Cards on the Table: The Winning Move Is Not in the Rules

Anonymous CVPR submission

Paper ID *****

Abstract

Playing a collectible card game well takes more than following the rules. A strong player must perceive the visible board, know which moves are legal, recall what has been played, reason about hidden information, and judge which legal move is best. In these games, individual card effects take precedence over the base rules, so the winning move usually cannot be deduced from the rulebook alone. We introduce ***Cards on the Table****, a benchmark that probes these abilities in language models (LLMs) and vision-language models (VLMs). It is built on* Aurora's Call*, a purpose-built two-player card game whose deterministic engine provides exact ground truth and the full set of legal moves for any state. From recorded games we automatically generate engine-verified questions in* ***seven categories****: perception, move legality, illegal-action diagnosis, public-card tracking, resource reasoning, state-change prediction, and replay reasoning, each posed in matched text (LLM) and image (VLM) form, and we complement the question benchmark with* ***full-game play*** *against scripted bots. Across five open models (Qwen2.5, Llama 3.1, Qwen2.5-VL, InternVL3, and LLaVA-OneVision), the best model answers only* 57.5% *of questions correctly, with counterfactual resource arithmetic below* 40% *for every model; in full games, models beat a random-move bot but lose the large majority of games to simple scripted heuristics that a human defeats* 100% *of the time. We further outline a memory-augmented agent that accumulates card knowledge across games to re-rank moves. The game engine, collected dataset, benchmark code, and memory module will be released.*

1. Introduction

Collectible card games (CCGs) such as *Pokémon*, *Magic: The Gathering* and *Hearthstone* are a popular hobby and a challenge for artificial intelligence [4, 7, 30]. What sets a CCG apart from other games is that it is defined by a large, open-ended library of cards whose printed effects can extend and override the base rules: a card may grant an extra draw, ignore a turn restriction, or claim a win condition the rulebook otherwise forbids. Skilled play therefore depends on knowing which cards exist, what they do, and how they combine, which is knowledge that accumulates over many games. A move that looks strong under the rules alone can be a blunder once the space of possible cards is considered, and vice versa. In short, the winning move is often not in the rules.

As large language models (LLMs) and vision-language models (VLMs) are increasingly used as interactive agents [15, 21, 22], video games have emerged as a natural testbed for the perception, memory, and reasoning such agents require. But a card game stresses several distinct abilities, which end-to-end win-rate metrics collapse into a single score that hides which one failed. To act, a player must perceive the visible board: its cards, resources, and their statuses, and judge which moves are legal under both the rules and the active card effects. The same decision also requires remembering what earlier turns revealed, reasoning about hidden information and remaining resources, and weighing which legal move is best. Existing game benchmarks for foundation models largely report aggregate returns in grid worlds, text games, or commercial video games, or target spatial puzzles [3, 15, 16, 25, 33]; they rarely isolate these sub-skills, and card-game related VLM capabilities remain mostly underexplored.

Our proposed benchmark **Cards on the Table** decomposes card-game competence into distinct abilities and measures each in both text and image form. The benchmark is built on *Aurora's Call*, a two-player CCG we developed with a deterministic, functional card-game engine (Sec. 3). A purpose-built game is deliberate: the engine yields exact ground truth and the complete legal-move set for any turn; the card pool is new, so models cannot have memorized strategy guides during pre-training. From recorded games we sample states, let the engine compute every answer by simulation, and export each in two forms: a structured text prompt (the LLM setting) and rendered board images (the VLM setting). The same infrastructure lets the models *play* full games against scripted opponents, so question-level understanding and game-level competence can be compared directly. The game will also allow us to release the cards,

generated data, and evaluation openly, opening new possibilities for extensions.

Our contributions are threefold:

- **Aurora's Call**, an openly released CCG with a deterministic engine that enables automatic dataset creation across seven categories.
- **Dataset and evaluation**: 7,868 engine-verified questions (3,934 per modality) sampled from recorded games, in matched text and image form, plus a full-game evaluation against scripted bots; we benchmark five open LLMs/VLMs on both.
- **A memory-augmented agent** (Sec. 6) that accumulates card knowledge across games to re-rank candidate moves; ongoing.

2. Related Work

Game-playing AI for card games. Card games are classic testbeds for decision-making under partial information. Specialized systems reach expert or superhuman play through search and reinforcement learning, like Pluribus in poker [1], Suphx in Mahjong [11], DouZero in DouDizhu [31], and RLCard [30] standardizes reinforcement-learning research across card games. Collectible card games add an open-ended, effect-driven card pool that surveys of the genre identify as a distinct AI challenge [4, 7]. These agents are game-specific, trained on hand-engineered state, and judged by playing strength alone; none ask whether a *general* model understands the game it is playing.

LLMs and VLMs as game players. A second line evaluates foundation models as agents in games. SmartPlay [25], BALROG [15], GameBench [2], and GTBench [3] measure how well an agent *plays* through end-to-end metrics such as win rate and return; Orak [16] extends this to twelve commercial video games; and ING-VP [33] shows that even simple vision-based games defeat current VLMs. For card games specifically, PokerBench [36] scores best-action accuracy on static text poker scenarios, and follow-up work probes the game-theoretic reasoning behind LLM poker play with agentic tool use [13]; UNO Arena [18] evaluates sequential decision-making in a text-only UNO environment; AvalonBench [12] pits LLMs against rule-based bots in a social-deduction game; PokéChamp [8] reaches expert-level Pokémon battling by wrapping an LLM in minimax search; and Cardiverse [10] uses LLMs to *prototype* new card games rather than to evaluate model understanding. These efforts either collapse competence into a single end-to-end score or study games with established theory, solvers, and abundant pre-training exposure. We ask a complementary, prerequisite question: can a general model read a rulebook and a board and reason about them? Cards on the Table decomposes this into seven engine-verified skills, each posed in matched text and image form, on an original card pool that models cannot have memorized; because we build the game and engine ourselves, the engine acts as a deterministic ground-truth oracle, and the data, code, and evaluation can be released openly. Following the common protocol of the LLM game-playing literature [8, 12, 18], we retain full games against scripted bots as a downstream test, so per-skill accuracy and playing strength can be compared on the same game.

Multimodal benchmarks. Our visual setting also differs from static multimodal benchmarks such as MMMU [29] and MathVista [14], which evaluate perception and reasoning over fixed images: our board is a dynamic, partially observable state that changes every turn, and every question exists in an information-equivalent text form, so the cost of perceiving the board from pixels can be isolated from the reasoning task. To our knowledge, this is the first benchmark to test VLMs on reading a CCG board under hidden information.

Memory-augmented agents. Since card-game skill grows with knowledge of the cards seen, our work relates to agents that learn across episodes. Early lifelong agents accumulate skill libraries, verbal reflections, and episodic streams [17, 21, 22]. A recent wave builds structured agent memory: Memp [5] distills past trajectories into reusable procedural memory; Meta-Policy Reflexion [24] consolidates reflections into reusable rules with hard admissibility checks for illegal actions; coarse-to-fine grounded memory [28] organizes experience at multiple granularities for planning; A-MEM [26] stores interlinked memory notes in the Zettelkasten style; ACE [34] evolves the context itself as an accumulating playbook; Memory-R1 [27] learns memory add/update/delete decisions with reinforcement learning; and MIRIX [23] and G-Memory [32] propose multi-type and hierarchical memory systems. None of these target a growing catalog of cards and their interactions, nor measure whether such memory improves play; Cards on the Table is built to be exactly that measurement instrument (Sec. 6).

3. Aurora's Call

Cards on the Table is built on *Aurora's Call*, a two-player CCG we designed to be simple enough to specify exactly, yet rich enough to demand real strategy. We summarize the rules needed to follow the benchmark; the full ruleset is packed with the code.

Objective and components. Each player brings a 40-card deck; the top four cards are set aside face down as that player's *Fate*. An unblocked attack forces the defender to claim one of their own Fate cards (the claimed card joins their hand, a built-in comeback mechanic), and a player

loses upon claiming their fourth Fate, or upon having to draw from an empty deck. The card pool contains 71 unique cards across four types: *Creatures*, *Tactics* (one-time effects), *Items* (attachments), and *Sites* (persistent effects), and four factions (*Ember*, *Tide*, *Grove*, *Dusk*), one starter deck per faction. Playing any card costs *Aura*, produced by exhausting *Sources*. There are no dedicated Source cards, but once per turn, any hand card may be committed as a Source, creating a persistent tension between resources and playables.

Turns, combat, and keywords. On a turn a player may play one card as a Source, gather Aura, deploy cards, and attack. A new Creature has *Awakening Delay* and cannot attack the turn it arrives, though it may defend. An attacker either attacks the opponent directly, who may block with one ready Creature, or attacks an exhausted enemy Creature. An unblocked direct attack forces the opponent to claim a Fate. Keywords bend these rules: *Cleave* claims Fate through a block when Attack exceeds the blocker's Defense, *Eager* removes Awakening Delay, *Tideguard* lets a Creature block while exhausted, *Frenzy* scales Attack with board width, and *Siphon* destroys the opponent's Sources; further effects grant draws, bounce or freeze Creatures, block Fate claims (*Fog*), or revive discards. The property that matters for our benchmark is that these effects *take precedence over the base rules*: the legality and value of a move depend on the specific cards in play, not the rulebook alone. The four factions form a deliberate balance, including swarm aggro, control, ramp, and tempo archetypes, so no single strategy dominates and matchup knowledge matters.

Engine and instrumentation. Aurora's Call runs on a deterministic Python engine that is the single source of truth for the rules. For any state it enumerates the legal actions $\mathcal{L}(s)$, applies an action via $\mathcal{T}$ to produce the next state, and logs every interaction as a structured replay in a database; a browser front end renders any recorded state to pixels. The same engine powers a roster of scripted opponents (random, rule-based, greedy one-step, and a Monte Carlo tree search bot) and LLM/VLM agents that select among the enumerated legal moves, so games can be generated at scale. This instrumentation is what makes automatic, ground-truthed benchmark generation possible (Sec. 4): every answer is computed by simulating the game rather than annotated by hand, and every state exports simultaneously as text and as an image. Because the card pool is original, it is unlikely to appear in any model's pre-training data, reducing contamination relative to benchmarks built on commercial games.

4. The Cards on the Table Benchmark

Cards on the Table turns recorded games of Aurora's Call into engine-verified questions, each asked in two matched modalities. We first formalize the setting (Sec. 4.1), then describe the generation pipeline (Sec. 4.2), the input representation (Sec. 4.3), the seven categories (Sec. 4.4), and the scoring protocol (Sec. 4.5).

4.1. Formalization

The engine is a deterministic transition system. Let $s \in \mathcal{S}$ be a game state, $a \in \mathcal{A}$ an action, $\mathcal{T} : \mathcal{S} \times \mathcal{A} \to \mathcal{S}$ the transition that applies an action, and $\mathcal{L}(s) \subseteq \mathcal{A}$ the set of legal actions in s. Crucially, $\mathcal{L}$ and $\mathcal{T}$ already encode the precedence of card effects over the base rules, so they are the authoritative oracle for legality and dynamics.

A benchmark item is a tuple

$$x = \big(s,\, h,\, q,\, y,\, \tau,\, m\big), \tag{1}$$

where h is the replay history visible to the active player, q a question, y its ground-truth answer, τ the answer type, and $m \in \{\text{TXT}, \text{IMG}\}$ the modality. Every item exists in both modalities with identical (s, q, y), differing only in how s is presented. Ground truth is computed from the oracle, not annotated. For a legality question about action a,

$$y \;=\; \mathbf{1}\big[\, a \in \mathcal{L}(s) \,\big]; \tag{2}$$

for a state-change question, $y = \phi\big(\mathcal{T}(s, a)\big)$ for a readout ϕ (*e.g.* the resulting hand size). For replay questions the relevant action is the one that *produced* the shown state from its predecessor s^-; we recover it by matching transitions,

$$a^\star \;=\; \big\{\, a \in \mathcal{L}(s^-) \;:\; \sigma\big(\mathcal{T}(s^-, a)\big) = \sigma(s) \,\big\}, \tag{3}$$

where $\sigma(\cdot)$ is a canonical state signature, and draw distractors from $\mathcal{L}(s^-) \setminus \{a^\star\}$.

4.2. Generation pipeline

Each item is produced automatically. **(1) Play.** We collect games from humans (browser client), scripted self-play bots (random, rule-based, greedy, and MCTS), and LLM/VLM agents versus bots, logging every interaction as a replay. **(2) Sample.** From each recorded game we sample non-trivial decision states (skipping mid-resolution states) from either player's perspective, and record the history h visible up to that point. **(3) Label.** For each state and applicable template the engine computes y by reading s, querying $\mathcal{L}(s)$, or simulating a move via $\mathcal{T}$ (Sec. 4.1). **(4) Render.** The state is exported as text and as rendered board images, and the item is stored with its answer, type, and provenance. Because y comes from the engine, the dataset scales to thousands of items with no annotation cost or noise. The released dataset samples five decision states from each of 20 recorded games (100 states), yielding 3,934 items per modality.

4.3. Dual-modality input representation

A model receives a fixed prompt: **(i)** a system instruction that fixes the task and constrains the output to a JSON object with keys `answer` and `explanation`; **(ii)** the **ruleset**, a compact plain-text specification of the rules and keywords; **(iii)** the **state and history**: in the TXT setting, a human-readable interaction log through the sampled event, including every card the active player has legitimately seen; in the IMG setting, the same log through the previous turn plus rendered board images for the current turn's events, the last image showing the sampled state; and **(iv)** the **question**. Replay-reasoning items instead present the previous and current decision states (as text, or as a before/after image pair). The short `answer` is graded exactly, while `explanation` is kept for analysis. Because the two modalities share (s, q, y), any accuracy gap between them measures the added cost of perceiving the board from pixels rather than a change in the reasoning task.

4.4. Question categories

The seven categories (Tab. 1) span the abilities of Sec. 1, from perception to dynamics.

OCR (perception). Reading the board: listing cards in hand or on the field, counting Sources, decks, and discards, reading a card's Attack/Defense and effect text, and reporting flags such as exhaustion or Awakening Delay. It is the category most affected by the visual modality and serves as a perception baseline for the others.

Move legality (rules). Whether a specific action, for example playing a card, playing it as a Source, gathering Aura, or attacking with or into a Creature, is currently legal. Ground truth is $\mathbf{1}[a \in \mathcal{L}(s)]$, so legality reflects both the base rules and the active card effects.

Illegal-action diagnosis. Beyond yes/no: the model must name *why* an action is legal or illegal, choosing among four options drawn from nine causes (*e.g.* turn ownership, insufficient Aura, exhaustion, Awakening Delay, field limit, target legality). This probes understanding of the rules' structure, including off-turn proposals that violate turn ownership.

Public-card tracking (memory). Memory and hidden information. From only the visible history h, the model decides whether a named card has appeared in a public zone (hand, board, discard, or claimed Fate) and counts hidden quantities such as face-down deck cards, combined discards, and the opponent's hand size. Correct answers require integrating information across turns, not reading the current board alone.

Resource reasoning. Counterfactual Aura arithmetic: how much Aura gathering all ready Sources yields, the resulting total, how many hand cards become affordable, and how much Aura remains after a purchase. This isolates quantitative planning from rule knowledge.

State-change prediction (dynamics). Predicting the effect of *taking* an action, like describing the hand size or Aura afterward, a card's destination zone, whether an attacked Creature is defeated, who survives a defended attack, and whether it forces a Fate claim. With ground truth $y = \phi(\mathcal{T}(s, a))$, this directly measures a model's forward model of the game, including card effects that override the base rules.

Replay reasoning (backward inference). Given a state, which single action produced it, as a four-way choice. The target is recovered by Eq. (3), yielding an unambiguous label and testing causal, backward reasoning over dynamics.

4.5. Metrics

Each item is scored correct or incorrect by its answer type and we report accuracy overall, by modality, and by category. Yes/no, multiple-choice, and categorical items use normalized exact match; numeric items require the exact integer; *name* items the normalized card name; and *list* items the predicted set of names to equal the ground-truth set. Writing $\hat{y}$ for the parsed prediction and norm_τ for type-specific normalization,

$$\mathrm{Acc} \;=\; \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \mathbf{1}\big[\,\mathrm{norm}_\tau(\hat{y}) = \mathrm{norm}_\tau(y)\,\big]\,. \qquad (4)$$

Free-text items (*e.g.* reading a card's effect text) are released for analysis but excluded from accuracy, leaving 3,800 scored items per modality. The matched-modality design lets us report a perception gap,

$$\Delta \;=\; \mathrm{Acc}_{\mathrm{TXT}} - \mathrm{Acc}_{\mathrm{IMG}}, \qquad (5)$$

which attributes the difference between the LLM and VLM settings to board perception while holding the reasoning task fixed. Finally, game-level competence is measured by *win rate* in full games against the scripted bots of Sec. 3 (Sec. 5.3).

4.6. Dataset statistics

The released benchmark comprises 7,868 items (3,934 per modality, 3,800 of them scored) drawn from 100 decision states across 20 recorded games, instantiated from 54 question templates; Tab. 2 reports per-category counts.

Category	Competency	Example question	Answer type	Ground truth
OCR	Perception	"List the cards visible in the player's hand." / "What is *X*'s visible Attack?"	list, numeric, yes/no	rendered state
Move legality	Rules	"Can the active player play *X*?" / "Can *X* attack *Y*?"	yes/no	legal-action set
Illegal action	Rule diagnosis	"Is this proposed action legal? Choose one: A) legal … D) Awakening Delay."	multiple choice (4 of 9)	engine
Public-card tracking	Memory / hidden info	"Has *X* appeared in a zone visible to the active player?" / "How many cards are face-down?"	yes/no, numeric	replay history
Resource reasoning	Counterfactual arithmetic	"If the active player gathers all ready Sources, how much total Aura will they have?"	numeric, yes/no	engine
State change	Dynamics (forward)	"If the active player plays *X*, which zone does it move to?" / "Will *Y* be defeated?"	numeric, categorical, yes/no, MC	engine simulation
Replay reasoning	Dynamics (backward)	"Which single action did the active player take to reach the shown state? A–D."	multiple choice (4)	signature matching

Table 1. **The seven categories of Cards on the Table.** Each category isolates a distinct competency, from reading the board (OCR) to inferring dynamics forward (state change) and backward (replay reasoning). Every question is generated automatically from a recorded game and is posed in both a text (LLM) and an image (VLM) form with identical ground truth.

Category	Tmpl.	Items	Scored
OCR	23	800	666
Move legality	5	769	769
Illegal action	5	713	713
Public tracking	4	600	600
Resource reasoning	5	545	545
State change	11	413	413
Replay reasoning	1	94	94
Total	54	3,934	3,800

Table 2. **Dataset composition.** Question templates per category (fixed by the generators) and generated items per modality; every item exists in both an LLM (text) and a VLM (image) form. *Scored* excludes free-text items (card effect text readout), which are released for analysis but not graded by exact match.

5. Experiments

5.1. Setup

We evaluate recent open models in both settings, zero-shot and without task-specific fine-tuning. The TXT (LLM) setting reads the text state and history; we evaluate Qwen2.5-32B-Instruct [19] and Llama-3.1-8B-Instruct [6]. The IMG (VLM) setting reads the rendered board; we evaluate Qwen2.5-VL-32B-Instruct [20], InternVL3-38B [35], and LLaVA-OneVision-7B [9]. All models are served locally with vLLM and queried with the prompt of Sec. 4.3, decoding greedily and parsing the JSON `answer`; unparseable responses count as incorrect. Every model sees the same items with the same ground truth, and the TXT and IMG items are matched, so models and modalities are directly comparable.

5.2. Question-level results

Tab. 3 reports accuracy by model, modality, and category. The best model, InternVL3-38B, answers 57.5% of items correctly; no model exceeds 58% overall, against a 25.8% random-guess baseline. Three patterns stand out. **(1) Perception is not the bottleneck.** VLMs read the rendered board well, InternVL3 reaches 83.3% OCR accuracy, and the two strongest models overall are VLMs. Within the Qwen2.5 family the matched-modality gap Δ (Eq. (5)) is -4.5 points overall: images *help* on perception-adjacent categories (OCR -15.0, state change -24.6) while text helps on rule application (legality $+16.4$) and replay reasoning ($+10.7$). **(2) Rule application, not rule recall, fails.** Despite receiving the full ruleset in every prompt, no VLM judges move legality better than the 50% coin-flip baseline (39.9–49.0%), and the best text model reaches only 58.4%; diagnosing *why* an action is illegal is above chance but far from solved. Turn-ownership violations are a systematic blind spot: on off-turn proposals models frequently declare the opponent's play legal. **(3) Counterfactual arithmetic**

Model	Mod.	Per-category accuracy (%)							Avg
		OCR	Legal	Illeg.	Track	Rsrc	State	Replay	
Qwen2.5-32B [19]	TXT	60.5	**58.4**	47.8	50.3	33.2	44.6	**79.8**	50.9
Llama-3.1-8B [6]	TXT	51.7	51.2	42.4	47.0	32.3	31.7	62.8	44.4
Qwen2.5-VL-32B [20]	IMG	75.5	42.0	52.2	57.8	**38.3**	**69.2**	69.1	55.4
InternVL3-38B [35]	IMG	**83.3**	39.9	**63.3**	**63.7**	29.9	67.1	54.3	**57.5**
LLaVA-OV-7B [9]	IMG	53.6	49.0	32.6	41.2	27.7	27.9	15.3	39.3
Random guess	–	5.0	50.0	25.0	33.1	22.8	8.8	25.0	25.8
Human	–	–	–	–	–	–	–	–	–

Table 3. **Main results.** Accuracy of LLMs (TXT, text state and history) and VLMs (IMG, rendered board images) on the seven categories over the 3,800 scored items per modality. *Avg* is overall accuracy across all scored items. *Random guess* is the expected accuracy of a uniform random choice among valid options (zero for numeric, name, and list answers). LLaVA-OV answered 3,476 of 3,800 items due to server timeouts; its accuracy is over answered items. Human reference is pending.

Model	Random	Rule	Greedy	MCTS	Avg
With game history in prompt					
Qwen2.5-32B	83.3	10.4	16.7	4.2	28.7
Llama-3.1-8B	29.2	4.2	8.3	0.0	10.4
Qwen2.5-VL-32B	91.5	10.9	15.9	8.7	31.8
InternVL3-38B	85.4	14.6	0.0	8.5	29.2
LLaVA-OV-7B	91.7	15.2	27.1	19.6	38.4
Without game history (current state only)					
Qwen2.5-32B	87.2	21.3	25.5	10.6	36.2
Llama-3.1-70B	84.1	12.5	10.6	4.3	27.9
Qwen2.5-VL-32B	84.4	21.3	17.0	12.5	33.8
InternVL3-38B	89.6	33.3	21.7	17.0	40.4
LLaVA-OV-7B	87.5	10.4	16.7	4.2	29.7
Human	100	100	100	100	100

Table 4. **Full-game win rate (%) against scripted bots**, 48 games per matchup. *Avg* is the mean over the Random, Rule-based, Greedy, and MCTS columns. Every model beats the random bot but loses the large majority of games to simple heuristics that a human defeats every time.

is hardest. Resource reasoning is the worst category for every model (27.7–38.3%, barely above its 22.8% baseline), showing that multi-step Aura arithmetic over the board state remains out of reach even when the state is read correctly.

5.3. Full-game play

We complement question-level accuracy with full games: at each decision point the model receives the ruleset, the current state (text or rendered images), and the engine-enumerated legal moves, rates every legal move, and the highest-rated move is played. Each model plays 48 games per scripted opponent, in two prompt variants: with the visible game history, and with only the current state. Tab. 4 reports win rates. All models comfortably beat the random bot (83–92%, except Llama-3.1-8B) yet lose the large majority of games to the simple rule-based checklist, the greedy one-step bot, and the MCTS bot, all opponents a human player defeats 100% of the time. Providing the full game history does not reliably help and often hurts (*e.g.* InternVL3 vs. rule-based: 14.6% with history, 33.3% without), suggesting current models fail to exploit and are sometimes distracted by exactly the accumulated context that skilled play requires. Together with Tab. 3, this locates the failure: models can partially read the board but cannot yet turn that reading into rule-sound, resource-aware decisions.

5.4. Analysis

The two evaluations answer the questions the benchmark was built for:

- **Perception vs. reasoning.** The matched-modality gap is small and often negative; failures persist in the TXT setting, so they are reasoning failures, not reading failures.
- **Category difficulty.** Resource arithmetic and move legality are hardest relative to their baselines; OCR and replay reasoning are easiest. The forward model (state change) splits by modality: seeing the board helps predict its next state.
- **Rules vs. knowledge.** Models that recite legality rules still misjudge specific, card-dependent cases, and win-rate collapses against any opponent that plays on-curve, consistent with the thesis that competence requires knowing the cards, motivating Sec. 6.
- **Human reference.** A small human study on the question categories is planned to contextualize per-category accuracy; on full games, a human defeats every scripted bot.

6. Toward a Memory-Augmented Agent

Our benchmark is designed to expose one weakness in particular: a model may recite the rules yet misjudge moves because it does not know *which cards exist* or how they interact. Sec. 5.3 shows exactly this, as models that beat a random opponent collapse against any opponent that plays on-curve, and handing them the raw game history does not help. Humans overcome this by learning across games: each match

reveals new cards, effects, and combinations that reshape later decisions. We therefore frame a **memory-augmented agent** as the natural method counterpart to the benchmark, and describe the design we are building.

Hypothesis. The central claim behind Cards on the Table is that, in a CCG, move quality is not a function of the current board and rulebook alone, but of accumulated knowledge K of the card pool:

$$\text{value}(a \mid s) \;=\; g\big(s,\, a,\, K\big), \tag{6}$$

where K grows with experience. A memoryless model approximates $g(s, a, \varnothing)$; the move that looks best under $\varnothing$ can be a mistake under a fuller K. The state-change category and the full-game evaluation are constructed precisely to be sensitive to this gap, and Tab. 4 suggests that *raw* history is not the right K: what is needed is distilled, reusable knowledge.

Memory store. The agent maintains an external memory $\mathcal{M}$ of *card-knowledge entries*. After each game it writes, for every card it observed, a structured record, consiting of name, type, faction, cost, statistics, effect text, the keyword interactions it witnessed (*e.g.* that *Cleave* claimed Fate through a block), and lightweight outcome statistics such as whether playing the card preceded a Fate swing. Entries are keyed by card, linked to related entries, and accumulate evidence across games, turning fleeting in-game observations into persistent, reusable knowledge. This design draws on the skill libraries and episodic memories of prior lifelong agents [17, 21, 22] and on recent structured agent memory: procedural distillation of trajectories into strategy procedures [5], linked memory notes [26], and learned add/update/delete management to keep the store compact [27], specialized to the card knowledge that Aurora's Call makes first-class, with hard rule-admissibility retained by the engine's legal-move oracle rather than learned memory [24].

Retrieval and move re-ranking. At a decision point with state s and legal moves $\mathcal{L}(s)$, the agent (i) perceives s, (ii) for each candidate a retrieves the entries $\mathcal{M}_a \subseteq \mathcal{M}$ relevant to the cards a involves and to the cards currently in play, and (iii) conditions on the retrieved knowledge to score and rank candidates,

$$a^{\star} \;=\; \arg\max_{a \in \mathcal{L}(s)} \; g_\theta\big(s,\, a,\, \mathcal{M}_a\big), \tag{7}$$

where g_θ is the LLM/VLM acting as a scorer prompted with the retrieved entries. Retrieval keeps the prompt compact even as $\mathcal{M}$ grows, so the agent can draw on knowledge from many past games without exceeding the context window, in contrast to the with-history prompts of Sec. 5.3, which grow with the game and dilute the relevant facts. Conceptually, Eq. (7) replaces the memoryless estimate $g(s, a, \varnothing)$ of a plain prompted model with an estimate informed by what the agent has learned the cards can do.

What the benchmark measures. Cards on the Table is the measurement instrument for this agent. We will compare the memory-augmented agent against the memoryless baselines of Sec. 5 on (i) *state-change* accuracy: testing if remembering card effects improve prediction of a move's outcome, (ii) *full-game win rate* against the fixed bot roster of Tab. 4 examining accumulated knowledge convert board reading into wins, and (iii) question-level accuracy on categories that depend on card knowledge. A positive result would substantiate the paper's thesis: that knowing the cards, not just the rules, is what separates competent from incompetent play, and that this knowledge can be *accumulated* rather than assumed.

The memory-augmented agent is under construction; we present its design and the evaluation plan here, and report its empirical results in the camera-ready version.

7. Conclusion

We presented **Cards on the Table**, a benchmark that evaluates language and vision-language models on seven engine-verified competencies of card-game play, including perception, rule legality, illegal-action diagnosis, public-card memory, resource arithmetic, state-change prediction, and replay reasoning, together with full-game play, over *Aurora's Call*, a purpose-built game whose card effects override the base rules. By generating every item from recorded games and posing it in matched text and image forms, the benchmark separates perception from reasoning and rule-following from card knowledge. Across five open models the results pinpoint where current models fail: boards are read adequately, but legality judgments hover at chance, counterfactual resource arithmetic collapses, and no model comes close to beating simple scripted heuristics at the full game. The benchmark also defines the measurement instrument for a memory-augmented agent that accumulates card knowledge across games, closing the loop between the paper's thesis and a method that acts on it.

Limitations. The benchmark currently covers a single game; while its categories are game-agnostic in spirit, extending to other titles would strengthen external validity. The visual modality uses rendered boards rather than photographs, which simplifies perception relative to physical play. The evaluated models are open-weight models up to 38B parameters; closed-source frontier models and a human study are planned. Finally, the memory-augmented

agent is ongoing: its empirical evaluation will appear in the camera-ready version.

Outlook. Cards on the Table makes the gap between rule-following and knowledge-aware play measurable. We hope it supports agents that, like human players, *improve after every game*, and we will release the game, engine, generation pipeline, and benchmark to that end.

References

[1] Noam Brown and Tuomas Sandholm. Superhuman AI for multiplayer poker. *Science*, 365(6456):885–890, 2019. 2

[2] Anthony Costarelli, Mat Allen, Roman Hauksson, Grace Sodunke, Suhas Hariharan, Carlson Cheng, Wenjie Li, Joshua Clymer, and Arjun Yadav. GameBench: Evaluating strategic reasoning abilities of LLM agents. *arXiv preprint arXiv:2406.06613*, 2024. 2

[3] Jinhao Duan, Renming Zhang, James Diffenderfer, Bhavya Kailkhura, Lichao Sun, Elias Stengel-Eskin, Mohit Bansal, Tianlong Chen, and Kaidi Xu. GTBench: Uncovering the strategic reasoning limitations of LLMs via game-theoretic evaluations. In *NeurIPS*, 2024. 1, 2

[4] Ronaldo e Silva Vieira, Anderson Rocha Tavares, and Luiz Chaimowicz. *A Taxonomy of Collectible Card Games from a Game-Playing AI Perspective*, page 163–177. Springer Nature Switzerland, 2024. 1, 2

[5] Runnan Fang, Yuan Liang, Xiaobin Wang, Jialong Wu, Shuofei Qiao, Pengjun Xie, Fei Huang, Huajun Chen, and Ningyu Zhang. Memp: Exploring agent procedural memory, 2026. 2, 7

[6] Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024. 5, 6

[7] Amy K. Hoover, Julian Togelius, Scott Lee, and Fernando de Mesentier Silva. The many AI challenges of Hearthstone. *KI - Künstliche Intelligenz*, 34(1):33–43, 2020. 1, 2

[8] Seth Karten, Andy Luu Nguyen, and Chi Jin. Pokéchamp: an expert-level minimax language agent, 2025. 2

[9] Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. Llava-onevision: Easy visual task transfer, 2024. 5, 6

[10] Danrui Li, Sen Zhang, Sam S. Sohn, Kaidong Hu, Muhammad Usman, and Mubbasir Kapadia. Cardiverse: Harnessing llms for novel card game prototyping, 2025. 2

[11] Junjie Li, Sotetsu Koyamada, Qiwei Ye, Guoqing Liu, Chao Wang, Ruihan Yang, Li Zhao, Tao Qin, Tie-Yan Liu, and Hsiao-Wuen Hon. Suphx: Mastering mahjong with deep reinforcement learning. *arXiv preprint arXiv:2003.13590*, 2020. 2

[12] Jonathan Light, Min Cai, Sheng Shen, and Ziniu Hu. Avalonbench: Evaluating llms playing the game of avalon, 2023. 2

[13] Minhua Lin, Enyan Dai, Hui Liu, Xianfeng Tang, Yuliang Yan, Zhenwei Dai, Jingying Zeng, Zhiwei Zhang, Fali Wang, Hongcheng Gao, Chen Luo, Xiang Zhang, Qi He, and Suhang Wang. How far are llms from professional poker players? revisiting game-theoretic reasoning with agentic tool use, 2026. 2

[14] Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chunyuan Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. MathVista: Evaluating mathematical reasoning of foundation models in visual contexts. In *ICLR*, 2024. 2

[15] Davide Paglieri, Bartłomiej Cupiał, Sam Coward, Ulyana Piterbarg, Maciej Wołczyk, Akbir Khan, Eduardo Pignatelli, Łukasz Kuciński, Lerrel Pinto, Rob Fergus, Jakob N. Foerster, Jack Parker-Holder, and Tim Rocktäschel. BALROG: Benchmarking agentic LLM and VLM reasoning on games. *arXiv preprint arXiv:2411.13543*, 2024. 1, 2

[16] Dongmin Park, Minkyu Kim, Beongjun Choi, Junhyuck Kim, Keon Lee, Jonghyun Lee, Inkyu Park, Byeong-Uk Lee, Jaeyoung Hwang, Jaewoo Ahn, Ameya S. Mahabaleshwarkar, Bilal Kartal, Pritam Biswas, Yoshi Suhara, Kangwook Lee, and Jaewoong Cho. Orak: A foundational benchmark for training and evaluating llm agents on diverse video games, 2026. 1, 2

[17] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *ACM Symposium on User Interface Software and Technology (UIST)*, 2023. 2, 7

[18] Zhanyue Qin, Haochuan Wang, Deyuan Liu, Ziyang Song, Cunhang Fan, Zhao Lv, Jinlin Wu, Zhen Lei, Zhiying Tu, Dianhui Chu, Xiaoyan Yu, and Dianbo Sui. Uno arena for evaluating sequential decision-making capability of large language models, 2024. 2

[19] Qwen Team. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 5, 6

[20] Qwen Team. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. 5, 6

[21] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. In *NeurIPS*, 2023. 1, 2, 7

[22] Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. Voyager: An open-ended embodied agent with large language models. *Transactions on Machine Learning Research (TMLR)*, 2024. 1, 2, 7

[23] Yu Wang and Xi Chen. Mirix: Multi-agent memory system for llm-based agents, 2025. 2

[24] Chunlong Wu, Ye Luo, Zhibo Qu, and Min Wang. Meta-policy reflexion: Reusable reflective memory and rule admissibility for resource-efficient llm agent, 2025. 2, 7

[25] Yue Wu, Xuan Tang, Tom M. Mitchell, and Yuanzhi Li. SmartPlay: A benchmark for LLMs as intelligent agents. In *ICLR*, 2024. 1, 2

[26] Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. A-mem: Agentic memory for llm agents, 2025. 2, 7

[27] Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian

Kersting, Jeff Z. Pan, Hinrich Schütze, Volker Tresp, and Yunpu Ma. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*, 2025. 2, 7

[28] Wei Yang, Jinwei Xiao, Hongming Zhang, Qingyang Zhang, Yanna Wang, and Bo Xu. Coarse-to-fine grounded memory for llm agent planning, 2026. 2

[29] Xiang Yue, Yuansheng Ni, Kai Zhang, Tianyu Zheng, Ruoqi Liu, Ge Zhang, Samuel Stevens, Dongfu Jiang, Weiming Ren, Yuxuan Sun, et al. MMMU: A massive multi-discipline multimodal understanding and reasoning benchmark for expert AGI. In *CVPR*, 2024. 2

[30] Daochen Zha, Kwei-Herng Lai, Yuanpu Cao, Songyi Huang, Ruzhe Wei, Junyu Guo, and Xia Hu. RLCard: A toolkit for reinforcement learning in card games. *arXiv preprint arXiv:1910.04376*, 2019. 1, 2

[31] Daochen Zha, Jingru Xie, Wenye Ma, Sheng Zhang, Xiangru Lian, Xia Hu, and Ji Liu. DouZero: Mastering DouDizhu with self-play deep reinforcement learning. In *International Conference on Machine Learning (ICML)*, 2021. 2

[32] Guibin Zhang, Muxin Fu, Guancheng Wan, Miao Yu, Kun Wang, and Shuicheng Yan. G-Memory: Tracing hierarchical memory for multi-agent systems. *arXiv preprint arXiv:2506.07398*, 2025. 2

[33] Haoran Zhang, Hangyu Guo, Shuyue Guo, Meng Cao, Wenhao Huang, Jiaheng Liu, and Ge Zhang. ING-VP: MLLMs cannot play easy vision-based games yet. *arXiv preprint arXiv:2410.06555*, 2024. 1, 2

[34] Qizheng Zhang, Changran Hu, Shubhangi Upasani, Boyuan Ma, Fenglu Hong, Vamsidhar Kamanuru, Jay Rainton, Chen Wu, Mengmeng Ji, Hanchen Li, Urmish Thakker, James Zou, and Kunle Olukotun. Agentic context engineering: Evolving contexts for self-improving language models, 2026. 2

[35] Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025. 5, 6

[36] Richard Zhuang, Akshat Gupta, Richard Yang, Aniket Rahane, Zhengyu Li, and Gopala Anumanchipalli. Pokerbench: Training large language models to become professional poker players, 2025. 2