VISTA: Storage and Read-Time Factorization for One-State Temporal Coexistence

Anonymous ACL submission

Abstract

Most model-editing evaluations only check whether the latest write survives a stream of updates. We study a stricter diagnostic, *one-state temporal coexistence*: one deployed model state must answer four temporal queries about the same fact—its `current` value, its immediately `previous` value, its value at a queried earlier revision (`as_of_revision`), and `[UNKNOWN_TIME]` when the queried time falls outside the visible append-only edit log (`unsupported_time`). VISTA addresses this with three LoRA adapters on a frozen backbone—*hot* for recent writes, *snapshot* for the current state, *archive* for older supported values—and a read path that first gates on visible-stream validity and only then scores fact-local historical candidates. On a cleaned WikiBigEdit version-selection benchmark, VISTA improves `current` and `previous` retrieval and keeps `unsupported_time` abstention at 1.0; an ablation isolates the read path—validity gating alone fixes `unsupported_time` but not `previous`, while the full path does both. VISTA leads completed 3B EasyEdit baselines on four official-source streams (Visa Bulletin, IRS, FDA, HHS), and matched-read WISE/UltraEdit controls—given the same gate and candidate set—narrow but do not close the gap. The shift transfers to Llama-3.2/3.1. Native `as_of_revision` remains a stress probe; the positive claim is closed-answer `current`/`previous` selection plus `unsupported_time` abstention.

1 Introduction

Most post-deployment editing evaluations ask only whether the latest write survives a stream of updates (Hartvigsen et al., 2023; Yu et al., 2023; Wang et al., 2024a; Gu et al., 2025; Wang et al., 2025; Luo et al., 2026). That criterion misses a common deployment requirement: the same edited state may have to answer four temporal questions about one fact—its `current` value, its immediately `previous` value, its value at a queried earlier revision (`as_of_revision`), and an explicit `[UNKNOWN_TIME]` response when the queried time falls outside the append-only edit log the system has seen. We call this *one-state temporal coexistence*. The difficulty is not just storing the newest fact, but preserving multiple supported versions in one final state and knowing when no supported version applies. This matters wherever a wrong historical answer is consequential rather than merely stale: regulatory, medical, and legal questions often turn on the rule that held at a specific time, and our official-source streams therefore draw on IRS tax rules, State Department visa cutoffs, FDA drug-label histories, and HHS poverty thresholds.

Existing editors are not built for this target. They mainly preserve the latest write, while temporal coexistence requires multiversion access plus explicit unknown-time handling. Recent temporal knowledge editing benchmarks ask related historical-retention questions, but typically evaluate direct temporal fact edits or structured temporal QA rather than one final edited state under a fixed inference-time read path. On our cleaned 8-file WikiBigEdit version-selection benchmark (Thede et al., 2025), even replay breaks down once evaluation moves beyond the newest value.

VISTA combines one storage choice with one read-path choice. Storage keeps three LoRA roles separate during training—*hot* for recent writes, *snapshot* for the current state, and *archive* for older supported values—and the read path first asks whether the queried time is in the visible edit stream, then scores supported historical candidates from that fact's revision chain. The read contract is inference-time logic over visible-edit-log metadata; it adds no learned parameters. To separate it from storage layout, we run *native-read* comparisons (each baseline as released) and *matched-read* controls (WISE and UltraEdit given the same validity

gate and fact-local candidate set), so any residual gap reflects only how the model itself stores edits. The contribution is the deployed memory contract together with an empirical diagnosis of where nearby editors fail, not a new optimization rule.

The evidence is strongest for current/previous version selection and unsupported-time validity. On the main batch-0224 surface, VISTA reaches 0.594 `current`, 0.659 `previous`, and 1.0 `unsupported_time` on Qwen2.5-3B (overall 0.721); matched-read WISE and UltraEdit close part of the gap but stay at 0.468 and 0.465 overall, and the same shift from a non-factorized read to VISTA appears on Llama-3.2-3B and Llama-3.1-8B. Native `as_of_revision` stays a stress probe: only eight such examples sit in batch 0224, so broad as-of retrieval is outside the claim. Our contributions are threefold. First, we define one-state temporal coexistence as a diagnostic for current/previous version selection plus unsupported-time abstention in one deployed state. Second, we present VISTA, a temporal-memory system that combines three separated storage roles with an inference-time read path over visible-edit-log metadata. Third, we introduce matched-read WISE and UltraEdit controls, plus cross-family, continuation, and transfer checks, that isolate read-contract gains from storage-layout gains.

2 Problem Setting

We study a frozen causal language model augmented with internal parametric memory modules. After deployment, the system receives append-only factual updates. At inference time, it does not retrieve from a text corpus or query an external knowledge base, although the read path may use structured metadata derived from the visible update stream. Our claim concerns this deployed memory system under an explicit temporal read contract, not open-book retrieval.

The diagnostic contains four query families: `current` (latest supported value), `previous` (immediately preceding supported value), `as_of_revision` (value at a queried earlier revision), and `unsupported_time` (the queried time lies outside the supported state and should map to `[UNKNOWN_TIME]`). If a subject-relation chain moves from A at r_1 to B at r_2 to C at r_3, the expected answers are C, B, B, and `[UNKNOWN_TIME]`. The main evaluation treats `current` and `previous` as supported retrieval, `unsupported_time` as validity selection, and `as_of_revision` as a harder stress probe.

This target is stricter than standard lifelong editing. A method can pass latest-value retention yet still fail the temporal task if it erases older values, merges neighboring revisions, or cannot tell a supported historical query from an unsupported one. The important distinction is between value *content* and value *validity*.

Test-time information contract. The global closed-answer scorer gives every method the same candidate pool and prompt. VISTA additionally uses two read-time structures built only from the visible append-only stream: $V(x)$, the supported revision times for the queried subject-relation chain, and $C(x, t)$, the fact-local values from that chain. $V(x)$ makes unsupported-time detection an explicit metadata decision; it is not evidence of learned historical retrieval by itself. $C(x, t)$ narrows supported historical scoring from the global pool to values in the relevant chain, but the model still has to rank the correct supported value. Because these structures are explicit metadata derived from the visible stream, we report native-read comparisons and matched-read WISE/UltraEdit controls that receive the same V and C; the matched-read rows are the fairness check that asks whether the gain survives when the read contract is held fixed across methods.

Our primary testbed is the deep 8-file version-selection benchmark over a cleaned WikiBigEdit stream. Evaluation uses closed-answer questions derived from visible revision chains, scored against one shared candidate set that contains all visible values, predecessor-to-latest transitions, and an explicit `[UNKNOWN_TIME]` option when `unsupported_time` queries are present. Thus `unsupported_time` measures selection of a dedicated unknown-time target, not free-form refusal. A prediction is correct only if it matches the target candidate exactly; chance at batch 0224 is about $1/258$. The slice contains 128 `current`, 88 `previous`, 88 `unsupported_time`, and only 8 `as_of_revision` queries. That imbalance matters: the benchmark can show whether a method supports current/previous coexistence and validity selection, but not whether durable as-of retention is solved at scale. The main evaluation therefore focuses on current/previous selection and dedicated unknown-time handling.

3 Method Overview

Storage roles. VISTA treats temporal memory as a storage-layout problem rather than only an edit-retention problem. As shown in Figure 1, the model keeps three separate write paths on a frozen Qwen2.5 backbone (Qwen Team, 2024): a plastic `hot` adapter (`level_0`) that absorbs recent streaming writes, a compacted `snapshot` adapter (`level_1_snapshot`) that holds the current supported state, and an `archive` adapter (`level_1_archive`) that retains older supported values from the visible revision chain. The decisive choice is that compaction builds *separate* snapshot and archive payloads instead of forcing both current and historical supervision through one shared representation; this avoids the latest-write-overwrites-history collapse we observe in the non-decoupled ablation rows.

Compaction schedule. Streaming writes accumulate in the hot adapter for k batches; a compaction step then distills the accumulated edits into the snapshot adapter (current values) and the archive adapter (predecessor values from the visible revision chain), and the hot adapter is reset. Crucially, VISTA inherits the same compaction schedule, optimization budget, and LoRA target modules used by the decoupled ablation row, so the only difference between those two rows is the read contract introduced below. Concrete hyperparameters are listed with the experimental setup and in the appendix.

Read-path semantics. VISTA's inference contract is built from two read-time structures over the visible append-only stream: $V(x)$, the supported revision times for the queried subject-relation chain, and $C(x,t)$, the fact-local supported values from that chain. $V(x)$ converts unsupported-time detection into an explicit metadata decision; on its own it is not evidence of learned historical retrieval. $C(x,t)$ narrows supported historical scoring from the global candidate pool to values in the relevant chain, so the model still has to rank the correct supported value rather than recall it from a flat lookup. Both structures are derived from the visible edit log alone and add no learned parameters, which is why the matched-read WISE and UltraEdit controls can be given exactly the same V and C.

Inference rule. Algorithm 1 combines the storage roles and the two read-path structures: `current` queries use the deployed hot-then-snapshot fallback, while historical queries first check $V(x)$ and only then score $C(x,t)$ on the archive adapter. Separating unsupported-time validity from supported version selection inside one deployed state is the single architectural difference between the decoupled ablation row and the VISTA row.

Algorithm 1 VISTA inference

Require: Query (x,t), hot adapter H, snapshot adapter S, archive adapter A, visible-stream validity index V, fact-local candidate set builder C

```
if x is a current query then
    score with the hot-then-snapshot fallback used by the deployed decoupled state
    return current-query prediction
else
    if t ∉ V(x) then
        return [UNKNOWN_TIME]
    end if
    build fact-local historical candidate set C(x, t)
    score C(x, t) with archive adapter A
    return highest-scoring supported historical candidate
end if
```

4 Experiments

4.1 Experimental Setup

Benchmark families. We use two benchmark families. The main mechanism diagnosis is a cleaned 8-file version-selection slice derived from WikiBigEdit (Thede et al., 2025). The late matched point is batch 0224 with 128 `current`, 88 `previous`, 88 `unsupported_time`, and 8 `as_of_revision` examples. We also analyze a 35-batch continuation ending at batch 0273, whose correlated prefixes contribute 119 native `as_of_revision` examples. WikiBigEdit serves as a diagnostic coexistence surface, not evidence that durable historical retention is solved at scale. The transfer suite contains four official-source streams built from the Visa Bulletin archive, IRS inflation-adjustment releases, DailyMed SPL histories, and the ASPE poverty-guidelines archive (U.S. Department of State, 2026; Internal Revenue Service, 2026; U.S. National Library of Medicine, 2026; Office of the Assistant

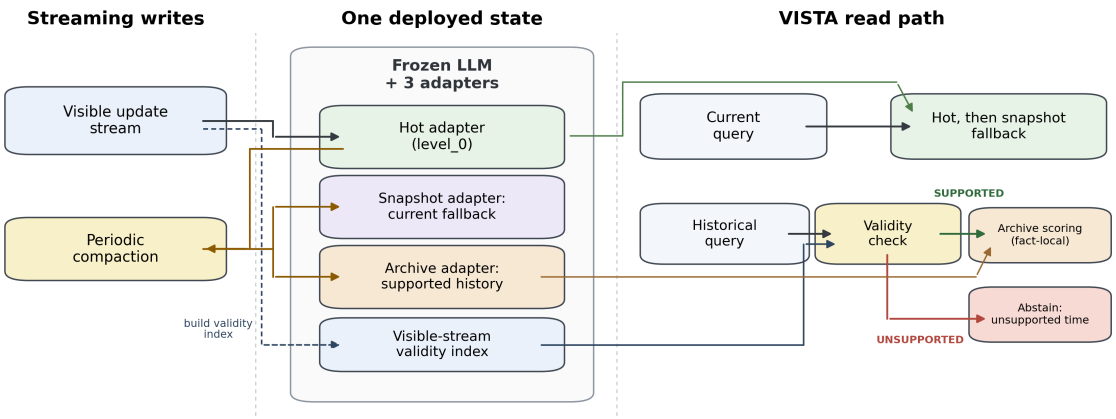

Figure 1: VISTA method overview. Streaming updates write through a plastic `hot` adapter and are periodically compacted into separate `snapshot` and `archive` adapters; `current` queries use hot-then-snapshot fallback, while historical queries first pass through a visible-stream validity gate (returning `[UNKNOWN_TIME]` on unsupported times) and then score only fact-local archive candidates.

Secretary for Planning and Evaluation, U.S. Department of Health and Human Services, 2026).

Compared systems. The internal ablation ladder has five rows on a fixed Qwen2.5-3B backbone (Qwen Team, 2024): replay, split, indexed, decoupled, and VISTA. We add one matched Qwen2.5-7B VISTA rerun and 7B EasyEdit implementations of WISE and UltraEdit. We focus matched external comparisons on this lifelong-editor family because it targets sequential update streams; locate-then-edit methods such as ROME, MEMIT, and AlphaEdit (Meng et al., 2022a,b; Fang et al., 2025) address one-shot or batch edits, and a metadata-only multiversion store tests a different upper bound. For cross-family evidence, Llama-3.2-3B (Meta Llama Team, 2024) contributes a paired decoupled-versus-VISTA comparison on the same late slice, and Llama-3.1-8B (Grattafiori et al., 2024) contributes one VISTA row. All main-text rows come from finished runs; smaller 3B external reruns are in Appendix Figure 4.

Shared training protocol. The four internal branches share a cleaned 8-file stream, seed 7, bf16 Qwen2.5 backbone, LoRA target modules `gate_proj`/`up_proj`/`down_proj`, 28 streaming batches of size 8, compaction every 2 batches, up to 64 update steps per batch, and 8 training epochs. VISTA reuses the decoupled training and compaction schedule and only adds two live read keys (visible-stream validity gating and fact-local historical candidates), so the branch comparison varies storage layout and read-time factorization rather than the optimization schedule.

External baseline protocol. WISE and UltraEdit use the official EasyEdit implementations (Zhang et al., 2024) with repo-local hparams. The matched-scale main-benchmark control uses `Qwen/Qwen2.5-7B-Instruct` (Qwen Team, 2024). All methods share the same prompt surface and closed-answer scorer. Because VISTA changes the read contract, Table 3 reports both native-read EasyEdit rows and scorer-held-constant WISE/UltraEdit rows that receive the same validity gate and fact-local restriction. The native rows test each system as implemented; the matched-read rows test whether read-contract alignment alone explains the gap.

Metrics and reporting. We report per-task accuracy from final-summary artifacts; predictions are chosen by maximum conditional mean log-probability over the candidate set. We treat `current` and `previous` as the primary supported-retrieval metrics and `unsupported_time` as a separate validity metric, because a visible-stream gate can solve unsupported-time selection without retrieving a historical value. Overall accuracy is reported only as a surface summary. Batch 0224 has only 8 native `as_of_revision` examples, so `as_of_revision` is a stress probe rather than headline evidence. We compute conditional

bootstrap intervals with 2,000 seed-7 resamples; Appendix Table 9 collects the main confidence ranges together with the 35-batch native-as-of continuation and broader as-of-form checks. Late endpoints have 257–258 candidates, so chance is about 0.0039. For official-source streams we report stream-average supported macro and overall mean over the full run; the small IRS/FDA suites are transfer checks, not scale claims.

4.2 Main End-to-End Result and Architectural Ablations

Table 1 gives the main result: VISTA is the first row in the 3B ladder that improves supported retrieval while also selecting the unsupported-time target. Replay collapses; split recovers supported retrieval but misses the unknown-time target; indexed flips that trade-off; decoupled recovers retrieval again; and VISTA combines previous retrieval with explicit validity. At batch 0224, VISTA reaches 76/128 `current`, 58/88 `previous`, and 88/88 `unsupported_time`. The 3/8 native `as_of_revision` cell is shown for completeness but is not used as headline evidence.

Because batch 0224 contains only 8 native `as_of_revision` examples, the headline result excludes `as_of_revision`. The bootstrap ranges in Appendix Table 9 preserve the current/previous ordering, while the native `as_of_revision` interval remains wide. On this surface, the frozen model plus the VISTA read path improves previous-value retrieval and unknown-time handling at the same time.

The read-path progression separates metadata validity from value retrieval. On the same 3B batch-0224 endpoint, a global resolver reaches overall 0.3365 with `previous` at 0.3636. Adding only the visible-stream validity gate raises overall to 0.6186 by fixing `unsupported_time`, but `previous` remains at 0.3636. The full VISTA read path then raises `previous` to 0.6591 and overall to 0.7212. Unsupported-time accuracy is not the main retrieval result; the key retrieval gain is the previous column after fact-local historical scoring is added.

4.3 Broader 3B Robustness Controls

The main effect is not unique to batch 0224. We run three 3B checks targeted at that concern. First, a saved-checkpoint observable historical readout covers all 18 prefixes from batch 0088 to 0224 that contain historical queries; saved historical accuracy lies between 0.5000 and 0.8864 with mean 0.8257, and `unsupported_time` stays at 1.0 throughout. Second, a 35-batch continuation enlarges the native `as_of_revision` surface from 8 examples at batch 0224 to 119 correlated prefix-level decisions, yielding aggregate native `as_of_revision` accuracy $62/119 = 0.5210$. Third, the completed 28-prefix all-endpoint candidate-control sweep evaluates the live factorized read path across 757 as-of-form examples; Appendix Table 10 reports 0.8158 under fact-local candidates and 0.7830 under same-relation hard negatives on as-of-form examples, with overall means 0.5244 and 0.5034. These checks keep `as_of_revision` visible as a stress signal, but they do not make broad as-of retrieval a solved result.

4.4 Cross-Family Check

A similar transition appears on Llama. Table 2 shows that Llama-3.2-3B with the decoupled live read stops at overall 0.3301 with `unsupported_time`= 0.0, while the matched Llama-3.2-3B VISTA row reaches overall 0.6891 with `unsupported_time`= 1.0: `current` stays flat, `previous` rises from 0.4091 to 0.6705, and unknown-time selection flips from failure to success. Llama-3.1-8B with VISTA lands at overall 0.7244 with `unsupported_time`= 1.0. These rows suggest the same mechanism boundary without replacing the Qwen-centered ablation and fairness analysis.

4.5 Matched-Scale Baseline Check

Table 3 shows that the 7B VISTA rerun remains above WISE and UltraEdit. The matched-read controls confirm that read-contract alignment narrows the gap but does not close it: both baselines improve, yet both remain well below VISTA.

4.6 Official-Source External Validity

We construct four closed-answer temporal streams from official public sources: Visa Bulletin employment cutoffs, IRS tax-year rules, FDA label revisions, and HHS poverty guidelines. These are practical temporal-risk settings rather than synthetic analogues: a deployed system may need the current rule, a previous rule, a time-anchored diagnostic answer for audit, or the explicit unknown-time option when no supported period exists. The base four-domain suite used for cross-domain comparison contains 59/15/19/72 records for Visa /

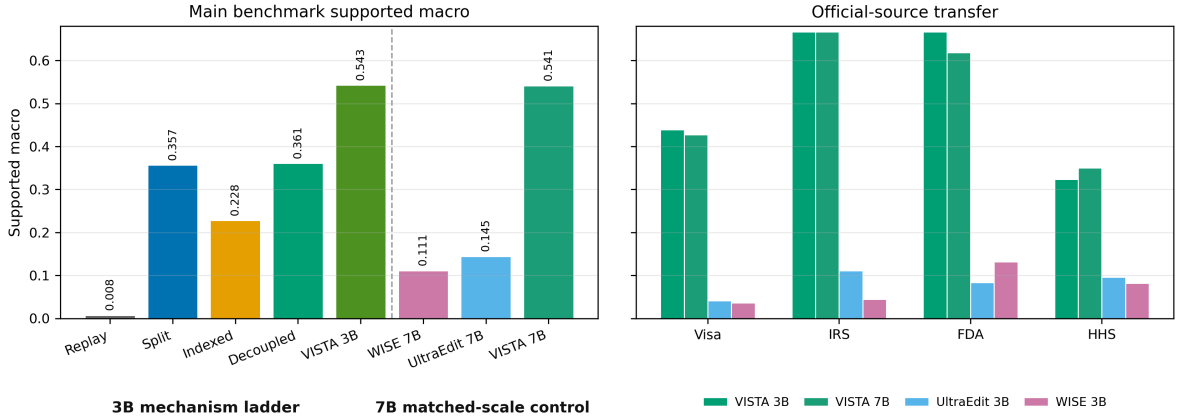

Figure 2: Supported macro on the main benchmark and the official-source transfer streams. Left: the 3B mechanism ladder on WikiBigEdit batch 0224 plus the matched 7B block. Right: stream-average supported macro on the completed official-source runs.

Table 1: Final-slice batch-0224 accuracy for the Qwen2.5-3B architectural ablations plus VISTA. The retrieval claim rests on current and previous; unsupported is a separate validity target, and as-of is shown only as an 8-example stress probe.

Method	Current ↑	Previous ↑	As-of ↑	Unsupported ↑	Overall ↑
Replay control (3B)	0.000	0.023	0.000	0.000	0.006
Split internal (3B)	0.594	0.352	0.125	0.000	0.346
Indexed internal (3B)	0.594	0.091	0.000	0.000	0.269
Decoupled internal (3B)	0.570	0.386	0.125	0.000	0.346
VISTA live read (3B)	0.594	0.659	0.375	1.000	0.721

IRS / FDA / HHS (12/5/8/12 chains; mean depth 4.9/3.0/2.4/6.0), and the 7B replications reuse the same streams. Because IRS and FDA are small, this block tests transfer of the read contract under limited construction sizes rather than serving as an application-scale benchmark.

Figure 2 and Table 4 show the transfer results. The VISTA rows lead the completed 3B EasyEdit baselines on supported macro and overall in every domain. These streams expose the failure mode that latest-only editors leave open: a temporally plausible but wrong historical value in place of the correct one or the explicit unknown-time option. Because the external baselines are completed only at 3B and the domains remain small, this block is transfer evidence rather than a matched-scale co-existence benchmark. In particular, IRS and FDA are best read as modest-domain checks rather than broad validation.

Expanded IRS/FDA follow-ups, prompt-paraphrase checks, visible-history noise, and timestamp-shuffle controls are reported in Appendix Table 18, Table 14, Table 15, and Appendix A.13. The expanded FDA stream is the clearest stress case: VISTA preserves current

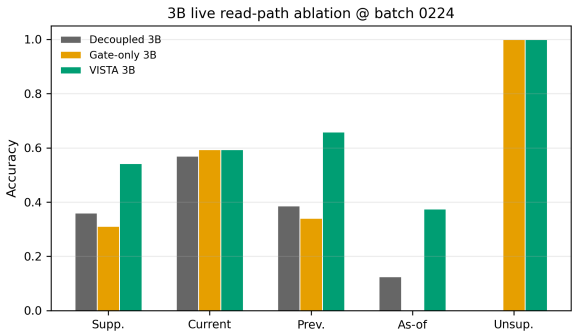

Figure 3: Live read-path attribution on the 3B batch-0224 endpoint. Validity gating alone restores `unsupported_time` but does not recover supported retrieval; the full VISTA read path restores both.

and mostly preserves unsupported-time behavior, but late `as_of_revision` falls sharply. We treat that as evidence that deeper as-of retention remains unsolved, not as a contradiction of the current/previous coexistence claim.

4.7 Supporting Analyses: Read-Path Attribution

Figure 3 shows where the final gain comes from. On the 3B batch-0224 endpoint, visible-stream va-

Table 2: Cross-family late-slice check at batch 0224. The Llama rows test whether the same decoupled-to-VISTA transition appears on a second backbone family.

Family	Setting	Current ↑	Previous ↑	As-of ↑	Unsupported ↑	Overall ↑
Qwen2.5-3B	VISTA	0.594	0.659	0.375	1.000	0.721
Qwen2.5-7B	VISTA	**0.602**	0.648	0.375	1.000	0.721
Llama-3.2-3B	Decoupled	0.523	0.409	0.000	0.000	0.330
Llama-3.2-3B	VISTA	0.523	0.671	0.125	1.000	0.689
Llama-3.1-8B	VISTA	0.578	**0.682**	**0.500**	1.000	**0.724**

Table 3: Matched Qwen2.5-7B late-slice check at batch 0224. The native-read rows compare finished systems; the matched-read rows give WISE and UltraEdit the same validity gate and fact-local restriction as VISTA to test whether read-contract alignment alone explains the gap.

Method	Current ↑	Previous ↑	As-of ↑	Unsupported ↑	Overall ↑
VISTA internal (7B)	0.602	0.648	0.375	1.000	0.721
WISE 7B	0.141	0.068	0.125	0.000	0.080
WISE 7B + matched read	0.133	0.443	0.250	1.000	0.468
UltraEdit 7B	0.195	0.114	0.125	0.000	0.115
UltraEdit 7B + matched read	0.195	0.341	0.250	1.000	0.465

lidity gating alone fixes `unsupported_time` but leaves supported historical retrieval weak; adding fact-local candidate restriction restores the full VISTA row in Table 11. The separate WISE and UltraEdit fairness controls in Table 3 show that read-contract alignment matters but does not by itself close the gap to VISTA. Detailed numeric rows and saved diagnostics appear in Appendix Table 11, Table 12, Table 5, and Table 6.

5 Related Work

Most post-deployment editing methods optimize latest-write retention, while our task asks one deployed state to support several temporal views of the same fact.

Model editing and writable internal memory. Most editing methods are optimized for getting the latest value into the model and keeping it there. Direct locate-then-edit methods such as ROME, MEMIT, and AlphaEdit rewrite localized factual associations once or in bulk, with AlphaEdit adding a null-space projection to better preserve unrelated knowledge under sequential edits (Meng et al., 2022a,b; Fang et al., 2025); MEND and SERAC learn how to transform or retrieve edits more scalably (Mitchell et al., 2022a,b). Lifelong editors such as GRACE, MELO, WISE, UltraEdit, MEMOIR, and SoLA reduce overwrite under long edit streams through routing, one-step editing, sparse residual memory, or reversible updates (Hartvigsen et al., 2023; Yu et al., 2023; Wang et al., 2024a; Gu et al., 2025; Wang et al., 2025; Luo et al., 2026). MemoryLLM and its FFN-memory extension ask a related question from the memory-architecture side (Wang et al., 2024b; Jaiswal et al., 2026). We focus matched comparisons on the lifelong family (WISE, UltraEdit, GRACE) because it directly targets sequential update streams; locate-then-edit methods are closer in spirit to one-shot or batch direct edits and a symbolic or metadata-only multiversion store would test a different upper bound. Our target is also different: one frozen end state must answer $p(y \mid x, t)$ for several valid temporal slices and abstain when t is unsupported.

Temporal knowledge editing. Temporal knowledge editing benchmarks are direct task-level relatives. AToKe introduced historical/current temporal editing and METO adds a time objective to improve historical retention after edits (Yin et al., 2024). MTKE and SPIKE extend this line to multi-granularity timestamps and sparse parameter injection (Zhao et al., 2026). Temple-MQA/TKeMqa studies multi-hop QA under temporal knowledge edits with a time-aware graph and structural retrieval (Cheng et al., 2024). VISTA differs in interface: it evaluates one final edited state under a declared visible-stream contract, separates unsupported-time validity from supported value retrieval, and includes matched-read controls to test how much of the gain comes from read-time metadata rather than the editor alone. The point is not to replace temporal QA, but to ask a narrower sys-

Table 4: Official-source temporal benchmarks from completed VISTA runs. Each row aggregates a full streaming run on one public revision source, so this is a stream-average transfer check rather than a late final-slice coexistence endpoint. Supported macro is the mean of `current`, `previous`, and `as_of_revision` over the full stream and should be read as transfer evidence, not as proof of broad as-of retrieval. VISTA leads the completed 3B EasyEdit baselines on supported macro and overall on all four domains, while the 7B replication stays close to the 3B row and keeps `unsupported_time` at 1.0 on every completed stream.

	Supported macro ↑				Overall ↑			
Benchmark	VISTA 3B	VISTA 7B	UltraEdit 3B	WISE 3B	VISTA 3B	VISTA 7B	UltraEdit 3B	WISE 3B
Visa Bulletin	**0.439**	0.427	0.042	0.036	**0.477**	0.464	0.073	0.029
IRS tax year	**0.667**	**0.667**	0.111	0.044	**0.717**	**0.717**	0.244	0.133
FDA label revision	**0.667**	0.618	0.083	0.132	**0.642**	0.602	0.133	0.026
HHS poverty guidelines	0.323	**0.350**	0.096	0.082	0.465	**0.491**	0.122	0.073

tems question about what remains available in one edited state after the update stream has ended.

Temporal QA and time-indexed reasoning. Temporal QA studies related questions under a different interface. TempLAMA evaluates time-conditioned factual prediction from pretrained states rather than post-deployment multiversion editing (Dhingra et al., 2022); recent work on temporal conflict and TRACK shows that multi-step reasoning still degrades under conflicting supplied evidence (Özer and Yıldız, 2025; Feng et al., 2026). Temporal knowledge graph methods treat time as a first-class input over explicit timestamped graphs rather than a frozen model with internal edited memory (Cai et al., 2022, 2024; Hou et al., 2024). A symbolic multiversion store would likewise answer a different question—how well explicit external structure serves temporal queries, not how much temporal discrimination remains inside an edited parametric state under a fixed read contract.

Continual learning and multiversion semantics. The mechanism story echoes continual-learning accounts of interference (EWC, PackNet) (Kirkpatrick et al., 2017; Mallya and Lazebnik, 2018), with associative-memory theory offering vocabulary for content-addressable reads (Hu et al., 2025). We also borrow framing from bitemporal and multiversion storage systems, where current and historical reads are different workloads rather than variants of one lookup (Rost et al., 2021; Coccimiglio et al., 2026)—separating those responsibilities is what latest-only editing tends to collapse.

6 Discussion

The ablation ladder points to an architectural bottleneck: only VISTA combines previous-value retrieval with explicit unknown-time selection in the same deployed state, and the matched-read WISE and UltraEdit controls show that read-contract alignment alone does not recreate the VISTA endpoint. Resolver diagnostics show the same separation: a global resolver reaches only 0.3365 overall; the visible-stream validity gate lifts that to 0.6186 without improving `previous`; the full VISTA read path reaches 0.7212 and raises `previous` to 0.6591. That progression makes the result a combination of storage layout and read-contract factorization, not a raw metadata lookup.

Cross-method evidence supports the same boundary: the Llama rows suggest the transition is not specific to Qwen, the official-source streams show the contract transfers to practical temporal records at the scale tested here, and the longer continuation reinforces why `as_of_revision` should remain a stress probe.

We do not claim to outperform a purpose-built symbolic or metadata-only multiversion store; that is a different baseline question. The strongest evidence here is current/previous version selection plus validity handling, not open-ended temporal generation or broad native as-of retrieval.

7 Conclusion

VISTA shows that one-state temporal coexistence depends on both storage layout and read-time semantics: separating unsupported-time validity from supported historical selection is the decisive read-time step, while distinct hot, snapshot, and archive roles preserve the relevant values during training. The same shift appears across the Qwen ladder, matched-read controls, Llama, and the transfer suite, so temporal behavior in edited models should be evaluated as a system interface, not just latest-write retention.

Limitations

- **No symbolic or metadata-only upper bound.** A purpose-built symbolic multiversion store or a no-adapter metadata-only scorer would be the right comparator for whether parametric edited memory is the best interface for temporal storage; we do not run that comparison. Our claim is internal to edited model states under a shared closed-answer interface.
- **Native `as_of_revision` stress probe.** Batch 0224 has only 8 native `as_of_revision` examples and the 35-batch continuation is correlated within the same stream, so we do not claim to solve broad as-of retrieval.
- **Read-contract assumptions.** Native-read and matched-read comparisons answer different questions, and the VISTA read contract assumes a clean visible revision timeline.
- **Bounded evidence base.** The main benchmark is one cleaned 8-file WikiBigEdit stream, and the official-source suites are still small relative to application-scale temporal records.
- **Closed-answer evaluation.** The strongest results use a fixed closed-answer scorer; we do not claim comparable strength for open-ended temporal generation.

Ethics Statement

We study temporal factual access in frozen language models on public benchmarks and public official-source records (IRS, FDA/DailyMed, HHS, and State Department Visa Bulletin). No human subjects or private data are involved. The main societal risk we see is over-reliance on a partial temporal memory in compliance-sensitive settings such as tax, health, or immigration guidance. For that reason the system is built to abstain on unsupported times, and we frame the result as a diagnostic finding rather than a deployment-ready tool. Licenses of the underlying benchmark, model families, and EasyEdit baselines are respected as released.

Anonymous code for this paper is available at our anonymous repository README.

References

Borui Cai, Yong Xiang, Longxiang Gao, He Zhang, Yunfeng Li, and Jianxin Li. 2022. Temporal knowledge graph completion: A survey. *Preprint*, arXiv:2201.08236.

Li Cai, Xin Mao, Yuhao Zhou, Zhaoguang Long, Changxu Wu, and Man Lan. 2024. A survey on temporal knowledge graph: Representation learning and applications. *Preprint*, arXiv:2403.04782.

Keyuan Cheng, Gang Lin, Haoyang Fei, Yuxuan Zhai, Lu Yu, Muhammad Asif Ali, Lijie Hu, and Di Wang. 2024. Multi-hop question answering under temporal knowledge editing. *Preprint*, arXiv:2404.00492.

Gaetano Coccimiglio, Trevor Brown, and Srivatsan Ravi. 2026. Multiverse: Transactional memory with dynamic multiversioning. *Preprint*, arXiv:2601.09735.

Bhuwan Dhingra, Jeremy R. Cole, Julian Martin Eisenschlos, Daniel Gillick, Jacob Eisenstein, and William W. Cohen. 2022. Time-aware language models as temporal knowledge bases. *Transactions of the Association for Computational Linguistics*, 10:257–273.

Junfeng Fang, Houcheng Jiang, Kun Wang, Yunshan Ma, Jie Shi, Xiang Wang, Xiangnan He, and Tat-Seng Chua. 2025. AlphaEdit: Null-space constrained knowledge editing for language models. In *The Thirteenth International Conference on Learning Representations*.

Yiyang Feng, Zeming Chen, Haotian Wu, Jiawei Zhou, and Antoine Bosselut. 2026. Tracking the limits of knowledge propagation: How LLMs fail at multi-step reasoning with conflicting knowledge. In *Proceedings of the 19th Conference of the European Chapter of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5813–5847, Rabat, Morocco. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, and 1 others. 2024. The Llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Xiaojie Gu, Ziying Huang, Jia-Chen Gu, and Kai Zhang. 2025. UltraEdit: Training-, subject-, and memory-free lifelong editing in language models. *Preprint*, arXiv:2505.14679.

Thomas Hartvigsen, Swami Sankaranarayanan, Hamid Palangi, Yoon Kim, and Marzyeh Ghassemi. 2023. Aging with GRACE: Lifelong model editing with discrete key-value adaptors. In *Advances in Neural Information Processing Systems*, volume 36.

Zhongni Hou, Xiaolong Jin, Zixuan Li, Long Bai, Jiafeng Guo, and Xueqi Cheng. 2024. Selective temporal knowledge graph reasoning. *Preprint*, arXiv:2404.01695.

Jerry Yao-Chieh Hu, Bo-Yu Chen, Dennis Wu, Feng Ruan, and Han Liu. 2025. Nonparametric modern hopfield models. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 24232–24269.

Internal Revenue Service. 2026. Inflation Adjustment and Tax Tables.

Ajay Jaiswal, Lauren Hannah, Han-Byul Kim, Duc Hoang, Arnav Kundu, Mehrdad Farajtabar, and Minsik Cho. 2026. MemoryLLM: Plug-n-play interpretable feed-forward memory for transformers. *Preprint*, arXiv:2602.00398.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A. Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, Demis Hassabis, Claudia Clopath, Dharshan Kumaran, and Raia Hadsell. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the National Academy of Sciences*, 114(13):3521–3526.

Haihua Luo, Xuming Ran, Tommi Kärkkäinen, Zhonghua Chen, Jiangrong Shen, Qi Xu, and Fengyu Cong. 2026. Reversible lifelong model editing via semantic routing-based LoRA. *Preprint*, arXiv:2603.11239.

Arun Mallya and Svetlana Lazebnik. 2018. Packnet: Adding multiple tasks to a single network by iterative pruning. *Preprint*, arXiv:1711.05769.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2022a. Locating and editing factual associations in GPT. In *Advances in Neural Information Processing Systems*, volume 35.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2022b. Mass-editing memory in a transformer. *Preprint*, arXiv:2210.07229.

Meta Llama Team. 2024. Llama 3.2 model card.

Eric Mitchell, Charles Lin, Antoine Bosselut, Chelsea Finn, and Christopher D. Manning. 2022a. Fast model editing at scale. In *The Tenth International Conference on Learning Representations*.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022b. Memory-based model editing at scale. In *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15817–15831.

Office of the Assistant Secretary for Planning and Evaluation, U.S. Department of Health and Human Services. 2026. Poverty Guidelines.

Atahan Özer and Çağatay Yıldız. 2025. Question answering under temporal conflict: Evaluating and organizing evolving knowledge with LLMs. *Preprint*, arXiv:2506.07270.

Qwen Team. 2024. Qwen2.5 Technical Report. *Preprint*, arXiv:2412.15115.

Christopher Rost, Philip Fritzsche, Lucas Schons, Maximilian Zimmer, Dieter Gawlick, and Erhard Rahm. 2021. Bitemporal property graphs to organize evolving systems. *Preprint*, arXiv:2111.13499.

Lukas Thede, Karsten Roth, Matthias Bethge, Zeynep Akata, and Thomas Hartvigsen. 2025. WikiBigEdit: Understanding the limits of lifelong knowledge editing in LLMs. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 59326–59354.

U.S. Department of State. 2026. Visa Bulletin.

U.S. National Library of Medicine. 2026. DailyMed SPLs SetID History API.

Ke Wang, Yiming Qin, Nikolaos Dimitriadis, Alessandro Favero, and Pascal Frossard. 2025. MEMOIR: Lifelong model editing with minimal overwrite and informed retention for LLMs. In *Advances in Neural Information Processing Systems*, volume 38.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024a. WISE: Rethinking the knowledge memory for lifelong model editing of large language models. *Preprint*, arXiv:2405.14768.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, Jingbo Shang, and Julian McAuley. 2024b. MEMORYLLM: Towards self-updatable large language models. *Preprint*, arXiv:2402.04624.

Xunjian Yin, Jin Jiang, Liming Yang, and Xiaojun Wan. 2024. History matters: Temporal knowledge editing in large language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 19413–19421.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2023. MELO: Enhancing model editing with neuron-indexed dynamic LoRA. *Preprint*, arXiv:2312.11795.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Ziwen Xu, Wenhao Liang, Kai Zhang, Fei Huang, and Huajun Chen. 2024. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

Simiao Zhao, Ning Pang, Zhen Tan, Yanli Hu, Weidong Xiao, and Xiang Zhao. 2026. Multi-granularity temporal knowledge editing over large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, pages 28742–28750.

Table 5: Post-hoc abstention operating points from saved Qwen2.5-3B split-versus-replay probes. Panel A shows the original full version-selection probe. Panel B shows the deeper 8-file carryover surface from the finished paper suite at each method's selected threshold. In both settings, split preserves useful supported behavior while replay reaches high `unsupported_time` rejection mainly by abstaining on almost everything.

Method	Threshold	Current	Previous	Unsupported	Supported overall	Coverage
Panel A: full version-selection probe						
Split post-hoc	0.0947	0.9063	0.0156	1.0000	0.4609	0.6250
Replay post-hoc	0.5490	0.0000	0.0313	1.0000	0.0156	0.0234
Panel B: deep 8-file carryover suite						
Split carryover	0.3445	0.4297	0.0568	0.9886	0.2679	0.3259
Replay carryover	0.1059	0.0000	0.0000	0.9886	0.0000	0.0045

Table 6: Late-band read-time diagnostics at batch 0224 for Qwen2.5-3B decoupled and VISTA readouts. Global keeps the original saved global candidate surface. Global + validity gate adds a post-hoc visible-stream temporal index over the saved evaluation slice. The live VISTA row is the end-to-end branch that combines that validity stage with fact-local historical selection. Observable is a saved historical-only readout whose overall column is a composite estimate formed from the saved current slice plus the stronger historical resolver.

Readout	Overall	Current	Previous	As-of	Unsupported
Live decoupled (3B)	0.3462	0.5703	0.3864	0.1250	0.0000
Global resolver	0.3365	0.5625	0.3636	0.1250	0.0000
Global + validity gate	0.6186	0.5625	0.3636	0.1250	1.0000
Live VISTA read (3B)	0.7212	0.5938	0.6591	0.3750	1.0000
Observable saved readout	0.7179	0.5703	0.6591	0.6250	1.0000

A Additional Ablations and Supporting Analyses

A.1 Saved-Checkpoint Abstention and Resolver Diagnostics

This appendix collects support-only intermediate variants, saved-checkpoint analyses, robustness probes, and implementation details for the final VISTA method.

These tables separate abstention from retrieval. High unsupported-time rejection is useful only when supported retrieval survives at the same operating point: split preserves useful supported behavior under matched abstention floors, while replay largely abstains on everything. The saved decoupled checkpoints also expose the two ingredients later combined in the live VISTA branch. A visible-stream validity gate fixes unsupported-time on the saved slice without harming the global saved readout, and fact-local historical selection improves saved predecessor/as-of retrieval beyond the plain global resolver.

A.2 Abstention Sensitivity Across Operating Points

Table 7 expands the selected post-hoc abstention threshold into matched unsupported-time operating points. On the original full saved probe, split keeps supported overall fixed at 0.4609 while unsupported-time rejection tightens from 0.8125 to 1.0000, whereas replay falls from 0.0938 to 0.0156. On the deeper 8-file carryover surface, split again dominates replay at every matched unsupported-time floor, although both methods lose supported coverage as the abstention target becomes stricter. We use abstention as supporting evidence for stored signal, not as proof that the live architecture already solves historical retrieval.

A.3 Late-Band Temporal Readout Across Checkpoints

Table 8 expands the saved late-band comparison across checkpoints 0200, 0208, 0216, and 0224. The same pattern holds throughout the late band: fact-local temporal scoping consistently improves supported retrieval relative to the saved decoupled live state, and the observable checkpoint readout further improves `as_of_revision` while restoring perfect `unsupported_time` handling in the saved historical-only analysis. These remain saved-checkpoint analyses rather than the live end-to-end result, but they are consistent with the ingredients later used by the live VISTA branch.

Table 7: Matched unsupported-time operating points for the saved abstention analyses. Supported overall and coverage are reported after abstaining below the threshold needed to reach each unsupported-time floor. Split preserves materially more supported utility than replay on both the original full probe and the deeper 8-file carryover probe.

Surface	Method	Unsupported floor	Supported overall	Coverage
Full saved version-selection probe				
Full	Split	0.8125	0.4609	0.7422
Full	Split	0.9063	0.4609	0.6875
Full	Split	0.9531	0.4609	0.6406
Full	Split	1.0000	0.4609	0.6250
Full	Replay	0.8125	0.0938	0.2031
Full	Replay	0.9063	0.0469	0.1094
Full	Replay	0.9531	0.0391	0.0625
Full	Replay	1.0000	0.0156	0.0234
Deep 8-file carryover probe				
Deep 8-file	Split	0.8068	0.3839	0.5491
Deep 8-file	Split	0.9205	0.3661	0.5000
Deep 8-file	Split	0.9545	0.3080	0.3750
Deep 8-file	Split	1.0000	0.1562	0.1652
Deep 8-file	Replay	0.8182	0.0000	0.1429
Deep 8-file	Replay	0.9091	0.0000	0.0938
Deep 8-file	Replay	0.9545	0.0000	0.0357
Deep 8-file	Replay	1.0000	0.0000	0.0045

A.4 Compact Robustness Summary

Table 9 summarizes the bootstrap ranges on the main batch-0224 surface together with the 35-batch native-as-of continuation. The continuation is correlated within the same stream, so it documents the stress probe rather than establishing broad as-of retrieval.

A.5 Run Configuration and Reproducibility

The main Qwen2.5-3B VISTA run uses `Qwen/Qwen2.5-3B-Instruct` in bf16 with LoRA adapters on `gate_proj`, `up_proj`, and `down_proj`. The three memory roles use rank 16, alpha 32, and dropout 0.0; capacities are 64 for `level_0` and 256 each for `level_1_snapshot` and `level_1_archive`. Training uses seed 7, maximum length 256, learning rate 5×10^{-4}, batch size 1, gradient accumulation 1, and 8 epochs. The stream has 28 batches of size 8, compaction every 2 batches, a final compaction, and at most 64 update steps per batch.

The final VISTA read contract sets `validity_gate_mode=visible_stream_valid_time` and `historical_candidate_mode=fact_local`. These keys build $V(x)$ and $C(x, t)$ only from the visible append-only stream. The matched-read WISE and UltraEdit controls receive the same two read-time structures, so their comparison isolates whether the VISTA endpoint is explained only by metadata alignment. Prompt templates, preprocessing, and run scripts are included in the anonymous code release; the main text reports the configuration values needed to identify the completed artifacts.

A.6 Candidate-Control Sweep

Table 10 records the completed 28-prefix candidate-control sweep. Fact-local candidates remain the main reference point, same-relation hard negatives are slightly harder, and the size-matched random control remains the easiest setting. This ordering is consistent with the factorized reader exploiting chain structure rather than only raw candidate count.

A.7 Older Storage-Role Attribution

Table 11 preserves the compact live 3B read-path table referenced from the main text. We move the numeric rows here in the submission version so the body can spend its page budget on the larger end-to-end and transfer evidence objects.

Table 12 preserves the earlier split-versus-replay component table that motivated the final VISTA read path. We moved it out of the main text because the paper's strongest current claim is now about the

Table 8: Late-band temporal diagnostics across the Qwen2.5-3B decoupled checkpoints. Panel A reports the saved live state and the fact-local readout. Panel B reports the stronger observable historical-only readout. Splitting the table keeps the late-band comparisons readable without collapsing the font.

Panel A: live state vs. fact-local readout						
Checkpoint	Live overall	Live prev.	Live as-of	Fact-local overall	Fact-local prev.	Fact-local as-of
0200	0.3483	0.3134	0.5000	0.4719	0.7612	1.0000
0208	0.2908	0.3243	0.1667	0.4043	0.7162	0.6667
0216	0.2651	0.2927	0.1667	0.3826	0.6829	0.6667
0224	0.3462	0.3864	0.1250	0.4263	0.6591	0.3750

Panel B: observable historical-only readout				
Checkpoint	Observable prev.	Observable as-of	Observable unsupported	Observable overall (est.)
0200	0.6567	0.6667	1.0000	0.6891
0208	0.6622	0.6667	1.0000	0.6525
0216	0.6341	0.6667	1.0000	0.6443
0224	0.6591	0.6250	1.0000	0.7179

Table 9: Compact robustness summary for the main 3B surface. The bootstrap intervals quantify uncertainty on the observed batch-0224 slice, and the fullprefix continuation documents the native as-of stress probe on a correlated extension of the same stream.

Evidence	Surface	Result
Replay bootstrap supported macro	WikiBigEdit batch 0224	$[0.000, 0.019]$
Decoupled bootstrap supported macro	WikiBigEdit batch 0224	$[0.287, 0.460]$
VISTA bootstrap supported macro	WikiBigEdit batch 0224	$[0.426, 0.665]$
Native as-of continuation	35-batch fullprefix, 119 examples	$62/119 = 0.5210$

completed VISTA method and the matched fairness control, not about the older split checkpoint story by itself.

A.8 Smaller-Scale External Editors

Figure 4 keeps the cross-scale view focused on WISE and UltraEdit because they are the only finished 3B/7B EasyEdit pairs. Table 13 adds the supporting 3B-only FT and GRACE reruns together with the practical 3B WISE matched-read proxy. The qualitative boundary stays the same: the 3B external-editor rows remain far below the final internal VISTA system on the late slice.

A.9 Prompt-Paraphrase Robustness

Table 14 adds the finished prompt-paraphrase checks that were run on the same late-slice Qwen2.5-3B surface. The check targets one simple objection: a single question template might be carrying the main result. VISTA stays well above both matched-read controls on all three paraphrase sets, with overall in $[0.8077, 0.8173]$, `current` in $[0.7969, 0.8281]$, and `unsupported_time` fixed at 1.0. The matched-read WISE reruns stay far lower with overall in $[0.4391, 0.4583]$ and `current` only in $[0.0547, 0.0938]$, while UltraEdit remains similarly weak with overall in $[0.3974, 0.4423]$ and `current` dropping as low as 0.0391. The gap therefore survives modest question rewrites, although this is not a test of strict template invariance.

A.10 Timestamp-Shuffle Controls

Table 15 probes whether the visible-stream read contract depends on exact timestamp identities rather than on coherent within-chain temporal structure. We evaluate two controls on the late-four observable historical sweep. *Within-fact shuffle* permutes timestamps only inside each fact chain; *cross-fact shuffle* reassigns timestamps across different chains. The result is asymmetric. Within-fact shuffling leaves overall roughly unchanged ($0.6875 \rightarrow 0.6925$) and even raises observable `as_of_revision`, which suggests the factorized reader is not memorizing one rigid numeric timestamp pattern. Cross-fact shuffling is more

Table 10: Completed 28-prefix candidate-control sweep on the Qwen2.5-3B live factorized read path. The same sweep shows that the factorized reader remains strong on as-of-form examples even when the historical candidate pool is constrained.

Control	As-of forms	As-of-form mean ↑	Overall mean ↑
Fact-local candidates	757	0.8158	0.5244
Same-relation hard negatives	757	0.7830	0.5034
Size-matched random	757	0.9192	0.5909

Table 11: Read-path attribution on the completed 3B live batch-0224 surface. Validity gating alone restores `unsupported_time` but does not recover supported retrieval, while the full VISTA read path restores both.

Setting	Current ↑	Previous ↑	As-of ↑	Unsupported ↑	Overall ↑
Decoupled live (3B)	0.570	0.386	0.125	0.000	0.346
Validity-gated live (3B)	0.594	0.341	0.000	1.000	0.622
VISTA live (3B)	0.594	0.659	0.375	1.000	0.721

damaging: overall falls to 0.6750 and observable `as_of_revision` falls from 0.4375 to 0.3125, while `unsupported_time` remains at 1.0. We read the contract as relying on chain-consistent temporal assignment rather than on exact absolute timestamp identities.

A.11 Matched Read-Contract Probe and Runtime Cost

Table 16 adds the finished scorer-held-constant WISE and UltraEdit controls on the same batch-0224 slice used by the main table: we re-evaluate the finished 7B states with the same visible-stream validity gate and fact-local historical candidate restriction used by the internal VISTA method. These controls confirm that read-path engineering matters. For WISE, previous retrieval rises from 0.0682 to 0.4432, unsupported-time flips from 0.0 to 1.0, and overall rises from 0.0801 to 0.4679. For UltraEdit, previous retrieval rises from 0.1136 to 0.3409, unsupported-time flips from 0.0 to 1.0, and overall rises from 0.1154 to 0.4647. The same controls also show why the final claim cannot collapse into read-time filtering alone: WISE current stays weak at 0.1328, UltraEdit current stays flat at 0.1953, `as_of_revision` rises only to 0.2500 on an 8-example slice for both baselines, and both matched-read controls still trail the completed internal VISTA system by a wide margin.

The same artifact summarizes runtime cost. Across completed 7B full runs, WISE uses 9460.1 seconds of cumulative write time versus 972.5 for the internal VISTA method, while UltraEdit uses less write time (159.8 seconds) but still takes slightly longer end-to-end latency (14826.3 versus 13964.0 seconds). The late-slice fairness checks still add nontrivial runtime (868.4 seconds latency and 1584.2 seconds write for WISE; 887.4 seconds latency and 10.0 seconds write for UltraEdit). Figure 5 visualizes the same trade-off: the internal VISTA method is far cheaper to write than WISE, not as cheap to write as UltraEdit, and exhaustive matched-read sweeps across all baselines are a real budget constraint on one GPU.

A.12 MemoryLLM-8B Partial Comparison

Table 17 reports the partial MemoryLLM-8B run referenced in the limitations. The setup uses the public MemoryLLM-8B implementation on the same cleaned 8-file WikiBigEdit stream as the main benchmark, with the same prompt surface and closed-answer scoring rule used for VISTA, WISE, and UltraEdit. We report the most recent saved checkpoint (`batch_0088`, 11 of 28 batches) before stopping the run. At that slice MemoryLLM-8B is below UltraEdit-3B and only marginally above WISE-3B, with previous and `unsupported_time` both at 0.0; its per-example latency is roughly 26× WISE-3B and 75× UltraEdit-3B. Under our one-GPU budget, completing the remaining 17 batches would have required 1–2 additional days for a run that was weak on the metrics that matter here. We treat the row as a documented cost point, not as a finished matched baseline.

Table 12: Older storage-role attribution on the late deep 8-file split and replay checkpoints. The split model shows the earlier division of labor that motivated the final VISTA read path: snapshot carries most current recall, while archive carries older-version retrieval.

Checkpoint	Hot cur.	Snapshot cur.	Archive prev.	Archive as-of	Replay cur.	Replay prev.	Best routed
0200	0.1181	0.5433	0.2985	0.5000	0.0630	0.0000	0.3558
0208	0.1016	0.3594	0.3108	0.1667	0.0000	0.0405	0.2589
0216	0.3672	0.3281	0.2561	0.1667	0.0156	0.0122	0.2685
0224	0.1562	0.5859	0.3523	0.1250	0.0000	0.0227	0.3558

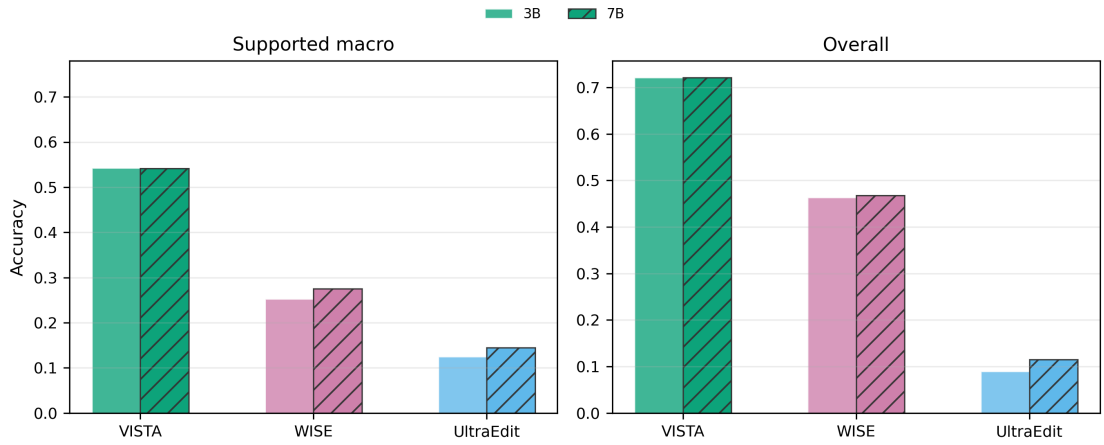

Figure 4: Cross-scale comparison for the main-surface external editors. The 7B EasyEdit reruns improve slightly over their 3B counterparts, but neither WISE nor UltraEdit approaches the completed VISTA system on supported macro or overall.

A.13 Noisy-Timeline Robustness Probe

The main paper assumes a clean visible-stream validity index $V(x)$. Figure 6 summarizes the corresponding robustness probe, where we uniformly drop 5%, 10%, and 20% of eligible historical visible records from $V(x)$ before scoring on the same Qwen2.5-3B `batch_0224` endpoint used in the main table. Two qualitative findings hold across the sweep. First, `unsupported_time` stays at 1.0 at every noise level, so the abstention behavior does not collapse under modest visible-history corruption. Second, overall degradation is small and monotone: $0.7212 \rightarrow 0.7051 \rightarrow 0.6987 \rightarrow 0.6795$ from 0% to 20% dropout, while `previous` degrades monotonically as expected when historical evidence is removed. The native `as_of_revision` column still wobbles because the slice has only 8 examples, so we do not read directional conclusions from it. This is evidence of graceful degradation, not a full study of adversarial or timestamp-jitter corruption modes.

A.14 Expanded Official-Source Follow-Ups

Base stream construction. The base official-source suite contains 59 Visa Bulletin records across 12 chains (mean depth 4.9, 8 batches), 15 IRS records across 5 chains (mean depth 3.0, 3 batches), 19 FDA label revisions across 8 chains (mean depth 2.4, 3 batches), and 72 HHS poverty-guideline updates across 12 chains (mean depth 6.0, 9 batches). These counts are setup context for the transfer block.

Table 18 collects the larger 3B IRS and FDA runs referenced in the main text. These follow-ups matter because the base official-source block is small, especially for IRS and FDA. The expanded IRS stream remains supportive at the final slice: VISTA reaches final overall 0.7500, final `as_of_revision` 0.6000, and final `unsupported_time` 1.0. The expanded FDA stream is the opposite stress test: much longer revision chains preserve final `current` and mostly preserve `unsupported_time`, but late `as_of_revision` falls to 0.0417. The corresponding UltraEdit follow-up is effectively collapsed even when we freeze a practical late-slice proxy: mean overall is only 0.0233, final overall is 0.0104, final `current` is 0.0417, and both final `as_of_revision` and final `unsupported_time` are 0.0.

Table 13: Supporting 3B EasyEdit baselines on the same late-slice surface. These smaller reruns preserve the same qualitative conclusion as the matched 7B control in Table 3: the 3B external-editor rows remain far below the final internal system on the late slice. The WISE matched-read row is a late-slice proxy rather than a separate finished native 0224 rerun.

Method	Supported ↑	Current ↑	Previous ↑	As-of ↑	Unsup. ↑	Overall ↑
FT 3B	0.0052	0.0156	0.0000	0.0000	0.0000	0.0064
GRACE 3B	0.0968	0.0859	0.0795	0.1250	0.0000	0.0609
WISE 3B + matched read	0.2528	0.1406	0.4512	0.1667	1.0000	0.4631
UltraEdit 3B	0.1250	0.1250	0.1250	0.1250	0.0000	0.0897

Table 14: Prompt-paraphrase robustness on the late-slice Qwen2.5-3B matched-read surface. The reference rows use the finished default template, and the paraphrase rows rewrite the same temporal questions with alternate surface forms. VISTA remains well above the matched-read WISE and UltraEdit controls on every variant, so the late-slice gap is not specific to one exact wording.

Setting	Overall ↑	Current ↑	Previous ↑	As-of ↑	Unsupported ↑
VISTA 3B baseline	0.7212	0.5938	0.6591	0.3750	1.0000
VISTA 3B paraphrase v1	0.8173	0.8281	0.6364	0.6250	1.0000
VISTA 3B paraphrase v2	0.8077	0.8125	0.6250	0.6250	1.0000
VISTA 3B paraphrase v3	0.8077	0.7969	0.6364	0.7500	1.0000
WISE 3B matched-read ref.	0.4631	0.1406	0.4512	0.1667	1.0000
WISE 3B paraphrase v1	0.4583	0.0938	0.4545	0.3750	1.0000
WISE 3B paraphrase v2	0.4487	0.0781	0.4432	0.3750	1.0000
WISE 3B paraphrase v3	0.4391	0.0547	0.4432	0.3750	1.0000
UltraEdit 3B baseline	0.4487	0.1250	0.3750	0.3750	1.0000
UltraEdit 3B paraphrase v1	0.4391	0.1094	0.3636	0.3750	1.0000
UltraEdit 3B paraphrase v2	0.4423	0.1016	0.3864	0.3750	1.0000
UltraEdit 3B paraphrase v3	0.3974	0.0391	0.3182	0.3750	1.0000

The visible-stream validity contract transfers and remains strong for explicit abstention, but supported historical selection is still sensitive to deeper, noisier official revision chains.

A.15 Selected Qualitative Cases

The qualitative rows come from the completed VISTA 3B and UltraEdit 3B real-benchmark runs rather than hand-written illustrative cases. This keeps the appendix examples tied to the practical transfer streams without duplicating a separate standalone WikiBigEdit case table.

Table 15: Timestamp-shuffle controls on the late-four observable historical sweep of the Qwen2.5-3B VISTA read path. The baseline rows are the finished observable late-four endpoints from the same full-prefix artifact. *Within-fact shuffle* permutes timestamps only within each fact chain; *cross-fact shuffle* reassigns timestamps across different chains. Unsupported-time remains fixed at 1.0, while cross-fact shuffling hurts historical selection more than within-fact shuffling.

Setting	Overall ↑	Previous ↑	As-of ↑	Unsupported ↑
Baseline late-four observable @0273	0.6875	0.6484	0.4375	1.0000
Within-fact timestamp shuffle	0.6925	0.6406	0.6250	1.0000
Cross-fact timestamp shuffle	0.6750	0.6250	0.3125	1.0000

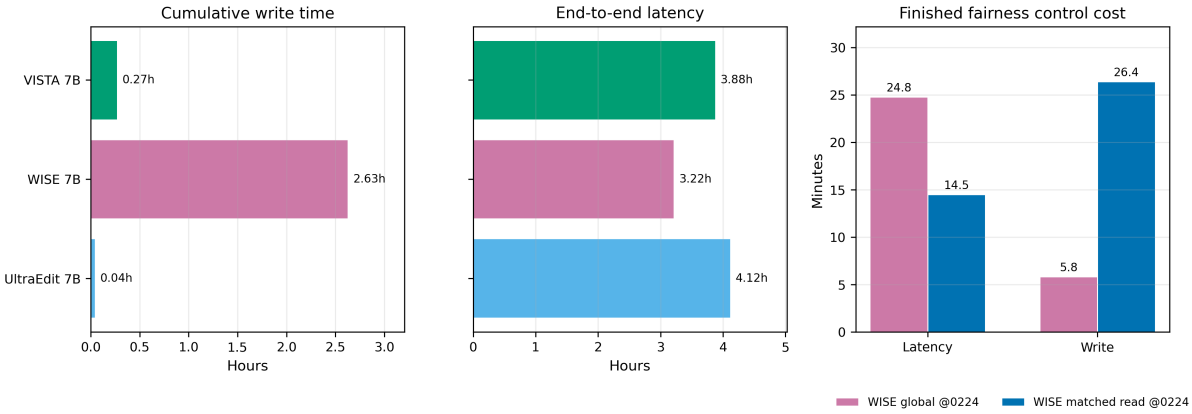

Figure 5: Runtime and overhead summary from completed local artifacts. Left and middle: cumulative write time and end-to-end latency for the finished 7B VISTA, WISE, and UltraEdit runs on the main benchmark. Right: the finished batch-0224 WISE fairness control still adds cost.

Table 16: Scorer-held-constant probes for the WISE and UltraEdit 7B baselines on the matched batch-0224 slice, together with the local runtime cost summary. The matched-read rows reuse the finished native states but apply the same visible-stream validity gate and fact-local historical candidate restriction used by the internal VISTA method. Supported macro is the mean of `current`, `previous`, and `as_of_revision`.

Setting	Supported macro ↑	Current ↑	Previous ↑	As-of ↑	Unsupported ↑	Overall ↑	Latency (s) ↓	Write (s) ↓
WISE 7B global read @0224	0.1113	0.1406	0.0682	0.1250	0.0000	0.0801	1486.4	349.0
WISE 7B matched read @0224	0.2753	0.1328	0.4432	0.2500	1.0000	0.4679	868.4	1584.2
UltraEdit 7B global read @0224	0.1446	0.1953	0.1136	0.1250	0.0000	0.1154	2508.9	5.9
UltraEdit 7B matched read @0224	0.2621	0.1953	0.3409	0.2500	1.0000	0.4647	887.4	10.0
Internal VISTA 7B full run	0.5414	0.6016	0.6477	0.3750	1.0000	0.7212	13964.0	972.5
WISE 7B full run	0.1113	0.1406	0.0682	0.1250	0.0000	0.0801	11580.8	9460.1
UltraEdit 7B full run	0.1446	0.1953	0.1136	0.1250	0.0000	0.1154	14826.3	159.8

Method	Scale	Slice	Overall	Current	Previous	`unsupported_time`
MemoryLLM (partial)	8B	batch_0088	0.226	0.253	0.000	0.000
WISE	3B	batch_0088	0.194	0.205	0.200	0.000
UltraEdit	3B	batch_0088	0.280	0.301	0.200	0.000
VISTA (ours)	3B	batch_0224	0.721	0.594	0.659	1.000

Table 17: Partial MemoryLLM-8B comparison on the main stream. The MemoryLLM row is the last saved checkpoint (11 of 28 batches) before the run was stopped; per-example latency was about 74 seconds, against ~ 2.7 seconds for WISE-3B and ~ 0.95 seconds for UltraEdit-3B on the same slice. The VISTA row is the final batch-0224 endpoint reported in the main text and is included only as a reference point.

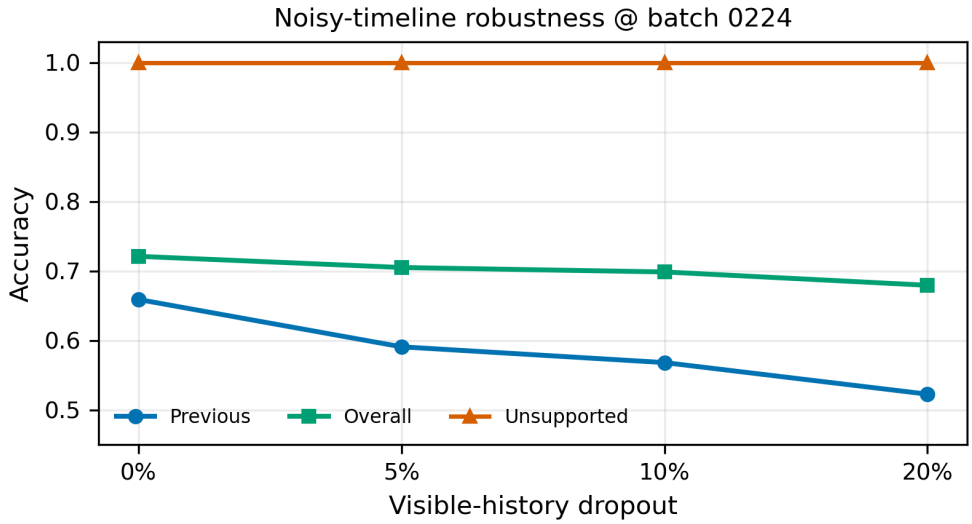

Figure 6: Noisy-timeline robustness on the Qwen2.5-3B VISTA `batch_0224` endpoint. Modest visible-history dropout degrades `previous` and overall gradually while leaving `unsupported_time` fixed at 1.0.

Table 18: Expanded official-source follow-ups on the larger IRS and FDA streams. The VISTA rows report the finished internal 3B runs. The UltraEdit FDA row is the practical late-slice proxy we retain for reporting because the remaining batches were too weak to change the conclusion.

System	Stream	Records	Mean overall ↑	Final overall ↑	Final current ↑	Final as-of ↑	Final unsupported ↑
VISTA 3B	IRS expanded	30	0.6139	0.7500	1.0000	0.6000	1.0000
VISTA 3B	FDA expanded	319	0.4652	0.5729	1.0000	0.0417	0.9167
UltraEdit 3B	FDA expanded	319	0.0233	0.0104	0.0417	0.0000	0.0000

Table 19: Selected qualitative cases from completed local 3B runs. VISTA preserves the correct temporal version or selects the explicit `[UNKNOWN_TIME]` option, abbreviated as `UNK`, while UltraEdit falls back to a temporally invalid value. The rows combine official-source transfer examples with one WikiBigEdit main-benchmark example.

Case	Query	Example	Gold	Ours	UltraEdit	Why it matters
IRS	previous	What was the 37% marginal-rate income threshold for single taxpayers at rev...	$626,350	$626,350	$14,600 to $15,000	Wrong past tax-year values can misstate deductions or audit-era thresholds.
FDA	previous	What was the published label date for NICOTINE (NICOTINE POLACRILEX) GUM, C...	Oct 25, 2024	Oct 25, 2024	Jul 06, 2021 – Nov 09, 2023	Clinical review needs the label version valid at prescription time.
HHS	unk.	What was the annual HHS poverty guideline for a 4-person household as of rev...	`UNK`	`UNK`	$39,000	Benefit eligibility depends on the correct historical guideline or on rejecting unsupported times.
WBE	unk.	What was KSI's sport as of revision 0?	`UNK`	`UNK`	Mtskheta-Mtianeti	One final state must reject unsupported revisions without losing supported temporal answers.