Harness, Context, and Diversity:

A Short Review of Recent Evidence on Running LLM Agents

Mike Murunov Maksim Borisov

October 22, 2026 · Review preprint

Abstract

Practitioners building LLM agents use a shared vocabulary of terms such as *agent*, *harness*, *context*, *compaction*, and *effort*, but the terms are used inconsistently, and many working heuristics spread faster than the evidence behind them. This note sets out working definitions and then reviews 28 papers, most from 2025–2026, alongside three security incidents disclosed during the week of writing. The review asks which common heuristics have empirical support. Four findings stand out. First, the harness matters, and it matters most for models below the frontier. Second, long single sessions degrade, and *forgetting* (periodically restarting from the best candidate with a fresh context) beats running longer or running in parallel at equal compute. Third, LLM idea diversity is low by default. Persona prompting and verbalised sampling help at the start of a run, but closed-loop multi-agent systems still converge semantically. Fourth, compaction is lossy for future questions and is also a security boundary. The note ends with practical recommendations and a list of claims for which no supporting evidence was found.

---

1 Introduction

Over the past six months, agent tooling has changed faster than shared terminology. In discussions, the word *harness* can mean a prompt set, a tool set, or a whole orchestration layer, depending on the speaker. Some heuristics are repeated without sources: "clear the context at half the window", "more reasoning effort is always better", "a swarm of cheap models hallucinates more than one strong model". This note is an attempt to tie those heuristics to published evidence.

Scope and method. Papers were retrieved from arXiv by topic queries; most were submitted between late 2025 and October 2026. A few foundational references are included for context. Every quantitative statement below is taken from the cited paper's abstract. We did not reproduce any of the results. Statements marked *practitioner observation* are anecdotal and untested.

2 Working Definitions

Table 1 gives the definitions used throughout. The most useful compact formulation is **agent = model + harness**. Under this formulation the harness defines the agent's behaviour, and the model determines how well that behaviour is carried out. This matches the operational definition used in recent harness-synthesis work, where the harness is "the program that fixes which roles exist, how they pass information, which tools each may call, and how retries are coordinated" [1].

Table 1: Working definitions used in this note.

Term	Definition
Agent	An LLM that decides, inside a loop, which actions or tools to invoke and when to stop. A fixed script that calls tools is a *workflow*, not an agent.
Harness	Everything in the agent except the model: tools, prompts, context management, control-flow topology (loop or graph), and the protocol between agents. Multi-agent coordination is part of the harness.
ReAct loop	The interleaved reason → act → observe cycle [2]. It is still useful wherever each step has to be controllable and auditable.
MCP / A2A	MCP is a contract for invoking tools and services. A2A is a separate agent-to-agent protocol, now governed by the Linux Foundation, designed to *complement* MCP rather than to reimplement it [3].
Context	All tokens the model conditions on at a given step. It is the main resource the harness manages.
Compaction	Summarising earlier context and replacing the original with the summary.
Effort	A provider-exposed setting for reasoning depth. Its implementation is opaque to the user; its cost and latency are not.
Idea basin	A localised region of solution space in which an iterative search stalls [4].

3 The Harness Matters, Mostly Below the Frontier

A multi-model study of LLM-driven vulnerability discovery evaluated 24 model–runner configurations on a production codebase whose vulnerabilities were not public at scan time [5]. Of 265 confirmed root causes, the best two-run configuration covered 60.4%. A further 14.3% were found by exactly one model, and no single configuration found everything. Moving from a minimal prompt to an engineered prompt with a CodeQL-derived call graph raised criticality-weighted yield by 20–86% for small and mid-tier models, but by only 2% for the frontier model. Strong lower-cost models came close to frontier performance at about one-tenth of the cost. In a separate study, holding the model fixed and changing only the harness changed success rates several-fold. An automatically synthesised harness reached 84.3% on TerminalBench-2 and found ten previously unknown Chrome vulnerabilities [1].

Two older results point the same way. Combining heterogeneous models layer by layer outperforms any single member [6]. Adaptive tree search that chooses between "going wider" (new candidates) and "going deeper" (refining existing ones) outperforms repeated sampling [7].

> **Takeaway.** Match the model, prompt, and context to the task, and combine configurations to get coverage. The intuition that an orchestrator over several providers beats any single model is supported, at least for search-like tasks with verifiable outputs.

4 Long-Horizon Degradation: Context Rot and Compaction

4.1 Context rot

Models use information in the middle of a long context less reliably than information at the beginning or end [8]. In agent settings the effect shows up as behaviour. A white-box study of a code-audit workflow found a coding agent passing 8 of 10 runs with an 11K-character context, but only 3 of 10 with a 299K-character context. The drop was the same whether the added material was relevant or irrelevant; it is a trend-level result ($p \approx 0.07$) [9]. In long-horizon search, extensive context causes *premature termination*: the model gives up or commits to an uncertain wrong answer well before the window is full, and the rate rises with context length [10]. We found no evidence for a fixed usable fraction of the window, such as the commonly quoted "40%". Degradation is gradual and depends on the task.

4.2 Compaction is lossy and a trust boundary

Compaction tuned for the current question does not serve later questions. On later questions, no sentence-selection method tested did detectably better than deleting sentences at random at the same token budget. A question-agnostic word-level compressor did better, beating random sentence deletion by 12.5 points on later MultiRC questions [11]. Compaction is also a security boundary. Harnesses that flatten typed messages into a single string with text labels such as `[User]:` lose provenance. In those harnesses, a forged user turn planted in any content the agent reads survives compaction as a genuine instruction in 87–100% of trials. Harnesses that keep role typing (Codex, Gemini CLI, Qwen Code, Continue) showed a 0% success rate for the same attack [12].

4.3 Fresh contexts for sub-tasks

When reusable procedures ("skills") have clear input–output contracts, running them as sub-agents in fresh context windows outperforms loading their instructions into the main context. The cost is coordination tokens [13].

> **Takeaway.** Neither aggressive clearing nor routine compaction is free. Prefer structured resets with explicit handoffs (Section 5), push bounded sub-tasks into fresh sub-agents, and use harnesses that keep message roles typed.

5 Escaping Idea Basins: Forgetting Beats Running Longer

The strongest result in this review concerns open-ended, iterative "autoresearch" agents. Table 2 summarises four papers that reach the same conclusion by different routes.

Table 2: Context-reset and initialisation interventions for iterative research agents. Gains are as reported in each paper's abstract and are not comparable across rows.

Paper	Intervention	Reported effect
Scaling Discovery Requires Forgetting [14]	Periodically discard every agent's session *and* the candidate history; restart the team from the best candidate.	Up to +15% vs. parallel agents and up to +39% vs. a single agent, on four tasks (heuristics, kernels, model training).
Fork-and-Flush [4]	Fork into parallel trajectories that keep the workspace but start from a fresh chat context; continue from the best one after a fixed horizon.	+66.0% vs. single run and +44.4% vs. best-of-N at equal compute, across 13 tasks with runs lasting up to several days.
Humanize MLE Agents [15]	Two agents alternate on a shared workspace, at most k experiments per session, context reset between sessions.	78.2% any-medal rate on MLE-bench (75 tasks), +8.0 points over the strongest open baseline at a quarter of its time budget.
Initialization Improves Discovery [16]	A parallel exploration stage first, then iterative optimisation.	Consistent gains across 12 harnesses and 5 discovery tasks; early discovery quality predicts final success.

The papers share a mechanism. Agents anchor on their own history and make progressively smaller edits [14]. Independent runs of the same agent plateau at very different scores [4]. Reasoning intensity falls as a single session lengthens (*long-horizon deceleration*), and the benefits of custom harnesses fade as base models improve (*harness staleness*) [15]. Harness staleness is consistent with the practitioner observation that recent models keep working toward a goal stated in the prompt without external "keep-going" loops.

Mode collapse, a sharp drop in the diversity of iterates, is the common failure. Diversity-inducing harness tweaks give inconsistent gains against it, whereas investing in a diverse initial population does help [16].

What to hand off. Handoff content matters because context is "contagious". In homogeneous multi-agent debate, agents adopted the majority answer in up to 85.5% of cases, and isolated self-correction outperformed debate [17]. A practical reading is to pass the goal, the constraints, and distilled insights (which directions paid off and which were dead ends) to new branches, and not the raw trajectories.

> **Takeaway.** To get past a plateau, fork, flush, and restart from the best candidate with a compact handoff. A human-level analogue also seems to work (practitioner observation): two engineers build harnesses independently against a shared success metric and merge when either runs out of ideas.

6 Diversity: Personas, Hiveminds, and Collapse

"Agent temperature" implemented through persona prompts is a popular way to generate ideas, and the evidence on it is mixed.

- Independent LLM samples are less diverse than independent human ideas. Two mechanisms are identified: *fixation*, where early outputs constrain later ones (chain-of-thought reduces it), and *knowledge aggregation*. *Ordinary* personas counter aggregation better than "creative entrepreneur" personas such as Steve Jobs, and combining both interventions beat human diversity [18]. This matches the practitioner observation that famous-figure personas (pitting historical personalities against one another) work worse than expected.
- Without intervention, inter-response similarity stays around 0.80–0.90 even at high temperature. A self-selected idiosyncratic persona combined with filtered high-temperature sampling raises diversity [19]. Persona pairing and collaboration mode also shape which domains brainstormed ideas come from [20].
- Asking the model to *verbalise a distribution* over several candidate responses increases creative-writing diversity by 1.6–2.1× without loss of factuality [21].
- Interacting multi-LLM systems run in closed loops for 200–1000 rounds show *semantic collapse* across model families. None of thirteen interventions (decoding, prompting, agent composition, activation steering, RL) reliably restored diversity [22].

> **Takeaway.** Personas and verbalised sampling widen the *initial* pool. Long closed loops erode it again, so diversity has to be re-injected through resets (Section 5), not only at the start. Persona-based "temperature" still lacks a metric; one candidate for bug-finding work is time-to-k-confirmed-findings at a fixed budget.

7 Reasoning Effort

Higher effort costs more and takes longer, but it does not always perform better. The inverse-scaling study constructs tasks in which longer reasoning lowers accuracy. Failure modes include distraction by irrelevant information, overfitting to how the problem is framed, and drifting toward spurious correlations [23]. In vulnerability discovery, frontier models had low false-report rates but did not consistently produce the highest yield [5].

> **Takeaway.** Set effort per phase, not per session. Use high effort for open-ended investigation and final verification, and medium effort for well-specified implementation. Treat any top-tier default as a cost decision that needs a reason.

8 Cross-Project Memory and Distillation

One pattern is a single agent layer spanning all of a practitioner's projects. The agent indexes documents, builds a knowledge graph, and keeps summaries of best practices and conventions, so that a new project starts with that context instead of from scratch. The closest published mechanism is Agentic Context Engineering [24], which treats context as an evolving *playbook* built through generation, reflection, and curation. ACE names two failure modes of naive approaches: *brevity bias*, where summaries drop domain insights, and *context collapse*, where iterative rewriting erodes detail. Structured incremental updates prevent both. ACE reports +10.6% on agent benchmarks and adapts from execution feedback without labels.

9 Security of the Agent Stack

Protocol and harness layers are now an attack surface in their own right:

- **MCP.** In typical local deployments, approving a tool call grants the server process broad filesystem permissions. Attacker-steered arguments can then read credential files outside the tool's declared contract (*filesystem capability laundering*) [25].
- **A2A.** A formal state-machine model extracted from the specification (37 states, 76 transitions) supports a systematic analysis of protocol-level vulnerabilities [3].
- **Compaction.** Provenance loss lets forged instructions persist in memory [12].

Incidents disclosed during the week of writing. Three incidents in the 48 hours before this note illustrate both how fast exploitation moves and the role of AI in finding vulnerabilities: (i) *PHP2Shell*

(CVE-2026-109618/109619, CVSS 9.2 each) is a pair of unauthenticated RCE flaws in the PHP core, found with AI assistance by Calif and, independently, by Wiz. Fixes shipped in PHP 8.5.12, 8.4.27, 8.3.36 and 8.2.35, and an exploitation attempt was observed four days before disclosure [26], [27], [28]. (ii) *Railway* reported that an unauthorised party may have accessed customers' encrypted environment variables and decrypted some of them, and urged all users to rotate their keys [29]. (iii) *LiteLLM* v1.107.0 was published to PyPI and Docker from a compromised release pipeline with a credential stealer. The package was live on PyPI for about two hours and was downloaded roughly 49,000 times [30], [31].

10 Human Factors

Delusion will not disappear by itself. Hallucination persists in part because training and evaluation reward guessing over acknowledging uncertainty [32]. As capability grows, confabulation moves to levels that are harder to check, such as invented APIs, invented methods, and plausible but unverifiable mathematics. Verifying it then needs more domain expertise.

Cognitive offloading. In an EEG study, participants who wrote with an LLM showed the weakest brain connectivity of three groups. When they then wrote without it, they remained under-engaged [33]. Deskilling has been proposed as an overlooked AI risk category [34].

Working principles. We hold to the following norms. They are positions, not findings: (1) do not run an agent without understanding the task; (2) where the operator does understand the task, their judgement outranks the model's; (3) responsibility stays with humans; (4) do not outsource genuine novelty, because agents tend to produce *acceptable* ideas rather than new ones; (5) do not hand meaningful human communication over to an agent.

11 Practical Recommendations

1. Define a success metric before building the harness. Without one, none of the interventions above can be evaluated.
2. For search-like tasks, combine heterogeneous models and configurations, because coverage, not peak score, is what grows [5], [6].
3. Spend compute early on a diverse parallel initial population [16].
4. Fork and flush at plateaus. Restart from the best candidate with a compact handoff of goal, constraints, and insights, and not the raw context [4], [14], [15].
5. Push bounded sub-tasks into fresh-context sub-agents [13]. Use role-typed harnesses where compaction is unavoidable [12].
6. Use ordinary personas and verbalised sampling for ideation, and filter the output hard [18], [21].
7. Allocate reasoning effort by phase [23].
8. Pin dependencies, limit the filesystem scope of MCP servers, and rotate secrets stored on third-party platforms.

12 Open Questions and Unsupported Claims

For the following claims we found no direct supporting study:

- that freshly cached context is used worse than context that has "settled" over several iterations;
- that there is a fixed usable fraction of the context window;
- that a strong model benefits from consulting a weaker model as a "mentor";
- how large a team of humans and agents can work on one problem before the cost of merging outweighs the diversity gained.

Pointers to evidence on any of these are welcome.

Acknowledgements. This note grew out of an informal reading-group discussion. We thank the participants for the questions that shaped it.

References

[1] H. Liu *et al.*, "Synthesizing Multi-Agent Harnesses for Vulnerability Discovery." [Online]. Available: https://arxiv.org/abs/2604.20801

[2] S. Yao *et al.*, "ReAct: Synergizing Reasoning and Acting in Language Models." [Online]. Available: https://arxiv.org/abs/2210.03629

[3] A. Lotfi, M. M. Rahman, I. Karim, and E. Bertino, "A2ABreak: Systematic Security Analysis of the A2A Protocol." [Online]. Available: https://arxiv.org/abs/2609.10871

[4] Z. Cai *et al.*, "Fork-and-Flush: Escaping Idea Basins in Autoresearch Agents." [Online]. Available: https://arxiv.org/abs/2610.07447

[5] G. G. Curmi and B. Wilde, "How Much Does the Harness Matter, and for Whom? A Multi-Model Study of Context in LLM-Driven Vulnerability Discovery." [Online]. Available: https://arxiv.org/abs/2610.28802

[6] J. Wang, J. Wang, B. Athiwaratkun, C. Zhang, and J. Zou, "Mixture-of-Agents Enhances Large Language Model Capabilities." [Online]. Available: https://arxiv.org/abs/2406.04692

[7] Y. Inoue, K. Misaki, Y. Imajuku, S. Kuroki, T. Nakamura, and T. Akiba, "Wider or Deeper? Scaling LLM Inference-Time Compute with Adaptive Branching Tree Search." [Online]. Available: https://arxiv.org/abs/2503.04412

[8] N. F. Liu *et al.*, "Lost in the Middle: How Language Models Use Long Contexts." [Online]. Available: https://arxiv.org/abs/2307.03172

[9] Y. Xue, "When and How Context Rot Appears in Coding Agents: A White-Box Study of Agent Skills in Code Auditing." [Online]. Available: https://arxiv.org/abs/2607.17937

[10] S. Xia, Y. Wang, Z. Huang, and P. Liu, "Diagnosing and Mitigating Context Rot in Long-horizon Search." [Online]. Available: https://arxiv.org/abs/2606.29718

[11] J. Yao *et al.*, "Evaluating Context Compaction on the Questions That Come Later." [Online]. Available: https://arxiv.org/abs/2610.27301

[12] A. Dabirsiaghi, "Compaction Laundering: Provenance Destruction in Context-Compacting Coding Agents." [Online]. Available: https://arxiv.org/abs/2610.29869

[13] W. T. Piriyakulkij, R. Lawrence, A. Curth, S. Karmalkar, and N. Prasad, "Subagents vs Agent Skills: Executing Reusable Knowledge for Long-Horizon Agentic Tasks." [Online]. Available: https://arxiv.org/abs/2609.09233

[14] S. Tan *et al.*, "Scaling Discovery Requires Forgetting." [Online]. Available: https://arxiv.org/abs/2610.20563

[15] C. Li *et al.*, "Humanize MLE Agents: Multi-Agent Session Alternation Beyond Single-Agent Limits." [Online]. Available: https://arxiv.org/abs/2610.13494

[16] M. Sakarvadia, M. Ciccone, and C. Raffel, "Initialization Improves LLM-Driven Discovery." [Online]. Available: https://arxiv.org/abs/2610.00707

[17] B. Bertalanič and C. Fortuna, "The Cost of Consensus: Isolated Self-Correction Prevails Over Unguided Homogeneous Multi-Agent Debate." [Online]. Available: https://arxiv.org/abs/2605.00914

[18] Y. Deng, M. Brucks, and O. Toubia, "Examining and Addressing Barriers to Diversity in LLM-Generated Ideas." [Online]. Available: https://arxiv.org/abs/2602.20408

[19] T. Fu, J. Conde, C. Arriaga, G. Martínez, P. Reviriego, and J. Coronado-Blázquez, "Beyond the Hivemind: Escaping LLM Homogeneity via Meta-Persona Anchoring and Sequential Temperature Scaling." [Online]. Available: https://arxiv.org/abs/2608.02618

[20] N. Straub, S. Khan, K. Jay, B. Cabral, and O. Linde, "Persona-based Multi-Agent Collaboration for Brainstorming." [Online]. Available: https://arxiv.org/abs/2512.04488

[21] J. Zhang *et al.*, "Verbalized Sampling: How to Mitigate Mode Collapse and Unlock LLM Diversity." [Online]. Available: https://arxiv.org/abs/2510.01171

[22] W. Kong, S. Lai, J. Piao, and J. Evans, "Multi-LLM Systems Exhibit Robust Semantic Collapse." [Online]. Available: https://arxiv.org/abs/2605.17193

[23] A. P. Gema *et al.*, "Inverse Scaling in Test-Time Compute." [Online]. Available: https://arxiv.org/abs/2507.14417

[24] Q. Zhang *et al.*, "Agentic Context Engineering: Evolving Contexts for Self-Improving Language Models." [Online]. Available: https://arxiv.org/abs/2510.04618

[25] Z. Huang, M. Tang, A. Guan, Q. Sun, and P. He, "My Credentials, Please: Detecting Filesystem Capability Laundering in MCP Servers at Scale." [Online]. Available: https://arxiv.org/abs/2610.22301

[26] S. Gatlan, "PHP ships emergency patches for critical PHP2Shell flaws." [Online]. Available: https://www.bleepingcomputer.com/news/security/php-ships-emergency-patches-for-critical-php2shell-flaws/

[27] C. Tolomia, "PHP releases patches for two critical remote code flaws found with AI." [Online]. Available: https://qz.com/php-critical-flaws-ai-patch-remote-code-execution-102126

[28] The Hacker News, "Someone tried to exploit PHP2Shell 4 days before the vulnerabilities were publicly disclosed." [Online]. Available: https://thehackernews.com/2026/10/someone-tried-to-exploit-php2shell-4.html

[29] TechCrunch, "Developer platform Railway urges "all users" to revoke secret keys after cyberattack." [Online]. Available: https://techcrunch.com/2026/10/22/developer-platform-railway-urges-all-users-to-revoke-secret-keys-after-cyberattack/

[30] LiteLLM, "Contained: Oct 21, 2026 LiteLLM release v1.107.0 compromised." [Online]. Available: https://docs.litellm.ai/blog/security-incident-oct-21-2026

[31] B. Read and R. McCarthy, "LiteLLM compromised, steals credentials." [Online]. Available: https://www.wiz.io/blog/litellm-compromised-steals-credentials

[32] A. T. Kalai, O. Nachum, S. S. Vempala, and E. Zhang, "Why Language Models Hallucinate." [Online]. Available: https://arxiv.org/abs/2509.04664

[33] N. Kosmyna *et al.*, "Your Brain on ChatGPT: Accumulation of Cognitive Debt when Using an AI Assistant for Essay Writing Task." [Online]. Available: https://arxiv.org/abs/2506.08872

[34] I. Chalkidis and A. Søgaard, "Brainrot: Deskilling and Addiction are Overlooked AI Risks." [Online]. Available: https://arxiv.org/abs/2605.03512