Experience-Sensitive Game Learning: A Behavioral Study of Humans and Language Agents

Anonymous ACL submission

Abstract

Large language model agents are increasingly evaluated through games, but most benchmarks emphasize final outcomes rather than how players learn from repeated interaction. We study *experience-sensitive game learning*: how gameplay experience changes the decision-making behavior of humans and language agents. We formulate experience-sensitive game learning as a framework for analyzing behavioral change across repeated gameplay, rather than only final score or win rate. We introduce a suite of interactive games with reusable strategic structure, together with cross-game greedy-to-global metrics and game-specific behavioral diagnostics that make experience-driven change observable from action traces. We also collect repeated-game trajectories from human players and evaluate recent self-evolving language agents in the same behavioral metric space. Our results show that human players exhibit interpretable and relatively stable shifts from locally greedy heuristics toward more global strategic decisions. In contrast, current self-evolving agents often show noisy and transient gains, suggesting that existing self-evolution methods remain limited in converting gameplay experience into durable changes in decision-making behavior.

1 Introduction

Games have long served as testbeds for intelligence. For large language model (LLM) agents, games are especially appealing because they combine explicit rules, sequential decision-making, repeatable episodes, and rich interaction traces. Success in such games can arise from different sources of competence. One source is strong within-episode computation: the ability to reason from the rules, evaluate states, and search over possible actions in a particular game trajectory. Another source is accumulated gameplay experience: repeated interaction may help a player discover reusable strategies, abandon misleading heuristics, and transfer lessons from earlier episodes to later decisions. Although these two sources of competence are closely related in practice, they are conceptually different. Existing studies on game-based LLM agent evaluation have largely emphasized within-episode competence, treating games as tests of reasoning, planning, or task completion within individual episodes. Much less attention has been paid to the second form: whether repeated interaction changes the way an agent evaluates states, selects actions, and reuses experience across episodes.

In this work, we study *experience-sensitive game learning*. We use this term to describe settings in which repeated gameplay can change a player's future decisions by inducing reusable strategic knowledge. In an experience-sensitive game, effective play is not determined only by reasoning from the rules. It also depends on whether the player can discover patterns, revise misleading heuristics, learn and reuse lessons from prior episodes. This formulation applies to both language agents and human players. Our goal is to analyze how gameplay experience changes observable decision-making behavior.

Consider Othello as a motivating example. A novice player may look at a board state and prefer the move that flips the largest number of discs in the current turn, since flipping more discs appears to make immediate progress under a simple reading of the rules. However, with experience, players often learn that such locally greedy moves can be strategically weak, as they may reduce future mobility, expose unstable frontier discs, or give the opponent access to valuable positions.

The same board state is, therefore, interpreted differently before and after gaining experience. What changes is not merely the player's final outcome, but the heuristic used to evaluate actions. This makes Othello an example of experience-sensitive learning: repeated gameplay can reshape

which features of a state are considered important.

To study this form of learning, we instantiate our framework with a suite of experience-sensitive games selected for their reusable strategic structure and rich behavioral traces. For each game, we define diagnostic behavioral metrics that capture interpretable aspects of play. These metrics are not the goal of the study but instruments for making experience-driven change observable. They allow us to ask whether a player becomes less greedy, more exploratory, more consistent, or more capable of long-horizon planning as gameplay experience accumulates. By turning raw gameplay trajectories into behavioral profiles, we can analyze learning dynamics within a player and compare learning patterns across different types of players.

Another component of our study is human gameplay data. We do not use humans merely as a performance ceiling or a source of final scores. Rather, we treat human players as learning subjects whose behavioral changes can be analyzed using the same metrics. Human trajectories provide a reference for what experience-sensitive learning looks like in practice: which heuristics disappear, which strategies emerge, and which behavioral metrics change as players become more familiar with a game. This allows us to compare not only whether modern agents perform as well as humans, but also whether they improve in similar ways.

Our contributions are threefold: (1) We formulate experience-sensitive game learning as a framework for studying how humans and language agents change their behavior through repeated gameplay; (2) We introduce a suite of experience-sensitive games and game-specific behavioral metrics that make experience-driven behavioral change observable beyond final performance; and (3) we collect human gameplay trajectories and evaluate recent self-evolving language agents in the same metric space, enabling us to analyze human learning, agent self-improvement, and human-agent behavioral differences. Together, our work turns games from static tests of task success into interactive environments for studying how experience reshapes decision-making behavior.

2 Related Work

Game Benchmarks. Games have long been used as controlled environments for evaluating language-based decision making. Early text-game platforms such as TextWorld (Côté et al., 2019) and Jericho (Hausknecht et al., 2020) provide standardized environments for training and evaluating agents in text-based interactive worlds. These environments have enabled a line of work on action generation, exploration, and policy learning in text games, including CALM (Yao et al., 2020), which trains action language models from human gameplay trajectories. Recent work has expanded game-based evaluation beyond simple rule-following tasks toward broader interactive, strategic, and multimodal settings. Several benchmarks evaluate LLM agents in out-of-distribution or competitive games, emphasizing strategic reasoning, long-horizon decision making, and comparisons with human or model opponents (Costarelli et al., 2024; Guertler et al., 2025; Tang et al., 2026). Other efforts further broaden the scope to large collections of text-based, visual, and video-game environments, enabling evaluation of interactive gameplay, visual grounding, and multi-turn reasoning across LLMs and VLMs (Shi et al., 2025; Hu et al., 2025; Paglieri et al., 2025; Hu et al., 2026; Park et al., 2026).

Self-Evolving Language Agents. Prior work has shown that language models can act as agents in complex sequential environments through reasoning, acting, and search, but these methods mainly improve within-episode deliberation rather than persistent learning from accumulated experience (Wei et al., 2022; Yao et al., 2023b,a; Zhou et al., 2024). Another line of work equips agents with memory or reusable experience representations, allowing them to store and reuse reflections, skills, causal abstractions, rules, or procedural knowledge across trials (Shinn et al., 2023; Wang et al., 2024; Majumder et al., 2024). Most related to our setting are recent studies of test-time learning and self-evolving agents. EvoTest formalizes repeated gameplay as J-TTL, where agents are evaluated by improvement across episodes in Jericho games; JitRL and Meta-TTL further study training-free policy adaptation and learnable adaptation policies for agents interacting repeatedly with an environment (He et al., 2026; Li et al., 2026; Lou et al., 2026). ReasoningBank distills successful and failed experiences into reusable reasoning strategies for test-time self-evolution, while CEL uses repeated gameplay to induce explicit environment rules and a strategic playbook (Ouyang et al., 2026; Wang et al., 2025). These works mainly propose mechanisms for improving agent performance from experience.

Game	Interface	Greedy Baseline	Global Evaluator	Learning Signal
Four in a Row *Adversarial search*		*Local line completion*: extend or block the most visible line. This action can look tactically useful but may miss forced threats, multi-step threats, or center control.	**Solver value** over legal columns. The evaluator rewards moves that are strong under global adversarial search rather than only locally visible patterns.	**Pattern completion → solver-aligned search.** Improvement means moving from obvious local connections toward moves that match globally strong play.
Othello6 *Greedy-trap strategy*		*Maximum-flip move*: choose the legal action that flips the most discs immediately. This reward proxy is visually salient but can reduce future mobility or expose unstable positions.	**Search value** from depth-limited minimax or related evaluation. The evaluator favors mobility-aware and positionally robust moves over immediate disc gain.	**Flip maximization → mobility control.** Improvement means avoiding short-sighted flips when they harm long-term positional value.
CircleCat *Spatial containment*		*Nearest-cell block*: place a wall on the legal cell closest to the cat. This can feel urgent but may leave multiple escape corridors or reachable boundary exits open.	**Containment score** based on shortest escape paths, reachable exits, and reachable area. The evaluator rewards walls that reshape the global escape structure.	**Local blocking → global containment.** Improvement means placing walls to close bottlenecks, reduce exits, and compress the reachable region.

Table 1: Summary of the three game environments. Each row shows the representative interface, the natural greedy baseline, the corresponding global evaluator, and the behavioral shift used to diagnose experience-sensitive learning.

Process-Level Evaluation and Human Gameplay. Many works evaluate agents using final outcomes such as win rate, task success, or accumulated score. Recent work has argued that such metrics can hide important differences in how agents reason, communicate, and act, and has therefore introduced more fine-grained evaluation protocols based on decision tracking, behavioral trajectories, reasoning traces, planning, revision, and resource-aware decision making (Tang et al., 2026; Hu et al., 2025; Xie et al., 2026; Ma et al., 2024; Yuan et al., 2025). Our work follows this process-level perspective but focuses specifically on whether repeated gameplay produces persistent and interpretable changes.

Human data have also been used in prior game-based NLP work, either as demonstrations for training agents or as reference points for contextualizing model performance (Yao et al., 2020; Costarelli et al., 2024; Guertler et al., 2025; Hu et al., 2025). However, these works typically treat humans as a source of supervision or as a performance ceiling, rather than as learning subjects whose behavior changes through repeated gameplay. Our perspective is motivated by classic studies of expertise and cognitive learning, which show that human decision making is shaped by reusable perceptual chunks, templates, heuristics, and abstractions acquired from experience (Chase and Simon, 1973; Gobet and Simon, 1996, 1998; Gershman et al., 2015; Lieder and Griffiths, 2020; Gershman and Goodman, 2014; Dasgupta et al., 2018, 2020). We operationalize this view in interactive games: by collecting human trajectories across repeated play, we study not only whether players improve, but also which behavioral patterns change, which misleading heuristics disappear, and which strategies emerge over time.

3 Environments and Behavioral Metrics

We use games because repeated experience can produce observable changes in action choices. We select environments with a common diagnostic structure: players may initially favor locally appealing actions, whereas stronger play requires evaluating longer-term strategic consequences. This allows us to test whether repeated play induces a greedy-to-global shift in decision-making, rather than merely improving final outcomes such as win rate or score.

3.1 Environment Suite

We study three games: Four in a Row, Othello6, and CircleCat. They were selected according to four criteria:

1. success should require reusable strategic principles rather than only local pattern matching;
2. each game should contain a natural but potentially misleading greedy heuristic;
3. each game should admit interpretable process-level metrics computable from action traces;
4. each game should be suitable for both human data collection and agent evaluation under a shared interface.

Under these criteria, the three environments instantiate complementary forms of the same greedy-to-global structure: Four in a Row tests movement from local line completion toward solver-aligned adversarial search; Othello6 tests movement from immediate disc maximization toward mobility-aware positional evaluation; and CircleCat tests movement from proximity-based blocking toward global containment planning. We summarize these design choices in Table 1.

Four in a Row is an adversarial game played on a vertical grid with seven columns. On each turn, the player chooses a column, and the piece falls to the lowest available cell in that column. The first player to form four consecutive pieces horizontally, vertically, or diagonally wins; if the board is filled without a winner, the game ends in a draw.

Four in a Row serves as our solver-aligned adversarial search environment. Its rules are simple enough for repeated human and agent play, but strong play requires more than extending visible lines or blocking immediate threats. Players must learn to recognize immediate wins, forced threats, multi-step threats, and the strategic value of central columns. Because legal actions can be evaluated against a strong solver, this environment allows us to measure whether repeated gameplay shifts players from locally salient pattern-based moves toward solver-aligned strategic choices, rather than only improving the final win rate.

Othello6 is a 6x6 version of Othello. Players take turns placing discs on the board. A legal move must bracket at least one line of opponent discs horizontally, vertically, or diagonally, after which the bracketed discs are flipped to the player's color. If a player has no legal move, the turn is passed. The game ends when no further moves are available, and the player with more discs wins.

Othello6 serves as our environment for studying greedy-trap avoidance. It exposes a common form of myopic play: maximizing the number of discs flipped in the current move. This heuristic is visually salient and often appears reasonable to novice players, but it can reduce future mobility, expose unstable frontier discs, open dangerous positions, or give the opponent access to stronger regions of the board. The strategic learning target is therefore a shift from immediate flip maximization toward mobility-aware and positionally robust evaluation. By comparing flip count, mobility, and positional metrics over repeated play, Othello6 allows us to test whether experience changes the heuristic used to evaluate actions.

CircleCat is a spatial containment game on a hex-neighbor board. On each turn, the player places one wall on an empty cell, after which the cat moves one step. The player wins if the cat becomes trapped and can no longer reach the boundary, and loses if the cat reaches the boundary.

CircleCat complements the adversarial board games by testing spatial containment rather than opponent modeling. Its natural novice heuristic is to block cells close to the cat, but effective play requires reasoning about the global escape structure of the board. A locally urgent wall may leave multiple escape corridors open, whereas a farther wall may close a bottleneck, remove several reachable exits, or shrink the cat's reachable region.

By tracking wall placement relative to the shortest escape paths, reachable boundary cells, and containment regions, CircleCat allows us to test whether repeated play shifts players from local reaction to global containment planning.

3.2 Metric Design

Principles. We use process-level metrics to make experience-sensitive behavioral changes observable. Rather than treating final outcomes such as win rate, score, or game length as the only evidence of improvement, these metrics characterize how players select actions across repeated games. They are computed from observable actions and can be applied consistently to both human players and language agents. Together, they connect game-specific behavior to a shared cross-game hypothesis: repeated gameplay should move players away from locally attractive but strategically weak actions and toward globally stronger decisions.

Cross-Game Greedy-to-Global Metrics We define two cross-game metrics to quantify whether repeated play moves decisions from local heuristics toward global evaluation. Greedy Value Difference (GVD) measures the average improvement of a player's actions over a game-specific greedy baseline, while Greedy Trap Avoidance (GTA) focuses on states where that greedy baseline leads to a strategically poor action.

Let s_t denote the game state at decision step t, a_t denote the player's action, and $A(s_t)$ denote the legal action set. For each game, we define a greedy baseline action $g_t \in A(s_t)$ that captures the natural local heuristic in that environment. We also define a

Game	Metric	Strategic Signal	Computation
Four in a Row	EAR	Solver alignment.	Fraction of player moves that match the solver-best column, measuring overall alignment with globally strong play.
	CPQ	Critical-state decision quality.	Accuracy on positions where the solver-best action is substantially better than the alternatives, focusing on high-stakes decisions.
	OCI	Opening center control.	Center preference score over the player's early moves, with more central columns receiving higher scores because they participate in more potential winning lines.
Othello6	MCI	Mobility control.	Difference between the player's future legal-move count and the opponent's future legal-move count after the player's move, measuring whether the player preserves options while restricting the opponent.
	CPQ	Search-critical decision quality.	Accuracy on search-critical positions where the search-best move is substantially better than the alternatives, capturing decision quality when immediate flip maximization may be misleading.
CircleCat	EPD	Escape-path lengthening.	Increase in the cat's shortest path distance to any boundary cell after the wall is placed, computed from the pre- and post-wall board states.
	EPC	Escape-option reduction.	Reduction in the number of boundary cells reachable by the cat after the wall is placed, measuring whether the wall removes alternative exits.
	CAR	Reachable-region compression.	Reduction in the size of the region reachable from the cat's current position after the wall is placed, measuring global containment of the cat's escape area.

Table 2: Game-specific behavioral metrics used to diagnose how strategic improvement occurs within each environment. These metrics complement the cross-game GVD and GTA measures by capturing environment-specific forms of strategic behavior. All listed metrics are computed from action traces, and larger values indicate stronger behavior along the corresponding strategic dimension.

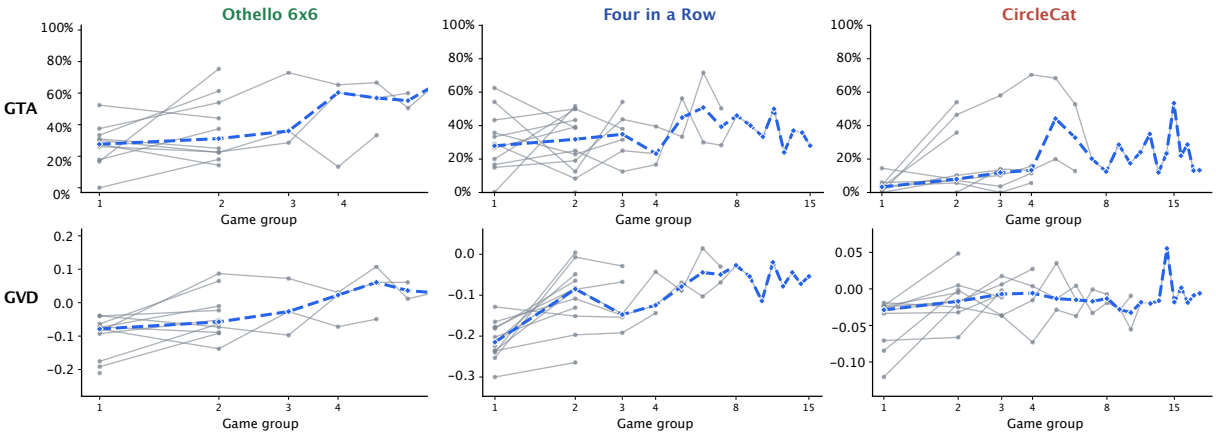


Figure 1: Human trajectories on cross-game greedy-to-global metrics. Gray lines show individual participants, and the dashed blue line shows the human median across game groups.

game-specific action evaluator $V(s, a)$, normalized so that larger values indicate actions that are better for the player. Depending on the environment, V may be derived from a solver, a depth-limited search procedure, or a spatial containment score.

Greedy Value Difference (GVD) measures whether the player's action is better than the game-specific greedy baseline. Because the raw evaluator scale differs across games, we use a softmax-normalized value difference:

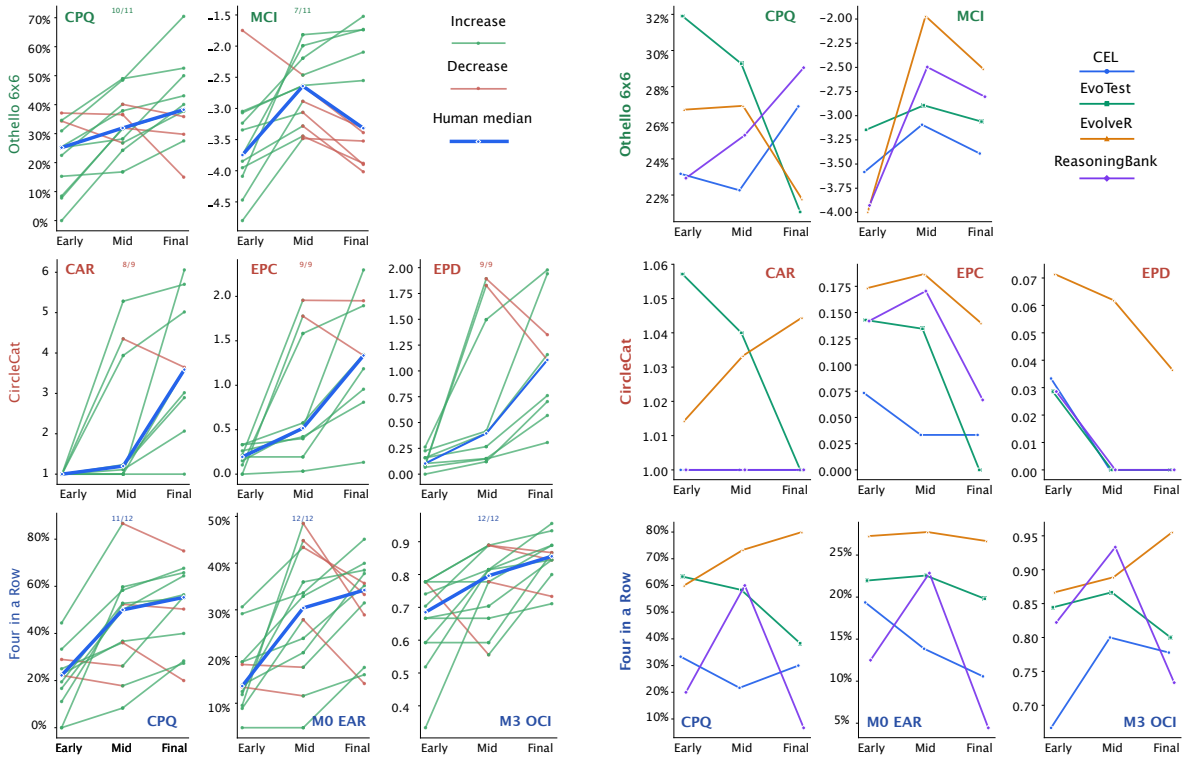
$$\mathrm{GVD} = \frac{1}{|\mathcal{T}|} \sum_{t \in \mathcal{T}} \left[P_t(a_t) - P_t(g_t) \right], \qquad P_t(a) = \frac{\exp(V(s_t, a)/\tau_s)}{\sum_{a' \in A(s_t)} \exp(V(s_t, a')/\tau_s)}. \tag{1}$$

Here $V(s_t, a)$ is the game-specific action evaluator, g_t is the locally greedy action, and τ_s is a temperature parameter. Higher GVD indicates that the player chooses actions with greater normalized value than the greedy baseline.

Greedy Trap Avoidance (GTA) focuses only on states where the greedy baseline is substantially worse than some available alternative. We define the set of greedy-trap opportunities as

$$\mathcal{T}_{\mathrm{trap}} = \left\{ t \in \mathcal{T} : \max_{a \in A(s_t)} V(s_t, a) - V(s_t, g_t) \geq \tau \right\}, \tag{2}$$

where τ is a game-specific threshold for identifying states in which the greedy action is meaningfully suboptimal. On these states, GTA measures the frac-



(a) Human trajectories. (b) Agent trajectories.

Figure 2: Game-specific behavioral metrics for humans and self-evolving agents. The left panel shows human trajectories across Early, Mid, and Final stages, where thin lines denote individual participants and the thick blue line denotes the human median. The right panel shows the corresponding trajectories of self-evolving agent methods over the same stages. Panel annotations report the fraction of runs or participants whose metric improves.

tion of times the player avoids the greedy trap by choosing an action that is better than the greedy baseline:

$$\mathrm{GTA} = \frac{1}{|\mathcal{T}_{\mathrm{trap}}|} \sum_{t \in \mathcal{T}_{\mathrm{trap}}} \mathbf{1}\left[V(s_t, a_t) - V(s_t, g_t) > 0\right]. \tag{3}$$

Thus, GVD captures the magnitude of average improvement over the greedy baseline, whereas GTA captures the frequency with which players avoid clearly harmful greedy choices.

Game-Specific Behavioral Metrics The cross-game metrics GVD and GTA measure whether players move away from locally greedy actions toward globally stronger decisions. To further characterize how this shift occurs within each environment, we define a small set of game-specific behavioral metrics targeting the main strategic structure of each game. These metrics are diagnostic complements rather than separate evaluation goals: they indicate whether improvement is associated with solver alignment, critical-state decision quality, mobility control, or spatial containment. Table 2 summarizes these metrics, their strategic interpretation, how they are computed from action traces, and the preferred direction of change. Detailed computation procedures for all metrics are provided in the supplementary appendix.

4 Results

4.1 Human Gameplay

Participants and protocol. We collected human gameplay trajectories for the three environments from 32 participants, including master and doctoral students, resulting in a total of 709 games. After filtering incomplete or invalid sessions, the retained data consist of 12 participants for Four in a Row, 11 for Othello6, and 9 for CircleCat. To reduce the influence of pre-existing game-specific strategies, participants were required to report no recent experience with the corresponding games. Each participant completed at least 11 full games. For analysis, we group consecutive games into five-game bins. We divide each trajectory into Early, Mid, and Final stages, corresponding to the first three games, the middle five games, and the final three games, respectively; when participants played

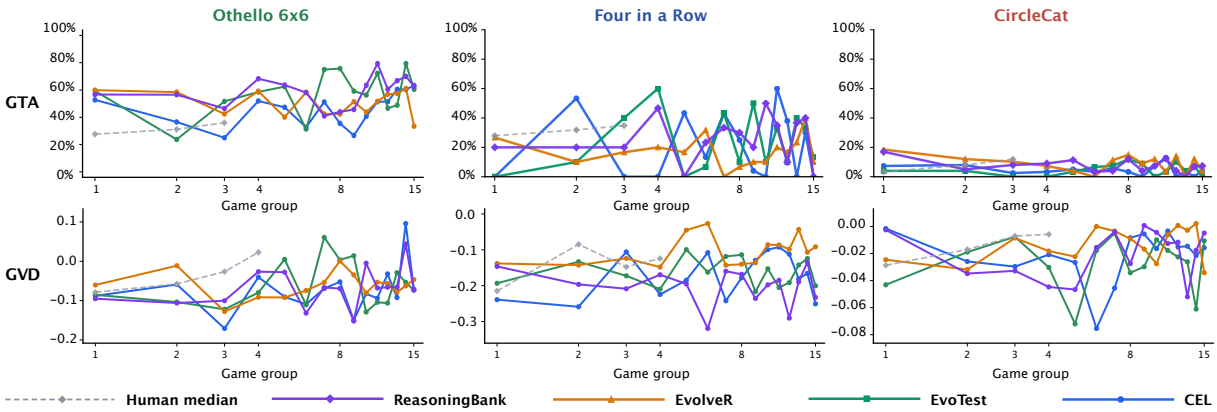


Figure 3: Cross-game greedy-to-global metrics for self-evolving agents. Solid lines show agent trajectories across repeated gameplay, and the dashed gray line shows the human median for reference.

additional games, later bins are included in the full learning curves.

Cross-game greedy-to-global trends. Figure 1 shows the human trajectories for the cross-game greedy-to-global metrics, GTA and GVD. Across the three environments, the median human trajectory improves within the first two game groups, suggesting that participants begin to move away from locally greedy heuristics after limited gameplay experience. The form of this shift differs by environment: in Othello6, improvement corresponds to avoiding flip-greedy traps; in Four in a Row, it corresponds to selecting moves with stronger global or solver value; and in CircleCat, the noisier trajectory still suggests an early movement from proximity-based blocking toward more global containment decisions. Thus, the cross-game metrics indicate that repeated play changes how participants evaluate actions, rather than only making them more familiar with the interface.

Game-specific behavioral changes. Figure 2a further examines whether this greedy-to-global shift is accompanied by interpretable changes in game-specific strategic behavior. The results show broad improvement along the strategic dimensions targeted by each environment. In Othello6, CPQ improves for $10/11$ participants and MCI improves for $7/11$, indicating gains in search-critical decisions and mobility-aware play. In CircleCat, containment metrics improve for most or all participants: CAR improves for $8/9$, while EPC and EPD improve for $9/9$, showing that repeated play helps participants reduce reachable space and escape options. In Four in a Row, CPQ improves for $11/12$ participants, while EAR and OCI improve for $12/12$, suggesting stronger solver alignment and increased early center control. Together, these results show that human learning is visible not only in cross-game greedy-to-global metrics, but also in the environment-specific mechanisms through which better play emerges.

Takeaway. The human data provide a behavioral reference for experience-sensitive game learning. Human players tend to move from simple local heuristics toward more global decision criteria over repeated play, and this shift is visible in both cross-game and game-specific metrics. This provides a target for adaptive language agents: an agent should not only improve final performance, but also exhibit durable and interpretable changes in decision-making behavior.

4.2 Evaluating Self-Evolving Agents

We next evaluate whether recent self-evolving language agents exhibit the experience-sensitive behavioral changes observed in human gameplay. We consider four representative agent designs: CEL (Wang et al., 2025), EvoTest (He et al., 2026), EvolveR (Wu et al., 2025), and ReasoningBank (Ouyang et al., 2026). For each agent, we run repeated gameplay in Four in a Row, Othello6, and CircleCat, collecting 128 gameplay episodes under the same logging and evaluation protocol. We compute both the cross-game greedy-to-global metrics, GVD and GTA, and the game-specific behavioral metrics in Table 2. Figures 3 and 2b summarize the resulting trajectories, with the human median included only as a behavioral reference.

Across the three games, current self-evolving agents do not exhibit the stable behavioral improvement observed in human trajectories. Although individual methods sometimes improve

over short stretches, the gains are often noisy, method-specific, and not consistently maintained across later episodes. This instability appears in both types of metrics: GVD and GTA show that agents only intermittently move away from locally greedy actions, while the game-specific metrics show that improvements in solver alignment, mobility control, critical-state decisions, or spatial containment can regress after temporary gains. Thus, the issue is not simply that agents achieve lower final performance than humans, but that their behavioral changes are less durable across repeated gameplay. The evaluated agents sometimes discover better actions, but they do not reliably consolidate these discoveries into stable decision policies. In contrast to the more consistent greedy-to-global shift observed in human gameplay, current self-evolution mechanisms remain limited in their ability to turn gameplay experience into persistent strategic change.

4.3 Why Self-Evolving Agents Fail

The failures of self-evolving agents suggest that their weaknesses are better understood as a hierarchy of bottlenecks rather than a single failure mode. We therefore perform a closer qualitative analysis of agent failures and organize the recurring patterns into a four-level hierarchy, ranging from low-level state simulation errors to high-level failures of experience integration.

Bottlenecks in self-evolving gameplay. Qualitative traces suggest four recurring bottlenecks that prevent agents from turning experience into stable strategic improvement. **Level 1: the simulation bottleneck** concerns whether the agent can reliably compute the next state after an action; for example, in Four in a Row, agents may verbalize immediate wins or blocks but still miscompute gravity, landing positions, or line completion. **Level 2: the evaluation bottleneck** concerns whether the agent can compare candidate actions by downstream value, as reflected by metrics such as CPQ, MCI, EPD, EPC, and CAR, which test whether agents move beyond locally attractive actions toward solver-aligned, mobility-aware, or containment-aware decisions. **Level 3: the proceduralization bottleneck** concerns whether useful strategic language becomes executable decision-time constraints: agents often store principles such as checking threats, preserving mobility, sealing gateways, or cutting shortest paths, but do not consistently translate them into procedures that filter moves, verify threats, or enforce global constraints. **Level 4: the experience-integration bottleneck** concerns whether repeated interaction produces stable behavioral change; even after accumulating substantial memory or reflection text, agents may repeat structurally similar errors, such as tactical mis-simulation in Four in a Row, near-boundary emergency blocking in CircleCat, or endgame mobility collapse in Othello6.

This hierarchy explains why self-evolving agents can sound strategically sophisticated while still failing behaviorally. A failure at Level 1 corrupts the state information used by later reasoning. A failure at Level 2 prevents the agent from choosing among legal actions by future value. A failure at Level 3 prevents verbal strategies from becoming operational policies. Finally, a failure at Level 4 prevents accumulated experience from producing persistent behavioral adaptation across episodes. Detailed qualitative traces for each level are provided in the supplementary material, where we further decompose these four bottlenecks into finer failure modes and episode-level case studies.

Taken together, these failures suggest that current self-evolving agents learn the vocabulary of strategy more readily than the executable procedures required to apply strategy. Their central limitation is not the absence of reflection, but the failure to convert reflection into reliable, verifiable, and reusable changes in decision-making behavior.

5 Conclusion

We introduced experience-sensitive game learning as a framework for measuring how humans and language agents change their decision-making through repeated gameplay. Our human trajectories show clear and interpretable behavioral changes: players increasingly avoid greedy traps and improve along strategic dimensions such as solver alignment, mobility control, and spatial containment. In contrast, recent self-evolving language agents show noisier and less durable gains. Although self-evolving agents often generate plausible strategic reflections, their main bottleneck remains converting gameplay experience into executable, reusable, and persistent behavioral change across episodes. Overall, our environments and metrics provide a shared setting for comparing human learning and agent self-evolution through observable decision-making behavior.

Limitations

Limited number of environments. A primary limitation of this work is that our detailed analysis is conducted on three game environments. Although this number is modest, it reflects a deliberate choice rather than an attempt to cover games broadly. In the early stage of this project, we experimented with a wider range of games, including 2048, Merge Fall, Four in a Row, Othello, Tic-Tac-Toe, Sudoku, CrossNumber, and Nuts and Bolts. However, we found that only a subset of games satisfied the requirements of the present study: they needed to support repeated play, expose reusable strategic structure, produce interpretable action traces, and admit process-level behavioral metrics beyond final score or win rate.

Some games were too difficult for the type of experience-sensitive analysis targeted in this paper. For example, Go and Chess require substantial domain knowledge and long-term study, making them unsuitable for observing meaningful learning from a small number of repeated games, both for current agents and for many human participants. Other games, such as 2048 and Merge Fall, involve long-horizon action sequences in which a single episode may contain hundreds of operations. Although such games are interesting and we have collected preliminary data for them, their feedback signals are sparse and delayed, making it difficult to attribute behavioral changes to specific strategic shifts at the current stage.

Therefore, the limited number of environments remains an important limitation. Our current benchmark prioritizes validated environments with clear behavioral metrics over broad game coverage. We are actively extending the evaluation protocol and benchmark to include more verified environments, with the goal of supporting a broader study of self-evolving agents. In this paper, however, we intentionally favor quality over quantity, because the validity and interpretability of each environment are crucial for analyzing experience-sensitive behavioral change.

Variability and cost of human data. A second limitation is the difficulty of collecting and interpreting human gameplay data. Human trajectories are valuable because they provide a behavioral reference for experience-sensitive learning, but they are also inherently noisy. Participants differ substantially in attention, prior knowledge, motivation, and learning style. Some participants may become distracted during gameplay, producing trajectories that appear random or inconsistent. Others may learn, but in a direction that does not align with the intended strategic principle of the environment. As a result, human learning curves can contain substantial variability, even when the task interface and instructions are fixed.

We observed such variability in the collected trajectories. Understanding it more deeply will require a larger and more diverse human dataset, as well as more careful modeling of individual differences. However, collecting human gameplay trajectories is costly, especially when the goal is not merely to obtain final scores but to observe repeated-game behavioral change over time. We therefore view the current human study as an initial reference rather than a definitive characterization of human learning. Expanding the scale, diversity, and reliability of human data is an important direction for future work.

Comparison with trainable game agents. Another limitation is that we do not compare against reinforcement-learning agents or language models trained directly in the target game environments. This is a deliberate scope choice. Reinforcement learning follows a different adaptation paradigm: an agent is optimized for a specific environment through repeated training, often achieving performance far beyond ordinary human players and, in some games, even expert-level play. However, such environment-specific optimization does not directly reflect the type of experience-sensitive behavioral change studied in this work. Our goal is not to test whether a trainable policy can eventually master a game, but to examine whether language agents can use limited repeated interaction to revise their decision-making in a way that is behaviorally comparable to human learning.

This distinction is important because large language model agents occupy a different setting from conventional trainable game agents. They enter the environment with broad prior knowledge, natural-language reasoning, memory, and reflection mechanisms, but without gradient-based training on the specific game. We therefore focus on whether these agents can convert gameplay experience into reusable strategic behavior through interaction alone. Comparing this form of interaction-based adaptation with fully trainable reinforcement-learning systems is an important direction for future work, but it requires a different experimental proto-

col and should be interpreted separately from the experience-sensitive learning setting studied here.

Ethical Considerations

This work involves the collection and analysis of human gameplay trajectories. Participants were informed about the purpose of the study, the nature of the gameplay tasks, and the types of data collected before participation. Participation was voluntary, and participants could stop playing at any time. The collected data consist of gameplay interaction traces, such as game states, actions, and episode-level outcomes. We did not collect sensitive personal attributes, and our analysis focuses on aggregate behavioral patterns rather than individual performance evaluation. The study procedure was reviewed and approved by the project supervisor through the available internal project oversight process before data collection.

To protect participant privacy, human gameplay records were anonymized before analysis. Any identifiers used during data collection were removed or replaced with anonymous participant IDs. The reported results are aggregated across participants, and no individual participant is identifiable from the figures, tables, or qualitative analysis. The tasks used in this study are simple puzzle or board-game environments and involve minimal risk beyond ordinary interaction with computer games.

Because human gameplay can vary substantially across individuals, we use the data as a behavioral reference for studying experience-sensitive learning rather than as a normative judgment of participant ability. We do not use the data to make claims about participants' cognitive capacity, intelligence, or personal traits. If required by the venue or institution, the collection protocol should be reviewed under the applicable human-subjects research guidelines.

References

William G Chase and Herbert A Simon. 1973. Perception in chess. *Cognitive psychology*, 4(1):55–81.

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

Marc-Alexandre Côté, Ákos Kádár, Xingdi Yuan, Ben Kybartas, Tavian Barnes, Emery Fine, James Moore, Matthew Hausknecht, Layla El Asri, Mahmoud Adada, Wendy Tay, and Adam Trischler. 2019. Textworld: A learning environment for text-based games. In *Computer Games*, pages 41–75, Cham. Springer International Publishing.

Ishita Dasgupta, Eric Schulz, Noah D. Goodman, and Samuel J. Gershman. 2018. Remembrance of inferences past: Amortization in human hypothesis generation. *Cognition*, 178:67–81.

Ishita Dasgupta, Eric Schulz, Joshua B Tenenbaum, and Samuel J Gershman. 2020. A theory of learning to infer. *Psychological review*, 127(3):412.

Samuel Gershman and Noah D. Goodman. 2014. Amortized inference in probabilistic reasoning. In *Proceedings of the 36th Annual Meeting of the Cognitive Science Society, CogSci 2014, Quebec City, Canada, July 23-26, 2014*. cognitivesciencesociety.org.

Samuel J. Gershman, Eric J. Horvitz, and Joshua B. Tenenbaum. 2015. Computational rationality: A converging paradigm for intelligence in brains, minds, and machines. *Science*, 349(6245):273–278.

Fernand Gobet and Herbert A Simon. 1996. Templates in chess memory: A mechanism for recalling several boards. *Cognitive Psychology*, 31(1):1–40.

Fernand R. Gobet and Herbert A. Simon. 1998. Pattern recognition makes search possible: Comments on holding (1992). *Psychological Research*, 61:204–208.

Leon Guertler, Bobby Cheng, Simon Yu, Bo Liu, Leshem Choshen, and Cheston Tan. 2025. Textarena. *arXiv preprint arXiv:2504.11442*.

Matthew Hausknecht, Prithviraj Ammanabrolu, Marc-Alexandre Côté, and Xingdi Yuan. 2020. Interactive fiction games: A colossal adventure. *Proceedings of the AAAI Conference on Artificial Intelligence*, 34(05):7903–7910.

Yufei He, Juncheng Liu, Yue Liu, Yibo Li, Tri Cao, Zhiyuan Hu, Xinxing Xu, and Bryan Hooi. 2026. Evotest: Evolutionary test-time learning for self-improving agentic systems. In *The Fourteenth International Conference on Learning Representations*.

Lanxiang Hu, Mingjia Huo, Yuxuan Zhang, Haoyang Yu, Eric P. Xing, Ion Stoica, Tajana Rosing, Haojian Jin, and Hao Zhang. 2026. lmgame-bench: How good are LLMs at playing games? In *The Fourteenth International Conference on Learning Representations*.

Lanxiang Hu, Qiyu Li, Anze Xie, Nan Jiang, Ion Stoica, Haojian Jin, and Hao Zhang. 2025. Gamearena: Evaluating LLM reasoning through live computer games. In *International Conference on Learning Representations*, volume 2025, pages 33278–33309.

Yibo Li, Zijie Lin, Ailin Deng, Xuan Zhang, Yufei He, Shuo Ji, Tri Cao, and Bryan Hooi. 2026. Just-in-time reinforcement learning: Continual learning in

llm agents without gradient updates. *arXiv preprint arXiv:2601.18510*.

Falk Lieder and Thomas L. Griffiths. 2020. Resource-rational analysis: Understanding human cognition as the optimal use of limited computational resources. *Behavioral and Brain Sciences*, 43:e1.

Zhanzhi Lou, Hui Chen, Yibo Li, Qian Wang, and Bryan Hooi. 2026. Learning to learn-at-test-time: Language agents with learnable adaptation policies. *arXiv preprint arXiv:2604.00830*.

Chang Ma, Junlei Zhang, Zhihao Zhu, Cheng Yang, Yujiu Yang, Yaohui Jin, Zhenzhong Lan, Lingpeng Kong, and Junxian He. 2024. Agentboard: An analytical evaluation board of multi-turn llm agents. In *Advances in Neural Information Processing Systems*, volume 37, pages 74325–74362. Curran Associates, Inc.

Bodhisattwa Prasad Majumder, Bhavana Dalvi Mishra, Peter Jansen, Oyvind Tafjord, Niket Tandon, Li Zhang, Chris Callison-Burch, and Peter Clark. 2024. CLIN: A continually learning language agent for rapid task adaptation and generalization. In *First Conference on Language Modeling*.

Siru Ouyang, Jun Yan, I-Hung Hsu, Yanfei Chen, Ke Jiang, Zifeng Wang, Rujun Han, Long Le, Samira Daruki, Xiangru Tang, Vishy Tirumalashetty, George Lee, Mahsan Rofouei, Hangfei Lin, Jiawei Han, Chen-Yu Lee, and Tomas Pfister. 2026. Reasoningbank: Scaling agent self-evolving with reasoning memory. In *The Fourteenth International Conference on Learning Representations*.

Davide Paglieri, Bartł omiej Cupiał, Samuel Coward, Ulyana Piterbarg, Maciej Wołczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob Foerster, Jack Parker-Holder, and Tim Rocktaeschel. 2025. BALROG: Benchmarking agentic LLM and VLM reasoning on games. In *International Conference on Learning Representations*, volume 2025, pages 96666–96702.

Dongmin Park, Minkyu Kim, Beongjun Choi, Junhyuck Kim, Keon Lee, Jonghyun Lee, Inkyu Park, Byeong-Uk Lee, Jaeyoung Hwang, Jaewoo Ahn, Ameya Sunil Mahabaleshwarkar, Bilal Kartal, Pritam Biswas, Yoshi Suhara, Kangwook Lee, and Jaewoong Cho. 2026. Orak: A foundational benchmark for training and evaluating LLM agents on diverse video games. In *The Fourteenth International Conference on Learning Representations*.

Jiajun Shi, Jian Yang, Jiaheng Liu, Xingyuan Bu, Jiangjie Chen, Junting Zhou, Kaijing Ma, Zhoufutu Wen, Bingli Wang, Yancheng He, Liang Song, Hualei Zhu, Shilong Li, Xingjian Wang, Wei Zhang, Ruibin Yuan, Yifan Yao, Wenjun Yang, Yunli Wang, and 10 others. 2025. KORGym: A dynamic game platform for llm reasoning evaluation. In *Advances in Neural Information Processing Systems*, volume 38, pages 161286–161314. Curran Associates, Inc.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2023. Reflexion: language agents with verbal reinforcement learning. In *Advances in Neural Information Processing Systems*, volume 36, pages 8634–8652. Curran Associates, Inc.

Wenjie Tang, Yuan Zhou, Keyan Cheng, Erqiang Xu, Liquan Xiao, and Minne Li. 2026. Dsgbench: A diverse strategic game benchmark for evaluating llm-based agents in complex decision-making environments. In *ICASSP 2026 - 2026 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pages 16987–16991. IEEE.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2024. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research*.

Sai Wang, Yu Wu, and Zhongwen Xu. 2025. Cogito, ergo ludo: An agent that learns to play by reasoning and planning. *arXiv preprint arXiv:2509.25052*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*, volume 35, pages 24824–24837. Curran Associates, Inc.

Rong Wu, Xiaoman Wang, Jianbiao Mei, Pinlong Cai, Daocheng Fu, Cheng Yang, Licheng Wen, Xuemeng Yang, Yufan Shen, Yuxin Wang, and Botian Shi. 2025. Evolver: Self-evolving llm agents through an experience-driven lifecycle. *arXiv preprint arXiv:2510.16079*.

Sixiong Xie, Zhuofan Shi, Haiyang Shen, Gang Huang, Yun Ma, and Xiang Jing. 2026. M3-bench: Process-aware evaluation of llm agents social behaviors in mixed-motive games. *arXiv preprint arXiv:2601.08462*.

Shunyu Yao, Rohan Rao, Matthew Hausknecht, and Karthik Narasimhan. 2020. Keep CALM and explore: Language models for action generation in text-based games. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 8736–8754, Online. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 11809–11822. Curran Associates, Inc.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik R Narasimhan, and Yuan Cao. 2023b. ReAct: Synergizing reasoning and acting in language models. In *The Eleventh International Conference on Learning Representations*.

Xiaopeng Yuan, Xingjian Zhang, Ke Xu, Yifan Xu, Lijun Yu, Jindong Wang, Yushun Dong, and Haohan Wang. 2025. Tracing llm reasoning processes with strategic games: A framework for planning, revision, and resource-constrained decision making. *arXiv preprint arXiv:2506.12012*.

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

Appendix

A Detailed Failure Analysis of Self-Evolving Agents

This appendix provides a qualitative failure analysis of self-evolving agents around later episodes. We organize the analysis using the four-level bottleneck hierarchy introduced in the main paper. The four levels are nested. Level 1 concerns whether the agent can simulate the next state. Level 2 concerns whether it can evaluate candidate actions by future value. Level 3 concerns whether verbal strategies become executable action constraints. Level 4 concerns whether accumulated experience produces persistent behavioral change across episodes.

A.1 Level 1: Simulation Bottleneck

Failure Analysis 1: State-Transition Simulation Failure. Explanation. This failure occurs when an agent knows the game rules in language but cannot reliably simulate the next state after an action. In board games, this means that the agent may correctly state that it should check legal moves, immediate wins, blocks, gravity, or line completion, yet still miscompute the actual board transition. This is the lowest-level bottleneck in the hierarchy. If the agent cannot accurately compute the state after a candidate move, then its later strategic evaluation is built on an unreliable representation of the game.

Case: Four in a Row, CEL, Episode 50. A clear example appears in Four in a Row, Episode 50, with the CEL agent. By this point, the agent had already accumulated a very long strategy text, with `strategy_before_len = 29012`. Nevertheless, the episode ended in `LOSE` with `score = bot_win`. The failure is therefore not an early-stage failure caused by lack of exposure. It occurs after substantial accumulated strategy text.

In step 13, the agent selected action 4 and described it as an immediate winning move. Its reasoning claimed that the piece would land in column 4, row 2, completing a vertical four across rows 2, 3, 4, and 5. However, the game did not end with an agent win. The final result was still `bot_win`. This indicates that the agent's action choice was based on a faulty simulation of the board transition, not merely on a weak high-level strategy.

The same episode also shows instability in tactical scanning. In step 17, the agent cycled through several possible immediate wins and bot threats. It first considered column 1 as a possible immediate winning move, then rejected it. It then checked whether the bot had an immediate win in column 5, rejected that as well, and eventually identified column 3 as a possible bot winning threat. This pattern shows that the agent knows the correct verbal routine, namely to scan immediate wins and blocks, but the scan itself is unstable.

Interpretation. This case shows that late-stage self-evolving agents may possess the language of tactical reasoning without a reliable internal state simulator. The agent can say that it is checking immediate threats, but it does not consistently compute piece placement, gravity, or line completion correctly. In the four-level hierarchy, this is a Level 1 failure because it precedes strategic evaluation. If the agent cannot trust its own next-state computation, then solver alignment and critical-state decision quality cannot be stable.

A.2 Level 2: Evaluation Bottleneck

Failure Analysis 2: Counterfactual Search Failure. Explanation. This failure occurs when the agent recognizes that multiple candidate actions must be compared, but does not reliably evaluate their downstream consequences. The agent may say that a state requires concrete sequence analysis, minimax-like reasoning, or future mobility evaluation. However, its chosen action does not reflect a stable comparison among possible future trajectories. This is a Level 2 failure because the agent's problem is not only computing the next state, but assigning value to different possible futures.

This bottleneck is directly connected to our process-level metrics. Four in a Row uses solver alignment and critical-state decision quality to test whether the agent chooses globally strong columns. Othello6 uses mobility control and search-critical decision quality to test whether the agent avoids moves that look good immediately but damage future options. CircleCat uses escape-path lengthening, escape-option reduction, and reachable-region compression to test whether the agent evaluates walls by their effect on the global escape structure.

Case: Othello6, ReasoningBank, Episode 50. In Othello6, Episode 50, the ReasoningBank agent entered the episode with substantial memory and strategy text. The metadata reports `strategy_before_len = 15199`, yet the final score was `12:24`, a large loss. The agent's memory already included a corner-adjacency audit, instructing it to test each open corner for the opponent's next move using the exact bracket rule.

The most revealing moment occurs in step 27.

Level	Bottleneck	Detailed failure modes
Level 1	Simulation bottleneck	State-transition simulation failure: the agent verbally checks legal moves, immediate wins, blocks, or line completion, but fails to reliably simulate the next state.
Level 2	Evaluation bottleneck	Counterfactual search failure; reactive local-salience failure. The agent recognizes that future consequences matter, but still fails to compare candidate actions by their downstream value.
Level 3	Proceduralization bottleneck	Verbal strategy without proceduralization; abstraction grounding failure. The agent stores useful strategic principles, but does not reliably convert them into action-time constraints.
Level 4	Experience-integration bottleneck	Long-horizon credit assignment failure; memory retrieval and compression failure. The agent accumulates memory and reflection, but does not turn them into persistent experience-sensitive behavioral change.

Table 3: A four-level hierarchy of failure modes in self-evolving agents. The levels are nested: reliable state simulation supports future-oriented evaluation; evaluation supports proceduralized strategy execution; and proceduralized execution is required for experience-sensitive behavioral change across episodes.

The agent explicitly recognized that it was in a high-importance endgame state and that only two legal moves remained. It stated that the best action should be chosen through concrete sequence analysis rather than broad heuristics such as central control or edge shape. This is the right verbal diagnosis of the situation. However, the later trajectory still produced forced passes in step 31 and step 33, where Black had no legal move and had to pass.

Interpretation. The agent can name the required evaluation procedure, but it does not execute that procedure reliably. If it had successfully compared the legal moves by their future mobility consequences, it should have been able to anticipate which move would preserve legal options and which move would collapse into forced passes. The failure therefore lies in counterfactual evaluation: the agent describes future-sensitive reasoning, but its action does not reflect a correct comparison of future states.

Failure Analysis 3: Reactive Local-Salience Failure. Explanation. This failure occurs when the agent remains driven by the most locally urgent or visually salient threat, rather than evaluating the global structure of the game. It is especially clear in CircleCat. The natural novice heuristic is to place a wall near the cat, but effective play requires reasoning about the global escape structure. A wall that looks urgent locally may leave several escape corridors open, while a farther wall may close a bottleneck, remove multiple reachable exits, or shrink the cat's reachable region.

Case: CircleCat, CEL, Episode 51. CircleCat, Episode 51, shows this failure recurring after prior reflection. The agent entered the episode with `strategy_before_len = 20965`, meaning that it had already updated its strategy after the Episode 50 failure. The result was still `score = 0`, and the strategy grew again to `strategy_after_len = 21670`.

In step 7, the cat was located at `(2,7)` in the upper-right region. The agent correctly identified that the nearest escape routes were toward the top and right edges. By step 9, the cat had moved to `(1,6)`, very close to the top boundary. At that point, two boundary cells, `(0,5)` and `(0,6)`, were open. The agent selected a move that blocked one of these immediate exits, but it explicitly acknowledged that the other exit remained open. By step 13, the cat reached `(0,5)`, which placed it on the boundary and made the position effectively lost.

Interpretation. The agent does recognize danger, but only after the danger has become locally urgent. By the time the cat reaches row 1 with two open boundary exits, blocking one cell is insufficient. The correct strategic behavior would have been to reshape the top-right escape structure earlier, before the cat entered the near-boundary zone. This is a Level 2 evaluation failure because the agent evaluates the most immediate threat but fails to evaluate the global consequences of earlier wall placements.

A.3 Level 3: Proceduralization Bottleneck

Failure Analysis 4: Verbal Strategy Without Proceduralization. Explanation. This failure occurs when the agent stores useful strategic principles in natural language, but these principles do not become executable action-selection procedures. The agent may write down strong strategies such as preserving mobility, sealing shared gateways,

cutting multiple shortest paths, or checking immediate wins and blocks. However, these remain descriptive rather than operational. They do not become a reliable decision-time structure of the form: recognize a state pattern, trigger a rule, filter candidate actions, verify the consequence, and select the move.

Case: CircleCat, CEL, Episode 50. CircleCat, Episode 50, provides a strong example. Before the episode, the CEL agent's strategy text already contained global containment principles. It stated that the best walls should build continuous structures, close gaps between wall groups, seal gateways shared by multiple shortest paths, and reduce the cat's live escape tree rather than merely occupy cells near the cat. The metadata reports `strategy_before_len = 18962` and `strategy_after_len = 20965`, showing that the agent had accumulated substantial strategic text.

However, the actual gameplay did not implement these principles early enough. In step 7, the cat was at `(2,5)`, and the agent recognized that the cat was already in the upper-middle approach zone. The agent also recognized that row 1 was only one move away and that row 0 would then be the boundary. In step 9, the cat reached `(1,4)`, and only then did the agent place a wall at `(0,4)` to remove an immediate top-boundary escape. By step 11, the cat had reached `(0,5)`, which was already on the boundary and effectively ended the game.

Interpretation. This case demonstrates that the agent has learned the language of global containment but has not proceduralized it. The agent knows that it should seal shared gateways and cut multiple shortest paths, but it does not convert this principle into early wall placement. The strategy remains a textual principle, not a constraint that forces the agent to act before the cat reaches the near-boundary zone.

Failure Analysis 5: Abstraction Grounding Failure. Explanation. This failure occurs when the agent can name abstract strategic concepts, but those concepts are not grounded in concrete state features or action constraints. For example, an agent may use terms such as mobility, corner safety, shortest paths, feeder bands, bottlenecks, or reachable exits. However, these concepts do not consistently trigger the right behavior in the relevant state. The abstraction is linguistically available, but behaviorally under-grounded.

This failure is distinct from not knowing the concept. The agent may correctly state that mobility is important, or that a shared gateway should be sealed. The failure is that the concept does not become a state recognizer or a candidate-action constraint.

Case: Othello6, ReasoningBank, Episode 51. Othello6, Episode 51, illustrates this failure. The agent's strategy already included corner-related and mobility-related rules. It contained instructions to audit every edge or corner-adjacent move, and after losing a corner, to switch from shape-based play to a corner race. The episode metadata reports `strategy_before_len = 15223`, but the final score was still `11:25`.

In step 29, the agent explicitly described the position as a low-mobility endgame. It also stated that the key question was which move would preserve Black's ability to keep playing after White's best reply. This is the correct abstract diagnosis. However, in step 31 and step 33, Black had no legal placement and had to pass. The reflection then noted that the game collapsed when Black ran out of legal moves and White finished the remaining edge squares.

Interpretation. The agent can use the word mobility correctly, but it does not consistently ground mobility in concrete future legal-move counts. The concept functions as an after-the-fact explanation rather than as a move-selection constraint. A grounded use of mobility would require the agent to reject actions that leave Black with no legal moves after White's likely reply, unless an immediate tactical gain justifies that risk. The trace suggests that this grounding is missing.

A.4 Level 4: Experience-Integration Bottleneck

Failure Analysis 6: Long-Horizon Credit Assignment Failure. Explanation. This failure occurs when the final loss is caused by decisions made much earlier, but the agent's reflection focuses mainly on the terminal symptom. In games such as Othello6, an endgame forced pass is usually not the true causal mistake. It is the visible consequence of earlier decisions that reduced future mobility, exposed unstable frontier discs, or allowed the opponent to secure stable corners. If the agent only records that it ran out of moves, without identifying which earlier choices caused the collapse, then the next episode may repeat the same structural error.

Case: Othello6, ReasoningBank, Episodes 50

and 51. Episode 50 shows the first part of this pattern. The agent already had a memory item instructing it to audit corner adjacency and check open corners using the exact bracket rule. Nevertheless, the episode ended with a `12:24` loss. The later steps included repeated forced passes. The reflection also acknowledged that the agent often talked about corner denial, but several move choices still allowed White to take stable corners and convert them into a large endgame advantage.

Episode 51 shows that the issue was not resolved. The agent again described the endgame as low-mobility and identified preservation of future legal moves as the key criterion. Yet step 31 and step 33 again resulted in Black having no legal placement, and the final score was `11:25`. The repeated pattern suggests that the agent did not successfully trace the terminal mobility collapse back to earlier positional concessions.

Interpretation. This is a Level 4 failure because the problem lies in integrating experience across time. The agent's reflection correctly names the symptom, namely forced passes and mobility collapse. However, it does not reliably identify the upstream decisions that produced the symptom. As a result, the agent continues to enter endgames where no legal moves remain. Experience is recorded, but causal credit is not assigned to the earlier choices that need to change.

Failure Analysis 7: Memory Retrieval and Compression Failure. Explanation. This failure concerns the self-evolution mechanism itself. Longer memory does not necessarily imply better adaptation. As strategy, memory, and reflection grow longer, the agent still needs to retrieve the relevant lesson, compress it into an actionable form, and apply it to the current state. If memory remains an unstructured collection of natural-language advice, the most important lessons may be diluted by generic principles, repeated observations, or stale heuristics.

Case: CircleCat, CEL, Episodes 50 and 51. CircleCat Episodes 50 and 51 show this failure clearly. In Episode 50, the CEL agent's strategy grew from `strategy_before_len = 18962` to `strategy_after_len = 20965` after a failure in which the cat reached the top boundary. The agent had already written principles about continuous wall structures, shared gateways, and shortest paths. Yet the episode still failed because the cat moved from `(2,5)` to `(1,4)`, and then to `(0,5)`.

In Episode 51, the agent entered with the updated strategy text, now `strategy_before_len = 20965`. Nevertheless, a structurally similar failure recurred. The cat moved from `(2,7)` to `(1,6)`, gained two immediate boundary exits, and escaped through `(0,5)`. After this episode, the strategy length increased again to `strategy_after_len = 21670`. The memory became longer, but the behavior did not reliably change.

Interpretation. This case suggests that self-evolution may produce more text without producing more usable control. The agent stores the right strategic ideas, and the memory grows after failure, but the next episode still repeats a similar near-boundary emergency. The missing step is retrieval and compression: the agent does not convert the prior failure into a compact decision rule such as "when the cat enters the upper approach zone, prioritize closing shared top exits before row 1 becomes reachable." Without that conversion, accumulated memory remains descriptive rather than behaviorally effective.

A.5 Summary

The detailed cases support the four-level interpretation in the main paper. Some failures occur at the level of state simulation, where the agent cannot reliably compute the immediate consequences of an action. Other failures occur at the level of evaluation, where the agent recognizes that future consequences matter but does not compare candidate actions correctly. Still others occur at the level of proceduralization, where the agent has useful strategic language but does not convert it into action constraints. Finally, repeated failures across consecutive episodes indicate an experience-integration bottleneck: memory and reflection accumulate, but they do not reliably produce stable experience-sensitive behavioral change.

These traces suggest that current self-evolving agents do not simply lack strategic descriptions. Rather, they fail at successive stages of converting experience into behavior: simulating states, evaluating futures, proceduralizing strategies, and integrating prior failures into reusable decision rules.

B Metric Implementation Details

This appendix specifies the metric implementation used in our experiments. All metrics are computed from logged action traces by reconstructing the legal action set at each decision state and comparing

	Four in a Row		Othello6		CircleCat	
Method	First 10	Final 10	First 10	Final 10	First 10	Final 10
Human median	0.0%	10.0%	0.0%	0.0%	20.0%	20.0%
CEL	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
EvoTest	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%
EvolveR	10.0%	0.0%	0.0%	0.0%	10.0%	0.0%
ReasoningBank	0.0%	0.0%	0.0%	0.0%	0.0%	0.0%

Table 4: Win/loss outcome diagnostic for humans and language agents. Agent First 10 and Final 10 are computed over episodes 1–10 and 66–75. Human values are medians over players after per-player first/final window aggregation. The table shows that outcome-level win rate is sparse in these hard games: most language agents show no visible improvement. Thus, win/loss alone provides a weak and insensitive signal for experience-sensitive learning.

the observed action with a game-specific greedy baseline.

B.1 Cross-Game Greedy-to-Global Metrics

For each state, we assign evaluator scores to all legal actions and use a state-wise softmax to normalize scores within the action set. `GVD` is the normalized value difference between the player's action and the best tied greedy action. `GTA` is defined only on trap opportunities, where the best legal action exceeds the greedy baseline by a game-specific threshold; success requires choosing an action with higher evaluator score than the greedy baseline.

B.2 Four in a Row Metrics

We use the Pascal Pons solver as the action evaluator. The greedy baseline is a local line-completion heuristic: for each legal column, we simulate the drop and score the move by the larger of the player's longest extended line and the opponent's longest blocked line. `GTA` uses `TRAP_THRESHOLD = 1`. For game-specific metrics, `EAR` is solver-best agreement, `OCI` measures opening centrality, and `CPQ` uses `CPQ_THRESHOLD = 2` to identify solver-critical positions.

B.3 Othello6 Metrics

The greedy baseline is the legal move that flips the most discs immediately. Legal actions are evaluated by depth-limited minimax with `MINIMAX_DEPTH = 5`. The leaf evaluator combines positional weights, a mobility term $5(|\mathcal{A}_{\text{human}}| - |\mathcal{A}_{\text{bot}}|)$, and an endgame disc-count term $3(n_{\text{human}} - n_{\text{bot}})$ when fewer than ten empty cells remain; terminal outcomes receive scores of approximately ± 10000. `CPQ` and `GTA` use thresholds of `5.0` for search-critical and greedy-trap states, respectively.

The mobility-control metric is computed after the player's move:

$$\text{MCI} = |\mathcal{A}_{\text{human}}(s_{t+1})| - |\mathcal{A}_{\text{bot}}(s_{t+1})|, \quad (4)$$

Other quantities such as dangerous-square avoidance, blunder severity, and max-flip rate were used during metric exploration but are not included in the main metric table.

B.4 CircleCat Metrics

The greedy baseline is the nearest legal wall placement to the cat, with ties resolved by the highest containment score. Candidate walls are evaluated by recomputing the cat's reachable escape structure with breadth-first search. The containment score is a weighted sum of escape-path distance gain, escape-option reduction, and reachable-area reduction: $S(a) = 3.0\,\text{EPD}(a) + 1.5\,\text{EPC}(a) + 0.05\,\text{CAR}(a) + B(a)$. Here $B(a) = 20.0$ if the wall eliminates all reachable boundary cells, and $B(a) = 0$ otherwise. We treat a state as a greedy-trap opportunity when the best containment score exceeds the nearest-wall greedy baseline by at least 1.0.

B.5 Trajectory Aggregation and Filtering

Human games are sorted chronologically per player and aggregated into non-overlapping five-game groups; group curves report the median across player-level group values. Human three-stage figures use the first three, middle five, and final three games. For Othello6 greedy curves, unfinished games with fewer than five human moves are removed. Agent curves use 75 episodes per game, aggregated into five-episode groups; agent three-stage figures use the first three, middle five, and final three episodes. For the win/loss table, agent First 10 and Final 10 correspond to episodes 1–10 and 66–75, while human values are computed per player and summarized by the median.