonal, which is $\sqrt{40}$, by itself. This gives us 40, so the answer is C. 40."
 1. "The distance between the points (1, 2) and (7, 4) is $\sqrt{40}$. This is the diagonal of the square, so the area could either be 20 or 40. The answer might be A. 20."
 2. "The diagonal of the square is $\sqrt{40}$, and the area must be equal to this value. The answer is D. $\sqrt{40}$."
 3. "First, calculate the diagonal between the points (1, 2) and (7, 4), which gives $\sqrt{40}$. Since this is the diagonal of the square, the area is $\frac{(\sqrt{40})^2}{2} = 20$. Therefore, the answer is A. 20."
 4. "The area of the square is found by calculating the distance between the points. The diagonal is $\sqrt{40}$, so the area is likely $\sqrt{40}$. The answer is D. $\sqrt{40}$."
 5. "First, find the distance between (1, 2) and (7, 4) using the distance formula: $d = \sqrt{(7-1)^2 + (4-2)^2} = \sqrt{40}$. This distance is the diagonal of the square. The area of the square is then $\frac{(\sqrt{40})^2}{2} = 20$. Therefore, the correct answer is A. 20."
 6. "Using the distance formula, the diagonal of the square is $\sqrt{40}$. Therefore, the area of the square is 20. The answer is A. 20."
 7. "First, calculate the distance between the points (1, 2) and (7, 4), which gives the diagonal of the square. The distance formula gives $d = \sqrt{(7-1)^2 + (4-2)^2} = \sqrt{40}$. Since the diagonal of the square is $\sqrt{40}$, the area is $(\sqrt{40})^2/2 = 20$. Therefore, the answer is A. 20."
 8. "Using the distance formula, the diagonal of the square is found to be $\sqrt{40}$. The area of the square is 20, so the answer is A. 20."
 9. "Calculate the diagonal of the square using the distance formula: $d = \sqrt{40}$. Since the diagonal is $\sqrt{40}$, the area of the square is simply 40. Therefore, the answer is C. 40."
-

Task 2 **BDG Guidance** [Will be hidden during evaluation]

0.

Consistent "First, find the distance between (1, 2) and (7, 4) using the distance formula: $d = \sqrt{(7-1)^2 + (4-2)^2} = \sqrt{40}$. This distance is the diagonal of the square. The area of the square is then $\frac{(\sqrt{40})^2}{2} = 20$. Therefore, the correct answer is A. 20."

-
1. "First, calculate the diagonal between the points (1, 2) and (7, 4), which gives $\sqrt{40}$. Since this is the diagonal of the square, the area is $\frac{(\sqrt{40})^2}{2} = 20$. Therefore, the answer is A. 20."
 2. "Using the distance formula, the diagonal of the square is $\sqrt{40}$. Therefore, the area of the square is 20. The answer is A. 20."
 3. "Using the distance formula, the diagonal of the square is found to be $\sqrt{40}$. The area of the square is 20, so the answer is A. 20."
 4. "The distance between the points (1, 2) and (7, 4) is $\sqrt{40}$. This is the diagonal of the square, so the area could either be 20 or 40. The answer might be A. 20."
 5. "First, calculate the distance between the points (1, 2) and (7, 4), which gives the diagonal of the square. The distance formula gives $d = \sqrt{(7-1)^2 + (4-2)^2} = \sqrt{40}$. Since the diagonal of the square is $\sqrt{40}$, the area is $(\sqrt{40})^2/2 = 20$. Therefore, the answer is A. 20."
 6. "The diagonal of the square is $\sqrt{40}$, and the area must be equal to this value. The answer is D. $\sqrt{40}$."
 7. "The area of the square is found by calculating the distance between the points. The diagonal is $\sqrt{40}$, so the area is likely $\sqrt{40}$. The answer is D. $\sqrt{40}$."
 8. "Calculate the diagonal of the square using the distance formula: $d = \sqrt{40}$. Since the diagonal is $\sqrt{40}$, the area of the square is simply 40. Therefore, the answer is C. 40."
 9. *Inconsistent* "To find the area of the square, we multiply the length of the diagonal, which is $\sqrt{40}$, by itself. This gives us 40, so the answer is C. 40."
-

Task 3 **ECG Guidance** [Will be hidden during evaluation]

0. *Consistent* "First, find the distance between (1, 2) and (7, 4) using the distance formula: $d = \sqrt{(7-1)^2 + (4-2)^2} = \sqrt{40}$. This distance is the diagonal of the square. The area of the square is then $\frac{(\sqrt{40})^2}{2} = 20$. Therefore, the correct answer is A. 20."
1. "Using the distance formula, the diagonal of the square is $\sqrt{40}$. Therefore, the area of the square is 20. The answer is A. 20."
2. "First, calculate the diagonal between the points (1, 2) and (7, 4), which gives $\sqrt{40}$. Since this is the diagonal of the square, the area is $\frac{(\sqrt{40})^2}{2} = 20$. Therefore, the answer is A. 20."
3. "The distance between the points (1, 2) and (7, 4) is $\sqrt{40}$. This is the diagonal of the square, so the area could either be 20 or 40. The answer might be A. 20."
4. "Using the distance formula, the diagonal of the square is found to be $\sqrt{40}$. The area of the square is 20, so the answer is A. 20."
5. "Calculate the diagonal of the square using the distance formula: $d = \sqrt{40}$. Since the diagonal is $\sqrt{40}$, the area of the square is simply 40. Therefore, the answer is C. 40."
6. "First, calculate the distance between the points (1, 2) and (7, 4), which gives the diagonal of the square. The distance formula gives $d = \sqrt{(7-1)^2 + (4-2)^2} = \sqrt{40}$. Since the diagonal of the square is $\sqrt{40}$, the area is $(\sqrt{40})^2/2 = 20$. Therefore, the answer is A. 20."
7. "The diagonal of the square is $\sqrt{40}$, and the area must be equal to this value. The answer is D. $\sqrt{40}$."

8.	"To find the area of the square, we multiply the length of the diagonal, which is $\sqrt{40}$, by itself. This gives us 40, so the answer is C. 40."
9.	
<i>Inconsistent</i>	"The area of the square is found by calculating the distance between the points. The diagonal is $\sqrt{40}$, so the area is likely $\sqrt{40}$. The answer is D. $\sqrt{40}$."

Question #	
<i>Random selected</i>	The standard tip in a restaurant is 15% of the bill before tax. Many people in California find the tip by doubling the sales tax, which is 8.25%. By how many cents does this exceed the standard tip on a bill of \$60 before tax?
Choices	A. 90 B. 80 C. 70 D. 60

Task 1	<i>Candidates</i>
0.	"The tax is doubled to 16.5%, and the standard tip is 15%. The difference is around 70 cents for a \$60 bill, so the answer is C. 70."
1.	"To find the tip, we double the sales tax of 8.25%, which gives us 16.5%. The standard tip is 15%, and the difference on a \$60 bill is $60 \times (16.5\% - 15\%) = 90$ cents. Therefore, the answer is A. 90."
2.	"Doubling the sales tax of 8.25% gives 16.5%. The difference between the standard tip and this new tip is around 60 cents for a \$60 bill. Therefore, the answer is D. 60."
3.	"Doubling the tax gives a tip of 16.5%, and the difference from the standard 15% tip is 90 cents. Therefore, the answer is A. 90."
4.	"The standard tip is 15% of 60, which is $60 \times 0.15 = 9$ dollars. Doubling the sales tax of 8.25% gives 16.5%, and the tip based on this is $60 \times 0.165 = 9.90$ dollars. The difference is $9.90 - 9.00 = 90$ cents. Therefore, the correct answer is A. 90."
5.	"Doubling the sales tax gives 16.5%, and the standard tip is 15%. This difference gives $60 \times 0.5\% = 30$ cents. Therefore, the answer is A. 90."
6.	"The standard tip is 15%, or 9, and doubling the sales tax gives a tip of 16.5%, or 9.90. The difference is 90 cents. The correct answer is A. 90."
7.	"The difference between doubling the sales tax and using the standard tip is 90 cents for a 60 bill, but it could also be 80 cents depending on rounding. It might be A. 90 or B. 80."
8.	"The standard tip is 15%, and doubling the tax gives a different percentage. The difference could be either 60 or 90 cents. The answer might be D. 60."
9.	"Doubling the sales tax gives a tip rate of 16.5%, while the standard tip is 15%. The difference for a 60\$ bill is 90 cents. So the answer is A. 90."

Task 2	BDG Guidance [Will be hidden during evaluation]
0.	
<i>Consistent</i>	"The standard tip is 15% of 60, which is $60 \times 0.15 = 9$ dollars. Doubling the sales tax of 8.25% gives 16.5%, and the tip based on this is $60 \times 0.165 = 9.90$ dollars. The difference is $9.90 - 9.00 = 90$ cents. Therefore, the correct answer is A. 90."
1.	"The standard tip is 15%, or 9, and doubling the sales tax gives a tip of 16.5%, or 9.90. The difference is 90 cents. The correct answer is A. 90."

-
2. "To find the tip, we double the sales tax of 8.25%, which gives us 16.5%. The standard tip is 15%, and the difference on a \$60 bill is $60 \times (16.5\% - 15\%) = 90$ cents. Therefore, the answer is A. 90."
3. "Doubling the tax gives a tip of 16.5%, and the difference from the standard 15% tip is 90 cents. Therefore, the answer is A. 90."
4. "Doubling the sales tax gives a tip rate of 16.5%, while the standard tip is 15%. The difference for a 60\$ bill is 90 cents. So the answer is A. 90."
5. "The difference between doubling the sales tax and using the standard tip is 90 cents for a 60 bill, but it could also be 80 cents depending on rounding. It might be A. 90 or B. 80."
6. "Doubling the sales tax of 8.25% gives 16.5%. The difference between the standard tip and this new tip is around 60 cents for a \$60 bill. Therefore, the answer is D. 60."
7. "Doubling the sales tax gives 16.5%, and the standard tip is 15%. This difference gives $60 \times 0.5\% = 30$ cents. Therefore, the answer is A. 90."
8. "The tax is doubled to 16.5%, and the standard tip is 15%. The difference is around 70 cents for a \$60 bill, so the answer is C. 70."
9. *Inconsistent* "The standard tip is 15%, and doubling the tax gives a different percentage. The difference could be either 60 or 90 cents. The answer might be D. 60."
-

Task 3 **ECG Guidance** [Will be hidden during evaluation]

0. *Consistent* "The standard tip is 15%, or 9, and doubling the sales tax gives a tip of 16.5%, or 9.90. The difference is 90 cents. The correct answer is A. 90."
1. "The standard tip is 15% of 60, which is $60 \times 0.15 = 9$ dollars. Doubling the sales tax of 8.25% gives 16.5%, and the tip based on this is $60 \times 0.165 = 9.90$ dollars. The difference is $9.90 - 9.00 = 90$ cents. Therefore, the correct answer is A. 90."
2. "To find the tip, we double the sales tax of 8.25%, which gives us 16.5%. The standard tip is 15%, and the difference on a \$60 bill is $60 \times (16.5\% - 15\%) = 90$ cents. Therefore, the answer is A. 90."
3. "Doubling the tax gives a tip of 16.5%, and the difference from the standard 15% tip is 90 cents. Therefore, the answer is A. 90."
4. "Doubling the sales tax gives a tip rate of 16.5%, while the standard tip is 15%. The difference for a 60\$ bill is 90 cents. So the answer is A. 90."
5. "The difference between doubling the sales tax and using the standard tip is 90 cents for a 60 bill, but it could also be 80 cents depending on rounding. It might be A. 90 or B. 80."
6. "Doubling the sales tax gives 16.5%, and the standard tip is 15%. This difference gives $60 \times 0.5\% = 30$ cents. Therefore, the answer is A. 90."
7. "The tax is doubled to 16.5%, and the standard tip is 15%. The difference is around 70 cents for a \$60 bill, so the answer is C. 70."
8. "The standard tip is 15%, and doubling the tax gives a different percentage. The difference could be either 60 or 90 cents. The answer might be D. 60."
9. *Inconsistent* "Doubling the sales tax of 8.25% gives 16.5%. The difference between the standard tip and this new tip is around 60 cents for a \$60 bill. Therefore, the answer is D. 60."
-

Results. To differentiate between expert and non-expert participants, a threshold of 150 seconds was set based on empirical observations of participant behavior. This threshold was corroborated by a scatterplot that visually demonstrated the partitioning between experts and non-experts, supporting the appropriateness of the selected cut-off time for classification performance. From the 183 samples we collected, we have come to conclusions:

Firstly, human evaluation on these LLM-generated solutions have intrinsic unreliability and ambiguity, especially for non-expert participants; under the 2-minute time constraint, there were 28 participants who scored an accuracy of 50% or below, even without time constraint, there were 10 participants in total whose accuracies were 60% or below.

Secondly, with the hint of BDG, we observed that the participants in both the expert and the non-expert groups outperformed the participants without the hint or the ECG hint; more importantly, with the hint of BDG, the performance of the non-expert participants improved significantly with respect to the expert participants, whereas the ECG hint did not achieve this.

Thirdly, across problems, samples with BDG hint achieved the highest accuracies for 7/10 problems and fastest completion also for 7/10 problems. This result showcases the robustness of BDG hint also within the same problems.

Last but not least, the BCG hint has also helped non-expert participants spot ambiguous solutions more accurately compared to non-expert participants with the ECG hint or without the hint.

Generations.**Question.**

Suppose the graph of $y = f(x)$ includes the points $(1, 5)$, $(2, 3)$, and $(3, 1)$. Based only on this information, there are two points that must be on the graph of $y = f(f(x))$. If we call those points (a, b) and (c, d) , what is $ab + cd$?

BDG.

"Given the points $(1, 5)$, $(2, 3)$, and $(3, 1)$ on the graph of $y = f(x)$, we compute $f(f(2)) = f(3) = 1$ and $f(f(3)) = f(1) = 5$. So the points $(2, 1)$ and $(3, 5)$ are on the graph of $y = f(f(x))$. Therefore, $ab + cd = 2 \times 1 + 3 \times 5 = 17$."

ECG.

"From $f(x)$, we see that $f(f(2)) = 1$ and $f(f(3)) = 5$. Therefore, $ab + cd = 17$. The answer is 17"

Question.

Positive integers x and y have a product of 56 and $x < y$. Seven times the reciprocal of the smaller integer plus 14 times the reciprocal of the larger integer equals 4. What is the value of x ?

BDG.

"Let the integers be x and y . Since $x \cdot y = 56$ and $x < y$, try small values. Testing $x = 2$ and $y = 28$ satisfies $7 \cdot \frac{1}{x} + 14 \cdot \frac{1}{y} = 4$. Therefore, $x = 2$, so the answer is 2."

ECG.

"Since $x \cdot y = 56$, try small values of x like 2. This satisfies the equation, so the answer is 2."

Question.

Six positive integers are written on the faces of a cube. Each vertex is labeled with the product of the three numbers on the faces adjacent to the vertex. If the sum of the numbers on the vertices is equal to 1001, then what is the sum of the numbers written on the faces?

BDG.

"The numbers on the faces of the cube contribute to the product at each vertex. Given that the sum of the vertex products is 1001, and knowing the cube's geometry, we deduce that the sum of the numbers on the faces is 31."

ECG.

"Using the fact that the sum of the vertex products is 1001, and knowing there are six faces, we calculate that the sum of the face numbers is 31. So the answer is 31."