The Agzamov Test:
A Benchmark Proposal for Measuring Augmented AI Capabilities Under Adversarial Conditions

Ali Agzamov[1]

[1]BrainOps Limited, Queenstown, New Zealand

February 2026

Abstract

Every major AI benchmark is an exam. A question is asked, an answer is given, a score is assigned. The model is tested naked—no tools, no memory, no context. In laboratory conditions—fixed questions with known answers. On tasks that can be memorized—training data contamination is an unsolved problem.

None of this reflects how AI is actually used. In production, models operate with tools, memory, and orchestration layers. They face tasks that cannot be memorized. They work against adversaries who adapt.

The Agzamov Test is designed to address a gap in current evaluation: how AI models perform in a real adversarial environment, at every level of augmentation. Two agents play repeated games in two environments—Chess960 (complete information) and poker (incomplete information)—across four augmentation levels—memory, tools, retrieval (RAG), and full orchestration—against a naked baseline. The test produces a single headline metric, the Agzamov Score, with breakdown by environment and augmentation level.

Chess960 minimizes the value of memorized openings. Poker introduces incomplete information. Every position is effectively unique. Every opponent adapts. "Smart model" is not a claim—it is a number.

Status: This paper presents the benchmark design, theoretical motivation, and infrastructure validation (Phase 0, chess environment). The poker environment is designed but not yet validated. Phases 1–3 are in progress. Preliminary results demonstrating non-zero Δ_a will be reported in subsequent work.

Keywords: AI benchmarks, augmented AI, tool use, memory systems, game theory, adversarial evaluation, Chess960, poker

1 The Problem with Current Benchmarks

1.1 They Test Naked Models

All major benchmarks (MMLU [Hendrycks et al., 2021], HumanEval [Chen et al., 2021], HELM [Liang et al., 2023], ARC-AGI-2) evaluate a model in isolation. No tools, no memory, no persistent context. But nobody uses a naked model. Every production deployment includes retrieval, tool access, memory, and orchestration. The gap between "how models are tested" and "how models are used" is total.

There is no way to measure how much value augmentation adds. A company builds a memory system—how does it prove the system works? A lab releases a new model—how does it show the model uses tools better than the previous version? Today the answer is: marketing claims. There is no independent, reproducible measurement.

1.2 They Use Laboratory Conditions

Benchmarks present fixed questions with known answers. Models can memorize them. Training data contamination is documented across MMLU, HumanEval, and GSM8K. Even the ARC-AGI family—the most contamination-resistant benchmarks available—yields to brute-force program synthesis: on ARC-AGI v1, sampling ∼8,000 candidate programs per task with GPT-4o reached 50% [Greenblatt, 2024]; on ARC-AGI-2, test-time training of small models reached 28% [Sorokin and Puget, 2025]. Both versions share the same task structure—finite, example-verifiable puzzles amenable to search.

Real tasks do not have answer keys. Real opponents adapt. Real environments change.

1.3 They Measure Theory, Not Practice

A model scores 90% on MMLU. What does this mean in practice? Nothing. It means the model is good at answering multiple-choice questions from textbooks. It says nothing about whether the model can solve a real problem where the correct answer is unknown, the environment is adversarial, and yesterday's strategy may be obsolete today.

"Smart model" should not be a press release. It should be a verifiable number measuring what the model can actually do.

1.4 The ARC-AGI-2 Illusion

ARC-AGI-2 [Chollet, 2025] is the most credible reasoning benchmark available—used on model cards by all four major labs (OpenAI, Google, Anthropic, xAI). It presents novel visual-logic grid puzzles that cannot be solved by memorization. Humans average 60%; base LLMs without augmentation score near 0%.

In February 2026, Google announced Gemini 3.1 Pro scored 77.1% [Google DeepMind, 2026b,a]. Headlines declared a "reasoning breakthrough." But examine *how* these scores are achieved:

Program synthesis, not reasoning. The top solutions generate thousands of candidate programs per task and verify which one matches the examples. ARC Prize's own analysis confirms accuracy scales log-linearly with compute—the signature of brute-force search [ARC Prize, 2025]. The benchmark's creators acknowledge: "Current AI reasoning performance is tied to model knowledge… Human reasoning capability is not bound to knowledge."

What ARC-AGI-2 actually measures: the ability to search a finite, verifiable space of grid transformations. Given 2–4 examples, generate candidate rules, test against examples, output the one that fits.

What it does not measure: performance in open-ended adversarial environments where candidate solutions cannot be verified against examples, the environment changes, and the opponent adapts.

Table 1: Task types and LLM capability landscape.

Task Type	Example	LLM Capability	Why
Knowledge recall	MMLU, GPQA	Strong	Pattern matching on training d
Pattern completion	ARC-AGI-2	Partial (via search)	Finite verifiable search space
Positional evaluation	Chess: "who is winning?"	Partial	Pattern matching on comment
Adversarial adaptation	**Repeated games**	**Unmeasured**	**No existing benchmark**

The Agzamov Test is designed to fill this gap.

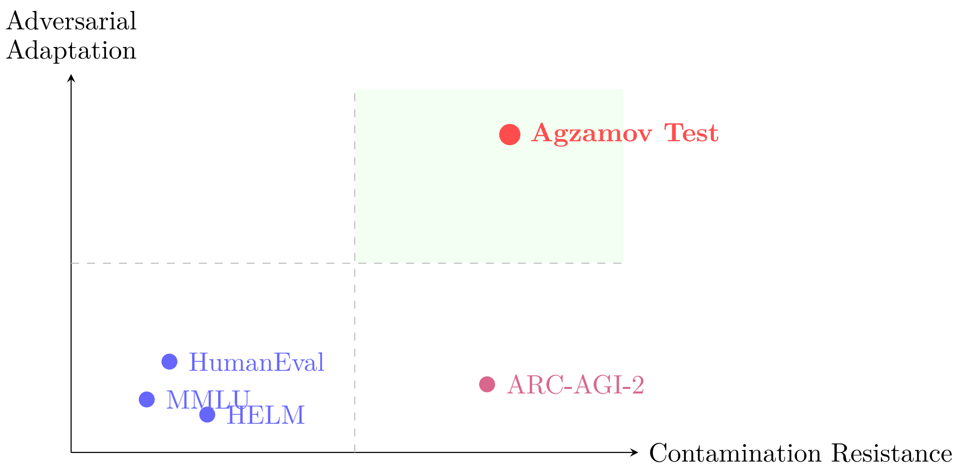

Figure 1: Benchmark landscape. Existing benchmarks cluster in the low-adaptation region. ARC-AGI-2 resists contamination but tests static puzzles, not adversarial adaptation. The Agzamov Test occupies the high-resistance, high-adaptation quadrant.

2 What the Agzamov Test Measures

The Agzamov Test answers one question: **how does augmentation change what an AI model can actually do?**

2.1 Simple Explanation

Imagine two chess players. Both equally skilled. But one forgets everything after each game. The other remembers: how the opponent played, what mistakes they made, what positions they avoid.

Who wins after 100 games? The one with memory. Even if the forgetful one is slightly smarter.

Now give the second player a calculator that can check tactics. And a notebook of the opponent's previous games. And a coach whispering strategic advice.

The Agzamov Test measures exactly this—at each level of augmentation, in two different environments, producing one number: the Agzamov Score.

2.2 Augmentation Levels

Table 2: Augmentation levels in the Agzamov Test.

Level	**What the model has**	**What it tests**
Naked	Nothing. Raw model.	Baseline capability
+ Memory	Persistent memory across games	Adaptation, opponent modeling
+ Tools	External calculation (e.g., Stockfish)	Tool use effectiveness
+ RAG	Context injection from external data	Retrieval quality
+ Full Stack	Memory + Tools + RAG + Orchestration	Complete agent architecture

Table 3: Comparison of chess and poker as evaluation environments.

Property	**Chess960**	**Poker (NLHE)**
Information	Complete—both see full board	Incomplete—hidden cards
Randomness	None—deterministic	High—card distribution
Contamination risk	Eliminated—960 random starts	Low—no memorizable "answers"
Memory value	Pattern exploitation, preferences	Bet sizing tells, bluff profiling
Tool value	Tactical calculation (Stockfish)	GTO solvers
What it reveals	Performance under full information	Performance under uncertainty

2.3 Two Environments

A model that improves with augmentation in both environments demonstrates general capability. A model that improves in one but not the other reveals domain-specific limitations.

2.4 Why Games

Games are among the few domains that satisfy all requirements simultaneously:

1. **Real adversarial pressure.** The opponent adapts, punishes mistakes, and actively tries to make your strategy obsolete.
2. **Cannot be memorized.** Chess960 eliminates opening theory. Every position is unique. There is no answer key.
3. **Objective measurement.** Win/loss/draw. Big blinds per 100 hands. No subjective evaluation.
4. **Established baselines.** Decades of human and engine performance data for calibration, including superhuman AI systems in both chess [Silver et al., 2018] and poker [Brown and Sandholm, 2019].
5. **Repeatable at low cost.** Thousands of games can be run for dollars.
6. **Spectator-friendly.** A chessboard is universally understood. "With memory wins, without memory loses" requires no technical explanation.

3 The Agzamov Score

3.1 Headline Metric

The Agzamov Score is a single number (0–100) that captures a model's overall performance across all environments and augmentation levels:

$$A = 100 \cdot \sum_{i=1}^{7} w_i \, \hat{S}_i, \qquad \sum_i w_i = 1, \quad \hat{S}_i \in [0,1] \tag{1}$$

where each $\hat{S}_i$ is a normalized sub-score:

Sigmoid scaling constant c and clamp bounds $\Delta_{\max}$, $\tau_{\max}$ are calibrated from Phase 1–2 data. Weights for v0.1: uniform ($w_i = 1/7$). Final weights for v1.0 will be determined via leave-one-model-out cross-validation and published alongside the reference implementation. The score is deterministic given the same test data and weight vector.

3.2 Agzamov Delta (Δ_a)

The core sub-metric. Difference in performance between augmented and naked play:

$$\Delta_a = P(\text{model} + \text{augmentation}) - P(\text{model, naked}) \tag{2}$$

Table 4: A-Score sub-components and normalization.

i	Component	Raw metric	Normalization $\hat{S}_i$
1	Chess baseline	Win rate $\in [0,1]$	Identity
2	Chess augmented	Win rate $\in [0,1]$	Identity
3	Poker baseline	bb/100	Sigmoid: $\sigma(x/c)$
4	Poker augmented	bb/100	Sigmoid: $\sigma(x/c)$
5	Δ_a (chess)	Win-rate pp	Clamp $[0, \Delta_{\max}]$, rescale
6	Δ_a (poker)	bb/100	Clamp $[0, \Delta_{\max}]$, rescale
7	Convergence τ	Games to 95%	$1 - \tau/\tau_{\max}$ (faster = higher)

where P denotes the environment-specific performance metric: win rate for chess, bb/100 for poker. The theoretical basis for this quantity is developed in Section 6.

Higher delta = augmentation adds more value. Negative delta = augmentation is hurting performance (retrieval noise, tool misuse).

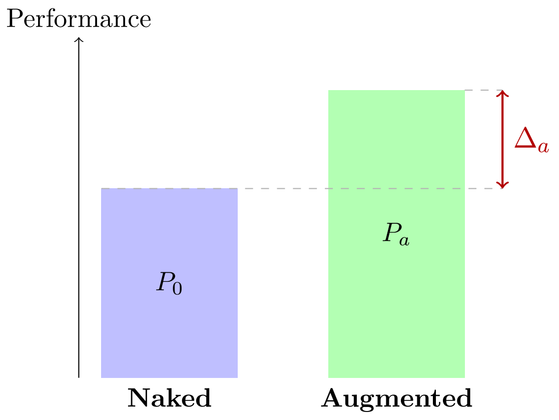

Figure 2: The Augmentation Delta (Δ_a): performance difference between augmented and naked play. $\Delta_a > 0$ indicates augmentation adds value; $\Delta_a < 0$ indicates augmentation degrades performance.

3.3 Convergence Rate (τ)

How quickly augmentation starts helping. Measured as the number of games/hands needed to reach 95% of maximum performance.

Two systems with identical Δ_a but different τ are fundamentally different: one is useful from day one, the other after months of data collection.

Recovery τ. When the opponent changes strategy, how quickly does the agent adapt? This measures resilience under adversarial shift.

3.4 Model × Augmentation Matrix

The full breakdown. Take N models and M augmentation configurations. Run all combinations in both environments:

Reading the matrix:

- **Rows** (fixed model, varying augmentation): what does each augmentation level add for this model?
- **Columns** (fixed augmentation, varying model): which model uses this augmentation best?

Table 5: Hypothetical model×augmentation matrix for Chess960 (win rate %).

	Naked	+ Memory	+ Tools	+ RAG	+ Full Stack
Claude	...	...	...	...	...
GPT	...	...	...	...	...
Gemini	...	...	...	...	...

- **Cross-environment**: consistent gains across both ⇒ general capability. Gains in one only ⇒ domain-specific limitation.
- **Critical insight**: if a weaker model + better augmentation beats a stronger model + worse augmentation, that proves infrastructure can compensate for model capability.

3.5 Derived Diagnostics

Glicko-2 Rating. Running rating updated after every game/hand, using the Glicko-2 system [Glickman, 1999] rather than classical Elo. Glicko-2 tracks rating deviation (uncertainty) alongside the point estimate, which is critical for early-phase measurements where few games have been played and rating confidence is low. It also accounts for rating volatility—a model that fluctuates wildly receives a wider confidence interval than one that performs consistently. Captures trajectory—improvement speed, plateau timing, recovery after opponent adaptation.

Why not classical Elo. With $K = 32$ and 500-game series, classical Elo fluctuates excessively and provides no uncertainty estimate. In a two-agent system, Elo is also purely relative: if Agent A improves and Agent B improves more, Agent A's Elo drops despite getting stronger. Glicko-2's deviation parameter makes this uncertainty explicit.

Game Quality Index (GQI). Average move quality measured against a strong oracle (Stockfish for chess). Two draws can be radically different: a 15-move repetition is stagnation; an 80-move endgame battle is mastery. GQI detects improvement even when outcomes don't change, and provides early warning of memory poisoning (retrieved stale information degrades decisions before it affects win rate).

GQI limitation. In Phases 4a–4c, Stockfish serves as both the tactical tool available to the agent and the evaluation oracle for GQI. This creates a circularity: the agent's move quality is judged by the same engine it consults. GQI in tool-augmented phases should therefore be interpreted as an *engine-alignment* metric (how well the agent follows Stockfish's advice) rather than an independent quality measure. For tool-augmented phases, win rate and Glicko-2 remain the primary metrics; GQI is reported but flagged.

4 Test Protocol

4.1 Phase 0: Sanity Gate

Each model plays 30 games against a random-move opponent (no augmentation). Pass criteria: >70% win rate, <20% error rate (invalid/illegal moves), binomial $p < 0.05$. Models that fail Phase 0 are excluded from subsequent phases. This gate ensures models can play legal, purposeful chess before measuring augmentation effects.

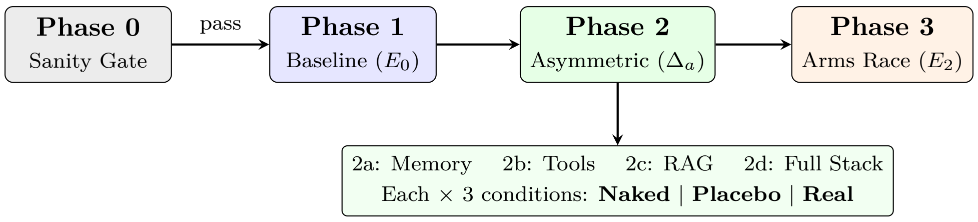

Figure 3: Protocol flow. Phase 0 gates entry; Phase 1 establishes naked baselines; Phase 2 measures Δ_a across four augmentation types under three experimental conditions; Phase 3 tests equilibrium when both agents are augmented.

4.2 Phase 1: Baseline (E_0)

Both agents play without augmentation. Chess: $N \geq 500$ games, alternating colors. Poker: $N \geq 10{,}000$ hands. Establishes baseline performance.

Self-play vs cross-model. Phase 1 uses self-play (same model as both agents) to establish each model's individual baseline. Cross-model comparisons are derived from the matrix (Section 3.4) by comparing baseline rows, not from direct head-to-head play without augmentation. This avoids confounding baseline measurement with model strength differences.

4.3 Phase 2: Asymmetric (Δ_a Measurement)

Agent A receives augmentation; Agent B plays naked (same model). Same N as Phase 1. The performance difference $= \Delta_a$. This phase is repeated for each augmentation type to isolate individual effects:

- Phase 2a: + Memory only
- Phase 2b: + Tools only (e.g., Stockfish)
- Phase 2c: + RAG only
- Phase 2d: + Full stack (all augmentations combined)

The difference $\Delta_a^{2d} - (\Delta_a^{2a} + \Delta_a^{2b} + \Delta_a^{2c})$ reveals interaction effects: positive means synergy, negative means redundancy.

Three experimental conditions (applied within each sub-phase):

- **Naked:** No augmentation (control).
- **Placebo:** Randomized augmentation of equal size—shuffled memory entries, random tool outputs, irrelevant retrieved documents (active control).
- **Real:** Genuine augmentation (treatment).

Real augmentation must outperform placebo to demonstrate that augmentation *quality* matters, not just additional context in the prompt. Without placebo, "memory helps" could mean "more tokens in context helps." The three conditions are orthogonal to the four augmentation types (2a–2d): each sub-phase can be run under all three conditions.

Run separately for each model × augmentation combination.

4.4 Phase 3: Arms Race (E_2)

Both agents receive augmentation (same or different systems). Same N. Measures equilibrium when both sides adapt.

Recovery τ protocol. At a pre-defined trigger point (configurable), one agent's strategy is forcibly shifted. Measures adaptation speed. Recovery τ is games/hands until performance returns within 5% of pre-shift baseline.

4.5 Phase 4: Full Orchestration (Chess Only)—*Extended Protocol*

LLM + Stockfish (as tool) + Memory vs human player. Tests the complete agent architecture.

Note: Phase 4 requires human opponents, making it expensive, slow, and difficult to standardize. It is an optional extended protocol—not required for computing the Agzamov Score. The core benchmark (Phases 0–3) is fully automated. Phase 4 results are reported separately when available.

Sub-phases for attribution:

$$\begin{aligned}
&\text{Phase 4a:} && \text{Stockfish alone vs Human} && \rightarrow \text{engine baseline}\\
&\text{Phase 4b:} && \text{LLM + Stockfish vs Human} && \rightarrow \text{orchestration effect}\\
&\text{Phase 4c:} && \text{LLM + Stockfish + Memory vs Human} && \rightarrow \text{full stack}
\end{aligned}$$

$$\Delta_{\text{orchestration}} = 4\text{b} - 4\text{a}, \quad \Delta_{\text{memory}} = 4\text{c} - 4\text{b}, \quad \Delta_{\text{full}} = 4\text{c} - 4\text{a} \tag{3}$$

4.6 Phase 5: Positional Stress Test (Chess Only)

Agents placed into pre-configured positions with known evaluations:

- **Equal** (eval 0.0 ± 0.5): does augmentation improve quality when outcome is uncertain?
- **Slight disadvantage** (-1.5 to -3.0): can augmentation turn a loss into a draw?
- **Severe disadvantage** (-3.0 to -5.0): measures survival time under pressure.

Competence is revealed under pressure, not in comfort. An agent that fights intelligently from losing positions—leveraging memory of opponent-specific weaknesses—demonstrates adaptive resilience.

5 Environments

5.1 Chess960

Two AI agents play $N \geq 500$ games of Chess960 (Fischer Random). Colors alternate every game. Win = 1 point, draw = 0.5, loss = 0.

Why Chess960, not standard chess. LLMs have extensive knowledge of standard openings from training data. Chess960 randomizes the starting position (960 configurations), greatly reducing the value of memorized opening book knowledge. Strategic insight must come primarily from real-time reasoning or retrieved memory. Parametric knowledge of standard opening theory becomes largely irrelevant.

5.1.1 Synthetic Opponent Patterns

To further isolate augmentation effects, agents face opponents with **injected behavioral patterns**—configurable tendencies that must be inferred online from match evidence only:

- "Opponent always castles within 8 moves when possible"
- "Opponent avoids trading queens until forced"

Patterns are enforced via system prompt constraints on the opponent agent—it still plays autonomously, just with a behavioral tendency. This is distinct from move substitution. System prompt constraints produce naturalistic behavior with a detectable statistical signature—exactly the kind of pattern a memory-equipped agent should learn to exploit.

Note on contamination. We do not claim these patterns are absent from training data—such a claim is unprovable for any closed-weight model. Instead, the protocol relies on Chess960's positional novelty and the combinatorial context (specific pattern × specific random starting position × specific game state) to ensure that rote recall is insufficient. The test measures whether an agent can detect and exploit a *specific opponent's* tendency within a match, not whether it has general knowledge of similar tendencies.

5.2 Poker

Two AI agents play $N \geq 10{,}000$ hands of No-Limit Texas Hold'em (heads-up). Measured in bb/100.

Poker is the ideal adversarial testbed because:

- **Nash Equilibrium is well-defined.** GTO strategy [Nash, 1950] is the unexploitable baseline (E_0). Any deviation is either a mistake to punish or a trap.
- **Opponent modeling is everything.** Without memory, an agent can only play GTO. With memory, it detects that the opponent folds to 3-bets 80% of the time and exploits.
- **Variance requires volume.** More hands needed for significance—also tests memory's ability to extract signal from noise.

Synthetic patterns for poker: "Opponent always min-raises with pocket pairs" or "Opponent folds to 3-bets 85% on the button."

Statistical note. Standard deviation in HU NLHE ≈ 80–100 bb/100. To detect 3–5 bb/100 effect at $p < 0.05$ with 80% power, minimum ∼7,000–15,000 hands. 10,000 hand minimum detects moderate effects (≥ 4 bb/100).

5.2.1 Poker State Representation and the Naked Baseline Problem

LLMs have finite context windows. 10,000 hands of poker history cannot fit in a single prompt. This creates a methodological tension: in the Naked condition, the model receives only the current hand state (hole cards, board, pot, action history for this hand). It has no memory of previous hands and plays each hand independently—effectively a one-shot GTO approximation.

This means the Naked baseline in poker is fundamentally different from the Naked baseline in chess. In chess, the model plays a full game with sequential moves and accumulates within-game context. In poker, each hand is a fresh decision with no cross-hand information.

This is a feature, not a bug. The Naked poker baseline establishes what a model can do with zero opponent history—pure strategy from first principles. The Δ_a then measures exactly what cross-hand memory adds: opponent modeling, pattern exploitation, adaptation. The gap between "play each hand in isolation" and "play with opponent history" is the core measurement.

Naked state input per hand:

```
Hole cards: [Ah, Kd]
Board: [Qs, Jh, 3c, 7d]
Pot: 12 BB | To call: 4 BB
Action: Opponent bet 4 BB
Position: Button
```

Augmented state input per hand: Same as above, plus a memory-retrieved summary:

```
Opponent profile (last 200 hands):
- VPIP: 68% | PFR: 42% | 3-Bet: 12%
- Folds to c-bet: 55% | Check-raise freq: 8%
- Bluff-to-value ratio on river: 2.1:1
```

The augmented agent receives the same hand state plus a structured opponent summary from external memory. Memory quality determines summary accuracy; the benchmark measures whether better memory produces better exploitation.

6 Theoretical Foundation

6.1 Game Theory Basis

The test is grounded in the distinction between one-shot and repeated games [von Neumann and Morgenstern, 1944]. The Folk Theorem [see e.g., Aumann and Shapley, 1994] establishes that in repeated games, any individually rational outcome can be sustained as Nash Equilibrium through adaptive strategies—a principle demonstrated empirically in Axelrod [1981]'s iterated prisoner's dilemma tournaments. The Augmentation Delta measures exactly this shift:

$$\Delta_a = E_{\text{repeated}} - E_{\text{one-shot}} \tag{4}$$

This is the theoretical basis for the operational metric Δ_a defined in Equation (2): E_{repeated} corresponds to augmented performance P_a (where memory enables cross-game learning), and $E_{\text{one-shot}}$ corresponds to naked baseline P_0.

6.2 Disentangling Model vs. Augmentation

A predictable objection: "You are measuring model and augmentation together—how do you know which one is responsible?"

The matrix (Section 3.4) is the analytical instrument:

- If $\Delta_a \approx 0$ across an entire row (fixed model, all augmentations): the bottleneck is the model.
- If Δ_a is consistently higher in one column (fixed augmentation, all models): that augmentation provides value independent of model.
- If Δ_a varies by row and column: both matter, and interactions reveal synergies vs redundancies.

6.3 The Evaluation-Calculation Gap

LLMs possess two separable chess competencies: positional *evaluation* (pattern-matching assessment of who stands better) and tactical *calculation* (tree-search computation of forced sequences). Evaluation is strong—models correctly identify winning positions. Calculation is weak—current models struggle to compute forced mating sequences even in trivially won positions.

This is not a problem for the benchmark. It is a finding. The benchmark *measures* this gap rather than assuming it away. At each augmentation level, the gap manifests differently:

- **Naked:** Model relies on evaluation only. Calculation gap limits performance.
- **+ Memory:** Memory helps evaluation (opponent patterns, positional preferences). Calculation gap persists.
- **+ Tools:** Stockfish provides calculation. Gap is filled by tool use.
- **+ Full Stack:** Memory directs tool use toward opponent-specific weaknesses. Synergy.

The benchmark tracks how this gap closes across augmentation levels—a dimension not captured by existing benchmarks.

6.4 Production Relevance

The evaluation-calculation gap is not a chess curiosity—it maps directly to failure modes in production AI systems. A medical AI that correctly identifies a disease (evaluation) but cannot plan a multi-step treatment protocol (calculation). A financial model that recognizes an undervalued asset (evaluation) but cannot construct a hedging strategy across correlated instruments (calculation). A coding assistant that understands what a function should do (evaluation) but cannot reason through a multi-step refactor without introducing bugs (calculation).

In all these domains, the pattern is identical: pattern-matching competence paired with sequential-reasoning fragility. The Agzamov Test provides a controlled environment to measure this gap and to quantify how augmentation (memory, tools, orchestration) can compensate for it. Results from chess and poker are not directly transferable to medicine or finance—but the *structure* of the capability gap, and the *degree* to which augmentation closes it, are informative for any domain where AI systems must combine evaluation with multi-step planning.

7 Hypotheses

7.1 Primary Hypotheses (H1–H5)

Bonferroni correction applied across all five; per-test $\alpha = 0.01$.

H1: Augmentation helps. $\Delta_a > 0$ for real augmentation in both environments.
H2: Real memory > placebo. Real memory outperforms random/fake memory of equal size, demonstrating that memory quality matters, not just additional context.
H3: Environment interaction. Δ_a differs between chess (complete information) and poker (incomplete information), revealing how augmentation value depends on information structure.
H4: Synergy. Full-stack Δ_a (Phase 2d) exceeds the sum of individual augmentation deltas ($\Delta_a^{2a} + \Delta_a^{2b} + \Delta_a^{2c}$). Memory + tools are synergistic, not additive—memory directs calculation toward opponent-specific weaknesses.
H5: Speed matters. τ varies significantly across augmentation systems even when Δ_a is similar—convergence speed is an independent quality dimension.

7.2 Exploratory Hypotheses (E1–E12)

Not corrected for multiple comparisons. Reported with uncorrected p-values and effect sizes; Benjamini-Hochberg FDR correction reported alongside for transparency.

E1: Weaker models (Haiku-class) may show $\Delta_a \approx 0$ even with high-quality augmentation—model capability floor.
E2: Δ_a is compressed in chess vs poker due to high draw rates between equal models.
E3: Memory-equipped agents show largest relative improvement in slight-disadvantage positions (Phase 5).
E4: Temporally weighted memory outperforms uniform memory in Phase 3 (arms race), where older information is stale.
E5: Memory augmentation improves evaluation-dependent play (middlegame) more than calculation-dependent play (endgame).
E6: The evaluation-calculation gap is measurable as Evaluation Accuracy >85% and Calculation Accuracy <30% for current frontier models.
E7: ARC-AGI-2 scores do not correlate with Agzamov Calculation Efficiency scores.
E8: Recovery τ (after opponent strategy shift) correlates more with augmentation quality than initial τ.
E9: The ratio $\tau_{\text{poker}}/\tau_{\text{chess}}$ characterizes a system's noise tolerance.
E10: Memory-equipped agents show equal or lower invalid move rates vs naked agents.
E11: Without material adjudication, won positions are recorded as draws, biasing Δ_a.
E12: Bad memory (high retrieval noise) produces $\Delta_a < 0$—worse than no memory.

8 Open Questions

1. Does the Δ_a–augmentation quality relationship scale linearly, or are there phase transitions?
2. Can superior augmentation compensate for inferior model capability?
3. Is there a ceiling to Δ_a regardless of augmentation quality?
4. Does multi-game memory transfer across opponents (general learning vs opponent-specific memorization)?
5. In Phase 3, does an arms race emerge or does the system converge to equilibrium?
6. Can the test extend to multi-agent environments (3+ players in poker)?

9 Implementation

9.1 Technical Requirements

- Chess engine: `python-chess` (Chess960 mode); Stockfish via MCP for Phase 4
- Poker engine: heads-up NLHE with standard hand evaluation
- Memory systems: BrainOps Memory MCP as primary; competitor systems for matrix
- Model API access: Claude, GPT, Gemini (minimum 3 providers)
- Statistical framework: see Section 9.5
- Game/hand history storage for reproducibility
- Error tracking: invalid/illegal moves per agent per game

9.2 Memory Audit Protocol

Problem. Without content restrictions, a memory system can be pre-loaded with external knowledge. This measures knowledge injection, not augmentation quality.

Rule. Memory systems may only store information derived from the current match.

Table 6: Augmentation Audit Protocol: allowed and forbidden content.

Allowed	**Forbidden**
Game/hand IDs and timestamps	Pre-loaded opening databases
Observed moves and actions	External GTO charts or solvers
Derived patterns with evidence trail	Opponent data from outside the match
Consolidated analytical summaries	General strategy guides

Enforcement. (1) Pre-match audit—memory store verified empty. (2) Post-match dump—full contents exported for review. (3) Content hash chain—every write logged with source game ID. (4) Automated validation—orphan entries = contamination flag. Audit logs published alongside results. Runs without audit logs are unverified.

9.3 Sampling Determinism

LLMs are stochastic: temperature > 0 introduces response variance independent of augmentation effects. This noise can mask or inflate Δ_a measurements.

Default protocol. All models run at temperature $= 0$ (greedy decoding) for the primary benchmark. This maximizes reproducibility and isolates augmentation effects from sampling noise.

Exception. Reasoning models (OpenAI o-series) do not support temperature control—they use internal chain-of-thought with provider-managed sampling. These models are benchmarked in their default configuration, and their inherent stochasticity is documented.

Robustness check. A subset of games (≥ 50 per condition) is re-run at temperature = 0.3 to measure sensitivity. If Δ_a changes by more than 1 SE between temperature settings, both results are reported.

9.4 Error Handling

1. Invalid move $\rightarrow$ random legal move substitution.
2. Error rate tracked per agent, per phase.
3. Δ_a calculated with and without error-containing games.
4. Agent exceeding 5% error rate = flagged as unreliable.

9.5 Statistical Framework

Significance threshold. $\alpha = 0.05$ for all tests, with corrections as described below.

Primary hypotheses (H1–H5). Bonferroni correction applied across the 5 primary hypotheses, yielding per-test $\alpha = 0.01$. These are the confirmatory tests; results are reported as significant only if they survive correction.

Exploratory hypotheses (E1–E12). No correction applied. These are clearly labeled as exploratory and reported with uncorrected *p*-values and effect sizes. Benjamini-Hochberg FDR correction is reported alongside for transparency, but individual E-hypotheses are not claimed as confirmed findings.

Confidence intervals. 95% bootstrap CIs (10,000 resamples) reported for all point estimates (win rates, Δ_a, τ, GQI). Glicko-2 ratings reported with ± 1 SE.

Effect sizes. Cohen's *d* or equivalent reported alongside *p*-values for all primary hypotheses. Statistical significance without meaningful effect size is noted but not emphasized.

Multiple model comparisons. When comparing N models in the matrix (Section 3.4), pairwise comparisons use Tukey's HSD or equivalent. The matrix is presented as descriptive; only pre-registered contrasts are tested for significance.

Chess-specific. Win rates tested via binomial test (Phase 0) or paired comparison (alternating colors). Poker: Welch's *t*-test on session-level bb/100 means, with sessions of 100 hands.

9.6 Compute Budget Fairness

Models differ dramatically in per-move compute: reasoning models (OpenAI o3, o4) consume 10–50× more tokens and wall-clock time than standard models (Sonnet, GPT-4o) for a single move. This creates a fairness question: is a reasoning model "better" at chess, or just spending more compute?

The Agzamov Test does not attempt to equalize compute. Each model runs in its default mode with provider-default parameters. Reasoning models use chain-of-thought by design; suppressing it would measure a crippled model. Instead, compute is tracked and reported:

- **Tokens per move** (input + output, including reasoning tokens)

- **Wall-clock time per move** (mean, median, p95)
- **API cost per game** (provider-reported)
- **Total run cost** per phase

This allows readers to construct their own efficiency frontier: performance vs compute, performance vs cost. A model that wins 80% of games at $0.10/game is arguably more useful than one that wins 85% at $5.00/game. The benchmark reports both dimensions; the tradeoff is left to the reader.

10 Phase 0: Infrastructure Validation

Before measuring augmentation effects, Phase 0 validates that the LLM agent can play legal Chess960 moves consistently, that game recording is accurate, and that the cost model supports the full protocol.

10.1 Setup

A single model (Claude Sonnet 4, Anthropic) played 30 games of Chess960 against a random-move opponent—no augmentation on either side. Colors alternated. Each game used a distinct randomly-selected Chess960 starting position. Games were terminated by checkmate, insufficient material, or Stockfish adjudication when evaluation exceeded ±10.0 pawns for 3 consecutive moves.

The random opponent provides a deterministic lower bound: any competent agent should achieve near-perfect results. Phase 0 validates infrastructure, not strategic capability.

10.2 Results

Table 7: Phase 0 validation results: Claude Sonnet 4 vs. random opponent, 30 Chess960 games (no augmentation).

Metric	**Value**
Games played	30
Model wins	29 (96.7%)
Draws	1 (3.3%)
Model losses	0 (0.0%)
Model score	0.983
p-value (binomial)	< 0.001
Total API calls	857
Format errors	0 (0.0%)
Illegal move errors	0 (0.0%)
Average game length	54.3 moves
Average game duration	270 seconds
Total cost (API)	$6.11 USD
Termination reasons	
Adjudication (Stockfish)	28
Checkmate	1
Insufficient material	1

10.3 Observations

Three patterns emerge that inform the full protocol:

Zero error rate validates the harness. Across 857 API calls, the model produced zero format errors, zero illegal moves, and zero nonsensical outputs. Modern LLMs can reliably interface with Chess960 game state representations. The 5% error threshold (Section 9.4) was never approached.

Adjudication dominates checkmate 28:1. The model consistently achieved decisive material advantages but converted only once via actual checkmate. The single draw (game 27, 170 moves) illustrates the extreme case: the model captured all opposing material but traded its own in the process, reaching king-versus-king. This "greedy capture" behavior—prioritizing material gain over mating-net construction—is a measurable weakness and a potential target for augmentation: memory of endgame patterns from prior games, or tool-assisted endgame tablebases, could shift this ratio. Phase 2 will test this directly.

Cost model scales. Phase 0 total API cost was $6.11 for 30 games ($0.20/game against a random opponent). Per-game cost will differ in subsequent phases where both agents are LLMs; actual costs are tracked per phase and reported alongside results.

10.4 Limitations

Phase 0 establishes infrastructure reliability, not augmentation effects. A random opponent provides no adversarial pressure. Phase 1 introduces model-versus-model play where the baseline win rate approaches 50%, and augmentation effects become measurable.

11 Why "Agzamov"

The benchmark is named the Agzamov Test—a double reference to the author's surname, and to Georgy Agzamov (1954–1986), the first chess grandmaster from Central Asia. Georgy was known as the "nightmare of top grandmasters," defeating Tal and drawing Karpov through tenacity, pattern recognition, and counterattack rather than raw calculation. The benchmark aspires to a similar reputation among AI models: a nightmare that rewards adaptation over brute force.

12 Normative Specification (v0.1-draft)

Implementations claiming Agzamov Test compliance must adhere to the parameters below. Items marked † are configurable within stated bounds; all others are fixed.

Versioning. Any change to the values in Table 8 or to the A-Score formula constitutes a new protocol version. Results must state the protocol version used.

13 Roadmap

1. **Phase 0 (complete for 1 model)**—30 games vs random, Claude Sonnet 4 validated. Remaining models in progress.
2. **Preliminary data**—100+ games with/without memory, first Δ_a numbers.
3. **Paper**—Methodology + preliminary findings on arXiv (establish priority).
4. **MVP**—Full protocol: two agents, chess + poker, four augmentation levels.

Table 8: Normative protocol parameters.

Parameter	**Value**	**Ref**
Chess960		
Games per phase	≥ 500	§5.1
Phase 0 sanity games	30	§4.1
Phase 0 pass criteria	>70% win, <20% error, $p < 0.05$	§4.1
Color alternation	Every game	§5.1
Adjudication	Stockfish $\lvert\text{eval}\rvert > 10.0$ for 3 moves	§10
Poker (NLHE heads-up)		
Hands per phase	$\geq 10{,}000$	§5.2
Blind structure	Fixed (1/2 BB)†	§5.2
Starting stack	100 BB†	—
General		
Temperature	0 (default); 0.3 (robustness)	§9.3
Error threshold	5% invalid moves	§9.4
Significance	$\alpha = 0.05$; Bonferroni for H1–H5	§9.5
Bootstrap CIs	10,000 resamples, 95%	§9.5
Memory audit	Pre/post dump, content hash chain	§9.2
A-Score (Eq. 1)		
Sub-scores	7 (Table 4)	§3
Weights (v0.1)	Uniform: $w_i = 1/7$	§3
Range	0–100	§3

5. **Matrix**—3+ models × 3+ augmentation configs × 2 environments.
6. **Leaderboard**—Public, updatable (agzamovtest.com or HuggingFace Spaces).
7. **Standard**—Propose as standard evaluation framework for augmented AI.

14 Reference Implementation

The reference harness, evaluation scripts, and raw game data are available at:

`https://github.com/brainops-pub/agzamov-test`

The implementation includes: Chess960 agent harness (`python-chess`), Stockfish integration via MCP, memory audit tooling, statistical analysis pipeline, and a live game observation dashboard. All Phase 0 results reported in this paper were produced with this codebase.

Version. Results correspond to tag `v0.1.0` (to be created at publication). Subsequent protocol revisions will increment the minor version; changes to the A-Score formula or normative parameters increment the major version.

License. Code: MIT. Paper and specification: CC BY 4.0. The benchmark protocol is open; any memory system, model provider, or orchestration framework may be evaluated.

Competing Interests

The Agzamov Test uses BrainOps Memory MCP as its primary memory system in augmented conditions. BrainOps Memory MCP is developed by BrainOps Limited, the author's company. To mitigate conflict of interest: (1) the benchmark protocol is fully open and any memory system can be substituted; (2) the matrix design (Section 3.4) explicitly compares multiple memory systems, not just the author's; (3) all code, configuration, and raw game data are published for independent verification. The benchmark measures augmentation quality in general—it is not a product evaluation for any specific memory system.

Acknowledgments

This work was developed at BrainOps Limited with assistance from Claude (Anthropic) for iterative refinement. External review was provided by Gemini 2.5 Pro (Google DeepMind), GLM-4 (Zhipu AI), and GPT-4o (OpenAI)—these are LLM-based reviews, not human peer review.

References

ARC Prize. ARC prize 2025 results and analysis. ARC Prize blog, 2025. URL https://arcprize.org/blog/arc-prize-2025-results-analysis. Accessed February 2026.

Robert J. Aumann and Lloyd S. Shapley. Long-term competition—a game-theoretic analysis. In *Essays in Game Theory*, pages 1–15. Springer, 1994.

Robert Axelrod. The evolution of cooperation. *Science*, 211(4489):1390–1396, 1981.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*, 2021.

François Chollet. ARC-AGI-2: A new benchmark for artificial general intelligence. https://arcprize.org/blog/arc-agi-2, 2025.

Mark E. Glickman. Parameter estimation in large dynamic paired comparison experiments. *Journal of the Royal Statistical Society: Series C (Applied Statistics)*, 48(3):377–394, 1999.

Google DeepMind. Gemini 3.1 Pro — model card. Model card, 2026a. URL https://deepmind.google/models/model-cards/gemini-3-1-pro/. Accessed February 2026.

Google DeepMind. Gemini 3.1 Pro technical report. https://deepmind.google/technologies/gemini/, 2026b. Accessed February 2026.

Ryan Greenblatt. Getting 50% (SoTA) on ARC-AGI with GPT-4o. Redwood Research blog, 2024. URL https://blog.redwoodresearch.org/p/getting-50-sota-on-arc-agi-with-gpt. Accessed February 2026.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021.

Percy Liang, Rishi Bommasani, Tony Lee, Dimitris Tsipras, Dilara Soylu, Michihiro Yasunaga, Yian Zhang, Deepak Narayanan, Yuhuai Wu, Ananya Kumar, et al. Holistic evaluation of language models. *Annals of the New York Academy of Sciences*, 1525(1):140–146, 2023.

John F. Nash. Equilibrium points in n-person games. *Proceedings of the National Academy of Sciences*, 36(1):48–49, 1950.

David Silver, Thomas Hubert, Julian Schrittwieser, Ioannis Antonoglou, Matthew Lai, Arthur Guez, Marc Lanctot, Laurent Sifre, Dharshan Kumaran, Thore Graepel, et al. A general reinforcement learning algorithm that masters chess, shogi, and Go through self-play. *Science*, 362(6419):1140–1144, 2018.

Ivan Sorokin and Jean-François Puget. NVARC: Test-time training for ARC-AGI-2. https://github.com/1ytic/NVARC, 2025. 1st place, ARC Prize 2025 (27.6% on public leaderboard).

John von Neumann and Oskar Morgenstern. *Theory of Games and Economic Behavior*. Princeton University Press, 1944.