Research Square

Preprints are preliminary reports that have not undergone peer review.
They should not be considered conclusive, used to inform clinical practice,
or referenced by the media as validated information.

Replication Instability in Experience Replay for Multi-Agent LLM Pipelines

Jianhao Wang
South China Normal University
Kaijia Wu
South China Normal University
Dong Liu
201003@gwng.edu.cn

South China Business College Guangdong University of Foreign Studies
Shouqiang Liu
South China Normal University

Research Article

Keywords: Multi-agent systems, Organizational memory, Coordination failure, Experience replay, Evaluation methodology, Replication

Posted Date: November 12th, 2026

DOI: https://doi.org/10.21203/rs.3.rs-11409743/v1

License: This work is licensed under a Creative Commons Attribution 4.0 International License.
Read Full License

Additional Declarations: No competing interests reported.

Replication Instability in Experience Replay for Multi-Agent LLM Pipelines

Jianhao Wang[1], Kaijia Wu[1], Dong Liu[2,*], Shouqiang Liu[1,*]

[1] School of Artificial Intelligence, South China Normal University, Foshan 528225, China

[2] School of Computer Science, South China Business College of Guangdong University of Foreign Studies, Guangzhou 510545, China

[*] Corresponding author: Dong Liu, 201003@gwng.edu.cn; Shouqiang Liu, liusq@m.scnu.edu.cn

ORCID: Jianhao Wang, 0009-0006-9764-8496; Kaijia Wu, 0009-0000-8134-2755; Dong Liu, 0009-0008-5311-0226; Shouqiang Liu, 0000-0002-2795-0685

Abstract. Experience replay, the re-injection of highly rated past outputs as in-context exemplars, gives large language model (LLM) agents memory without parameter updates. In a multi-agent system such a store is the organization's memory, and retrieval a coordination mechanism. Its effect is typically measured in a single stochastic run, which this paper shows cannot support the claim drawn from it. In a five-agent LLM pipeline, after repairing three coordination failure modes that had prevented the loop from closing, fifteen controlled comparisons across two models produced no replay effect that reproduces. Significance did not survive replication even where the sign did. Single-run evaluations produced large, significant effects of *opposite sign* on the two models (Δ=-0.172, p=0.041 and Δ=+0.147, p=0.040), and neither survived replication. A pool-size comparison that looked decisive in single runs did not survive either. Ten independent runs of one identical configuration, five per model, spanned a range of 0.138 and disagreed about significance and, on one model, about the sign. What relevance-matched retrieval can return is bounded by the resolution of its key rather than by the size of the store, and we prove the bound for key-dominated scoring; the parameter the literature reports is not the one that governs the outcome. An objective task with a deterministic answer key reproduced the instability. We conclude with a seven-point protocol to make such evaluations informative rather than merely confident. Code, data, and the full call log are available at https://github.com/wjh55224545/replication-instability-in-experience-replay.

Keywords Multi-agent systems · Organizational memory · Coordination failure · Experience replay · Evaluation methodology · Replication

Acknowledgments

We thank Prof. Meng Xiong and his team in the Department of Psychology, Hubei University, for independently evaluating the reports generated in this study, and Dr. Kehan Xiong, President of the Boze Institute for Talent Research, for his support of this work.

1. Introduction

Multi-agent systems increasingly coordinate through shared memory. Rather than updating model parameters, an agent stores past experience (prompts, outputs, outcomes) in a store and retrieves relevant entries to condition future behavior [1, 2, 3, 4]. In such a system the organization's memory also serves as a coordination mechanism, since it is how one agent's experience, or one working day's, is made to reach another. The mechanism is attractive for practical reasons. It requires no training, is simple to implement, and carries an intuitive justification, in that outputs humans rated highly ought to make future outputs better. It is now deployed widely enough that its effects are reported as findings about memory in agent systems generally.

That dual role creates a measurement problem. The mechanism is assessed by comparing outcomes across conditions, but each outcome is produced by a stochastic process, and the effect of the mechanism cannot be separated from the particular run that produced it unless the comparison is repeated. The pattern this paper examines is the one in which it is not. A single evaluation is compared against a baseline, a difference is observed, and the mechanism is reported as beneficial or harmful. Because the variance of that difference is never estimated, there is no way to tell a property of the mechanism from a property of the draw. We study this pattern because it is the one our own field uses when it evaluates a deployed pipeline over a fixed item set, and because its failure mode is demonstrable; we do not claim a base rate for how often it is used in place of repeated evaluation. The problem is acute when the expected effect is small relative to the spread of the output metric, which is the regime a context manipulation of this kind occupies.

The mechanism is organizational memory. The system we study is a five-agent pipeline of a kind that has become a mainstream paradigm for LLM-based multi-agent systems. Specialized roles execute in sequence under a central orchestrator, in the manner of MetaGPT and AutoGen [5, 6, 7]. Such a pipeline is a cooperative multi-agent organization in the classical sense, with agents that hold distinct capabilities and are coordinated by an explicit structure assigning roles and sequencing activity. Its experience store is the organization's *memory*, a repository of past experience that the collective consults when performing new tasks [8].

Reframing replay in these terms changes what the mechanism is expected to do, and yields predictions our experiments test. Two of them matter here.

First, a shared memory must discriminate among queries to contribute information. Transactive-memory theory holds that a group's performance depends not only on what its members know but on a directory specifying *who knows what*, and that such a directory is useful only insofar as it resolves distinct queries [9, 10]. An experience store queried by attribute matching is precisely such a directory. If the store is small relative to the diversity of queries, every query within a category resolves to the same few entries, and the store supplies repetition rather than information. The theory therefore predicts that the measured benefit of relevance-matched retrieval depends on store size, and that below some size a comparison between retrieval policies is uninterpretable. The reason is not that the policies are equivalent but that the directory cannot express a difference. We test this prediction in Section 5.3, and it fails in an instructive way. The directory does stop discriminating among queries, and retrieval returns the same handful of exemplars for most inputs. It does so at *every* store size we tried, however, because what bounds retrieval diversity is the resolution of the retrieval key rather than the number of entries behind it. The prediction located the constraint in the wrong quantity.

Second, failure of a feedback loop is a property of the information architecture rather than of agent competence. Coordination theory locates coordination failure in the structure that routes information, not in the reasoning of the parties being coordinated [11]. A loop implemented as separate store, filter, and retrieval components can fail to close when their semantics are misaligned, even though each component behaves as specified in isolation. This predicts that such failures are identifiable by inspecting component interfaces rather than by mining agent traces. Our audit (Section 3.3) is of that kind, and finds three.

The two predictions concern the organization rather than any agent's reasoning, and we return to what this implies in Section 6.

This paper reports what happened when we subjected one such mechanism to replication. The mechanism is implemented in a five-agent LLM pipeline that generates structured mental-health reports for school counselors. This is a high-stakes generation setting in which output quality, not task-completion accuracy, is the object of interest. The pipeline stores every generated report, accepts 1–10 ratings, and re-injects highly rated reports into the prompt as "historical successful cases".

Our study began as an attempt to improve this pipeline and became a study of why such improvements are hard to demonstrate. Three contributions follow.

1. **Mechanism.** We identify and repair three coordination failure modes of a multi-agent feedback loop. Feedback is routed to a category disconnected from the artifact it concerns; a quality filter implements semantics that differ from its apparent intent; and the retrieval policy is blind to the attribute the mechanism exists to exploit. Alongside these repairs we show that the diversity of what relevance-matched retrieval can return is bounded by the *resolution of its key* rather than by the size of the store, so that enlarging a store whose keys are coarse adds candidates to each key value without adding distinct key values (Section 3.3 and Section 5.3).

2. **Evidence.** Across fifteen controlled comparisons on two language models, no replay effect reproduces. Significance did not survive replication even where the sign did. Single-run evaluation produced significant effects of opposite sign; a pool-size comparison that had looked decisive in single runs dissolved under replication; and ten independent runs of one identical configuration, five on each model, produced estimates that disagreed about the magnitude of the effect and, on one model, about its direction (Section 5).

3. **Method.** We propose a seven-point evaluation protocol for memory mechanisms in agentic systems, intended to make such studies falsifiable rather than merely confident (Section 6.5).

The two predictions are tested in different parts of the paper. The first, that a shared memory must discriminate among queries to contribute information, is tested by the pool-size comparison (Section 5.3) and by the ablation of retrieval parameters (Section 5.7). It fails as stated, and the manner of its failure is what yields the key-resolution result. The second, that failure of a feedback loop is a property of the information architecture rather than of agent competence, is substantiated by the interface audit of Section 3.3, which finds three failure modes that no execution trace would show.

The consequence for this community is direct. Shared experience memory is how one agent's experience is made to reach another, and this literature reports on it from single runs. We show that at realistic effect sizes such reports cannot separate the mechanism's effect from the sampling process that produced the run. We also identify the design parameter these reports omit, namely the resolution of the retrieval key, and show that it governs what the mechanism can return at all, in place of the store size that is usually reported.

The claim is a narrow one. We do not claim that experience replay is universally ineffective, only that in this setting and under replication its effect is not measurable, and that evaluation practices common in this literature are not sufficient to detect that fact. Nor are the failure modes we report peculiar to this implementation. They are structural properties of how such loops are composed, as we argue in Section 6.

The remainder of the paper is organized as follows. Section 2 reviews related work, including the multi-agent systems literature on organizational memory and coordination. Section 3 describes the pipeline, the replay mechanism, and the three failure modes. Section 4 details the experimental design. Section 5 reports results. Section 6 discusses implications, boundaries, and limitations, and Section 7 concludes.

2. Background and Related Work

2.1 Memory and Experience Replay in LLM Agents

A substantial body of work equips LLM agents with external memory in place of parameter updates. Reflexion stores verbal self-feedback in episodic memory and retrieves it in later trials [1]. Self-Refine iteratively critiques and revises a single output [12]. Generative Agents introduce a memory stream with retrieval and reflection as the

substrate for long-horizon behavior [2]. Contextual Experience Replay accumulates and synthesizes past experiences into a dynamic buffer for in-context self-improvement of web agents [3]. Evo-Memory formalizes self-evolving memory as a search–synthesis–evolve cycle and provides a streaming benchmark [4]. In parallel, the older but closely related line of work on reinforcement learning from human feedback [13] establishes the value of human ratings for shaping model behavior, but does so through gradient updates rather than prompt injection.

Recent work has begun to qualify the optimism of this line. Hu et al. show that external memory does not resolve the stability–plasticity dilemma of continual learning but relocates it to the level of memory representation and retrieval design [14]. Their finding and ours are complementary, and together they point the same way. They show that the difficulty is not solved by adding memory, but moved into how memory is represented and consulted; we show that in a deployed retrieval mechanism the binding quantity is the resolution of the retrieval key rather than the volume of the store (Section 5.3). The joint implication is that the design parameters of a memory mechanism matter more than its capacity. This is the opposite of what the reporting conventions of this literature emphasize, since store size is commonly stated and the indexing scheme almost never is. Several recent studies report null or negative results for memory scaffolds in long-horizon agent tasks. Our study is consistent with this skeptical turn but differs in method. Rather than proposing a new memory design and evaluating it, we take a *fixed* replay design and quantify how unstable its measured effect is across replications, and we identify the design parameter that governs what the retrieval policy can return, namely the resolution of its key, which the literature does not report.

2.2 Role-Decomposed Multi-Agent LLM Systems and Their Failure Analysis

LLM-based multi-agent frameworks decompose tasks across specialized roles, arranging them either in conversation or in a workflow graph [5, 6, 7]. The system studied here belongs to the serial workflow variant of this paradigm, with four execution roles under a central orchestrator. It differs from the systems usually benchmarked in this literature in that its output is a natural-language artifact judged for quality rather than a task action judged for success. That difference matters for method, because quality judgments are compressed and noisy in a way that binary task success is not.

The concern that animates our study is not confined to this paradigm. Wei et al. study a coordination mechanism in multi-agent reinforcement learning and find that parameter sharing drives agents toward behavioral homogeneity, limiting the diversity of coordination the system can express; they counter it by having a language model generate semantic role labels that supervise contrastive role learning [15]. Our result is the mirror image, obtained in a different mechanism. Where they enrich the semantics of roles to buy diversity, we show that a retrieval key of fixed resolution places a mechanical ceiling on the diversity an experience store can deliver, and that enlarging the store behind that key does not raise the ceiling (Section 5.3). The two findings point at the same place. In both, the binding constraint sits in the representation the mechanism is indexed by, not in the agents that use it.

Systematic failure analysis for this class of systems is emerging. Section 2.4 discusses how our findings relate to the most prominent such catalog, MAST [16]. Briefly, the failures we report leave no trace signature and are therefore not of the kind that trace-based analysis recovers. Our audit yields one further observation, which we carry into the discussion. Repairing defects of this kind restores the intended semantics of the mechanism without producing a measurable improvement in output quality. "The loop now closes" is not evidence that the loop helps.

2.3 LLM-as-a-Judge for Evaluating Generated Text

Because expert evaluation of generated text is expensive, LLM-as-a-judge has become a standard proxy for quality. MT-Bench and Chatbot Arena show that strong models approximate human preference with reasonable agreement [17]; G-Eval uses chain-of-thought scoring [18]. A growing literature documents the limits of this proxy, including self-preference bias and weak agreement with human judgment on subjective dimensions [19], and statistical work has established limits to what judge-based comparison can resolve at a given sample size [20].

We contribute a concrete instance to this literature. In a subjective, high-stakes generation task, judge-based estimates of a context-manipulation effect are large in one run and absent in the next; judge identity shifts score levels by roughly 1.6 points on a ten-point scale, and two judges who agree moderately at the dimension level *reverse* their ordering of the same reports when the dimensions are averaged; and a third judge exhibited a ceiling effect that rendered a comparison uninformative. Because our conclusions rest on within-condition paired differences, a level shift does not by itself invalidate them, but the ceiling case shows that evaluator choice can silently destroy a study's resolving power, and the reversal case shows that it can invert one.

2.4 Organizational Memory and Coordination in MASs

Section 1 framed the replay mechanism as organizational memory and derived two predictions from that framing. Here we locate the paper within the multi-agent systems literature those predictions come from, and state what we take to be its contribution to it. That literature has treated coordination among autonomous agents as its central subject from the outset [21], and the mechanism we study is one instance of it.

Organizational and transactive memory. Walsh and Ungson characterize organizational memory as a set of retention facilities and argue that its value depends on how retrieval is *structured* rather than on how much is retained [8]. Wegner's transactive memory adds the directory that records *who knows what* [9], and Ren and Argote's integrative review identifies directory specialization and credibility as the dimensions along which such systems vary [10]. Our findings speak to the specialization dimension, though not in the way we first expected. What we measured is that the directory's *resolution*, the number of distinct key values it distinguishes, bounds how much of the stored experience can reach a query at all, and that adding entries behind an unchanged key does not raise that bound (Section 5.3). The limit is mechanical rather than a failure of any agent's judgment. It is a property of the directory's resolution, not of the agents reading it.

Multi-agent systems research supplies the reason to expect such structural effects in the first place. Horling and Lesser survey the organizational paradigms deployed in agent systems and record that the organizational design chosen has a significant, quantitative effect on performance characteristics [22]. Zoethout et al. show in simulation that the variety of tasks a group confronts determines whether it settles into specialization or generalization, and that the two arrangements trade off differently as that variety changes [23]. Their manipulation is task variety; ours is the granularity of the retrieval key, which fixes how much variety the memory can represent at all before any agent reads it. Our study is therefore a case in which the structural effect that this literature leads one to expect could not be recovered. We enlarged the organization's retention capacity, and output quality did not reproducibly move, because the directory indexing it was unchanged.

Coordination and information flow. Malone and Crowston define coordination as the management of dependencies between activities and locate coordination failure in the dependency structure itself [11]. Our three failure modes are all instances of it. A feedback channel's dependency on a specific artifact is not encoded in its routing key; a filter's implemented predicate diverges from its documented intent; and a retrieval policy is blind to the very dependency it exists to manage. None is an error of agent reasoning, which is why none is visible in an execution trace.

Work on coordination middleware takes the opposite route to the same problem, making the coordination explicit so that it can be checked. Cranefield defines group goals and plans with formal semantics, models the coordination they require as a shared state machine, and verifies the intended behavior by model checking [24]. The failures we report are precisely the case that route does not reach. Each of F1–F3 is a dependency that holds in fact but is represented nowhere. It appears in no routing key, no filter predicate, and no retrieval criterion. There is therefore no specification against which to check it, and no error raised when it is violated. Our failures are what remains after the coordination has been left implicit in the glue between components, which is the common case in the role-decomposed frameworks discussed above.

Failure taxonomies for multi-agent LLM systems. MAST organizes fourteen failure modes observed across a large corpus of agent traces into three categories, namely specification issues, inter-agent misalignment, and task verification [16]. Our findings are complementary in both kind and method. In kind, they fall at the *seam between components* rather than inside any agent. Each component behaves as specified, and the loop fails because their semantics are not aligned. In method, they are recovered by reading the components rather than by mining traces. The distinction matters because a trace-derived taxonomy systematically under-represents silent failures. Here the system runs to completion, emits plausible reports, and raises no error, so there is no trace signature for a trace-based pipeline to find. We read this as an argument for combining trace mining with interface-level auditing rather than as a criticism of either.

Human-agent interaction. The rating channel makes the human a participant in the loop, so its design is a human-agent interaction problem. Two of our findings bear on it. The interface must carry enough structure for feedback to be routed to the artifact it concerns. Otherwise the channel is decorative, which is exactly failure mode F1, and a user who supplies ratings has no way to know they are inert. Expectation-setting also matters. Because replay did not reliably improve output quality in our experiments, an interface that presents replay as an improvement mechanism would mislead the person using it.

The evaluation side of human–AI ensembles has been treated directly. Dell'Anna et al. derive a quality model for hybrid intelligence teams from a study with fifty researchers and validate its attributes against forty-eight usage scenarios, locating quality in properties of the ensemble rather than of any single output [25]. Our evaluator results are a caution for measurement in that setting. Scores from the instrument on which such assessments often rest compressed onto a narrow band of the scale, and two judges that agreed moderately at the dimension level reversed their ordering of the same reports once the dimensions were averaged (Section 5.5). A quality model of the kind just described is only as informative as the resolving power of the instrument used to apply it.

Taken together, this positions the paper's multi-agent systems contribution narrowly but, we hope, usefully. It is not a new coordination mechanism, but evidence about when an existing and widely deployed one, shared experience memory, transmits information between agents, and about failure modes that can make such memory appear to work when it does not.

2.5 Replication and Reliability in Empirical Machine Learning

Replication has long been a concern in the behavioral sciences, where large-scale efforts have found that a substantial fraction of published effects fail to reproduce or shrink considerably on replication [26]. Comparable scrutiny of empirical machine learning and NLP is more recent. Our study is in this tradition, applied to a specific and currently popular mechanism. We are explicit that we report a null result under replication rather than a new method.

3. System and Replay Mechanism

3.1 Pipeline

The system is a serial pipeline of five agents.

Perception agent. Fuses two multimodal emotion signals into a valence/arousal estimate with a confidence value. The signals are facial expression from video and head micro-vibration captured by a contactless video-based estimator.

Analysis agent. Aggregates a seven-day emotion series into twelve derived indicators (emotional stability, fluctuation entropy, negative-emotion accumulation, diurnal trend, arousal abnormality, recovery speed, stress accumulation, positive-emotion ratio, abrupt-change count, and an overall score, among others).

Report agent. Generates a structured daily report in Chinese with a fixed section structure comprising emotional overview, indicator table, key findings, risk analysis, next-day prediction, and suggestions.

Alert agent. Maps the derived risk tier to one of three notification levels with escalating channels.

Orchestrator. Sequences the agents and streams the collaboration trace to a human-facing panel via server-sent events.

Each agent has an independent system prompt and tool set. Prompts are written in Chinese to match the target output language. Reports are the artifact of interest in this study; the perception, analysis, and alert components are held fixed and are not evaluated here. Fig. 1 shows the arrangement.

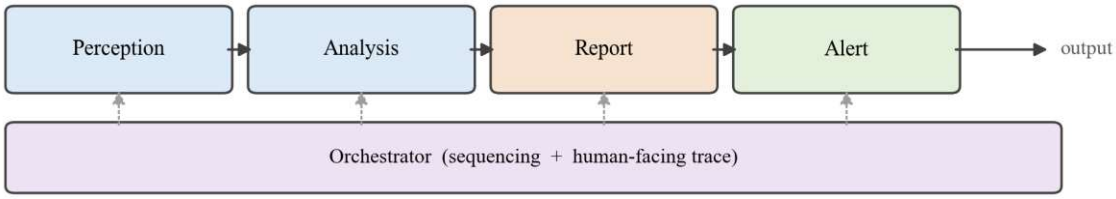


Fig. 1 The five-agent pipeline. Four execution agents run in series under a central orchestrator, which sequences them and streams the collaboration trace to a human-facing panel. Only the report-generation component is evaluated here; the other three are held fixed. The experience-replay loop that closes through the store is shown separately in Fig. 2

3.2 The Replay Mechanism as Intended

After each generation the report text is appended to an experience store, a JSON-Lines file. A separate feedback endpoint accepts a 1–10 rating. Before a new generation, entries are retrieved and injected into the system prompt as "historical successful cases". The intended behavior is experience replay with positive-example prompting, in which highly rated past outputs guide future generations toward higher quality. Fig. 2 shows the loop as intended, together with the three points at which the implementation departs from it.

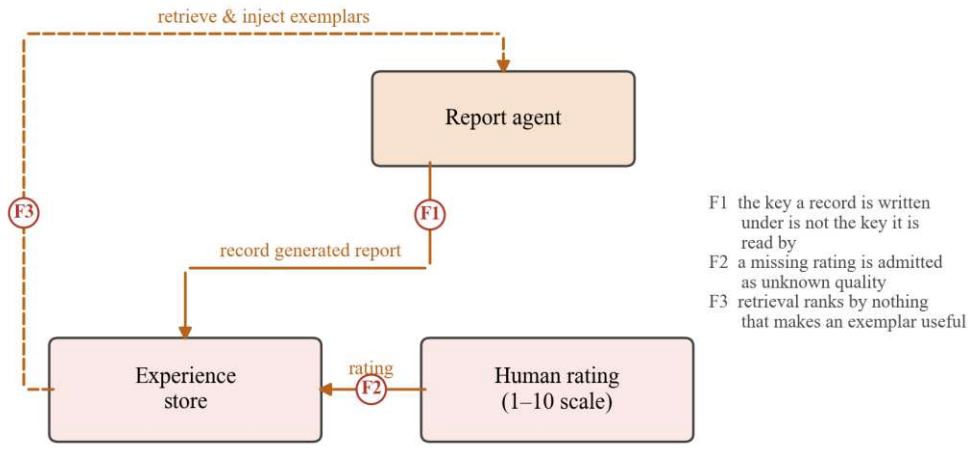


Fig. 2 The experience-replay loop. A generated report is written to the store along the solid path, a human rating is attached to it, and highly rated entries are retrieved and injected back into the report agent's prompt along the dashed path. The three coordination failure modes of Section 3.3 each sit at a seam in this loop, marked F1 to F3 and described in the legend

3.3 Three Coordination Failure Modes

Reading the implementation, we found that the loop does not close in three distinct ways. We describe each as a coordination failure rather than as a coding error, and we make the case for that description after the three.

F1. Feedback routed away from its artifact. Generation-time records were written without a rating, while user feedback was stored under a separate category. The retrieval step filters by category, so no rated entry was ever retrievable for the artifact category it concerned. The rating signal never reached the generation path.

F2. Filter semantics diverging from stated intent. The retrieval predicate discarded entries whose rating fell below a threshold but retained entries whose rating was absent, treating unknown quality as acceptable quality. The documented "high-score-first" policy was therefore not the implemented one. Arbitrary unrated generations were eligible for injection.

F3. Retrieval blind to the dependency it manages. Even among rated entries, retrieval sampled uniformly at random. Injected exemplars were therefore unrelated to the case at hand. In a tone-sensitive task, a report written for a stable student could be injected into the context for a student exhibiting distress.

Table 1 states the three modes in a form that later sections refer back to, with the mismatched pair each one consists of.

Table 1. The three coordination failure modes, each stated as an unaligned pair.

	Component	What it assumes	What the other component does
F1	Feedback routing	Records are written under the key they will be read by	The write key and the read key are chosen independently, so no rated entry is retrievable for the category it concerns
F2	Quality filter	A missing rating means low quality	A missing rating is treated as unknown quality and admitted

	Component	What it assumes	What the other component does
F3	Retrieval policy	The retrieved exemplar is relevant to the case at hand	Retrieval samples uniformly, blind to the attribute that makes an exemplar useful

Why these are structural rather than incidental. Each failure lies at the seam between components, not inside one. The store, the filter, and the retrieval policy each behave as specified; what remains unspecified is the *agreement among them* about what a record means. F1 arises because the key used to write a record and the key used to read it are chosen by different components. F2 arises because a predicate's treatment of missing values is a semantic decision that no interface declares. F3 arises because the attribute that makes an exemplar useful is known to the mechanism's designer but is not required by its signature.

A system that composes a feedback loop from three independently implemented pieces inherits all three hazards unless the pieces are made to agree explicitly. This is a claim about composition rather than about our code, and it is why we expect these failure modes to recur elsewhere. None of them requires a mistake a careful programmer would notice, because none of them is visible within any single component.

Coordination theory gives this a name. Malone and Crowston define coordination as the management of dependencies between activities, and locate coordination failure in the dependency structure rather than in the competence of the parties being coordinated [11]. Read that way, each of our three failures is a dependency that exists in fact and is represented nowhere. F1 is the dependency between the key used to write a record and the key used to read it, which no interface declares. F2 is the dependency between a predicate's treatment of missing values and the semantics its documentation asserts. F3 is the dependency between the retrieval criterion and the task attribute that makes an exemplar worth retrieving. In each case the dependency is real, in that violating it changes what the mechanism does, but it appears in no signature, so no component has violated a specification. That is what makes these coordination failures rather than coding errors, and it is why they are silent. There is no contract to breach.

The same separation appears in the mainstream frameworks. We checked the two frameworks we cite as exemplars of this architecture, reading their memory modules rather than their documentation.

MetaGPT's Memory keeps a storage list alongside a separate index keyed by message.cause_by. add() appends to storage and files the message under that key [5]. Retrieval is exposed through recency (get), role (get_by_role), substring match (get_by_content, try_remember), and action (get_by_action). Nothing in the interface requires the key used to retrieve to be the key used to store; a mismatch yields an empty result rather than an error. That is failure mode F1 as an architectural possibility, not as a claim about MetaGPT's correctness.

AutoGen's Memory protocol attaches to each MemoryContent an untyped metadata: Dict[str, Any] [6]. What a writer puts there and what a query matches on are decided independently at the two ends, and the type system has nothing to say about their agreement. This is F1 again, in the shape of an intentionally generic interface.

We are not claiming that either framework exhibits these failures; we did not run them. The claim is narrower and, we think, sufficient. The store, the write key, and the retrieval policy are separate components in the systems the field builds on, and the interfaces do not enforce that they agree. A hazard that no interface rules out is one that each implementation must rule out for itself.

3.4 Repair

We repaired all three.

1. Feedback is now written back to the artifact category it concerns, so ratings become retrievable for that category.

2. Unrated entries are treated as unknown quality and excluded; only entries at or above the quality threshold are eligible.

3. Retrieval ranks candidates by a structured match on risk tier and dominant emotion, with tier weighted above emotion. The weighting reflects the domain logic that risk tier determines *whether* to intervene while emotion refines *how*; we validate the qualitative ordering in the ablation reported in Section 5.7.

We deliberately describe this as *structured relevance matching* rather than "semantic retrieval", because no embedding model or vector similarity is involved. The term matters because the label "semantic" would imply a mechanism we did not implement.

All experiments reported below use the repaired loop.

4. Experimental Design

4.1 Inputs

We construct 44 student emotion profiles. The anonymized record store spans seven days and three students; aggregating it per student-day yields 19 profiles, of which 16 fall in the medium-risk band and 3 in the low-risk band, with none in the high-risk band. All 19 enter the pool unchanged. To bring each band to a comparable size we synthesize the remainder, drawing indicator values *uniformly* over the range observed for individual records in the relevant band, with a fixed random seed. The low-risk draw spans 0.70–0.95 against an observed record range of 0.700–0.947, and the high-risk draw spans 0.10–0.40 against an observed record range of 0.178–0.327. The pool is therefore 14 low-risk profiles (3 real, 11 synthetic), 16 medium-risk (all real), and 14 high-risk (all synthetic). The synthetic profiles reproduce the *range* of the recorded data in their band rather than its shape, and the pool is deliberately balanced across bands rather than mirroring the store's concentration in one of them.

That no high-risk student-day exists in the store is a limitation we cannot repair after the fact. High-risk *records* do occur, but the unit of analysis here is the student-day, and no student-day in the available data averages below the high-risk threshold; the aggregation removes the band entirely. All fourteen high-risk profiles are consequently synthetic, so the high-risk comparisons rest on constructed inputs in a way the low- and medium-risk comparisons do not.

We emphasize that profiles are *constructed experimental units*, not a sample of a student population. No population-level claim is made anywhere in this paper. The profiles exist to hold the input constant while the injected context varies.

4.2 Models

Two language models serve as generators, both accessed through OpenAI-compatible endpoints and both used unchanged from their public releases.

M1 is Lingshu-32B, a 32-billion-parameter Chinese medical model served through the MoArk platform. *M2* is DeepSeek's deepseek-chat endpoint, accessed in September 2026.

We flag one caveat about M2 because it bears directly on this paper's subject. deepseek-chat is a *rolling alias*. It names an endpoint rather than a frozen model version, and the backing weights may change without notice. M2 should therefore be read as "the DeepSeek chat endpoint as of the stated access date," and a replication attempted

later may not be running the same generator. We report this rather than gloss over it, because it is an instance of exactly the reporting problem this paper documents. A design parameter that materially affects results is easy not to mention.

Traceability is weaker than a replication study would ideally be, and we state this plainly. The runs reported here were made between 18 August and 12 September 2026. We did not record a per-run timestamp or a model revision identifier, and neither endpoint exposes one. deepseek-chat resolves to a moving alias, and the MoArk platform serves Lingshu-32B without a published revision tag. A study whose object is reproducibility should record both. We report the omission rather than describing the endpoints as pinned, and Section 6.7 sets out what it implies for interpretation.

We make no claim about the relative capability of M1 and M2. We did not run an independent capability benchmark, and the labels serve only to distinguish two generators. Where our results differ between them, we report the difference as an empirical fact about these two endpoints rather than as an effect of model scale.

In the main comparisons the judge is the same model as the generator under evaluation (Section 4.5); the cross-evaluator analysis of Section 5.5 deliberately breaks that correspondence.

4.3 Conditions

Baseline. No injected context; the system prompt alone.

Random replay. Two entries drawn uniformly at random from the eligible experience pool.

Matched replay. The two highest-ranked entries under structured relevance matching.

Other-tier replay. Two entries deliberately drawn from tiers and emotions that match neither the profile's tier nor its dominant emotion, used in one experiment to probe whether any non-matching exemplar behaves like a random one.

4.4 Experience Pool Construction

The pool is seeded by generating reports for the non-test profiles. In the *small-pool* setting each seeding profile is generated once, yielding 15 entries. In the *large-pool* setting each seeding profile is generated three times, yielding 45 entries. All pool entries are rated at or above the quality threshold, so what varies between the two settings is pool *size*, not entry quality. Because the reports in the pool are themselves generated by the system, the pool's content differs across runs even when its size is held fixed; we return to this point in Section 5.4.

4.5 Judge

Reports are scored by an LLM judge at temperature 0 using a fixed rubric, on five dimensions (clinical appropriateness, relevance, empathy, actionability, and completeness), each on a 1–10 integer scale. The dependent variable is the arithmetic mean of the five sub-scores. The rubric instructs the judge to use the full range, to place ordinary reports between 5 and 7, and to reserve scores above 8 for exceptional reports. The five sub-scores show acceptable internal consistency (Cronbach's α=0.81), which establishes score coherence but not construct validity. Full rubric text is given in Appendix.

4.6 Replication Design

Each profile × condition cell is generated three times; the mean of the three generations is used as the cell value. All replay conditions share a single baseline per repetition, generated once and reused, so that condition differences are

not confounded with baseline re-sampling. Generation temperature is 0.7; the judge runs at temperature 0. Injection format, prompt composition, and the ranking function are given in Appendix.

Three repetitions is a floor rather than a target. It is the smallest number at which the within-condition variance becomes estimable at all; without it, a comparison over 29 profiles would still rest on a single draw per profile. It is not a count chosen to detect a particular effect size, and no count of this order could be. Section 5.7 sets out why. The binding constraint is the resolution of the instrument rather than the size of the sample. At 29 profiles the paired design can detect a difference of about 0.11 points on a ten-point scale at 80% power, which sits at the upper end of the effects we observe; resolving the smaller effects in play would require several hundred profiles per condition.

Two features of the design bear on the objection that a failure to replicate reflects insufficient power rather than genuine instability. The effects the single runs reported lie above the detection threshold, so the design is not blind to differences of the size at issue. The smaller of the two was detected in a single run at p=0.041. And what the study characterizes is the spread of the estimate across repetitions, a question about the estimator rather than about any one effect size. Adding profiles would narrow the interval around each run's estimate without reducing that spread, which is why replication rather than sample size is the appropriate response. What the study sets out to characterize is how far the estimate moves across repetitions, and that is what three repetitions, applied to the whole experiment rather than to a single generation, are there to expose.

4.7 Statistical Analysis

All comparisons are paired within profile. We report the mean of the paired differences, the two-tailed Wilcoxon signed-rank p-value, and Cohen's d_z computed on the difference scores. Confidence intervals shown in figures are t-based intervals on the mean difference. The test is two-tailed because the hypothesis under study is non-directional. Whether replay helps or harms is part of what is being characterized, not a claim being advanced. Tests are not corrected for multiplicity, because the purpose of the study is not to establish any single effect but to characterize the instability of the estimates. We report below what happens under a false-discovery-rate correction; the conclusion is unchanged.

5. Results

Table 2 lists all fifteen comparisons and Fig. 3 plots them.

Table 2. All fifteen comparisons. Δ is the mean paired difference in report quality (1–10 scale) relative to the shared no-replay baseline; p is two-tailed Wilcoxon signed-rank; d_z is the paired effect size. M1 is Lingshu-32B and M2 is DeepSeek's deepseek-chat endpoint (Section 4); no capability ordering between them is claimed (n=29 for all rows except the two M2 single runs, where n=15). Rows 13–15 report a second independent run of the same nominal configuration as rows 11–12; the remaining three runs of that configuration appear in Table 4.

Model	Design	Condition	Δ	p	d_z
M1	single run	random	-0.172	0.041	-0.44
M1	single run	matched	-0.007	0.679	-0.02
M1	3 reps, pool 15	random	-0.048	0.442	-0.19
M1	3 reps, pool 15	matched	-0.062	0.130	-0.29
M1	3 reps, pool 45	random	-0.106	0.044	-0.42
M1	3 reps, pool 45	matched	-0.090	0.017	-0.51
M2	single run	random	+0.093	0.239	+0.34
M2	single run	matched	+0.147	0.040	+0.63

Model	Design	Condition	Δ	p	d_z
M2	3 reps, pool 15	random	+0.147	0.0005	+0.74
M2	3 reps, pool 15	matched	+0.046	0.241	+0.24
M2	3 reps, pool 45	random	+0.030	0.273	+0.12
M2	3 reps, pool 45	matched	+0.012	0.747	+0.05
M2	3 reps, pool 45 (run 2)	random	+0.092	0.014	+0.50
M2	3 reps, pool 45 (run 2)	matched	+0.009	0.929	+0.04
M2	3 reps, pool 45 (run 2)	other-tier	+0.032	0.293	+0.15

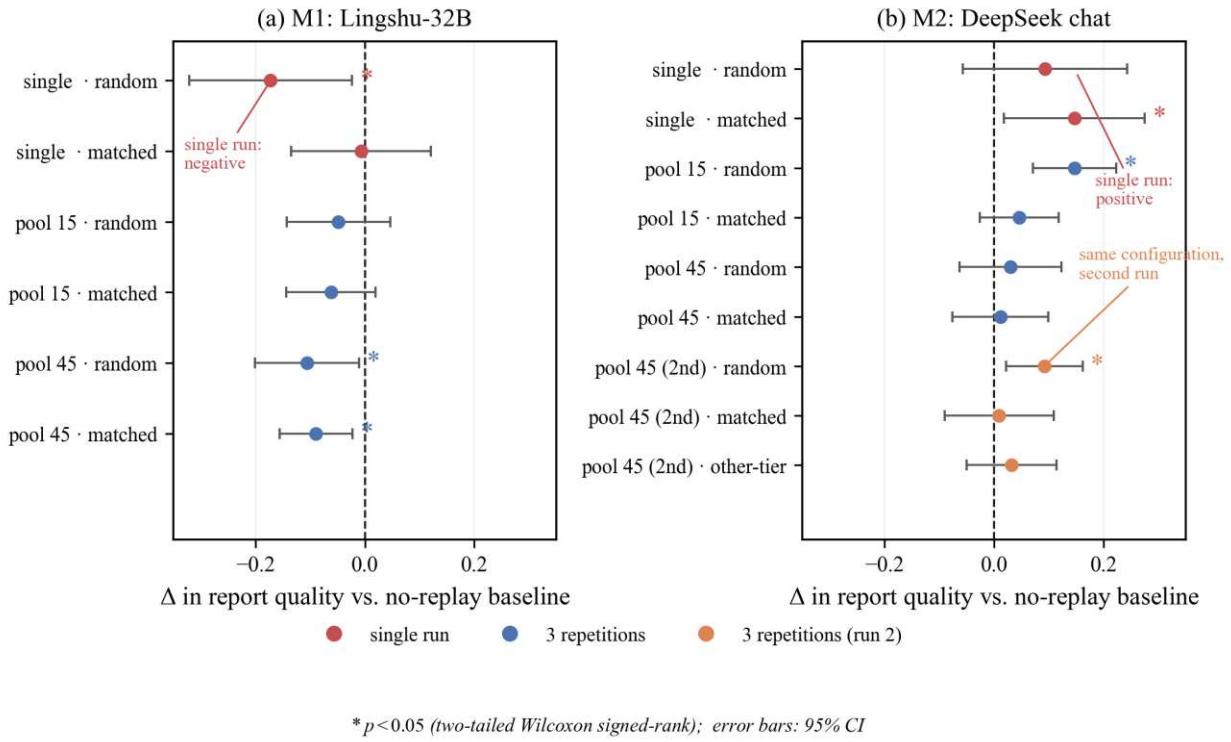


Fig. 3 Effect of experience replay on report quality, for all fifteen comparisons, split by model so that the two panels share a common scale. Points are mean paired differences relative to the shared no-replay baseline; bars are 95% confidence intervals; asterisks mark p<0.05 (two-tailed Wilcoxon signed-rank). Rows marked *single* are single runs, and every other row uses three repetitions per cell. Three comparisons are annotated. The single-run estimates point in opposite directions on the two models; the two pool sizes are indistinguishable once replicated (Mann–Whitney p=0.42 on M1 and p=1.00 on M2); and a second run of an identical configuration (orange) disagrees with the first (blue)

5.1 Single-Run Evaluation Suggests Large Effects in Opposite Directions

On model M1, a single run indicated that random replay substantially *degrades* report quality (Δ=-0.172, p=0.041, d_z=-0.44, a medium negative effect). On model M2, a single run indicated that matched replay significantly *improves* it (Δ=+0.147, p=0.040, d_z=+0.63, a medium-to-large positive effect). The two runs are not matched in sample size. The M1 run covers the full 29 test profiles, while the M2 runs use the 15 seeding profiles alone, so the two estimates differ in precision as well as in sign. We report them as two instances of a single-run estimate rather than as a controlled contrast between the models.

Taken at face value, these two runs alone would support a confident published claim about the effect of experience replay, and the sign of that claim would depend on which of the two models was selected as the experimental subject.

5.2 Replication Removes Both Effects

With three repetitions per cell, neither effect survives. On M1, the random-replay effect shrinks by roughly three quarters and becomes non-significant (Δ=-0.048, p=0.442). On M2, the matched-replay effect that a single run had declared significant becomes non-significant (Δ=+0.046, p=0.241). Neither single-run finding reproduces.

5.3 A Pool-Size Comparison That Does Not Survive Replication

Our first pass at this question compared a pool of 15 entries against one of 45, one run each, and the comparison looked decisive. On M1 the larger pool gave significant harmful effects (Δ=-0.106, p=0.044; matched -0.090, p=0.017) where the smaller pool had given nothing (p=0.442, p=0.130); on M2 the larger pool gave nothing (Δ=+0.030, p=0.273) where the smaller pool had given a large benefit (Δ=+0.147, p=0.0005). Read together these suggested that pool size is a design degree of freedom capable of reversing a conclusion, in a model-specific direction.

That reading does not survive replication. We re-ran each of the four configurations five times independently, holding everything but the pool size fixed, and the two pool sizes are not distinguishable. Table 3 gives all twenty runs.

Table 3. Effect of random replay by experience-pool size, five independent runs per cell. Δ is the mean paired difference in report quality against the shared no-replay baseline.

Model	Pool	Δ across five runs	Mean	p
M1	15	-0.101, -0.064, +0.046, -0.035, -0.127	-0.056	0.42
M1	45	-0.060, +0.002, -0.099, +0.039, -0.014	-0.026	
M2	15	+0.037, +0.042, +0.025, +0.092, +0.083	+0.056	1.00
M2	45	+0.030, +0.092, +0.025, +0.129, +0.028	+0.061	

On M1 the two distributions overlap over 80% of their joint span (Mann–Whitney p=0.42); on M2 they are indistinguishable (p=1.00). The difference between the two means is +0.030 on M1 and +0.005 on M2, while the spread across runs *within* a single pool size is 0.17 and 0.10 respectively. On M1 the direction is moreover the opposite of what the single-run comparison suggested. The smaller pool produced the larger average harmful effect. We therefore withdraw the claim that pool size moved the estimates in a model-specific direction. The two readings in our first pass came from single runs that happened to fall on opposite sides of a noisy distribution. This is the phenomenon the paper documents, occurring a fourth time, here inside our own analysis.

Why the pool size does not matter here. The null result has a mechanical explanation, and it is not the one we first offered. We instrumented retrieval directly, recording for each test profile which store entries the matched-retrieval function returned. Both pools draw their entries from the same fifteen seeding profiles, so both populate the same *seven* of the twenty-four possible (risk tier, emotion) cells, the product of three risk tiers and eight emotions; the pool of 45 simply holds three times as many entries *within* each of those seven cells. Retrieval ranks candidates by a score dominated by the tier match (+2), with the emotion match (+1) and the rating (0.1×) serving as tie-breakers, so the top two entries within a cell are the same however many candidates the cell contains. We observed a mean of 7.0

distinct returned pairs across 29 test profiles at pool size 45 and 7.3 at pool size 15, against a theoretical maximum of 990 for the larger pool, so the two pool sizes are indistinguishable on this measure as well.

The quantity that governs retrieval diversity is therefore the number of distinct *keys* in the store, not the number of entries. Tripling the store multiplied entries per cell without adding a cell, and the set of possible retrieval outcomes was unchanged, which is why the effect estimates were unchanged too. This is a sharper statement than the one we first made, and it points at a different design question. What a retrieval policy can express is bounded by the *resolution* of its key, and enlarging a store whose keys are coarse adds candidates rather than diversity. Studies comparing retrieval policies should report the key, the number of distinct key values, and the number of entries per key value. These are not, to our knowledge, standard statistics, and without them a null result of the kind we report cannot be interpreted, because we would have had no way to tell a policy that does not help from a policy that cannot express a difference.

A bound on retrieval diversity. The observation generalizes beyond this store, and the generalization is what makes the key the right parameter to report. Write the retrieval score as $s(e,q) = f(k(e), k(q)) + g(e)$, where $k(\cdot)$ is the entry's key, $k(q)$ the query's, and g a query-independent term such as the rating. Retrieval returns the m highest-scoring entries. Two consequences follow. First, every entry that retrieval can return lies within the top m of its own key cell under g. Within a cell f is constant, so g fixes the internal order, and an entry outside its cell's top m is outranked by m entries that share its f value and therefore cannot enter the global top m under any query. Second, adding entries to a cell that do not enter its top m under g leaves every retrieval outcome unchanged, because such entries can outrank the cell's existing leaders under no query. The number of distinct outcomes a query set can produce is therefore bounded by the populated cells together with their within-cell orderings, and is invariant to store size wherever the additions fall below that rank. For this mechanism f is the tier and emotion match and g is the rating, which is why tripling the store changed nothing. The 7-of-24 measurement above is an instance of the bound, not a coincidence of this dataset. The bound also says what would change the outcome. A finer key, or a g that depends on the query, would.

5.4 Fixed Configuration, Repeated Runs

To characterize the estimator directly, we ran the same nominal configuration five times independently on *each* model, with an experience pool of 45 entries, three repetitions per cell, and the random replay condition. Across the ten runs, the only things that differed were the random draws that populated the experience pool and sampled exemplars within it. Table 4 reports all ten; Fig. 4 plots the M2 series.

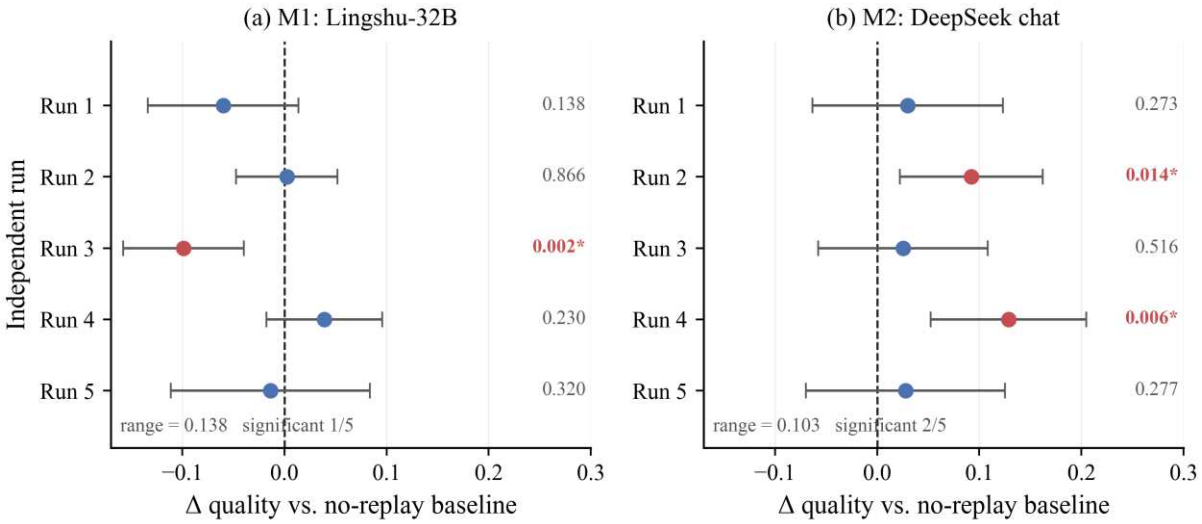


Fig. 4 Ten independent runs of one identical configuration (experience pool of 45 entries, three repetitions per cell, random replay), five per model. Points are mean paired differences against the shared no-replay baseline; bars are 95% confidence intervals; red points mark runs reaching $p<0.05$, with the p-value printed at the right of each row. On M2 the estimates span 0.103 and every run is positive, so the sign is stable and only the magnitude moves. On M1 they span 0.138 and the sign reverses across runs

Table 4. All ten independent runs of the identical configuration (experience pool of 45 entries, three repetitions per cell, random replay), five per model. Δ is the mean paired difference in report quality relative to the shared no-replay baseline; p is two-tailed Wilcoxon signed-rank. Run 2 of M2 is also reported in rows 13–15 of Table 2.

	M1 (Lingshu-32B)		M2 (DeepSeek chat)	
Run	**Δ**	**p**	**Δ**	**p**
1	-0.060	0.138	+0.030	0.273
2	+0.002	0.866	+0.092	0.014
3	-0.099	0.002	+0.025	0.516
4	+0.039	0.230	+0.129	0.006
5	-0.014	0.320	+0.028	0.277

Range across runs, M1 0.138, M2 0.103. Bold p denotes significance at 0.05.

The result is the central observation of the paper, and running both models makes it harder to dismiss as an artifact of one system. On M2 the estimated effect spans a fivefold range, Δ=+0.025 to +0.129, and the significance verdict flips. Two runs report a significant benefit (p=0.014, p=0.006) while three report none (p=0.273, p=0.516, p=0.277). The sign, however, is stable. Every M2 run is positive, so what varies is the magnitude and with it the verdict.

On M1 the instability is more severe. The estimates run from Δ=-0.099 to +0.039, and the *sign itself* reverses across runs (-, +, -, +, -), with the most extreme run reaching significance. A reader of any single M1 run would not merely misjudge how large the effect is; they would misjudge which way it points.

This is the central observation of the paper. It is not that the effect is small and therefore merely hard to detect. It is that the standard error of the estimator is of the same order as the estimate itself, so that the outcome of a single

comparison is close to arbitrary. A "significant" result under such conditions carries almost no information about the underlying parameter, and a non-significant result does not license the conclusion that the effect is absent. The practical consequence is that a literature reporting single-run comparisons of this mechanism would contain a mixture of confident positive findings and confident null findings, all generated by the same underlying process.

Quantifying the instability. A meta-analytic reading makes the source of the instability explicit, and shows that the two models are unstable in different ways. On M2, treating the five runs as independent estimates of a common effect, the pooled estimate is Δ=+0.069; Cochran's Q=5.70 on four degrees of freedom does not reject homogeneity (p=0.22), and the share of variance attributable to between-run differences is I^2=29.8% with τ=0.026. The M2 runs are thus not detectably estimating *different* effects. They are estimating the same effect with a standard error comparable to the effect itself. The ratio of the estimate to its standard error ranges from 0.6 to 3.5 across the five runs, straddling the value required for significance at this sample size. That is why the verdict flips while the sign does not.

On M1 the picture is different. The same analysis gives Q=14.1 on four degrees of freedom, which *does* reject homogeneity (p=0.007), with I^2=71.6% and τ=0.049. Here the runs are detectably estimating different effects, not merely the same effect imprecisely, which is what one would expect if the content of the experience pool, re-drawn each run, is itself part of what determines the outcome. The two models thus illustrate two routes to the same practical failure. One is an estimate too noisy to resolve; the other is an estimate whose value genuinely depends on an unreported draw.

The contrast is worth naming, because the two routes have different origins. M2's instability is a property of the *estimator*. The runs estimate a common effect and simply cannot resolve it at this sample size. M1's is a property of the *coordination mechanism*. What the mechanism returns depends on which entries the store happens to hold, so re-drawing the store redraws the effect. Both end in the same place, since a single run settles nothing, but only the second is about the system rather than about the measurement. A field that reports one run per configuration is exposed to both, and cannot tell which it is looking at.

Whether the second reading is the right one for M1 is testable in principle. It predicts that runs sharing a pool should agree more closely than runs differing in pool. We did not design the study to test that, and we flag it as an open question rather than a finding.

The implication for both models is the same. A single run's estimate is too noisy, or too pool-dependent, to settle the question, and reporting any one of these ten runs would have produced a confident conclusion from what is, on this analysis, a single draw from a distribution whose mean is not distinguishable from zero.

Multiplicity. The comparisons above are reported without correction, on the grounds that the study characterizes an instability rather than establishes an effect. That reasoning is not a substitute for checking whether correction would change the picture, and it does not. Applying Benjamini–Hochberg control at α=0.05 across all twenty-five comparisons in Table 2 and Table 4 leaves three standing. These are the pool-15 comparison on M2 (p=0.0005) and two of the ten identical-configuration runs, +0.129 on M2 (p=0.006) and -0.099 on M1 (p=0.002). Every one of these is an instance of the same manipulation, random replay against a shared baseline, and they disagree in sign across the two models. Correction therefore does not rescue a single claim from this study; it selects, as the most robust findings in the set, evidence pointing in opposite directions. That is the outcome this paper predicts, and it is why the argument is framed as a property of the estimator rather than as a positive or negative result about replay.

5.5 Evaluator Analysis

We additionally scored a fixed set of twelve reports with two different judges, the system's own judge and an external one drawn from a different model family.

Level shift. The judgments differ systematically in level. The self-judge placed its own outputs in the range 8.0–8.2 while the external judge placed the same reports in 6.0–7.2, a shift of 1.61 points on a ten-point scale. Nothing about the reports changed between the two passes; only the evaluator did.

Rank agreement. The two judges agree only moderately at the dimension level ($r=0.55$, $n=60$ paired observations, $p<10^{-5}$). At the report level, where the five dimensions are averaged and their non-independence is removed, agreement does not merely weaken but *reverses*. The correlation there is $r=-0.70$ ($n=12$, $p=0.012$). To a substantial degree, the judge that scored a report high was the judge that scored it low. We report the second figure with the caveat that twelve reports is a small sample, but the sign is not marginal, and the direction of the discrepancy is the one that matters. An evaluation pipeline that pooled the two judges would be averaging quantities that order the same objects in opposite directions. That the two correlations differ so widely is also a caution about how agreement is reported. Pooling non-independent dimensions inflates apparent concordance, and a pooled figure alone would have concealed the reversal.

Ceiling effect. A third judge, drawn from a still different model family, assigned nearly identical scores (8.2–8.6) to both conditions across fifteen items, returning the same score for the two conditions on twelve and eleven of them respectively. It was therefore unable to discriminate between conditions at all. Since our conclusions rest on within-condition paired differences rather than absolute scores, the level shift between the first two judges does not by itself invalidate them; but the ceiling case shows that an evaluator can silently render a comparison uninformative even when the rubric, prompt, and model are held fixed.

Expert human raters. The same twelve reports were also scored by five expert raters from a university psychology department, blind in the sense that matters here. The reports reached them identified only by number, with no indication of condition, and they were told the study concerned the quality of automatically generated reports rather than the effect of a retrieval mechanism. Each rated all twelve on eight dimensions, of which the first five correspond to the five the judge scores. Three features of the resulting data bear on this paper.

The level disagreement among experts is *larger* than between the two LLM judges. The five raters' mean scores span 2.60 points (9.22 to 6.62), against 1.61 between the judges. Absolute agreement is close to zero ($ICC(2,1)=0.07$) and rank agreement is weak ($ICC(3,1)=0.33$; mean pairwise $r=0.35$, Spearman $\rho=0.31$, $n=12$ reports). The five do not behave as one panel but as two. Within each cluster the raters agree closely ($r=0.96$ and 0.98), while across clusters they are essentially uncorrelated ($r=-0.08$, 0.01, 0.09). Removing any single rater leaves $ICC(2,1)$ between 0.05 and 0.10, so this is not the work of one outlier.

The experts and the system's judge do not agree either. Across the twelve reports the correlation is $r=0.18$ ($p=0.58$). The experts' report-level scores span 7.20–8.68 (SD=0.40) where the judge's span 8.00–8.20 (SD=0.09). We note that this is not the reversal seen between the two LLM judges. No pair of experts orders the reports oppositely in the way that pair did. What the experts show is not disagreement of that kind but the absence of agreement of any kind.

Most consequentially, the experts found defects the judge did not. Three of the twelve reports carry a concrete fault. One has an indicator table missing the risk-level row that the surrounding prose then cites. One reports 52.3% positive against 47.7% negative emotion while assigning the highest risk tier. One contains an English word left inside a Chinese sentence. Two raters independently identified the first two in written comments, and two identified the third. The judge scored those three reports 8.00, 8.20 and 8.20, indistinguishable from the rest of the set. An instrument that returns the same value for a report missing its risk tier and for one that is not cannot be measuring the property its rubric names, and averaging over runs would not recover the difference.

5.6 Dimension-Level Breakdown

For the M1 single-run condition, the one that appeared to show degradation, we examined which dimensions carried the effect. The largest contribution came from actionability (Δ=-0.276), followed by clinical appropriateness, relevance, and completeness (Δ=-0.172 each), with empathy least affected (Δ=-0.069). This pattern is consistent with injection interfering most with the concrete, actionable content of a report. We report it as a description of one run, not as an established effect, for the reasons given above.

5.7 Convergence and Ablation

Two additional experiments probe the mechanism under more forgiving designs. Both are exploratory relative to the main study; neither yields evidence of a parameter effect, and the second is instructive about the evaluation instrument as much as about the mechanism.

Multi-round replay. We simulated five rounds of the full loop (generate, score, record, retrieve, regenerate), with the experience pool growing from 8 to 40 entries. In the first run, mean quality across rounds was 8.200 → 7.850 → 8.050 → 8.025 → 8.175, from which we read the absence of a monotone trend. Repeating the whole procedure five times supports that reading without removing the ambiguity. Four of the five runs show no monotone trend, while one declines strictly (8.250 → 8.150 → 8.025 → 7.950 → 7.875, a drop of 0.375). Pooled across runs the round means are flat (8.165, 8.080, 8.115, 8.075, 8.110), and the spread within a single run (0.13 to 0.38) is larger than any change across rounds in the pooled series. We therefore state the finding as a tendency rather than a result, which is what our first run licensed and what the replications support. Table 8 gives the round means for all five runs.

Ablation of retrieval parameters. We varied three parameters independently, holding the others at their defaults, over twelve profiles per setting (Table 7, Appendix). No parameter produced a reliable effect. Top-k gave means of 8.10 (k=1), 8.12 (k=2), and 8.07 (k=3), a spread of 0.05 on a ten-point scale with no monotone ordering (one-way ANOVA F=0.056, p=0.95). Rating threshold and injection position likewise produced differences of at most 0.10 and no ordering.

We report this null result against an earlier finding of our own. A first run of the same ablation, with five profiles per setting, had shown a monotone trend favoring smaller k (8.20, 8.12, 7.80; F=2.02, p=0.18), and we had been prepared to interpret it mechanistically, as each additional exemplar competing with the system prompt for the generator's attention. Tripling the sample dissolved the trend. We report the reversal rather than the tidier first result because it is a third instance of the phenomenon this paper documents, occurring inside our own analysis. A small-n comparison produced an ordered pattern that invited a mechanism story, and the pattern did not survive more data. We note also what the ablation illustrates about the instrument. Of the 96 scores it produced, 73 were exactly 8.2 and the full range was 7.2 to 8.6. A dependent variable with eight-tenths of its mass on a single value cannot resolve differences of the size these parameters might plausibly produce. The appropriate conclusion is not that the parameters are inert, but that this instrument cannot tell us whether they matter.

The ablation is not merely underpowered. That objection is testable, and it does not hold. Profiles are paired across settings, so the relevant test is paired. We report the detectable effect under two choices of standard deviation, so that the conclusion does not rest on which is preferred. Within settings, the pooled standard deviation of judge scores is 0.307; between settings, the standard deviation of the paired differences is 0.491. At twelve profiles per setting and 80% power (two-tailed α=0.05), the smallest difference detectable is 0.273 points on the first reading and 0.437 on the second. The largest difference the ablation actually produced was 0.05. Detecting an effect of 0.10, the middle of the range our main experiments produce, would require about 77 profiles per setting on the more favorable reading and 192 on the less favorable one; an effect of 0.03 would require roughly 827 and 2108 respectively. The ceiling is set by the instrument rather than by the sample. Among the scores it emits, 76% take the same value, and its entire range spans 1.4 points. A comparison run on this instrument, at any size a study of this kind would contemplate, returns a null, and that null carries no information. We therefore report it as a limit of the measuring device, not as evidence about the retrieval parameters.

The same limit applies to the main experiments, which is the point of the paper. There the single-generation scores take a modal value in 77% of cases on M1 and 45% pooled across both models, with standard deviations of 0.264 and 0.390 respectively; the effects we are trying to measure are of the same order as the instrument's resolution. That is why replication rather than larger samples is the appropriate response. At 29 profiles the paired design can detect a difference of about 0.11 points, at the upper end of the effects in play, so adding profiles narrows the interval around an estimate whose scale the instrument can barely resolve, whereas repeating the whole experiment is what reveals that the estimate itself moves.

5.8 Rule-Based Safety Checks

Because mean judge scores can conceal rare but severe errors, we added deterministic text checks over report content. These ask whether the stated risk tier contradicts the input tier, whether the report makes diagnostic assertions (for example, asserting a diagnosis rather than describing a tendency), and whether low-risk profiles receive alarmist language. Across the evaluated reports these checks found no violations.

We report this as a null result and explicitly decline the stronger reading. The checks operationalize only three coarse properties, with rules validated by inspection rather than by expert annotation; they cannot support a claim that the reports are safe, only that these three particular failure signatures were absent.

5.9 Cross-Task Validation on an Objective Task

Every result so far comes from a task whose dependent variable is a compressed subjective score. To ask whether the instability is a property of that instrument or a more general phenomenon, we constructed a second task with a deterministic answer key and ran the same design on it.

Task and instrument. The system is shown ten students, each described by six emotion indicators, and must apply a stated rule to identify the four who most need immediate intervention. The rule combines a three-level risk classification with a lexicographic ordering on two further indicators. The dependent variable is the overlap between the selected and correct sets, taking values in 0, 0.25, 0.5, 0.75, 1, scored by exact match with no judge involved. Thirty fixed groups of ten students serve as items, and conditions are paired within group. Replay takes the same form as before, with past outputs retrieved and injected as worked examples, where a past output is now a group together with its correct selection. Two properties make this a fair test. The dependent variable is graded over five levels rather than compressed onto one, and its scoring is deterministic, so no evaluator noise enters. Pilot measurement confirmed a usable range (mean overlap 0.79 on M1 and 0.91 on M2, against a chance level of 0.45).

A confound and its control. The injected worked examples are longer than those of the report task, ten lines against roughly three, so a harmful effect could reflect context length rather than replay content. We therefore added a fourth condition, *filler*, which injects two pseudo-examples of identical format and length (1770 versus 1766 characters) whose indicator values and answer are randomly generated. Filler isolates length and format from informational content.

Table 5. Cross-task validation on an objective task (select 4 of 10 students by a stated rule). Δ is the mean paired difference in overlap accuracy against the no-injection baseline over five independent runs; p is two-tailed Wilcoxon signed-rank. *Filler* injects length-matched but uninformative pseudo-examples.

Model	Condition	Δ range	Δ/SE range	Significant
M1	random	-0.031 to +0.036	-1.20 to +1.19	0/5
M1	matched	-0.019 to +0.042	-0.71 to +1.40	0/5
M1	filler	-0.031 to +0.014	-1.06 to +0.71	0/5

Model	Condition	Δ range	Δ/SE range	Significant
M2	random	-0.047 to -0.025	-2.48 to -1.33	1/5
M2	matched	-0.036 to -0.014	-1.58 to -0.76	0/5
M2	filler	-0.022 to +0.006	-1.22 to +0.42	0/5

Report task, M2, for comparison. Δ range 0.103, 2/5 significant.

Table 5 reports five independent runs per model and condition. Three findings follow.

The confound is largely ruled out. On M2, filler shifted accuracy by -0.011 while random replay shifted it by -0.039. Length and format therefore account for roughly a third of the effect, and the remainder is attributable to replay's informational content. The harmful effect is not an artifact of a longer prompt.

The magnitude of the estimate is far more stable. The range of Δ across five runs is 0.022 on M2, against 0.103 in the report task, a factor of 4.7. In this narrow sense the objective instrument behaves as one would hope.

But the verdict still flips. One of the five M2 runs reaches significance and four do not, with the test statistic ranging from -2.48 to -1.33, straddling the critical value of 2.05. On M1 no run reaches significance, but the sign of the estimate reverses (-0.031 to +0.036). A reader of any single one of these ten runs would draw a confident conclusion, and the conclusions would not agree.

The explanation is visible in the same numbers, and we regard it as the substantive result of this section. Raising measurement precision compressed the signal and the noise *together*. The absolute variability of the estimate fell by a factor of 4.7, but so did its standard error, leaving the ratio that determines the verdict, and therefore the verdict itself, about as unstable as before.

The two observations are therefore not in tension, and it is worth stating the relation plainly, because the distinction is easy to miss. *Measurement precision determines the absolute variability of an estimate; it does not determine whether the estimate resolves the effect.* What governs the second is the ratio of the effect to the noise of a single draw, and improving the instrument moves both terms together. A study can therefore report a visibly tighter estimate and still be unable to say which way the effect points. Instability is not a property of subjective evaluation as such. What moves the ratio, and with it the verdict, is the size of the effect or the number of items, not the fineness of the scale.

We note two limits of this experiment. The task is one of rule application, not of planning, tool use, or task allocation, so it tests the measurement properties of an objective instrument rather than the generality of our findings across task families. And the injection format differs from the report task's, so the comparison across tasks is one of instruments rather than of identical manipulations. That second limit is what motivates the task below.

5.10 Trajectory Task: A Large Effect That Is Stable

The task above answers a single question and scores the answer. The boundary this paper identifies is a ratio rather than a task type, which makes the complementary test one in which the effect is large. If the account is right, the instability should not appear there.

Task. A student is described by two indicators, a stress index and a negative-emotion ratio, and begins in the highest risk tier. Over five days the system selects one intervention per day from a menu of five, each usable at most once. Transitions are deterministic, and an action whose precondition is unmet, or one already used, wastes the day. One action reduces stress far more than any other but is legal only while the student remains in the highest tier, and two

others each reduce stress enough to leave that tier in one step. Order therefore decides the outcome. A general stress reducer used first makes the strong action permanently unavailable and puts the goal out of reach. Scoring is a deterministic function of the final state and involves no evaluator, so the task has an exact answer key and contributes no measurement noise of its own. We generated thirty starting states and verified that every one is solvable in the correct order and unsolvable in the wrong one.

Result. Replay helps a great deal here, and the verdict does not move. Averaged over five independent runs, the matched condition improves the score by +0.057 on M1 and +0.173 on M2 against the shared no-replay baseline, and every one of the five runs reaches $p<0.05$ on both models. The spread of the estimate across runs is 0.035 on M1 and 0.006 on M2, and the sign is positive in all ten runs. To put the contrast plainly, the ratio of the estimate to its standard error, the quantity that Section 5.4 identifies as governing whether a single run settles anything, ranges from 2.7 to 5.2 on M1 and from 67 to 70 on M2, against 0.6 to 3.5 in the report task. A large ratio means the estimator resolves the effect, and the verdict is then stable across runs. The length-matched filler condition sits at the baseline on both models (0.698 against 0.704 on M2), so the gain comes from the content of what is injected rather than from a longer prompt.

What this adds, and what it does not. Together with Section 5.9 the two objective tasks bracket the phenomenon. A single-selection task with a small effect reproduced the instability, and a trajectory task with a large effect did not. That is the boundary condition the paper states, tested from the side that predicts stability, and it locates the risk in the ratio of effect to single-draw noise rather than in the subjectivity of the measurement or in the shape of the task. Two features of this task limit how far the result travels, and we state them rather than leave them to be inferred. The injected material is a past intervention plan for a similar student, and such a plan transfers almost directly, which makes this a more forgiving analog of the mechanism than the report task is; retrieval also cannot fail to return something useful, because every stored plan solves the same trap. And our starting states vary over a narrow band, so the contrast between relevance-matched and random retrieval, which separates the two in the report task, is uninformative here by construction. We therefore read the result as evidence about when the instability occurs, not as evidence that trajectory tasks are generally immune to it.

6. Discussion

6.1 Why Single-Run Estimates Mislead Here

The reports produced by this pipeline are stylistically homogeneous and receive compressed judge scores clustered near the top of the scale. The compression is severe. Across the 96 scored reports of the ablation experiment (Section 5.7), 73 received exactly the same mean score, 8.2, and the entire range spanned 7.2 to 8.6 on a ten-point scale. In this regime the residual variance across generations is comparable to the effect of a context manipulation. A single draw can therefore land far from the mean in either direction, and with a sample of 29 profiles it can reach conventional significance.

Both of our single-run results were of this kind. One was a spurious large negative effect on M1, the other a spurious large positive effect on M2, each individually significant and neither reproducible. The mechanism is not specific to mental-health reporting. It applies whenever the dependent variable is a compressed subjective score and the manipulation is a modest change to context.

6.2 Implications for Agent Memory Research

Our results do not show that experience replay cannot help. They show that, for this task and these models, a commonly used evaluation procedure cannot distinguish its effect from zero, and that the same procedure readily reports a confident and reproducible-looking finding in either direction.

Relation to transactive-memory theory. The mechanism we identify, namely that a directory informs a collective only to the extent that it can distinguish the queries put to it, is the transactive-memory account of directory resolution [9, 10], carried into a setting where the directory itself is inspectable. The theory was developed for human teams, where the directory is a cognitive construct inferred from behavior. In an LLM multi-agent system the directory is a data structure, and its resolution can be measured rather than inferred. We can count the distinct key values it holds and record what each query returns (Section 5.3). That makes the mechanism testable in a way the human-team literature cannot manage, and it locates the constraint in a different place than we first assumed, in the *resolution of the key* rather than in the volume of the store. What the theory calls directory specialization is, in a deployed agent memory, a property of the indexing scheme; and indexing schemes are chosen, usually without discussion, by whoever implements retrieval. Our data also suggest that the resolution threshold, where it exists, will depend on the generator, since a more capable model extracts more from a repeated exemplar; but we did not design a study to establish that, and we flag it as a prediction rather than a finding.

This matters because the number of memory mechanisms being proposed and evaluated is large, and because this mechanism is widely deployed. Some reported benefits may be of the kind documented here. We do not claim that they are, only that a single-run study cannot rule it out, and that the field would benefit from a norm of replication before claims of benefit. Nor do we claim that single-run evaluation is the norm; our claim is about what that design can support wherever it is used, which is the question a reader of any one such study needs answered.

A related implication concerns the reporting of design parameters. Store size is rarely reported in detail, and when we tried to establish its effect we could not. Five runs at each of two sizes were indistinguishable on either model. But the reason turned out to be more useful than the effect would have been. What bounds retrieval is not the number of entries but the number of distinct key values they are filed under, and that number is a property of the indexing scheme, which no paper in this literature reports at all. The parameter we were chasing was the wrong one. A study should state the retrieval key, how many distinct values it takes in the store, and how many entries share each value; without those three numbers, a null result of the kind we report is uninterpretable, and a positive one is uninterpretable for the same reason.

6.3 Implications for Multi-Agent System Design

The results bear on three decisions a designer of such a system actually faces.

Whether to deploy a shared memory at all. The results do not show that replay cannot help; they show that its effect is not measurable at this scale, so a single evaluation cannot tell a designer which way to go. What changes is the ground on which a deployment decision can rest. Where an effect cannot be resolved, the defensible reasons to adopt such a mechanism are the ones that do not require it to be measured. It makes the system's behavior auditable, it lets a rating reach the artifact it concerns, and it is cheap. A reason of the form "we evaluated it and quality improved" requires a comparison that, in this regime, the field is not currently positioned to run, and Section 6.5 sets out what running one would take.

What it implies for ablation studies. The ablation in Section 5.7 has the common shape of a design claim in this literature. Hold the pipeline fixed, vary one mechanism parameter, and report which setting wins. On an instrument of this resolution that procedure cannot discriminate among settings, and its null result is uninterpretable in both directions. A reader cannot tell whether the parameter does not matter or whether the study could not have detected it mattering. The remedy is not a larger sample but a different instrument, or at least an explicit statement of the difference that would have been detected. A reported "no difference" is informative only alongside that number.

Where the finding generalizes. Key resolution bounds what an attribute-indexed memory can return. The same question can be asked of any coordination mechanism that routes by a key rather than by content. Examples include a message bus dispatching on a coarse topic, a task allocator bucketing by role, and a cache keyed on a hashed

context. In each case the resolution of the routing key bounds how many distinct destinations the mechanism can reach, and enlarging the pool behind it does not raise that bound. We have tested this only for retrieval over an experience store and state it here as a question to ask of other mechanisms rather than as a result about them. The general form of the question is to ask what a coordination mechanism can express before asking whether it helps.

6.4 Repairing a Loop Is Not the Same as Improving It

The three failure modes we report are of a kind that plausibly recurs wherever a feedback loop is composed from independently implemented pieces. A feedback channel may not reach the artifact it concerns; a filter may implement semantics that differ from its stated intent; and a retrieval step may be blind to the property the mechanism is supposed to exploit. Each is silent. Each leaves the system running and producing plausible output. And as Section 3.3 argues, none requires a mistake that would be visible inside any single component, which is what makes them a hazard of composition rather than of this particular codebase.

Repairing them restores the intended semantics of the loop, since feedback now reaches the artifact it concerns, the quality filter now filters, and retrieval now attends to relevance. It does not, in our experiments, produce a measurable gain in output quality.

Designers and reviewers should therefore resist treating "the loop now closes" as evidence that the loop helps. Closure is necessary for the mechanism to do anything; it is not sufficient for the mechanism to do something measurable. The practical form of this warning is procedural. Closure should be verified by exercising the loop end to end and inspecting what actually reaches the prompt, rather than inferred from the presence of the components in an architecture diagram.

6.5 Recommended Evaluation Protocol

From these results we distill seven checks for evaluations of memory mechanisms in agentic systems. Table 6 states each as a concrete action together with the failure it guards against; we intend it to be applied before a result is reported rather than after it is questioned. A fill-in version of the same seven checks, for use during a study rather than after it, is provided as Supplementary Information (Supplementary Checklist, 2026). The protocol is stated in general terms. How much it can be expected to change a result depends on the setting, and Section 6.6 sets out the conditions under which the instability we document appears.

The checks divide into two groups, and the division matters in practice. Checks 1–4 are properties of the *design*. They must be built in beforehand and cannot be applied to a completed study. Checks 5–7 are properties of the *report* and can be satisfied at writing time. The costly ones are in the first group, and in our experience the temptation to economize falls precisely there, most of all on check 4, repeating the whole experiment rather than the generation within it. That is also the check most likely to change the answer. In our own study it was the only procedure that exposed the instability, and it is the one we would have been least likely to run had we not been asked to.

Where resources allow only some of these, the checks are not equally costly and we would not treat them as equally optional. Checks 1 and 2 cost additional generation and nothing else, and without them the variance of the estimate is unknown, so we would regard them as mandatory in any study that makes a comparative claim. Check 3 costs a paragraph of reporting; its omission is what makes the existing literature unreadable, and it is the cheapest of the seven. Check 4 is the expensive one, and it is the one that changed our answer. If only one check can be afforded, this is the one to keep. Checks 5–7 are analysis and reporting conventions that can be applied to a study already run, and are therefore the ones to add when the design can no longer be changed.

These checks complement, rather than replace, an existing testing tradition in multi-agent systems engineering. Baiardi et al. test BDI agent specifications by mapping their control flow onto discrete event simulation, motivated

by the divergence between deployed and simulated systems when surrogates are used [27]. The concerns are adjacent but distinct. That line of work asks whether an implementation behaves as specified, whereas the checks below ask whether an *experiment* can support the claim drawn from it. A system may pass every implementation test and still yield a null result that is an artifact of how it was measured.

Table 6. A checklist for evaluations of memory mechanisms in agentic systems.

Check	Concrete action	Guards against
Repeat generation	Generate at least three independent outputs per condition and aggregate them before testing; report the number of repetitions and the within-condition variance. Three is a floor at which the variance becomes estimable, not a target. Size the count from the ratio of expected effect to single-draw noise	An estimator whose variance is unknown, and therefore a difference that cannot be attributed
Share the baseline	Generate one baseline per repetition and reuse it across all conditions	Confounding condition effects with baseline re-sampling
Report the retrieval key and its resolution	State the key, how many distinct values it takes in the store, and how many entries share each value; record what retrieval returns for a sample of queries	A policy that cannot express a difference, reported as a policy that found none
Repeat the whole experiment	Re-populate the store and re-run end to end at least three times; report the run-to-run variance of the estimate alongside the estimate itself, and report every run rather than a selected one. Three is again a floor at which that variance becomes estimable	Treating a reproducible generation as a reproducible result
Use more than one evaluator	Use at least three judges drawn from more than one model family and independent of the generator under test; report judge identity; test explicitly for ceiling and floor effects; prefer rank-based agreement over correlations pooled across non-independent dimensions	An evaluator that cannot discriminate, which silently destroys the study's resolving power
Use paired within-item designs	Compare conditions within each item; report effect sizes with confidence intervals rather than p-values alone	Losing power to between-item variance
Add deterministic checks	Test safety-critical properties by rule rather than by mean score	Rare severe errors hidden by an acceptable average

Two of the seven carry a number, and both are floors rather than targets. Generate at least three outputs per condition, and run the whole procedure at least three times, because below three neither the within-condition variance nor the run-to-run variance can be estimated at all. Above that floor the count follows the ratio of the effect to the noise of a single draw, and a pilot estimates that ratio directly. For a paired design at 80% power and two-tailed $\alpha=0.05$, the number of items required is close to $8/d_z^2$, where d_z is the effect divided by the standard deviation of the paired differences. An effect half that standard deviation therefore needs on the order of thirty items, one of a fifth needs on the order of two hundred, and one of a tenth needs on the order of eight hundred. Our own ablation illustrates the last case, where a pooled standard deviation of 0.307 against effects of 0.03 to 0.10 put the requirement between 77 and 827 profiles per setting (Section 5.7). A study whose design is sized to a pilot in this way will either resolve its effect or know that it cannot.

6.6 Boundaries of the Claim

We state the scope of our conclusions explicitly, because the boundary conditions are narrow and the claim is easy to over-extend.

What we claim to have shown. That in this pipeline, on this task, with these two generators and this evaluation instrument, the measured effect of experience replay does not reproduce across independent runs, and that the standard error of the estimator is of the same order as the estimate itself. This is a claim about *measurement under these conditions*, not a claim that replay is ineffective in general.

Properties of the setting we expect to matter. Four features of our design seem to us necessary for the phenomenon to appear, and we would not expect it to transfer where they are absent.

1. *A dependent variable whose single-draw noise is comparable to the effect.* Reports are stylistically homogeneous and judge scores cluster near the ceiling, so residual variance across generations is of the same order as the manipulation's effect, and a single draw can land far from the mean. Section 5.9 shows that what matters is the *ratio*, not the subjectivity of the measurement. An objective instrument with an order of magnitude less absolute noise reproduced the same instability, because its standard error shrank along with the signal.

2. *A modest manipulation.* Replay changes the prompt, not the model. Effects of the size we observe (0.03–0.17 on a 10-point scale) are the regime in which replication is decisive. A manipulation that changes the task or the model itself will not be at risk in the same way.

3. *A capable generator with a strong baseline.* Both generators already produce plausible structured reports without replay, which leaves limited headroom. A weaker generator with a poorer baseline has more room for exemplar injection to help, and we would expect effects there to be larger and possibly reproducible.

4. *A single-turn generation task.* Replay here conditions one generation on static exemplars. Whether the same instability obtains where memory is updated across many turns, or where an agent must decide *when* to consult memory, is open; it is also the variant most relevant to long-horizon multi-agent systems.

How far the boundary reaches. Two objective tasks now bracket the phenomenon. The first, a single-selection task with a small effect, reproduced the instability (Section 5.9). The second, a multi-day planning task with a large effect, did not (Section 5.10). The boundary is therefore the ratio of effect to single-draw noise rather than the subjectivity of the measurement or the shape of the task, which is the claim the paper set out to test. What remains open is the regime between the two. Both of our objective tasks sit at one extreme or the other, and we have no estimate of the ratio at which a verdict becomes unreliable. Negotiation, tool use, and task allocation over much longer horizons are also untested, and a longer horizon gives errors more room to compound, so a manipulation could be large in its consequences while remaining small in any single comparison. We would not extrapolate our results to such settings, in either direction.

Generality of the failure modes. The claim of Section 3.3 concerns composition rather than our codebase. A loop assembled from independently implemented store, filter, and retrieval components inherits these hazards unless their semantics are aligned explicitly. We have not surveyed other implementations, and we do not claim a base rate for these failures. What we claim is that they are the kind that interface-level auditing can find and trace-level analysis cannot. That is a claim about method, not about frequency.

Because a failure mode that only its authors can recognize is of limited use, we state the audit as three questions that can be asked of any such implementation without running it. (i) Are the key used to *write* a memory entry and the key used to *retrieve* it produced by the same component, and is their agreement enforced anywhere in the interface?

(ii) Does the filter's treatment of a missing or unrated value match what its documentation says it does? (iii) Does the retrieval function take, as an explicit parameter, the task-relevant attribute the mechanism exists to exploit? A negative answer to any of the three is the hazard; whether it has actually bitten depends on the deployment, which is why we claim the hazard rather than a base rate. Applied to our own system before the repair of Section 3.3, all three answers were negative.

6.7 Threats to Validity

External validity. The study covers one pipeline, one task family, and two model endpoints (Section 4). Neither generator is characterized beyond its behavior in these experiments, and the two are not ordered by an independent capability benchmark, so our model-to-model differences should be read as facts about these two endpoints rather than as effects of scale. The task is a single generative one whose dependent variable is a compressed subjective score; what this implies for the scope of our conclusions is set out in Section 6.6. Inputs are constructed profiles rather than real students, so nothing here licenses a claim about student populations or about deployed school systems.

Model drift. M2's endpoint is a rolling alias. It names a service rather than a frozen set of weights, so the five runs reported here are not guaranteed to have been produced by the same generator, and a replication attempted later may run a different one. This is a threat we cannot quantify. The runs were made against a live endpoint, the weights in use at each point are not retrievable, and no archived copy of the endpoint exists to compare against. We therefore cannot apportion M2's run-to-run spread between estimator noise and drift, and we do not attempt to. Three considerations bound how far this reaches. First, drift is not needed to explain M2. Its five runs are not detectably estimating different effects ($Q=5.70$, $p=0.22$, $I^2=29.8\%$). The spread there is fully accounted for by an estimator whose standard error is comparable to the effect it estimates, and a moving generator would have to operate in addition to a mechanism already sufficient. Second, the caveat was raised for M2 alone, since M1 names a specific model rather than a service alias, and M1 is where the runs disagree most strongly, with heterogeneity that a stable generator leaves unexplained ($Q=14.1$, $p=0.007$, $I^2=71.6\%$). Third, the paper's central claim concerns the resolving power of a single-run estimate, which is a property of the estimator's variance structure; drift could only raise that variance, not lower it, and so could reinforce the conclusion rather than undermine it.

Construct validity. The dependent variable is an LLM judge's mean score over five dimensions. We report internal consistency and, in Section 5.5, both cross-evaluator agreement and a comparison against five expert raters from a psychology department. What those expert ratings show is not encouraging for the instrument. Absolute agreement among the experts is close to zero, rank agreement is weak, and they detected concrete defects in reports the judge scored identically to the rest. What they do not do is settle the condition contrast, because the experts rated a fixed set of twelve reports rather than pairs drawn from the comparisons on which our conclusions are most fragile. The dimensions most central to a mental-health report, namely emotional tone and clinical appropriateness, are precisely those on which automated judges are least reliable. A study with counselor raters could reach different conclusions about the *absolute* quality of these reports; we do not think it would change the central observation about estimator instability, and the expert data offer some support for that, in that the experts were themselves no more able to establish a stable ordering of these reports than the judges were.

The test that would settle the condition contrast remains cheap, and we did not run it. A handful of counselors rating a stratified sample of report pairs, blind to condition, would establish whether expert ordering of the conditions agrees with the judge's on the dimensions we rely on. The sample should be drawn to include the comparisons on which our conclusions are most fragile, namely the runs that disagreed, rather than reports at random, since the question is not the absolute quality of the outputs but whether a human would have reached a different verdict on the same contrast.

Self-judged comparisons. In the main comparisons the judge is the same model as the generator under evaluation (Section 4.5), and its preference for its own output is measurable in our data, at 1.61 points on a ten-point scale (Section 5.5). Two features of the design bound what that can explain. The comparison is paired within profile, and both arms of every pair are generated by the same model, so a preference that is constant across conditions subtracts out of the paired difference. What would not subtract out is a preference that depends on the condition, for example if reports resembling the injected exemplars were rated more favorably, those exemplars being the model's own highly rated outputs. Our design does not separate that possibility from the mechanism, and the expert analysis does not cover it either, since the experts rated a fixed set of reports rather than condition pairs. A bias of that kind would move the level of the estimate, and the claim this paper makes concerns its spread across repetitions, which no stable evaluator preference can account for.

Feedback realism. In our loop, feedback is a scalar rating produced for analysis rather than by working counselors. Real feedback is heterogeneous, noisy, and conditioned on professional context in ways our setup does not represent. Two things should be separated here. At the level of the *mechanism*, the failure modes F1–F3 are properties of how the loop is composed and would arise regardless of who supplies the rating, since none of them concerns a rating's content, only its routing, its filtering, and what it is retrieved against. At the level of *effect size*, real feedback would plausibly make matters worse rather than better. A rating produced for analysis carries no expectation that it will be used, whereas a counselor who rates a report expects the rating to take effect, and F1 and F2 are precisely the failures under which it silently does not. The interface would then present itself as responsive while being inert. That is the same defect the protocol of Section 6.5 is meant to catch, with a human on the receiving end. The effect sizes of replay under realistic feedback nevertheless remain unknown, and this is a prediction we did not test.

Self-generated experience. The experience pool is populated by the same system that is then evaluated against it. Highly rated reports are the system's own outputs, selected by a rating and re-injected as exemplars, so what the loop tests is whether the system benefits from reviewing a subset of its own past output, not whether it benefits from experience acquired elsewhere. This is the architecture the mechanism implies, since an organization's memory is by definition its own record, but it has a consequence for how our null should be read. To the extent that the pool carries little beyond the generator's existing style, retrieval cannot supply much that the generator would not have produced anyway, and a null result then speaks partly to the redundancy of the pool rather than to replay as such. One observation from our own data bears on this. Pool size made no difference because both pools were built from the same fifteen warm-up profiles and returned the same entries (Section 5.3), so within this study the pool's informational content is bounded by those profiles in a way we did not manipulate. Whether replay helps against an externally sourced or human-authored pool is a separate question, and we did not test it. It is the extension we would run first, because it separates the mechanism from the provenance of its content.

7. Conclusion

We subjected the experience-replay mechanism of a five-agent LLM pipeline to replication. After repairing three coordination failure modes that prevented the loop from closing, fifteen controlled comparisons across two models produced no replay effect that reproduces across runs. Significance did not survive replication even where the sign did. Single-run evaluation produced two large and significant effects of opposite sign; a pool-size comparison that had looked decisive in single runs and dissolved under replication; and ten independent runs of an identical configuration, five on each model, produced effect estimates that disagreed about the magnitude of the effect and, on one model, about its direction.

The broader point is methodological. Experience replay and related memory mechanisms are widely adopted on the strength of single-run comparisons, in a regime where the estimator's variance is comparable to the effect being estimated. Under those conditions a single significant result is weak evidence. We offer a seven-point protocol

(Section 6.5) intended to make evaluations of agent memory informative rather than merely confident, and we hope the contradictory pair of runs we report illustrates why it is needed.

For multi-agent systems research the contribution is twofold. Methodologically, the effect of a coordination mechanism, in this case shared experience memory, cannot be reliably estimated from single runs at the effect sizes such mechanisms produce, and a literature that reports them once will accumulate confident findings in both directions out of the same data-generating process. Theoretically, what governs what the mechanism can return is the resolution of its retrieval key rather than the size of the store. That makes the key a first-class design variable. It is chosen, usually without discussion, by whoever implements retrieval, and it bounds what the mechanism can express before any agent consults it. Both results follow from asking what a coordination mechanism can *express* rather than what it *contains*, which we take to be the more productive question for this community to ask of the memory mechanisms it now builds on.

Statements and Declarations

• **Funding.** This work was supported by the following projects: the 2026 University-Level "Pilot Program" Approved List of 100 New-Quality Courses (Teaching [2026] No. 40, No. 82); the 2026 Faculty Capacity Improvement Project of the School of Artificial Intelligence, "Course Development for Physical AI and Digital Twin Application Development"; the 2025 Guangdong Provincial Undergraduate Higher Education Teaching Quality and Teaching Reform Approved Project (Yue Jiao Gao Han [2026] No. 4, No. 97); the South China Normal University University-Level Special Teaching Reform Project (Teaching [2025] No. 7, Item 46); the Guangdong Provincial Joint Postgraduate Training Demonstration Base (866-Yue-Jiao-Yan-Han [2024] No. 1, Item 032); the 2025 South China Normal University Teaching Quality and Teaching Reform Approved Project (Teaching [2025] No. 44, No. 11); the 2025 South China Normal University Special University-Level Teaching Reform Project, "Research and Practice on Teaching Reform of the Course Fundamentals of Cyberspace Security Empowered by Large Models"; and the South China Normal University Special University-Level Teaching Reform Project (Teaching [2023] No. 71, No. 027).

• **Competing interests.** The authors declare no competing interests.

• **Relationship to prior publication.** JAAMAS requires that a submission identify its relationship to any prior archival publication and state its added value. No part of this work has been published elsewhere, and it is not under consideration elsewhere. One *non-archival* prior dissemination is declared for transparency. The platform studied here was submitted as a system entry to the 8th CCF Open Source Innovation Competition (Domestic Open-Source GPU AI Innovation Ecosystem track), with a technical report and a defense presentation (Wang, Wu, He and Liu, 2026). That entry describes the platform as an engineering artifact, covering architecture, GPU acceleration, deployment, and system-level test results, and reports no experimental study of the replay mechanism. The added value of the present submission is the whole of its research content. This comprises the framing of replay as organizational memory and the two predictions derived from it; the interface-level audit and the three coordination failure modes; the fifteen controlled comparisons and the replication study; the retrieval-key resolution analysis and the direct verification of the diversity bound; the evaluator analysis; the cross-task validation on an objective task; and the seven-point evaluation protocol.

• **Ethics approval.** Not applicable. This study uses constructed experimental profiles derived from an anonymized record store. The five expert raters described in Section 5.5 evaluated AI-generated reports; they were not themselves the subjects of study, and no personal data about them or about any student were collected. The record store contains placeholder names only and no contact details.

• **Data availability.** The experimental scripts, the raw result files of every experiment, the complete log of LLM calls, and the figures are openly available at https://github.com/wjh55224545/replication-instability-in-experience-replay. The constructed student profiles, the source record store, and the twelve reports used in the evaluator analysis are included in that repository.

• **Code availability.** The code implementing the pipeline, the replay mechanism, and every analysis reported here is available at the same repository, https://github.com/wjh55224545/replication-instability-in-experience-replay.

• **Author contributions.** Jianhao Wang: Conceptualization, Methodology, Software, Validation, Formal analysis, Investigation, Data curation, Visualization, Writing – original draft; Kaijia Wu: Software, Resources; Dong Liu: Writing – review & editing,Sunding acquisition, Project administration, Supervision; Shouqiang Liu: Conceptualization, Methodology, Resources, Funding acquisition, Project administration, Supervision, Writing – review & editing. All authors read and approved the final version.

• **Use of large language models.** The system under study is itself a pipeline of large language models, and the models used as its components, as the judge, and as the external evaluator are specified in Section 4. Language models were used to assist with the preparation of the manuscript, including English-language editing of author-written text. They were not used to generate experimental results, to compute any statistic reported here, or to decide which results to report; every number in this paper is produced by the analysis scripts released with it.

A. Judge Rubric

The judge received the report text together with the input profile and returned a one-line JSON object containing five integer scores on a 1–10 scale. The dimension definitions were as follows.

• **Clinical appropriateness.** Psychological soundness and terminology, professional but not overly academic.

• **Relevance.** Fidelity to the input emotion data (score, dominant emotion, trend).

• **Empathy.** Warm and constructive, without causing unnecessary alarm.

• **Actionability.** Concrete, executable, targeted suggestions.

• **Completeness.** Presence of the required sections (overview, indicator table, key findings, risk analysis, prediction, suggestions).

The prompt instructed the judge to use the full 1–10 range, not to give all reports high scores, to place ordinary reports in the 5–7 band, to reserve scores above 8 for exceptional reports, and to output only the JSON object.

B. Generation and Injection Details

Generation. Temperature 0.7, top-p 1.0, maximum output 2048 tokens, issued through an OpenAI-compatible endpoint. Each condition within a repetition used an independently sampled generation.

Injection. Retrieved entries were appended to the system prompt as a numbered list under the heading "historical successful cases". Each entry reported the source profile's name, its rating, its risk tier, its dominant emotion, and a 120-character preview of the generated text.

Ranking. Matched retrieval ranked candidates by +2 for a matching risk tier, +1 for a matching dominant emotion, and a bonus of 0.1×rating, taking the top two. Other-tier retrieval restricted candidates to entries matching neither tier nor emotion.

C. Full Result Tables

Per-condition means for the repeated experiments are reported in the data files accompanying this submission. Table 7 gives the ablation results discussed in Section 5.7.

Table 7. Ablation of retrieval parameters. Means are judge scores on a 1–10 scale. The right-hand column reports an earlier run of the same ablation with five profiles per setting (Section 5.7); that run used a different, smaller profile sample and the two columns are not paired.

			n=12		n=5, earlier
Parameter	**Setting**	**Mean**	**SE**	**Mean**	**SE**
top-k	1	8.10	0.127	8.20	0.000
top-k	2	8.12	0.067	8.12	0.080
top-k	3	8.07	0.119	7.80	0.245
rating threshold	3	8.12	0.083	7.60	0.245
rating threshold	4	8.15	0.036	7.72	0.224
rating threshold	5	8.12	0.046	8.00	0.200
injection position	system	8.10	0.083	8.20	0.000
injection position	user	8.05	0.105	8.20	0.000

Table 8. Mean judge score by round under multi-round replay, for the original run and for five independent repetitions of the whole procedure. The experience pool grows from 8 to 40 entries in every run.

	Round 1	Round 2	Round 3	Round 4	Round 5
Original run	8.200	7.850	8.050	8.025	8.175
Repetition 1	8.100	8.075	8.125	8.175	8.025
Repetition 2	8.250	8.150	8.025	7.950	7.875
Repetition 3	8.225	8.175	8.125	8.150	8.250
Repetition 4	8.250	7.975	8.125	8.150	8.200
Repetition 5	8.000	8.025	8.175	7.950	8.200
Pooled	8.165	8.080	8.115	8.075	8.110
SD across repetitions	0.111	0.084	0.055	0.115	0.157
SE across repetitions	0.050	0.037	0.025	0.051	0.070
Experience pool	8	16	24	32	40

References

[1] Shinn, N., Cassano, F., Gopinath, A., Narasimhan, K., Yao, S. Reflexion: Language Agents with Verbal Reinforcement Learning. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2023). https://doi.org/10.52202/075280-0377

[2] Park, J. S., O'Brien, J. C., Cai, C. J., Morris, M. R., Liang, P., Bernstein, M. S. Generative Agents: Interactive Simulacra of Human Behavior. In: *Proceedings of UIST* (2023). https://doi.org/10.1145/3586183.3606763

[3] Liu, Y., Si, C., Narasimhan, K., Yao, S. Contextual Experience Replay for Self-Improvement of Language Agents. In: *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL), Volume 1: Long Papers*, pp. 14179–14198. Association for Computational Linguistics, Vienna, Austria (2025). https://doi.org/10.18653/v1/2025.acl-long.694

[4] Wei, T., Sachdeva, N., Coleman, B., et al. Evo-Memory: Benchmarking LLM Agent Test-Time Learning with Self-Evolving Memory. arXiv preprint arXiv:2511.20857 (2025). https://doi.org/10.48550/arXiv.2511.20857

[5] Hong, S., Zhuge, M., Chen, J., et al. MetaGPT: Meta Programming for a Multi-Agent Collaborative Framework. In: *International Conference on Learning Representations (ICLR)* (2024). https://doi.org/10.48550/arXiv.2308.00352

[6] Wu, Q., Bansal, G., Zhang, J., et al. AutoGen: Enabling Next-Gen LLM Applications via Multi-Agent Conversation. arXiv preprint arXiv:2308.08155 (2023). https://doi.org/10.48550/arXiv.2308.08155

[7] Guo, T., Chen, X., Wang, Y., et al. Large Language Model Based Multi-Agents: A Survey of Progress and Challenges. arXiv preprint arXiv:2402.01680 (2024). https://doi.org/10.48550/arXiv.2402.01680

[8] Walsh, J. P., Ungson, G. R. Organizational Memory. *Academy of Management Review* 16(1), 57–91 (1991). https://doi.org/10.2307/258607

[9] Wegner, D. M. Transactive Memory: A Contemporary Analysis of the Group Mind. In: *Theories of Group Behavior*, pp. 185–208. Springer (1987). https://doi.org/10.1007/978-1-4612-4634-39

[10] Ren, Y., Argote, L. Transactive Memory Systems 1985–2010: An Integrative Framework of Key Dimensions, Antecedents, and Consequences. *Academy of Management Annals* 5(1), 189–229 (2011). https://doi.org/10.5465/19416520.2011.590300

[11] Malone, T. W., Crowston, K. The Interdisciplinary Study of Coordination. *ACM Computing Surveys* 26(1), 87–119 (1994). https://doi.org/10.1145/174666.174668

[12] Madaan, A., Tandon, N., Gupta, P., et al. Self-Refine: Iterative Refinement with Self-Feedback. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2023). https://doi.org/10.52202/075280-2019

[13] Ouyang, L., Wu, J., Jiang, X., et al. Training Language Models to Follow Instructions with Human Feedback. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2022). https://doi.org/10.48550/arXiv.2203.02155

[14] Hu, Q., Long, Q., Wang, W. When Continual Learning Moves to Memory: A Study of Experience Reuse in LLM Agents. arXiv preprint arXiv:2604.27003 (2026). https://doi.org/10.48550/arXiv.2604.27003

[15] Wei, T., Zheng, Y., Wang, Z., Yi, C., Zeng, C., Lu, L., Chen, S., Chen, J. LLM-Guided Contrastive Role Representation for Diverse Multi-Agent Coordination. *Autonomous Agents and Multi-Agent Systems* 40(2), 41 (2026). https://doi.org/10.1007/s10458-026-09769-1

[16] Cemri, M., Pan, M. Z., Yang, S., et al. Why Do Multi-Agent LLM Systems Fail? In: *Advances in Neural Information Processing Systems (NeurIPS)* (2025). https://doi.org/10.52202/085713-4082

[17] Zheng, L., Chiang, W., Sheng, Y., et al. Judging LLM-as-a-Judge with MT-Bench and Chatbot Arena. In: *Advances in Neural Information Processing Systems (NeurIPS)* (2023). https://doi.org/10.52202/075280-2020

[18] Liu, Y., Iter, D., Xu, Y., Wang, S., Xu, R., Zhu, C. G-Eval: NLG Evaluation Using GPT-4 with Better Human Alignment. In: *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pp. 2511–2522. Association for Computational Linguistics, Singapore (2023). https://doi.org/10.18653/v1/2023.emnlp-main.153

[19] Norman, J. D., Rivera, M. U., Hughes, D. A. Reliability Without Validity: A Systematic, Large-Scale Evaluation of LLM-as-a-Judge Models Across Agreement, Consistency, and Bias. arXiv preprint arXiv:2606.19544 (2026). https://doi.org/10.48550/arXiv.2606.19544

[20] Dorner, F. E., Nastl, V. Y., Hardt, M. Limits to Scalable Evaluation at the Frontier: LLM as Judge Won't Beat Twice the Data. In: *International Conference on Learning Representations (ICLR)* (2025). https://doi.org/10.48550/arXiv.2410.13341

[21] Wooldridge, M., Jennings, N. R. Intelligent Agents: Theory and Practice. *The Knowledge Engineering Review* 10(2), 115–152 (1995). https://doi.org/10.1017/S0269888900008122

[22] Horling, B., Lesser, V. A Survey of Multi-Agent Organizational Paradigms. *The Knowledge Engineering Review* 19(4), 281–316 (2004). https://doi.org/10.1017/S0269888905000317

[23] Zoethout, K., Jager, W., Molleman, E. Task Dynamics in Self-Organising Task Groups: Expertise, Motivational, and Performance Differences of Specialists and Generalists. *Autonomous Agents and Multi-Agent Systems* 16(1), 75–94 (2008). https://doi.org/10.1007/s10458-007-9022-9

[24] Cranefield, S. Enabling BDI Group Plans with Coordination Middleware: Semantics and Implementation. *Autonomous Agents and Multi-Agent Systems* 35(2), 45 (2021). https://doi.org/10.1007/s10458-021-09525-7

[25] Dell'Anna, D., Murukannaiah, P. K., Yurrita, M., Dudzik, B., Grossi, D., Jonker, C. M., Oertel, C., Yolum, P. From Human Teams to Hybrid Intelligence Teams: Identifying, Characterizing, and Evaluating Foundational Quality Attributes. *Autonomous Agents and Multi-Agent Systems* 40(1), 10 (2026). https://doi.org/10.1007/s10458-025-09730-8

[26] Open Science Collaboration. Estimating the Reproducibility of Psychological Science. *Science* 349(6251), aac4716 (2015). https://doi.org/10.1126/science.aac4716

[27] Baiardi, M., Burattini, S., Ciatto, G., Pianini, D. Testing BDI-Based Multi-Agent Systems Using Discrete Event Simulation. *Autonomous Agents and Multi-Agent Systems* 40(1), 18 (2026). https://doi.org/10.1007/s10458-026-09744-w