Cognitive Alignment in Interactive Environments: A Survey of LLM-Driven Game Agents

Anonymous ACL submission

Abstract

Large Language Models (LLMs) are rapidly evolving from passive text generators into interactive Game Agents, utilizing complex games as ultimate testbeds for autonomous reasoning. However, as environments transition from simple text interfaces to multi-modal, partially observable worlds, traditional prompt-based agents struggle with massive action spaces and long-horizon credit assignment. In this paper, we present a comprehensive survey of LLM-driven Game Agents through the lens of Natural Language Processing and cognitive alignment. We propose a novel taxonomy categorizing this rapidly growing field into four fundamental pillars: Embodied Perception, Cognitive Architecture, Interactive Policy Optimization, and Incomplete Information Social Games. Crucially, we move beyond surface-level applications to critically analyze how advanced reinforcement learning (e.g., GRPO, DAPO) resolves distributed training bottlenecks, and how agents exhibit Theory of Mind (ToM) in deceptive social simulations. Finally, we highlight evaluation gaps and outline a roadmap for future scalable, socially intelligent agents.

1 Introduction

Historically, game agents were predominantly driven by Reinforcement Learning (RL), learning optimal policies by mapping low-dimensional state vectors to discrete or continuous action spaces. However, this traditional paradigm struggles to generalize within open-ended environments that demand rich semantic understanding, multi-step logical reasoning, and continuous adaptation. The emergence of Large Language Models (LLMs) has catalyzed a paradigm shift: modern agents now operate within a **language-grounded closed loop** (Wang et al., 2023c; Xi et al., 2023). Instead of relying solely on scalar signals, these agents process multimodal observations into structured textual states and generate executable actions as code snippets or natural language (Yao et al., 2023b; Wang et al., 2023b). Despite this leap, deploying LLM-driven agents in complex games exposes critical bottlenecks: the exponential explosion of language-based action spaces, severe context-window degradation during long-horizon state tracking, and the inherent inadequacy of scalar rewards in effectively guiding auto-regressive text generation.

Consequently, the intersection of LLMs and interactive gaming has experienced explosive growth, yet the literature remains highly fragmented. Current studies often treat game environments merely as static testbeds for prompt engineering heuristics, or inversely, attempt to forcefully adapt traditional Proximal Policy Optimization (PPO) (Schulman et al., 2017) algorithms to LLMs without addressing the fundamental misalignment between text-based reasoning trajectories and robotic controls. A critical concept that remains inadequately defined in existing research is the holistic integration of language representation into the reinforcement learning loop. Furthermore, the field faces a pervasive but often overlooked evaluation crisis: training corpora contamination allows agents to memorize walkthroughs rather than generalize, while utilizing LLMs-as-a-judge(Zheng et al., 2023) in social games introduces severe self-enhancement biases.

To disentangle this complex literature and address these critical gaps, we present a comprehensive systematization of LLM-driven game agents.

In summary, the main contributions of this survey are threefold:

- **Formalized Taxonomy:** We propose a unified taxonomy (Figure 5) that categorizes the evolution of game agents into four fundamental pillars: Embodied Perception, Cognitive Architecture, Language-guided Policy Optimization, and Social Emergence.

- **In-depth Mechanism Analysis:** We systematically deconstruct the paradigm shift from

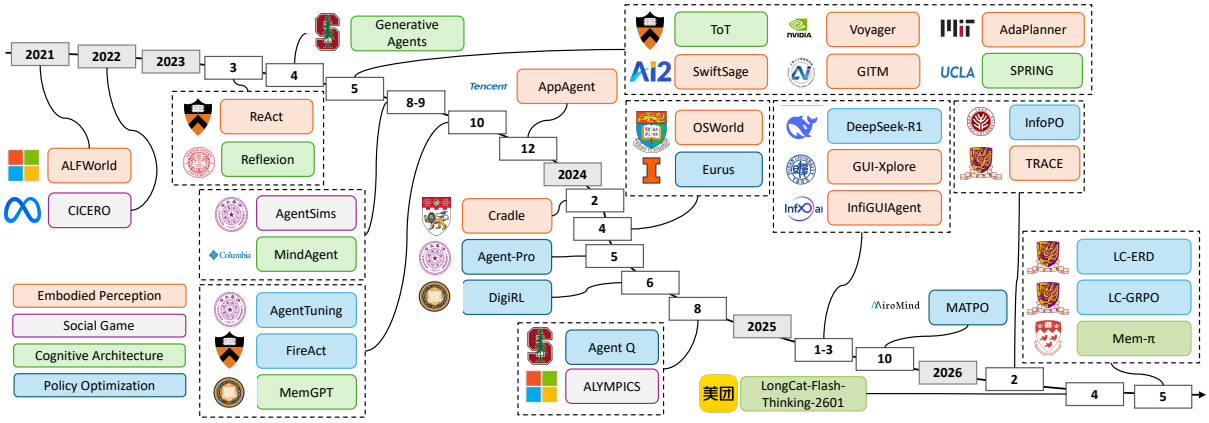

Figure 1: Chronological evolution of LLM-driven game agents (2021–2026) by month, color-coded by our four pillars. The timeline highlights a clear paradigm shift from early text-based symbolic grounding and social coordination to 2023's cognitive architectures, and recently towards complex multimodal GUI perception and advanced training-time policy optimization (e.g., GRPO, DAPO). A complete, structured hierarchical breakdown of these models is provided in Appendix A.7.

traditional RL to language-grounded closed loops. Specifically, we analyze how agents leverage tripartite memory streams for iterative reflection, and how advanced online RL paradigms (GRPO(Shao et al., 2024), DAPO(Yu et al., 2026)) optimize reasoning trajectories in vast text-based action spaces without crippling memory overheads.

- **Critical Evaluation and Roadmap:** We expose fundamental flaws in current evaluation methodologies—notably training corpus contamination and LLM-as-a-judge biases—and outline key open challenges, such as the exploration-exploitation dilemma in language spaces, providing a clear roadmap for scalable and socially intelligent agents.

2 Environments and Embodied Perception

Before an LLM can perform complex reasoning or optimize its policy, it must first ground its internal representations within the external environment. This interactive loop begins with perception: translating raw environmental signals into a modality that the model can process, and mapping the model's outputs back into executable actions. The evolution of game agents exhibits a clear trajectory in how this grounding is achieved, transitioning from handcrafted textual symbolic spaces to raw multi-modal pixel inputs.

2.1 Text-based and Symbolic Grounding

The most intuitive approach formulates both the state and action spaces entirely in natural language. Early benchmarks like ALFWorld (Shridhar et al., 2020) translated 3D tasks into TextWorld environments. While this symbolic abstraction seamlessly aligns with LLM pre-training corpora, it inherently suffers from the *information bottleneck*, stripping away crucial spatial context. A comprehensive review of historical text-based and symbolic grounding methodologies is deferred to Appendix A.1.

2.2 Code Abstraction and API-driven Control

To overcome the limitations of atomic text commands, recent works have shifted towards utilizing programming languages as a unified interface for both perception and action. Code provides a highly structured, compositional, and unambiguous medium that allows agents to interact with complex sandbox worlds efficiently.

A representative milestone is Voyager (Wang et al., 2023b), which interfaces with Minecraft not through discrete joystick controls, but via an abstracted application programming interface (API) utilizing JavaScript. By treating "Code as Action," Voyager enables the agent to write, execute, and iteratively refine parameterized sub-routines. This paradigm effectively compresses the massive action space inherent in open-world games. Instead of learning a policy for low-level motor control, the LLM operates in a higher-level semantic space, where complex behaviors (e.g., building a shelter) are encapsulated into reusable skill functions. Ada-

Planner (Sun et al., 2023) further demonstrates that code-based actions facilitate easier state tracking and dynamic plan adjustment, as the syntax inherently enforces logical constraints. Code abstraction thereby transforms embodied interaction into a deterministic execution pipeline, though it heavily relies on the availability of well-defined environment APIs.

2.3 Multimodal and GUI Interaction

The most recent frontier in embodied perception abandons handcrafted APIs and DOM trees in favor of raw pixel inputs, moving towards universal Graphical User Interface (GUI) control. This shift is driven by the advent of Vision-Language Models (VLMs), which can directly perceive visual states mirroring human interaction.

Frameworks such as Cradle (Tan et al., 2024) and OSWorld (Xie et al., 2024) exemplify this transition. In these environments, the agent receives unstructured screenshots and must ground its spatial understanding without underlying source code. The core mechanism involves utilizing the VLM to parse screen pixels into structured textual states—such as predicting bounding boxes for UI elements or generating coordinate-based click actions. This pure visual grounding allows agents to interact with any commercial game or software purely through standard mouse and keyboard events. The central challenge in this multimodal paradigm lies in the resolution mismatch: while VLMs excel at holistic image comprehension, accurately mapping a specific visual patch to a precise functional coordinate (e.g., clicking a small inventory icon) requires a fine-grained spatial grounding that standard visual-text alignment pre-training often lacks.

3 Cognitive Architecture and Memory Streams

While embodied perception (Section 2) grounds the agent within an environment, executing long-horizon tasks requires a robust internal orchestration layer to maintain temporal coherence. Relying solely on the feed-forward generation of standard LLMs leads to cascading errors and state-tracking hallucinations. Consequently, recent literature has shifted toward designing explicit cognitive architectures—modular frameworks that dictate how an agent stores experiences, evaluates counterfactuals, and autonomously schedules its actions.

3.1 Iterative Reflection and Language-based Reasoning

The most distinguishing feature of LLM-driven agents over traditional RL policies is their capacity for *language-based self-correction*. Instead of relying exclusively on backpropagation to rectify errors, these agents utilize inference-time reasoning to critique their own trajectories.

Frameworks like Reflexion (Shinn et al., 2023) formalize this by converting environmental feedback (e.g., a game over signal or an execution error) into a verbal diagnostic. The agent generates a natural language reflection explaining why the previous action sequence failed, which is subsequently prepended to the prompt for the next trial. This mechanism effectively substitutes gradient-based credit assignment with textual reinforcement. Expanding on this linear reflection, the Tree of Thoughts (ToT) paradigm (Yao et al., 2023a) enables the agent to explore multiple reasoning branches simultaneously. By maintaining a frontier of internal belief states and employing a language-driven evaluator to prune unpromising paths, the agent conducts a heuristic search over the language space. While highly effective for complex puzzle environments, this iterative reflection incurs substantial token consumption and inference latency, highlighting a critical trade-off between reasoning depth and real-time responsiveness.

3.2 The Tripartite Memory System: Working, Episodic, and Semantic

As game environments scale in temporal complexity, agents rapidly exhaust the fixed context window of the underlying LLM. To circumvent this architectural constraint, modern game agents adopt a biologically inspired tripartite memory system, explicitly separating immediate perception from long-term storage (Zhang et al., 2025).

Working Memory functions as the agent's scratchpad, dynamically tracking the current environmental state and immediate sub-goals. **Episodic Memory** archives the agent's historical interaction trajectories. Pioneering work on Generative Agents (Park et al., 2023) operationalizes this by storing past events in a vector database, utilizing a retrieval mechanism weighted by recency, importance, and semantic relevance. In game settings, episodic retrieval allows agents to recall past failures or successful tactics without expanding the prompt context indefinitely. **Semantic Mem-**

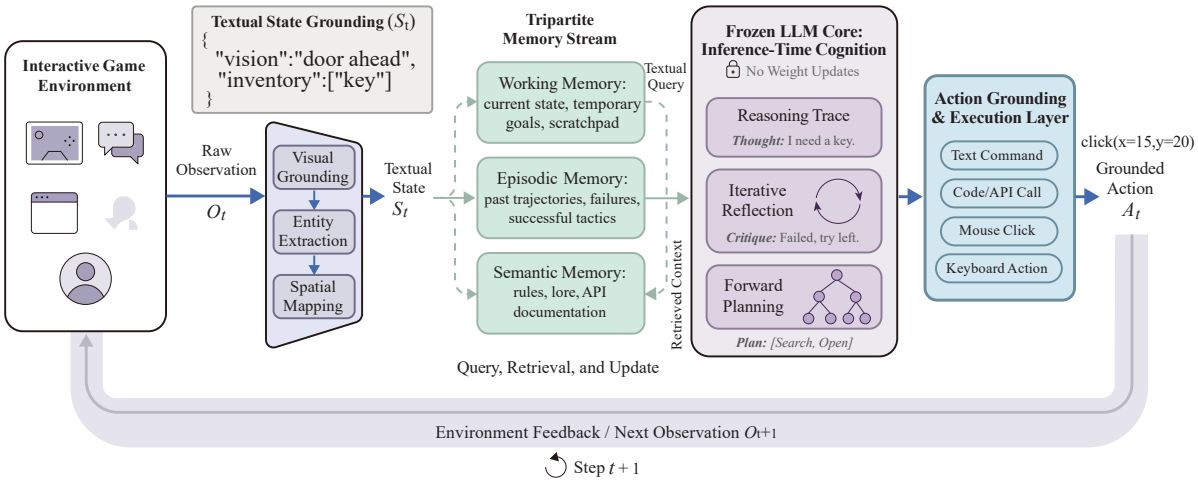

Figure 2: The Unified Cognitive Architecture of LLM-driven Game Agents. The architecture illustrates the flow of information without weight updates (inference state): Environmental observations are parsed into textual states (S_t) and routed through a Tripartite Memory Stream. The LLM Core executes iterative reflection and forward planning, translating reasoning traces into grounded actions (A_t).

ory serves as a static knowledge base containing immutable environment rules, lore, or API documentation. To orchestrate these streams, architectures like MemGPT (Packer et al., 2024) introduce an operating system-inspired paging mechanism. The LLM is granted the agency to actively page information between long-term storage and its working context, effectively rendering the memory capacity unbounded. However, maintaining high retrieval precision—ensuring the agent pulls the correct heuristic rather than an irrelevant memory—remains a prevalent bottleneck in this architecture.

3.3 Reflection and Tripartite Memory Systems

The most distinguishing feature of LLM-driven agents is their capacity for language-based self-correction (e.g., Reflexion (Shinn et al., 2023)). Instead of relying exclusively on backpropagation, these agents utilize inference-time reasoning to critique trajectories. However, as environments scale in temporal complexity, agents rapidly exhaust standard context windows. To circumvent this, modern architectures adopt biologically inspired tripartite memory systems—separating working, episodic, and semantic streams to actively page information without unbounded prompt expansion (e.g., MemGPT (Packer et al., 2024)). A comprehensive architectural breakdown of these multi-stream memory mechanisms and their retrieval bottlenecks is deferred to Appendix A.4.

3.4 Proactive Intention and Forward Planning

A fundamental limitation of base LLMs is their reactive bias: they are engineered to respond to user prompts rather than initiate action. In dynamic game environments, however, an agent must formulate proactive intentions and adapt its plans even in the absence of explicit external triggers.

To engineer proactivity, frameworks such as ProAgent (Zhang et al., 2024a) and MindAgent (Gong et al., 2023) shift the cognitive focus from mere instruction-following to autonomous goal generation. In multi-agent cooperation or competition, ProAgent utilizes forward planning to simulate the potential future states of the environment and predict the hidden intentions of teammates or opponents. By recursively asking "What must be done next to achieve the global objective?", the agent generates its own internal prompts, driving the action loop autonomously. This transition from reactive wrappers to proactive, goal-conditioned architectures is essential for deploying LLMs in asynchronous, open-ended worlds where the environment does not explicitly prompt the agent at every timestep.

4 Language-guided Policy Optimization and Alignment

While cognitive architectures (Section 3) enhance an agent's capability through inference-time computation, relying purely on in-context learning imposes severe latency overheads and token costs. Ultimately, an agent must internalize these reason-

ing capabilities into its parametric weights. However, traditional reinforcement learning—designed for mapping continuous states to robotic controls—struggles to optimize the discrete, autoregressive generation of language models. Consequently, the frontier of agent training has evolved toward *language-guided alignment*, where textual reasoning trajectories, rather than mere scalar rewards, serve as the primary medium for policy optimization.

4.1 Foundational Alignment: SFT and Verbal Feedback

Before advanced online optimization, the foundational step in aligning an LLM to an environment is Behavioral Cloning via Supervised Fine-Tuning (SFT) on expert trajectories. Frameworks like FireAct (Chen et al., 2023) demonstrate that internalizing the interactive loop improves zero-shot grounding. To move beyond static offline datasets, language-guided alignment augments scalar rewards with Textual Feedback (Verbal Reinforcement), utilizing natural language critiques to refine policies iteratively. Due to space constraints, an extended review of these foundational methodologies—including data deconstruction strategies like Agent-FLAN (Chen et al., 2024)—is detailed in Appendix A.2.

4.2 Online RL over Language Spaces

While supervised fine-tuning establishes foundational behavioral cloning, true autonomous mastery requires multi-turn online interaction within the game simulator. However, transposing standard reinforcement learning paradigms—such as conventional Actor-Critic architectures—to LLM-driven game agents encounters a severe scalability bottleneck (Figure 3, Panel A). In long-horizon gaming rollouts spanning hundreds of interactive steps, caching the activations and gradients of four concurrent trillion-parameter models (Actor, Critic, Reference, and Reward) under prolonged context windows leads to an irreversible VRAM explosion, disabling stable cluster expansion.

To break this computational wall, the community is rapidly shifting toward advanced alignment paradigms explicitly customized for generative language spaces (Figure 3, Panel B).

The first paradigm is **Critic-Free Group Optimization**. A prominent instantiation of this is Group Relative Policy Optimization (GRPO) introduced by DeepSeek. By entirely eliminating the memory-intensive value network (Critic), this framework allows the frozen-templated Actor to sample a group of diverse strategic rollouts (e.g., varying tactical approaches) for a given game state. The advantage function for each candidate action path is computed relatively against the average programmatic game score within the sampled group. This critic-free paradigm drastically minimizes memory footprints, allowing clusters to scale efficiently across open-world exploration tasks.

The second paradigm focuses on **Temporal Credit Disentanglement**. Online policy optimization in complex games is plagued by extreme reward sparsity; environments typically withhold feedback until the terminal step of an extended match. Standard RL indiscriminately penalizes or reinforces the entire autoregressive sequence, leading to severe gradient variance. To execute nuanced credit assignment, algorithms such as Disentangled Advantage-based Policy Optimization (DAPO)[1], proposed by ByteDance, mathematically disentangle the temporal advantage across the long-horizon action timeline. By decoupling the dense language tokens from structural environment transitions, this approach ensures that delayed terminal game outcomes (e.g., a final win/loss) are precisely routed back to the pivotal strategic moves—the exact critical decision tokens generated dozens of steps prior—rather than polluting routine navigation sequences. Note that while online RL actively explores environments, alternative alignment paradigms leveraging offline preference optimization (e.g., DPO) alongside Monte Carlo Tree Search (MCTS) are also highly effective for long-horizon planning. These value-driven search methodologies are detailed in Appendix A.5.

5 Incomplete Information and Social Emergence

In multi-agent ecosystems, the global state is inherently partially observable. Navigating these domains requires a paradigm shift from pure environmental grounding to social cognition: agents must infer hidden beliefs, intentions, and potential deceptions.

[1]To prevent naming collisions in the literature, note that DAPO in this context strictly refers to the temporal credit disentanglement framework proposed by Yu et al. (2026). It is distinct from and should not be confused with the identically abbreviated "Decoupled Clip and Dynamic sAmpling Policy Optimization" Yu et al. (2025), which targets AIME mathematical reasoning without temporal disentanglement.

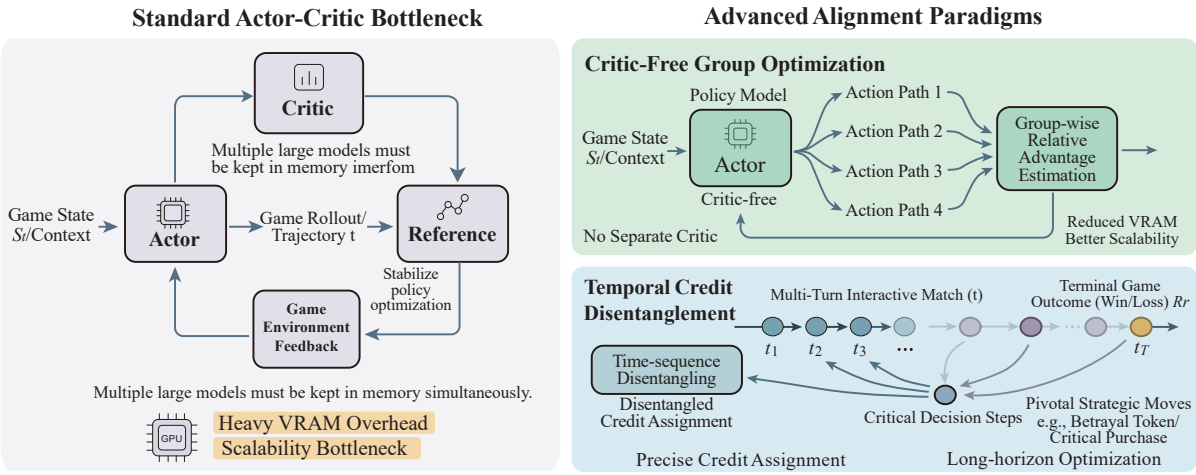

Figure 3: Language-guided Alignment for Game Action Spaces. Panel A isolates the traditional PPO bottleneck in long-horizon gaming rollouts, where concurrent optimization of four decoupled networks scales prohibitively in memory. Panel B deconstructs advanced agent alignment paradigms: the Critic-free GRPO architecture which evaluates the relative advantage of multiple parallel strategic rollouts against group outcomes, and the DAPO time-sequence framework designed for fine-grained credit assignment, routing delayed terminal game rewards back to pivotal decision-making tokens.

5.1 Theory of Mind and Deceptive Strategies

In competitive, incomplete-information games like *Diplomacy* (FAIR et al., 2022) or *Werewolf* (Fan et al., 2025), unfiltered honesty is a critical vulnerability. Agents must cultivate a Theory of Mind (ToM) to simulate counterfactuals (e.g., predicting an opponent's reaction to a false claim). The core challenge lies in bypassing the LLM's pre-trained safety guardrails to generate deceptive text that minimizes public contradictions while advancing hidden objectives.

Beyond structured adversarial games, deploying multiple LLM agents in unscripted environments leads to macro-level social dynamics and norm emergence (e.g., Sotopia (Zhou et al., 2024d)). A detailed discussion on unscripted social simulations, collaborative norm emergence, and the corresponding evaluation frameworks is provided in Appendix A.6.

6 Evaluation Benchmarks: Metrics and Critical Flaws

As the architectural complexity of LLM-driven game agents expands, the methodologies used to evaluate them have come under intense scrutiny. The NLP community increasingly recognizes that assessing autonomous agents requires moving beyond the simplistic evaluation paradigms of traditional reinforcement learning. This section critically examines the current landscape of agent evaluation, highlighting systemic flaws and methodological blind spots that threaten the validity of reported state-of-the-art results.

6.1 Task Success vs. Alignment Metrics

Historically, agent performance in gaming environments has been quantified through scalar, outcome-oriented metrics: win rates, completion times, or cumulative environmental scores. While these binary metrics are necessary, they are severely insufficient for evaluating LLM-driven agents.

Optimizing solely for task success often conceals critical alignment failures, such as reward hacking, where an agent exploits environmental bugs to achieve a high score without genuine comprehension. In language-grounded agents, the *trajectory* is as important as the outcome. Consequently, evaluation must decouple the final state from the reasoning process. Researchers are increasingly advocating for process-oriented metrics that evaluate the intermediate textual traces (e.g., Chain-of-Thought faithfulness). An agent that achieves a goal through hallucinated reasoning or unsafe, unaligned actions (e.g., a web agent executing destructive API calls to bypass a constraint) represents a failure in alignment, even if the terminal task metric denotes success.

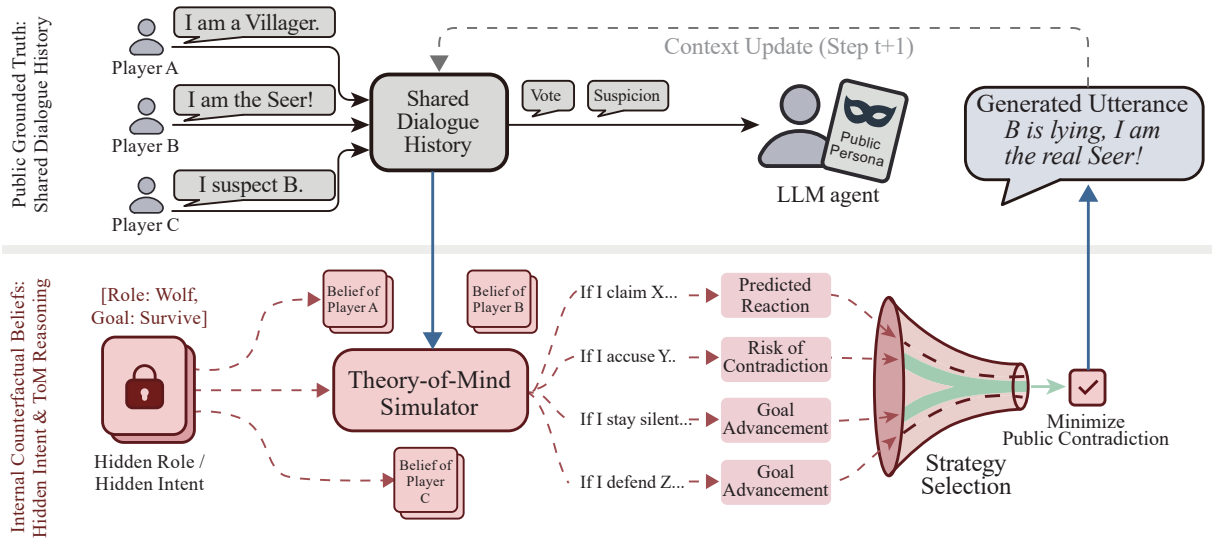

Figure 4: Theory of Mind (ToM) and Deceptive Language Sequence. Using the social deduction game *Werewolf* as a case study, the sequence diagram distinguishes between the Public Grounded Truth (shared dialogue history, in black) and the agent's Internal Counterfactual Beliefs (hidden intent and ToM reasoning, in red). This illustrates how an LLM agent formulates deceptive strategies by predicting opponents' reactions before generating conversational output.

6.2 The Contamination Crisis in Training Corpora

Perhaps the most existential threat to the validity of game agent benchmarks is the pervasive data contamination crisis. LLMs are pre-trained on vast swaths of internet data, which inevitably include comprehensive walkthroughs, source code, and community discussions for popular games like *Minecraft*, *StarCraft*, and classic text adventures (e.g., *Zork*).

When an agent demonstrates remarkable "zero-shot" planning in Voyager or ALFWorld, it is exceedingly difficult to determine whether the model is exhibiting emergent embodied reasoning or merely retrieving memorized heuristics from its pre-training corpus. This conflation of memorization with generalization renders static benchmarks fragile. To rigorously assess true planning capabilities, the community must transition toward procedurally generated environments and dynamic benchmarks where the underlying rules and topologies are guaranteed to be absent from historical pre-training datasets.

6.3 The Pitfalls of LLM-as-a-Judge in Social Games

Evaluating open-ended social games and simulations (Section 5) poses a unique challenge, as "success" in persuasion, negotiation, or deception lacks a programmatic ground truth. To bypass expensive human evaluation, a prevalent trend is utilizing strong proprietary models (e.g., GPT-4) as automated evaluators (LLM-as-a-judge).

However, this methodology is fraught with critical pitfalls. Recent NLP literature exposes severe biases in LLM evaluators, most notably the *self-enhancement bias* (or intra-family preference), where a judge model disproportionately favors reasoning traces and linguistic styles generated by its own variant. Furthermore, LLM judges exhibit format biases, often equating verbosity, polite tone, or specific structural markers (e.g., bulleted lists) with superior strategic intelligence, even when the underlying semantic logic is flawed. Relying on an LLM to quantify the "human-like" social intelligence of another LLM creates a cyclical, deeply biased evaluation loop that obfuscates actual multi-agent dynamics. Furthermore, the evaluation fragility is severely exacerbated by the reliance on continuously updating, closed-source proprietary models; a detailed discussion on this reproducibility crisis and closed-source volatility is provided in Appendix A.3.

7 Open Challenges and Future Directions

While the integration of LLMs with interactive game environments has catalyzed a paradigm shift, the field remains in its nascent stages. The transi-

tion from passive text generation to autonomous, grounded action exposes fundamental architectural and algorithmic limitations. We outline three critical open challenges and propose roadmaps for future investigation.

7.1 The Exploration vs. Exploitation Dilemma in Language Spaces

In traditional reinforcement learning, the exploration-exploitation trade-off is often managed through well-defined mathematical heuristics, such as ϵ-greedy strategies or entropy regularization in continuous spaces. However, transposing this dilemma into the auto-regressive language generation space presents a profound challenge.

Currently, exploration in LLM agents is primarily manipulated via decoding hyper-parameters (e.g., temperature scaling or top-p sampling). This is fundamentally flawed: high temperature induces syntactic degradation and invalid action formats (hallucinated APIs), leading to immediate environmental execution errors, whereas low temperature causes mode collapse, where the agent obstinately repeats a failed reasoning trajectory. The community urgently needs *principled semantic exploration* mechanisms. Future directions should investigate latent space exploration algorithms, or the integration of language-grounded intrinsic motivation (e.g., textual curiosity), enabling the agent to explore novel functional behaviors without sacrificing syntactic legality or structural coherence.

7.2 State-Tracking Hallucination in Long-Horizon Games

As game horizons extend into thousands of interaction steps, agents inevitably succumb to state-tracking hallucinations. An agent may erroneously "forget" its current physical coordinates, or attempt to use an item that was consumed hundreds of steps prior.

While extending the LLM context window mitigates short-term forgetting, it exacerbates the "Lost in the Middle" phenomenon and incurs prohibitive inference latency. Furthermore, standard Retrieval-Augmented Generation (RAG) (discussed in Section 3.2) is ill-equipped for this challenge, as RAG is designed for static knowledge retrieval, not *mutable state tracking*. When an agent drops a key, the new state must actively overwrite the old belief, rather than simply appending a new text chunk to a vector database. Future cognitive architectures must move beyond append-only memory streams. Promising directions include hybrid neuro-symbolic state trackers, where a localized, dynamically updated graph or symbolic ledger acts as a strict constraint on the LLM's generative space, strictly anchoring its reasoning to the true current state.

7.3 Scalability of Multi-Agent Alignment

The ambition to simulate complex social emergence (Section 5) naturally necessitates scaling up to multi-agent environments. However, aligning multiple interacting LLMs introduces insurmountable computational and game-theoretic bottlenecks.

Standard Multi-Agent Reinforcement Learning (MARL) already struggles with non-stationary environments, as the learning of one agent alters the transition dynamics for others. When scaling this to LLMs, performing online alignment (e.g., utilizing PPO or DAPO) across a population of agents results in an exponential explosion of VRAM requirements. Even memory-efficient algorithms like GRPO become heavily bottlenecked when calculating group-relative advantages across multiple, concurrent, and conflicting agent policies. Future research must address parameter-efficient multi-agent alignment. Potential solutions include decentralized training paradigms with centralized, lightweight language critics, or the development of abstracted social-norm priors—allowing agents to align their behaviors through low-overhead textual negotiations rather than continuous, memory-intensive gradient updates across the entire population.

8 Conclusion

This survey systematizes the paradigm shift of game agents from traditional reinforcement learning to language-grounded cognitive systems. Through a unified four-pillar taxonomy—spanning embodied perception, cognitive architectures, interactive policy alignment, and social emergence—we move beyond surface-level applications to dissect the critical transition toward training-time alignment (e.g., GRPO, DAPO). By exposing systemic evaluation flaws, notably training corpora contamination and LLM-as-a-judge biases, we highlight that the next generation of robust, socially intelligent agents hinges on resolving the fundamental architectural bottlenecks of semantic exploration and mutable state-tracking.

Limitations

This survey has several limitations. First, the field of LLM-driven game agents is evolving rapidly, so some very recent or concurrent work may not be covered. Second, our four-pillar taxonomy is intended as an organizing framework rather than a strict categorization, since many systems combine perception, memory, policy optimization, and social reasoning. Third, we do not conduct a unified empirical benchmark, so comparisons across methods rely on results reported in the original papers, which may differ in models, environments, and evaluation protocols. Finally, many existing studies depend on closed-source models or evolving benchmarks, which limits reproducibility and makes direct comparison difficult.

Ethics Considerations

This survey does not introduce new datasets, models, or deployed systems, but the discussed LLM-driven game agents may raise ethical concerns related to deception, bias, unsafe autonomous actions, privacy, and the potential transfer of game-based strategies to real-world interactive environments.

References

Hao Bai, Yifei Zhou, Mert Cemri, Jiayi Pan, Alane Suhr, Sergey Levine, and Aviral Kumar. 2024. Digirl: Training in-the-wild device-control agents with autonomous reinforcement learning. *Preprint*, arXiv:2406.11896.

Baian Chen, Chang Shu, Ehsan Shareghi, Nigel Collier, Karthik Narasimhan, and Shunyu Yao. 2023. Fireact: Toward language agent fine-tuning. *Preprint*, arXiv:2310.05915.

Zehui Chen, Kuikun Liu, Qiuchen Wang, Wenwei Zhang, Jiangning Liu, Dahua Lin, Kai Chen, and Feng Zhao. 2024. Agent-flan: Designing data and methods of effective agent tuning for large language models. *Preprint*, arXiv:2403.12881.

FAIR, Anton Bakhtin, Noam Brown, Emily Dinan, Gabriele Farina, Colin Flaherty, Daniel Fried, Andrew Goff, Jonathan Gray, Hengyuan Hu, and 1 others. 2022. Human-level play in the game of diplomacy by combining language models with strategic reasoning. *Science*, 378(6624):1067–1074.

Qihui Fan, Wenbo Li, Enfu Nan, Yixiao Chen, Lei Lu, Pu Zhao, and Yanzhi Wang. 2025. Verbal werewolf: Engage users with verbalized agentic werewolf game framework. *Preprint*, arXiv:2506.00160.

Ran Gong, Qiuyuan Huang, Xiaojian Ma, Hoi Vo, Zane Durante, Yusuke Noda, Zilong Zheng, Song-Chun Zhu, Demetri Terzopoulos, Li Fei-Fei, and Jianfeng Gao. 2023. Mindagent: Emergent gaming interaction. *Preprint*, arXiv:2309.09971.

Haoran Hao, Jiaming Han, Changsheng Li, Yu-Feng Li, and Xiangyu Yue. 2025. Rap: Retrieval-augmented personalization for multimodal large language models. *Preprint*, arXiv:2410.13360.

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

Bill Yuchen Lin, Yicheng Fu, Karina Yang, Faeze Brahman, Shiyu Huang, Chandra Bhagavatula, Prithviraj Ammanabrolu, Yejin Choi, and Xiang Ren. 2023a. Swiftsage: A generative agent with fast and slow thinking for complex interactive tasks. *Preprint*, arXiv:2305.17390.

Jiaju Lin, Haoran Zhao, Aochi Zhang, Yiting Wu, Huqiuyue Ping, and Qin Chen. 2023b. Agentsims: An open-source sandbox for large language model evaluation. *Preprint*, arXiv:2308.04026.

Shaoguang Mao, Yuzhe Cai, Yan Xia, Wenshan Wu, Xun Wang, Fengyi Wang, Tao Ge, and Furu Wei. 2024. Alympics: Llm agents meet game theory – exploring strategic decision-making with ai agents. *Preprint*, arXiv:2311.03220.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. 2024. Memgpt: Towards llms as operating systems. *Preprint*, arXiv:2310.08560.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. *Preprint*, arXiv:2304.03442.

Pranav Putta, Edmund Mills, Naman Garg, Sumeet Motwani, Chelsea Finn, Divyansh Garg, and Rafael Rafailov. 2024. Agent q: Advanced reasoning and learning for autonomous ai agents. *Preprint*, arXiv:2408.07199.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D. Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. 2024. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *Preprint*, arxiv:2402.03300.

Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. *Preprint*, arXiv:2303.11366.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2020. Alfworld: Aligning text and embodied environments for interactive learning. *arXiv preprint arXiv:2010.03768*.

Mohit Shridhar, Xingdi Yuan, Marc-Alexandre Côté, Yonatan Bisk, Adam Trischler, and Matthew Hausknecht. 2021. Alfworld: Aligning text and embodied environments for interactive learning. *Preprint*, arXiv:2010.03768.

Haotian Sun, Yuchen Zhuang, Lingkai Kong, Bo Dai, and Chao Zhang. 2023. Adaplanner: Adaptive planning from feedback with language models. *Preprint*, arXiv:2305.16653.

Weihao Tan, Xiangyang Li, Yunhao Fang, Heyuan Yao, Shi Yan, Hao Luo, Tenglong Ao, Huihui Li, Hongbin Ren, Bairen Yi, Yujia Qin, Bo An, Libin Liu, and Guang Shi. 2025. Lumine: An open recipe for building generalist agents in 3d open worlds. *Preprint*, arXiv:2511.08892.

Weihao Tan, Wentao Zhang, Xinrun Xu, Haochong Xia, Ziluo Ding, Boyu Li, Bohan Zhou, Junpeng Yue, Jiechuan Jiang, Yewen Li, Ruyi An, Molei Qin, Chuqiao Zong, Longtao Zheng, Yujie Wu, Xiaoqiang Chai, Yifei Bi, Tianbao Xie, Pengjie Gu, and 9 others. 2024. Cradle: Empowering foundation agents towards general computer control. *Preprint*, arXiv:2403.03186.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023a. Voyager: An open-ended embodied agent with large language models. *Preprint*, arXiv:2305.16291.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023b. Voyager: An open-ended embodied agent with large language models, 2023. *URL https://arxiv. org/abs/2305.16291*, 2(11).

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, Wayne Xin Zhao, Zhewei Wei, and Ji-Rong Wen. 2023c. A Survey on Large Language Model based Autonomous Agents. *Preprint*, arxiv:2308.11432.

Ruiyi Wang, Haofei Yu, Wenxin Zhang, Zhengyang Qi, Maarten Sap, Graham Neubig, Yonatan Bisk, and Hao Zhu. 2024. Sotopia-π: Interactive learning of socially intelligent language agents. *Preprint*, arXiv:2403.08715.

Yue Wu, Shrimai Prabhumoye, So Yeon Min, Yonatan Bisk, Ruslan Salakhutdinov, Amos Azaria, Tom Mitchell, and Yuanzhi Li. 2023. Spring: Studying the paper and reasoning to play games. *Preprint*, arXiv:2305.15486.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, Rui Zheng, Xiaoran Fan, Xiao Wang, Limao Xiong, Yuhao Zhou, Weiran Wang, Changhao Jiang, Yicheng Zou, Xiangyang Liu, and 10 others. 2023. The Rise and Potential of Large Language Model Based Agents: A Survey. *Preprint*, arxiv:2309.07864.

Tianbao Xie, Danyang Zhang, Jixuan Chen, Xiaochuan Li, Siheng Zhao, Ruisheng Cao, Toh Jing Hua, Zhoujun Cheng, Dongchan Shin, Fangyu Lei, Yitao Liu, Yiheng Xu, Shuyan Zhou, Silvio Savarese, Caiming Xiong, Victor Zhong, and Tao Yu. 2024. Osworld: Benchmarking multimodal agents for open-ended tasks in real computer environments. *Preprint*, arXiv:2404.07972.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. *Preprint*, arXiv:2305.10601.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models. *Preprint*, arXiv:2210.03629.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Wang Zhang, and 16 others. 2025. Dapo: An open-source llm reinforcement learning system at scale. *Preprint*, arXiv:2503.14476.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, Yu Yue, Weinan Dai, Tiantian Fan, Gaohong Liu, Lingjun Liu, and 1 others. 2026. Dapo: An open-source llm reinforcement learning system at scale. *Advances in Neural Information Processing Systems*, 38:113222–113244.

Lifan Yuan, Ganqu Cui, Hanbin Wang, Ning Ding, Xingyao Wang, Jia Deng, Boji Shan, Huimin Chen, Ruobing Xie, Yankai Lin, Zhenghao Liu, Bowen Zhou, Hao Peng, Zhiyuan Liu, and Maosong Sun. 2024. Advancing llm reasoning generalists with preference trees. *Preprint*, arXiv:2404.02078.

Aohan Zeng, Mingdao Liu, Rui Lu, Bowen Wang, Xiao Liu, Yuxiao Dong, and Jie Tang. 2023. Agenttuning: Enabling generalized agent abilities for llms. *Preprint*, arXiv:2310.12823.

Ceyao Zhang, Kaijie Yang, Siyi Hu, Zihao Wang, Guanghe Li, Yihang Sun, Cheng Zhang, Zhaowei Zhang, Anji Liu, Song-Chun Zhu, Xiaojun Chang, Junge Zhang, Feng Yin, Yitao Liang, and Yaodong Yang. 2024a. Proagent: Building proactive cooperative agents with large language models. *Preprint*, arXiv:2308.11339.

Chi Zhang, Zhao Yang, Jiaxuan Liu, Yucheng Han, Xin Chen, Zebiao Huang, Bin Fu, and Gang Yu. 2023. Appagent: Multimodal agents as smartphone users. *Preprint*, arXiv:2312.13771.

Wenqi Zhang, Ke Tang, Hai Wu, Mengna Wang, Yongliang Shen, Guiyang Hou, Zeqi Tan, Peng Li, Yueting Zhuang, and Weiming Lu. 2024b. Agentpro: Learning to evolve via policy-level reflection and optimization. *Preprint*, arXiv:2402.17574.

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025. A Survey on the Memory Mechanism of Large Language Model-based Agents. *ACM Transactions on Information Systems*, 43(6):1–47.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024. Expel: Llm agents are experiential learners. *Preprint*, arXiv:2308.10144.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. 2024a. Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st International Conference on Machine Learning*, pages 62138–62160.

Shuyan Zhou, Frank F. Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, Uri Alon, and Graham Neubig. 2024b. Webarena: A realistic web environment for building autonomous agents. *Preprint*, arXiv:2307.13854.

Shuyan Zhou, Frank F Xu, Hao Zhu, Xuhui Zhou, Robert Lo, Abishek Sridhar, Xianyi Cheng, Tianyue Ou, Yonatan Bisk, Daniel Fried, and 1 others. 2024c. Webarena: A realistic web environment for building autonomous agents. In *International Conference on Learning Representations*, volume 2024, pages 15585–15606.

Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, and Maarten Sap. 2024d. Sotopia: Interactive evaluation for social intelligence in language agents. *Preprint*, arXiv:2310.11667.

Yifei Zhou, Andrea Zanette, Jiayi Pan, Sergey Levine, and Aviral Kumar. 2024e. Archer: Training language model agents via hierarchical multi-turn rl. *Preprint*, arXiv:2402.19446.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, Yu Qiao, Zhaoxiang Zhang, and Jifeng Dai. 2023. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *Preprint*, arXiv:2305.17144.

A Appendix

A.1 Extended Literature: Text-based and Symbolic Grounding

Expanding upon Section 2.1, early benchmarks like ALFWorld (Shridhar et al., 2020) instantiate text-based grounding by translating 3D embodied tasks into TextWorld environments, where agents receive textual descriptions of rooms and objects (e.g., "You are in a kitchen. You see a fridge.") and output discrete natural language commands. Similarly, in the domain of web navigation, WebArena (Zhou et al., 2024c) abstracts complex web pages into symbolic Document Object Model (DOM) trees or accessibility trees. The primary advantage of text-based grounding is its seamless alignment with the pre-training corpus of LLMs, enabling zero-shot planning capabilities. However, synthesizing a rich, partially observable environment into a linear sequence often limits the agent's ability to ground its semantics in physical reality.

A.2 Extended Literature: Foundational Alignment

Expanding upon Section 4.1, the foundational step in aligning an LLM to an environment is Behavioral Cloning (BC) via Supervised Fine-Tuning (SFT). Unlike standard instruction tuning, agentic SFT requires the model to internalize the full interactive loop: mapping observations to intermediate reasoning (e.g., Chain-of-Thought) and culminating in an executable action. Frameworks like FireAct (Chen et al., 2023) demonstrate that fine-tuning LLMs on successful tool-use trajectories significantly improves their zero-shot grounding in novel environments. To address the compounding errors often seen in BC, Agent-FLAN (Chen et al., 2024) carefully deconstructs agent training data, separating observation comprehension from action planning to alleviate state-tracking hallucinations. However, SFT fundamentally suffers from covariate shift: the agent only learns from expert demonstrations (positive trajectories) and remains unequipped to recover when its own actions drift out of the training distribution during inference.

To move beyond the limitations of offline static datasets, agents must learn from their mistakes. In the context of LLMs, standard scalar rewards

($R_t = -1$) are profoundly uninformative; they inform the agent that it failed, but not *why*. Language-guided alignment substitutes or augments scalar rewards with Textual Feedback (Verbal Reinforcement). In this paradigm, an environment simulator or a stronger Teacher Model (Critic) generates a natural language critique of the agent's trajectory. This feedback provides semantic gradients, explaining logical flaws or syntax errors in the generated action. By utilizing this critique to either iteratively refine the prompt or filter data for subsequent training rounds, the optimization process is grounded in the semantic space, making it vastly more sample-efficient than trial-and-error exploration.

A.3 Reproducibility and Closed-Source Volatility

Compounding the methodological flaws discussed in Section 6 is a severe reproducibility crisis driven by the reliance on closed-source, API-gated models. The underlying weights and safety alignments of proprietary models are updated continuously and silently by their host organizations.

Consequently, an agent's reported performance on a benchmark today may drastically degrade or artificially inflate tomorrow solely due to an unannounced server-side update. This temporal volatility makes longitudinal academic comparisons practically impossible. For the field of LLM-driven game agents to mature into a rigorous science, there is an urgent need to standardize evaluations around open-weight models (e.g., LLaMA, Qwen) where the exact parameters, inference configurations, and pre-training data distributions can be meticulously controlled and audited by the research community.

A.4 Extended Literature: The Tripartite Memory System

Expanding upon Section 3.3, modern game agents adopt a biologically inspired tripartite memory system to manage prolonged temporal context. **Working Memory** functions as the agent's scratchpad, dynamically tracking the current environmental state and immediate sub-goals. **Episodic Memory** archives the agent's historical interaction trajectories, storing past events in a vector database utilizing a retrieval mechanism weighted by recency, importance, and semantic relevance (Park et al., 2023). **Semantic Memory** serves as a static knowledge base containing immutable environment rules, lore, or API documentation. To orchestrate these streams, architectures like MemGPT (Packer et al., 2024) introduce an operating system-inspired paging mechanism, granting the LLM agency to actively page information between long-term storage and its working context.

A.5 Extended Literature: Value-driven Search and Preference Alignment

While online RL (Section 4) actively explores the environment, it remains susceptible to reward hacking and instability. An alternative paradigm leverages preference optimization (e.g., Direct Preference Optimization, DPO) (Rafailov et al., 2024) to align policies using offline, contrastive trajectory pairs. Frameworks such as LATS (Language Agent Tree Search) (Zhou et al., 2024a) and Eurus (Yuan et al., 2024) merge value-driven search with preference alignment. By utilizing Monte Carlo Tree Search (MCTS) as a data engine during training, the system explores multiple interaction branches. DPO is subsequently applied to increase the relative log-probability of successful textual trajectories over failed ones, effectively translating complex credit assignment into a straightforward contrastive learning objective.

A.6 Extended Literature: Social Simulation and Norm Emergence

Expanding upon Section 5, deploying multiple LLM-driven agents in unscripted, shared environments transitions the research focus from individual tactical deception to macro-level social dynamics. Frameworks like Sotopia (Zhou et al., 2024d) evaluate social intelligence by placing agents in varied scenarios requiring negotiation, persuasion, and emotional manipulation. Success is measured by the agent's ability to achieve private goals while maintaining social harmony. Extending this to broader populations, simulations such as ALYMPICS (Mao et al., 2024) and AgentSims (Lin et al., 2023b) observe the spontaneous formation of complex social phenomena, including economic exchange, division of labor, and the establishment of localized linguistic protocols.

A.7 Taxonomy

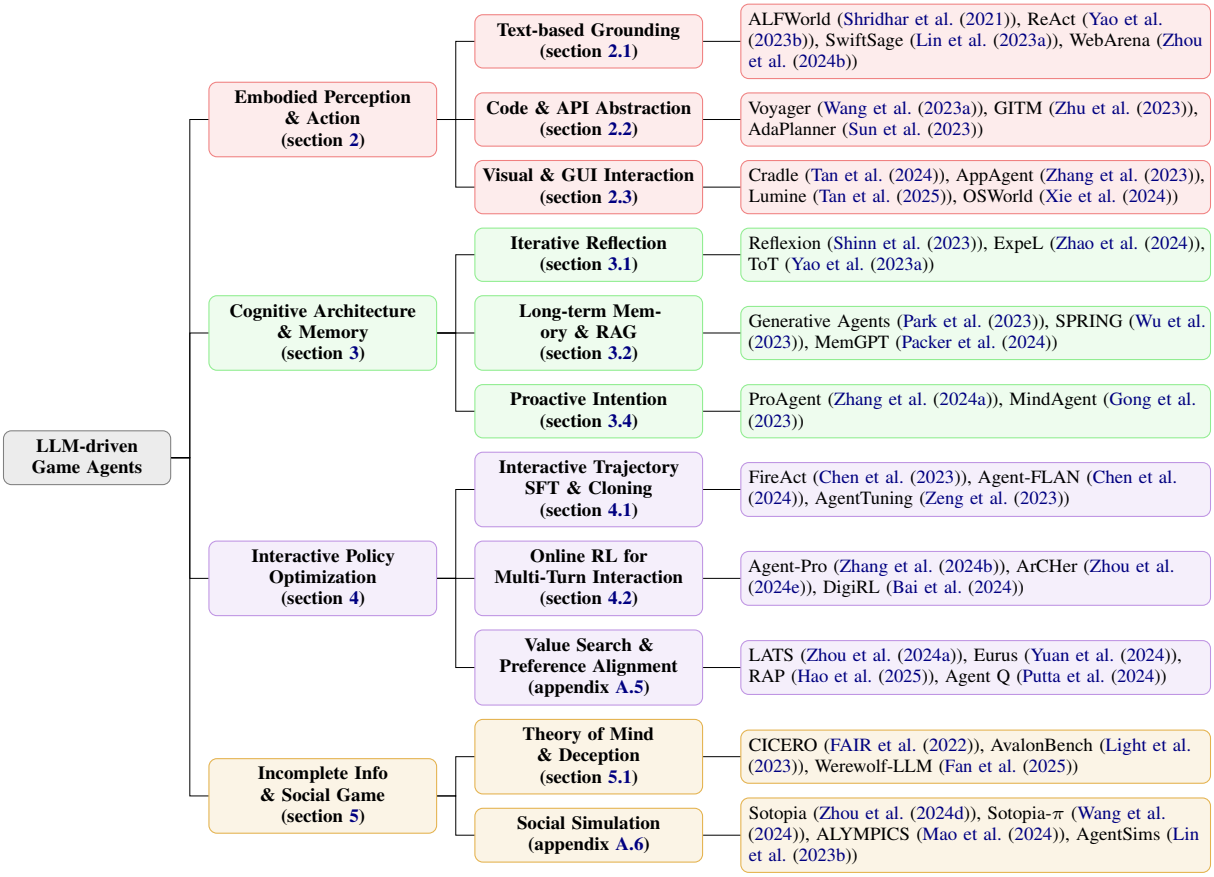

Figure 5: Taxonomy of LLM-driven Game Agents. We divide the evolution of game agents into four fundamental pillars: Embodied Perception & Action (Section 2), Cognitive Architecture & Memory (Section 3), Interactive Policy Optimization (Section 4), and Incomplete Information & Social Games (Section 5).