Two Ways to Lose a Fact: When Partitioning a Compaction Carrier Helps, and When It Cannot

A capacity-matched ablation across twenty recent models, and a remedy that works for one failure mode and not the other

Aleksandr Tikhonov

July 28, 2026

Abstract

Long-running language agents compact their history into a running summary, folding it forward one chunk at a time. The fold is delta-only, so whatever it drops is unrecoverable. A widely used carrier is a single free-text field inside a schema, and a widely offered remedy is to divide that field into named ones. We test the remedy where it is supposed to work, holding declared capacity exactly fixed: one field permitted 9,600 characters against five permitted 1,920 each, with the system prompt, the input, the fold protocol, the temperature and the output budget held constant. Across twenty models published within six months, the undivided carrier loses planted identifier-shaped facts on eight — a real hazard, but a minority one. Partitioning rescues six of those eight, and which six is predictable: the failing models separate cleanly into two modes by how much of the declared ceiling their carrier filled. Six models filled 59-100% with description of the conversation rather than its contents, and partitioning rescued all six; two wrote a short gist instead, filling 6% and 36%, and partitioning rescued neither. Fill fraction does not predict whether a model fails ($r = -0.23$ over twenty-five measured models, a seven-model estimate of -0.89 having failed to replicate) but it separates which failure a model has, and therefore whether the remedy applies. Where the undivided carrier already works, partitioning costs 0.075 recall on average. The operative variable is whether the carrier arrives divided at all: a five-field schema guarantees it on 134 of 134 rows, a single field leaves it to the model, and recall tracks the division that arrived ($r = +0.61$ over 570 rows) rather than the one requested. We also report two negative results about instruments: a capability probe routed through an aggregating gateway measures the conjunction of model, endpoint and request, and our own request shape produced eight false "model cannot do structured output" verdicts; and a declared ceiling behaves as a target models write to and stop at (80 of 80 folds finished naturally), not as a truncation bound. We release the harness, the fixtures and sanitized aggregates.

1 Introduction

An agent that outlives its context window must compact. The common design folds the conversation forward: each fold receives the previous running summary plus a new chunk of history, and returns a replacement summary. The operation is delta-only. Nothing re-reads the original transcript, so a fact the fold declines to carry is gone for the remainder of the session, and no later turn can notice its absence.

That the operation loses facts is established. Zahn and Chana (2026) report summarization destroying 60% of stored facts and cascading compaction eroding 54% of project constraints, and locate the same behaviour across several frontier models; the pre-LLM ancestry runs back to iterative human retelling (Horta Ribeiro et al. 2019). That the shape of a model's requested output changes its behaviour is also established, for single-turn tasks (Tam et al. 2024) and, more precisely, as a capacity phenomenon: Fan (2026) shows that models with headroom pay no penalty for JSON while models near their limit pay a large one, and attributes roughly 87% of one model's penalty to token exhaustion.

What is not established is *which* property of a structured request is responsible. Fan (2026) is explicit about this, and names the gap as future work: "Our gradient varies prompt length, field count, and nesting depth simultaneously; future work should isolate each factor." A gradient that moves field count and nesting and prompt length together cannot say whether a schema hurts because it is a schema, because it is long, or because of how it divides the model's output.

This paper isolates one of those factors — the partition — inside an iterative memory fold. The comparison that carries the argument holds *declared capacity exactly constant*: one field permitted 9,600 characters against five fields permitted 1,920 characters each. The system prompt is byte-identical between the two arms (same recorded digest), as are the input, the fold protocol, the sampling temperature and the output-token budget.

One thing is not identical, and it does not run the way one would expect. A five-field schema serialises to more characters than a one-field schema, so the decomposed arm's *request* is larger. Its *prompt* is nonetheless **smaller**: the fold prompt is dominated by the summary carried in from the previous fold, and the decomposed arm carries far less of it. Measured, the decomposed arm sends **1,534 fewer** prompt tokens than the undivided one pooled over every sweep, and 2,319 and 3,089 fewer in the two sweeps where the partition contrast was run. An earlier version of this paper asserted the opposite and called it a small unavoidable advantage to the decomposed arm; it is neither the direction we claimed nor an advantage, and prompt size cannot explain the effect in either direction.

Our contributions:

1. A **capacity-matched partition ablation across twenty recent models**. At an identical declared ceiling, moving from one opaque field to five named ones is measured on every model where both arms run. Because total capacity is equal by construction, a capacity account is excluded by design rather than by argument.
2. **A scope for the remedy, including where it fails.** The undivided carrier loses facts on 8 of 20 models; partitioning rescues 6 of those 8, is useless on one of the other 2 and harmful on the other, and costs 0.075 recall on average where the carrier already worked.
3. **A mechanism that is structural rather than persuasive.** A five-field schema guarantees the carrier arrives divided (134 of 134 rows); a single field leaves it to the model, which complies on fewer than half of its rows. Recall tracks the division that arrived, not the one requested ($r = +0.61$ over 570 rows).
4. **Two failure modes, and the limit of the remedy.** Enforcing the partition rescues models that failed by filling an undivided slot with description; it does not rescue models that write 621 characters into a 9,600-character allowance, which receive the five sections and leave them empty.
5. **Two direct manipulations of the capacity channel**, which prior work leaves observational, plus evidence that a declared ceiling is a target rather than a bound.
6. **Evidence from the carrier itself.** We record the carried summary, not only its length, which lets us show that the fold is the entire failure surface and that the residual failure mode is coerced fabrication (Usman 2026) compounding upstream compaction loss.

We do not claim that structured output is harmful; two of our best arms are JSON. We do not claim to have discovered compaction loss, the capacity axis, or schema-coerced fabrication. We do not claim that partitioning is a general remedy — Section 4.7 is largely an account of when it is not. Section 6 states what is scoped to the models tested and what is not established.

2 Related work

Compaction destroys facts. Zahn and Chana (2026) is the closest prior result on the phenomenon and reports it at scale, proposing an external store as the remedy. An (2026) reaches a compatible conclusion from the retrieval side, reporting that verbatim chunks beat lossy artifact extraction by 15.9 points on one long-conversation benchmark (43.9% against 28.0%) and 22.0 points on another (67.4% against 45.4%), and attributing the loss to lossy distillation rather than to structure as such. Our remedy class is different from both: we stay inside the summary carrier and change only how the summariser is asked to divide its output. Our result is consistent with An's diagnosis and sharpens it — the distillation is lossy in a specific, fixable way, and the fix is not to abandon structure but to divide it.

Capacity. Fan (2026) is the work we position against most carefully. It establishes that the penalty for structured output scales with schema complexity, is not explained by prompt length alone, and largely disappears for models with headroom; it also shows a budget sweep in which truncation falls from 49% to 2% as accuracy rises from 52.5% to 84.0%. Two points of contact matter. First, its complexity axis runs *more fields and deeper nesting is worse*, while our partition axis runs *more named fields is better*. These are not in conflict — Fan measures reasoning accuracy under a structured answer, we measure verbatim survival through an iterated carrier — but the divergence shows the effect of schema shape is

task-dependent, and that for a memory carrier decomposition is protective. Second, Fan considered and rejected a verbosity account, observing that API JSON mode drops accuracy slightly while emitting *fewer* tokens than chain-of-thought. Any claim of the form "structure induces verbosity" must therefore be argued against a published negative result; we do not make that claim, and our capacity-matched design does not rest on it.

Serialisation cost. A line of work measures the token overhead of notation itself. Kutschka and Geiger (2026) benchmarks token-optimised alternatives to JSON across four agent benchmarks and five models, reporting reductions up to 27% and 18% against accuracy tradeoffs. That quantity is serialisation overhead — the cost of the syntax. Ours is content volume — how much material the model elects to write into the slot it is given. The two are independent, and conflating them would read as a rediscovery of the JSON tax.

Structured representations inside an iterative update. Hwang et al. (2024) is the closest prior work on our mechanism, and it reaches a conclusion that must be engaged rather than filed away. It updates a structured JSON knowledge representation in place as new sources arrive — rather than rebuilding it each time, which is our fold — and reports that doing so beats an unstructured-memory baseline by 40% and 14% across two datasets, with a further 7% and 4% from dynamically augmenting the representation. Comparing that to our own arms is harder than it looks: the contrast that would speak to it directly pairs our unheaded plain arm against our decomposed JSON arm, and those two differ in system prompt as well as format (Section 4.7), so we decline to read an ordering off it.

We do not think these conflict, and we do not claim to overturn it. Their comparison is structured-JSON against an undifferentiated memory, scored by entity and fact F1 over a document stream; ours varies how a summariser's output is partitioned, scored by verbatim identifier match in a conversation fold. On the axis both studies share — organised carrier against undifferentiated one — we agree, and the clean form of our agreement is the ladder in Section 4.7: at identical declared capacity, dividing the field raises recall by 0.700. The metrics also differ in a way that matters, since entity F1 credits a paraphrased entity and verbatim identifier match does not. The honest statement is that the two results are not directly comparable, and that neither supports "JSON is better" or "plain text is better" as a general rule.

The mechanism itself has standard references, and we use it rather than propose it. Wang et al. (2023) is the canonical statement of recursive summarisation for dialogue memory — previous memory plus new context yields new memory — which is the fold this paper measures; Packer et al. (2023) is the standard systems treatment of tiered virtual context around it. We contribute nothing to the mechanism and study only what its summariser is asked to emit.

Why a bespoke fixture rather than a public benchmark. Wu et al. (2024) and the LoCoMo corpus used by An (2026) are the obvious substrates, and we do not use them. The reason is the dependent variable: this study needs facts that are non-derivable and graded by exact string, planted at known positions so the fold that lost one can be identified, and those benchmarks grade by judge or by normalised overlap, which tolerates paraphrase. That is a real limitation on comparability and we state it rather than defend it.

Length regulation under iteration. Adams et al. (2023) establishes that summary length is controllable independently of information content, by explicit instruction. Chang et al. (2024) characterises running-summary folding directly, and is the canonical prior on the incremental-update regime we operate in. Our observation is adjacent and, as far as we can find, unstated: partitioning the output regulates its length implicitly, without any instruction about length at all.

Structure and fabrication. Usman (2026) demonstrates across thirteen models that required schema fields coerce fabrication, driving it to 100% in ten of them, and that an explicit "insufficient evidence" escape is largely ineffective for open-weight models. Our recall turn asks the model to list values, which is itself a form demanding an answer, so the fabrication we observe is best read as this phenomenon occurring *downstream of compaction loss* rather than as a new failure mode.

Kwon (2026) makes the closely related point for memory specifically, and states the danger more sharply than we would have: a memory that keeps a wrong conclusion but drops the work behind it leads a model to re-emit the stale value confidently, "where an empty memory leads it to abstain" — studied under a fixed token budget, as ours is. We take that as established and do not re-claim it. What our setup adds is that we record the carrier as well as the answer, so we can show the fact was absent from the carrier rather than inferring it, and can attribute the loss to the fold rather than to the answer turn.

One candidate mechanism for *which* values get invented comes from Parikh (2026), which finds that requesting JSON collapses answer diversity toward the modal response across 44 models — the modal answer's share rises from 41% to 64%, distinct answers fall from 52 to 36. That paper concerns valid answer spaces rather than missing information, so the connection is a conjecture rather than a result. But it would explain the specific character of what we observe: the substituted values are not arbitrary, they are the highest-frequency instance of their type — port 8080, `example.com`, the RFC 4122 example UUID.

Causal caution. Yuan et al. (2025) reports that apparent format effects largely vanish under causal analysis, finding no causal impact in 43 of 48 scenarios, and enumerates the collider structures that produce spurious format effects. Conditioning on a post-treatment variable — such as whether a summary hit its cap — is exactly the error it warns about. We avoid it by manipulating the capacity channel directly rather than conditioning on it, and we report the observational analysis separately and label it as such.

Structured memory. Contrary to a simple "plain text wins" reading, Sharma et al. (2026) finds JSON extraction reaching 0.96 feasibility against 0.48 for narrative hand-off, *despite* narrative producing the smallest payload — which by itself refutes "shorter is better". Petrov et al. (2026) reports strong results for schema-grounded memory. Our data agrees with this direction: named-field JSON is among our best arms, and the failure is specific to an undivided slot.

3 Method

3.1 The fold

A synthetic session of 150,000 tokens is divided into chunks and folded run by run. Each fold receives the previous running summary and one chunk, and returns a replacement. Nothing re-reads the transcript. After the final fold, a recall turn asks for fourteen planted facts by label, and the answer is graded against the recorded values.

Fifty facts are planted; **fourteen are graded**, and the two sets are not the same. The graded values are two ports, a filesystem path, a shard count, a UUID, three UTC timestamps, a cache origin URL, a DNS zone, a signing-key id, an on-call handle, a release tag, and one Cyrillic department name. Cluster ids, tenant codes, licence keys and image digests are planted but never asked, so a description of the planted set overstates what the score measures; this paper previously made that error.

None of the graded values can be reconstructed from another, from a range, or from a pattern. But "recall it or fail" is slightly too strong: four of them carry a unit word (`19 shards`, and the three timestamps), and an answer that keeps the number while dropping the unit fails the check while having carried the fact. The carrier cross-tab in Section 4.10 bounds how often that can matter. This is a deliberate choice of the hardest fact class and it scopes the claim; see Section 6.

3.2 Arms

Arms differ only in the representation the summariser is asked to produce. Prompt, input, fold protocol, and output budget are held fixed within a comparison.

Arm	Representation	Declared capacity
`f`	Deployed shape: shared schema, one `summary` string	1,024 chars
`n`	Same schema, ceiling lifted	9,600 chars
`j`	Dedicated schema, `summary` field only	9,600 chars, 1 field
`p`	As `j`, with an expanded field description	9,600 chars, 1 field
`q`	Decomposed schema, five named string fields	1,920 chars x 5 = 9,600
`k`	Semantic schema, several named fields	—

Arm	Representation	Declared capacity
`o`	Plain text, no headings	—
`m`	Plain text under five fixed headings	—
`t`	As `j`, declared ceiling halved	4,800 chars, 1 field
`u`	As `m`, carry cap reduced 2,400 -> 1,400 tokens	—
`l`	Plain text under the same five headings, at the deployed 900-token budget	—
`g`	The deployed shared schema, parsed rather than carried raw	1,024 chars

The `j` to `q` contrast is the paper's centre. `q` declares five fields whose ceilings sum to exactly the 9,600 characters `j` declares on its single field, so the pair moves the partition and not the capacity. This is enforced in code rather than asserted: the per-field ceiling is the single-field ceiling divided by the section count, and every row records the ceilings it actually declared.

3.3 Execution and validity

Arms are interleaved within each sweep on a rotating Latin square, so every arm leads equally often and execution position is balanced against endpoint drift. This balances position but **not** first-order carryover — each arm has the same single predecessor in most repeats — and a Williams design (Williams 1949) would be required for that. One sweep's rotation does not close, and is disclosed where used.

A single validity rule decides whether a row may enter any aggregate; every reported statistic reads that one rule. Rows invalidated by infrastructure faults may be retried; rows invalidated by an arm's own behaviour may not, since retrying those would select for the arm succeeding.

3.4 Models and provider pinning

Nine models carry the full replication sweep, selected for open weights where available, recency, small-to-medium size, and family diversity. Hosted models are reached through a gateway that routes across multiple endpoints per model. Left unpinned this is a **validity** defect and not merely a cost one: endpoints for one model differed in quantization (bf16, fp8, fp4) and in whether they honour strict schema requests at all, so an unpinned sweep can silently vary the treatment. Every sweep pins one endpoint with fallbacks disabled, and records the pin, its quantization, its context length and its price.

Three further models were attempted and produced no measurable rows. An earlier version of this paper described that as a schema-compliance failure. **It was not, and the error was ours**; the corrected account is in Section 4.2, because the mistake is instructive for anyone measuring capability through an aggregating gateway.

4 Results

> **Note**
>
> Recall is effectively binary at the row level: rows land at or near 0 or 1. Over all 620 valid rows of this scenario, a one-way random-effects decomposition of the fourteen binary checks gives **ICC = 0.750** and a design effect of **10.745**, so a fourteen-check row is worth about **1.30** independent observations rather than fourteen. Cells are therefore reported with their denominators wherever a table has room, and where a table gives means without them — the partition table, the iteration table and the failure-mode table — the per-cell row counts are in the released package rather than the page; the fourteen checks within a row are not independent trials (Miller 2024). Earlier versions of this paper quoted 0.851/12.06 and then 0.787/11.23, each from a smaller row set; the figures above are recomputed by `generate.py` from the released row- and check-level artifacts on every build, so they cannot drift from the data again.

4.1 The deployed shape fails on every model measured

The starting point was a deployed configuration: a shared schema carrying one `summary` string with a 1,024-character ceiling. It fails on all eight models on which it was measured, while plain text under fixed headings succeeds on all nine — but at 2,400 fold tokens against `f`'s 900, so that pair is not budget-matched. The arm that *is* matched to `f` at 900 tokens is `l`, which also succeeds on all nine (0.694-0.929); that is the comparison to read.

Table 1: Verbatim recall on the distributed-facts scenario, by model and arm. Cells are means over three to seven valid rows, with per-cell denominators in `s5-recall-by-model-arm.csv` of the released package; rows are effectively binary, so a mean here stands for a small count. The `qwen3.6-flash` row comes from the one sweep whose rotation does not close (2 arms over 7 repeats, so `l` leads four times and `m` three); every other sweep closes. The rotation is also a pure cycle, so first-order carryover is confounded with arm throughout.

Model	`f` deployed	`n` shared schema, ceiling lifted	`k` named-field JSON	`l` plain, 900 tok	`m` plain, 2,400 tok
Qwen3.6 (local reference)	0.051	0.061	0.969	0.725	0.969
qwen3.6-35b-a3b	0.000	0.051	0.878	0.735	0.990
qwen3.6-flash	—	—	—	0.786	0.990
tencent/hy3	0.000	0.857	0.816	0.694	0.786
z-ai/glm-5.2	0.086	0.000	0.800	0.929	1.000
minimax/minimax-m3	0.000	0.750	1.000	0.871	1.000
mistral-small-2603	0.257	0.786	1.000	0.829	0.986
deepseek-v4-flash	0.071	0.171	0.929	0.929	1.000
xiaomi/mimo-v2.5	0.000	0.000	0.857	0.857	1.000

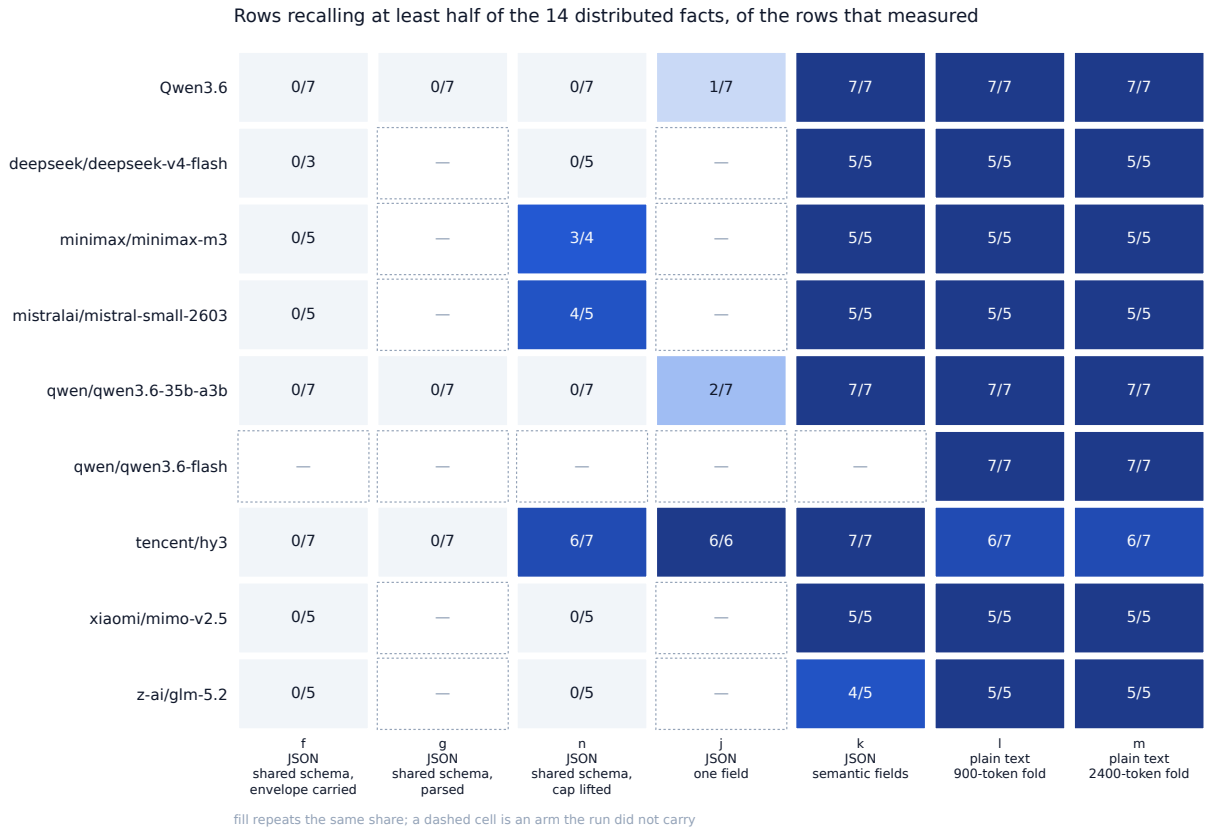

Figure 1: Rows recalling at least half of the fourteen distributed facts, by arm and model. A dashed cell is an arm the run did not carry.

Two cautions on reading Table 1. First, `f` differs from `m` in more than one respect — schema, ceiling and budget all change — so it does not isolate anything; its 1,024-character ceiling cannot hold the material regardless of shape, and it is reported as a floor result rather than as an ablation. Second, `n` is the arm that shows how model-dependent the schema penalty is, ranging from 0.000 to 0.857 across models. That variance is the reason the isolating comparisons below hold the model fixed.

Named-field JSON (`k`) scores 0.80 to 1.00 throughout, which is the first indication that the problem is not schemas. Paired within model, `k` nonetheless trails `m` by 0.066 with a bootstrap interval excluding zero, and an equivalence test at a margin of 0.10 in either direction does not pass; we therefore do not claim the two are equivalent, only that both are far above the undivided arms.

4.2 Capability failures we caused, and one we did not

Three models — `granite-4.1-8b`, `ling-2.6-flash` and `trinity-large-thinking` — produced no measurable rows. An earlier version of this paper reported that as evidence that they "advertise strict schema support and do not honour it". That was wrong, our own released verdicts said so in words, and the correction matters more than the original claim did.

The recorded verdicts are `probe_rejected_no_endpoint_for_parameters` for the first two and `probe_rejected_reasoning_not_disableable` for the third — routing and parameter-support failures, not schema-compliance failures. Neither of the first two was ever sent a schema: only the two plain-text arms were attempted on them, and those arms put no `response_format` on the request at all.

The cause was our own request. The gateway enforces `require_parameters` against **every** parameter on the wire, and this harness adds a reasoning-suppression parameter to every call that has nothing to do with what it measures. An endpoint that honours `response_format` but does not *declare* `reasoning` is refused with "no endpoints found that can handle the requested parameters" — a message that is indistinguishable, from outside, from the model lacking the capability. Re-probed with that parameter relaxed, **eight models return JSON conforming to a strict schema**: `granite-4.1-8b`, `ling-2.6-flash`, three Ministral sizes, two Gemma sizes and `llama-3.2-3b`. Only `gpt-oss-20b` still refuses, for the separate

reason below. The re-probe is a capability check rather than a measurement and produced no sweep rows, so it is reported here and is not in the released package.

`trinity-large-thinking` is a genuine incompatibility of a different kind: reasoning is mandatory on its endpoint and cannot be disabled, which the protocol requires. The same is true of the `gpt-oss` family at both 20B and 120B — a *family* property, not a size property.

The one real schema non-conformance in this study is **`qwen/qwen3.6-flash`**, recorded as `schema_advertised_reply_nonconforming`: it advertises structured outputs, returned JSON, and the JSON omitted a required property. Its sweep still ran and still measured, and it is included in Table 1 on the plain-text arms, which send no schema. It is named here because the earlier version of this paper reported three schema failures that were not, and stayed silent about the one that was.

The lesson is about instruments, not models. A capability probe routed through an aggregator measures the conjunction of the model, the endpoint, and the exact parameter set the harness happens to send. Reporting the conjunction as a property of the model is the error we made.

4.3 A clean ladder, and why we do not report a 2x2

An earlier version of this paper reported a format-by-partition 2x2 over `o`, `m`, `j` and `q`. That grid is not clean and we withdraw it. `o` is the only arm of the four that carries a **different system prompt** (`unstructured_prose`, digest `5ebf263a...`) from the other three (`fixed_headings`, digest `62a20557...`). Any contrast involving `o` therefore moves the instruction as well as the representation, and the two cannot be separated in that comparison.

Three arms do share one system prompt exactly, and they form a ladder (Table 2) in which each rung moves one thing:

Table 2: The three arms that share one system prompt exactly, each rung moving one thing.

rung	change	S5 recall	effect
`j`	one opaque field, ceiling 9,600	0.071	—
`q`	the same 9,600 split into five named fields	0.771	**+0.700**
`m`	the same five sections, schema removed	0.986	**+0.214**

The second rung deserves the same standard we apply elsewhere. `q` is a large improvement on `j` and it is **not** a return to `m`: it reaches full recall on **1 of 10** rows against `m`'s **8 of 10** (Fisher exact $\mathbf{p = 0.0055}$), a gap of 0.214. That is more than three times the `k-m` difference this paper declines to call equivalence, so we do not call `q` a recovery either. Enforcing a partition inside a schema recovers most of what an undivided field loses; removing the schema recovers the rest.

Both rungs are real and they are separable. Partitioning the declared capacity is worth about three times what removing the schema is worth, once the system prompt is held fixed. This is the comparison that survives; the `o` arm appears below only as an observation with its confound stated.

The `o` arm, for completeness and with its confound restated, scores 0.921 on the same sweeps. Read against `j` it would suggest that removing the schema is worth more than partitioning; read against `m` it would suggest partitioning in plain text is worth almost nothing. Neither reading is available, because `o` also changes the system prompt. We report the number and draw nothing from it.

A third arm adds a 1,835-character natural-language description to the single field and scores 0.016. It would be convenient to read that as "explaining the field harder does not help, so the problem is not comprehension" — and we did, in an earlier version. The harness's own precondition for that reading is a token count, and it fails: if the description reached the model on all five folds the arm's prompt should exceed the undivided arm's by roughly **2,294 tokens**, and the observed difference is **+298** — most of which is accounted for by this arm carrying a longer summary anyway. **We therefore withdraw the**

inference. The arm bounds nothing about comprehension, because we cannot show the description arrived.

On which measurement of `j` the ladder uses. The 0.071 above is the undivided arm as measured in the two decomposition sweeps (n = 10), which is where `q` was also measured; a contrast has to come from one design. The same arm on the same model over the larger anchor corpus scores **0.214** (n = 28), and over every model and sweep in this study its mean is higher still. The ladder's +0.700 is therefore a within-design effect, not a claim that 0.071 is the arm's characteristic value. Section Section 4.6 gives the range across models, which is wide and is the point.

4.4 Iteration is what makes it catastrophic

The same ablation run on a scenario whose facts all arrive in a single chunk, rather than distributed across folds, separates a mild effect from a severe one:

Arm	Facts in one chunk	Facts distributed across folds
`j` undivided JSON	0.300	**0.071**
`p` undivided JSON, described	0.300	**0.016**
`q` decomposed JSON	1.000	0.771
`o` plain, no headings	1.000	0.921
`m` plain, headings	1.000	0.986

This comparison moves more than iteration, and the difference is not small. The single-chunk scenario grades **7** of its 28 planted facts; the distributed one grades **14** of 50, and uses a different grader and a different set of planted values. So the contrast above is between two scenarios, not between two levels of one factor, and the "fourteen graded facts" stated throughout this paper describes the distributed scenario only. The grader change at least is empirically inert here: the label-bound and presence counts agree on **619 of 620** valid distributed rows.

Every healthy arm is perfect when the summariser sees the facts together and merely good when it must accumulate them; the undivided field is already impaired in the easy case and collapses in the hard one. The representation penalty is therefore not a property of summarisation as such but of summarisation *iterated*: each fold rewrites a description of a description, and an undivided slot gives the model no place to keep a fact that its own prose is drifting away from. This is what distinguishes the setting from single-turn format studies, where the model gets one chance to serialise material it can still see.

4.5 Capacity does not account for it

The capacity channel is manipulated directly rather than conditioned on.

Arm	Change	n	Rows recalled	Mean	Carried chars	Folds clipped at cap
`j`	ceiling 9,600	11	3	0.325	7,818	17
`t`	ceiling **4,800**	12	1	0.131	4,697	0
`m`	plain	4	4	1.000	6,269	0
`u`	carry cap **1,400**	4	4	1.000	5,505	13

Halving the declared ceiling halves the carried text and does not recover recall; the point estimate moves in the *opposite* direction to the capacity prediction (paired over the two models that ran both arms: 17/47 against 3/16, Fisher exact p = 0.23). The failing arm carries less text than either healthy arm. And assigning a reduced carry cap to a healthy representation — which clips it on 13 of 20 folds, the most clipping of any arm — leaves recall at 1.000, while `j`, clipped on 17 of about 53 folds, recalls 0.325. Pooled, the representation contrast is 4/23 against 8/8 (Fisher exact p = 6.3e-05).

The declared ceiling is not inert, however. It governs whether the model returns schema-valid output at all: `j` fails to produce a conforming fold on 4 of 60 folds (6.7%), silently degrading the fold, while `t` fails on none. Halving the ceiling fixed well-formedness and left fact retention untouched, which separates the two effects inside a single arm pair.

4.6 The ceiling result replicates; the collapse itself is model-dependent

Repeating the ceiling manipulation on a second model (`z-ai/glm-5.2`, four repeats per arm) gives a result worth reporting in full because half of it does not transfer.

Model	`j` ceiling 9,600	`t` ceiling 4,800	`j` fill of its ceiling
qwen3.6-35b-a3b	6/28 rows	1/12 rows	81%
z-ai/glm-5.2	**4/4 rows**	2/4 rows	**56%**

What replicates: halving the declared ceiling never improves recall. On both models `t` is at best equal to `j` and here again lands below it. The concern that our headline was an artifact of the particular ceiling we chose is answered on two models, in both significance and sign.

What does not: the collapse of the undivided arm is not universal. On `z-ai/glm-5.2` the undivided schema arm recalls perfectly.

A mechanism we proposed and then had to withdraw. On the first seven models we measured, the fraction of its declared ceiling a model chose to fill was strongly inversely associated with recall (r = -0.89), which would have explained the non-replications neatly. Extended to **twenty-five measured models it does not hold: r = -0.23**, indistinguishable from no relationship. We report the retraction rather than the seven-model version because the failure is informative.

What breaks it is that there are at least **two** ways to lose the facts, and they sit at opposite ends of the fill scale:

model	rows	carried chars	of a 9,600 ceiling	recall
`p`, described field	9	9,044	94%	0.016
qwen3.6-35b-a3b	28	8,501	89%	0.214
deepseek-v4-pro	4	4,740	49%	**1.000**
granite-4.1-8b	16	3,238	34%	0.250
ling-2.6-flash	16	621	**6%**	0.000

One failure writes prose until the slot is full and carries description instead of content. The other writes a short gist — `ling-2.6-flash` carries 621 characters on average summarising which maintenance steps passed, and not one identifier. Healthy carriers sit in between. Fill fraction cannot separate these because the relationship is not monotone, and any single-number summary of it is a small-sample artifact. What the two failures share is not length; it is that neither carrier is organised into places where an identifier belongs.

4.7 Partitioning across twenty models: not a general fix

The ladder above is one model. Extending the capacity-matched pair to every model where both arms could be run (**?@tbl-partition**) gives a much less comfortable picture, and it is the one that should be believed.

model	`j` undivided	`q` five fields	gap	`j` fill
ling-2.6-flash	0.000	0.161	+0.161	6%
qwen3.5-122b-a10b	0.000	0.714	**+0.714**	100%
qwen3.5-35b-a3b	0.000	0.625	**+0.625**	100%
mimo-v2.5	0.000	0.929	**+0.929**	99%
granite-4.1-8b	0.304	0.250	-0.054	36%
deepseek-v4-flash	0.161	0.947	**+0.786**	81%

model	j undivided	q five fields	gap	j fill
glm-5.2	0.487	0.518	+0.031	59%
qwen3.5-9b	0.482	0.661	+0.179	81%
qwen3.5-397b-a17b	0.625	0.821	+0.196	79%
minimax-m3	0.643	0.875	+0.232	93%
qwen3.6-27b	0.750	0.536	-0.214	68%
nemotron-3-ultra-550b-a55b	0.875	0.929	+0.054	60%
hy3	0.881	0.661	-0.220	62%
ling-2.6-1t	0.929	0.929	+0.000	61%
gemma-4-26b-a4b-it	0.947	0.857	-0.089	69%
mistral-small-2603	0.982	0.929	-0.053	71%
nemotron-3-super-120b-a12b	0.982	0.589	**-0.393**	69%
deepseek-v4-pro	1.000	1.000	+0.000	49%
gemma-4-31b-it	1.000	1.000	+0.000	63%
mimo-v2.5-pro	1.000	0.589	**-0.411**	58%

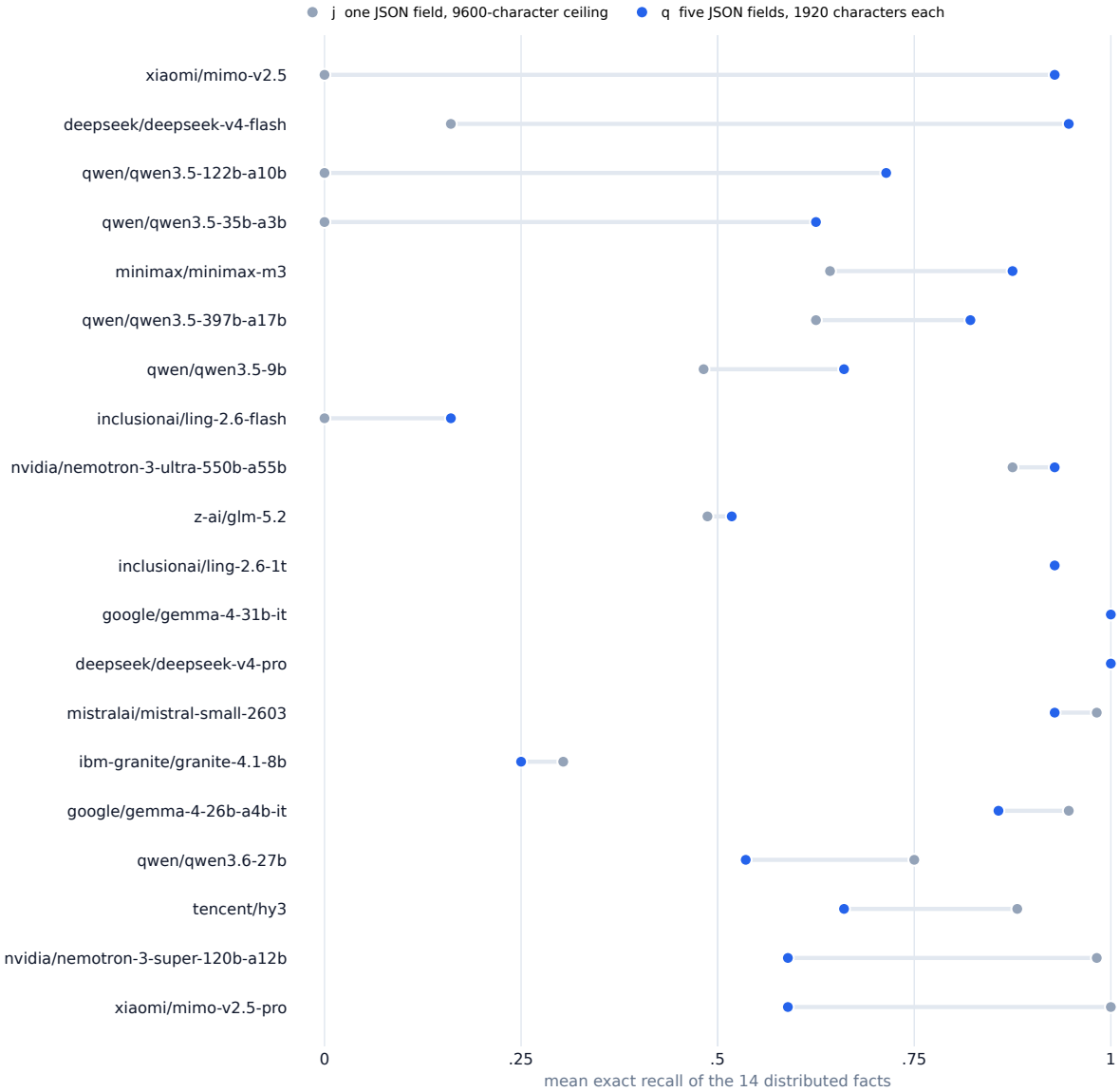

Figure 2: The same contrast drawn: each model's undivided arm against its partitioned one, at identical declared capacity.

: The capacity-matched partition contrast on every in-scope model where both arms ran, ordered by the undivided arm. `j` fill is the share of the declared 9,600-character ceiling the undivided carrier used. {#tbl-partition}

Three facts follow, and none is what the single-model ladder suggests.

The undivided slot fails on 8 of 20 models, not on all of them. It is a real hazard and a minority one.

Partitioning rescues 6 of those 8 — and which 6 is not arbitrary. Ordering the failing models by how much of the declared ceiling their undivided carrier filled separates the outcomes completely, with no overlap:

failure mode	fill of the 9,600-char ceiling	models	partitioning rescues
fills the slot with description	59%, 81%, 81%, 99%, 100%, 100%	6	**6 of 6**
writes a short gist instead	6%, 36%	2	**0 of 2**

Where the undivided slot already works, partitioning costs 0.075 recall on average across the twelve such models, and 0.411 on `mimo-v2.5-pro`. It is not a free intervention.

So fill fraction is not a predictor of *whether* a model loses facts — we showed above that it is not, at r = -0.23 — but it is a clean diagnostic of *which* failure a model has, and that determines whether the standard remedy applies. Dividing a carrier into five named destinations helps a model that was going to write 9,000 characters of prose regardless. It cannot help a model that was never going to write more than 900, because the problem there is not where the content goes but that it was never produced.

Each contrast is paired within a sweep. A model's `j` and `q` means come from the same run, never pooled across runs. This matters and we got it wrong at first: `z-ai/glm-5.2` scores 1.000 on the undivided arm in one sweep and 0.393 in another, so pooling its `j` rows across sweeps moved it out of the failing group entirely. The within-sweep pairing is the only one a contrast can use, and the between-sweep spread is itself a reminder that these rows are close to all-or-nothing.

The borderline cases, and the separation with and without them. Two models sit within 0.02 of the threshold on the undivided arm and cannot be confidently classed either way. `glm-5.2`, re-run at twelve rows per arm, gives `j` 0.487 and `q` 0.518 — a gap of +0.031, with 5 of 11 and 6 of 12 rows above half. `qwen3.5-9b` gives `j` 0.482. Calling either a failure or a success is an artifact of where the line is drawn.

The separation does not depend on that call:

classification	rescued fills	non-rescued fills	margin
all models	59, 81, 81, 99, 100, 100	6, 36	23 pts
excluding both borderline models	81, 99, 100, 100	6, 36	**45 pts**

Dropping the ambiguous models *widens* the gap rather than closing it, which is the direction a real effect should move in and the opposite of what a threshold artifact would do.

How strong is a complete separation of eight models? Not as strong as the word "complete" sounds. The split is post-hoc, on a continuous variable, at a threshold chosen after seeing the outcomes. Granting the direction, drawing exactly the two lowest-fill models as the non-rescued pair has probability 1 in C(8,2), about 3.6%, under random assignment — so the separation carries roughly the weight of p = 0.04 at n = 8, not the weight the absence of overlap suggests.

The negative half was re-run at three times the depth, and it holds. Because two models is a thin basis for a claim, both were repeated at twelve rows per arm rather than four:

model	arm	n	mean	rows >= 0.5	fill	headings carried
granite-4.1-8b	`j`	12	0.304	6	36%	0 on 10 rows, 5 on 2
granite-4.1-8b	`q`	12	**0.250**	**0**	**20%**	5 on all 12
ling-2.6-flash	`j`	12	0.000	0	6%	0 on 11 rows, 1 on 1
ling-2.6-flash	`q`	12	0.161	2	13%	5 on all 12

Neither is rescued at depth, and `granite-4.1-8b` produces the sharpest result in the paper against the remedy: **enforcing the partition made it worse** (0.250 against 0.304, and 0 of 12 rows above half against 6 of 12), while also making it write *less* — 20% of the allowance under five enforced fields against 36% under one. Dividing the carrier five ways did not distribute this model's effort; it fragmented it.

Its undivided arm is also the clearest within-model case for Section 4.8: the 14 rows that carried no headings average 0.194, the 2 that carried five average 0.643. The model does better when it happens to partition, and worse when the partition is imposed — which is a real tension and we do not resolve it.

The honest scope of the remedy, then: partitioning a compaction carrier is effective for the over-writing failure, useless for the under-writing one, and on one model actively harmful. Two models is still two models; what the depth run buys is that their non-rescue is no longer a four-row accident, and one of the six positives remains marginal.

4.8 The partition that arrives, not the one that is requested

Every arm in this study receives the **same** system prompt asking for the same five named sections — the recorded digest is identical across `j`, `q`, `m`, `p`, `t` and `u`. What differs is whether the carrier that comes back is actually divided. The harness records this per row as `carried_fixed_headings`, precisely so that "the prompt asked for five headings" is never mistaken for "five headings arrived", and it is the strongest signal in the study.

Within the single-field-schema arms, recall is monotone in how many of the five sections the carrier actually contained:

headings that arrived	rows	mean recall	rows >= 0.5
0	38	0.086	5
1	58	0.041	2
2	5	0.457	2
3	5	0.529	3
4	11	0.610	7
5	69	**0.900**	64

Over all 570 valid rows where the count was recorded the correlation is $\mathbf{r = +0.61}$, against -0.23 for the fill fraction we withdrew above. Across *all* arms the relation is not monotone at the bottom, and the exception is instructive: the unheaded plain arm carries zero sections by construction and does well, because it is not a schema at all. Zero sections is harmless without a schema. One section inside a schema is where the facts go missing.

This is what the partition manipulation actually does. A five-field schema does not merely suggest division — it *guarantees* it, because the carry is the concatenation of the named fields under their headings: `q` carries five sections on **134 of 134** rows, and `k` on 39 of 39. A single field leaves division to the model, and models differ sharply in whether they comply: of `j`'s 161 rows, 37 carry no section at all, 40 carry one, and 69 carry all five.

That reframes the cross-model result of Section 4.7 in a way nothing else did. The models whose undivided arm already works are the ones that partition it unprompted — `deepseek-v4-pro` and `gemma-4-31b-it` both carry five sections on 4 of 4 rows and recall 1.000. The models that partitioning rescues are the ones that do not: `deepseek-v4-flash` carries one section per row and recalls 0.161, and the same model given five enforced fields carries five and recalls 0.947.

And it explains the two it does not rescue, by failing to. `ling-2.6-flash` and `granite-4.1-8b` receive all five headings under `q` — on every row — and still recall 0.161 and 0.250. Enforcing the partition worked; it just did not help, because these are the models that write 621 and 3,238 characters into a 9,600-character allowance. A carrier divided into five labelled places is no use to a model that will not write enough to fill one.

So the mechanism has two parts and the second is not optional: **the carrier has to arrive partitioned, and it has to arrive populated.** The schema controls the first and nothing in this study controls the second.

Stated as an association, deliberately. Headings carried is a post-treatment variable, and conditioning on it causally would be exactly the error Yuan et al. (2025) warns of. What is assigned is the arm; what is structural is that the five-field arm forces compliance on every row. The causal claim runs arm → guaranteed partition → recall, and the table above is the observational evidence that the middle term is the operative one.

4.9 What the carrier actually contains

Under one opaque field the model writes *about* the session — one carrier states that "the assistant has recorded 50 configuration decisions so far" — counting the facts without carrying any. Under five named headings the same model writes the decisions themselves, as an enumerated list of literal identifiers. Halving the ceiling shortens the description without making it factual.

Models treat a declared ceiling as a target rather than a bound. This is worth establishing rather than asserting, because if the ceiling were a guillotine — if the halved arm were simply being cut off — it would manipulate something different from what a capacity account is about.

It is not a guillotine. Across every fold in the halved-ceiling arm, **80 of 80 calls finished with `stop`**: none hit the output bound and none was truncated, while the carried text landed within 0.5% of the declared ceiling on 14 of 16 rows. The model wrote to the offered ceiling and stopped there of its own accord. The undivided arm at the full ceiling behaves the same way on the large majority of its calls — 694 `stop` against 20 that hit the length bound, the latter being the schema-invalid folds reported above.

So an undivided slot is an invitation to produce prose to length, the invitation is accepted voluntarily, and prose is the wrong carrier for an identifier.

4.10 The carrier bounds recall, but reading it is lossy too

Because the carried summary is recorded, not merely measured, we can ask of every row how many graded values are present in the carrier and how many the answer then gets right. Over **215 valid rows** the two agree on 171. An earlier version of this paper reported exact agreement on every row; that held on the 32 rows then available and does not hold at this scale, and the shape of the disagreement is more useful than the agreement was.

All 44 disagreements run the same way. In every one, the carrier held the fact and the answer turn failed to produce it. There is **not a single row** in which the answer was right about a fact the carrier did not contain.

arm	rows	carrier had it, answer missed it
`q` five named fields	88	**31 (35.2%)**
`j` one opaque field	103	13 (12.6%)
`m` plain, five headings	4	0
`t` halved ceiling	16	0
`u` reduced carry cap	4	0

Two things follow. First, **the carrier is a strict upper bound on recall**: nothing is recovered after the fold, so every effect reported above is an effect on what the fold wrote down, and the presence-vs-exact diagnostic that cannot separate the two losses is retired either way.

Second, and this qualifies the remedy, **retrieval from the carrier is itself lossy and most lossy in the partitioned arm**. `q` gets the facts into the carrier far more often than `j` does — that is the whole effect of Section 4.8 — and then loses about a third of them again at the answer turn. Some of that is arithmetic: an arm whose carrier holds more facts has more to lose. But it means the measured benefit of partitioning is a *net* of a large gain in the fold and a real loss in the read, and a design that improved the read would show partitioning in a better light than we do.

4.11 Sometimes the loss is silent, and which arm decides that

An earlier version of this paper asserted that a fold-dropped fact never returns as an omission. That is not what the answers show, and the correction matters because the original claim was the paper's most quotable and its least supported.

Classifying every failing row (recall < 0.5) by whether its answer contains any abstention marker at all:

arm	fabricates with no abstention	abstains somewhere
`j` one opaque field	23	64
`q` five named fields	4	48
`n` shared schema	12	20
`t` halved ceiling	8	5
`f` the deployed shape	**1**	**44**

So silent fabrication is real but it is a minority behaviour overall, and it is concentrated in the undivided-schema arms. The deployed configuration this study began with does the opposite: in 44 of its 45 failing rows the model says it does not know. That is a loud failure, and a system that logs abstention would have caught it.

Where fabrication does occur it has the character Usman (2026) describes, arising here downstream of compaction loss rather than from absent input. The substituted values are drawn from the high-frequency instance of their type: in one failing row, port `8443` for `62114`, `/var/lib/migration/state.json` for `/opt/kestrel/state/install.lock`, `02:00 UTC` for `01:26 UTC`, `16 shards` for `19 shards`, a canonical example UUID for the planted one, and an `example.com` origin for an internal host. Round numbers, stock ports and documentation placeholders form a greppable signature, which is the practical value of the observation.

We cannot say that row fabricated on *all fourteen* items: only the first 400 characters of each answer were retained, which reaches item nine. The counts above are likewise computed on that prefix, so they classify what the answer began by doing, not necessarily what it did throughout.

5 Discussion

The practical rule is not "avoid structured output" — but our evidence for that is narrower than it first looks, and the narrowing is worth stating. **Only one arm in this study carries JSON forward, and it is the deployed one.** `carried_json_envelope` is false on 575 of the 620 valid rows; the 45 that carry a raw envelope are all `f`, the shipped configuration — which fails on every model measured. Every arm we call healthy is a JSON *writing* format whose fields are rendered back to headed text before the next fold reads them. So we show that a JSON **request** is safe when it forces a partitioned carrier. We do not show that a JSON **carrier** is safe: the only carrier of that kind we ran is the one this study began by replacing, and one arm is no basis for a claim either way. A paper recommending prose over schemas would still contradict Sharma et al. (2026), but it would not be contradicted by us.

The same distinction disposes of what looks like a counterexample in our own table. The shared-schema arm declares *three* fields to the undivided arm's one, and does no better — on some models worse. It is more decomposed in the schema and not in the carrier: two of its three fields (`preserved_paths`, `preserved_tool_names`) are discarded rather than carried, so what reaches the next fold is again a single

free-text slot, and its headings-carried distribution looks like the undivided arm's rather than the five-field arm's. Schema field count is not the axis. What the carrier is divided into is. The rule is that **a memory carrier should not contain an undivided free-text slot**. Whatever a slot is named, a model asked for a "summary" of a fact list will tend to produce a description of the list; asked for constraints, decisions and identifiers under separate headings, it produces them.

That partitioning also *shortens* the carrier is the counter-intuitive part, and it runs opposite to the serialisation-overhead literature, where structure costs tokens. The two quantities differ: notation overhead is a property of the syntax, while what we measure is how much material the model elects to write. Decomposition appears to act as an implicit length regulator in an iterated rewrite loop, doing without instruction what Adams et al. (2023) achieves by instructing.

6 Limitations

Scope of the claim. Results are scoped to the models tested, one fold protocol, and one scenario family. The partition contrast ran on twenty models published within six months of the study; the supporting arms (`f`, `n`, `k`, `l`, `m`) ran on a smaller and partly older set, and Table 1 should be read as that set rather than as this one.

The negative half rests on two models. Six of eight rescues and zero of two non-rescues is a complete separation, but two is a small denominator for the claim that partitioning cannot help an under-writing carrier. We regard the mechanism as interpretable and the sample as thin, and we would not be surprised by a third mode.

Selection. Models were required to have a serving endpoint advertising structured outputs at 131,072 tokens or more, which the fold prompt needs. That requirement excludes most small models on the aggregator we used, so the small end of our range is not a random sample of small models.

Effect size is model-dependent. We do not claim a universal magnitude for the undivided-carrier penalty; it ranges from total (0.000) to absent (1.000) across the models measured.

Fact class. The planted facts are high-entropy, non-derivable identifiers, chosen as the hardest case. We do not establish that the effect holds for facts a model could paraphrase or reconstruct, and the claim should be read as scoped to verbatim retention of non-derivable tokens.

Scenario weight. One scenario carries the decisive comparisons. Its five planted segments land in three folds rather than five, so it exercises accumulation less than its design intends.

Non-random exclusion. An earlier version of this section claimed that rows carrying an arm's own parse failures are dropped, so schema-arm recall is conditioned on the schema having worked. That was wrong in both directions and we correct it. Parse failures **stay in the means by design** — the validity rule says so in terms, holding that a schema the model cannot satisfy at its budget is the arm's own behaviour and one of the things these arms exist to measure. Empirically, of the 23 single-field-schema rows carrying a parse failure, **18 are valid** and enter the aggregates, and they average **0.560** recall against **0.399** for the 175 clean rows of the same arms — so they are not dragging that mean down either. Over all arms the counts are 63 rows carrying a parse failure, 57 valid, averaging 0.476 against 0.553 for the 563 clean rows; the two populations differ because the arms do, which is why the schema-arm figures are the ones the claim needs.

The exclusion that *is* arm-correlated is a different one, and we disclose it here. A distributed-facts row is invalidated when the fold's catch-up branch fires, because its chunks were then folded together and it answers a different question. That happened on **6 rows, every one of them a schema arm** (`j` four times, `p` once, `q` once) and never on a plain arm. Their recalls were 1.000, 1.000, 0.000, 0.000, 0.000 and 0.000 — two removed from an arm that was working and four from arms that were failing — so the direction is not systematically flattering, but the correlation with arm is real and a reader should have it. Parse-failure rate is reported beside recall regardless.

Sampling and dispersion. Sampling is stochastic and no seed was sent, so runs are not bit-reproducible; this is a reproducibility limitation we disclose rather than one we corrected retroactively (Biderman et al. 2024).

Order control. The rotating square balances execution position but not first-order carryover, and one sweep's rotation does not close.

Token accounting. Prompt sizing used a character-ratio estimator for gateway sweeps and an exact tokenizer for the local reference, so absolute token figures are not comparable across those two. The estimator is also content-dependent in both directions: measured against a reference tokenizer it *under*counts identifier-dense text and *over*counts prose. We report the measured per-class ratios rather than a single multiplier, and note that this asymmetry is itself why a character-denominated ceiling is easier to fill with prose than with identifiers.

Grading. The scenario's fourteen graded checks bind a value to its label and require it to appear as a whole token. A weaker substring grader is used for the presence diagnostic on other scenarios; it was given the same boundary rule for the release, but figures collected before that change were not re-graded, because full answer text was not retained. No graded value is a substring of another, which bounds the effect.

Observational analyses. Where we report associations between carry-budget consumption and recall, these are observational and conditioning on them would be conditioning on a post-treatment variable (Yuan et al. 2025). The causal claims in this paper rest on the manipulations, not on those associations.

7 Data availability

What is distributed with this paper is the sanitized evidence package: aggregate and row-level CSV files, the evidence manifest, the study configuration, a provenance lock pinning the retained private inputs by hash, and `generate.py`, which validates every artifact and recomputes the figures, the tables and four inferential quantities from them — the headings correlation, the intraclass correlation and design effect, the withdrawn fill correlation, and the paired ceiling contrast. The remaining tests quoted in the text are computed here and not in the package, and are marked as such. The benchmark harness and the scenario fixtures are **not** in this release. Account state, credentials, private endpoints and local paths are excluded from the release.

8 Conclusion

Inside an iterative memory fold, how a summariser is asked to divide its output governs what survives it, and does so independently of how much it is allowed to write. One opaque field loses what five named fields keep, at identical declared capacity and with 45% less text carried. The damage is confined to a single configuration — an undivided free-text slot inside a schema — and it is not reliably announced: the undivided-schema arms supply a confident wrong value in a substantial minority of failing rows, while the deployed shape mostly abstains instead. Whether the loss is visible depends on the arm, which is why logging abstention catches some of these failures and none of the others.

Adams, Griffin, Alexander Fabbri, Faisal Ladhak, Eric Lehman, and Noémie Elhadad. 2023. *From Sparse to Dense: GPT-4 Summarization with Chain of Density Prompting*. https://arxiv.org/abs/2309.04269.

An, Tao. 2026. *Fidelity Before Structure: Verbatim Chunks Beat Lossy Artifact Extraction in Long-Conversation LLM Memory*. https://arxiv.org/abs/2601.00821.

Biderman, Stella, Hailey Schoelkopf, Lintang Sutawika, et al. 2024. *Lessons from the Trenches on Reproducible Evaluation of Language Models*. https://arxiv.org/abs/2405.14782.

Chang, Yapei, Kyle Lo, Tanya Goyal, and Mohit Iyyer. 2024. "BooookScore: A Systematic Exploration of Book-Length Summarization in the Era of LLMs." *The Twelfth International Conference on Learning Representations*. https://arxiv.org/abs/2310.00785.

Fan, Hengxin. 2026. *Capacity, Not Format: Rethinking Structured Reasoning Failures*. https://arxiv.org/abs/2606.09410.

Horta Ribeiro, Manoel, Kristina Gligorić, and Robert West. 2019. *Message Distortion in Information Cascades*. https://arxiv.org/abs/1902.09197.

Hwang, EunJeong, Yichao Zhou, James Bradley Wendt, et al. 2024. *Enhancing Incremental Summarization with Structured Representations*. https://arxiv.org/abs/2407.15021.

Kutschka, Lorenz, and Bernhard Geiger. 2026. *Notation Matters: A Benchmark Study of Token-Optimized Formats in Agentic AI Systems*. https://arxiv.org/abs/2605.29676.

Kwon, Alex. 2026. *Reclaim Evaluation: A Lossy Memory Is Worse Than an Empty One*. https://arxiv.org/abs/2606.25449.

Miller, Evan. 2024. *Adding Error Bars to Evals: A Statistical Approach to Language Model Evaluations*. https://arxiv.org/abs/2411.00640.

Packer, Charles, Sarah Wooders, Kevin Lin, et al. 2023. *MemGPT: Towards LLMs as Operating Systems*. https://arxiv.org/abs/2310.08560.

Parikh, Tapan. 2026. *Structured Output Collapses Answer Diversity Across 44 Language Models*. https://arxiv.org/abs/2607.18476.

Petrov, Alex, Alexander Gusak, Denis Mukha, and Dima Korolev. 2026. *From Unstructured Recall to Schema-Grounded Memory: Reliable AI Memory via Iterative, Schema-Aware Extraction*. https://arxiv.org/abs/2604.27906.

Sharma, Anantha, Sheeba Elizabeth John, Kaarthik Senthil Kumar, and Saratsuhas Vijayababu. 2026. *State Compression in Two-Agent LLM Relays: A Closed-World Study of Constraint Preservation*. https://arxiv.org/abs/2607.18265.

Tam, Zhi Rui, Cheng-Kuang Wu, Yi-Lin Tsai, Chieh-Yen Lin, Hung-yi Lee, and Yun-Nung Chen. 2024. "Let Me Speak Freely? A Study on the Impact of Format Restrictions on Performance of Large Language Models." *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, 1218–36. https://doi.org/10.18653/v1/2024.emnlp-industry.91.

Usman, Rana Muhammad. 2026. *PhantomFill: When the Form Demands an Answer, Language Models Invent One*. https://arxiv.org/abs/2607.20492.

Wang, Qingyue, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. 2023. *Recursively Summarizing Enables Long-Term Dialogue Memory in Large Language Models*. https://arxiv.org/abs/2308.15022.

Williams, E. J. 1949. "Experimental Designs Balanced for the Estimation of Residual Effects of Treatments." *Australian Journal of Scientific Research A* 2 (2): 149–68. https://doi.org/10.1071/CH9490149.

Wu, Di, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024. *LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory*. https://arxiv.org/abs/2410.10813.

Yuan, Han, Yue Zhao, Li Zhang, Wuqiong Luo, and Zheng Ma. 2025. *Quantifying the Impact of Structured Output Format on Large Language Models Through Causal Inference*. https://arxiv.org/abs/2509.21791.

Zahn, Oliver, and Simran Chana. 2026. *Facts as First Class Objects: Knowledge Objects for Persistent LLM Memory*. https://arxiv.org/abs/2603.17781.