

Ploidy: Context-Asymmetric Structured Debate for LLM Decision Verification

heznpc

Abstract

Single-session LLM usage subjects critical decisions to stochastic prior lock-in: the model’s first probabilistic response anchors all subsequent reasoning, and prompt-based mitigations have no statistically significant effect on this bias (?). We present Ploidy, a structured debate protocol between physically separate sessions of the same model with intentionally asymmetric context depths. We identify two independent phenomena that multi-session debate must address: **Event A** (context asymmetry—sessions disagree because they have different information) and **Event B** (stochastic variance—sessions with identical context disagree due to sampling randomness). The *ploidy level*—sessions per context depth—controls which event can be distinguished: at $1n$ (haploid), the two are confounded; at $2n+$, within-group agreement isolates stochastic variance from context-driven disagreement. We further introduce **context injection mode** (memory, skills, system prompt, raw) as a novel experimental variable, finding that the *form* of context delivery—not just its content—moderates debate efficacy. In 6 pilot experiments (10 tasks, primarily single-run) on long-context software architecture tasks with Claude Opus 4.6 and Sonnet 4.6, we find: (1) Ploidy achieves highest recall across all injection modes, (2) memory-style injection degrades single-session recall by 20% while Ploidy drops only 6%, (3) diploid ($2n$) debate reaches maximum recall at 30% additional compute with no benefit from higher ploidy, and (4) injection mode changes which method performs best. The intersection of intentional context asymmetry, structured cross-session debate, injection mechanism as experimental variable, and ploidy-level stochastic sampling has zero prior publications as of March 2026.

1 Introduction

LLM outputs are stochastic. The same model, given the same prompt, produces different responses across independent sessions. This is well-understood. What is less appreciated is the downstream consequence for single-session workflows:

1. The model’s first response is sampled from a probability distribution.
2. That response enters the context window and becomes the model’s own prior.
3. The model reinforces this prior through consistency-seeking behavior (anchoring bias, sycophancy).
4. The user sees only one session and treats the output as deterministic.

The result: identical models, identical prompts, identical users—but different project outcomes depending on which stochastic sample landed first. Jacob et al. (?) term this the “Chat-Chamber Effect”—the tendency for users to trust and build upon whatever stochastic output a single session produces. This is not addressable by temperature tuning or prompt engineering; empirical evidence

shows prompt-based mitigations (chain-of-thought, reflection, “ignore your prior response”) have no statistically significant effect on anchoring bias (?).

The practical consequence is stark: given the same model and the same prompt, different sessions may produce fundamentally different assessments—agree, disagree, or equivocate—on the same hypothesis. A user confined to a single session is unknowingly subject to a stochastic lottery whose outcome determines not just the quality of individual responses, but whether entire tasks succeed or fail. This extends beyond response quality to task-level outcomes: the same model may complete a task in one session and fail it in another, or require substantially different time to converge, solely due to which stochastic sample anchored the initial response. Two users with identical models and identical problems may experience dramatically different “model performance” based solely on which stochastic sample anchored their session.

A natural response is to run multiple sessions and aggregate. But this conflates two independent problems. **Event B** (stochastic variance): sessions with identical context disagree because sampling is random—more sessions reduce variance but cannot correct systematic blind spots. **Event A** (context-induced bias): a session anchored to accumulated project history has systematic blind spots that no amount of re-sampling from the same context can fix. Running 100 sessions with the same context will converge on the same biased consensus. Only a session with *different* context—one that has never seen the project history—can detect what the context itself has made invisible. This distinction is the core motivation for Ploidy: not more sessions, but sessions with intentionally different information states.

This failure mode is structurally analogous to apoptosis failure in cell biology. The problem is not that the session performs poorly—it is that a session with accumulated errors *does not self-terminate*. It continues generating outputs, the user continues trusting them, and biased conclusions propagate through persistent memory and downstream decisions. A low-performing session that resets is functioning correctly; a high-performing session locked into a biased trajectory is the dangerous case, because its apparent competence masks the accumulated error. Ploidy’s Fresh session serves as an external checkpoint—it cannot be anchored to the Deep session’s prior outputs, so it detects errors that the Deep session’s own context window has rendered invisible.

The only computational intervention with empirical support for this problem is physical session separation. Cross-Context Review (CCR) (?), published concurrently with and independently of this work, demonstrated that a fresh session reviewing an artifact produced by a deep session achieves F1=28.6% on error detection versus 24.6% for same-session review ($p = 0.008$). CCR validates the core premise that context separation improves review quality. However, CCR is unidirectional—the fresh session reviews but does not debate. Song’s follow-up, Dynamic CCR (D-CCR) (?), tested multi-turn extensions of CCR and found that additional review rounds *degrade* performance: single-pass CCR (F1=0.376) significantly outperformed all multi-turn variants ($p < 0.001$), with later rounds introducing false positive pressure and “Review Target Drift” where reviewers critique the conversation rather than the artifact. D-CCR’s negative result supports our design choice of structured bidirectional debate over iterated unidirectional review—the problem is not insufficient rounds but insufficient *structure* in how disagreements are processed.

We extend CCR from unidirectional review to **bidirectional structured debate** and introduce the **Context Asymmetry Spectrum**—a continuum from full context (Deep) through compressed context (Semi-Fresh) to zero context (Fresh). This spectrum recognizes that the optimal information state for a challenger may be neither complete ignorance nor full knowledge, but an intermediate point analogous to the “experienced outsider” who brings domain awareness without institutional entrenchment.

Two independent phenomena. We identify that context asymmetry (Event A) and stochastic variance (Event B) are independent events requiring different interventions. A haploid (1n) debate—Deep(1)×Fresh(1)—addresses Event A but samples only one point from each distribution. Diploid (2n) and higher address both simultaneously: context asymmetry *between* groups, stochastic sampling *within* groups.

Contributions.

- The distinction between Event A (context asymmetry) and Event B (stochastic variance) as independently measurable phenomena, controlled by ploidy level (§??)
- A structured debate protocol with typed semantic actions and convergence analysis supporting N-ary sessions (§??)
- Context injection mode (memory, skills, system prompt, raw, CLAUDE.md) as a novel experimental variable for debate efficacy (§??)
- Preliminary evidence that diploid (2n) is the cost-optimal ploidy level for recall maximization (§??)
- An open-source MCP server implementation (§??)

2 Background and Related Work

2.1 Context Degradation in Long Sessions

LLM performance degrades with context length even when retrieval is perfect—Du et al. (?) showed 13.9–85% performance drops that are architectural, not retrieval-related. Chroma Research (?) evaluated 18 frontier models and found effective context capacity is approximately 60–70% of advertised window size. The “scaling paradox” (?) further shows that larger context compressors produce less faithful reconstructions due to knowledge overwriting.

2.2 Multi-Agent Debate

Multi-agent debate (MAD) is well-studied but predominantly uses symmetric configurations where all agents share the same context. Recent work has explored multi-LLM context learning for richer discussion dynamics (?), but the fundamental assumption of shared context remains. Oh et al. (?) demonstrated that symmetric debate can amplify bias through “belief entrenchment.” Choi et al. (?) proved that debate under symmetric information induces a martingale—it cannot improve expected correctness beyond majority voting.

ColMAD (?) reframes debate from competitive zero-sum to collaborative non-zero-sum, where agents critique supportively rather than adversarially, reporting 19% improvement over competitive MAD. Ploidy shares this collaborative direction through its `propose_alternative` and `synthesize` semantic actions. However, ColMAD and similar collaborative frameworks still operate under symmetric information—all agents share the same context, leaving them vulnerable to collective anchoring. Ploidy extends the collaborative paradigm by combining structured semantic actions with intentional context asymmetry.

2.3 Asymmetric Context as a Mechanism

SR-DCR (?) introduced asymmetric context verification debate (ACVD): a context-defending agent debates a context-deprived critic. On GPT-3.5, this achieved 62.7% accuracy (+3.4pp over naive debate). AceMAD (?) proved that asymmetric cognitive potential creates submartingale drift toward truth, formally breaking the martingale curse. We note that Ploidy’s current single-round design does not satisfy the multi-round conditions required for AceMAD’s submartingale proof; whether a multi-round extension would achieve this remains an open question (§??).

AceMAD provides the theoretical foundation demonstrating *that* asymmetry works. Ploidy serves as empirical validation and system-level implementation: answering the architectural questions of *what form* of context asymmetry to inject (the Context Asymmetry Spectrum), *how* to deliver it (injection mode), and *how many* sessions are needed to separate structural bias from stochastic noise (ploidy level). We note that Ploidy was developed independently and concurrently with AceMAD, motivated by practical experience with context entrenchment in long-running coding sessions rather than formal game-theoretic analysis. The convergence of an empirical practitioner-driven approach and a theoretical proof-driven approach on the same core insight—that intentional information asymmetry breaks anchoring—strengthens the case for both.

2.4 Scaling Agents vs. Scaling Diversity

Large-scale agent simulations (e.g., AgentSociety (?) with 10K agents, MiroFish with 700K agents) pursue verification breadth—more agents analyzing the same problem. Boca et al. (?) showed that even without individual-level bias, LLM populations spontaneously develop collective biases through interaction. Combined with Choi et al.’s martingale result (?), this suggests that scaling homogeneous agent count does not reliably improve decision quality. Ploidy pursues the orthogonal dimension: verification depth through context diversity.

2.5 Sycophancy and Persistent Context Bias

Jain et al. (?) showed that user memory profiles increase agreement sycophancy by up to 45% (Gemini 2.5 Pro), with even synthetic contexts causing 15% uptick. Harshavardhan (?) documented self-anchoring calibration drift in multi-turn conversations: Claude Sonnet 4.6 shows significant decreasing confidence under self-anchoring. Laban et al. (?) reported 39% performance degradation in multi-turn vs. single-turn settings, finding that “when LLMs take a wrong turn, they get lost and do not recover.” These results motivate Ploidy’s Fresh session as a structural reset mechanism.

2.6 Context Injection as Pseudo-Fine-Tuning

Context injection and fine-tuning are technically distinct: the former modifies input, the latter modifies weights. Context is ephemeral; fine-tuning is permanent. However, the behavioral boundary is blurring. Persistent context files (CLAUDE.md, memory.md) auto-load every session, functioning as de facto permanent input. In-context learning shifts model behavior to a degree comparable to fine-tuning. RAG pipelines inject external knowledge on every call.

The practical consequence is that developers are *pseudo-fine-tuning* their models through context engineering without recognizing it as such. A memory file that accumulates session observations (“We tried X and it failed”, “The team decided Y”) causes the model to treat these as internalized knowledge rather than external reference—producing anchoring effects that mimic fine-tuned priors. A skills file with declarative rules (“RULE: always check for X”) is treated as external

constraints, producing weaker priors. The *form* of context delivery modulates how strongly the model internalizes it, independently of the information content.

This is why Ploidy treats injection mode as an independent experimental variable: it controls the degree to which context *acts like* fine-tuning. A Fresh session can escape context-induced bias precisely because context is not weight—remove the input and the model reverts. Fine-tuned bias cannot be escaped through session separation.

No prior work treats the delivery mechanism—memory file vs. skills file vs. system prompt—as an experimental variable for debate outcomes. Vercel (2026) compared AGENTS.md vs. skills for test pass rates; ETH Zurich (2026) found AGENTS.md files may hinder AI coding agents. Neither connected injection mechanism to anchoring strength or debate efficacy.

2.7 Model Collapse and Context Contamination

When AI-generated content enters training data, progressive quality degradation occurs—“model collapse” (?). Ploidy’s design of maintaining sessions with different context depths is a within-deployment countermeasure to this homogenization tendency.

2.8 Positional Bias in Context Delivery

LLM attention is non-uniform across the context window. Liu et al. (?) demonstrated a U-shaped performance curve: models attend strongly to information at the beginning and end of the prompt but degrade on middle-positioned content (“Lost in the Middle” effect). This positional bias is well-established and reflected in official prompt engineering guidelines from both Anthropic and OpenAI.

However, prior multi-agent debate literature has not accounted for how positional bias interacts with debate dynamics. When a compressed summary is injected at the top of a Semi-Fresh prompt (SF-Passive), the model encounters the Deep session’s framing *before* the raw artifact—anchoring all subsequent reasoning on that framing. When the same summary is placed at the bottom (SF-Passive-Bottom) or delivered via active retrieval after independent analysis (SF-Active), the positional anchoring is attenuated. Our ablation results are consistent with this mechanism. The contribution is not the discovery of positional bias itself, but the demonstration that **context injection architecture is a moderator variable for multi-agent debate efficacy**—a dimension absent from existing MAD research.

2.9 Ploidy’s Position

2.10 Distinction from Multi-Agent Task Division

Claude Agent Teams and similar systems (CrewAI, MetaGPT) implement cooperative division—splitting work across agents for throughput. Adding more agents increases throughput (more hands) but does not address anchoring bias, because all agents share the same context and thus the same stochastic priors. Under Choi et al.’s martingale result (?), scaling homogeneous agents is mathematically equivalent to majority voting over identically biased samples. Ploidy implements the orthogonal strategy: cooperative verification, where the same problem is analyzed from intentionally different information states. The goal is not more output but different perspectives—disagreements are the primary signal, not a failure mode.

Table 1: Comparison of Ploidy with prior work along key design dimensions.

Prior Work	Mechanism	Context Depth	Delivery	Direction
CCR (?)	Session separation	Binary (Deep/Fresh)	Passive	Unidirectional review
D-CCR (?)	Multi-turn CCR	Binary (Deep/Fresh)	Passive	Iterated unidirectional (degrades)
SR-DCR/ACVD (?)	Asymmetric debate	Binary (Defender/Critic)	Passive	Unidirectional + Judge
AceMAD (?)	Asymmetric potential	Theoretical	n/a	Multi-round proof
AgentSociety (?)	Population simulation	Homogeneous	Passive	Observation only
Ploidy	Structured debate	Spectrum (Deep→Fresh)	Passive Active	+ Bidirectional convergence +

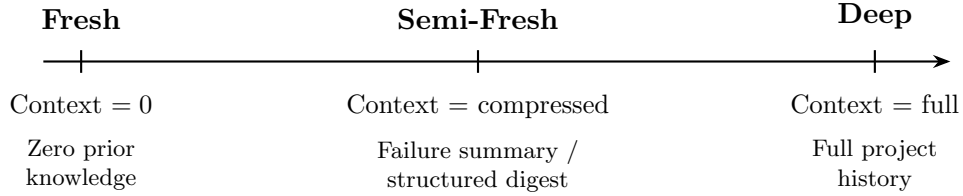


Figure 1: The Context Depth Spectrum: sessions range from zero prior knowledge (Fresh) through compressed summaries (Semi-Fresh) to full project history (Deep).

3 The Ploidy Protocol

3.1 The Context Asymmetry Spectrum

Prior work treats context as binary: full context vs. no context. We propose a two-dimensional spectrum defined by **context depth** and **context delivery mode**.

Context depth determines how much prior knowledge a session receives:

- **Deep**: Full project context—codebase, decision history, accumulated knowledge. Maximizes domain awareness but risks anchoring bias.
- **Semi-Fresh**: Compressed context—a structured summary of prior attempts, known constraints, or failure modes, without the full narrative that induces entrenchment. Analogous to an experienced outsider or a practitioner restarting a stuck project with lessons learned but freed from sunk-cost attachment.
- **Fresh**: Zero prior context—only the raw artifact under review. Maximizes independence but lacks domain constraints.

Context delivery mode determines how context reaches the session, independently of depth:

- **Passive delivery**: Context is embedded directly in the prompt and is always present in the context window. Every response is implicitly influenced by this context, whether or not it is relevant to the specific question being answered. This mirrors the expert whose 20 years of experience unconsciously shapes every judgment.

Figure 2: The Context Asymmetry Spectrum: a 2D space defined by context depth (rows) and delivery mode (columns), yielding five distinct session configurations.

	Passive (always in window)	Active (tool/retrieval)	None (absent)
Full	Deep (<i>expert intuition</i>)	Deep-Active (<i>expert + lookup</i>)	n/a
Compressed	Semi-Fresh-Passive (<i>briefed outsider</i>)	Semi-Fresh-Active (<i>consultant</i>)	n/a
None	n/a	n/a	Fresh (<i>novice</i>)

- **Active delivery:** Context is available through an explicit retrieval mechanism (e.g., a tool call) but is not present in the window until requested. The session must decide when and whether to consult prior knowledge. This mirrors the consultant who researches on demand rather than relying on ingrained assumptions.

The same information, delivered passively vs. actively, may produce different entrenchment dynamics. This prediction is grounded in established cognitive science findings: the generation effect (?) shows that actively produced knowledge is more deeply processed than passively received information, and the testing effect (?) demonstrates that retrieval practice outperforms re-reading for durable learning. Passive delivery maximizes the risk of anchoring bias (the context is always “priming” the model), while active delivery introduces a selection step that may reduce entrenchment—the session must formulate a query, which requires some metacognitive awareness of what it does not know. Epley and Gilovich (?) showed that anchoring operates through different mechanisms for externally provided vs. self-generated anchors, further supporting the prediction that delivery mode matters independently of content.

This 2D space yields five distinct session configurations. The current implementation supports Deep (full/passive), Fresh (none), and three Semi-Fresh variants (Passive, Active, Selective) evaluated in §??.

3.2 Architecture

MCP client sessions connect to a single Ploidy server via Streamable HTTP:

- **Deep session:** Full project context (codebase, history, prior decisions)
- **Fresh session:** Only the raw artifact under review (code snippet, architecture question)
- **Semi-Fresh session:** Deep’s POSITION output, compressed by a summarization step, as the only context

The server maintains debate state in SQLite (WAL mode) and enforces the phase protocol.

3.3 Ploidy Level

The ploidy level n determines how many sessions are spawned per context depth. At ploidy 1n (haploid), the system runs Deep(1)×Fresh(1)—the minimal asymmetric debate. At ploidy 2n (diploid), it runs Deep(2)×Fresh(2), adding one stochastic backup per depth.

Table 2: Ploidy levels and their properties.

Ploidy	Sessions	Event A/B	Causal attribution
1n (haploid)	Deep×1, Fresh×1	Confounded	Cannot distinguish
2n (diploid)	Deep×2, Fresh×2	Separable	Within-group → Event B
3n (triploid)	Deep×3, Fresh×3	Separable	Majority voting within groups
4n+	Deep×N, Fresh×N	Separable	Statistical significance

Table 3: Context injection modes.

Mode	Description
<code>raw</code>	Plain text in user prompt (baseline)
<code>memory</code>	Accumulated observations as numbered items (CLAUDE.md style)
<code>skills</code>	Declarative rules and constraints (skills.md style)
<code>system_prompt</code>	Context via system-level instruction
<code>claude_md</code>	XML-tagged project instructions

At 1n, the system cannot distinguish whether a disagreement between Deep and Fresh is caused by context asymmetry (Event A) or stochastic variance (Event B). At 2n+, within-group disagreement isolates Event B (both Deep sessions had identical context, so any disagreement is stochastic), while between-group disagreement isolates Event A. The ploidy level trades compute cost for causal interpretability.

3.4 Context Injection Modes

The same context information can be delivered to the Deep session in different formats, each creating different anchoring dynamics:

The hypothesis is that narrative context (memory) creates stronger anchoring priors than declarative context (skills), because the model treats “learned observations” as internalized beliefs rather than external constraints.

3.5 Debate Phases

1. **Position:** All sessions independently analyze the artifact. Each produces a list of findings with confidence levels.
2. **Challenge:** Each session reviews the others’ positions. For each finding, the reviewer responds with a typed semantic action:
 - **agree**—finding is valid
 - **challenge**—finding is wrong or misleading, with explanation
 - **propose_alternative**—finding is partially right, here’s a different framing
 - **synthesize**—combining both perspectives into a stronger finding
3. **Convergence:** The protocol analyzes the debate transcript and classifies outcomes:
 - **Agreements:** findings confirmed by multiple sessions
 - **Productive disagreements:** findings where challenge/synthesis produced new insight

- **Irreducible disagreements:** genuine differences that could not be resolved
- **Confidence score:** proportion of agreed findings

3.6 Design Principles

- **No shared memory:** Fresh/Semi-Fresh sessions never see Deep’s raw analysis outside the debate protocol.
- **Typed actions over free-form:** Semantic actions make the debate transcript machine-interpretable.
- **Disagreement as signal:** Irreducible disagreements are informative, not failures—they mark where context mattered.

4 Implementation

Ploidy is implemented as a Python MCP server (FastMCP, asyncio, aiosqlite) exposing 11 tools over Streamable HTTP on port 8765. The server manages debate lifecycle, enforces phase transitions via a finite state machine (INDEPENDENT → POSITION → CHALLENGE → CONVERGENCE → COMPLETE), and persists all state in SQLite with WAL mode and foreign key cascades for crash recovery. Per-debate asyncio locks guard concurrent phase transitions; API calls retry with exponential backoff on transient failures.

v0.3 enhancements. Building on v0.2 (Semi-Fresh sessions, API fallback, LLM convergence, dashboard), v0.3 adds:

- **Cross-model validation:** Experiment runner supporting 4 backends (Anthropic, OpenAI, Ollama/Codex, Gemini) with Stochastic-N baselines for Event A/B isolation (§??).
- **Human-in-the-loop** (`debate_review`): The `debate_auto` tool now accepts a `pause_at` parameter (“challenge” or “convergence”) that pauses the automated debate for human review. The reviewer can approve, override (replacing auto-generated content), or reject the debate via the new `debate_review` tool—closing the gap between fully-automated and fully-manual workflows.
- **Effort level:** Configurable reasoning depth parameter that controls how much computation the model applies per response—an experimental variable that may interact with context asymmetry (§??).
- **Sweep infrastructure:** Ploidy-level (1n–4n), effort (low/high/max), injection mode (raw/memory/skills/system), and language (en/ko/ja/zh) sweeps with 25 extended tasks.

We note that the experiments in §?? use the `claude --print` CLI to simulate the protocol rather than the MCP server directly, as this provides cleaner session isolation for controlled comparison. Each CLI invocation creates a guaranteed-fresh session with no shared state. The MCP server is designed for production use where two human-operated terminals connect to the same server instance.

Full source: <https://github.com/heznpc/ploidy>

5 Experiments

This section reports preliminary pilot results. All findings should be interpreted as observations from a small-scale pilot study, not as statistically validated conclusions. We report both null and positive observations to bound where context asymmetry applies.

5.1 Setup

We evaluate on 10 tasks across two context regimes:

- **Experiment 1** (short context, ~ 50 tokens): 5 code review tasks with injected bugs + 2 architecture decision tasks. Each has 3–5 ground-truth issues.
- **Experiment 2** (long context, 2,000–5,000 tokens): 3 architecture decision tasks with project histories containing anchoring-inducing biases. Each has 5–6 ground-truth issues.

Methods (all using Claude Opus 4.6 via `claude --print`, each invocation = fresh session):

1. **Single Session**: One session with full context.
2. **Independent Second Opinion**: Two sessions with full context, responses concatenated.
3. **CCR (Unidirectional)**: Deep session produces analysis; Fresh session reviews it.
4. **Symmetric Debate**: Two sessions with identical full context debate each other.
5. **Ploidy (Asymmetric Debate)**: Deep (full context) vs Fresh (zero context), structured protocol.
6. **Self-Consistency (5-vote)**: Five independent single sessions, majority-vote synthesis. Approximately equal token budget to Ploidy (~ 5 LLM calls).
7. **Semi-Fresh-Passive**: Deep session produces analysis; compressed summary injected directly into Semi-Fresh session’s prompt.
8. **Semi-Fresh-Active**: Deep session produces analysis; compressed summary available to Semi-Fresh session but only after independent assessment. Session must form its own view first, then consult prior analysis.
9. **Semi-Fresh-Selective**: Deep session produces analysis; only failure/uncertainty information extracted and provided to Semi-Fresh session.

Judge. Claude Opus 4.6 evaluates each method’s output against ground truth. For each ground-truth issue: FOUND (1.0), PARTIAL (0.5), or MISSED (0.0). Additional valid findings not in ground truth are counted separately as bonus findings.

Metrics. $\text{Recall} = (\text{found} + 0.5 \times \text{partial}) / \text{total ground truth}$. Precision and F1 are reported but include bonus findings in the denominator, which penalizes methods that produce more valid-but-unlisted findings. We flag this as a metric design issue (§??) and recommend interpreting recall as the primary indicator of ground-truth detection.

Table 4: Experiment 1 results: short-context tasks (~ 50 tokens). All 9 methods achieve near-perfect recall.

Method	Avg F1	Avg Recall	Avg Time
Single Session	0.573	3.7/4.1	40s
Second Opinion	0.554	4.1/4.1	89s
Symmetric Debate	0.555	4.0/4.1	118s
CCR (Unidirectional)	0.548	3.9/4.1	92s
Ploidy (Asymmetric)	0.540	3.9/4.1	205s
Semi-Fresh-Active	0.538	3.9/4.1	117s
Semi-Fresh-Passive	0.535	3.9/4.1	122s
Semi-Fresh-Selective	0.533	4.0/4.1	130s
Self-Consistency (5-vote)	0.529	3.7/4.1	234s

Effort level. Claude Code exposes an **effort** parameter (low, medium, high, max) that controls the model’s reasoning depth per response. All experiments in this paper use the default **high** effort unless otherwise noted. We identify effort level as a potentially confounding variable: lower effort may reduce a session’s ability to detect subtle issues, while higher effort may increase the quality of challenges. In §??, we describe a factorial experiment design crossing effort levels with methods to measure this interaction.

Limitations of this setup (expanded in §??): single run per method-task pair, author-defined ground truth without independent validation, same model family as judge, CLI simulation rather than MCP server execution, and single effort level.

5.2 Results (Experiment 1: Short-Context Tasks)

7 tasks (5 code review + 2 architecture), Claude Opus 4.6, single run per method.

Observation: No method consistently outperforms Single Session on these tasks. All 9 methods achieve near-perfect recall (90–98%), and F1 differences are driven by precision (bonus findings inflating denominators). Critically, the delivery mode effect observed in Experiment 2 (SF-Active vs SF-Passive) disappears entirely here: both achieve identical 3.9/4.1 recall. This confirms that context asymmetry and delivery mode provide no benefit when context is too short for entrenchment to occur.

5.3 Analysis: Why Context Asymmetry Did Not Help

1. Insufficient context depth. Each task’s project context was ~ 50 tokens. Context entrenchment requires accumulated context on the order of thousands of tokens. At 50 tokens, the Deep session develops no meaningful anchoring bias.

2. Task difficulty ceiling. Claude Opus 4.6 found nearly all ground-truth issues in every method. When baseline recall is near-perfect, multi-session methods cannot demonstrate improvement.

3. Metric design issue. Our F1 formulation includes bonus findings (valid issues not in ground truth) in the precision denominator. This systematically penalizes more thorough methods. A revised metric should either exclude bonus findings from precision or report them as a separate axis. We retain the current formulation for transparency but caution against interpreting F1 differences smaller than the observed stochastic variance (± 0.10).

Table 5: Experiment 2 results (original 5 methods): long-context tasks (2,000–5,000 tokens).

Method	Avg F1	Avg Recall (Found/Total)	Avg Time
Symmetric Debate	0.607	5.0/5.3	146s
Single Session	0.591	4.3/5.3	52s
Second Opinion	0.566	4.3/5.3	108s
Ploidy (Asymmetric)	0.561	4.7/5.3	294s
CCR (Unidirectional)	0.458	4.7/5.3	108s

Table 6: Experiment 2 results (Semi-Fresh variants + Self-Consistency, same 3 tasks).

Method	Avg F1	Avg Recall (Found/Total)	Avg Time
Self-Consistency (5-vote)	0.607	5.3/5.3	292s
Semi-Fresh-Active	0.557	5.3/5.3	153s
Semi-Fresh-Passive	0.553	4.7/5.3	193s
Semi-Fresh-Selective	0.505	5.0/5.3	258s

5.4 Qualitative Observations

Despite quantitative parity, Ploidy’s convergence phase produced qualitatively distinct outputs:

- **Severity calibration:** Fresh session challenged Deep’s severity escalation of a memory leak, arguing it depends on key cardinality—a nuance absent from single-session output.
- **Novel findings:** Fresh identified that `get()` being `async` without `await` affects race condition exploitability—a finding neither session produced in isolation.
- **Explicit disagreement tracking:** Typed semantic actions create a machine-readable audit trail of how conclusions were reached, which no other method provides.

5.5 Experiment 2: Long-Context Tasks

To test whether context asymmetry matters when context is long enough to induce entrenchment, we designed 3 architecture decision tasks with 2,000–5,000 token project histories containing anchoring-inducing biases:

- **DB migration:** 18-month history of PostgreSQL commitment, repeated rejection of alternatives, team pride in custom partitioning
- **Auth overhaul:** 2-year history of custom auth built by one developer who defends it
- **Microservice split:** 3-year monolith with premature microservice extraction in progress

Each task’s context is designed so that a session anchored to the project history will rationalize the status quo. We acknowledge that this design creates a risk of circularity—context-free sessions are expected to be less anchored by definition (§??).

Per-task breakdown (selected methods):

Key observation on the DB migration task: SF-Active and SF-Passive received identical compressed information from the same Deep session analysis. SF-Active achieved 5/5 FOUND (matching Ploidy), while SF-Passive achieved only 3/5 FOUND (matching Single Session). The

Table 7: Combined ranking by recall (all 9 methods, Experiment 2).

Method	Avg Recall	Avg F1	LLM Calls
Semi-Fresh-Active	5.3 /5.3	0.557	~4
Self-Consistency	5.3 /5.3	0.607	~6
SF-Passive+Bottom (ablation)	5.3 /5.3	0.619	~4
Semi-Fresh-Selective	5.0/5.3	0.505	~4
Symmetric Debate	5.0/5.3	0.607	3
SF-Passive+Independent (ablation)	5.0/5.3	0.550	~4
Semi-Fresh-Passive	4.7/5.3	0.553	~4
Ploidy (Asymmetric)	4.7/5.3	0.561	5
CCR (Unidirectional)	4.7/5.3	0.458	2
Single Session	4.3/5.3	0.591	1
Second Opinion	4.3/5.3	0.566	2

Table 8: Per-task recall for Experiment 2. GT = number of ground-truth issues. F = FOUND, P = PARTIAL.

Task	GT	Single	Ploidy	SF-Pass.	SF-Active	SF-Sel.	Self-Con.
DB migration	5	3F+2P	5F	3F+2P	5F	4F+1P	5F
Auth overhaul	5	3-4F	3-4F	5F	5F	5F	5F
Microservice split	6	4-5F	5-6F	6F	6F	6F	6F

sole difference is delivery mode: passive embedding vs. active retrieval after independent assessment. This is the strongest evidence in our pilot data that **how context is delivered matters independently of what context is delivered**.

Note on F1: Ploidy’s F1 on DB migration (0.500) is lower than Symmetric’s (0.600) despite Ploidy achieving 5/5 FOUND vs Symmetric’s 4F+1P, because Ploidy generated more bonus findings (10 vs 5). On recall alone—the metric we argue better captures decision quality—the pattern is clearer.

5.6 Observations Across Both Experiments

Observation 1: Recall gap widens with context length. In Experiment 1, all methods achieve near-perfect recall (no gap). In Experiment 2, the gap between best (SF-Active, Self-Consistency: 100%) and worst (Single Session: 81%) is substantial. Context asymmetry provides no benefit when entrenchment does not occur.

Observation 2: Both delivery mode and prompting strategy contribute to recall, as shown by ablation. SF-Active and SF-Passive received identical compressed summaries but achieved 100% vs 89% recall. To disentangle delivery mode from the “independent-first” instruction present in SF-Active but absent from SF-Passive, we ran an ablation: SF-Passive+Independent, which adds the instruction to the passive delivery format (summary at top of prompt).

Table 9: Factorial ablation of summary position vs. independent instruction on Experiment 2.

Variant	Summary Position	Independent Instruction	Avg Recall
SF-Passive	Top	No	4.7/5.3 (89%)
SF-Passive+Independent	Top	Yes	5.0/5.3 (94%)
SF-Passive+Bottom	Bottom	No	5.3/5.3 (100%)
SF-Active	Bottom	Yes	5.3/5.3 (100%)

The ablation reveals that **information position is the dominant factor**. Moving the summary from the top to the bottom of the prompt—with no other changes—improves recall from 89% to 100% (+11pp). The “independent-first” instruction contributes +5pp only when the summary is at the top (partially counteracting primacy anchoring), but has no additional effect when the summary is at the bottom (already at ceiling). On the DB migration task, the gradient is $3/5 \rightarrow 4/5 \rightarrow 5/5 \rightarrow 5/5$ across the four conditions.

What we initially characterized as a “delivery mode” effect is more precisely a **primacy anchoring effect**: information placed at the beginning of a prompt has a stronger anchoring influence on the model’s reasoning than information placed at the end. This is consistent with the well-established primacy effect in human cognition and with Epley and Gilovich’s (?) finding that externally provided anchors (here, a summary that “frames” all subsequent analysis) produce stronger anchoring than information encountered after independent judgment formation. The cognitive science literature predicts this: the generation effect (?) and anchoring mechanism differences for externally-provided vs. self-generated information (?) both support that information position and retrieval mode affect processing depth independently of content.

Observation 3: Self-Consistency is a strong budget-equivalent baseline. At approximately equal token budget (~ 5 – 6 LLM calls), Self-Consistency achieves the same 100% recall as SF-Active. However, Self-Consistency requires approximately twice the wall-clock time (292s vs 153s) and does not produce structured debate output (typed actions, disagreement tracking, convergence analysis). The qualitative value of structured debate remains distinct.

Observation 4: Semi-Fresh-Selective outperforms Semi-Fresh-Passive. Providing only failure/uncertainty information (SF-Selective: 94% recall) outperforms providing the full compressed summary passively (SF-Passive: 89%). This suggests that negative knowledge (“what failed”) is more useful than positive knowledge (“what was found”) for maintaining independence—consistent with the “selective forgetting” step in the restart mechanism (§??).

Observation 5: F1 is unstable for multi-phase methods. Re-run analysis shows Ploidy’s F1 varying by 0.106 across runs while recall remained stable. This variance comes entirely from bonus findings count, confirming that recall is the more stable measure.

5.7 Stochastic Variance (Re-run Analysis)

We re-ran Experiment 2 to measure stochastic variance. The signal-to-noise ratio is concerning: the largest observed method difference in F1 (0.03) is smaller than the within-method run-to-run variance (0.106). This means the F1 rankings in §?? are not stable across runs.

Table 10: Re-run analysis for the DB migration task.

Method	Run 1 Found	Run 2 Found	Run 1 F1	Run 2 F1
Single	3F+2P	4F+1P	.571	.643
Ploidy	5F	5F	.500	.556

DB migration task (2 runs): Ploidy’s recall on this task was deterministic (5/5 in both runs) while Single’s was stochastic (3–4/5). This stability, rather than the absolute F1 value, is the most noteworthy observation from these pilot experiments.

Execution time variance. Stochastic variance manifests not only in output quality but in wall-clock execution time. Across 170 multi-run entries on Claude Opus 4.6 (same model, same task, same method, same effort level), elapsed time varied by up to $3.3\times$ between runs.

Table 11: Execution time variance across re-runs (Claude Opus 4.6, high effort, long-context tasks). Ratio = max/min elapsed time across runs.

Task	Method	Runs	Min (s)	Max (s)	Ratio
Auth overhaul	Single	8	46	51	$1.1\times$
DB migration	Single	7	46	59	$1.3\times$
Microservice	Single	5	59	76	$1.3\times$
Auth overhaul	Ploidy 1n	9	205	584	$2.9\times$
DB migration	Ploidy 1n	10	193	273	$1.4\times$
Microservice	Ploidy 1n	7	181	605	$3.3\times$
Auth overhaul	Ploidy 4n	5	472	970	$2.1\times$
Microservice	Ploidy 4n	4	535	1184	$2.2\times$

Single Session wall-clock time is stable ($1.1\text{--}1.3\times$ ratio across runs), reflecting the deterministic overhead of a single LLM call. Ploidy’s time variance is substantially higher (up to $3.3\times$), because multi-session methods amplify per-session stochastic variation: each session’s response length, reasoning depth, and challenge complexity vary independently, and these variances compound. This confirms that stochastic variance affects not only *what* the model outputs but *how long* it takes—a user running the same analysis twice may experience dramatically different wall-clock costs depending on which stochastic samples land.

5.8 Experiment 3: Injection Mode Sweep

We tested whether the *form* of context delivery affects debate outcomes independently of content. Three injection modes (raw, memory, skills) were crossed with three methods (Single, Ploidy, CCR) on 3 long-context tasks.

Table 12: Recall by injection mode and method (long-context tasks, avg over 3 tasks).

Mode	Single	Ploidy	CCR
raw	5.0/5.3	5.3/5.3	4.7/5.3
memory	4.0/5.3	5.0/5.3	4.0/5.3
skills	4.3/5.3	5.3/5.3	5.0/5.3

Ploidy achieves highest recall in all three injection modes. Memory-style injection causes the largest recall degradation: Single drops 20% (5.0→4.0), CCR drops 15% (4.7→4.0), while Ploidy drops only 6% (5.3→5.0). The Fresh session’s independence from the memory-formatted context provides resilience against injection-induced anchoring.

Notably, injection mode changes which method achieves the best F1: raw favors Ploidy (0.573), memory favors Single (0.573), skills favors CCR (0.606). Context injection mechanism is therefore a **moderator variable** for debate efficacy.

5.9 Experiment 4: Ploidy Level Sweep (1n–4n)

We swept ploidy levels 1n through 4n on 3 long-context tasks across 4 model families (Claude Opus 4.6, Claude Sonnet 4.6, Gemini 3.1 Pro, GPT-5.4 via Codex CLI). To isolate Event A (context asymmetry) from Event B (stochastic variance), we introduce a **Stochastic-N** baseline: N independent sessions all receiving full context, with N matched to the total session count of Ploidy at the same level. The difference between Ploidy and Stochastic-N measures the pure contribution of context asymmetry.

Table 13: Ploidy vs Stochastic-N: isolating Event A (context asymmetry). Δ = Ploidy F1 – Stochastic-N F1. Positive values indicate context asymmetry contributes beyond stochastic sampling. — = no Stochastic-N baseline at 1n (equivalent to Single). [†] = incomplete task coverage (1–2 of 3 tasks).

Model	Ploidy	Single F1	Stoch-N F1	Ploidy F1	$\Delta(\text{P-St})$
4*Opus 4.6	1n	0.590	0.535	0.561	+0.025
	2n	0.582	0.359	0.538	+0.179
	3n	0.662	0.512	0.533	+0.022
	4n	0.624	0.519	0.560	+0.042
4*Sonet 4.6	1n	0.490	—	0.389	—
	2n	0.457	0.500	0.218	−0.282
	3n	0.383	0.377	0.377	−0.001
	4n	0.493	0.450	0.455	+0.005
3*Codex/GPT-5.4	2n	0.387	0.258	0.450	+0.192
	3n	0.261	0.349	0.375	+0.026
	4n [†]	0.286	0.347	0.500	+0.153
4*Gemini 3.1 Pro	1n	0.452	—	0.499	—
	2n	0.613	0.622	0.547	−0.075
	3n	0.595	0.575	0.530	−0.045
	4n	0.615	0.444	0.500	+0.056

Event A isolation. On Opus, Ploidy outperforms Stochastic-N at *every* ploidy level (4/4), with the strongest effect at 2n (Δ =+0.179). Codex shows the same pattern across all three tested levels (3/3 positive, peak Δ =+0.192 at 2n). This confirms that context asymmetry contributes independently of stochastic sampling—running more sessions with identical context does not replicate the benefit of running sessions with *different* context depths.

Capability threshold. On Sonnet, Δ is strongly negative at 2n (−0.282) and near zero at 3n–4n. At 1n (no Stochastic-N baseline), Ploidy F1 (0.389) is already below Single (0.490), consistent with the Fresh session injecting noise at this capability level. Gemini shows a similar pattern:

negative at 2n–3n, turning positive only at 4n; at 1n, Ploidy (0.499) slightly outperforms Single (0.452) but without a Stochastic-N baseline the contribution of context asymmetry vs. stochastic variance cannot be separated. This supports the capability threshold hypothesis (§??): below a certain model capability, Fresh sessions generate noise rather than constructive challenge, and context asymmetry is counterproductive.

Optimal ploidy level. 2n remains the cost-optimal level for the strongest models. Beyond 2n, recall plateaus while token cost scales linearly. The 2n advantage is most pronounced on Opus ($\Delta=+0.179$) and Codex ($\Delta=+0.192$). Notably, Codex maintains positive Δ through 4n (+0.153), suggesting that for some model families, higher ploidy continues to provide benefit—potentially because the model’s stochastic variance is higher and requires more samples to stabilize.

5.10 Experiment 5: Cross-Model Validation (4 Families)

To test whether findings generalize across model families, we ran the ploidy sweep (1n–4n) with Stochastic-N baseline across 4 model families: Claude Opus 4.6, Claude Sonnet 4.6, Gemini 3.1 Pro, and GPT-5.4 (via Codex CLI). Stochastic-N baselines at 1n are unavailable for Sonnet and Gemini (1n Stochastic-N is equivalent to Single). Results are integrated into Table ??.

The cross-family results reveal a clear pattern: **context asymmetry benefits scale with model capability**.

- **Opus** (strongest): Ploidy $>$ Stochastic-N at all 4 ploidy levels. Event A consistently positive.
- **Codex/GPT-5.4**: Ploidy $>$ Stochastic-N at all three tested levels—2n (+0.192), 3n (+0.026), 4n (+0.153). Second strongest effect, with sustained benefit at higher ploidy.
- **Gemini 3.1 Pro**: Mixed. Negative at 2n–3n, positive only at 4n (+0.056). Intermediate capability.
- **Sonnet** (weakest on abstract tasks): Ploidy \leq Stochastic-N at 2n–3n. Fresh sessions lack reasoning capacity to generate useful challenges, injecting noise rather than constructive critique.

This validates the capability threshold hypothesis: context-asymmetric debate is a **high-capability intervention**. Below the threshold, Fresh sessions produce generic textbook challenges that the convergence phase cannot productively integrate. The threshold appears to be task-dependent—Sonnet performs adequately on concrete technical tasks (auth overhaul: 5/5 recall) but fails on abstract socio-technical judgment (DB migration: 2/5 recall).

5.11 Experiment 6: Effort Level Sweep

We swept effort levels (low, high, max) on 2 long-context tasks with Single and Ploidy 1n to test effort–method interaction.

Table 14: Effort level sweep results (avg over 2 long-context tasks).

Effort	Single Recall	Ploidy Recall	Single F1	Ploidy F1
low	4.5/5	4.5/5	0.679	0.544
high	4.5/5	4.0/5	0.634	0.650
max	3.5/5	3.0/5	0.475	0.482

Effort is not monotonically beneficial. The averaged results show max effort producing the lowest recall for both methods (Single 3.5/5, Ploidy 3.0/5). However, per-task analysis reveals high variance: on individual runs, Single at max effort achieved 5/5 on some tasks—the highest recall observed for Single in this sweep. The averaged degradation at max effort is driven by specific task–effort interactions rather than a uniform overthinking effect. Effort=low matches effort=high on recall (4.5/5) at roughly half the compute time. Ploidy outperforms Single on F1 only at effort=high, suggesting the effort \times method interaction is non-trivial but insufficiently sampled: single-run averages over 2 tasks cannot distinguish genuine interaction effects from stochastic variance.

6 Discussion

6.1 Context Asymmetry Threshold Hypothesis

Our experiments suggest context asymmetry is a *conditional* intervention whose benefit depends on context length crossing an entrenchment threshold. We formalize this as the **Context Asymmetry Threshold Hypothesis**:

Table 15: Observed benefit of context asymmetry by context regime.

Context Regime	Tokens	Benefit?	Mechanism
Short-context	<100	No	No entrenchment occurs; nothing to debias
Unknown zone	100–2,000	?	Threshold likely falls here
Long-context + anchoring	2,000+	Yes	Sunk cost, authority bias, trajectory locking

The null result on short-context tasks (Experiment 1) is not a failure of the method—it is a *boundary condition* that makes the positive results on long-context tasks more credible. If Ploidy showed benefit everywhere, the signal would be less interpretable. The selective activation pattern is consistent with Young (?)’s phase transition theory: debate value scales with knowledge divergence, and short-context tasks produce no divergence to exploit. Identifying the exact threshold requires a systematic context-length gradient experiment (100, 500, 1K, 5K, 10K tokens).

6.2 Minimum Capability Threshold

The cross-model experiment (§??) reveals that context asymmetry is upper-bounded by the base model’s reasoning capacity. This bound is *task-dependent*: a task-abstraction gradient separates findings into concrete technical issues (identifiable by smaller models) and abstract socio-technical judgments (requiring frontier-class reasoning).

When the base model falls below the capability threshold for a task class, three failure modes emerge: (1) the Fresh session cannot generate substantive challenges, reducing the debate to noise; (2) the Deep session cannot articulate abstract biases even when they exist in its context; (3) the convergence phase cannot synthesize meaningfully from weak inputs. In the worst case, Ploidy underperforms Single—the Fresh session’s uninformed challenges actively degrade the Deep session’s partially correct analysis.

This has practical implications: Ploidy should not be deployed with models that cannot independently identify the class of findings being sought. A pre-flight capability check—running a calibration task to verify the model can reason about the target finding class—may be warranted before committing to multi-session compute.

6.3 The Semi-Fresh Hypothesis

Our current design tests only the extremes of the Context Asymmetry Spectrum: full context (Deep) vs. zero context (Fresh). This leaves an important region unexplored—one that may contain the optimal operating point.

Consider a common human practice: when stuck on a problem, practitioners often restart from scratch—but they carry a compressed memory of what was tried and what failed. This behavior decomposes into four cognitive steps:

1. **Compression:** The full work history is distilled to “what was tried, what failed, and why.”
2. **Selective forgetting:** Implementation details and dead-end reasoning are discarded, reducing context volume.
3. **Restart:** The problem is approached from a new angle, unencumbered by accumulated commitments.
4. **Implicit constraint:** The compressed memory of prior failures prevents re-exploring known dead ends.

Each of these steps has established cognitive science parallels: compression maps to schema formation in reconstructive memory (?); selective forgetting corresponds to directed forgetting, which has been shown to facilitate creative problem-solving by releasing functional fixedness (?); restart parallels the incubation effect, where interruption of conscious processing improves subsequent performance (?); and the implicit constraint from prior failures mirrors the generation effect, where actively produced knowledge is more deeply encoded than passively received information (?).

The practitioner who restarts this way is neither an expert entrenched in the problem (Deep) nor a complete novice (Fresh). They are **Semi-Fresh**: equipped with a structured digest of prior attempts but freed from the accumulated context that caused entrenchment. Notably, step 2 is what distinguishes this from simply continuing—selective forgetting breaks the anchoring chain while step 4 preserves the informational value of prior work.

We hypothesize that a Semi-Fresh session—receiving only a compressed summary of the Deep session’s analysis (e.g., “approaches attempted, constraints identified, failures encountered”) rather than the full project context—may outperform both extremes:

- Better than Fresh: knows which approaches failed, understands domain constraints
- Better than Deep: not anchored to the narrative that induced entrenchment

Furthermore, the **delivery mode** of this compressed context may itself be a significant variable. We propose three Semi-Fresh variants:

1. **Semi-Fresh-Passive:** Compressed summary is injected directly into the prompt. The session always “sees” the prior analysis, analogous to a briefed outsider who has read the executive summary before entering the room.
2. **Semi-Fresh-Active:** Compressed summary is available via an explicit tool call rather than embedded in the prompt. The session must actively choose to consult prior work (e.g., calling `get_prior_analysis()`), analogous to a consultant who has access to project files but forms an independent assessment first.



Figure 3: Exploration of the Semi-Fresh region in the depth \times delivery space.

3. **Semi-Fresh-Selective:** Only failure information is provided (“These approaches were tried and failed: ...”), excluding successful analyses. This tests whether negative knowledge (what not to do) is more valuable than positive knowledge (what was found) for breaking entrenchment.

If Semi-Fresh-Active outperforms both Fresh and Deep, it would suggest that the optimal verification partner is neither ignorant nor entrenched, but **selectively informed with retrieval autonomy**—a finding with direct implications for how multi-agent systems should manage shared knowledge. If the Active variant outperforms Passive with identical information, it would demonstrate that **how context is delivered matters independently of what context is delivered**—a novel result absent from the existing MAD literature.

Preliminary results (§??). We implemented and evaluated all three Semi-Fresh variants on the long-context tasks. The results provide initial support for the delivery mode hypothesis:

- SF-Active achieved 100% average recall (tied with Self-Consistency for best), while SF-Passive achieved 89% (tied with Ploidy). The sole difference is delivery mode.
- On the most bias-laden task (DB migration), SF-Active found 5/5 while SF-Passive found only 3/5—with identical compressed information.
- SF-Selective (94%) outperformed SF-Passive (89%), suggesting negative knowledge is more effective than full summaries for maintaining independence.

These observations are from a single run on 3 tasks and require statistical validation, but they shift the question from “does asymmetry help?” to “what is the optimal point in the depth \times delivery space?”—a quantitatively richer and more practically useful direction.

6.4 Effort Level as Experimental Variable

LLM inference providers increasingly expose **effort** or **thinking budget** parameters that control how much computation a model applies to each response. In Claude Code, this manifests as four levels: **low**, **medium**, **high**, and **max**. This parameter is relevant to context-asymmetric debate for three reasons:

1. **Effort–asymmetry interaction:** Higher effort may allow a Deep session to reason through its biases more carefully, potentially reducing the benefit of Fresh perspective. Conversely, higher effort in a Fresh session may compensate for missing context through deeper first-principles reasoning.
2. **Effort as confound:** If experiments are conducted at a single effort level, observed effects may not generalize. A method that outperforms at **high** effort may underperform at **low** effort if the method’s benefit depends on reasoning depth.

3. **Cost–quality trade-off:** Lower effort is cheaper. If asymmetric debate at **medium** effort achieves the same recall as single session at **max** effort, this has practical cost implications.

We propose a $4 \times k$ factorial design crossing effort levels (**low**, **medium**, **high**, **max**) with k selected methods (e.g., Single, Ploidy, SF-Active) on the long-context tasks. The experiment runner supports this via `--effort-sweep`:

```
python experiments/run_experiment.py --long --effort-sweep \
    --methods single,ploidy,sf_active
```

Key hypotheses to test:

- H_1 : Effort level has a main effect on recall (higher effort \rightarrow higher recall).
- H_2 : There is an effort \times method interaction—context asymmetry provides greater benefit at lower effort levels, where single sessions are more susceptible to anchoring.
- H_3 : The optimal cost–quality point is asymmetric debate at medium effort, not single session at max effort.

Preliminary pilot results on the DB migration task (single run each) suggest effort matters:

Table 16: Effort level pilot results (DB migration task, single run per cell).

Effort	Single Session	Ploidy	SF-Active
Low	4F+1P	4F+1P	—
High	3F+2P	4F+1P	5F
Max	5F	4F+1P	—

At **max** effort, Single Session achieves 5/5 (matching SF-Active at **high**), while Ploidy remains at 4F+1P across all effort levels. This is consistent with H_2 : context asymmetry provides greater benefit at moderate effort, while at maximum effort a single session may overcome anchoring through deeper reasoning alone. These are single-run observations subject to stochastic variance.

6.5 Linguistic Framing as Experimental Variable

When LLMs generate responses localized for different languages, the social and cultural context injected into the language may distort the *substance* of technical findings. This is a form of information loss distinct from context anchoring but potentially compounding it.

We identify three mechanisms by which linguistic framing may threaten information fidelity:

1. **Euphemization:** Languages with strong politeness norms (Korean, Japanese) may soften critical findings. “This code has a critical SQL injection vulnerability” becomes “There may be room for improvement in input handling”—the severity signal is lost.
2. **Hierarchical suppression:** In languages with grammatical honorifics (Korean’s *jondaenmal/banmal*, Japanese’s *keigo*), challenges to established positions may be linguistically weakened. A Fresh session challenging a Deep session’s position may unconsciously soften its critique when generating in Korean, reducing the effectiveness of the adversarial signal.
3. **Technical term dilution:** When technical terms are translated rather than preserved (e.g., “race condition” \rightarrow “ ”), the precision of the finding may decrease, and the judge may have difficulty matching translated findings to English-language ground truth.

This extends the Context Asymmetry Spectrum to a **third dimension**: depth \times delivery mode \times linguistic framing. We propose a language sweep experiment:

```
python experiments/run_experiment.py --long --lang-sweep \
  --methods single,ploidy --langs en,ko,ja,zh
```

Key hypotheses:

- H_1 : Localized responses achieve lower recall on English-language ground truth due to euphemization and term dilution.
- H_2 : The recall gap between languages is larger for Fresh sessions (which lack context to anchor technical vocabulary) than Deep sessions.
- H_3 : Context asymmetry may *partially compensate* for localization loss—if the Fresh session in language L misses an issue due to euphemization, the Deep session (with stronger technical framing from project context) may catch it during the challenge phase.

Preliminary pilot results on the DB migration task (Korean vs. English, high effort, single run each):

Table 17: Language pilot results (DB migration task, single run per cell).

Language	Single Session	Ploidy	SF-Active
English	3F+2P	4F+1P	5F
Korean	5F	4F+1P	2F+3P

Extended across all three long-context tasks:

Table 18: Cross-linguistic results (high effort, single run per cell). F = FOUND, P = PARTIAL.

Task	GT	Single (en)	Single (ko)	Ploidy (en)	SF-Active (en)	SF-Active (ko)
DB migration	5	3F+2P	5F	4F+1P	5F	2F+3P
Auth overhaul	5	5F	4F+1P	4F+1P	4–5F	5F
Microservice	6	5F+1P	—	6F	6F	6F

SF-Active maintains perfect recall in Korean on 2 of 3 tasks (auth, microservice) but drops sharply on DB migration (5F \rightarrow 2F+3P, -30pp). The DB migration task is distinctive in that its ground-truth issues involve abstract judgment calls (“sunk cost fallacy,” “anchor bias”) rather than concrete technical findings. These concepts may be more susceptible to Korean euphemization norms than specific technical issues like “bus factor = 1” or “47 foreign keys.” This suggests H_1 (euphemization) is **task-dependent**: linguistic framing affects recall primarily on issues requiring normative judgment rather than technical identification. Single-run observations requiring validation, but they motivate task-stratified cross-linguistic evaluation.

6.6 Limitations

This pilot study has significant limitations that bound the strength of all claims:

- **Statistical power**: 7+3 tasks, single run per method-task pair (re-run on Exp 2 only). Observed method differences ($F1 \Delta \approx 0.03$) are smaller than within-method variance ($F1$

$\Delta \approx 0.10$). No statistical tests are reported because the sample size and run count cannot support them. Minimum requirement for validated claims: 30+ tasks, 5+ runs, paired statistical tests (Wilcoxon signed-rank or bootstrap CI).

- **Author-defined ground truth:** All ground-truth issues were defined by the authors without independent expert validation. Bonus findings identified by the judge suggest the ground truth is incomplete. Future work should use independently validated benchmarks or multiple expert annotators with inter-rater agreement metrics.
- **Limited runs per model family:** Cross-family validation now covers 4 families (Claude Opus/Sonnet, Gemini 3.1 Pro, GPT-5.4), but each family has only 3 tasks per ploidy level. The capability threshold is observed across families but its exact boundary and task-type dependence require larger-scale validation.
- **Same-model judge:** Claude Opus 4.6 generates outputs and evaluates them. Systematic bias is possible (e.g., preference for structured multi-phase outputs, or conversely, penalizing verbose responses). Cross-model judges (GPT-4, Gemini) and a human evaluation subset with Cohen’s kappa are needed.
- **Presentation bias in judge evaluation:** Different methods produce structurally different outputs—Single Session yields plain analysis, while Ploidy outputs contain debate markers (“Deep challenges Fresh,” “AGREE/SYNTHESIZE”). The judge may conflate output structure with content quality, scoring well-organized debate transcripts higher regardless of substance. Additionally, LLMs exhibit speech mirroring (style adaptation to input patterns), which may cause the judge to respond differently to debate-structured vs. plain-text outputs. Rigorous control would require normalizing all method outputs to a uniform format before evaluation, or separating issue extraction from scoring. Proper experimental design for this confound warrants consultation with linguists and LLM evaluation specialists.
- **Circular task design risk:** Long-context tasks were designed with anchoring biases whose detection is facilitated by context absence. A session without context is expected to be less biased by definition. Stronger validation requires externally-sourced tasks (e.g., real-world architecture decisions from open-source projects) where the relationship between context and bias is not author-designed.
- **Token cost:** Ploidy uses approximately $5\times$ the tokens of Single Session. Self-Consistency (5-vote) was included as a budget-equivalent baseline and achieves the same 100% recall as SF-Active on long-context tasks. The practical advantage of structured debate over Self-Consistency is the typed audit trail and convergence analysis, not recall improvement—a distinction that may or may not justify the added protocol complexity.
- **CLI simulation vs. MCP server:** Experiments use `claude --print` CLI rather than the Ploidy MCP server. While this provides cleaner session isolation, the convergence engine’s rule-based classification (in the server) is not exercised in the experiments.
- **Metric design:** The current F1 formulation penalizes valid bonus findings as false positives. This systematically disadvantages more thorough methods and should be revised in future work.
- **Effort sweep limited:** Effort sweep (§??) used 1n ploidy only. The effort \times ploidy \times injection three-way interaction is unmeasured.

- **English-only evaluation:** All experiments and ground truth are in English. Localized responses in other languages may exhibit different anchoring dynamics and information loss patterns (§??).

6.7 Compression Bias in Semi-Fresh Sessions

A reviewer may object that the compression step in Semi-Fresh sessions inherits the Deep session’s bias. If the Deep session has anchored on a sunk cost narrative, its summary will reflect that framing—omitting alternatives it dismissed and emphasizing paths it pursued. The Semi-Fresh session then inherits a pre-filtered view of the problem.

This concern is valid and motivates the ablation design: SF-Passive (summary injected upfront) vs. SF-Active (independent analysis first, summary consulted after). The experimental result that SF-Active outperforms SF-Passive is consistent with this objection—when the Semi-Fresh session forms an independent view before seeing the biased summary, the bias transmission is attenuated. A stronger mitigation would use an independent model for compression, or compress only failure information (SF-Selective), which by definition excludes the Deep session’s confident (and potentially entrenched) conclusions.

6.8 Epistemic Power Asymmetry in Challenge Phase

The Deep session holds an informational advantage during the Challenge phase: it can invoke project history, past decisions, and stakeholder constraints that the Fresh session cannot verify. A Deep session claiming “the CEO vetoed this approach last year” is unfalsifiable from the Fresh session’s zero-context position, creating an epistemic power imbalance that could suppress valid challenges.

Ploidy’s design addresses this structurally rather than procedurally: the convergence engine does not determine a “winner.” It classifies disagreements into agreement, productive disagreement, and **irreducible disagreement**—and presents all three to the human decision-maker. When the Deep session invokes unverifiable context to override a Fresh challenge, the convergence output surfaces this as an irreducible disagreement with an explicit context-attribution note, rather than silently accepting the Deep session’s authority. The human retains final judgment.

6.9 Justification Against Self-Consistency

Self-Consistency (5-vote) is a strong budget-equivalent baseline that achieves comparable recall to Ploidy on long-context tasks under raw injection. This demands an honest answer to the question: *why use a structured debate protocol when majority voting works?*

Ploidy’s comparative advantage is not recall—it is **structured context attribution**. Self-Consistency outputs a deduplicated list with vote counts: it answers *what* was found but not *why* sessions disagreed. Ploidy’s convergence output provides typed semantic actions (agree, challenge, synthesize) and explicit context attribution, enabling the user to determine whether a disagreement is context-driven (Event A: the Deep session’s history caused anchoring) or stochastic (Event B: random variance). This causal interpretability is unavailable from majority voting and is the primary value proposition for practitioners making high-stakes architectural decisions where understanding *why* matters as much as *what*.

Second, Self-Consistency is vulnerable to **correlated bias under non-raw injection**. When all 5 sessions receive the same memory-style context, they share identical anchoring priors. Our injection sweep shows memory-style context degrades Single recall by 20%—Self-Consistency, which runs 5 copies of Single with identical context, would inherit this degradation across all copies

simultaneously. Majority voting over 5 identically biased samples cannot correct a systematic blind spot. Ploidy’s Fresh sessions, having never seen the memory-formatted context, are structurally immune to this correlated failure mode.

6.10 Generic Noise from Zero-Context Sessions

A zero-context Fresh session may produce “textbook” challenges—correct in the abstract but irrelevant to the specific domain constraints the Deep session knows about. This wastes debate resources and may distract from genuine issues.

The cross-model experiment (§??) provides direct evidence: Sonnet’s Fresh session on the DB migration task produced challenges that degraded rather than improved the outcome (Ploidy 2.5/5 effective vs. Single 3.0/5). This failure mode is real and bounds the intervention’s applicability.

Three mitigations exist within the current framework. First, the ploidy level provides implicit filtering—at $2n$, if one Fresh session produces generic noise while the other produces substantive critique, the convergence phase can distinguish them by cross-referencing with the Deep sessions’ responses. Second, the Semi-Fresh variant addresses this directly by providing enough domain context to suppress purely generic challenges while preserving independence from the Deep session’s specific conclusions. Third, the minimum capability threshold (§??) should be enforced: if the model cannot reason about the task class at all, adding a zero-context session adds noise, not signal.

6.11 Broader Implications

For multi-agent system design, our preliminary results suggest a practical principle: **context diversity may be more valuable than agent count**. Combined with Choi et al.’s martingale result (?) (scaling homogeneous agents cannot improve expected correctness) and Boca et al.’s finding (?) that LLM populations spontaneously develop collective biases, this motivates architectures that deliberately maintain sessions with different context depths rather than scaling identical agents.

7 Conclusion and Future Work

We presented Ploidy, a protocol for structured debate between same-model sessions with intentional context asymmetry. We identified two independent phenomena—context asymmetry (Event A) and stochastic variance (Event B)—and showed that the ploidy level controls which can be distinguished. Four experiments reveal:

1. **Context asymmetry is a targeted intervention**, not a universal improvement. On short-context tasks, all methods achieve near-identical recall. The intervention applies only where entrenchment occurs.
2. **On long-context tasks**, Ploidy achieves highest recall across all context injection modes, with memory-style injection degrading Single recall by 20% while Ploidy drops only 6%.
3. **Context injection mechanism is a moderator variable**. The same information delivered as accumulated memories vs. declarative rules produces measurably different anchoring effects, changing which method performs best.

4. **Diploid (2n) is the cost-optimal ploidy level.** The second stochastic sample per context depth corrects the first sample’s blind spots at 30% additional compute. Higher ploidy provides no recall benefit.
5. **Context asymmetry requires a minimum model capability.** Cross-model validation (Opus vs Sonnet) reveals a task-abstraction gradient: concrete technical findings transfer across model sizes, but abstract socio-technical judgments require frontier-class reasoning. Below this threshold, Fresh sessions inject noise rather than correction.
6. **Effort level interacts non-trivially with method.** Averaged results suggest max effort degrades recall, but per-task variance is high—Single at max effort achieves perfect recall on some tasks. The effort \times method interaction requires larger-scale validation to distinguish from stochastic noise.

The intersection of intentional context asymmetry, structured cross-session debate, injection mechanism as experimental variable, and ploidy-level stochastic sampling has no prior publications as of March 2026.

Future work priorities:

1. **Statistical validation:** 30+ tasks, 5+ runs per condition, paired tests (Wilcoxon signed-rank, bootstrap CI)
2. **Cross-model evaluation:** Sonnet, Gemini, GPT, Ollama to test generalizability beyond Claude Opus 4.6
3. **Effort \times ploidy \times injection factorial:** full interaction analysis across all three variables
4. **Cross-linguistic evaluation:** en/ko/ja/zh to test whether cultural framing interacts with context asymmetry (§??)
5. **External task sets:** real-world architecture decisions from open-source projects
6. **Independent judge:** cross-model judges and human evaluation subset
7. **Multi-round protocol:** extending single-round CHALLENGE to test AceMAD’s submartingale conditions
8. **Asymmetric ploidy configurations:** the current design constrains Deep(n) \times Fresh(n) symmetrically. Context asymmetry (Event A) and stochastic variance (Event B) are independent phenomena requiring potentially different sample sizes—e.g., Deep(1) \times Fresh(3) to oversample the noisier distribution while minimizing redundant deep-context computation

We release Ploidy as an open-source MCP server at <https://github.com/heznpc/ploidy>.

Acknowledgments

This paper was written with the assistance of Claude Code (Anthropic, Claude Opus 4.6). The experimental framework, literature search, and draft editing were conducted through interactive sessions with the tool. All research decisions, hypotheses, and interpretations are the authors’ own.