THE HIDDEN FOOTPRINT: MAKING STORAGE A FIRST-CLASS METRIC FOR LLM AGENT EVALUATION

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM agent benchmarks measure task completion, reliability, and inference cost, but not the persistent data an agent run leaves on local disk, including logs, context snapshots, checkpoints, and debug traces. These bytes are absent from the leaderboards we survey, yet bear on whether an agent system can be deployed on desktops, under data-retention compliance regimes, or at fleet scale. To our knowledge, AGENTFOOTPRINT is the first systematic cross-framework benchmark of post-run agent storage footprint. Its serialization-aware metric suite covers total retention, channel composition, duplication, growth exponent, compressibility, and a conversation-history reconstructability score. It addresses a measurement trap: naïve byte-level measurement understates duplication by an order of magnitude, because database paging and JSON escaping obscure repeated content. The footprint has two determinants: the logical volume generated by the agent's behavior and the amplification added by the persistence layer. A fixed-trace control compares the latter: the same scripted content replayed through each persisting framework's adapter yields retained sizes differing by $6.7\times$. With the same model, tools, and tasks, LangGraph and OpenAI Agents both achieve 100% accuracy, attain the highest conversation-reconstructability score, and support fresh-process continuation, yet their default retention differs by $15.7\times$. Three full-history configurations grow superlinearly over the measured horizons of a repeated-observation stress task. On a minimal write task producing 39-byte summaries, framework residue exceeds the delivered output files by orders of magnitude. Exported trajectories from 108 instance-normalized SWE-bench Verified submissions span three orders of magnitude in per-instance volume with no detectable correlation with resolve rate. Offline content addressing with zstd reduces retention by 4.8–$32.7\times$ while preserving reconstructability scores after restoration. Reconstructing recorded conversation content does not require megabytes, and these storage metrics can be reported alongside inference cost.

1 INTRODUCTION

LLM agents—systems that pursue multi-step tasks by interleaving model calls with tool use—are moving from demos into production, and the community evaluates them along an expanding set of axes: task success on GAIA (Mialon et al., 2024) and SWE-bench (Jimenez et al., 2024), reliability via pass^k on τ-bench (Yao et al., 2025), and, following Kapoor et al. (2025), inference cost. A recent survey of agent evaluation catalogs more than a dozen such metrics (Mohammadi et al., 2025). None includes the *storage footprint*, defined here as the bytes an agent execution leaves behind on disk.

Unlike inference cost, which is consumed when execution ends, retention is a stock that persists and *accumulates* across runs. This persistence is simultaneously useful and costly. It is the substrate for replay debugging, recovery, and compliance audits, but it also enlarges long-term storage, governance, and deletion obligations. The retained volumes are easy to underestimate. In a production deployment that we measured directly (a supply-chain quoting system; Appendix P), a single-task inventory totaled 131 MiB: 49.2 MiB of staged inputs, 79.2 MiB of framework residue, and 2.6 MiB of deliverables. A separate eight-directory audit found occupancy from 248 B to 81.4 MiB, with a median of 504 KiB. The full deployment accumulated $\sim$56 GB over $\sim$1,000 recorded rounds (operator-attested).

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Our goal is to make this axis measurable and comparable across agent frameworks. Doing so is challenging for three reasons. First, *attribution*: an agent run scatters bytes across working directories, session databases, hidden home-directory state, and tokenizer caches. Deciding which bytes are task-attributable requires per-run isolation, not a disk-usage command. Second, *serialization masking*: the same file content is re-stored many times inside SQLite pages and JSON-escaped message arrays, where page fragmentation and escape sequences defeat byte-level fingerprinting. Raw-file chunking can report $D{=}1.01$ for a store logically holding the same content twelve times over. Third, *semantics*: raw byte counts alone cannot distinguish an agent that retains much to support auditing from one that retains much because of avoidable storage amplification. The metric suite must indicate which recovery and audit capabilities the retained bytes support.

We present AGENTFOOTPRINT, a benchmark and measurement harness for these challenges. Each (framework, task) pair runs in a fresh sandbox whose workspace and home delta is captured and attributed (challenge one). Our meter analyzes *logical content streams* (SQLite cells and JSONL records) rather than raw files, and injects content probes that are matched under raw and JSON-escaped encodings (challenge two). On top of this substrate we define six measurements: total retention S_{total}, a descriptive storage-channel composition, duplication factor D, growth exponent α fitted over controllable-horizon tasks, long-window compressibility C, and a 0–3 history-reconstructability score R asking whether retained bytes reconstruct the conversation history behind step k (challenge three). Storage is read jointly with accuracy and R. A fixed-trace control compares persistence under fixed scripted content (§5.6).

Measuring eight representative frameworks (LangGraph, AutoGen, CrewAI, SmolAgents, OpenAI Agents SDK, LlamaIndex, Agno, and InfiAgent) under identical backends, tools, and tasks yields three findings. (i) *Equal-accuracy dispersion*: LangGraph and OpenAI Agents both achieve 100% task accuracy, $R{=}3$, and fresh-process continuation, yet their default retention differs by $15.7\times$. (ii) *Observed growth*: full-history configurations have fitted α up to 1.95 over 25–200 rounds: a 200-round monitoring loop over one 2 KiB status file leaves 323 MiB, while frameworks with windowed context management stay near-linear. (iii) *Residue dominates artifacts*: when agents produce 39-byte summary files, framework residue exceeds the deliverables by up to $24{,}033\times$; the production case above shows about $30\times$.

Public agent submissions exhibit an analogous dispersion in exported trajectory volume: across 108 instance-normalized SWE-bench Verified submissions, per-instance volume spans $1{,}617\times$ with no detectable correlation with resolve rate. A content-addressed store removes most retained bytes while all restore checks pass and every R score is reproduced.

Our contributions are:

- **A metric suite and measurement methodology** for agent storage footprint (S_{total}, composition, D, α, C, R), including the finding that serialization masks duplication from naïve measurement by up to $12\times$ (§3).
- **AGENTFOOTPRINT**, a reproducible harness and task suite measuring eight frameworks over 1,051 sandboxed executions and ten post-hoc channel variants, with container-validated isolation (§4–§5; full accounting in Appendix C).
- **An in-the-wild study** harvesting all 134 SWE-bench Verified submissions (108 instance-normalized), finding a $1{,}617\times$ volume spread with no detectable correlation between exported volume and success ($\tau_b = 0.080$, $p = 0.222$) (§6).
- **A headroom demonstration**: a content-addressed compactor shows that the measured reconstructability score survives at 4.8–$32.7\times$ fewer bytes than the frameworks' defaults (§7).

2 RELATED WORK

AGENTFOOTPRINT expands *what* agent benchmarks measure rather than *how well* agents score. Kapoor et al. (2025) introduced inference cost as a mandatory reporting axis and showed simple baselines dominate the cost–accuracy Pareto frontier. τ-bench added reliability via pass^k (Yao et al., 2025). Storage footprint differs from inference cost because it persists and compounds after the task. Harness disclosure makes configuration part of evaluation (Zhang et al., 2026). Efficient Agents

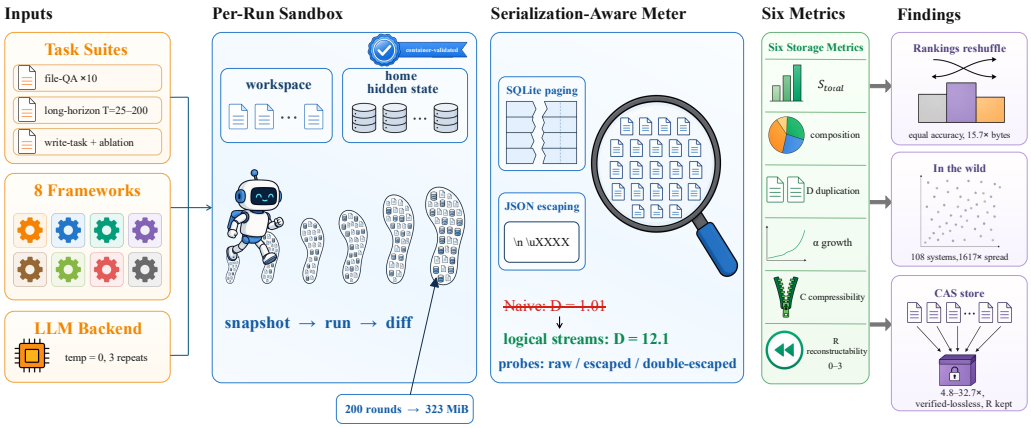


Figure 1: AGENTFOOTPRINT pipeline: each (framework, task, repetition) runs in a fresh sandbox. The filesystem delta is analyzed as *logical content streams* rather than raw bytes, yielding the six metrics in §3 that drive §5–§7.

studies effectiveness against inference cost (Wang et al., 2025). MAFBench reports architecture-dependent latency and coordination costs (Orogat et al., 2026). Neither measures persistent residue, which AGENTFOOTPRINT isolates.

A second line optimizes what agents keep *in context or in memory stores* (Wang et al., 2026). MemRefine (Kim et al., 2026) compresses memory banks for retrieval; Slipstream (Chen et al., 2026) shortens prompts. VISTA archives context blocks as recoverable payloads (Xu et al., 2026). These methods target model-facing state. The persistence layer can still checkpoint every intermediate underneath. We measure the *system-wide residue*, coupled to context policy through α.

Trajectory platforms (LangSmith, Langfuse, AgentOps) persist agent runs as a product feature but publish no efficiency measures (potential reporters of such metrics, not sources). CLP (Rodrigues et al., 2021) and zstd (Collet & Kucherawy, 2021) compress syntactically (zstd is our baseline C). Our semantic metrics capture what compression neither reveals nor repairs: what is duplicated, how retention scales, whether bytes reconstruct history.

Finally, the gap is explicit in the surveys: Mohammadi et al. (2025)'s metric catalog spans completion, step-efficiency, tool correctness, plan quality, latency, and cost, but no storage dimension. GAIA, SWE-bench, and τ-bench report none. To our knowledge, AGENTFOOTPRINT is the first systematic cross-framework benchmark of post-run storage footprint for LLM agents. Adjacent lines (workflow provenance (Davidson & Freire, 2008; Herschel et al., 2017), LSM storage amplification (O'Neil et al., 1996)) instrument different objects. We import their techniques as mitigations.

3 STORAGE FOOTPRINT METRICS

3.1 WHAT COUNTS: BOUNDARY AND TAXONOMY

We define the storage footprint as post-run local retention attributable to an execution: files created or grown in the workspace and framework-side stores (session/checkpoint databases, logs, snapshots, and traces), including hidden home-directory state. The baseline follows framework setup and input staging; remote telemetry lies outside this filesystem boundary. *Footprint* and *retention* are used interchangeably; *residue* is the framework-side portion (all but agent-produced workspace artifacts). Environment artifacts shared across tasks (model weights, tokenizer/package caches) are excluded, although their volume is reported for auditability. Measurement uses a per-run sandbox: a fresh workspace and home whose file inventory is snapshotted before and diffed after the run (§4.3). Appendix N audits the filesystem accounting.

Retained bytes are not equally justified. We classify them as *decisions* (model outputs, generally not regenerable), *observations* (tool and environment returns, some of which are likewise unrecov-

erable), *derivations* (framework-generated intermediates: snapshots, summaries, checkpoints), and *duplicates* (the second through n-th appearance of the same bytes), separating what must be kept, what may be referenced, and what is overhead. The taxonomy is conceptual; the benchmark reports deterministic storage-channel and duplication measurements, not automated semantic attribution of every byte.

3.2 Six Metrics

On this foundation we define six measurements per (framework, task) run:

S_{total} Retained bytes (apparent-length filesystem delta; Appendix N), divided into workspace artifacts and framework-side residue.

Store composition S_{total} grouped into state/checkpoints, debug/event logs, conversation/actions, and other channels by deterministic, adapter-audited path rules.

D Duplication factor $S_{\text{total}}^{\text{logical}}/S_{\text{unique}}$. Here, S_{unique} sums unique content-defined chunks (FastCDC (Xia et al., 2016) with SHA-256) over *logical streams*. SQLite cell extraction excludes page headers, index structures, and free pages. $D{=}2$ means the average extracted content byte is stored twice. This is an exact-match lower bound (Appendix K); comparing S_{total} with the logical total reveals physical-layout overhead.

α Growth exponent from fitting $\log S_{\text{total}} \sim \alpha \log T$ at $T \in \{25, 50, 100, 200\}$ on the repeated-observation stress task—a descriptive statistic; per-horizon retention is primary (Appendix G).

C Compressibility: S_{total} divided by its size under zstd -19 with a 128 MiB long-range window over the concatenated retained files. This is a compression bound, not an archive. We report it to separate syntactic compression gains from semantic structure.

R Conversation-history reconstructability, 0–3: given call boundaries (§5.4), can the retained bytes reconstruct the conversation history the model received at step k? $R{=}0$: nothing retained; $R{=}1$: bytes but no per-call structure; $R{=}2$: per-call structure but no complete copy of the observations; $R{=}3$: at least one channel holds a complete record under the reconstruction contract. Automated candidate grades require field-by-field reconstruction validation (§5.4; construct check in Appendix O).

3.3 Measurement Methodology: Serialization Masks Duplication

A key methodological finding is that raw-file chunking substantially understates duplication. Running content-defined chunking over raw retained files reports $D = 1.01$ for LangGraph's checkpoint store, although zstd compresses it by $142\times$. This apparent discrepancy indicates missed logical repetition. Two serialization mechanisms mask the duplication. SQLite fragments large values across 4 KiB pages interleaved with headers and pointers, disrupting chunk alignment across copies. JSON escaping rewrites the bytes of embedded content (`\n`, `\uXXXX`), so trajectory copies no longer match the source bytes. Our meter therefore extracts *logical streams* before fingerprinting: qualifying SQLite cells and JSONL records form separate streams, while other files are whole streams. FastCDC uses minimum, target, and maximum chunk sizes of 1, 4, and 32 KiB (extraction rules in Appendix K). Under logical streams the same store measures $D = 12.1$ (a single store; Table 1's 30-run means are D 12.8, C 136); the duplication was present but obscured by serialization.

Cross-representation copies still evade chunk matching, so we add *content probes*: deterministic 72-character substrings from the input corpus, matched under raw and JSON-escaped encodings. Their mean occurrence count, *echo*, measures corpus-content repetition in retained streams. Unread files remain in the denominator, so echo can be below one even when all observed content is retained. Separate whole-line probes test observation coverage for R. Appendix K specifies both procedures and calibrates D and echo on known-duplication stores.

4 THE AGENTFOOTPRINT BENCHMARK

4.1 TASK SUITES

AGENTFOOTPRINT uses six task families, each isolating one storage behavior, plus a control. The *file-QA suite* (10 tasks) gives the agent ten ∼60 KiB documents (600 KiB total) and five sequential questions whose answers are unique registry codes embedded in the files; questions revisit files in an A,B,A,C,A pattern with instructed re-reading (read amplification). Grading is exact substring match: no LLM judge. The *long-horizon suite* fixes a 2 KiB status file re-read for $T \in \{25, 50, 100, 200\}$ rounds, isolating α from content growth. The *write-task suite* (5 tasks) reads two documents and produces a summary file via a `write_file` tool (legitimate artifacts vs. residue). The *exploratory-retrieval suite* (5 tasks) lets agents choose which files to inspect (§5.5). Two *heterogeneous families* (five tasks each), *edit* and *data analysis*, are read–transform–write tasks graded *jointly*: correct answer *and* artifact delivered in the shared workspace (§5.5). Finally, *ablation runs* re-execute the file-QA suite with persistence disabled.

4.2 FRAMEWORKS AND FAIRNESS PROTOCOL

We measure eight representative frameworks: LangGraph 1.2.8 (LangChain AI, 2026), AutoGen 0.7.5 (Microsoft, 2025), CrewAI 1.15.2 (CrewAI Inc., 2026), SmolAgents 1.26.0 (Hugging Face, 2026), OpenAI Agents SDK 0.18.0 (OpenAI, 2026), LlamaIndex 0.14.23 (LlamaIndex Inc., 2026), Agno 2.7.1 (Agno AGI, 2026), and InfiAgent 3.12.24 (Yu et al., 2026). The roster covers widely used open-source agent APIs with distinct persistence mechanisms. Agno and InfiAgent provide two independent implementations of windowed context management (a runtime forgetting loop and a bounded per-request window), so the growth study (§5.2) can compare full-history and bounded-history policies.

We use each framework's *documented durable-session configuration*, selected by a predefined protocol (Table 1). Frameworks receive identical task text, the same minimal tools (list, read, write where applicable), and the same backend (DeepSeek-V4-Flash, served by one provider, temperature 0). The main file-QA study uses three independent repetitions. The exact adapter code ships in the artifact, and §8 reports a configuration-sensitivity check (AutoGen's save cadence).

The measured configurations and their capabilities are documented in Appendix R; comparison limits are discussed in §8.

4.3 HARNESS

The harness (Figure 1) runs each (framework, task, repetition) in a fresh sandbox. The workspace holds the task corpus, and `HOME`/XDG paths are redirected into the sandbox to capture hidden state. After framework setup and input staging, the meter records the baseline. The adapter runs the framework's quickstart-style agent, and the meter diffs both inventories. Re-running LangGraph (three tasks) in per-run Docker containers changed measurements by 0–8.5%, below the 11% repetition-level standard deviation. Harness, generators, adapters, and per-run records ship in the artifact. Appendix C enumerates all executions and post-hoc variants and states the statistical units and aggregation conventions.

5 CONTROLLED RESULTS

5.1 SAME ACCURACY, 15.7× THE BYTES

Table 1(a) summarizes the file-QA suite: eight frameworks, ten tasks, and three repetitions. LangGraph and OpenAI Agents answer every question correctly, achieve $R{=}3$, and support fresh-process continuation, yet retain 5.10 and 0.33 MiB per task, respectively. Their unrounded means differ by 15.7×. CrewAI retains a similar 0.32 MiB as task outputs, with $R{=}1$; SmolAgents retains zero bytes, with $R{=}0$. The full equal-accuracy default spread, from LangGraph to CrewAI, is also 15.7× (bootstrap 95% CI $[15.0, 16.3]$; Appendix C). Six frameworks answer every question correctly; Agno achieves 99.3% and AutoGen 86%.

Table 1: Main results: the eight measured configurations as one profile. **(a)** File-QA suite, 10 tasks $\times$ 3 repetitions: S_{total} mean$\pm$SD; R modal score; CAS first-repetition subset, paired reduction in parentheses (Table 12). α is the stress-task growth exponent (mean$\pm$SD over three seeds for the full-history trio, single run otherwise). **(b)** Persistence mechanism, channel composition, fresh-process resume, and backend replication on Qwen3.6-27B and Qwen2.5-7B. Sizes are in MiB; aggregation details appear in Appendix C.

(a) Framework	$S_{\text{total}} \downarrow$	$D \downarrow$	Echo	C	α	Acc. $\uparrow$	$R \uparrow$	CAS $\downarrow$
LangGraph	5.10 ± 0.58	12.8	8.1	136	1.74 ± 0.01	100%	3	0.50 (10.3$\times$)
AutoGen	2.49 ± 0.33	8.4	4.1	106	1.95 ± 0.18	86%	3	0.07 (32.7$\times$)
InfiAgent	2.12 ± 0.05	1.8	1.7	43	1.15	100%	3	0.36 (5.8$\times$)
Agno	1.15 ± 0.02	2.5	0.8	42	0.98	99.3%	3	0.06 (20.5$\times$)
LlamaIndex	$0.93 \pm <0.01$	4.2	1.3	40	1.88 ± 0.10	100%	3	0.09 (10.3$\times$)
OpenAI Agents	0.33 ± 0.02	1.6	0.5	14	0.73	100%	3	0.07 (4.8$\times$)
CrewAI	$0.32 \pm <0.01$	1.5	0.5	14	1.20	100%	1	0.05 (6.1$\times$)
SmolAgents	0	–	–	–	–	100%	0	–

(b) Framework	Persistence mechanism	State	Debug	Conv.	Resume	Qwen-27B	Qwen-7B
LangGraph	SQLite checkpointer	5.10	–	–	✓	5.20	2.49
AutoGen	Per-turn state + event log	0.83	1.65	–	✓	2.64	1.26
InfiAgent	Task stores + raw traces	0.07	0.86	1.20	✓	2.13	2.14
Agno	SQLite session database	1.15	–	–	✓	1.16	0.77
LlamaIndex	Serialized `Context`	0.93	–	–	✓	0.94	1.20
OpenAI Agents	`SQLiteSession`	0.33	–	–	✓	0.31	0.21
CrewAI	Task-output store	0.32	–	–	×	0.32	0.37
SmolAgents	None (in-memory)	–	–	–	–	0	0

Composition (Table 1b) shows retention concentrated in different channels: checkpoints (LangGraph), event logs (AutoGen), conversation stores and debug traces (InfiAgent). Including InfiAgent's input staging raises its mean from 2.12 to 2.72 MiB (Appendix N). The duplication factor separates architectural families: full-history checkpoint/snapshot designs (D: LangGraph 12.8, AutoGen 8.4, LlamaIndex 4.2) are most redundant, while windowed or append-only designs stay below 2.6. On the same corpus and prescribed read pattern, mean probe counts are 8.1 under LangGraph and 0.5 under OpenAI Agents.

5.2 Observed Growth under Repeated Observations

On the controlled repeated-observation stress task, the retention curves show three patterns (Figure 2; α in Table 1; three seeded runs per horizon for the full-history trio). Frameworks that persist the full message history at every step (LlamaIndex, AutoGen, and LangGraph; α 1.74–1.95) grow superlinearly: monitoring one 2 KiB status file for 200 rounds leaves 323 ± 26 MiB under LangGraph ($D \approx 49$), 102 ± 21 MiB under AutoGen. Frameworks with windowed context (InfiAgent's forgetting loop) or bounded per-round stores (CrewAI, Agno) stay near-linear (α 0.98–1.20): at $T=200$ (completed by every persisting framework) they retain 1.4–36.4 MiB against LangGraph's 323 ± 26 MiB.

At $T=200$, the median number of correct answers across persisting frameworks and seeds is 195, with a minimum of 171. Every seed of every full-history framework fits $\alpha \geq 1.72$ over the measured range. Per-horizon values, local slopes, and alternative fits appear in Appendix G. Archival policy matters as much as architecture: LlamaIndex's newest-snapshot-only form leaves 85–645 KiB ($\alpha \approx 0.98$); superlinear growth follows from the *append-all* archival policy, not from durable resumption (Appendix G).

OpenAI Agents appends messages rather than re-serializing the entire history. Its retained size rises from 124 to 540 KiB over 25–200 rounds, with a fitted $\alpha=0.73$. An exponent below one can reflect fixed storage overhead as well as changing per-round additions (Appendix G). At $T=200$, echo

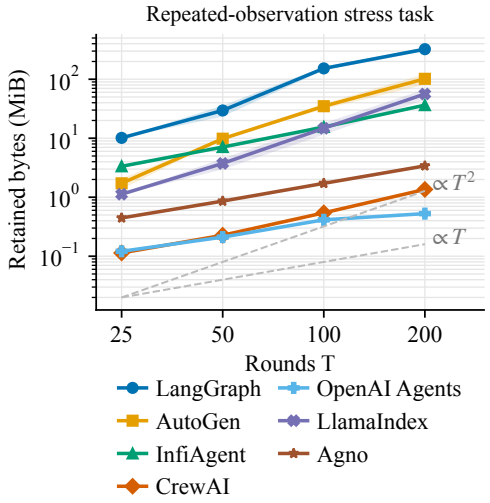


Figure 2: Retention growth on the repeated-observation stress task (log–log). Lines: means; bands: ±1 SD over three seeds (full-history trio).

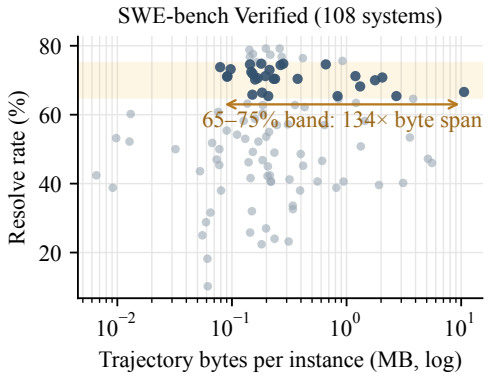


Figure 3: SWE-bench Verified, 108 instance-normalized systems: resolve rate vs. exported trajectory bytes per instance (log). No detectable volume–success correlation (τ_b=0.080, p=0.222).

reaches 120 while D=1.0, illustrating why probe matches and exact-chunk duplication provide complementary views of repetition.

5.3 Persistence Ablations

Retention is not required for immediate task success on this short retrieval suite: with persistence disabled, five frameworks (LangGraph, AutoGen, LlamaIndex, OpenAI Agents, Agno; five-task subset) keep their accuracy (100% for four; AutoGen within run variance: 22/25 vs. 19/25) and retain exactly zero bytes. Retention supports replay, resumption, and recovery rather than immediate task completion, and the ablated configurations—like SmolAgents' default—provide none (R=0). InfiAgent exposes no debug-channel switch, so we perform a post-hoc ablation by excluding its debug channel, which accounts for 40% of its footprint; removal *raises* D 1.75→2.1.

5.4 Reconstructability

Scoring one run per (framework, task) pair (80 runs), six frameworks achieve R=3 at retained sizes spanning 15×; CrewAI retains bytes without per-call structure (R=1), SmolAgents nothing. InfiAgent's raw-trace channel supplies what its truncated debug log lacks; multi-channel designs need channel-level scoring. We validate the grade by *conversation-content reconstruction*: each R=3 framework re-ran three tasks behind a logging proxy recording every request body (194 calls); we rebuilt the conversation from the retained bytes and compared field-by-field (roles, contents, tool-call ids/arguments, results, order): with call boundaries taken from the proxy log, 194/194 calls are reconstructible as contiguous subsequences after normalization (contracts in Appendix I). System prompts are recoverable from retention in Agno and InfiAgent; the rest keep them in code-side configuration (LangGraph configures none). A *process-boundary resume probe* (Appendix J) confirms the grades operationally: every R=3 store resumes in a fresh process; CrewAI's (R=1) does not.

Appendix S plots default and post-CAS retention with R and accuracy. The same reconstructability score, R=3, is obtained at retained sizes from 0.33 MiB (OpenAI Agents) to 5.10 MiB (LangGraph), and content addressing reduces all seven persisting configurations to 0.05–0.50 MiB (0.06–0.50 MiB among the R=3 configurations) through deduplication and compression. Across all six R=3 configurations, minimal history-retaining forms span 3.9×: three executed variants, two native forms, and one post-hoc channel split (Appendix Q). LangGraph and LlamaIndex's executed minimal history-

retaining forms converge at 0.31 MiB, close to OpenAI Agents' native 0.33 MiB, with accuracy and R unchanged (Appendix Q). Accuracy alone does not expose these storage differences: six frameworks score 100%, and Agno scores 99.3%. A success-only leaderboard penalizes AutoGen for its lower accuracy (86%) but not for its 2.49 MiB footprint, and it treats both LangGraph and OpenAI Agents as fully successful despite their $15.7\times$ retention difference. Section 6 examines exported trajectory volume across public agent submissions.

5.5 Beyond Adversarial Retrieval: Exploratory and Heterogeneous Tasks

In five multi-document tasks, agents choose which files to inspect through `list_files`, without prescribed re-reading. Dispersion persists (23/28 pairwise orderings agree; persisting frameworks span $8.1\times$) and echo *rises* (LangGraph $8.1\rightarrow11.8\times$). AutoGen and SmolAgents achieve 47% and 67% accuracy, respectively, and are excluded from equal-accuracy comparisons (Appendix E).

The two heterogeneous families (§4.1; all eight frameworks, 80 runs; Appendix D) require a correct answer *and* delivery of the specified artifact in the shared workspace. Five configurations achieve full joint success on both families (LangGraph, CrewAI, OpenAI Agents, LlamaIndex, and Agno) and span $6.9\times$ in retention on edit tasks (24–164 KiB) and $14.1\times$ on data tasks (8.8–124 KiB); answer-only accuracy is broader (seven frameworks reach 100% on edit) but conflates delivered and undelivered work. LangGraph's checkpointer echoes the input CSVs $7.8\times$ on data tasks. InfiAgent answers every question correctly yet fails joint success on all ten tasks, because it deposits the deliverable in its managed home-side store rather than the shared workspace. The workspace/home split of §3 makes this visible. AutoGen's default tool-iteration budget again fails the read–transform–write chain (0% on both families).

5.6 Fixed-Trace Control: Persistence under Fixed Content

The live-agent suites entangle agent policy with storage encoding. A scripted mock endpoint replays fixed content (three reads, 182.7 KiB of observations, one answer) through the seven persisting frameworks' adapters. Identical-content amplification spans $1.03\times$ (LlamaIndex) to $6.86\times$ (InfiAgent) under batched transport, a $6.7\times$ spread. Sequential transport raises LangGraph to $7.40\times$ (Appendix L). The control fixes scripted observations and responses; tool schemas and termination remain framework-specific. It measures which objects each configuration stores, how often, and in what serialization. The protocol is deterministic, offline, and model-free.

6 Storage in the Wild

Public trajectory exports provide a complementary view of what agent developers retain for evaluation. We harvested all 134 SWE-bench Verified (Jimenez et al., 2024) submissions in our July-2026 snapshot; keys are grouped by canonical instance ID (S3 layouts vary). Of 115 submissions with S3 prefixes, 113 expose size listings and 108 meet the 90%-coverage inclusion thresholds (instances and mapped bytes); 19 lack a public prefix, seven cannot be instance-normalized. The measured volume reflects each system's export format and logging policy.

Mean exported trajectory volume per mapped instance ranges from 6.6 kB to 10.6 MB across systems—$1{,}617\times$ (Figure 3). In the direct test, resolve rate shows no detectable correlation with bytes per instance in this sample (Kendall $\tau_b{=}0.080$, permutation $p{=}0.222$). The 28 systems in the 65–75% band alone span $134\times$ (IQR 152–703 kB; the spread is not driven solely by extremes). We account for repeated submissions (63 families; 25 multi-entry) with latest-per-family and family-mean analyses. Both yield the same conclusion ($\tau_b{\leq}0.074$, $p{\geq}0.40$). Detailed inspection of six sampled submissions shows intra-trajectory duplication from 1.0 (deduplicated exports) to 3.35.

7 A Reference Mitigation: Content-Addressed Trajectory Store

We evaluate content addressing with zstd as a reference for compact post-run storage. Our framework-agnostic implementation (about 300 lines) post-processes each run separately: strings and blobs of at least 1 KiB inside JSON, JSONL, log lines, and SQLite cells are moved into a

zstd-compressed content-addressed store and each is replaced by a hash reference with preview. Rehydration inverts the transform on demand and restores each framework's store schema.

Across 70 runs of the seven persisting frameworks, the compactor reduces retention by 4.8–$32.7\times$ (Table 1; Appendix H), and all 310 retained files pass the restore checks (JSON-object equality for structured text; row-set equality for SQLite user tables, internal/FTS shadow tables excluded as regenerable; pass-through byte-exact). Re-scoring R on the CAS-restored stores reproduces every framework's original score ($R{=}3$ for all six, $R{=}1$ for CrewAI). The compactor achieves its largest reduction on AutoGen ($32.7\times$), whose event log and per-turn snapshots repeatedly store the same messages. The measured gain combines shared-payload deduplication and per-payload compression.

Relation to whole-store compression. Whole-store zstd compression gives a smaller representation than CAS (LangGraph: ≈ 0.04 vs. 0.50 MiB), but answers a different question. C is a compression bound over concatenated files, not an archive. CAS blocks remain hash-addressable and can be restored on demand; the two compose (Appendix H).

The write-task suite quantifies how far defaults are from this reference point: a 39-byte summary file (a deliberately small write task) leaves 128 KiB–1 MiB of residue, up to $24{,}033\times$ the artifact (the production case of §1, with realistic outputs, shows $\sim 30\times$). With the ablations, these bracket the design space: retaining nothing yields no reconstructability, while defaults can retain megabytes even though the same score is achievable within 50–500 KiB after post-processing.

8 Discussion and Limitations

Configuration and workload. The controlled results describe the measured framework versions on retrieval and file-interaction tasks. Qwen3.6-27B preserves the storage ranking ($\rho{=}0.96$), with sizes 0.93–1.06$\times$ those of the main backend. Qwen2.5-7B also preserves much of the ranking ($\rho{=}0.89$), although context-window errors shorten some executions and reduce their retained volume (Appendix F). Containerized and sandbox measurements differ by 0–8.5%. Within AutoGen, an end-of-run save variant retains 2.1 versus 2.49 MiB, with fitted $\alpha{=}1.50$.

From individual runs to deployments. Persistence policy affects both what a run retains and how later tasks add to it. LangGraph's ten-task shared thread retains 95.2 MiB, compared with 5.8 MiB across ten fresh sandboxes (Appendix M). On file-QA, its minimal history-retaining form uses about one-sixteenth of the default's bytes at equal R (Appendix Q). Session reuse and retention policy therefore affect the capacity needed for deployment alongside framework and backend selection.

Recovery and storage lifecycle. R measures conversation-content recovery, while workflow state and audit provenance provide additional capabilities (Appendix R). The CAS experiment shows post-run savings with per-run object stores. Applying content addressing during execution would require write-path integration and corresponding deletion and recovery policies (Appendix A).

Resource costs. Storage retention entails both capacity costs and lifecycle obligations. TrendForce reported an approximately 80% quarter-on-quarter increase in enterprise SSD contract prices in Q1 2026 (TrendForce, 2026). Deleting state reduces storage but forfeits its supported capabilities (§5.3).

9 Conclusion

Agent evaluations report task success, reliability, and inference cost, but not persistent local storage. AgentFootprint makes this axis reportable through a serialization-aware suite, controlled and in-the-wild studies, and a reconstruction-preserving store. Equal-accuracy persisting defaults differ by $15.7\times$, while the fixed-trace control shows a $6.7\times$ configuration-level spread. The restoration experiments further show that recorded conversation content can be reconstructed without megabytes of retained data. Reporting storage together with accuracy, inference cost, and reconstructability makes the operational consequences of agent design choices visible before deployment.

AI USE STATEMENT

The authors prepared the initial manuscript, specified the benchmark, built and modified the code framework, and reviewed the data. Claude (Anthropic) and Codex (OpenAI) provided manuscript polishing, coding, and data-analysis assistance on author-provided materials. The authors reviewed and revised the manuscript and AI-assisted changes, checked the results against the code and data, and take responsibility for all content and released artifacts. The synthetic task data were produced by deterministic generators, not by LLM sampling.

ETHICS STATEMENT

The controlled corpora are synthetic and contain no personal data. Public trajectory analysis uses published submission metadata, object-size listings, and a limited sample of publicly released trajectories. Production measurements were collected with operator authorization and de-identified at the source. Released production records contain aggregate counts and categories rather than message content or business data. Content-addressed storage introduces privacy and deletion considerations described in Section 8 and Appendix A.

REPRODUCIBILITY STATEMENT

The anonymous supplement contains the harness, task generators, eight framework adapters, pinned dependencies, per-run measurements, and analysis scripts. Appendix C accounts for 1,051 executions and ten post-hoc channel variants. Appendices I and J specify reconstruction and continuation checks; Appendices K and N document measurement calibration and accounting audits. Representative retained stores allow independent remeasurement, and the fixed-trace mock endpoint provides an offline conformance path. The ICLR figures and summary checks are regenerated from the archived results without model access.

REFERENCES

Agno AGI. Agno. `https://pypi.org/project/agno/2.7.1/`, 2026. Version 2.7.1; accessed: 2026-07-13.

Zhuofu Chen, Rui Pan, Yinwei Dai, and Ravi Netravali. Slipstream: Trajectory-grounded compaction validation for long-horizon agents, 2026. arXiv:2605.08580.

Yann Collet and Murray Kucherawy. Zstandard compression and the 'application/zstd' media type. RFC 8878, RFC Editor, February 2021.

CrewAI Inc. CrewAI. `https://pypi.org/project/crewai/1.15.2/`, 2026. Version 1.15.2; accessed: 2026-07-13.

Susan B. Davidson and Juliana Freire. Provenance and scientific workflows: Challenges and opportunities. In *Proceedings of the ACM SIGMOD International Conference on Management of Data*, pp. 1345–1350. ACM, 2008. doi: 10.1145/1376616.1376772.

Melanie Herschel, Ralf Diestelkämper, and Houssem Ben Lahmar. A survey on provenance: What for? what form? what from? *The VLDB Journal*, 26(6):881–906, 2017. doi: 10.1007/s00778-017-0486-1.

Hugging Face. smolagents. `https://pypi.org/project/smolagents/1.26.0/`, 2026. Version 1.26.0; accessed: 2026-07-13.

Carlos E. Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik R. Narasimhan. SWE-bench: Can language models resolve real-world GitHub issues? In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Sayash Kapoor, Benedikt Stroebl, Zachary S. Siegel, Nitya Nadgir, and Arvind Narayanan. AI agents that matter. *Transactions on Machine Learning Research*, 2025.

Minjae Kim, Jinheon Baek, Soyeong Jeong, and Sung Ju Hwang. MemRefine: LLM-guided compression for long-term agent memory, 2026. arXiv:2606.13177.

LangChain AI. LangGraph. https://pypi.org/project/langgraph/1.2.8/, 2026. Version 1.2.8; accessed: 2026-07-13.

LlamaIndex Inc. LlamaIndex. https://pypi.org/project/llama-index-core/0.14.23/, 2026. Core version 0.14.23; accessed: 2026-07-13.

Grégoire Mialon, Clémentine Fourrier, Craig Swift, Thomas Wolf, Yann LeCun, and Thomas Scialom. GAIA: A benchmark for general AI assistants. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2024.

Microsoft. AutoGen. https://pypi.org/project/autogen-agentchat/0.7.5/, 2025. Version 0.7.5; accessed: 2026-07-13.

Mahmoud Mohammadi, Yipeng Li, Jane Lo, and Wendy Yip. Evaluation and benchmarking of LLM agents: A survey. In *Proceedings of the 31st ACM SIGKDD Conference on Knowledge Discovery and Data Mining V.2*, pp. 6129–6139. ACM, 2025. doi: 10.1145/3711896.3736570.

Patrick O'Neil, Edward Cheng, Dieter Gawlick, and Elizabeth O'Neil. The log-structured merge-tree (LSM-tree). *Acta Informatica*, 33(4):351–385, 1996. doi: 10.1007/s002360050048.

OpenAI. OpenAI Agents SDK. https://pypi.org/project/openai-agents/0.18.0/, 2026. Version 0.18.0; accessed: 2026-07-13.

Abdelghny Orogat, Ana Rostam, and Essam Mansour. Understanding multi-agent LLM frameworks: A unified benchmark and experimental analysis, 2026. arXiv:2602.03128v1.

Kirk Rodrigues, Yu Luo, and Ding Yuan. CLP: Efficient and scalable search on compressed text logs. In *15th USENIX Symposium on Operating Systems Design and Implementation (OSDI 21)*, pp. 183–198. USENIX Association, July 2021. ISBN 978-1-939133-22-9.

TrendForce. AI agent boom triggers enterprise SSD supply crunch; top five enterprise SSD brands post record US$18.46 billion revenue in 1Q26, says TrendForce. https://www.trendforce.com/presscenter/news/20260611-13092.html, 2026. June 11. Accessed September 8, 2026.

Ningning Wang, Xavier Hu, Pai Liu, He Zhu, Yue Hou, Heyuan Huang, Shengyu Zhang, Jian Yang, Jiaheng Liu, Ge Zhang, Changwang Zhang, Jun Wang, Yuchen Eleanor Jiang, and Wangchunshu Zhou. Efficient agents: Building effective agents while reducing cost, 2025. arXiv:2508.02694.

Yifei Wang, Ziteng Wang, Yuling Shi, Silin Chen, Xinrui Wang, Yueqi Wang, Beijun Shen, Linjing Li, Xiaodong Gu, Julian McAuley, and Daniel Dajun Zeng. Context compression for LLM agents: A survey of methods, failure modes, and evaluation. Preprints.org, 2026. doi: 10.20944/preprints202605.2065.v1.

Wen Xia, Yukun Zhou, Hong Jiang, Dan Feng, Yu Hua, Yuchong Hu, Yucheng Zhang, and Qing Liu. FastCDC: A fast and efficient content-defined chunking approach for data deduplication. In *2016 USENIX Annual Technical Conference (USENIX ATC 16)*, pp. 101–114, Denver, CO, June 2016. USENIX Association. ISBN 978-1-931971-30-0.

Binyan Xu, Haitao Li, and Kehuan Zhang. LLM agents are latent context managers: Eliciting self-managed context via state proprioception, 2026. arXiv:2606.30005.

Shunyu Yao, Noah Shinn, Pedram Razavi, and Karthik R. Narasimhan. τ-bench: A benchmark for tool-agent-user interaction in real-world domains. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2025.

Chenglin Yu, Yuchen Wang, Songmiao Wang, Hongxia Yang, and Li Ming. InfiAgent: An infinite-horizon framework for general-purpose autonomous agents. In *Findings of the Association for Computational Linguistics: ACL 2026*, pp. 35884–35894, San Diego, California, United States, 2026. Association for Computational Linguistics. doi: 10.18653/v1/2026.findings-acl.1787.

Yunbei Zhang, Janet Wang, Yingqiang Ge, Weijie Xu, Jihun Hamm, and Chandan K. Reddy. Stop comparing LLM agents without disclosing the harness, 2026. arXiv:2605.23950.

A Availability and Artifact Contents

The anonymous code-and-data supplement contains the harness, task generators, eight framework adapters, CAS implementation, and analysis scripts under the MIT license. It includes synthetic corpora, per-run measurements, reconstruction and CAS outputs, growth and write-task results, Docker checks, and cached public-trajectory statistics. A README, pinned dependencies, and version manifest describe how to reproduce the analyses.

One retained store per persisting framework supports remeasurement and restoration. Five are byte-exact. CrewAI's released example omits a credential key and user/device metadata (312 bytes), with separate post-removal measurements and CAS restoration checks; its task-output database is unchanged. InfiAgent's key-material placeholders and operator paths use length-preserving substitutions with rechecked measurements. Per-framework lockfiles identify exact package versions.

The SWE-bench data comprise public submission metadata, results files, object-size listings, and sampled trajectories from six submissions. The release includes the collection and instance-normalization scripts, cached tables, inclusion diagnostics, and statistical analyses. Synthetic corpora contain no personal data and use the repository's MIT license. The SWE-bench-derived tables retain the terms of their public sources.

Verification scripts. The artifact provides meter calibration, R construct checks, and regeneration of the file-QA and heterogeneous-task results. The fixed-trace protocol runs against an offline mock endpoint. The lockfiles cover seven frameworks in a container environment; InfiAgent installs from PyPI. The `envlocks/` directory records study versions.

CAS deployment obligations. Content hashes can reveal equality across trajectories or tenants. Shared blobs also require reference-aware deletion and collection of unreferenced content. Deduplication across trust boundaries can enable existence probes against predictable content. Deployments should isolate tenants, protect sensitive equality information, and couple deletion with reference-counted collection. The evaluated compactor creates a separate object store for each run, so the reported reductions use only within-run sharing. SQLite restoration preserves tested user-table rows rather than the original database-file bytes or image hash; forensic preservation of the original file requires a byte-preserving archive.

B Per-Framework Composition and Persistence Configuration

Table 2 gives the numeric decomposition behind Figure 4, together with each framework's persistence mechanism as configured in the harness (the exact adapter code ships in the artifact).

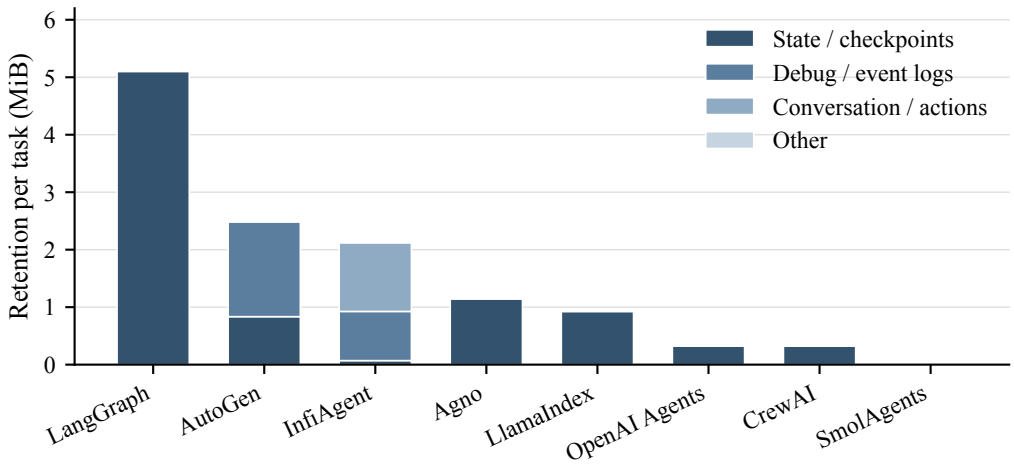


Figure 4: Composition of retained bytes on the file-QA suite. Frameworks are heavy in different places: checkpoints (LangGraph), debug/event logs (AutoGen), conversation stores (InfiAgent).

Table 2: Retained MiB per file-QA run by store category (mean over 30 runs), and the persistence mechanism measured.

Framework	State	Dbg/log	Conv.	Mechanism
LangGraph	5.10	–	–	SQLite checkpointer
AutoGen	0.83	1.65	–	per-turn state, event log
InfiAgent	0.07	0.86	1.20	task stores, raw traces
Agno	1.15	–	–	SQLite session db
LlamaIndex	0.93	–	–	serialized `Context`
OpenAI Agents	0.33	–	–	`SQLiteSession`
CrewAI	0.32	–	–	task-output store
SmolAgents	–	–	–	none (in-memory)

Channel asymmetries. AutoGen's recipe combines per-turn `save_state` with the tutorial's event logger. Removing the event log leaves 0.83 MiB per file-QA task. Both channels grow superlinearly on the stress task, with $\alpha = 1.70$ for the log and 1.86 for the snapshots. The log records each turn's full request. OpenAI Agents' hosted tracing is disabled because it stores data remotely, outside the measured local boundary.

Write-task residue by framework. Mean residue over the five write tasks: LangGraph 920 KiB, InfiAgent 1,022 KiB, AutoGen 374 KiB, Agno 285 KiB, OpenAI Agents 149 KiB, CrewAI 136 KiB, LlamaIndex 128 KiB, SmolAgents 0 B; per-run records ship in the artifact.

C Run Accounting

Table 3 accounts for 1,051 executions and ten post-hoc channel variants: 499 main-study executions and 552 replications and executed variants. The latter include two full-roster backends, additional growth seeds, executed history-retaining forms, continuing sessions, and two further proxied tasks per $R = 3$ framework. The original proxied task is included in the main-study subtotal. Each fresh sandbox has a baseline snapshot and measurement record in the artifact. The ten InfiAgent channel-ablation records are derived by copying existing first-repetition stores and removing debug channels, without rerunning the agent. They are counted separately from executions.

Computing infrastructure. All controlled runs executed on one Apple M5 workstation (16 GB RAM, macOS 26.4, arm64; Python 3.12), one framework process at a time; each per-run record stores the platform and Python version, and `envlocks/` pins framework and library versions. The measurements report retained bytes rather than storage-device timings.

Statistical units and aggregation. The framework–task pair is the primary comparison unit; the three repetitions quantify run-level variability. Recorded model calls (Section 5.4) and retained files (Section 7) are verification objects, not independent samples, and the 108 in-the-wild submissions carry family-level dependence, which Section 6's sensitivity analyses address. Ratio metrics (D, echo, C) are computed per run and then averaged. Table entries describe the expected per-run ratio. Decomposing Table 1's dispersion, between-task SD of task means versus mean within-task repetition SD is, in MiB: LangGraph 0.27/0.36, AutoGen 0.16/0.25, InfiAgent 0.03/0.02, Agno 0.01/0.01, LlamaIndex 0.00/0.00, OpenAI Agents 0.01/0.01, CrewAI 0.00/0.00. Repetition variability dominates for the two heaviest frameworks; task-content variability is second-order elsewhere. The in-the-wild study adds no sandboxed runs; it harvests public metadata for 134 submissions, of which 108 pass instance normalization. SmolAgents retains zero bytes at every measured horizon; its $T{=}200$ point was not run. A nonparametric bootstrap (20,000 resamples of run means within each framework) puts a 95% CI of $[15.0, 16.3]$ on the headline $15.7\times$ file-QA spread. The executed history-only $3.9\times$ is a ratio of 10-task means (per-configuration SDs in Appendix Q); the fixed-trace $6.7\times$ comes from a deterministic single-trajectory protocol and is reported as a point estimate.

For the write-suite comparison, we divide the mean residue by the mean delivered-file size over the five tasks. LangGraph's values are 942,080 bytes and 39.2 bytes, giving 24,032.65. The main text rounds this ratio to 24,033. This is an aggregate comparison rather than a maximum over runs.

The CAS experiment uses the first repetition of each file-QA task, giving 70 runs over seven persisting frameworks. The retention figure compares the default means over three repetitions with this CAS subset.

Table 3: Execution accounting. The ten post-hoc records derive from file-QA executions and are listed separately.

Experiment	Count	Composition
File-QA main study	240	8 fw $\times$ 10 tasks $\times$ 3 reps
Persistence ablations	25	5 frameworks $\times$ 5 tasks
AutoGen end-of-run save	8	5 file-QA + 3 horizon points
Second backend (4o-mini)	18	3 fw $\times$ 5 QA + 3 at $T{=}50$
Replay-verification source	6	1 proxied task per $R{=}3$ fw
Long-horizon	31	7 fw $\times$ 4 T + SmolAgents $\times$ 3
Write tasks	40	8 fw $\times$ 5 tasks
Exploratory retrieval	40	8 fw $\times$ 5 tasks
Edit family	40	8 fw $\times$ 5 tasks
Data-analysis family	40	8 fw $\times$ 5 tasks
Docker cross-check	3	LangGraph $\times$ 3 tasks
Fixed-trace (batched)	7	7 fw $\times$ 1 scripted trace
Fixed-trace (sequential)	1	LangGraph transport variant
Main-study executions	499	
Backend: Qwen3.6-27B	240	8 fw $\times$ 10 tasks $\times$ 3 reps
Backend: Qwen2.5-7B	240	8 fw $\times$ 10 tasks $\times$ 3 reps
Growth repetition seeds	24	3 fw $\times$ 4 T $\times$ 2 seeds
Executed history-only	30	3 configurations $\times$ 10 tasks
Continuous shared threads	6	6 fw $\times$ ten-task thread
Replay-source additions	12	2 further proxied tasks $\times$ 6 fw
Total executions	1,051	
Post-hoc channel variants	10	InfiAgent $\times$ 10 existing file-QA runs
Total evaluation records	1,061	Executions plus post-hoc variants

D HETEROGENEOUS TASK FAMILIES: PER-FAMILY RESULTS

Tables 4 and 5 give the per-framework results behind Section 5.5 (one run per task, five tasks per family; results are descriptive and no repetition variance is reported; S_{total} is the mean over the five tasks and includes the delivered artifact itself, under 1 KiB in every case). "Acc" is answer accuracy; "Artifact" counts tasks whose required artifact passes a *spec-exact* validator in the shared workspace (edit: every DATE line ISO and set-equal to the source ledger; data: exact header, all product totals correct, descending order); "Joint" counts tasks where both hold. This is the task-success criterion in Section 5.5. An earlier presence-based checker over-credited SmolAgents' data artifacts (3/5); spec-exact grading yields 1/5, and no other cell changes. InfiAgent's home-side artifacts pass the spec-exact check as well.

Table 4: Edit family (normalize ledger dates, deliver corrected copy). [a]InfiAgent's artifacts are all content-correct but deposited in its managed home-side store rather than the workspace. [b]AutoGen exhausts its default tool-iteration budget before the write step.

Framework	Acc	S_{total} (KiB)	D	Echo	Artifact	Joint
LangGraph	100%	164	1.6	4.8×	5/5	5/5
AutoGen[b]	0%	6.1	1.0	0.0×	0/5	0/5
InfiAgent[a]	100%	287	1.2	2.9×	0/5	0/5
CrewAI	100%	41	1.0	1.2×	5/5	5/5
SmolAgents	100%	2.9	1.0	0.2×	4/5	4/5
OpenAI Agents	100%	44	1.0	0.8×	5/5	5/5
LlamaIndex	100%	24	1.0	0.9×	5/5	5/5
Agno	100%	66	1.0	1.4×	5/5	5/5

Table 5: Data-analysis family (aggregate two sales CSVs, deliver `summary.csv`). Footnotes as in Table 4.

Framework	Acc	S_{total} (KiB)	D	Echo	Artifact	Joint
LangGraph	100%	124	1.4	7.8×	5/5	5/5
AutoGen[b]	0%	6.4	1.0	0.0×	0/5	0/5
InfiAgent[a]	100%	284	1.2	0.0×	0/5	0/5
CrewAI	100%	16	1.0	1.0×	5/5	5/5
SmolAgents	60%	0.1	1.0	0.0×	1/5	1/5
OpenAI Agents	100%	33	1.0	1.1×	5/5	5/5
LlamaIndex	100%	8.8	1.0	0.0×	5/5	5/5
Agno	100%	47	1.0	0.0×	5/5	5/5

E EXPLORATORY-RETRIEVAL SUITE: PER-FRAMEWORK RESULTS

Table 6 gives the per-framework results behind Section 5.5 (five multi-document tasks, one run each).

Table 6: Unprompted exploratory retrieval: per-framework means over the five tasks.

Framework	Acc	S_{total} (MiB)	D	Echo
LangGraph	100%	3.01	7.8	11.8×
AutoGen	47%	0.62	3.2	2.5×
InfiAgent	100%	3.00	1.7	6.7×
CrewAI	100%	0.68	2.2	2.8×
SmolAgents	67%	0.00	–	0.0×
OpenAI Agents	100%	0.37	1.4	1.4×
LlamaIndex	100%	0.93	3.1	3.1×
Agno	100%	1.17	1.8	2.0×

F BACKEND REPLICATIONS

Full-roster replications (Qwen3.6-27B and Qwen2.5-7B). All eight frameworks re-ran the complete file-QA suite (10 tasks $\times$ 3 repetitions per backend; 480 fresh sandboxes) on two further backends served through the same provider channel as the main study. Under Qwen3.6-27B, per-framework S_{total} lands within 0.93–1.06$\times$ of the main study, accuracy is 98–100% throughout, and the persisting-framework ranking is preserved at Spearman ρ=0.964 (spread 16.9$\times$ vs. 15.7$\times$). Under the deliberately weak Qwen2.5-7B, task accuracy collapses for four frameworks (35–39%), and the per-answer records identify the mechanism: 80–94 of each collapsed framework's 150 answers are HTTP-400 errors from full-history prompts exceeding the endpoint's 32,768-token window as the 60 KiB source files accumulate. These failures are context-window errors. InfiAgent (windowed context) never hits the ceiling and holds 100%; LlamaIndex, whose default chat memory bounds the prompt while its store appends everything, holds 92%. Yet the storage ranking persists at ρ=0.893 (spread 12.0$\times$): the retention signature is visible even when the model fails the task. Windowing before the model request affects both retained history and robustness to a small context limit.

Table 7: Full-roster backend replications (mean per-task values; 30 runs per cell).

	Qwen3.6-27B				Qwen2.5-7B			
Framework	S_{total}	D	Echo	Acc.	S_{total}	D	Echo	Acc.
LangGraph	5.20	13.2	8.2$\times$	100%	2.49	9.1	3.9$\times$	35%
AutoGen	2.64	8.5	4.3$\times$	98%	1.26	6.3	2.1$\times$	39%
InfiAgent	2.13	1.8	1.7$\times$	100%	2.14	1.8	1.7$\times$	100%
Agno	1.16	2.5	0.8$\times$	100%	0.77	2.0	0.6$\times$	39%
LlamaIndex	0.94	4.2	1.3$\times$	100%	1.20	5.5	1.7$\times$	92%
OpenAI Agents	0.31	1.5	0.5$\times$	100%	0.21	1.4	0.3$\times$	35%
CrewAI	0.32	1.5	0.5$\times$	100%	0.37	1.7	0.6$\times$	81%
SmolAgents	0	–	–	99%	0	–	–	60%

Serving-channel note. One cell required a serving substitution: the sole API provider offering tool support for Qwen2.5-7B could not parse SmolAgents' zero-argument tool schema (the router returned "no endpoints found that support tool use" once that provider was excluded), so the SmolAgents$\times$7B cell was served from the same weights on a self-hosted vLLM instance; all other cells used the shared API channel. Ten Qwen3.6-27B SmolAgents runs that hit the harness wall-clock limit during a provider-congestion window were treated as invalid and re-executed in fresh sandboxes under an extended limit, per the benchmark's task-identity rule.

GPT-4O-MINI SUBSET (EARLIER PROTOCOL)

Table 8: GPT-4o-mini subset (earlier three-framework, five-task protocol, superseded by the full-roster replications above; both columns use that protocol, so the DS value 5.27 is not comparable to Table 1's ten-task 5.10). Rankings and D transfer; absolute bytes shift with model verbosity.

	File-QA S_{total} (MiB)		D	
Framework	GPT	DS	GPT	DS
LangGraph	5.35	5.27	13.6	13.4
AutoGen	2.33	2.50	7.9	8.3
OpenAI Agents	0.33	0.33	1.6	1.6

G GROWTH REPETITIONS AND MODEL COMPARISON

Repetition protocol. Every horizon of the three full-history frameworks (LangGraph, AutoGen, LlamaIndex) was rerun under two additional independent seeds (24 added runs; identical harness, backend, and judging), giving three runs per point; the remaining frameworks keep single runs. Table 9 reports per-horizon means. Two repetition cells failed the validity screen applied to every

run (return code, accuracy band, store size within the sibling family) and were rerun; the failures coincided with an upstream serving-pool incident on the shared backend model (one harness-level timeout, runs with 20–34% of turns receiving HTTP-200 responses with empty content, one stalled connection). The final rerun enabled an environment-gated transport guard (90 s per-request timeout with client-side retries; re-issue of a request whose response arrives empty). The guard is inactive by default, was off for all other accepted runs, and does not alter persistence behavior. The session history stores only the final response of each call. The rejected attempts are preserved in the release archive (`longhorizon_runs/_anomalies/`): a manifest of all five attempts plus each attempt's measurement, answer, and adapter-log records where the failure mode produced them.

Between repetitions, scatter concentrates in absolute volume while the exponent is stable (per-seed fits in Table 9): LangGraph's $T{=}100$ point varies by ± 10 MiB while its α varies by ± 0.01. On seed 1, local slopes over $T = 25 \to 100$ are 1.96–1.98 for all three full-history frameworks. Over $100 \to 200$, they diverge: LlamaIndex 1.92, AutoGen 1.38, and LangGraph 0.96, with no failed rounds and token volumes far below context limits. We suspect checkpoint-layer consolidation; compaction events were not directly instrumented.

Table 9: Per-horizon retention in MiB, mean $\pm$ SD over three seeded runs. Per-seed full-range α fits: LangGraph $\{1.72, 1.75, 1.74\}$, AutoGen $\{1.76, 1.97, 2.12\}$, LlamaIndex $\{1.95, 1.77, 1.93\}$.

Framework	$T{=}25$	$T{=}50$	$T{=}100$	$T{=}200$
LangGraph	10.1±0.0	29.6±6.3	152.6±10.3	323.2±25.6
AutoGen	1.7±0.4	9.8±0.2	34.9±3.7	101.7±20.8
LlamaIndex	1.1±0.0	3.7±0.8	14.7±3.2	56.3±11.5

Model comparison. Table 10 compares power-law, linear, and quadratic fits over four horizons. With up to three fitted parameters, a quadratic can closely interpolate these points. The fits describe the measured range; four horizons do not identify an asymptotic functional form. Archival policy is part of the configuration: LlamaIndex's latest-only footprint (newest snapshot per horizon) is 85, 163, 336, and 645 KiB at $T \in \{25, 50, 100, 200\}$, fitting $\alpha{=}0.98$ against $1.88{\pm}0.10$ for the append-all series in the measured recipe. The two archival policies produce different growth patterns over the measured horizons.

OpenAI Agents retains 124, 216, 424, and 540 KiB at these four horizons. For an affine process $S(T) = a + bT$ with positive fixed overhead a, the local log–log slope is $bT/(a + bT) < 1$. Thus a fitted exponent below one can occur even with constant marginal growth. This is a possible contribution to the observed fit, not an identification of its storage-engine mechanism; the measured increments also vary across intervals.

Table 10: Model comparison for retention growth. R^2 of power-law ($\log S_{\text{total}} \sim \alpha \log T$), linear ($S_{\text{total}} \sim a{+}bT$), and quadratic fits. α: mean $\pm$ SD of per-seed fits where three seeds exist, single-seed fit otherwise; R^2 columns are evaluated on the seed-1 series.

Framework	α	R^2 (power)	R^2 (linear)	R^2 (quadr.)
LangGraph	1.74±0.01	0.976	0.984	0.987
AutoGen	1.95±0.18	0.988	0.996	1.000
InfiAgent	1.15	0.999	0.996	1.000
Agno	0.98	1.000	1.000	1.000
LlamaIndex	1.88±0.10	1.000	0.967	1.000
OpenAI Agents	0.73	0.970	0.909	0.995
CrewAI	1.20	0.996	0.992	1.000

H Whole-Store Compression vs. Content Addressing

Table 11 compares four representations. The zstd bound is derived from default retention and compressibility in Table 1. It compresses a concatenated stream without preserving file boundaries. The

CAS column reports the compacted store, and CAS+zstd reports a real archive of that store using the same compressor settings. Both CAS columns use the first-repetition runs.

Exact repetition and near-duplicate content behave differently. For AutoGen and Agno, archiving the CAS output approaches the whole-store compression bound. LangGraph's checkpoints change slightly each turn, so exact content addressing keeps distinct versions as separate blobs. Its archived CAS representation is about ten times the bound. This gap identifies similarities that zstd exploits across objects but the exact-match CAS transform does not.

Table 11: Per-task retained MiB under four representations (file-QA suite). "zstd bound" is derived from Table 1's C (stream-concatenated compression bound, not a restorable archive); "CAS+zstd" is measured on a real tar stream.

Framework	Default	zstd bound	CAS store	CAS+zstd
LangGraph	5.10	0.038	0.50	0.389
AutoGen	2.49	0.023	0.07	0.025
InfiAgent	2.12	0.049	0.36	0.177
Agno	1.15	0.027	0.06	0.027
LlamaIndex	0.93	0.023	0.09	0.054
OpenAI Agents	0.33	0.024	0.07	0.039
CrewAI	0.32	0.023	0.05	0.044

Paired CAS reductions. Table 12 uses all 70 archived first-repetition CAS transformations, with ten paired tasks per framework and no exclusions. Each before/after pair refers to the same retained store. The headline factor is the ratio of the summed original bytes to the summed compacted bytes within each framework, equivalently the ratio of the two paired means. It is not calculated by dividing Table 1's three-repetition default mean by its first-repetition CAS mean. For example, AutoGen's paired means are 2.392 and 0.073 MiB, yielding $32.7\times$ on unrounded values.

The median, quartiles, and range summarize variation in individual task ratios. Quartiles use linear interpolation at $(n-1)p$, with $n = 10$ and $p \in \{0.25, 0.75\}$ (type 7). These are descriptive summaries of the fixed task suite, not confidence intervals or estimates of between-run variability. No hypothesis tests are applied. The artifact includes every pair and the standard-library Python script `summarize_paired_cas.py`.

Table 12: Paired CAS retention and per-task reduction factors. Before and after are means over the same ten first-repetition tasks, in MiB. Factor is the ratio of paired totals, as in Table 1. Median, IQR, and range refer to individual task factors; all factors are dimensionless.

Framework	Before	After	Factor	Median [IQR]	Range
LangGraph	5.193	0.505	10.3	10.38 [10.36–10.41]	9.52–10.43
AutoGen	2.392	0.073	32.7	32.88 [31.02–36.11]	26.51–36.34
InfiAgent	2.115	0.363	5.8	5.84 [5.83–5.85]	5.77–5.86
Agno	1.151	0.056	20.5	20.51 [20.48–20.55]	20.44–20.61
LlamaIndex	0.928	0.090	10.3	10.33 [10.31–10.37]	10.26–10.41
OpenAI Agents	0.329	0.069	4.8	4.74 [4.74–4.78]	4.73–4.80
CrewAI	0.325	0.053	6.1	6.12 [6.11–6.13]	6.09–6.14

I Conversation-Reconstruction Protocol

An HTTP proxy between each adapter and model provider records every request body as validation ground truth. Each of the six $R = 3$ frameworks ran three proxied tasks. The 18 runs contain 194 calls, with totals of 65, 64, and 65 for the three tasks across frameworks.

Extractors reconstruct conversation content from the retained stores. LangGraph uses checkpoint snapshots. AutoGen uses per-turn state, with its thought-field encoding normalized. OpenAI Agents uses session rows, merging text and function-call items from the same assistant turn. LlamaIndex uses serialized context, Agno uses session-run rows, and InfiAgent uses raw traces.

Each recorded non-system message list must be a contiguous subsequence of a rebuilt history. InfiAgent supplies separate sequences for individual calls. Equality covers roles, content, tool-call identifiers, names and arguments, tool results, and order. Empty text is normalized to null, and parseable JSON arguments are compared with sorted keys. A windowed sender such as Agno is checked against the window it actually sent. All 194 calls satisfy this contract. System prompts are recoverable from retention in Agno and InfiAgent; other configurations keep them in code or configure none.

The proxy supplies reference requests and call boundaries for validation; reconstructed message content comes only from retained stores. The test establishes conversation-content reconstruction under this contract. It does not establish byte-identical provider requests or self-describing call boundaries. The archive includes per-call reconstruction diffs. The retained units vary: InfiAgent supplies per-call request arrays, LangGraph/AutoGen/LlamaIndex supply saved states or snapshots, and OpenAI Agents and Agno supply session records. A saved state or session record need not correspond one-to-one to a model request. Validation therefore uses the same recorded-request boundary contract for all six.

J Process-Boundary Resume Probe

The reconstructability grade asks whether retained bytes can reconstruct history; this probe asks whether they can *continue* it. Phase A runs each framework's unmodified benchmark adapter in its documented durable-session configuration: the agent is told a nonce registry code, acknowledges it, and the process exits (the crash point is between turns). Phase B starts a fresh process on the same sandbox home, reattaches to the same documented session identity, and asks for the code; workspace files do not contain it, so only the retained store can supply it. Because the AutoGen and LlamaIndex adapters persist state without ever reloading it, their phase B exercises the frameworks' official restore APIs (`load_state`, `Context.from_dict`) in the fresh process.

Table 13: Process-boundary resume: can a fresh process continue the session from retained bytes alone?

Framework	R	Resumed	Phase-B mechanism
LangGraph	3	yes	`thread_id` + SQLite checkpointer
AutoGen	3	yes	`load_state` of saved state
OpenAI Agents	3	yes	`SQLiteSession`
LlamaIndex	3	yes	`Context.from_dict`
Agno	3	yes	`session_id` + session db
InfiAgent	3	yes	task identity + task store
CrewAI	1	no	task-output store; no session API
SmolAgents	0	–	nothing retained

All six $R = 3$ frameworks return the exact code from retention. CrewAI reports that it has no record of the code. SmolAgents retains no session to reopen. The probe tests conversation continuation after a process exit between turns. Mid-turn tool state and workflow position are outside this test.

K Meter Calibration on Known-Duplication Stores

Logical streams and chunking. The meter uses `fastcdc==1.7.0`, SHA-256, and minimum, target, and maximum chunk lengths of 1,024, 4,096, and 32,768 bytes, respectively, with the library's default cut-point normalization. Streams of at most 1,024 bytes are hashed as a whole. Each SQLite text/blob cell longer than 64 bytes is one stream; numeric and NULL cells are skipped. For newline-delimited `.jsonl/.ndjson`, each line longer than 64 bytes is one stream. If extraction yields no qualifying cells or lines, the file is used as one stream; a database-open failure also falls back to file bytes. Other nonempty files are whole streams. Both terms of D use these same streams, whereas S_{total} counts retained file lengths. Appendix N reports sensitivity to the extraction threshold.

Echo. Let P be the multiset of sampled substrings, L the retained logical streams, and $V(p)$ the distinct byte encodings of probe p. The meter reports

$$\text{echo} = \frac{1}{|P|} \sum_{p \in P} \sum_{s \in L} \sum_{v \in V(p)} \text{count}(s, v),$$

where count is the number of non-overlapping exact occurrences. For each UTF-8 baseline workspace file with at least 288 characters, twelve 72-character probes are selected. For file length n, probe $i = 0, \ldots, 11$ starts at

$$o_i = \left\lfloor \frac{h + i(n - 72)}{12} \right\rfloor \bmod (n - 72),$$

where h is the integer represented by the first twelve hexadecimal digits of SHA-256 of the workspace-relative path, including its `workspace/` prefix. The three candidate encodings are raw UTF-8 and JSON-escaped text with `ensure_ascii=False` and `ensure_ascii=True`, after removing the outer quotation marks; identical encodings are counted once. No whitespace or semantic normalization is applied. Unmatched probes contribute zero, including probes from unread files. Unchanged baseline source files are outside the retained search set. Thus echo measures repetition relative to the supplied corpus, combining retrieval coverage and storage repetition. A fixed probe set is used across frameworks for each task; reported means summarize runs rather than random probe resampling. Empty probe sets return an undefined value.

Whole-line probes for R. The candidate scorer separately selects up to six lines longer than 80 characters from each source file named in the task questions, using `random.Random(20260708)` and sorted file names. It searches raw, once-escaped, and twice-escaped forms, and checks the fraction of lines found in each structurally eligible channel. At least 90% coverage yields a candidate grade of 3. The subsequent field-by-field reconstruction test checks actual messages, tool arguments, and order (Appendix I).

Calibration. The calibration generator creates stores with known copy counts. It splits a 40 KiB probe-source file into ten 4 KiB blocks and persists them under different encodings. "Naive D" applies the same chunker directly to the file bytes. The generator and stores are included in the artifact.

Table 14: Meter calibration. Five copies per block except small-lines (eight). "part": under $\geq$doubly-nested encoding only probes free of escapable characters match, so echo is a lower bound there.

Scenario	Encoding	D_{true}	D	naive D	echo_{exp}	echo
single-esc	JSONL, `json.dumps`	5.0	5.0	2.1	5.0	5.0
sqlite-exact	SQLite text cells	5.0	5.0	1.6	5.0	5.0
double-esc	JSON inside JSON	5.0	5.0	2.4	part	1.7
triple-esc	three nesting levels	5.0	5.0	2.1	part	1.7
near-dup	1% byte edits per copy	$\approx$1	1.0	1.0	–	2.0
zlib-prefix	compressed growing prefixes	$\approx$1	1.1	1.0	0	0.0
small-lines	400-char lines $\times$8	8.0	8.0	1.0	–	8.0

Logical-stream D recovers each exact copy count, including short records handled by whole-stream hashing. Raw-byte chunking reports only 1.6–2.4 for the five-copy encodings. Echo is exact under single JSON escaping. Nested escaping leaves only probes without escapable characters matching, so echo becomes a lower bound. Near-duplicate and compressed content yield $D \approx 1$. Compressibility still reveals similarity in the near-duplicate case, where $C = 17.7$.

L Fixed-Trace Protocol: First Instantiation

Per-framework results behind Section 5.6. The mock endpoint adapts tool schemas and termination contracts per framework (InfiAgent terminates via its `final_output` contract); every framework

Table 15: Fixed-trace results (batched transport). "×/obs" = retained bytes per observation byte. [a]CrewAI executes the full trace but paraphrases the scripted final answer, failing exact match; retention covers the completed trace. SmolAgents retains nothing and is out of scope.

Framework	Trace	Answer	S_{total} (KiB)	×/obs
InfiAgent	✓	✓	1,254	6.86
CrewAI[a]	✓	–	624	3.42
LangGraph	✓	✓	592	3.24
AutoGen	✓	✓	555	3.04
Agno	✓	✓	404	2.21
OpenAI Agents	✓	✓	212	1.16
LlamaIndex	✓	✓	188	1.03

receives the same scripted content, comprising three reads, 182.7 KiB of observations, and one answer.

Two observations. First, identical-content encoding amplification spans $1.03\times$ (LlamaIndex: this single-question trace serializes one context snapshot, consistent with the latest-only analysis of Appendix G) to $6.86\times$ (InfiAgent's multi-channel stores), giving a $6.7\times$ spread with scripted content fixed. Second, transport shape matters for per-step checkpointers: under a sequential-transport variant (one tool call per model turn), LangGraph rises from $3.24\times$ to $7.40\times$. The checkpointer saves once per superstep. The batched and sequential variants are archived separately.

M Continuous Shared-Store Protocol

Fresh sandboxes maximize attribution but leave open whether a long-lived shared backend amortizes per-task cost. This protocol runs ten identical fixed-trace tasks against one sandbox and one documented session identity per framework, measuring cumulative retention after each task; marginal cost is $\Delta S(k) = S(k) - S(k{-}1)$.

Table 16: Ten identical tasks on one shared store (mock endpoint; all answers correct for every included framework).

Framework	$S(1)$	$S(10)$	$\Delta S(1)$	$\Delta S(10)$	Shape
LangGraph	592 KiB	95.2 MiB	592 KiB	18.5 MiB	rising
Agno	404 KiB	7.0 MiB	404 KiB	748 KiB	mildly rising
AutoGen	555 KiB	3.8 MiB	555 KiB	371 KiB	flat
OpenAI Agents	212 KiB	1.8 MiB	212 KiB	184 KiB	flat
LlamaIndex	188 KiB	188 KiB	188 KiB	0	constant

LangGraph's ten tasks retain 95.2 MiB in one shared thread versus 5.8 MiB in ten fresh sandboxes. Each task checkpoints an increasingly long history. LlamaIndex instead overwrites its context snapshot, while AutoGen overwrites snapshots and appends event records. These policies yield different cumulative and marginal curves.

InfiAgent's documented task identities must not be reused across invocations. The attempted shared-identity run fails this protocol and is excluded from the comparison. It reaches the 60-round cap, consistent with the documented contamination behavior. Its diagnostic record remains in the artifact.

N Meter Implementation Audit and Threshold Sensitivity

Delta accounting. The runs in this paper were captured with a size-inventory baseline: a pre-existing file was treated as unchanged when its size was unchanged, and a changed file was counted at its full post-run size. The audit covers 509 stored sandboxes: 499 main-study executions and ten post-hoc channel-variant copies. It checks both approximations against Section 3.1. (i) Pre-existing bytes inside changed baseline files total 31,952 B against an aggregate S_{total} of 1,404.7 MiB

Table 17: Construct check of the automated R scorer on synthetic stores built from a real task corpus. "3*": content-complete per-call stores with order or argument defects receive an automated 3 by design; those properties are verified by reconstruction, which compares roles, contents, tool-call ids, arguments, and order field-by-field.

Constructed store	Intended	Auto R	Verdict
True per-call record	3	3	correct
Aggregate blob, spoofed name	1	1	correct
Aggregate blob, neutral name	1	1	correct
Per-call, shuffled order	3*	3	insensitive*
Per-call, missing tool args	3*	3	insensitive*
Per-call, truncated observations	2	2	correct
Per-call, summaries only	2	2	correct
Zero retained data	0	0	correct

(0.0022% overall, 0.056% in the largest affected run). The reported rounded values are unchanged. (ii) Same-size in-place rewrites: every baseline workspace file was byte-compared against its pristine task corpus source; zero modifications across all 509 stored sandboxes. (iii) Only InfiAgent populates the home directory before the run (its setup scaffolding); its same-size baseline files were verified separately: 300 corpus copies byte-identical to their pristine sources and 150 configuration files byte-identical across repetition seeds. These checks found no undetected rewrites. The released meter additionally hashes every baseline file (SHA-256) and reports an increment-based `S_total_delta` alongside the published accounting, closing this gap by construction.

Stream-length threshold. For cell- and line-level extraction, only records longer than 64 B enter D. If none qualify, the meter falls back to the whole file (Appendix K). Numeric and NULL cells do not enter cell-level streams. Recomputing D on the file-QA first-repetition subset at thresholds of 0, 16, 32, 64, and 128 B changes no framework's value by more than 0.1 (LangGraph 13.3–13.4; all others flat to one decimal), so the threshold does not drive any conclusion.

Setup staging and boundary. The baseline snapshot is taken after framework-side setup, so bytes a framework stages *before* execution are excluded from S_{total}. Only InfiAgent stages: its setup copies the task corpus into its managed task directory (0.60 MiB mean on file-QA). Under a total-operational view (staging + retained), its file-QA mean rises from 2.12 to 2.72 MiB, above AutoGen's 2.49 MiB. The 15.7× LangGraph/CrewAI spread is unchanged. We keep staged copies of immutable task inputs out of S_{total} because they duplicate provided inputs rather than execution history; both accountings ship in the artifact. Sizes are apparent file lengths (`st_size`), not allocated blocks. Sparse-file and copy-on-write allocation are outside this accounting. The redirected boundary does not cover system temp directories or remote telemetry; a full-filesystem container audit is the extension path for adversarial submissions.

O AUTOMATED R SCORER: CONSTRUCT VALIDATION

The automated scorer identifies per-call structure from audited channel signatures and requires JSONL channels to parse as JSON-object records. Whole-line probes check observation completeness. Table 17 includes eight archived constructions, including the zero-retention regression case. Field order and tool arguments are checked by the separate reconstruction protocol.

The JSON-record gate rejects the aggregate store with a misleading filename. Shuffled messages and missing arguments still receive an automated candidate grade of 3, showing why the reconstruction check is needed. Framework-level grades rely on the validated extraction contract remaining applicable across tasks at the measured version and configuration. The proxy records supply ground truth for checking that contract.

P PRODUCTION DEPLOYMENT MEASUREMENT

With the operator's authorization we measured a production InfiAgent deployment (supply-chain quoting). Collection was whitelist-based and irreversibly de-identified at the source: only byte counts, file counts, channel categories, coarsened months, and tool-name frequencies were exported. No message content, file names, user identifiers, or business data left the environment (the collector ships in the artifact; tool names are pseudonymized in the released aggregates).

Deployment root. The collector inventories 436.7 MiB across the deployment root, including eight task directories. Channel composition: raw traces 148.0 MiB, service logs 98.5 MiB, task stores 71.6 MiB, other 69.4 MiB, debug logs 44.1 MiB, conversation stores 4.8 MiB, configuration 0.3 MiB. Trace, log, and debug channels hold 290.6 MiB (67%) while conversation stores hold 1%. The eight task directories together occupy 141.8 MiB; the remainder lies in shared directories. Their sizes range from 248 B to 81.4 MiB, with a median of 503.7 KiB. The sorted directory sizes, in bytes, are 248; 14,340; 201,294; 299,151; 732,420; 1,708,933; 60,433,749; and 85,331,853. The median averages the fourth and fifth values. The legacy collector's cached p50 instead selected the fifth value (715 KiB), and its p90 selected the maximum. Raw counts and the corrected descriptive summary are both retained in the artifact.

Deployment-wide scale. As of July 2026 the operator reports $\sim$56 GB of accumulated retained data across $\sim$1,000 recorded agent rounds for this deployment. This is an operator-attested aggregate over the full store, of which the 436.7 MiB root above is the collector-audited subset. "Round" is the deployment's unit of work and is not calibrated to our task definition, so we derive no per-task mean from it; the figure establishes practical scale in one deployment, not population prevalence.

Single-task decomposition. A separate task-level inventory totals about 131 MiB: a staged input workbook (42.4 MiB, in the framework's hidden upload area), framework residue of 79.2 MiB (raw trace 47.7, runtime logs 29.7, action log 1.5, history database 0.3), staged reference material (6.8 MiB), and 2.6 MiB of delivered artifacts. Removing 49.2 MiB of staged inputs from this inventory leaves 81.8 MiB of residue and deliverables. These are category totals from the production inventory rather than a controlled pre/post-run delta. Residue outweighs deliverables by about 30$\times$.

Workload profile. The task issued 184 model calls; tool-name frequencies are dominated by profile updates (219), program steps (62), resource lookups (50), and structured reads (32), with file writes rare. This workload is dominated by reading and processing. The de-identified aggregates ship in the supplementary archive.

Q HISTORY-RETENTION CHANNEL DECOMPOSITION (EXECUTED)

The system track measures documented configurations that support different capabilities (Section 4.2). This decomposition measures how much of the default spread is associated with channels beyond conversation-history retention, by *executing* each configuration's minimal history-retaining form on the full file-QA suite (10 tasks, single repetition, main-study backend). Three forms required a change and were re-run end-to-end: LlamaIndex latest-only archival (delete the previous snapshot after each write), AutoGen events-off (the documented switch: no event logger, state channel kept), and LangGraph newest-checkpoint pruning (an adapter-level wrapper deleting superseded checkpoint rows, then `VACUUM`). Functional equality is verified on all 30 runs: accuracy is unchanged (150/150 questions correct) and every retained store still scores R=3. OpenAI Agents and Agno are natively history-only (their main-study runs are the executed evidence); InfiAgent's channels expose no switch, so it remains a disclosed post-hoc split. This remains a decomposition of the default spread, not a capability-normalized efficiency ranking: the executed forms match on history reconstruction, not on every capability a default bundle serves.

Table 18: History-only forms on file-QA (mean over 10 executed tasks where re-run). Source: executed (re-run end-to-end, accuracy and R verified), native (default already history-only; main-study runs), split (post-hoc channel exclusion; channels not switchable).

Configuration	MiB	Source
LlamaIndex (latest-only archival)	0.31±0.00	executed
LangGraph (newest checkpoint)	0.31±0.03	executed
OpenAI Agents (session)	0.33±0.00	native
AutoGen (state, events off)	0.90±0.00	executed
Agno (session db)	1.15±0.00	native
InfiAgent (conversation channel)	1.20	split

Across these six history-retaining forms, the spread is $3.9\times$, compared with $15.7\times$ among equal-accuracy defaults. LangGraph, LlamaIndex, and OpenAI Agents fall within 7% of one another (0.31–0.33 MiB). The remaining spread reflects differences in retained channels and their encoding.

R Capability Matrix (Documentation-Based)

Table 19 summarizes what each documented durable-session configuration persists and which recovery capabilities it offers, per vendor documentation; "resume" is the process-boundary probe of Appendix J (measured), the other columns are documentation claims we did not independently test. This is the boundary behind Section 4.2: equal-R frameworks are comparable on history reconstruction, not on every column below.

Table 19: Persisted objects and recovery capabilities. Resume is measured here (Appendix J); Wf. (workflow-state recovery) and Ev. (event/debug provenance) are vendor-documentation claims, untested.

Framework	Persisted objects	Resume	Wf.	Ev.
LangGraph	graph checkpoints	yes	yes	–
AutoGen	state snaps + events	yes	partial	yes
OpenAI Agents	message session	yes	–	–
LlamaIndex	serialized context	yes	partial	–
Agno	session runs (window)	yes	–	–
InfiAgent	task store + traces	yes	yes	yes
CrewAI	task outputs	no	–	–
SmolAgents	none	–	–	–

S Reconstructability–Retention Figure

Figure 5 plots each framework's default retention against its post-CAS retention, annotated with R and accuracy; the reading is discussed in Section 5.4.

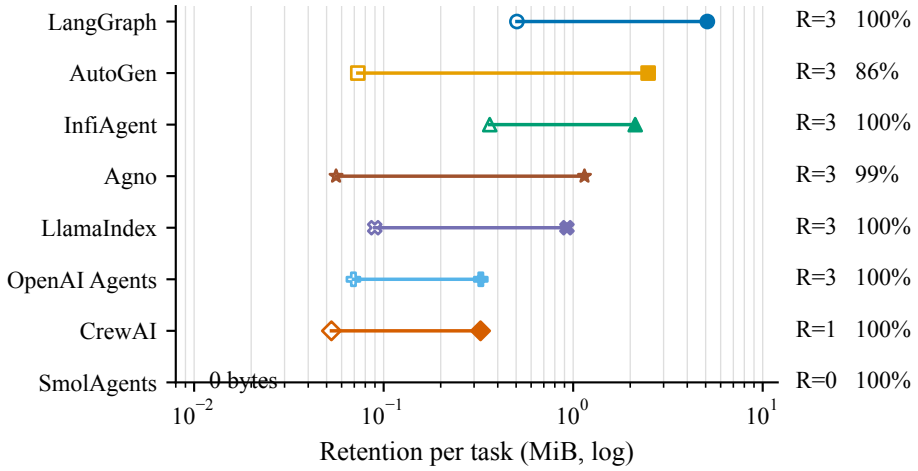


Figure 5: Reconstructability versus retention (discussed in §5.4 of the main paper). Filled: default retention per task; hollow: retention after the content-addressed store; right margin: R and accuracy. R=3 is obtained at 0.33–5.10 MiB under defaults and 0.06–0.50 MiB after content addressing (the 0.05 MiB post-CAS point is CrewAI, R=1).