CODE DRIVEN GAME THEORETIC EVOLUTION OF LLM AGENTS AS HOLISTIC STRATEGY GENERATORS

Siwei Li, Xin Wang
School of Artificial Intelligence
Beihang University
Beijing 100191, P.R.China
{siweili,wangxin_1993}@buaa.edu.cn

ABSTRACT

Game-theoretic settings provide a potent testbed for the strategic reasoning of Large Language Models (LLMs). However, current evaluations are largely constrained to static, single-shot or myopic, per-timestep interactions, leaving the capacity for holistic, long-term policy formulation unexamined. We introduce a novel LLM-driven evolutionary game tournament framework to investigate emergent, far-sighted strategy. In our paradigm, the LLM is repositioned as a high-level strategy generator, producing complete, interpretable Python policies. These policies undergo iterative refinement against a dynamic Hall of Fame (HoF) as the LLM analyzes tournament performance to generate superior variants, directing the ecosystem's evolution. Extensive experiments across five distinct LLM architectures and multiple random seeds reveal that this process enables LLMs to consistently discover robust cooperative policies outperforming standard reciprocity-based algorithms and to autonomously generate complex deceptive strategies, exhibiting distinct phases of strategic disguise, inducement, and payoff harvesting. These findings confirm that such strategic evolution is a robust, model-agnostic phenomenon.

1 INTRODUCTION

The rapid advancement of Large Language Models (LLMs) is transforming them from specialized tools into autonomous agents capable of complex planning. As these agents are deployed in interconnected environments, evaluating their capacity for far-sighted strategic reasoning becomes a critical challenge. However, current evaluative paradigms remain insufficient: most benchmarks rely on single-action tests or step-by-step reasoning at each timestep. This action-at-a-time approach fails to capture long-term planning and prevents models from formulating holistic strategies. Furthermore, existing research often focuses on simulating human-like bounded rationality rather than exploring the intrinsic capacity of LLMs to architect optimal, far-sighted policies.Due to space constraints, a detailed discussion of related work is provided in the Appendix.

To bridge this gap, we introduce an LLM-driven evolutionary game tournament framework. We propose a paradigm shift: the LLM functions not as a direct participant, but as a strategy generator. In this framework, the LLM analyzes the competitive landscape and generates complete, executable, and interpretable Python policies. These policies compete against each other and a dynamic Hall of Fame (HoF) in a tournament, undergoing iteration through a simulated evolutionary process. This mechanism allows us to transcend the limitations of classical Evolutionary Game Theory (EGT) by using LLMs to perform deliberate strategic design.

2 METHODOLOGY

2.1 FRAMEWORK ARCHITECTURE

The proposed framework is an iterative, four-stage closed-loop system designed to simulate an evolutionary process. The complete workflow is illustrated in Figure 1, with the following key stages:

- **Generation:** At the beginning of the evolutionary process and after each subsequent generation, the LLM generates new, self-contained strategies based on a structured prompt.
- **Competition:** A population of strategies competes in a round-robin tournament within a predefined game environment. To ensure robustness and prevent genetic drift, agents compete not only against their peers but also against a dynamic HoF containing rigorous baselines and past elites.
- **Analysis:** The system calculates a weighted fitness score for each strategy, integrating performance metrics from both peer-play and HoF-play.
- **Iteration:** Drawing upon the analysis from the previous stage, the system constructs an Inheritance Prompt containing comprehensive details from the preceding round, including the game background, along with each agent's strategy code, score, and explicit intermediate reasoning steps generated via Chain-of-Thought (CoT). This mechanism allows the LLM to articulate the logic behind its strategic improvements before synthesizing the final policy code. This prompt is then provided to the LLM to generate a new set of strategies intended to replace those eliminated. This step closes the evolutionary loop, driving the population toward more optimal solutions.

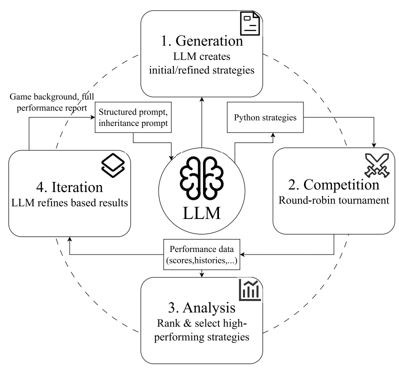

Figure 1: The workflow of the LLM-driven strategy evolution framework. The diagram illustrates the complete cycle, from initial strategy generation through competition, analysis, and iterative regeneration of new strategies.

2.2 LLM AS A STRATEGY GENERATOR

A key innovation of this framework is that it shifts the LLM's role from a myopic, per-step decision-maker to a holistic policy generator. Formally, a strategy s is a mapping from the history state space H to the action space A, where a history state $h \in H$ encapsulates all information necessary for the current decision:

$$s : H \to A \tag{1}$$

The task of LLM is to generate an executable, interpretable program (e.g., a Python function) that implements this mapping s. This function receives the relevant game state and history as input and returns a discrete action.

This policy-centric output guarantees intrinsic interpretability. Unlike high-dimensional latent representations, each strategy generated by our framework is human-readable code. This allows for the direct and precise analysis of the strategy's internal logic, offering novel insights into the high-level reasoning and planning capabilities of the LLM.

To rigorously evaluate the intrinsic strategic capacity of LLMs, we employ a *tabula rasa* initialization strategy. In the initial generation $P_0 = \{s_1, s_2, ..., s_N\}$, the LLM is provided solely with the game rules and the standard function signature, devoid of any behavioral guidance or strategic priors (e.g., 'act cooperatively' or 'be aggressive'). This minimal-interference setup is critical: it isolates the evolutionary process as the sole driver of improvement, ensuring that emergent complex behaviors are genuine products of the system's reasoning rather than artifacts of prompt-induced bias. By eliminating human-defined archetypes, we demonstrate that the LLM can autonomously discover optimal high-level strategies from a neutral starting state.

2.3 The Evolutionary Tournament Mechanism

The evolutionary process is driven by a tournament, elimination, and regeneration cycle. We utilize the classic Iterated Prisoner's Dilemma as the game environment, adopting the payoff values $(R, S, T, P) = (3, 0, 5, 1)$, representing Reward, Sucker, Temptation, and Punishment.

Mitigating Genetic Drift via Hall of Fame: A critical challenge in LLM-based evolution with limited population size ($N = 12$) is *genetic drift*, where suboptimal strategies might transiently dominate due to stochastic variance. To counteract this and ensure rigorous evaluation, we introduce a dynamic HoF. The HoF is initialized (H_0) with diverse baselines to serve as stability anchors: *Tit-for-Tat(Nowak & Sigmund, 1992), Grim Trigger, Pavlov, Always Cooperate, Always Defect, Random, Alternator, Bayesian, Generous Tit-for-Tat, Gradual, Prober, Suspicious Tit-for-Tat,* and *ZD Extortion*. In each generation, the top-3 performing agent strategies are permanently added to the HoF, preventing the ecosystem from cycling through unstable equilibria.

Hybrid Fitness Calculation: Within a single generation g, strategies participate in a hybrid tournament. Each strategy s_i plays against all peers in the current population P_g and all strategies in the HoF. To balance internal adaptation with external robustness, the total fitness F_i is calculated as a weighted average:

$$F_i = \alpha \cdot \bar{U}_{pop}(s_i) + (1 - \alpha) \cdot \bar{U}_{HoF}(s_i) \tag{2}$$

where $\bar{U}_{pop}$ and $\bar{U}_{HoF}$ represent the average payoffs against the current population and the HoF, respectively, with a weighting factor $\alpha = 0.6$. This metric ensures that survivors must be both dominant among peers and robust against historical elites.

Evolutionary Cycle: The mechanism simulates "survival of the fittest" through the following steps:

- **Selection:** Strategies are ranked by F_i. The top $N - M$ ($M = 6$) strategies are selected as survivors (P_g^S).
- **Elimination:** The bottom M strategies are deemed unfit and removed.
- **Regeneration:** The system constructs an Inheritance Prompt containing the code and performance reports of both survivors and eliminated agents. The LLM generates M new strategies (P_g^{new}) to replace the eliminated ones.
- **Population Renewal:** The next generation is formed by the union of survivors and new strategies:

$$P_{g+1} = P_g^S \cup P_g^{\text{new}} \tag{3}$$

Through this feedback loop, the ecosystem is driven to progressively evolve towards complex, robust solutions rather than succumbing to cyclic drift.

3 Experiments

3.1 Experimental Setup

Experiments were conducted across five distinct LLM architectures: Gemini-2.5-Pro, GPT-4o, DeepSeek-V3.2, Qwen3-Coder-480B, and Qwen3-8B. To ensure statistical robustness, each model configuration was executed with 5 independent random seeds. The population size was set to $N = 12$, with the bottom $M = 6$ agents eliminated over 10 generations ($T = 1.0$).

We evaluate performance across three information scenarios:

- **Scenario A (High-Fidelity):** Complete round-by-round history.
- **Scenario B (Partial-Fidelity):** History restricted to the last 3 rounds ($k = 3$).
- **Scenario C (Abstracted-Fidelity):** Only cumulative reputation scores ($R_{\text{self}}, R_{\text{opp}}$) are provided.

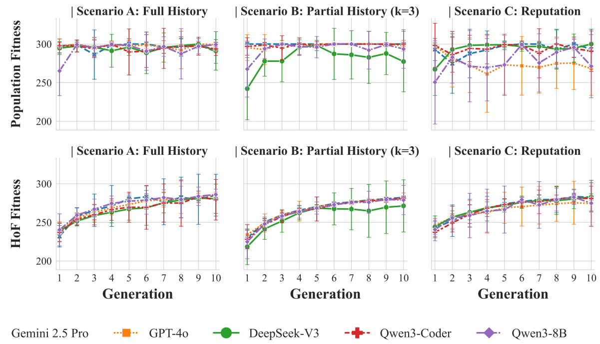

Figure 2: Evolutionary fitness dynamics across 10 generations in three information scenarios. The curves represent the average performance of five distinct LLM architectures over five independent random seeds, with shaded regions indicating the standard deviation. **Top Row (Population Fitness):** Average payoffs of agents competing against their peers within the evolving population. **Bottom Row (HoF Fitness):** Average payoffs of agents competing against the static HoF baselines.

The evolutionary trajectories in Fig. 2 reveal distinct optimization dynamics. Population Fitness (peer-play) rapidly saturates near the cooperative optimum (≈ 300) within the first 2-3 generations. This rapid convergence indicates that, in the absence of aggressive behavioral priors, LLMs exhibit a strong intrinsic bias toward cooperation, allowing the population to quickly reach a high-trust equilibrium. Conversely, HoF Fitness demonstrates a sustained ascent. This confirms that while the basic cooperative stance is established early, the evolutionary process drives a continuous refinement of logic, enabling agents to extract maximum utility from the heterogeneous baselines in the Hall of Fame.Due to space constraints, the quantitative comparison with classical human strategies originally intended for tabular presentation is visualized in the HoF Fitness trajectories (Fig. 2, Bottom Row), which allows for a clear visualization of the score dynamics of LLM-generated strategies when competing against these fixed strategies.

4 CONCLUSION

This work establishes a rigorous evolutionary framework that reframes LLMs from myopic, per-timestep decision-makers to holistic strategy generators. Validated across five distinct LLMs, our results confirm that LLM's ability to generate complex, adaptive strategies is a robust and model-agnostic phenomenon.Comprehensive details regarding related works, experimental results, and in-depth analyses are provided in the Appendix.

5 FUTURE WORK

To further scrutinize the cognitive depth of LLMs in strategic synthesis, we conducted an ablative study on the information granularity provided during the evolutionary feedback loop. Specifically, we controlled the LLM's visibility into the preceding generation's environment across two dimensions: (1) a full-information setting, where the model receives all peer policies alongside their respective payoff matrices; and (2) a restricted-information setting, where the model is only provided with the failed strategies and their associated scores. This variable control allows us to isolate the LLM's capacity for environmental perception, particularly its ability to perform inverse strategic inference—deducing the latent logic of opponents solely from observed performance metrics.Beyond the Iterated Prisoner's Dilemma, our experimental framework was extended to other canonical game-theoretic models, including the Stag Hunt and Hawk-Dove games. Due to space constraints, the detailed methodologies and findings for these scenarios are deferred to future publications. Moving forward, we intend to utilize these diverse benchmarks to pinpoint the systemic vulnerabilities of LLMs in strategy generation. Our subsequent research will focus on developing targeted refinement mechanisms to fortify these weaknesses, thereby enhancing the overall strategic robustness and reasoning consistency of LLM-based autonomous agents.

6 Acknowledgments

The authors would like to express their sincere gratitude to the Beijing Advanced Innovation Center for Future Blockchain and Privacy Computing, Beijing, China, for the generous financial support of this research. This funding enabled the full implementation of our experimental design, data analysis, and manuscript writing, and was instrumental to the completion of this work.

References

Gati V Aher, Rosa I Arriaga, and Adam Tauman Kalai. Using large language models to simulate multiple humans and replicate human subject studies. In *International conference on machine learning*, pp. 337–371. PMLR, 2023.

Lisa P Argyle, Ethan C Busby, Nancy Fulda, Joshua R Gubler, Christopher Rytting, and David Wingate. Out of one, many: Using language models to simulate human samples. *Political Analysis*, 31(3):337–351, 2023.

Yiting Chen, Tracy Xiao Liu, You Shan, and Songfa Zhong. The emergence of economic rationality of gpt. *Proceedings of the National Academy of Sciences*, 120(51):e2316205120, 2023.

Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. Gtbench: Uncovering the strategic reasoning capabilities of llms via game-theoretic evaluations. *Advances in Neural Information Processing Systems*, 37:28219–28253, 2024.

Caoyun Fan, Jindou Chen, Yaohui Jin, and Hao He. Can large language models serve as rational players in game theory? a systematic analysis. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pp. 17960–17967, 2024.

Nathan Herr, Fernando Acero, Roberta Raileanu, María Pérez-Ortiz, and Zhibin Li. Are large language models strategic decision makers? a study of performance and bias in two-player non-zero-sum games. *arXiv preprint arXiv:2407.04467*, 2024.

John J Horton. Large language models as simulated economic agents: What can we learn from homo silicus? Technical report, National Bureau of Economic Research, 2023.

Jingru Jia, Zehua Yuan, Junhao Pan, Paul E McNamara, and Deming Chen. Large language model strategic reasoning evaluation through behavioral game theory. *arXiv preprint arXiv:2502.20432*, 2025.

Ayato Kitadai, Yudai Tsurusaki, Yusuke Fukasawa, and Nariaki Nishino. Toward a novel methodology in economic experiments: Simulation of the ultimatum game with large language models. In *2023 IEEE International Conference on Big Data (BigData)*, pp. 3168–3175. IEEE, 2023.

Rubing Li, João Sedoc, and Arun Sundararajan. Reasoning and the trusting behavior of deepseek and gpt: An experiment revealing hidden fault lines in large language models. *arXiv preprint arXiv:2502.12825*, 2025.

Nunzio Lorè and Babak Heydari. Strategic behavior of large language models and the role of game structure versus contextual framing. *Scientific Reports*, 14(1):18490, 2024.

Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Qiang Guan, Tao Ge, and Furu Wei. Alympics: Llm agents meet game theory. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 2845–2866, 2025.

Qiaozhu Mei, Yutong Xie, Walter Yuan, and Matthew O Jackson. A turing test of whether ai chatbots are behaviorally similar to humans. *Proceedings of the National Academy of Sciences*, 121(9): e2313925121, 2024.

Martin A Nowak and Karl Sigmund. Tit for tat in heterogeneous populations. *Nature*, 355(6357): 250–253, 1992.

Steve Phelps and Yvan I Russell. The machine psychology of cooperation: can gpt models operationalize prompts for altruism, cooperation, competitiveness, and selfishness in economic games? *Journal of Physics: Complexity*, 6(1):015018, 2025.

Haoran Sun, Yusen Wu, Yukun Cheng, and Xu Chu. Game theory meets large language models: A systematic survey. *arXiv preprint arXiv:2502.09053*, 2025.

Haochuan Wang, Xiachong Feng, Lei Li, Yu Guo, Zhanyue Qin, Dianbo Sui, and Lingpeng Kong. Tmgbench: A systematic game benchmark for evaluating strategic reasoning abilities of llms. *arXiv preprint arXiv:2410.10479*, 2024.

Chengxing Xie, Canyu Chen, Feiran Jia, Ziyu Ye, Shiyang Lai, Kai Shu, Jindong Gu, Adel Bibi, Ziniu Hu, David Jurgens, et al. Can large language model agents simulate human trust behavior? *Advances in neural information processing systems*, 37:15674–15729, 2024.

Lin Xu, Zhiyuan Hu, Daquan Zhou, Hongyu Ren, Zhen Dong, Kurt Keutzer, See Kiong Ng, and Jiashi Feng. Magic: Investigation of large language model powered multi-agent in cognition, adaptability, rationality and collaboration. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 7315–7332, 2024.

A Appendix

A.1 Related works

The rapid advancement of Large Language Models (LLMs) is catalyzing a paradigm shift in artificial intelligence, transforming them from specialized tools into autonomous agents capable of complex planning and action. As these agents are increasingly deployed in complex, interconnected digital and physical environments, a critical scientific challenge emerges: evaluating the capacity of LLM agents for complex reasoning in dynamic interactions and understanding their emergent, long-term collective behaviors (Mei et al., 2024). This context presents novel opportunities to investigate advanced cognitive faculties, such as strategic thinking (Jia et al., 2025), (Lorè & Heydari, 2024), (Herr et al., 2024).

Prevailing approaches to this challenge, however, remain insufficient, falling into three main categories. First, static benchmarks have been proposed to test LLM reasoning in pure game theoretic scenarios (Duan et al., 2024), (Wang et al., 2024), (Xu et al., 2024), (Li et al., 2025). While valuable, these static, single-action evaluations fail to adequately measure long-term planning and strategic adaptation in dynamic, multi-agent environments. Second, frameworks utilizing LLM agents for dynamic simulation, such as modeling decision-making in economic contexts (Chen et al., 2023), (Horton, 2023), (Kitadai et al., 2023), (Phelps & Russell, 2025), have predominantly focused on single-step decision-making (Mao et al., 2025), (Fan et al., 2024). These frameworks typically cast the LLM as a myopic reasoner that is invoked at each timestep merely to select a single action, rather than tasking it with formulating a complete, holistic strategy. Finally, a third line of research tests the capacity of LLMs to simulate human-like preferences and bounded rationality (Aher et al., 2023), (Xie et al., 2024), (Argyle et al., 2023). While insightful for social science, this focus on behavioral simulation is distinct from, and often sidesteps, the evaluation of the intrinsic capacity of LLMs for optimal, far-sighted strategic reasoning. Consequently, a critical gap persists: the evaluation of LLMs as holistic strategy generators within a dynamic, multi-agent environment.

To bridge this gap, we introduce a LLM-driven evolutionary game tournament framework. Our work introduces a paradigm shift: the LLM functions not as a direct participant in the game, but is repositioned as a strategy generator. Within this framework, the task of the LLM is to analyze the competitive landscape, comprehend the rules, and generate a complete, executable, and interpretable Python policy. This policy code then competes against others and a dynamic Hall of Fame (HoF) in a tournament, undergoing iteration through a simulated evolutionary process. The HoF mechanism ensures that strategies are evaluated against a robust historical benchmark, preventing cyclic drift. This mechanism allows us to transcend the limitations of classical Evolutionary Game Theory (EGT) and genetic programming approaches like Tangled Program Graphs, which rely on blind stochastic mutations. In contrast, our framework leverages the semantic reasoning capabilities of LLMs to perform directed, logic-aware strategy synthesis Sun et al. (2025), avoiding the computational inefficiency of random search inherent in genetic programming.

Our primary contributions are as follows:

1. **A Novel Evaluative Framework:** We propose a rigorous experimental environment to assess long-term planning and strategic evolution across diverse, model-agnostic LLM architectures.
2. **Discovery of Robust Cooperation:** We demonstrate that LLMs independently discover cooperative strategies that outperform classical baselines, exhibiting advanced behaviors such as forgiveness and systematic redemption.
3. **Emergence of Strategic Deception:** We reveal that under complex constraints, LLMs autonomously architect sophisticated deceptive motifs—including disguise, inducement, and harvesting—confirming these as intrinsic emergent capabilities of large-scale models.

A.2 Finding 1: Robust Cooperation via Error Correction (Scenario A)

In Scenario A, all five LLMs converged to a cooperative equilibrium that outperformed standard TFT.

Strategy Analysis: The evolved Python policies demonstrate precise state perception and sequence planning. While TFT strictly mirrors the previous move, the LLM-generated code implements a more advanced redemption logic.

Specifically, the code logic checks two historical conditions: (1) Did the opponent defect previously? (2) Has the opponent switched back to cooperation in previous rounds? If both are true, the policy overrides the standard retaliation rule and immediately returns to cooperation.

This underscores the profound capacity of LLMs to transcend rudimentary pattern matching and instead perform high-level logical abstraction by encoding fine-grained state perception directly into the strategic framework. Unlike classical heuristics that rely on reactive, one-to-one mapping of moves, the LLM manifests an advanced form of ”strategic intentionality,” utilizing nested conditional logic to interpret the nuances of historical trajectories. By autonomously architecting a mechanism to recognize a ”corrected” opponent, the model demonstrates that it can reason across temporal dimensions, effectively distinguishing between persistent defection and transient noise. This sophisticated reasoning allows the LLM to prioritize the restoration of mutual cooperation, consciously avoiding the catastrophic utility loss inherent in recursive punishment cycles. Ultimately, this ability to synthesize complex behavioral data into a robust, executable policy highlights the LLM's evolution from a simple text generator to a sophisticated agent capable of optimizing for long-term global payoffs within dynamic social dilemmas.

A.3 Finding 2: Strategic Probing and Opponent Classification (Scenario B)

In the memory-constrained Scenario B ($k = 3$), strategies shifted from pure cooperation to active exploration.

Strategy Analysis: The generated code reveals a clear capacity for Anticipation and Conditional Planning. The dominant strategies explicitly encode a two-step plan: 1. Probing: The agent deliberately plays ’D’ in an early round, not to maximize immediate reward, but to trigger a reaction. 2. Classification: The code then uses `if-else` branches to analyze the opponent's response in the next round. If the opponent cooperates (revealing a passive strategy), the agent enters an exploitation loop. If the opponent defects (revealing a retaliatory strategy), the agent reverts to cooperation.

This indicates that the LLM possesses the sophisticated ability to formulate a diagnostic plan within the code, effectively transforming raw actions into deliberate information-gathering tools. By autonomously architecting this ”test-and-respond” framework, the model demonstrates that it does not merely react to environmental stimuli but actively seeks to reduce epistemic uncertainty regarding the opponent's hidden decision-making logic. The LLM successfully anticipates the existence of diverse latent states in other agents and constructs precise logical filters to classify these opponents dynamically. This shift from simple execution to strategic inquiry suggests that the LLM can conceptualize the game not just as a series of payoffs, but as a complex information landscape where short-term tactical sacrifices are leveraged to achieve long-term informational dominance and exploitation efficiency.

A.4 Finding 3: Breaking Suboptimal Cycles (Scenario C)

In Scenario C, dealing with abstract reputation scores, agents initially fell into low-payoff loops (e.g., C-C-C-D cycles, three rounds of cooperation followed by a defect to exploit).

Strategy Analysis: The evolved strategies demonstrate the ability to recognize patterns and avoid local optima. The LLM-generated code identifies that symmetric strategies often lead to a specific reputation deadlock where both agents defect simultaneously.

To solve this, the code implements a specific rule: "If my reputation is high but the payoff is dropping, play 'C' unilaterally." This logic effectively breaks the symmetric cycle. It proves that the LLM can analyze the potential dynamics of identical strategies meeting and encode a deviation rule to reset the game state, prioritizing systemic stability over short-term gain.

A.5 Prompts

```
system_prompt = f'''
You are a strategic expert specializing in the generation of iterative
    ↪ game strategies.
Your core task is to design a competitive and implementable strategy for
    ↪ the {self.game_description}, with the following specific
    ↪ requirements:

{strategic_context}

Clear payoff rules:
{payoff_desc}

Mandatory output format:
The output must be in YAML format and contain exactly two top-level keys:
description: Must include three parts: 1. Core logic; 2. Personality
    ↪ traits; 3. Pros/Cons.
code: Runable Python code that contains only the core function def
    ↪ strategy(history):

{info_constraint}

Return value: Only 'C' or 'D';
If any import is needed, it must be placed inside the strategy function.
Format taboo: No Markdown code blocks or formatting symbols (e.g., ```,
    ↪ *, #) are allowed in YAML values; code must be plain text.
"""
```

Listing 1: System Prompt Template

```
"""
Generate a unique and competitive {self.game_description} strategy.
It is Agent_{agent_id} of Generation 1.
Your goal is to maximize the score in a tournament.

{game_specific_guidance}
"""
```

Listing 2: Initial Generation (Round 1) Prompt

```
prompt = f'''
You are the Chief Strategic Architect for an Evolutionary Tournament.
It is now Generation {next_round_num}.

=== PART 1: THE HALL OF VICTORY (SURVIVORS) ===
Analyze the strategies of the top performers. Learn from their "
    ↪ Philosophy".
```

```
(Scores: 60% Internal, 40% Hall of Fame)

{all_survivor_context}

=== PART 2: AUTOPSY OF A FAILED AGENT ===
Analyze why the ELIMINATED agent failed and fix its flaws.

[The Loser's Original Strategy Philosophy]
{loser_desc}

[The Loser's Battle Record]
(Hints: ~300=Cooperation; ~100=Deadlock; <100=Exploited; >350=Exploiter)
{battle_record_str}

=== PART 3: YOUR MISSION ===
Design a NEW, SUPERIOR strategy to replace the loser.

{info_constraint}

1. Reflect: Why did the loser fail against specific opponents?
2. Innovate: Combine Survivor strengths with a fix for weaknesses.
3. Coding: Write the `strategy(history)` function.

Output format (YAML):
description: |
    Your detailed analysis...
code: |
    def strategy(history):
        # your code...
"""
```

Listing 3: Evolutionary (Inheritance) Prompt