

The Deterministic Sandwich: adding cross-run learning to a Tines AI Agent that has no memory

A method brief for Tines builders. Anthony Stapleton, 2026-09-29.

Abstract

A Tines AI Agent is stateless by design. Every run starts from its instructions and its prompt; whatever it worked out mid-run survives only in the event's `steps[]` until retention, plus whatever its tools wrote. That looks like a limitation. The literature on agent memory says it is the safest starting point you can have, because agents that write their own memory un-gated end up performing worse than agents with no memory at all. The method below keeps the agent stateless and moves every part of "learning" into deterministic Tines actions around it: a pre-hook that retrieves and splices, a post-hook that extracts and admits, a human gate that promotes, and a maintenance step that expires. Each design rule is tied to the research that motivates it, and each component is mapped to a Stories primitive you already have (Records, Resources, case metadata, Event Transforms, prompt pills, Send to Story, Pages).

1. The constraint, stated precisely

In Stories, an AI Agent in task mode is a function of `(instructions, prompt, tools) -> (output, steps[])`. It has no built-in store. It can reach one only through a tool you attach (a Records template, for instance), which is exactly the path this method keeps closed on the write side. Its tools are linked actions (Send to Story, templates, custom tools, remote MCP) that do work and return text. The only durable artifacts a run leaves are the emitted event, whatever its tools wrote, and whatever downstream actions write.

That gives three places to put memory, all outside the model:

Substrate	Read at runtime by	Write path	Fit
Records (typed rows, TTL, cap, evict-oldest)	a Record list action, then an Event Transform	a Record create/update action	telemetry, run traces, candidate queues
Resources (team-scoped JSON or text blobs)	<code>RESOURCE.name</code> in a formula (<code>JSON_PARSE</code> first: a JSON-shaped value arrives as a string)	REST whole-value update, or the element-level append / replace / remove endpoints with <code>if_value</code> for conflict-free edits, or the UI	small curated sets read every run
Case metadata (per-case key/value, filterable)	a case lookup with <code>and_metadata</code>	per-key upsert	per-case state, not lessons

Substrate	Read at runtime by	Write path	Fit
Skills (team-scoped markdown, attached to an AI Agent)	the agent itself: it sees <code>name</code> + <code>description</code> and loads <code>body</code> only when the <code>description</code> matches the task	REST <code>POST/PUT /api/v1/skills</code> (since Aug 2026) or the UI	on-demand curated doctrine; native progressive disclosure

Everything that follows uses these four and nothing else. No vector database, no external memory service.

2. Why not just give the agent a memory

The research question was: can a pipeline of stateless LLM agents improve across runs, on a platform with no agent memory, using only externalized deterministic steps and a human gate? Thirty-three sources were loaded (10 Tier 2, peer-reviewed venue; 19 Tier 3, arXiv and preprint surveys; 4 tool pages), and the load-bearing claims were verified by passage search against the loaded text. Four findings decide the architecture.

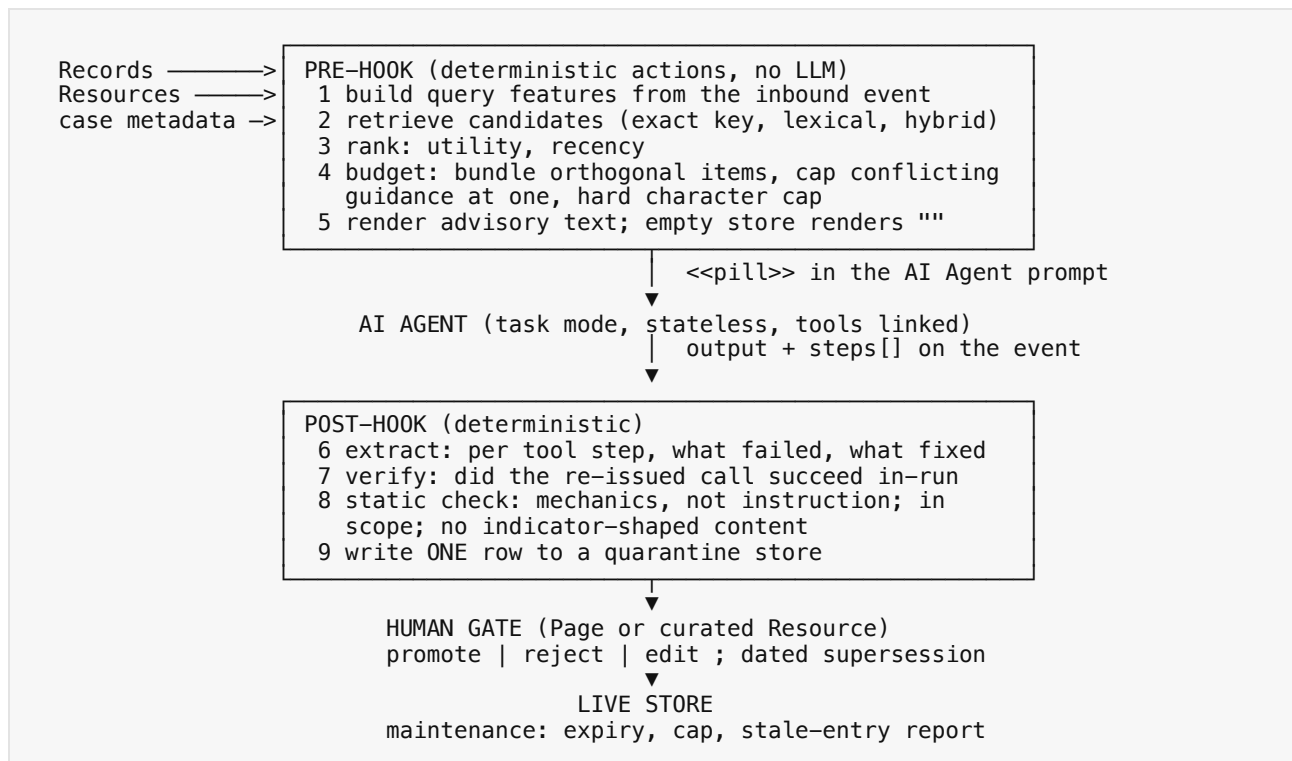
2.1 Un-gated self-written memory degrades below the no-memory baseline. The clearest evidence is the empirical study *How Memory Management Impacts LLM Agents* (ACL 2026, Tier 2) [914ec457]. It measures what the authors call experience-following: an agent's output tracks the similarity of what it retrieved with a correlation near 1, so a retrieved wrong trace is replicated and written back, and the error compounds. Its "Add-All" policy (store every experience) scored 13.05 percent against a 16.75 percent fixed no-memory baseline on EHRAgent, while a strict admission gate lifted AgentDriver from 32.32 percent un-gated to 51.00 percent gated. The *Always-On Agents* survey (Tier 3) [26502729] reports the same shape at the governance level: a governed reducer held all 15 of its safety invariants where a Mem0-style autonomous store held 3 to 4, and the no-memory configuration was the only one that never leaked authority or stale commitments. No source in the corpus reports autonomous memory beating gated memory. Treat the statelessness of the AI Agent as the feature that removes this failure class.

2.2 A success signal verifies the call, not the lesson. Several surveys converge here: *LLM Agents: A Survey* (Tier 3) [dfacca38] lists memory poisoning as an unsolved open problem; the *Automated Design of Agentic Systems* survey (Tier 3) [273a314e] documents reward hacking in self-evolving systems (ADAS and the Darwin Godel Machine removed the detector's markers to fake a passing score). A tool call can return HTTP 200 while ingesting an adversarial payload, and a naive manager stores the poison for a later clean session. So an execution verifier is necessary but not sufficient. The corpus names two gap-closers with empirical validation, both reported through the surveys rather than loaded as primary papers: multi-signal verification (execution plus static checks, as in Eco-Evolve) and capability restriction (as in CaMeL, which gates tool calls so untrusted data cannot drive a control action). Everything else in that space (consensus, provenance DAGs, origin-bound authority) is a 2025-2026 proposal.

2.3 For operational facts, store exact cases, not distilled rules. *ReasoningBank* (ICLR 2026, Tier 2) [49cee0b4] is the strongest distilled-rule system in the corpus and its distiller is instructed to strip specifics: "do not mention specific websites, queries, or string contents." That is correct for strategy and fatal for an operational fix, where the string content is the fix. *Learning from Supervision with Semantic and Episodic Memory* (Tier 3) [7ee08482] found natural-language critiques gave roughly zero benefit on a knowledge-precise task (PubMed), where zero-shot beat memory. *CASCADE: Case-Based Continual Adaptation* (Tier 3) [9c79a753] stores exact (query, fix, result) tuples and reports API-debugging and AIOps root-cause accuracy rising from 20.75 to 55.76 percent; a single study, so treat the magnitude as indicative and the direction as supported. The *Review of Case-Based Reasoning for LLM Agents* (Tier 3) [81ec1ccf] gives the theoretical frame. Layer them: distilled rules for high-level strategy, exact cases for operational facts, executable skills (*Voyager* [802f390d], *Graph-of-Skills* [a3f117fd]) when a fix is worth wrapping in code.

2.4 Memory only pays when problems recur with shared structure. The same PubMed result [7ee08482] shows the other edge: when facts are unique per instance, memory is prompt noise. The memory-management study [914ec457] shows the benefit is domain-sensitive rather than uniform across its benchmarks. For a SOC pipeline this is the scoping rule: platform and tool behavior recurs (a vendor's timeouts, a job's poll budget, a key-shape rule); indicators do not (this domain, this hash, this mailbox). Store the first class. Never store the second. The same rule also bounds the blast radius of a poisoned entry, because a lesson about a header can never touch a verdict about an indicator. And it means: **measure recurrence before you build**. Count distinct failure signatures over a window of runs and how many runs each appears in. If the distribution is long-tail unique, stop; the manual doc you already keep is the right answer.

3. The method



The agent never reads or writes the store directly. It sees advisory text in its prompt and produces output; the sandwich does the rest. Component by component, with the Stories mapping:

Pre-hook. An Event Transform builds the query (lane, tool, error class) from the inbound event. A Record list action or a `RESOURCE` read fetches candidates. Retrieval runs deterministically in formulas or a Run Script: exact composite key first, lexical normalization second (fold case, strip digits and volatile tokens), and only if recall proves inadequate a Run Script computing a hybrid score. The evidence for that ladder is *Graph-of-Skills* [a3f117fd]: exact-key match gave 0 percent recall on any lexical variation; flat vector retrieval over a large skill set scored below no retrieval at all (21.5 versus 27.4 percent) because it returned incomplete bundles; dependency-aware retrieval scored 34.4 percent with 56.7 percent fewer tokens. The lesson is that retrieval quality is a bundling problem as much as a similarity problem. Budget the splice: bundle orthogonal items up to a character cap (*Graph-of-Skills* used at most 8 items and about 12k characters); cap conflicting strategies at one, because *ReasoningBank* [49cee0b4] shows accuracy falling monotonically as more competing strategies are injected (49.7 to 44.4). Render the result as prose under a labeled header, and render the empty string when nothing qualifies, so an empty store is byte-identical to the baseline prompt.

Splice. A pill in the AI Agent prompt, for example `<<DEFAULT(retrieve_notes.text, "")>>`. Pills in the prompt resolve at runtime even though the MCP authoring schema flags the field `formula_compatible: false`; verify by reading `configuration.prompt` on the emitted event, not the action config. Keep static doctrine in `instructions` and per-run material in `prompt`. Add one sentence to the instructions naming the header and saying the text is advisory.

Stateless agent. Unchanged. If it has tools, it keeps its FAILURE link. The AI Agent has no `emit_failure_event` option the way HTTP Request does; measured, it emits nothing on failure without a FAILURE link, and `{"failed": true}` once with one.

Post-hook. An Event Transform maps `steps[]` where `role == "tool"` to `(tool, inputs, status, error text)`. The verifier is the platform's own signal: did the re-issued call return a success status in this run. The static check is a set of formula predicates: the candidate changes mechanics (a header, a parameter shape, a retry decision), not an instruction to the agent; its target is in an allow-list of tool and lane names; it contains no indicator-shaped tokens (IP, domain, hash, email) and no verdict words. Both pass, and one row goes to a quarantine Record type with TTL. Branch on your own presence flags, never on whether an upstream reference errors, because a reference to an action that did not run in this run can resolve to a stale event from an earlier run, or to null, and neither is an error you can catch.

Human gate. A Page listing quarantine rows with promote, reject and edit, or, for a small set, a human editing a Resource through a local validator that enforces the schema. Promotion writes to the live store with a date. If a live entry exists for the same trigger, promotion supersedes it; the old entry is retired, never left to coexist. *Beyond Dialogue Time: Temporal Semantic Memory* (ACL Findings 2026, Tier 2) [a4038bec] is the source for bi-temporal validity windows over silent coexistence, and the memory-management study [914ec457] frames the failure they prevent: contradictory entries accumulating is the canary for a store that has stopped learning.

Maintenance. A scheduled deterministic job: expire entries past their validity window, prune to cap, and produce a report of live entries whose trigger has not recurred in N days. This is where the recurrence measurement from section 2.4 becomes a standing metric rather than a one-time gate.

4. The design rules, each with its reason

1. **Single deterministic writer; agents never write.** Shared stores propagate corrupted state across scope boundaries with no safe contamination threshold (*Always-On Agents* [26502729], drawing on the MESI and TLA+ single-writer disciplines). One Record action writes; one human promotes.
2. **Injection is advisory, never a control path.** A retrieved entry shapes how the agent works, not what it may assert or act on. Enforce it downstream: a deterministic validator on the agent's return that rejects evidence keys and unknown promotions. This is the capability-restriction control the surveys rate as validated [dfacca38, 26502729].
3. **Scope to platform and tool behavior; exclude indicator verdicts.** Section 2.4. It is a recurrence filter and a blast-radius bound at once.
4. **Exact cases, dated, with provenance.** Symptom, wrong, right, rule, evidence reference, date. Section 2.3. Never lossy-consolidate: once the derivation chain is gone, a ground-truth entry and a hallucinated one look the same [914ec457, 26502729].
5. **Every entry anchors to an external outcome, never to another entry.** The loop stays stable only if its gain is below one; an entry derived from an entry is how the gain exceeds one [914ec457].

Tag derivation (ground-truth, reasoning, untrusted lineage) and route anything but ground-truth through the human gate.

6. **Verifier plus static check plus human; no single signal admits.** Section 2.2.
7. **Budget the splice and cap conflicts.** Section 3, pre-hook.
8. **Measure recurrence first, and keep measuring.** Sections 2.4 and 3, maintenance.

5. Two tiers, because not everything needs a human gate

The method has a lightweight form and a full form. Choose by what the entry is.

Telemetry tier (no human gate). Some cross-run facts are counts of outcomes, not lessons: this source has failed four of five calls in the last six hours; this job has missed its poll budget three runs running. Only enumerated statuses, source names and counts are stored, so no free text from a case reaches the store, and no reasoning is stored, so the post-hook writes them automatically and the pre-hook aggregates them into one line. A stateless agent gets what no single run can see: that a source is down right now. This tier often carries the first measurable win, because it stops the agent spending retries and minutes on a source that failed ten minutes ago in another run.

Curated tier (human gate). Anything with a rule in it: when to retry, how to classify a miss, a key-shape requirement, a parameter a tool needs. These are the exact cases of section 2.3, and they go through quarantine and promotion. Expect the set to stay small; a store you can read in one screen is a store a human can govern.

A worked note in the curated tier looks like this:

```
{
  "id": "rn-0001",
  "date": "2026-10-02",
  "trigger": {"lane": "reputation", "source": "vendor_x", "why_match": "gave no HTTP response"},
  "rule": "When vendor_x is the only failing source on a row and SOURCE HEALTH names it, do not re-demand; leave the row inconclusive.",
  "evidence_ref": "trace:11265540",
  "supersedes": null,
  "expires": "2026-12-31"
}
```

`why_match` is a literal substring the agent will see in a tool result, so retrieval is exact and cheap. `evidence_ref` is required. `expires` is bounded. A validator refuses two live notes on one trigger and any indicator-shaped token in `rule`.

6. Preload, a read tool, or a skill

Three composable read paths exist, and the write-side safety argument is the same for all of them because poisoning is a write concern. **A skill** is the platform's own progressive disclosure: the agent sees only the skill's name and description every run and loads the markdown body when the

description matches what it is doing. Since August 2026 skills can be created and updated over REST, so a human-curated notes skill can be written from a repo through the same validator as a Resource, and a rename rewrites the attachment. The cost is an extra model turn and the body's tokens when it loads; measure it for your body size before choosing. **Preload** (the pre-hook splice) is right when the live set is small enough to render under about a thousand characters: zero latency, and it also serves agents that have no tools. **A mounted Send-to-Story read tool** is right when the store is too large to preload; the agent calls it when it hits a problem, index first, detail on demand, and the returning story still ranks and budgets. Measured on our tenant, a Send-to-Story tool call costs about 13 seconds and 4 credits, plus a model decision. If the agent already runs near its time budget, preload. When mounting a tool, note that tools are linked objects that do not appear in the AI Agent's REST options or in the story's action list (the story export carries them by guid only), that "Remove" deletes the tool action with no confirmation (retire by disabling instead), and that the MCP `link_sts_tool` call wants the target story's internal record id, not the URL id.

7. Failure modes and their canaries

If you see	The cause is probably	From
Success falls as the store grows despite budgeting	retrieval returns incomplete or conflicting bundles	[a3f117fd, 49cee0b4]
A verified entry influenced a control decision	injection escaped advisory scope	[dfacca38, 273a314e]
Human review volume exceeds capacity	scope is too broad (indicator facts admitted) or the pre-filter is loose	[7ee08482]
Contradictory live entries for one trigger	supersession is not firing	[a4038bec, 914ec457]
Later iterations degrade on queries that were fine	exposure bias in an un-stopped correction loop	SCIM [072224f1]

8. Platform notes for anyone building this in Stories

Measured on a US-region tenant, September 2026. Probes available on request.

- `timeout` on an AI Agent is enforced when the agent has no tools and, as of late September 2026, was not enforced on a tool-using agent; a stalled request ran about 12.5 minutes to a socket timeout with no retry.
- The AI Agent has no `emit_failure_event` option (HTTP Request does). Without a FAILURE link it emits nothing on failure; with one it emits `{"failed": true}` once.
- `RESOURCE.x` arrives as a string in the formula context even for a JSON-shaped resource (the resources API returns `value` as a string as well); `JSON_PARSE` before any path access. For shared

counters or lists, the element-level append / replace / remove endpoints with `if_value` avoid the read-modify-write race.

- Record filters cannot see JSON or Large-text fields, and values compare as strings. The Record list action returned at most 100 rows a page whatever we asked for (measured August 2026); the REST list caps at 500. Since September 2026 the Records v2 API adds `POST /api/v2/records/aggregate` (COUNT, SUM, AVG, MIN, MAX, MEDIAN, grouped by up to two scalar fields, with a `created_at` time range) and `POST /api/v2/records/search` (filter tree plus time range). For a telemetry tier, write scalar rows and let aggregate do the counting: one call, no paging, no formula aggregation. Verified on our tenant.
- A record TEXT field caps at 512 characters; use JSON or Large text for anything structured.
- A reference to an action that did not run in this run can resolve to a stale event from an earlier run, or to null, and neither is an error. Branch on your own flags.
- Pills in the AI Agent `prompt` resolve at runtime; the schema flag says otherwise.
- Skills have a REST API (list, get, create, update) since August 2026, and an agent loads a skill body only on a description match. Our one measurement of a skill load was slow and credit-heavy, but it was a large body; treat the cost as proportional to body size and measure your own.
- An AI Agent's tools are linked objects that do not appear in the action's REST options; "Remove" deletes the tool action immediately. Retire a tool by disabling it.
- In Tines 3B the substrate differs (a named volume mounted into a step persists across runs) but the governance rules in section 4 are identical: designate a single writer, anchor every entry to an outcome, and keep the agent from writing its own memory un-gated.

Three things would make this pattern a one-afternoon build instead of a plan: a per-agent, human-approved notes list rendered into the prompt with per-note expiry and change control; `timeout` enforced on tool-using agents; and a first-class REST view of linked tools (today only the story export carries them, by guid) with a detach that preserves the target.

9. Sources

Loaded corpus of the research pass (tiers from the source audit: T2 peer-reviewed venue, T3 arXiv or preprint, T- tool documentation). Hash prefixes match the citations above. Note on evidence grade: the direction of every claim in sections 2 and 4 is supported by more than one source; the specific magnitudes quoted are single-study and should be read as indicative. Several named methods (CaMeL, Eco-Evolve, MESI, AOEP, Mem0, ExpeL, MemGPT) are discussed inside the surveys below and were not loaded as primary papers.

id	Tier	Source
914ec457	T2	How Memory Management Impacts LLM Agents: An Empirical Study of Experience-Following Behavior. ACL 2026. https://aclanthology.org/2026.acl-long.27.pdf

id	Tier	Source
49cee0b4	T2	ReasoningBank: Scaling Agent Self-Evolving with Reasoning Memory. ICLR 2026. https://proceedings.iclr.cc/paper_files/paper/2026/file/980ea04d23d1f6908964eba2a74afe45-Paper-Conference.pdf
a4038bec	T2	Beyond Dialogue Time: Temporal Semantic Memory for Personalized LLM Agents. ACL Findings 2026. https://aclanthology.org/2026.findings-acl.1496.pdf
fc55a6e8	T2	ReMem: Reasoning with Episodic Memory in Language Agents. ICLR 2026. https://proceedings.iclr.cc/paper_files/paper/2026/file/5b93eda3b396a7f665a26bf53655adc-Paper-Conference.pdf
d026af30	T2	In Prospect and Retrospect: Reflective Memory Management for Long-term Personalized Dialogue Agents. ACL 2025. https://aclanthology.org/2025.acl-long.413.pdf
bdc47655	T2	Re-ReST: Reflection-Reinforced Self-Training for Language Agents. EMNLP 2024. https://aclanthology.org/2024.emnlp-main.861.pdf
ae9294fe	T2	Deciphering the Interplay of Parametric and Non-parametric Memory in Retrieval-augmented Language Models. EMNLP 2024. https://aclanthology.org/2024.emnlp-main.943/
fb801ccc	T2	Multi-Agent Design: Optimizing Agents with Better Prompts and Topologies. OpenReview. https://openreview.net/pdf?id=I05H9RUzHB
802f390d	T2	Voyager: An Open-Ended Embodied Agent with Large Language Models. OpenReview. https://openreview.net/pdf/625b7da181479e7642abce270739da66290f0fa3.pdf
072224f1	T2	SCIM: Self-Correcting Iterative Mechanism for Retrieval-Augmented Generation. MDPI Electronics 15(5):996. https://www.mdpi.com/2079-9292/15/5/996
9c79a753	T3	CASCADE: Case-Based Continual Adaptation for Large Language Models During Deployment. arXiv 2605.06702. https://arxiv.org/html/2605.06702v1
a3f117fd	T3	Graph-of-Skills: Dependency-Aware Structural Retrieval for Massive Agent Skills. arXiv 2604.05333. https://arxiv.org/html/2604.05333v4
7ee08482	T3	Learning from Supervision with Semantic and Episodic Memory: A Reflective Approach to Agent Adaptation. arXiv 2510.19897. https://arxiv.org/pdf/2510.19897
26502729	T3	Always-On Agents: A Survey of Persistent Memory, State, and Governance in LLM Agents. arXiv 2606.30306. https://arxiv.org/html/2606.30306v1
dfacca38	T3	LLM Agents: A Survey. Preprints.org 202608.0265. https://www.preprints.org/manuscript/202608.0265
273a314e	T3	Automated Design of Agentic Systems: A Survey of Algorithms for Searching, Optimizing, and Evolving LLM Agents, Workflows, and Prompts. Preprints.org 202606.0238. https://www.preprints.org/manuscript/202606.0238
81ec1ccf	T3	Review of Case-Based Reasoning for LLM Agents: Theoretical Foundations, Architectural Components, and Cognitive Integration. arXiv 2504.06943. https://arxiv.org/html/2504.06943v1
91e2111b	T3	Cognitive Memory in Large Language Models. arXiv 2504.02441. https://arxiv.org/html/2504.02441v1

id	Tier	Source
e01b9662	T3	From Human Memory to AI Memory: A Survey on Memory Mechanisms in the Era of LLMs. arXiv 2504.15965. https://arxiv.org/html/2504.15965v1
92e23634	T3	From Storage to Experience: A Survey on the Evolution of LLM Agent Memory Mechanisms. Preprints.org 202601.0618. https://www.preprints.org/manuscript/202601.0618
c316d15e	T3	A Comprehensive Survey of the LLM-Based Agent: The Contextual Cognition Perspective. Preprints.org 202604.0935. https://www.preprints.org/manuscript/202604.0935
f8a7072b	T3	A Self-Reflective Multi-Agent Collaboration Framework. Preprints.org 202603.0129. https://www.preprints.org/manuscript/202603.0129/v1/download
21be2290	T3	Multiple Memory Systems for Enhancing the Long-term Memory of Agent. arXiv 2508.15294. https://arxiv.org/html/2508.15294v1
80d2c5fd	T3	Optimizing Instructions and Demonstrations for Multi-Stage Language Model Programs. arXiv 2406.11695. https://arxiv.org/html/2406.11695v1
de1cece5	T3	Remember When It Matters: Proactive Memory Agent for Long-Horizon Agents. arXiv 2607.08716. https://arxiv.org/html/2607.08716v1
84a4e990	T3	Skill Retrieval Augmentation for Agentic AI. arXiv 2604.24594. https://arxiv.org/html/2604.24594v1
2215e657	T3	Reflexion: Language Agents with Verbal Reinforcement Learning. arXiv 2303.11366. https://arxiv.org/abs/2303.11366
a753376b	T3	Generative Agents: Interactive Simulacra of Human Behavior. arXiv 2304.03442. https://arxiv.org/abs/2304.03442
9882e264	T3	Language Agent Tree Search Unifies Reasoning, Acting and Planning in Language Models. arXiv 2310.04406. https://arxiv.org/abs/2310.04406
34132c97	T-	SoK: Agentic Skills, Beyond Tool Use in LLM Agents. alphaXiv 2602.20867. https://www.alphaxiv.org/abs/2602.20867
bfa2c328	T-	DSPy: Overview. https://dspy.ai/
28c39fea	T-	TextGrad: Automatic "Differentiation" via Text. https://github.com/zou-group/textgrad
254ef3aa	T-	MemoryBank-SiliconFriend (source and demo). https://github.com/zhongwanjun/MemoryBank-SiliconFriend

Tines platform behavior in section 8 is from probes on our tenant and the public docs (<https://www.tines.com/docs/actions/types/ai-agent/>), not from the research corpus.