REPLAYCLAIM: EXECUTABLE RETURNED SUPPORT FOR PERSISTENT MULTIMODAL CLAIMS

Anonymous authors
Paper under double-blind review

ABSTRACT

A claim stored from an invoice can return with its original OCR crop and still be unsafe to reuse: the read path must bind the amount to the requested field, validate current service state, and suppress contradictory claims before an answer or tool consumes it. This missing read contract defines the *persistence gap*; ReplayClaim implements it as a returned-support object that stores evidence with typed verifier routes, calibration and model versions, write-time outputs, and provenance. Its read operator executes query-conditioned verification, constructs a contradiction-consistent served set, and sends invalid state to quarantine or re-grounding; *decision completeness* denotes whether the object reconstructs this declared read contract. Across 74 tool workflows with downstream queries at 5- and 10-session horizons, ReplayClaim raises $K{=}10$ success from $70.8 \pm 1.9\%$ to $72.6 \pm 1.8\%$ versus an equally provisioned evidence cache ($p{=}0.002$) and reduces unsupported actions from 47/900 to 38/900. A complementary one-write/one-read benchmark on DocVQA, ChartQA, and ScienceQA compares returned objects inside a pre-declared $83.5\% \pm 0.5\%$ coverage band: unsupported claims fall from $8.7 \pm 1.0\%$ to $7.6 \pm 1.0\%$ (paired 95% CI [0.5,1.7], $p{=}0.0018$), and method-blind truth error falls from 8.2% to 6.4%. Lifecycle enforcement further reduces stale-state attack success from 28.2% to 9.4%, while the cache gap persists with LLaVA-1.5-13B ($9.8 \pm 1.2\%$ to $8.6 \pm 1.1\%$) and after manual parser cleanup. The results make returned support an executable decision object and expose safe claim reuse as a property of the read contract, not evidence retention alone.

1 INTRODUCTION

A document agent stores the claim "the invoice total is $186.40" together with the crop that produced it. When a later session asks for the amount to pay, those pixels are necessary but not sufficient: the service must bind `186.40` to the *total* field requested now, check the active model and calibration state, and exclude a conflicting amount before a payment tool consumes the claim. The failure is consequential because one unsupported write can influence several later answers or actions. It also survives a seemingly strong design choice—returning the original evidence—when that evidence is merely displayed beside a historical score.

ReplayClaim turns the returned evidence into an executable object. It persists the claim, region or OCR witness, typed verifier route, calibration and model versions, write-time outputs, and provenance. At read time, the interface runs the declared query-conditioned gate, resolves the admitted claims as a set, and records serve, rejection, quarantine, or re-grounding. Figure 1 summarizes this transition from inspectable evidence to executable decision state. Across changed-query workflows, the resulting advantage over an evidence cache grows from 1.1 success points at $K{=}5$ to 1.8 at $K{=}10$; at the longer horizon, unsupported actions fall from 47/900 to 38/900.

We call the missing link between write support and read admissibility the *persistence gap*. The paper operationalizes it with a fixed target read contract: a reusable claim must pass typed verification for the current query and service version, coexist with the other served claims, and enter an explicit lifecycle state. A returned object is *decision-complete* when its persisted fields, the query, and bounded service state reconstruct those decisions. This criterion makes inspectable support and semantic truth separately testable, allowing the source audit below to verify whether the returned-object ordering also holds for correctness.

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

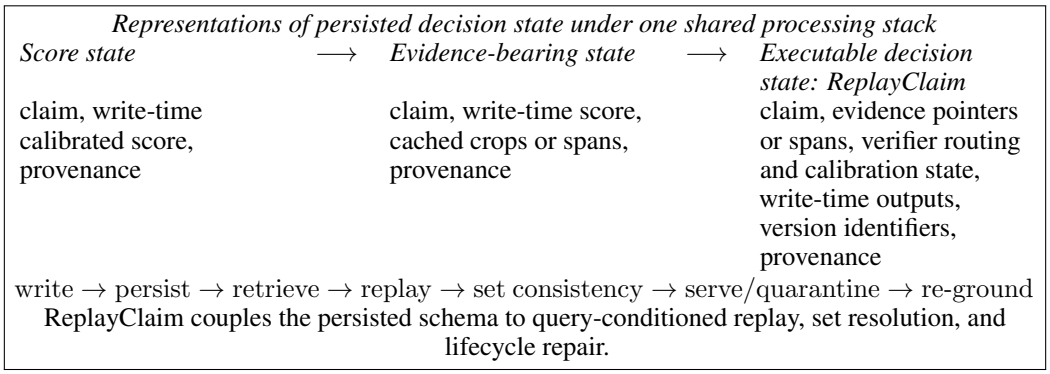


Figure 1: Decision-state schematic. The controls carry progressively richer state under identical processing components; ReplayClaim couples its persisted witness to query-conditioned replay, set resolution, and lifecycle repair.

Existing evaluations illuminate adjacent stages but use a different unit of analysis. Hallucination and citation benchmarks score current responses or answer-local attribution (Li et al., 2023b; Guan et al., 2024; Gao et al., 2023; Asai et al., 2024). LongMemEval and multimodal-memory benchmarks evaluate indexing, retrieval, updating, and reasoning over histories (Wu et al., 2025; Bei et al., 2026; Huang et al., 2026; Liu et al., 2026; Guo et al., 2026), while MIRIX, M2A, and AdaMM design stores for long-term multimodal interaction (Wang & Chen, 2025; Feng et al., 2026; Tian et al., 2026). TierMem escalates from summaries to provenance-linked raw evidence (Zhu et al., 2026), and LedgerMind binds complete reasoning trajectories to active evidence and typed repair (Du et al., 2026). Our unit begins after retrieval and serialization: one claim, its returned witness, and the exact operation that authorizes its next use. This exposes a comparison unavailable from storage or trajectory metrics alone—whether an evidence cache and an executable returned object behave differently when candidate evidence and processing components are held equal.

The primary contribution is a claim-level read-contract evaluation and its executable object. The contract specifies the persisted state, replay operation, set composition, and lifecycle transition; ReplayClaim implements all four with a versioned witness and typed replay operator. The main returned-object study then compares independently calibrated operating points at matched coverage and compute on DocVQA, ChartQA, and ScienceQA: ReplayClaim reduces unsupported returns from $8.7 \pm 1.0\%$ to $7.6 \pm 1.0\%$ relative to the evidence cache, with an independent truth audit showing $8.2\% \rightarrow 6.4\%$. Operator interventions, parser cleanup, distribution shifts, interface attacks, a second VLM, and the 74 multi-session workflows connect that object-level result to its failure modes and downstream consequences.

2 Decision-Complete Persistence with ReplayClaim

Target read contract. We fix one method-independent persistent-claim contract before comparing interfaces:

$$\mathcal{C}^\star = (\mathcal{O}^\star, R^\star_\theta, G^\star_\Gamma, U^\star),$$

where $\mathcal{O}^\star$ names the state required at reuse, $R^\star_\theta(\cdot; q)$ is the versioned gate for query q, $G^\star_\Gamma$ resolves a candidate set under contradiction policy Γ, and $U^\star$ updates lifecycle state. Here θ indexes the replay and model-version configuration, while Γ fixes the contradiction policy. For returned objects O_q retrieved for query q, the served set is

$$S_q = G^\star_\Gamma\big(\{\mathrm{claim}(o) : o \in O_q,\ R^\star_\theta(o; q) = 1\}\big),$$

and $U^\star$ records serve, rejection, quarantine, and re-grounding transitions. An object schema is decision-complete when the persisted object, q, and versioned service state suffice to reconstruct this fixed target contract. The criterion is not a truth certificate: a reconstructible serve decision can still be semantically wrong, which the independent source audit in Section 4 measures separately.

Write and persisted object. Given claim c from frame I, GroundingDINO-T (Swin-T) proposes n_{prop}=8–32 regions (Liu et al., 2024b), followed by SAM segmentation (Kirillov et al., 2023). The

applicable claim-type verifiers $V_t(c, r)$, $t \in T(c)$, score each candidate. For an evidence pack E and query context q, the calibrated aggregate is

$$P_{\text{agg}}(c, E; q) = \frac{1}{|T(c)|} \sum_{t \in T(c)} \sigma\big(V_{t,v(c)}(c, E; q)/T_t\big) .$$

At write time, candidates are sorted by their maximum applicable calibrated verifier probability, with proposal index breaking ties. The shared greedy pack is constructed once, before method-specific admission, and adds a candidate only when its intersection-over-union with every selected item is below 0.5. It stops when $P_{\text{agg}}(c, E; q_{\text{write}}) \geq \tau_{\text{w}}$ or the evidence cap k is exhausted. The write threshold τ_{w} is fixed across methods and is distinct from each method's read gate g_m. ReplayClaim persists

$$\text{RC}(c) = \big(c, E(c), M(c), s(c), \pi(c), v(c)\big),$$

where $M(c)$ records verifier types and calibration parameters, $s(c)$ stores write-time calibrated outputs, $\pi(c)$ records provenance, and $v(c)$ identifies the model/configuration version required for deterministic replay. The fields have separable reconstruction roles: removing $E(c)$ breaks returned support and verifier execution; removing $M(c)$ breaks route and calibration lookup; removing $\pi(c)$ or $v(c)$ breaks provenance and version validity; and removing the write-time output $s(c)$ prevents a faithful comparison with the original decision. The score-plus-cache object retains $c, E(c), s(c), \pi(c)$ but uses $s(c)$ as its read decision; ReplayClaim additionally makes route, calibration, and version state executable.

State transition and invariant. The service implements

$$\text{write} \to \text{persist} \to \text{retrieve} \to \text{replay} \to \text{resolve} \to \text{serve/quarantine} \to \text{re-ground}.$$

Its invariant is: every served claim carries the evidence and calibrated state required to reproduce its query-conditioned reuse decision, and every served set satisfies the declared contradiction policy. For query q, candidate set $C(q)$, and served set $S(q) \subseteq C(q)$, replay recomputes each applicable verifier logit from the persisted evidence,

$$\widetilde{z}_t(c, q) = V_{t,v(c)}(c, E(c); q),$$

and admits a claim-level candidate only when

$$R^{\star}_{\theta}(\text{RC}(c); q) = \mathbb{1}\left[\text{valid}(E(c), \pi(c), v(c)) \wedge \frac{1}{|T(c)|} \sum_{t \in T(c)} \sigma(\widetilde{z}_t(c, q)/T_t) \geq g_{\text{RC}}\right] .$$

Here valid checks evidence integrity, provenance, and version availability; the evidence-cache control instead applies its calibrated read gate g_{cache} to $s(c)$ while exposing $E(c)$ for inspection. Temperatures T_t are fitted on a held-out 20% training partition without test overlap (Guo et al., 2017). Analogous to carrying a checkable witness (Necula, 1997), the stored object supports the declared read gate; the source audit measures truth as a separate outcome.

Where the reuse query enters. Query conditioning is typed rather than appended to a generic score. For `OCRFact`, q supplies the requested entity, field role, and normalized value against which OCR spans are bound. For `Presence` and `Attribute`, it supplies the entity and attribute phrase to the crop probe; `Count` additionally defines the class and qualifiers whose non-overlapping instances are counted. For `Spatial`, q specifies the ordered entity pair and relation tested by CLIP and geometry. The query binding $\rho(c, q)$ is deterministic after parser normalization within the fixed applicable verifier family $T(c)$; unsupported bindings abstain. Thus the tested read semantics comprise current-query routing and verifier execution together.

Set consistency and revocation. OCR normalization and numeric/text rules create high-confidence contradiction edges; DeBERTa-v3-large NLI (He et al., 2021a) handles ambiguous pairs at threshold 0.70. The resolver builds an undirected contradiction graph over replay-admitted claims, sorts vertices by descending $P_{\text{agg}}(c, E(c); q)$, then by recency and stable claim identifier, and greedily inserts a vertex only if it has no edge to the selected set. This rule resolves isolated pairs, overlapping conflicts, and cycles without order ambiguity. Three rejections within a lifecycle window W quarantine the claim and trigger re-grounding; workflow experiments set W to the evaluated horizon. Set-level resolution is contract semantics: it prevents locally supported but jointly incompatible state from reaching a consumer.

Claim extraction and verifier routing. A fixed parser maps short declarative spans to `Presence`, `Attribute`, `Count`, `Spatial`, or `OCRFact`; it normalizes entities, coreference, numbers, and units before verifier routing and contradiction construction. `OCRFact` combines the PP-OCRv3 recognizer distributed in PaddleOCR v2.7 (Li et al., 2022a) with string agreement at confidence at least 0.85. `Presence`/`Attribute` use an InternVL2-8B crop probe (Chen et al., 2024) with GroundingDINO agreement, `Count` adds cross-crop consistency, and `Spatial` uses CLIP ViT-L/14 (Radford et al., 2021) with geometric checks. The parser is held identical across returned-object controls; a held-out intrinsic audit and a manually cleaned sensitivity slice quantify its contribution in Table 6.

Evaluation quantities. *Returned-support adequacy* asks whether the object surfaced at reuse supports the complete served claim. *Semantic correctness* asks whether that claim is true in the source media and task context. The former measures inspectable decision support; the latter supplies an independent truth-oriented bridge. Coverage is the fraction of queries receiving a non-abstained answer, false rejection (FR) is the rejection rate for adequate claims, and overhead (OH) is wall-clock increase over text-only memory under matched context, caching, and timeouts.

3 Write-to-Reuse Evaluation

Two complementary reuse estimands. Each logged reuse opportunity is a tuple

$$u = (i, q_{\text{write}}, c, \text{RC}(c), q_{\text{reuse}}, C(q_{\text{reuse}}), h),$$

where i is a source item, c is a declarative claim parsed from the write-pass answer, and h is the number of intervening sessions. The one-write/one-read VQA study processes an evaluation image and question, serializes the resulting claims, removes the generation trace, and reloads them with $q_{\text{reuse}} = q_{\text{write}}$ and $h = 0$. It estimates returned-object quality at coverage-matched operating points after serialization. The workflow study estimates the complementary changed-query effect: later tool decisions issue new downstream queries at $h \in \{5, 10\}$, requiring the stored witness to execute in a new read context. Calibration and gate selection use only a held-out 20% training partition; all reported opportunities come from the official evaluation partitions below, and no source item crosses the calibration/evaluation boundary.

Candidate and decision protocol. The study holds retrieval fixed rather than treating it as an experimental factor. For the VQA study, $C(q)$ is the logged, stably ordered set of parser outputs linked to the source item before any method-specific gate; for tool workflows, it is the common set of schema-eligible claims written in earlier sessions. Every method receives the same q, $C(q)$, evidence proposals, shared pack built with τ_{w}, and reuse-opportunity order. A method then applies its persisted-object gate, the shared contradiction policy, and its serve/abstain transition; the conditioned answer is scored only after that decision. This defines the estimands used below: support risk among served claims, query-level coverage and false rejection over all opportunities, task utility over the benchmark queries, and workflow success over complete chain runs.

Models, data, and hardware. The primary VLM is Qwen2.5-VL-32B-Instruct (Bai et al., 2025) in bf16. Perception uses GroundingDINO-T and SAM; OCR uses PaddleOCR v2.7; NLI uses DeBERTa-v3-large (MNLI); the spatial verifier uses CLIP ViT-L/14 with a two-layer MLP; and the VQA verifier is InternVL2-8B in 4-bit GPTQ. Experiments run on four NVIDIA A100 80 GB GPUs with CUDA 12.4, driver 550.54.15, PyTorch 2.3.1, and Flash Attention 2.5.8. We use five seeds $\{13, 37, 61, 71, 97\}$. Datasets are the DocVQA v1.0 validation and test evaluation partitions (5,188/5,349 questions) (Mathew et al., 2021), ChartQA test-human (1,250) (Masry et al., 2022), and the ScienceQA image test subset (4,241) (Lu et al., 2022). The primary query pool therefore contains 16,028 opportunities per seed; the two DocVQA partitions remain separate evaluation strata and are pooled only for the reported DocVQA task metric. The LLaVA-1.5-13B transfer rerun uses the common 10,840-opportunity test-only subset: 5,349 DocVQA test, 1,250 ChartQA, and 4,241 ScienceQA queries.

Parity and controls. All methods share the VLM, extractor, proposal/OCR outputs, verifier family, calibration partitions, prompts, context window (4,096 tokens), 500 ms verifier timeout, caches, contradiction rules, random seeds, and logged reuse opportunities. The controlled factors are the

persisted fields and required read operations. Broad baselines include text-only memory, region memory, citation-grounded generation (Gao et al., 2023), a post-hoc detector (Manakul et al., 2023), and verifier-gated memory. The decisive controls are score-only + set consistency, score + evidence cache + set consistency, and full ReplayClaim. In VQA, the cache comparison is an operating-point comparison between a persisted-score object and the complete returned object; the changed-query workflows identify the downstream effect of mandatory replay. Coverage-matched selection sweeps a predeclared read-gate family g_m and chooses the point closest to 83.5% inside a $\pm 0.5\%$ band; the shared write-pack threshold τ_{w} and evidence cap $k{=}2$ are never retuned by method, and budget matching fixes $B_V{=}500$ ms. Table 5 records the selected gates and realized coverages.

Labels and inference. Returned-support labels use three independent annotators and majority vote on 2,400 served claim–answer pairs per method (1,200 DocVQA, 600 ChartQA, 600 ScienceQA), allocated evenly across the five seed runs, with $\kappa{=}0.81$. Annotators inspect the claim, returned object, complete evidence set, query context, and conditioned answer; partial support is inadequate. Scores and replay outcomes cannot satisfy the rubric without visual or OCR evidence for every entity, value, attribute, and relation in the claim. False rejection uses the common pre-decision candidate-adequacy labels, including candidates rejected by a method, rather than the served-only support sample. Semantic correctness uses 2400 served claims per method, equally stratified over datasets and balanced by claim type. Three method-blind annotators inspect the source media and task context, achieve $\kappa{=}0.84$, and decide by majority vote. Table 2 applies the same support rubric to this semantic sample, so its adequacy/truth cross-tab is separate from the 1,200/600/600 primary support sample. Headline unsupported percentages and Table 1(a) are computed on the 2,400-item-per-method audited support sample; they are equal-weight means of the three dataset rates, and raw labeled counts remain stratified by dataset. Means and standard deviations are over five seeds. The cache–ReplayClaim support contrast uses 2,400 stratified, seed-aligned source-item pairs on which both methods serve; abstentions remain in query-level coverage and false-rejection outcomes. Truth-error differences use a separate 2,400-pair semantic sample, aligned by source item and seed; opportunities where either method abstains remain coverage outcomes rather than receiving an imputed truth label. Paired query-level bootstrap intervals and permutation tests use source items as the resampling and exchange unit at $\alpha{=}0.05$; workflow inference clusters repeated decisions by workflow identity. Frontier plots visualize the full gate sweep; inferential claims use the predeclared matched band and the tabulated paired tests.

4 Results

4.1 Returned-Object Quality at Matched Coverage

The results answer whether the returned object improves support and truth at a fixed serving rate, which operators matter under their declared interventions, and whether executable support changes decisions under later queries. Tables 1 and 2 address the first question. The first compares persisted-score and complete-object operating points under matched coverage and budget; the second tests the ordering with independent semantic adjudication. Figure 2 supplies the broader comparison, showing that answer-time and citation-oriented controls can retain similar coverage while returning less adequate support. At their predeclared coverage-matched gates, the evidence cache reaches $8.7 \pm 1.0\%$, while ReplayClaim reaches $7.6 \pm 1.0\%$: 1.1 points absolute and 12.6% relative (95% CI [0.5,1.7], $p{=}0.0018$). The score-only $\rightarrow$ cache $\rightarrow$ ReplayClaim ladder is $9.2\% \rightarrow 8.7\% \rightarrow 7.6\%$: retaining evidence improves the operating point, but does not recover the complete returned object's result. Against the verifier-gated anchor, the corresponding change is $9.6 \pm 1.1\% \rightarrow 7.6 \pm 1.0\%$, a 2.0-point absolute and 20.8% relative reduction (paired 95% CI [1.3,2.7], $p{=}0.0004$). ReplayClaim also lowers false rejection from $3.1 \pm 0.7\%$ to $2.8 \pm 0.6\%$, while p95 overhead changes from 11.1% to 11.6%.

Task utility moves consistently rather than being hidden in a heterogeneous aggregate. At matched coverage, ReplayClaim versus verifier-gated memory yields DocVQA ANLS 67.4% versus 65.2%, ChartQA relaxed accuracy 64.8% versus 63.1%, and ScienceQA exact match 66.0% versus 64.2%. The component-matched score-only control obtains 65.8%, 63.6%, and 64.7%; the evidence cache obtains 66.4%, 64.0%, and 65.3%. Raw method/dataset support denominators appear in Table 10.

Table 1: Claim-bearing returned-object comparison. Coverage matching selects predeclared gates inside the $83.5\% \pm 0.5\%$ band; budget matching fixes B_V=500 ms. Unsupported is the equal-weight mean of dataset-level percentages on the audited served-claim sample (n=2,400 per method).

Component-matched returned objects					
Method	Gate	Cov. (%)	Unsp. macro (%)	FR (%)	OH p95
Verifier gate	0.68	83.4	9.6 ± 1.1	3.6 ± 0.8	10.8%
Score-only	0.69	83.2	9.2 ± 1.1	3.4 ± 0.7	10.9%
Evidence cache	0.67	83.7	8.7 ± 1.0	3.1 ± 0.7	11.1%
ReplayClaim	0.70	83.5	7.6 ± 1.0	2.8 ± 0.6	11.6%

Table 2: Method-blind semantic-truth audit. Each row contains 2,400 truth judgments; the paired nearest-control ordering agrees with the returned-support comparison in Table 1.

Method	Adequate (true/false)	Inadequate (true/false)	Truth error (95% CI)
Verifier gate	2133 / 37	49 / 181	9.1% [7.9,10.3]
Evidence cache	2159 / 32	44 / 165	8.2% [7.1,9.3]
ReplayClaim	2196 / 22	50 / 132	6.4% [5.4,7.4]

Semantic adjudication gives the same conclusion. ReplayClaim lowers truth error by 1.8 points relative to the evidence cache (paired 95% CI [0.8,2.8], p=0.006). Both disagreement directions remain visible: adequate-but-false claims reveal source interpretation errors, whereas inadequate-but-true claims reveal correct conclusions with insufficient returned support. The returned object is therefore informative without making support adequacy tautological with truth.

The held-out parser audit contains 1800 gold reusable claims and 1773 outputs, with 94.1% precision and 92.7% recall. After manual canonicalization, the nearest-control unsupported rates are 8.3% for the evidence cache and 7.2% for ReplayClaim (paired delta CI [0.4,1.8]). Table 6 reports raw counts, confidence intervals, routing accuracy, normalization accuracy, and contradiction-edge errors.

4.2 Operator Attribution across Replay, Composition, and Repair

Table 3 tests four contract transitions at their declared matched settings. Complete-object admission improves on a cache carrying identical evidence; set resolution lowers unsupported returns at identical query coverage; version-aware recalibration recovers support and coverage under shift; and lifecycle enforcement sharply reduces stale-state attack success. Because the rows use different interventions and operating points, each supports its stated operator conclusion rather than a factorial decomposition of the headline effect. `Count` and `Spatial` remain the most demanding claim types, while the verifier-stack progression is reported with the operating protocol in Table 5.

4.3 Shift, Attack, and Cross-Stack Persistence

We test three distribution shifts (OCR blur, layout, and chart style), one proposal-pool stress, and a black-box memory-interface attacker. The attacker can select claims, evidence candidates exposed by the interface, and repeated verifier probes, but cannot modify source pixels or access model internals. Figure 3 shows that fixed thresholds degrade gracefully and 200 labels per claim type recover most calibration loss. Table 7 adds the component-matched cache and a LLaVA-1.5-13B base stack; Table 8 connects each condition to its protected object, trust boundary, comparator, measured outcome, and residual failure.

On the common 10,840-opportunity test-only subset with LLaVA-1.5-13B (Liu et al., 2024a), unsupported returns are $10.5 \pm 1.3\%$ for verifier-gated memory, $9.8 \pm 1.2\%$ for the evidence cache, and $8.6 \pm 1.1\%$ for ReplayClaim. The nearest-control difference is 1.2 points (95% CI [0.4,2.0], p=0.007). ReplayClaim improves DocVQA, ChartQA, and ScienceQA task metrics over the cache by 1.6, 1.2, and 1.4 points. The same returned-object ordering supports a shared read-contract explanation across the two evaluated base VLMs.

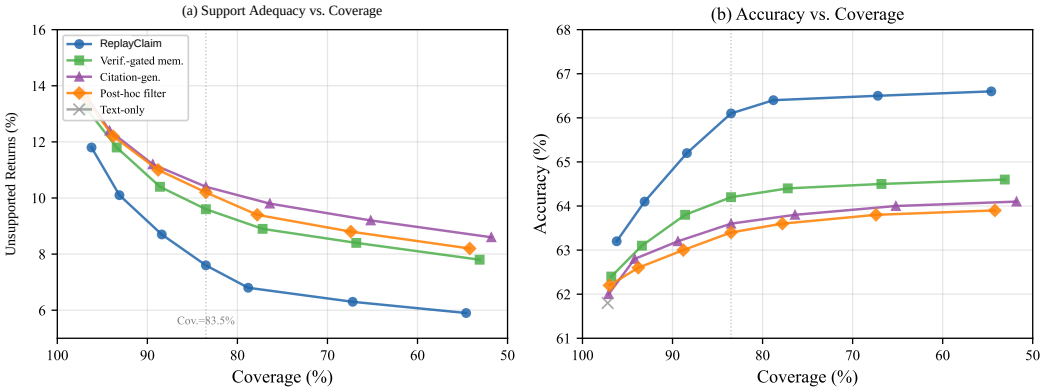


Figure 2: Returned-support and utility frontiers against answer-time, citation-oriented, and verifier controls. Panel (a) shows lower unsupported-return rates at the same served coverage; panel (b) shows the corresponding task utility. Both horizontal axes run from high coverage on the left to more selective serving on the right. The dashed point marks ReplayClaim at g_{RC}=0.70, coverage $= 83.5\%$; Table 1 provides the returned-object comparison in this operating band.

Table 3: Contract-operator interventions and claim-type diagnostics. Panel (a) reports one transition per row under its stated matched setting; the remaining stack is held fixed within that row, but settings differ across rows. ASR and FA denote attack success and false alarms. Panel (b) reports ReplayClaim by claim type, uses eligible queries as the coverage denominator, and defines ΔH as its absolute unsupported-return reduction relative to Text-only memory.

(a) Decision-completeness operators

Operation	Controlled transition and matched setting	Observed consequence
Complete-object admission	Cached write-score gate $\rightarrow$ replay-object gate; same evidence and $83.5\% \pm 0.5\%$ coverage	Unsupported $8.7 \pm 1.0 \rightarrow 7.6 \pm 1.0$; p95 OH $11.1 \rightarrow 11.6\%$
Set composition	Resolver off $\rightarrow$ consistent served set; both at 85.8% query coverage	Unsupported $9.4 \pm 1.1 \rightarrow 7.8 \pm 1.0$; accuracy $65.2 \rightarrow 65.8$
Calibration replay	Frozen $\rightarrow$ version-aware calibration; three shifts plus pool stress	Unsupported improves 1.3–1.8 points; coverage recovers 3.6–4.6 points
Lifecycle repair	Quarantine/re-grounding off $\rightarrow$ on; replay gate and resolver fixed; staleness attack	ASR $28.2 \pm 2.6 \rightarrow 9.4 \pm 1.6$; FA 2.6 ± 0.7

(b) ReplayClaim claim-type diagnostics

Type	Prec.	Type Cov.	ΔH	ECE	FR
`OCRFact`	95.6 ± 1.2	84.1 ± 2.0	8.4	$.04 \pm .01$	2.8 ± 0.6
Pres./Attr.	91.3 ± 1.4	78.6 ± 2.4	6.1	$.05 \pm .01$	4.2 ± 0.8
`Count`	85.7 ± 1.8	63.4 ± 3.1	4.8	$.07 \pm .02$	5.6 ± 1.1
`Spatial`	83.2 ± 2.0	58.9 ± 3.4	4.2	$.06 \pm .02$	5.9 ± 1.2

4.4 Deterministic Tool-Use Consequences

We construct 74 held-out deterministic workflows: 32 DocVQA form-filling chains, 18 ChartQA decision chains, and 24 ScienceQA lab-action chains, each spanning 5–12 tool calls. Every chain uses a frozen schema and executable success predicate. Earlier-session claims can influence later actions at $K \in \{5, 10\}$, directly testing error compounding across persistence. Each horizon evaluates the 74 workflow identities under five seeds and defines a distinct pool of 900 claim-conditioned tool

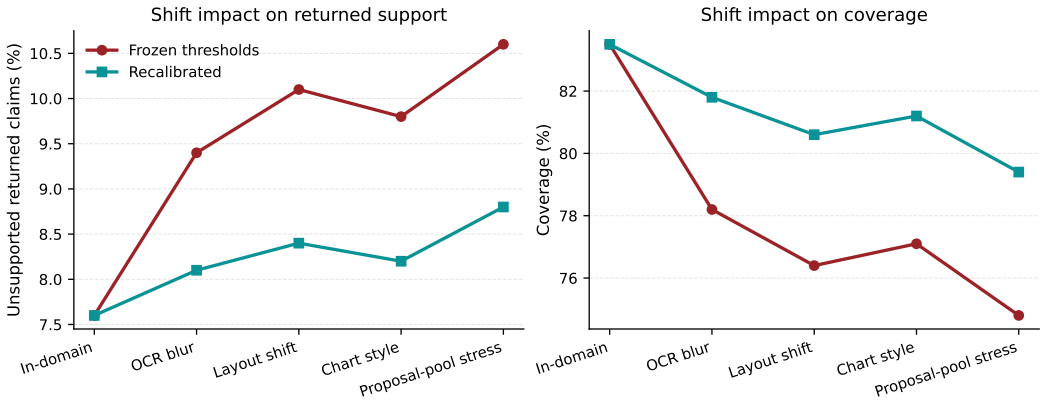


Figure 3: ReplayClaim under three distribution shifts and one proposal-pool stress. Frozen thresholds increase unsupported returns and reduce coverage; temperature recalibration with 200 labels per claim type recovers most of both changes without replacing the underlying models.

Table 4: Deterministic multi-session tool use over 74 chains. Percentages are five-seed means and standard deviations. The evidence-cache rows provide the same-evidence reference for mandatory replay in the downstream workflow; ReplayClaim improves success and unsupported-action rates at both horizons while preserving explicit abstention and re-grounding.

Method	Success	Action unsupported	Cov.	OH
Text-only (K=5)	68.4 ± 2.1	7.1 ± 1.0	96.8 ± 0.8	0%
Verifier-gated (K=5)	73.8 ± 1.8	4.6 ± 0.8	84.2 ± 1.9	10.4%
Evidence cache (K=5)	75.1 ± 1.7	3.8 ± 0.7	83.5 ± 1.9	11.9%
ReplayClaim (K=5)	76.2 ± 1.6	3.1 ± 0.6	82.8 ± 2.0	12.8%
Text-only (K=10)	62.8 ± 2.4	9.8 ± 1.4	96.2 ± 0.9	0%
Verifier-gated (K=10)	68.4 ± 2.0	6.4 ± 1.0	82.6 ± 2.1	11.2%
Evidence cache (K=10)	70.8 ± 1.9	5.2 ± 0.9	81.5 ± 2.1	13.4%
ReplayClaim (K=10)	72.6 ± 1.8	4.2 ± 0.8	80.4 ± 2.2	14.6%

actions; calls that do not consume persisted claims are excluded. Success percentages are means of the seed-level workflow aggregates, while paired win/loss/tie analysis uses all 370 workflow–seed outcomes. Table 4 shows higher end-to-end success and fewer unsupported actions at both horizons.

Relative to the evidence cache, ReplayClaim gains 1.1/1.8 success points (paired 95% CIs [0.3,1.9]/[0.7,2.9]; p=0.009/0.002), while unsupported actions fall from 34/900 to 28/900 and from 47/900 to 38/900. ReplayClaim attains these gains despite serving 0.7/1.1 fewer coverage points than the cache at K=5/10, so greater serving frequency does not explain the direction. Against verifier-gated memory, pairing each chain and seed yields 32 wins, 23 losses, and 315 ties at K=5, and 44/28/298 at K=10. The K=10 win/loss ledger corresponds to a 4.3-point paired difference; the separately rounded marginal means in Table 4 are 68.4% and 72.6%. The corresponding success-gain 95% CIs are [1.1,3.7] and [2.4,6.0]; unsupported actions fall from 41/900 to 28/900 and from 58/900 to 38/900. Success gains over verifier-gated memory at K=5/K=10 are 2.7/4.6 points for forms, 2.1/3.8 for chart decisions, and 2.4/4.1 for lab actions. The widening long-horizon advantage links returned-object quality to actionable persistence rather than a one-shot answer metric.

5 Discussion

The evaluation supports two claims at different temporal scales. The one-write/one-read benchmark shows that the complete returned object occupies a better support–coverage operating point than a persisted-score evidence cache (8.7% → 7.6%), and the independent source audit ties that ordering to truth error (8.2% → 6.4%). The changed-query workflows then show the downstream consequence

of mandatory execution: success gains widen with the reuse horizon while unsupported actions decline. Separate interventions establish that set resolution improves support at 85.8% coverage, recalibration recovers performance under shift, and lifecycle enforcement reduces stale-state attack success from 28.2% to 9.4%; they do not presume that one suboperator alone explains the full-object comparison.

The results motivate corresponding benchmark and API design choices. Persistent-memory benchmarks can preserve logged reuse opportunities, report coverage and support jointly, and score whether the returned object reconstructs the declared read decision. Memory APIs can expose evidence, calibration/model state, replay outcomes, conflicts, and revocation status as first-class fields. Precision-sensitive tool systems can then make quarantine and re-grounding explicit outcomes, preventing one unsupported write from silently multiplying across later actions.

The measured regime combines a shared parser and verifier family, fixed candidate retrieval, same-query serialization units, changed-query multi-session workflows, and attacks at the memory-interface boundary. Within it, `OCRFact` provides the strongest coverage while `Count` and `Spatial` choose more selective operating points; the cache comparison estimates the bundled returned-object intervention rather than a factorial effect of query routing, verifier execution, or version checks alone. Method-blind truth labels, parser cleanup, recalibrated shifts, two VLM stacks, and paired horizons preserve the ordering across these boundaries, while open-world retrieval and unconstrained tool policies remain separate evaluation targets.

6 Conclusion

A reusable multimodal claim needs more than evidence that can be displayed: it needs a returned object that can reconstruct the decision authorizing its next use. ReplayClaim supplies that object with a versioned witness, typed read gate, set resolution, and explicit repair. Its operating-point advantage survives independent truth adjudication, parser cleanup, and a second VLM, while changed-query workflows show larger gains at the longer reuse horizon. Claim-level read contracts therefore turn persistence from passive storage into a testable execution boundary.

Reproducibility Statement

Section 2 specifies the returned object, aggregate, typed query inputs, replay rule, deterministic resolver, and revocation transition; Section 3 defines write-to-reuse units, candidate sets, split isolation, models, labels, matching, and paired inference. Appendices A–C provide operating points, raw denominators, parser and contradiction audits, shift/attack conditions, and annotation examples.

AI Use Statement

Generative AI tools assisted manuscript preparation, including copyediting, formatting, and bibliographic verification. The existing research design, methods, measurements, analysis, and scientific claims were preserved during this preparation. The authors take responsibility for the final submission.

References

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob Menick, Sebastian Borgeaud, Andrew Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karen Simonyan. Flamingo: A visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, 2022.

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. Self-RAG: Learning to retrieve, generate, and critique through self-reflection. In *International Conference on Learning Representations*, 2024.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan,

Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025. URL https://arxiv.org/abs/2502.13923.

Yuanchen Bei, Tianxin Wei, Xuying Ning, Yanjun Zhao, Zhining Liu, Xiao Lin, Yada Zhu, Hendrik Hamann, Jingrui He, and Hanghang Tong. Mem-Gallery: Benchmarking multimodal long-term conversational memory for MLLM agents. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 40750–40784. Association for Computational Linguistics, 2026. doi: 10.18653/v1/2026.acl-long.1892. URL https://aclanthology.org/2026.acl-long.1892/.

Shiqi Chen, Yiran Zhao, Jinghan Zhang, I-Chun Chern, Siyang Gao, Pengfei Liu, and Junxian He. FELM: Benchmarking factuality evaluation of large language models. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, 2023. URL https://arxiv.org/abs/2310.00741.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, Bin Li, Ping Luo, Tong Lu, Yu Qiao, and Jifeng Dai. InternVL: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Enjun Du, Hange Zhou, Chenxu Du, Siyi Liu, Zirong Chen, Ziyu Zheng, and Yongqi Zhang. LEDGERMIND: Provenance-constrained multimodal agentic reasoning with a structured evidence ledger. *arXiv preprint arXiv:2607.28374*, 2026. URL https://arxiv.org/abs/2607.28374.

Junyu Feng, Binxiao Xu, Jiayi Chen, Mengyu Dai, Cenyang Wu, Haodong Li, Bohan Zeng, Yunliu Xie, Hao Liang, Ming Lu, and Wentao Zhang. M2A: Multimodal memory agent with dual-layer hybrid memory for long-term personalized interactions. *arXiv preprint arXiv:2602.07624*, 2026. URL https://arxiv.org/abs/2602.07624.

Tianyu Gao, Howard Yen, Jiatong Yu, and Danqi Chen. Enabling large language models to generate text with citations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Tianrui Guan, Fuxiao Liu, Xiyang Wu, Ruiqi Xian, Zongxia Li, Xiaoyu Liu, Xijun Wang, Lichang Chen, Furong Huang, Yaser Yacoob, Dinesh Manocha, and Tianyi Zhou. HallusionBench: An advanced diagnostic suite for entangled language hallucination and visual illusion in large vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 14375–14385, 2024.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. On calibration of modern neural networks. In *International Conference on Machine Learning*, 2017.

Minghao Guo, Qingyue Jiao, Zeru Shi, Yihao Quan, Boxuan Zhang, Danrui Li, Liwei Che, Wujiang Xu, Shilong Liu, Zirui Liu, Mubbasir Kapadia, Vladimir Pavlovic, Jiang Liu, Mengdi Wang, Yiyu Shi, Dimitris N. Metaxas, and Ruixiang Tang. MemEye: A visual-centric evaluation framework for multimodal agent memory. *arXiv preprint arXiv:2605.15128*, 2026. URL https://arxiv.org/abs/2605.15128.

Pengcheng He, Jianfeng Gao, and Weizhu Chen. DeBERTaV3: Improving DeBERTa using ELECTRA-style pre-training with gradient-disentangled embedding sharing. *arXiv preprint arXiv:2111.09543*, 2021a. URL https://arxiv.org/abs/2111.09543.

Pengcheng He, Xiaodong Liu, Jianfeng Gao, and Weizhu Chen. DeBERTa: Decoding-enhanced BERT with disentangled attention. In *International Conference on Learning Representations*, 2021b.

Zhengjun Huang, Wenxuan Liu, Zhoujin Tian, Wei Chen, Junle Chen, Yuqian Wu, Fangyuan Zhang, Qintian Guo, and Xiaofang Zhou. M^3Exam: Benchmarking multimodal memory for realistic user-agent interactions. *arXiv preprint arXiv:2606.07402*, 2026. URL https://arxiv.org/abs/2606.07402.

Alexander Kirillov, Eric Mintun, Nikhila Ravi, Hanzi Mao, Chloe Rolland, Laura Gustafson, Tete Xiao, Spencer Whitehead, Alexander C. Berg, Wan-Yen Lo, Piotr Dollár, and Ross Girshick. Segment anything. In *IEEE/CVF International Conference on Computer Vision*, 2023.

Chenxia Li, Weiwei Liu, Ruoyu Guo, Xiaoting Yin, Kaitao Jiang, Yongkun Du, Yuning Du, Lingfeng Zhu, Baohua Lai, Xiaoguang Hu, Dianhai Yu, and Yanjun Ma. PP-OCRv3: More attempts for the improvement of ultra lightweight OCR system. *arXiv preprint arXiv:2206.03001*, 2022a. URL https://arxiv.org/abs/2206.03001.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. BLIP: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *International Conference on Machine Learning*, 2022b.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. BLIP-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *International Conference on Machine Learning*, 2023a.

Yifan Li, Yifan Du, Kun Zhou, Jinpeng Wang, Wayne Xin Zhao, and Ji-Rong Wen. Evaluating object hallucination in large vision-language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023b.

Chengzhi Liu, Yuzhe Yang, Sophia Xiao Pu, Yepeng Liu, Lin Long, Yichen Guo, Nuo Chen, Zhaotian Weng, Elena Kochkina, Simerjot Kaur, Charese Smiley, Xiaomo Liu, James Zou, Sheng Liu, Yuheng Bu, Songyou Peng, and Xin Eric Wang. WorldMemArena: Evaluating multimodal agent memory through action-world interaction. *arXiv preprint arXiv:2605.29341*, 2026. URL https://arxiv.org/abs/2605.29341.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024a.

Shilong Liu, Zhaoyang Zeng, Tianhe Ren, Feng Li, Hao Zhang, Jie Yang, Qing Jiang, Chunyuan Li, Jianwei Yang, Hang Su, Jun Zhu, and Lei Zhang. Grounding DINO: Marrying DINO with grounded pre-training for open-set object detection. In *Computer Vision – ECCV 2024*, volume 15105 of *Lecture Notes in Computer Science*, pp. 38–55. Springer, 2024b. doi: 10.1007/978-3-031-72970-6_3.

Pan Lu, Swaroop Mishra, Tony Xia, Liang Qiu, Kai-Wei Chang, Song-Chun Zhu, Oyvind Tafjord, Peter Clark, and Ashwin Kalyan. Learn to explain: Multimodal reasoning via thought chains for science question answering. In *Advances in Neural Information Processing Systems*, 2022.

Potsawee Manakul, Adian Liusie, and Mark J. F. Gales. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq Joty, and Enamul Hoque. ChartQA: A benchmark for question answering about charts with visual and logical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2022*, 2022.

Minesh Mathew, Dimosthenis Karatzas, and C. V. Jawahar. DocVQA: A dataset for VQA on document images. In *IEEE Winter Conference on Applications of Computer Vision*, 2021.

Sewon Min, Kalpesh Krishna, Xinxi Lyu, Mike Lewis, Wen-tau Yih, Pang Wei Koh, Mohit Iyyer, Luke Zettlemoyer, and Hannaneh Hajishirzi. FActScore: Fine-grained atomic evaluation of factual precision in long form text generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

George C. Necula. Proof-carrying code. In *ACM SIGPLAN-SIGACT Symposium on Principles of Programming Languages*, 1997.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. MemGPT: Towards LLMs as operating systems. *arXiv preprint arXiv:2310.08560*, 2023. URL https://arxiv.org/abs/2310.08560.

Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior. In *ACM Symposium on User Interface Software and Technology*, 2023.

Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, Gretchen Krueger, and Ilya Sutskever. Learning transferable visual models from natural language supervision. In *International Conference on Machine Learning*, 2021.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. Toolformer: Language models can teach themselves to use tools. In *Advances in Neural Information Processing Systems*, 2023.

Zhoujin Tian, Hao Zhang, Yao Tian, Cheng Chen, Yakun Li, Lei Zhang, and Xiaofang Zhou. Beyond retrieval: Analytic memory for multimodal agents. *arXiv preprint arXiv:2607.29440*, 2026. URL https://arxiv.org/abs/2607.29440.

Yu Wang and Xi Chen. MIRIX: Multi-agent memory system for LLM-based agents. *arXiv preprint arXiv:2507.07957*, 2025. URL https://arxiv.org/abs/2507.07957.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. LongMemEval: Benchmarking chat assistants on long-term interactive memory. In *International Conference on Learning Representations*, 2025.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. ReAct: Synergizing reasoning and acting in language models. In *International Conference on Learning Representations*, 2023.

Qiming Zhu, Shunian Chen, Rui Yu, Zhehao Wu, and Benyou Wang. From lossy to verified: A provenance-aware tiered memory for agents. *arXiv preprint arXiv:2602.17913*, 2026. URL https://arxiv.org/abs/2602.17913.

A PROTOCOL, OPERATING POINTS, AND AUDITS

Worked read trace. Suppose the write pass extracts "the invoice shows $186.40" and stores OCR spans for `TOTAL`, `186.40`, and their coordinates, together with the `OCRFact` route, calibration state, and source/model versions. A later payment query changes the required binding from generic amount recognition to the invoice's total field. The evidence cache evaluates $\nVdash[s(c) \geq g_{\text{cache}}]$: the spans are inspectable, but they do not participate in that later binding. ReplayClaim reconstructs the field/value relation from the same spans under the payment query, applies the versioned temperature, checks provenance, and submits the admitted claim to the conflict graph. A version mismatch, missing field-role span, or higher-priority contradictory total therefore produces rejection or quarantine before the answer consumes the claim. The trace is the minimal two-query witness for why evidence retention and executable returned support are different object semantics.

Sampling ledger. The primary support audit contains 2,400 judgments per method, divided across five seeds as 240/120/120 DocVQA/ChartQA/ScienceQA items per seed. The semantic audit contains a separate 2,400-item sample per method, balanced across datasets and claim types; each item receives both the common support rubric and a method-blind source-truth label. The nearest-control intervals compare 2,400 seed-aligned source items served by both methods, while coverage and false rejection retain the full 16,028-opportunity-per-seed query pool. For tool workflows, success is paired over 370 workflow–seed outcomes per horizon, and unsupported-action counts use a distinct 900-action persisted-claim-consuming pool at each horizon.

Table 5: Operating points and verifier-stack diagnostic. (a) Every selected read gate lies inside the predeclared $83.5\% \pm 0.5\%$ coverage band; gates are selected on calibration data and reused without test-label optimization. The write-pack threshold τ_{w} is shared and fixed before these method-specific choices. (b) Reports the default-point verifier progression at g_{RC}=0.70.

(a) Fixed protocol and selected coverage-matched points

Method/group	Read gate/value	Realized Cov.	Selection rule
Text-only memory	0.54	83.1%	Nearest predeclared abstention gate
Region memory	0.58	83.6%	Nearest predeclared abstention gate
Citation generation	0.66	83.3%	Nearest citation-support gate
Post-hoc filter	0.65	83.8%	Nearest detector gate
Verifier-gated memory	0.68	83.4%	Nearest calibrated verifier gate
Score-only + consistency	0.69	83.2%	Nearest calibrated score gate
Score + evidence cache	0.67	83.7%	Nearest calibrated score gate
ReplayClaim	0.70	83.5%	Nearest replay-object gate
Evidence/reuse	k=2, $K \in \{5, 10\}$	–	Workflow horizons; W=K
Budgets	B_V=500 ms	–	$B_A \in \{1, 3, 5, 10, 20, 50\}$
Shift calibration	200/type	–	Temperature scaling only

(b) Default-point verifier-stack diagnostic

Variant	Unsupported (%)	Cov. (%)	OH p50/p95
Text-only	14.8 ± 1.6	97.2 ± 0.8	0/0%
Region, no verifier	12.4 ± 1.4	96.4 ± 0.9	1.2/3.8%
Single VQA verifier	9.8 ± 1.2	86.4 ± 1.6	3.1/8.7%
ReplayClaim ensemble	7.6 ± 1.0	83.5 ± 1.8	4.8/11.6%

Region proposals and OCR outputs are cached by media hash; verifier probes are keyed by claim type, crop/span, and model version; answer generations are not cached across evaluation queries. Coverage selection uses only the predeclared read-gate family and calibration data. Compute matching caps verifier wall time at 500 ms for every method.

Table 6: Contradiction and parser audits. (a) Edge metrics use contradiction-candidate claims as the denominator; system coverage is query-level. (b) The query-level resolver comparison holds coverage fixed in its third row. (c) The held-out parser audit reports source-item bootstrap 95% CIs and raw counts. Manual cleanup improves both nearest controls while preserving ReplayClaim's advantage.

(a) Claim-level contradiction edges

Type	Precision	Recall	False edge	Claim ΔCov.
`OCR` numeric	$.97 \pm .01$	$.82 \pm .04$	0.8%	-0.6
`OCR` text	$.94 \pm .02$	$.74 \pm .05$	1.4%	-0.8
Pres./Attr.	$.91 \pm .03$	$.68 \pm .06$	2.1%	-1.2
`Count`	$.88 \pm .04$	$.62 \pm .07$	2.8%	-1.6
`Spatial`	$.86 \pm .04$	$.58 \pm .08$	3.2%	-1.8
Overall	$.93 \pm .02$	$.72 \pm .04$	1.8%	-1.4

(b) Query-level system effect

Setting	Unsupported	Query Cov.	Accuracy
No contradiction	9.4 ± 1.1	85.8 ± 1.6	65.2 ± 1.1
With contradiction	7.6 ± 1.0	83.5 ± 1.8	66.1 ± 1.1
With contradiction, cov.-matched	7.8 ± 1.0	85.8 ± 1.6	65.8 ± 1.1

(c) Extraction/canonicalization audit

Quantity	Result	95% CI/count
Gold claims	1800	source statements
Parser outputs	1773	–
Matched claims	1669	–
Precision	94.1%	[93.0,95.1]
Recall	92.7%	[91.5,93.8]
Claim-type accuracy	94.6%	1579/1669; [93.5,95.6]
Number/unit normalization	96.8%	424/438; [94.9,98.1]
Coreference normalization	93.2%	288/309; [89.8,95.7]
False contradiction edges	2.2%	29/1324; [1.5,3.1]
Cleaned cache unsupported	8.3%	[7.3,9.4]
Cleaned ReplayClaim unsupported	7.2%	[6.2,8.2]

The parser error taxonomy separates missed and spurious claims, span boundaries, entity/coreference normalization, number/unit normalization, claim typing, verifier routing, and false contradiction edges. Blinded matching counts one parser output per gold claim and resolves duplicate outputs before computing precision and recall.

Latency policy. The default interface performs re-grounding synchronously after quarantine. ReplayClaim-SLA applies the same serve contract with a per-query cap on synchronous re-grounding; work beyond the cap remains quarantined and is repaired asynchronously. This variant changes tail scheduling, not admission semantics. At the selected k=2 operating point, Table 5 records 4.8%/11.6% p50/p95 overhead; the SLA variant bounds synchronous repair without changing which claims are eligible to serve.

B Shift, Cross-Stack, and Interface-Attack Details

The shift table separates calibration recovery from a backbone change. All rows preserve the source content and returned-object contract while changing the declared image transform, proposal pool, or base VLM.

Table 7: Distribution shift, proposal-pool stress, and cross-stack validation. (a) reports ReplayClaim with frozen and recalibrated thresholds. (b) applies identical shifts to the component-matched evidence cache. (c) changes only the base VLM to LLaVA-1.5-13B and reports five-seed variation plus unsupported/served counts over a common 10,840-opportunity pool.

(a) ReplayClaim with frozen threshold

Condition	Unsupported	Cov.	Prec.	ECE
In-domain	7.6 ± 1.0	83.5 ± 1.8	90.4 ± 1.4	.05
OCR blur	9.4 ± 1.3	78.2 ± 2.2	86.8 ± 1.8	.09
Layout shift	10.1 ± 1.4	76.4 ± 2.4	85.2 ± 2.0	.11
Chart style	9.8 ± 1.3	77.1 ± 2.3	86.1 ± 1.9	.10
Pool stress (n_{prop}=4)	10.6 ± 1.5	74.8 ± 2.6	84.4 ± 2.1	.12

(a) ReplayClaim with recalibrated threshold

Condition	Unsupported	Cov.	Prec.	ECE
In-domain	7.6 ± 1.0	83.5 ± 1.8	90.4 ± 1.4	.05
OCR blur	8.1 ± 1.1	81.8 ± 1.9	89.2 ± 1.5	.06
Layout shift	8.4 ± 1.2	80.6 ± 2.0	88.6 ± 1.6	.06
Chart style	8.2 ± 1.1	81.2 ± 1.9	89.0 ± 1.5	.06
Pool stress (n_{prop}=4)	8.8 ± 1.2	79.4 ± 2.1	87.8 ± 1.7	.07

(b) Score + evidence cache under the same conditions

Condition	Unsupported F/R	Coverage F/R	OH p95 F/R	ReplayClaim F/R
OCR blur	10.9/9.3	77.6/80.9	11.5/11.8	9.4/8.1
Layout shift	11.8/9.8	75.8/79.9	11.7/12.0	10.1/8.4
Chart style	11.4/9.6	76.5/80.6	11.6/11.9	9.8/8.2
Pool stress	12.2/10.1	73.9/78.5	11.9/12.2	10.6/8.8

(c) LLaVA-1.5-13B base VLM

Method	Unsp. (%)	Cov. (%)	Unsp./served	Acc. Δ (D/C/S)
Verifier gate	10.5 ± 1.3	83.4 ± 2.0	949/9041	–
Evidence cache	9.8 ± 1.2	83.6 ± 1.9	888/9062	reference
ReplayClaim	8.6 ± 1.1	83.3 ± 1.8	776/9029	+1.6/+1.2/+1.4

Across all four shifted conditions, ReplayClaim retains a lower unsupported-return rate than the evidence cache under both frozen and recalibrated settings. Changing the base VLM preserves the same ordering, with 776 unsupported claims among 9,029 served objects for ReplayClaim versus 888/9,062 for the cache.

Table 8: Direct attacks at the declared memory-interface boundary. ASR and FA denote attack success and false alarms. Each row names the protected state, allowed perturbation, trust assumption, comparator, and measured outcome; shift conditions are reported separately in Table 7.

State	Condition	Trusted input	Comparator	Observed outcome	Residual failure
OCR span	Span spoofing	Source pixels	Typed replay off	ASR $31.4 \pm 2.8 \rightarrow 8.6 \pm 1.4$; FA 2.1 ± 0.6	Wrong field relation
Region set	Region ambiguity	Source pixels	Typed replay off	ASR $24.8 \pm 3.1 \rightarrow 11.2 \pm 1.8$; FA 3.4 ± 0.8	Perceptual ambiguity
Verifier state	Score gaming	Fixed candidates	Verifier gate	ASR $18.6 \pm 2.4 \rightarrow 6.8 \pm 1.2$; FA 1.8 ± 0.5	Calibrated-score gaming
Provenance	Staleness	Version clock	Lifecycle repair off	ASR $28.2 \pm 2.6 \rightarrow 9.4 \pm 1.6$; FA 2.6 ± 0.7	Delayed invalidation

For the four direct attacks, p95 overhead is $+6.2\%$, $+8.4\%$, $+12.1\%$, and $+7.8\%$, respectively. The low false-alarm rates show that reduced attack success does not arise from indiscriminate quarantine. Region ambiguity remains the hardest condition because the candidate evidence itself is perceptually ambiguous. The adaptive verifier-gaming sweep preserves the same trust boundary while increasing the black-box query budget through $B_A \in \{1, 3, 5, 10, 20, 50\}$; ReplayClaim remains below text-only and verifier-based baselines throughout the sweep. The workflow distributions in Figure 4 connect fewer unsupported actions to faster recovery after a contradiction. The gap widens with persistence horizon rather than with additional serving coverage.

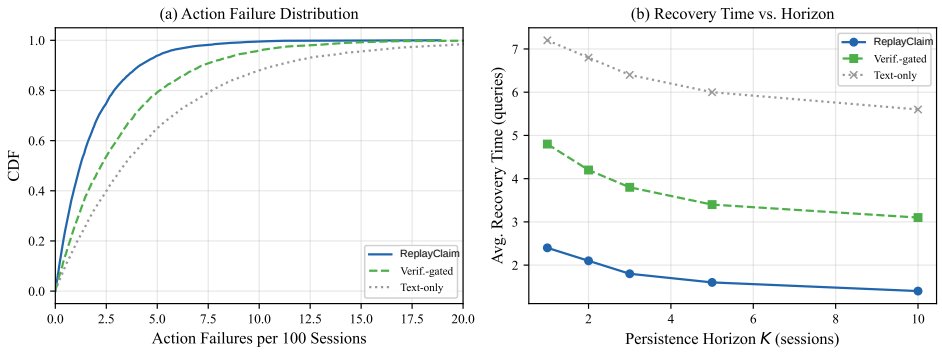


Figure 4: Deterministic workflow distributions. Panel (a) shows action failures per 100 sessions; panel (b) shows recovery time after contradiction-triggered quarantine and re-grounding. The distributional view explains the larger ReplayClaim advantage at K=10.

C Decision-State Positioning and Annotation Rubric

Table 9: Decision-state design map. The rows distinguish each family's primary stored object and read operation. ReplayClaim isolates whether persisted support reconstructs the later claim-level serve decision, complementing general memory and trajectory systems.

Family	Persisted state	Reuse gate	Set / lifecycle semantics
Citation-grounded generation	Answer with local citations	Retrieve cited text or regenerate a response	Task-specific filtering
Post-hoc filtering	Output with detector score	Threshold the persisted detector score	Reject or retain the current output
Verifier-gated memory	Claim, calibrated score, provenance	Threshold the stored score	Shared set-consistency control
Multimodal long-term memory	Text/visual histories, summaries, structured stores	Retrieve, update, or reason over past interactions	Conflict handling and memory evolution
Structured evidence runtimes	Provenance-linked tool outputs and dependencies	Ground reasoning steps in active ledger entries	Typed trajectory repair
Tool-state / trace memory	Actions, observations, outputs, execution state	Continue reasoning or execution	Controller handles invalid state
Long-memory evaluations	Histories, multimodal observations, evidence labels	Score stages from indexing through use	Update, abstention, stage diagnostics
Provenance-tiered memory	Summaries linked to raw evidence	Route or escalate to deeper evidence	Verified write-back
Score + evidence cache	Claim, score, crops or spans, provenance	Threshold with inspectable cached evidence	Shared set consistency and quarantine
ReplayClaim	Claim plus executable verifier/calibration witness	Replay the declared read gate	Set consistency, quarantine, re-grounding

One-shot factuality and hallucination benchmarks measure current-response errors (Li et al., 2023b; Guan et al., 2024; Chen et al., 2023; Min et al., 2023), while citation-grounded generation and Self-RAG improve answer-local attribution (Gao et al., 2023; Asai et al., 2024). Persistent agents manage summaries, messages, and execution traces across contexts (Packer et al., 2023; Park et al., 2023; Yao et al., 2023; Schick et al., 2023). MIRIX separates multimodal experience into specialized memory stores, M2A links raw and semantic personalized memory, and AdaMM adds queryable analytic structures (Wang & Chen, 2025; Feng et al., 2026; Tian et al., 2026). MemGallery, M^3Exam, WorldMemArena, and MemEye evaluate complementary aspects of visual-history retention, multimodal reasoning, interaction, conflict management, and evidence requirements (Bei et al., 2026; Huang et al., 2026; Liu et al., 2026; Guo et al., 2026). TierMem provides provenance-linked escalation from summaries to immutable raw evidence (Zhu et al., 2026); its read operation chooses an evidence tier rather than reconstructing a serialized claim's verifier and calibration decision. LedgerMind constrains complete multimodal trajectories to active evidence-ledger entries and typed repair (Du et al., 2026); ReplayClaim instead fixes the unit at one returned claim and pairs it with an evidence-identical cache. WorldMemArena and MemEye evaluate memory evolution and evidence use across interaction stages (Liu et al., 2026; Guo et al., 2026), whereas the present estimand holds candidate retrieval fixed and scores the object that authorizes service. These distinctions locate the contribution in claim-level read-contract reconstruction, not in a general storage, retrieval, or trajectory architecture.

Open-vocabulary grounding, OCR, calibration, and NLI provide the primitives used by the implementation (Liu et al., 2024b; Kirillov et al., 2023; Li et al., 2022a; Radford et al., 2021; Guo et al., 2017; He et al., 2021b). Broader multimodal systems improve answer-time representation and reasoning (Alayrac et al., 2022; Li et al., 2022b; 2023a; Dai et al., 2023; Liu et al., 2023); the persistence-gap audit begins when their answer-derived claim becomes reusable state.

Support rubric. Annotators inspect the complete returned tuple, including claim text, evidence crops or OCR spans, verifier state, provenance, replay outcome, conditioned answer/action, and query context. A claim is adequate only when the returned set supports the complete statement; partial evidence, wrong entity/field binding, ambiguous attribution, or missing relational context is inadequate. Annotators are instructed that confidence scores, verifier labels, and replay outcomes are metadata, not support: without evidence for the full statement, the label is inadequate. For "the invoice total is $186.40," the OCR spans must identify both the amount and its role as the total field. For "there are three red bars," the returned regions must cover all three bars; a partial crop showing two is inadequate. Baseline annotators inspect the object actually surfaced by that method rather than a reformatted ReplayClaim object.

Table 10: Raw returned-support audit. The table gives dataset-stratified unsupported/served labeled pairs for the four claim-bearing methods; Table 1 macro-averages dataset rates before averaging over seeds.

Unsupported/served support-audit pairs			
Method	DocVQA	ChartQA	ScienceQA
Verifier-gated	109/1200	62/600	56/600
Score-only + consistency	112/1200	59/600	51/600
Score + evidence cache	105/1200	54/600	50/600
ReplayClaim	86/1200	49/600	45/600

The raw ledger makes the dataset contribution to the macro average explicit: ReplayClaim records 86/1,200, 49/600, and 45/600 unsupported returns on DocVQA, ChartQA, and ScienceQA, respectively. Under the shared 500 ms verifier cap, budget-matched unsupported-return rates are $10.1 \pm 1.2\%$, $9.6 \pm 1.2\%$, $9.1 \pm 1.1\%$, and $8.2 \pm 1.1\%$ for verifier-gated, score-only + consistency, score + evidence cache, and ReplayClaim, respectively; these figures are separate from the coverage-matched comparison in Table 1.