Context-Switching Costs in Hybrid LLM-Algorithm Systems

Kevin Scroggins, Kweku Yamoah, Kevin Kurian, and Emmanuel Dorley
University of Florida
Gainesville, FL, United States
kscroggins,kyamoah,kuriankevin,edorley@ufl.edu

ABSTRACT

This paper explores Hybrid systems that combine large language models (LLMs) with classical decision theoretic algorithms. Such systems are often assumed to offer complementary strengths in iterative reasoning tasks, yet little empirical work examines how such components should be composed. We used Wordle as a controlled testbed for iterative reasoning under deterministic feedback. Building on prior results showing that while standalone LLMs achieve high task success they also exhibit higher constraint violations and slower convergence than classical algorithms. Our evaluated hybrid configurations vary handoff direction, alternation frequency, and LLM involvement. Our findings show that hybridization does not improve performance over strong classical baselines. Instead, alternating control introduces systematic degradation, slower convergence rate, and higher constraint violations. This occurs despite full information sharing, as after each guess the LLM receives the full guess history and remaining candidates.The resulting context switching cost reflects difficulty maintaining coherent reasoning when continuing from externally chosen actions rather than an information deficit. These findings suggest that naive hybridization can undermine iterative reasoning, and that effective hybrid agent design depends critically on control flow, state ownership, and minimizing mid-episode handoffs.

KEYWORDS

Hybrid agent architectures, Context switching, Iterative reasoning

ACM Reference Format:
Kevin Scroggins, Kweku Yamoah, Kevin Kurian, and Emmanuel Dorley. 2026. Context-Switching Costs in Hybrid LLM-Algorithm Systems. In *Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), Paphos, Cyprus, May 25 – 29, 2026*, IFAAMAS, 7 pages.

1 INTRODUCTION

Hybrid agent architectures that combine large language models (LLMs) with classical decision theoretic methods are increasingly proposed for iterative AI systems, motivated by the belief that learned language priors and principled reasoning offer complementary strengths [2, 6, 12]. However, how such heterogeneous components should be composed remains poorly understood, and it is unclear whether hybridization improves iterative reasoning or introduces new failure modes [7, 15].

We study this question using *Wordle* [20] as a controlled testbed for iterative reasoning under deterministic constraints, where agents must maintain consistency with accumulated feedback and progressively reduce uncertainty, making failures in state tracking directly observable.

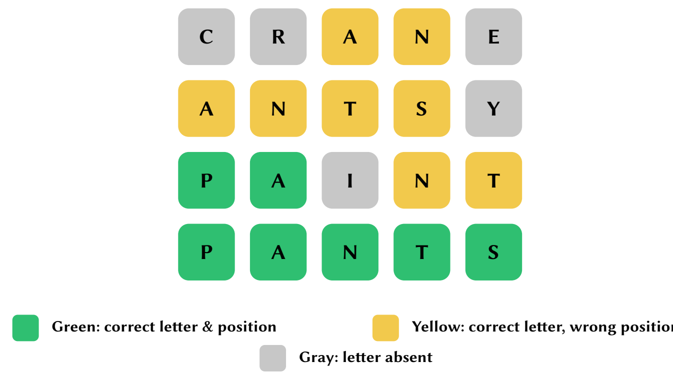

Figure 1: Wordle as a minimal diagnostic for iterative reasoning: each guess yields deterministic, position wise feedback (green/yellow/gray) that induces constraints and progressively reduces the candidate set until the target word (here, PANTS) is identified.

In prior work, we evaluated standalone LLM agents on this task and found that, while they achieve high win rates comparable to classical strategies, they converge more slowly and exhibit substantially higher constraint violation rates (see Appendix A). These standalone results are summarized in Appendix A for context; here, we focus on hybrid designs.

This workshop paper examines agents that combine LLM based and algorithmic reasoning within a single decision process. We compare pure classical strategies with hybrid configurations that vary handoff direction and alternation frequency. Our results show that hybrid performance depends critically on control structure: single handoff hybrids where an LLM initializes and a classical strategy completes the task achieve near optimal performance, while reverse and alternating handoffs perform substantially worse and incur significantly higher constraint violations, revealing a context switching cost when control shifts between reasoning modes.

Overall, we find no empirical evidence that hybridization improves performance over strong pure baselines. Instead, naive interleaving of learned and algorithmic reasoning degrades iterative behavior, even under full information sharing, highlighting the importance of control flow and state ownership in hybrid agent design.

Contributions. This paper (1) provides a focused empirical evaluation of hybrid LLM–algorithm agents in an iterative reasoning task, (2) identifies context switching costs as a primary source of hybrid degradation, and (3) demonstrates asymmetric handoff effects that inform practical hybrid design principles.

Proc. of the 25th International Conference on Autonomous Agents and Multiagent Systems (AAMAS 2026), C. Amato, L. Dennis, V. Mascardi, J. Thangarajah (eds.), May 25 – 29, 2026, Paphos, Cyprus.

2 BACKGROUND

2.1 Iterative Reasoning

Iterative reasoning refers to a sequential decision making process in which an agent maintains and updates an internal representation of the environment as new information becomes available over time. Rather than solving a problem in a single step, the agent repeatedly selects actions, observes feedback, updates its state or beliefs, and uses this updated information to guide future decisions [11]. This paradigm is central to many areas of artificial intelligence involving uncertainty and incomplete information, including decision analysis, Bayesian inference, planning, negotiation, and preference elicitation. Classical approaches formalize this process using explicit representations of belief, utility, and information value, enabling agents to systematically reduce uncertainty and improve decision quality across interaction rounds [16, 17]. In domains such as automated negotiation and preference elicitation, agents iteratively refine models of hidden preferences using Bayesian updates, value-of-information criteria, or regret based decision rules [3, 4]. These approaches are explicitly designed to maintain coherent state representations and to select actions that improve future decisions.

Large language models are increasingly deployed in similar iterative roles, including negotiation support, recommendation, and advisory systems. Unlike classical methods, however, LLMs typically encode state implicitly within textual context rather than maintaining explicit belief representations. Prior work has shown that this implicit state tracking can be brittle in multi-turn settings, with evidence of state drift and inconsistency across interaction rounds [1, 21]. This raises a fundamental question for iterative AI design: how should learned and formal reasoning components be combined when reliable state tracking is required?

2.2 Games as Diagnostic Benchmarks

Games provide controlled environments with explicit rules, structured feedback, and clear success criteria, making them useful for evaluating sequential decision making behavior [8, 19]. However, outcome based metrics alone can obscure important differences in reasoning processes. Agents may achieve similar win rates while employing qualitatively different strategies or levels of internal consistency [14]. As a result, recent work emphasizes the use of process level metrics such as convergence behavior, constraint adherence, and belief refinement—to characterize how agents reason over time rather than only whether they succeed.

2.3 Wordle as a Testbed for Hybrid Reasoning

We use *Wordle* as a minimal diagnostic testbed for iterative reasoning and hybrid agent design. An agent must infer a hidden five letter word through successive guesses, receiving deterministic, position wise feedback that induces logical constraints and progressively reduces a finite hypothesis space of 5,629 valid words.

Wordle's partial observability, enumerable state space, and unambiguous feedback make failures in state tracking directly observable, enabling fine grained analysis of convergence, search space pruning, and constraint violations, especially when control alternates between learned and algorithmic components.

This setting allows us to ask a precise question: *under what conditions does combining LLM-based and classical decision theoretic reasoning help or hinder iterative performance?*

3 EXPERIMENTAL DESIGN

3.1 Baselines

We first established baseline performance using classical decision theoretic approaches on a shared set of 100 target words. Classical strategies achieved win rates of 96% (CSS) and 99% (VOI). LLM agents across three size tiers achieved similar win rates (93–97%), with no significant differences between tiers.

Given this comparable baseline performance, we ask whether *hybrid agent designs*—combining LLM and algorithmic reasoning within a single decision process—can improve performance or instead introduce new failure modes. To preserve behavioral fidelity, we disabled fallback mechanisms that would replace invalid LLM outputs, ensuring that observed violations reflect internal state tracking rather than external correction.

3.2 Hybrid Design Space

We explore a structured hybrid design space spanning 38 configurations and $\approx 29{,}000$ games. Each configuration combines an LLM with a classical solver (CSS or VOI), varying how and when control transfers between components. Configurations differ across factors including handoff direction, alternation frequency, round allocation, repair mechanisms, and model selection.

Handoff Patterns (Group A). We test single handoff designs (LLM first, then classical), reverse handoffs (classical first, then LLM), and multi-stage handoffs, isolating the effect of handoff direction.

Round Allocation (Group B). We vary the number of rounds ($k \in \{1, 2, 3\}$) each component controls before handoff, testing whether increased LLM involvement improves or degrades outcomes.

Alternation Frequency (Group C). We evaluate designs that alternate control every round between LLM and classical, probing the impact of repeated context switching.

Repair Architectures (Group D). We test hybrids where algorithms constrain rather than replace LLM outputs, including constraint filtering and reranking, to assess whether lightweight safeguards improve reliability.

3.3 Models and Algorithms

Experiments use 8 LLMs (see Table 1) spanning small (7–8B), mid-sized (22–27B), and frontier (70B) scales. Algorithmic components include CSS, which minimizes worst-case remaining candidates via minimax regret [13], and VOI, which balances information gain and solution probability [5, 10]. Zero-shot prompting is used throughout unless otherwise noted; chain-of-thought (COT) variants are evaluated in Groups C and D. However as CoT did not alter any qualitative trends, it is omitted from this short paper for brevity.

3.4 Metrics

Performance was evaluated using six metrics: (1) **win rate**, which measures how often an agent successfully identifies the target word

Model	Parameters
mistral-7b-instruct	7B (small)
llama-3.1-8b-instruct	8B (small)
granite-3.3-8b-instruct	8B (small)
llama-3.1-nemotron-nano-8B-v1	8B (small)
codestral-22b	22B (mid)
gemma-3-27b-it	27B (mid)
llama-3.1-70b-instruct	70B (frontier)
llama-3.3-70b-instruct	70B (frontier)

Table 1: Models evaluated and parameter scales.

within the allowed guesses; (2) **attempts to win**, capturing efficiency via the average number of guesses required; (3) **constraint violations**, which count guesses that contradict earlier feedback (e.g., reusing gray letters, misplacing yellow letters, or violating confirmed green positions), indicating breakdowns in iterative reasoning; (4) **search space reduction**, measuring how effectively guesses eliminate impossible words and reflect information seeking behavior; (5) **Hamming distance** [9, 18], which tracks how many letter positions differ from the target word and how quickly guesses improve; and (6) **convergence rate**, describing how rapidly an agent's guesses move toward the correct solution over successive rounds.

4 RESULTS

4.0.1 Win Rate Analysis. Win rates were high across all approaches (93.7–99%), and although the effect of approach was statistically significant ($F(4, 21395) = 20.74$, $p < .001$), the effect size was small ($\eta_p^2 = .004$). Significant differences arose only within hybrid variants. *HybridLLMFirst* (96.7%) outperformed both *HybridAlternating* (95.6%, $p = .017$) and *HybridAlgoFirst* (93.7%, $p < .001$), while *HybridAlternating* also outperformed *HybridAlgoFirst*. This ordering indicates a strong handoff asymmetry: algorithms can recover from noisy LLM initial guesses, whereas LLMs struggle to continue coherent reasoning from algorithm-selected moves.

4.0.2 Attempts to Solve. The effect of approach on attempts was statistically significant ($F(4, 20489) = 14.22$, $p < .001$), though the effect size was small ($\eta_p^2 = .003$). Pure algorithms required the fewest guesses, with *CSS* (3.85) and *VOI* (3.93) performing equivalently and outperforming algorithm-first hybrids. Among hybrid designs, a clear ordering emerged: *HybridLLMFirst* (4.10) required fewer attempts than *HybridAlternating* (4.15), which outperformed *HybridAlgoFirst* (4.21). Neither *HybridLLMFirst* nor *HybridAlternating* differed significantly from pure algorithms. Overall, these results indicate that minimizing LLM involvement, especially avoiding algorithm-to-LLM handoffs, yields more efficient solving.

4.0.3 Constraint Violation Analysis. Constraint violations varied sharply by approach ($F(4, 21395) = 295.49$, $p < .001$, $\eta_p^2 = .052$), the largest effect across all metrics. Pure algorithms (*CSS*, *VOI*) incurred zero violations by design. Among hybrids, violations increased with context switching: *HybridLLMFirst* showed the fewest (0.19 per game), followed by *HybridAlgoFirst* (0.59) and *HybridAlternating* (0.91). *HybridAlternating* violated constraints significantly more than all others (all $p < .001$), providing direct evidence that repeated control switching disrupts state tracking in hybrid agents.

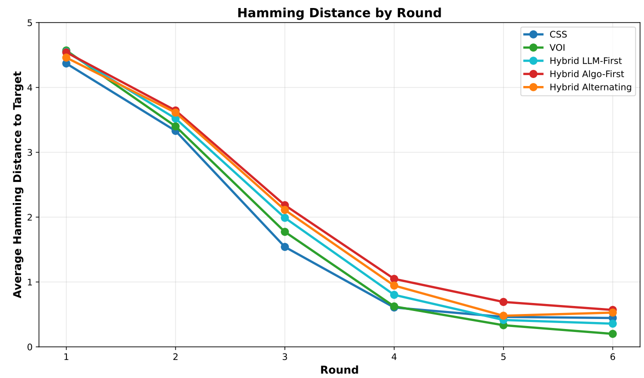

Figure 2: Average Hamming distance to the target word across rounds (lower is better).

4.0.4 Search Space Reduction. The effect of approach on first round search space reduction was statistically significant ($F(4, 18195) = 63.87$, $p < .001$, $\eta_p^2 = .014$). *HybridAlternating* (96.7%) and *CSS* (96.6%) achieved the strongest initial pruning and did not differ significantly from each other, while both significantly outperformed all other approaches. However, this early advantage did not translate into higher win rates or greater efficiency, indicating that aggressive first round pruning does not compensate for the downstream reasoning disruptions introduced by subsequent context switches.

4.0.5 Hamming Distance. Average Hamming distance differed significantly across approaches ($F(4, 21395) = 11.56$, $p < .001$, $\eta_p^2 = .002$), indicating differences in how quickly agents' guesses converged toward the target word over successive rounds. Pure algorithmic approaches remained consistently closer to the target throughout gameplay, with *CSS* achieving the lowest average Hamming distance overall. In contrast, *HybridAlgoFirst* maintained the largest distance from the target across rounds, suggesting weaker refinement after control transferred from the algorithm to the LLM. Alternating hybrids also converged more slowly than pure algorithms, though the overall effect size was small. Together, these results suggest that repeated or reverse-direction handoffs disrupt steady iterative refinement even when overall win rates remain high.

4.0.6 Convergence Rate. Convergence rate differed significantly by approach ($F(4, 21394) = 24.64$, $p < .001$, $\eta_p^2 = .005$). Pure algorithms (*VOI*, *CSS*) and *HybridLLMFirst* converged fastest, while *HybridAlternating* and *HybridAlgoFirst* converged more slowly. This pattern again indicates that context-switching, particularly algorithm-to-LLM or alternating control—impedes steady progress toward the target.

5 DISCUSSION

5.1 Context-Switching as the Primary Degradation Mechanism

Hybrid approaches incur substantially more constraint violations (0.28 per game) than pure LLM agents (0.04–0.16), despite pure algorithms producing zero violations, since candidate filtering explicitly prevents guesses that contradict prior feedback. This provides direct evidence of state inconsistency under alternating control: LLMs

Approach	Metrics				
	Win Rate	Avg Attempts to Win	Constraints Violations	Search Space Reduction	Convergence Rate
VOI	96.0%	3.85	0.00	96.6%	1.13
CSS	99.0%	3.93	0.00	92.7%	1.16
Hybrid LLM-First	96.7%	4.10	0.19	93.8%	1.11
Hybrid Algo-First	93.7%	4.21	0.59	94.2%	1.08
Hybrid Alternating	95.5%	4.15	0.91	96.7%	1.07

Table 2: Performance of five Wordle solving strategies under deterministic position wise feedback. We report win rate (%), average number of attempts on successful games, mean constraint violations (invalid guesses or inconsistencies with revealed feedback), the average fraction of the candidate lexicon eliminated per episode (search space reduction), and a convergence rate summarizing how quickly the hypothesis set collapses across guesses. Higher is better for win rate, search space reduction, and convergence; lower is better for attempts and violations.

fail to fully incorporate constraints implied by algorithmically chosen guesses. Critically, this degradation occurs despite full state sharing between components. After each guess, the candidate set is filtered using identical constraint logic, and the LLM receives both the updated candidates and complete guess history in its prompt. The resulting context-switching cost therefore reflects not an information deficit, but difficulty maintaining coherent reasoning when continuing from an externally chosen action the LLM did not generate.

This interpretation is reinforced by a strong handoff asymmetry. LLM-first hybrids achieve approximately 97% win rates, whereas algorithm-first hybrids drop to 91%. Algorithms readily recover from noisy LLM openings because feedback fully specifies constraints, but LLMs struggle to continue coherent reasoning from algorithm generated moves. This suggests that LLM state tracking depends on having generated prior actions itself.

5.2 Dose Response Relationship

In the hybrid design space, across Groups A and B, increased LLM involvement consistently correlates with worse performance, indicating that each context switch carries an accumulating cost. Single-handoff designs ($L_1 \rightarrow$ CSS, 97.1%) outperform multi-round LLM involvement ($L_3 \rightarrow$ CSS, 95.6%) and fully alternating designs (~95.5%). Some algorithm→LLM configurations with smaller models failed to complete due to instability when assuming control mid-game; these were excluded from aggregate metrics. These failures align with the broader handoff asymmetry and further support the claim that LLMs struggle to recover coherent state after mid-game control transfers.

5.3 Repair Architectures: Necessary but Insufficient

Repair hybrids perform better than alternating designs, with the best configuration (rerank + CSS, CoT) reaching 96.9% win rate. However, this still falls short of pure VOI (99%). Moreover, 45–100% of guesses in these hybrids required algorithmic intervention, indicating that when repair is frequent, hybrids offer little benefit over pure algorithmic play while adding complexity and latency.

5.4 Implications for Hybrid System Design

Overall, hybrid effectiveness depends on compatible state representations across components. Algorithms robustly enforce constraints regardless of prior guesses, while LLMs struggle to reconstruct state they did not generate. This asymmetry favors designs that either (1) use LLMs only for initialization followed by algorithmic control, or (2) avoid hybridization entirely when strong pure methods suffice. More broadly, these results caution against assuming complementary gains from hybridization: in domains with explicit constraints, context-switching costs can outweigh any benefits of combining learned and formal reasoning.

6 CONCLUSION

We evaluated hybrid LLM–algorithm systems on an iterative reasoning task across 38 configurations and approximately 29,000 games. No hybrid design outperformed the best pure algorithmic approach (VOI, 99% win rate), and most hybrids underperformed pure LLM agents. Performance degradation is driven by a context switching cost: alternating control disrupts LLM state tracking and substantially increases constraint violations, with more frequent handoffs leading to worse outcomes. Single handoff designs, where an LLM provides initialization and an algorithm completes the task—consistently outperform alternating hybrids. These findings suggest a clear design principle: **limiting control handoffs matters more than balancing reasoning components**. More broadly, comparable win rates can mask meaningful differences in reasoning quality that only emerge through process-level metrics such as constraint adherence and convergence behavior.

7 FUTURE WORK

Future work should investigate the mechanisms underlying context switching costs in hybrid LLM–algorithm systems. Although hybrids share identical candidate sets and histories, LLMs seem to struggle to maintain coherent constraint reasoning after mid episode handoffs, suggesting failures in implicit state reconstruction rather than information loss. Techniques such as representation similarity analysis, attention inspection, and activation patching may help identify how externally generated state is processed across turns. Future work should also explore structured state sharing mechanisms, such as explicit constraint sets or symbolic memory buffers, instead of relying solely on text based prompts. Finally, it remains important to test whether the observed handoff asymmetry generalizes beyond Wordle to iterative domains such as negotiation, planning, and scheduling.

A STANDALONE LLM PERFORMANCE ON WORDLE

Approach	Metrics				
	Win Rate	Avg Attempts to Win	Constraints Violations	Search Space Reduction	Convergence Rate
Small (< 10B)	94.3%	4.35	0.12	91.5%	1.066
Mid (10–30B)	96.5%	4.32	0.16	92.0%	1.039
Frontier (> 30B)	93.4%	4.31	0.04	92.8%	1.048

Table 3: Standalone LLM performance on Wordle (companion study). We report win rate (%), average attempts on successful games, mean constraint violations, search space reduction, and convergence rate. Higher is better for win rate, search space reduction, and convergence; lower is better for attempts and violations.

This appendix summarizes results from a companion study evaluating *standalone* large language model (LLM) agents on the Wordle task. These results are included for context and are not the primary focus of the present workshop paper, which examines hybrid LLM–algorithm systems.

A.1 Experimental Setup

Standalone LLM agents were evaluated on the same Wordle environment, target-word set, and deterministic feedback rules used throughout this paper. Agents generated guesses directly without algorithmic filtering, repair, or fallback mechanisms. All constraints were communicated solely through the standard Wordle feedback channel, requiring the model to maintain consistency implicitly via textual context.

A.2 Performance Summary

Across model sizes, standalone LLM agents achieved high win rates comparable to classical decision-theoretic strategies. Specifically, LLMs achieved win rates of 94.3% (small, ≤10B), 96.5% (mid-sized, 10–30B), and 93.4% (frontier, >30B), with no statistically significant differences across tiers ($p > .5$). Thus, when evaluated independently, both algorithmic (96–99%) and LLM-based approaches (93–97%) solved Wordle at high rates.

However, despite high task success, standalone LLMs required more attempts to solve the task, averaging 4.03–4.47 guesses compared to 3.85–3.96 for classical algorithms, and produced higher rates of constraint violations (0.04–0.16 per game versus 0.00 for algorithms). These results indicate difficulty maintaining a consistent internal state across rounds rather than a failure of local reasoning.

A.3 Implications for Hybrid Design

Taken together, these findings indicate that standalone LLMs are capable of strong local inference and information-seeking behavior but exhibit weaknesses in maintaining and applying accumulated constraints over multiple rounds. The hybrid experiments in the main paper build directly on this observation, investigating whether algorithmic components can compensate for these limitations when control is shared or transferred between reasoning systems.

REFERENCES

[1] Sanjeev Arora, Yuanzhi Li, Yicheng Pan, Yuhuai Wu, and Yi Zhang. 2023. Faithfulness of Reasoning in Large Language Models. *Transactions of the Association for Computational Linguistics* (2023).

[2] Tim Baarslag, Mark J. C. Hendrikx, Koen V. Hindriks, and Catholijn M. Jonker. 2016. Learning about the opponent in automated bilateral negotiation: a comprehensive survey of opponent modeling techniques. *Autonomous Agents and Multi-Agent Systems* 30, 5 (2016), 849–898. https://doi.org/10.1007/s10458-015-9309-1

[3] Tim Baarslag, Koen Hindriks, and Catholijn Jonker. 2016. Negotiation Decision Functions for Autonomous Agents. In *Proceedings of the International Conference on Autonomous Agents and Multiagent Systems (AAMAS)*. 1103–1111.

[4] Craig Boutilier. 2002. A POMDP Formulation of Preference Elicitation Problems. In *Proceedings of the National Conference on Artificial Intelligence (AAAI)*. 239–246.

[5] Craig Boutilier, Richard Dearden, and Moisés Goldszmidt. 2002. A Principled Analysis of Utility Elicitation. In *Proceedings of the Eighteenth Conference on Uncertainty in Artificial Intelligence (UAI)*. Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 102–109.

[6] Robin Burke. 2002. Hybrid Recommender Systems: Survey and Experiments. *User Modeling and User-Adapted Interaction* 12, 4 (2002), 331–370. https://doi.org/10.1023/A:1021240730564

[7] Michael K. Chen. 2025. A Comparative Study of Neurosymbolic AI Approaches to Interpretable Logical Reasoning. In *Proceedings of The 19th International Conference on Neurosymbolic Learning and Reasoning*, Leilani Gilpin, Eleonora Giunchiglia, Pascal Hitzler, and Emile van Krieken (Eds.), Vol. 284. Proceedings of Machine Learning Research, 447–462. https://proceedings.mlr.press/v284/chen25b.html

[8] Michael Genesereth, Nathaniel Love, and Barney Pell. 2005. General Game Playing: Overview of the AAAI Competition. In *AI Magazine*, Vol. 26. 62–72.

[9] Richard W. Hamming. 1950. Error Detecting and Error Correcting Codes. *Bell System Technical Journal* 29, 2 (1950), 147–160.

[10] Ronald A. Howard. 1966. Information Value Theory. *IEEE Transactions on Systems Science and Cybernetics* 2, 1 (1966), 22–26.

[11] Mykel J. Kochenderfer. 2015. *Decision Making Under Uncertainty: Theory and Application.* MIT Press, Cambridge, MA.

[12] Xiang Li. 2026. Task Planning and Decision-Making Methods for Intelligent Agents Based on Large Language Models. In *Proceedings of the 4th International Conference on Artificial Intelligence and Intelligent Information Processing (AIIIP 2025)*. Association for Computing Machinery, New York, NY, USA, 817–822. https://doi.org/10.1145/3778534.3778661

[13] Tyler Lu and Craig Boutilier. 2011. Robust Approximation and Incremental Elicitation in Voting Protocols. In *Proceedings of the 22nd International Joint Conference on Artificial Intelligence (IJCAI)*. IJCAI/AAAI Press, Barcelona, Catalonia, Spain, 287–293. https://www.ijcai.org/Proceedings/11/Papers/058.pdf

[14] Marlos C. Machado, Marc G. Bellemare, Erik Talvitie, Joel Veness, Matthew Hausknecht, and Michael Bowling. 2018. Revisiting the arcade learning environment: evaluation protocols and open problems for general agents. *J. Artif. Int. Res.* 61, 1 (Jan. 2018), 523–562.

[15] Uzma Nawaz, Mufti Anees-ur Rahaman, and Zubair Saeed. 2025. A Review of Neuro-Symbolic AI Integrating Reasoning and Learning for Advanced Cognitive Systems. *Intelligent Systems with Applications* 26 (2025), 200541. https://doi.org/10.1016/j.iswa.2025.200541

[16] Judea Pearl. 1988. *Probabilistic Reasoning in Intelligent Systems: Networks of Plausible Inference.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA.

[17] H. Raiffa. 1970. *Decision Analysis Introductory Lectures on Choices Under Uncertainty.* https://books.google.com/books?id=FKst-EYiDvEC

[18] Claude E. Shannon. 1948. A Mathematical Theory of Communication. *Bell System Technical Journal* 27, 3 (1948), 379–423.

[19] Claude E. Shannon. 1950. Programming a Computer for Playing Chess. *Philos. Mag.* 41, 314 (1950), 256–275.

[20] Josh Wardle. 2021. Wordle. https://www.powerlanguage.co.uk/wordle/. Online word game.

[21] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2022. Chain-of-Thought Prompting Elicits Reasoning in Large Language Models. In *Advances in Neural Information Processing Systems (NeurIPS)*.