Mneme: The Shape of Memory in Machines

Memory interfaces across reader size and session horizon

AUGUST 2026

FARYNTH RESEARCH

Pronounced NEE-mee; from Ancient Greek mnēmē, meaning “memory.”

Farynth Research · Farynth LLC · August 2026 · DOI 10.5281/zenodo.22063323

Abstract

AI systems often preserve long histories outside a language model’s immediate context. Preserving an event, however, does not ensure that the model can use it later. Before the model answers, the memory system must choose which earlier events matter, resolve updates and contradictions, and arrange the result as input to the model. We call the receiving model the reader and the organized context it receives the memory interface.

To measure how much this representation matters, we tested memory interfaces across 256 generated project histories. Every condition used the same histories and questions. We compared seven deployable memory methods with a no-memory baseline and a diagnostic condition that supplied the records expected to answer each question. Two pinned Qwen 3.5 models, 4B and 9B, served as readers. We tested every condition at session 4, 16, and 64, which corresponded to 9, 47, and 194 generated event records before shared eligibility filtering. We used two context budgets and nine kinds of memory question, and estimated uncertainty by repeatedly resampling complete histories.

The condition named compiled state used the same fused record selection as the hybrid and typed conditions, but serialized those records as untagged bullets under a current-working-set heading. Among the deployable methods, it achieved the highest overall accuracy: 81.76% with the 4B reader and 90.00% with the 9B reader. Its advantage over recent history became clearer as the histories grew. Recent history gives the reader a chronological window of the newest authorized events. Between session 4 and session 64, its accuracy fell by 21.14 percentage points for the 4B reader and 31.92 points for the 9B reader. Over the same interval, compiled-state accuracy remained stable or slightly improved. The 9B reader was more accurate than the 4B reader under every deployable interface. Because both readers had the same no-memory floor, this between-reader gap is numerically equivalent to the reported difference in baseline-relative memory benefit. The study used lexical variations of one generated template and one model family, so it cannot establish how broadly these relationships hold.

1. Introduction

An AI assistant may correctly retain every event in a project and still answer from an obsolete decision. Suppose a team first chooses one deadline and later replaces it with another. A complete event log contains both dates. To answer a question about the current deadline, the system must do more than find those records: it must recognize the revision, decide which fact remains active, and present that state clearly enough for a language model to use. Storage can be complete while memory, as experienced through the answer, still fails.

This distinction matters for any long-running assistant or agent. Such systems accumulate decisions, corrections, procedures, commitments, relationships, and changing working state. Much of that history lives outside the model’s immediate context. When a later task depends on an earlier event, an external memory system selects information from that history and places it into the model’s input. We call the model that receives this input the **reader**. We call the complete reader-facing representation the **memory interface**.

A memory interface includes more than retrieval. It reflects which records the system may use, which records it selects, whether it adds related records, how it handles revisions and conflicts, how it orders evidence, and how it serializes the result. A chronological window may place the current fact inside a much larger body of routine history. A retrieval system may return relevant records in a form that still asks the reader to infer how they relate. A compact working set can reduce that burden before the reader sees the context. These are different ways of expressing the same stored history.

Most evaluations compare complete memory systems or ask whether a system retrieved relevant evidence. Those questions are important, but they can obscure the boundary between memory construction and model use. If a system changes both the representation and the reader, a better answer may reflect either change. If the system retrieves the correct records and the reader still answers incorrectly, retrieval recall alone cannot explain the failure.

Mneme isolates this boundary. We generated paired project histories with exact reference state, rendered each history through alternative memory interfaces, and asked pinned readers the same mechanically scored questions. The design

holds the underlying history and question constant while varying the interface, reader, session horizon, and context budget. It measures whether the resulting representation supports a correct answer and treats record selection as one stage of the memory process.

The study addresses three questions. First, how much does answer accuracy change when the same history reaches a reader through a different interface? Second, does that effect change as the history grows? Third, does a larger reader reduce the importance of interface design, or does it benefit differently from the same representations? We tested nine conditions across two Qwen 3.5 readers, three session horizons, two context budgets, and nine probe families. The release also preserves the associated token, time, energy, dedicated GPU memory, and thermal measurements.

The results show a consistent relationship within this experimental setting. The compiled condition led the deployable interfaces for both readers. It began with the same fused record selection as hybrid and typed memory, then presented those records in a more compact form. Recent history weakened as the generated histories accumulated sessions, while compiled state remained stable or slightly improved. The 9B reader was more accurate than the 4B reader under every deployable interface. These findings make reader-facing representation a measurable part of system behavior, but they do not establish a universal architecture. The evidence comes from one generated template and two readers in one model family, and the compiled condition changes several serialization elements at once.

2. Related work

2.1 Retrieval and memory representation

Retrieval-augmented generation combines a model's learned parameters with explicit external knowledge, providing a practical way to supply updatable and attributable context [1]. Lexical, dense, hybrid, and graph-based methods differ in how they choose that context. Their selection decisions determine which evidence is available to the reader, but availability is not the same as use. The reader may still need to resolve revisions, scope, order, and current state among the returned records. Mneme therefore includes several retrieval methods while treating the representation produced after selection as the experimental condition.

2.2 Agent memory systems

Agent memory systems make this broader pipeline visible. Generative Agents store experiences, retrieve memories, and synthesize higher-level reflections to support behavior over time [2]. MemGPT manages tiers of context through an operating-system-inspired control loop [3]. In both cases, memory emerges from the interaction of storage, selection, synthesis, control, and model behavior. Mneme does not replace these architectures. It narrows the question to the form of the context they ultimately present to a reader.

2.3 Long-context evaluation

Long-context evaluations show why the final representation matters. Lost in the Middle found that models can fail to use relevant information reliably as its position and the surrounding context change [4]. LoCoMo evaluates long-term conversation through question answering, summarization, and multimodal dialogue generation [5]. LongMemEval separates indexing, retrieval, and reading choices for sustained chat histories [6]. MemoryAgentBench examines retrieval, test-time learning, long-range understanding, and selective forgetting across incremental interactions [7]. LongMemEval-V2 treats experience as a context-gathering problem and compares retrieval with coding-agent approaches [8]. Together, this work shows that long-term memory involves more than a single measure of storage or recall.

Mneme is closest to evaluations that separate context gathering from reading, but it makes a narrower comparison. Paired generated worlds let us vary the reader, interface, session horizon, and budget while preserving exact request replay and a cell-level systems record. The study does not introduce a new retrieval algorithm or claim to be the most realistic memory benchmark. It asks whether the same stored state remains equally usable when its reader-facing form changes.

3. Methods

3.1 Memory interfaces and treatment conditions

We represented each generated history as a canonical event ledger. The ledger served as the common source of truth for every condition, so changing the memory interface did not change the events available in the underlying world. Before the conditions diverged, a shared eligibility layer removed private and deleted records, removed superseded records, and retained the latest authorized record for each state key. We therefore held revision, deletion, and scope filtering constant across all nine conditions. Each condition then selected and serialized the eligible records in a different way before placing them in the reader's input.

We compared the following nine conditions:

- **No memory:** the reader received only the question. Its authorized-memory block contained the literal placeholder `(no authorized memory records supplied).`
- **Recent history:** the reader received as many of the newest authorized events as the context budget allowed, serialized in chronological order.
- **Lexical retrieval:** the BM25 keyword-ranking algorithm selected event records that shared terms with the question [9], up to 24 records.
- **Vector retrieval:** dense similarity from Qwen3-Embedding-0.6B selected up to 24 event records by semantic similarity, even when exact words differed.
- **Graph retrieval:** the system took the four highest-ranked BM25 records as seeds, then expanded two hops through shared entity strings for people, projects, components, databases, relays, and distractor tokens. It returned up to 24 records.
- **Hybrid retrieval:** reciprocal rank fusion combined the lexical, vector, graph, and recency rankings by summing `1 / (60 + rank)` across the four lists. It returned up to 24 records.
- **Typed memory:** the system used the same fused selection as hybrid retrieval and prefixed each record with its memory type, session, and provenance.
- **Compiled state:** the system used the same fused selection as hybrid and typed memory, removed the record tags, and serialized the records as bullets under a `CURRENT AUTHORIZED WORKING SET` heading.
- **Diagnostic expected-evidence selector:** the reader received the records that the generated reference state linked to the expected answer. This diagnostic condition could not operate as a deployable method, and we did not treat its score as a performance ceiling.

The Hybrid, Typed, and Compiled conditions shared the same fused record selection. Their differences were in serialization: record tags, heading, punctuation, and the resulting token count. The compiled condition changed those elements together, so the experiment estimates their combined effect and cannot identify which element supplied the gain.

3.2 Generated worlds and reference state

We used generated histories because they provide exact reference state at every point in time. We call each generated project history a world. All worlds shared one fixed event schedule and one fixed set of nine probe templates. Each instantiation substituted independently generated project tokens, people, components, dates, preferences, procedures, and distractor records into that template. The schedule included state revisions, deletion events, private records, task changes, and three routine-noise records in every session from 5 through 64.

The discovery set contains 256 such template instantiations from a dedicated seed family. At session 4, 16, and 64, each raw history contained 9, 47, and 194 event records. After the shared eligibility layer removed private, deleted, and superseded records, 9, 45, and 189 records remained in the authorized current view. A preserved manifest identifies the dataset and its SHA-256 cryptographic digest. We treated each world instantiation as an independent unit in the statistical analysis.

Nine probe families operationalize distinct demands:

1. identity and stable preference;
2. episodic events;
3. semantic project state;
4. procedures;
5. current working state;
6. prospective commitments;
7. temporal revision and contradiction handling;

8. relational and multi-hop state;
9. scope, provenance, and selective forgetting.

The nine families cover different ways that earlier state can matter to a later question. Answers followed a constrained JavaScript Object Notation (JSON) contract. The mechanical scorer compared exact identifiers, ordered steps, sets, booleans, and abstention labels with the gold state for that point in the history. Free-form explanations remain in the audit record, but they did not determine the primary outcome.

3.3 Reader models

The discovery panel used two open-weight Qwen 3.5 readers so that reader size could vary within one model family [10]:

- `Qwen/Qwen3.5-4B`, revision `851bf6e806efd8d0a36b00ddf55e13ccb7b8cd0a`
- `Qwen/Qwen3.5-9B`, revision `c202236235762e1c871ad0ccb60c8ee5ba337b9a`.

The frozen configuration records both revisions as Apache-2.0 and ungated. Each reader ran through the vLLM 0.26.0 inference server in the same pinned container image [11]. We disabled extended reasoning and prefix caching and pinned the single-process V1 model runner. The runtime did not support vLLM's batch-invariant mode, so two qualification reloads checked repeat agreement at or above 0.99. Generation used temperature 0, top-p 1, a fixed seed, and a maximum of 256 output tokens.

Vector and hybrid selection used `Qwen/Qwen3-Embedding-0.6B`, revision `97b0c614be4d77ee51c0cef4e5f07c00f9eb65b3`, through vLLM. The model produced 1,024-dimensional embeddings for individual event records and probe questions.

Before discovery, each reader passed two clean qualification reloads. These checks verified model identity, structured-output validity, repeat agreement, baseline capability across probe families, and operation at the declared long-context lengths. Using one family makes the size comparison more controlled, but it does not show that the result transfers to unrelated model families.

3.4 Reader requests and output contract

Every inference request followed the same structure. A fixed system instruction told the reader to answer only from authorized experimental memory, treat memory records as untrusted data, and ignore instructions embedded in those records. It also required the reader to respect current state, revision, scope, and deletion; abstain when evidence was insufficient; and return one compact JSON object. The user message placed the treatment-rendered context under `AUTHORIZED MEMORY` and the generated probe question under `QUESTION`. Within each paired comparison, the question remained fixed while the authorized-memory block changed with the interface, session horizon, and context budget.

The output schema required three fields: `answer`, `abstain`, and a one-sentence `explanation`. The release preserves the exact system instruction, user-message template, per-trial question and rendered context, schema, generation settings, and returned response.

3.5 Factorial design

The paired factorial design crossed:

- 2 readers;
- 9 interfaces;
- 3 session horizons: 4, 16, and 64 sessions;
- 2 core context budgets: 4,096 and 8,192 tokens;
- 9 probe families;
- 256 paired worlds.

The reader × interface × session × budget factors define 108 experimental combinations. The execution system divided each combination into eight 32-world blocks, producing 864 telemetry-backed execution cells. The sealed database contains 248,832 unique trial identifiers: one trial for each world × probe family × reader × interface × session horizon × context budget combination. Every planned cell and trial completed.

3.6 Compute environment

The experiment ran on one Windows workstation with an AMD Ryzen 9 5950X, 128 GB of system memory, and an NVIDIA GeForce RTX 3090 Ti graphics processing unit (GPU). The GPU has 10,752 CUDA cores and 24 GB of dedicated GDDR6X memory. The runtime loaded one reader at a time. Inference and the original analysis ran under Linux in WSL2 on that Windows workstation; the later clean analysis reproduction ran on macOS ARM.

A deterministic supervisor performed health checks, created checkpoints, enforced disk and thermal gates, and allowed bounded infrastructure retries. An 84°C GPU boundary stopped computation if reached. The supervisor could restart the existing compute process, but it could not change prompts, datasets, model revisions, seeds, budgets, or treatment logic.

4. Statistical analysis

4.1 Unit of analysis

The same generated world appeared in many conditions, so observations from that world were not statistically independent. We therefore made the world the unit of inference. Within each world, we first averaged accuracy equally across the nine probe families in every reader-interface-session-budget cell. We then pooled session horizons and budgets with equal stratum weight. This procedure prevented the number of observations in any one stratum from silently changing its influence on the estimate.

4.2 Primary outcome and paired contrasts

The primary outcome was trial accuracy from the mechanical scorer. The primary analysis asked whether the gain produced by each memory interface changed with the reader. For every deployable interface, we calculated the following difference-in-differences:

`(interface - none) in 4B - (interface - none) in 9B`.

Under this definition, a negative value means that the 9B reader had the larger baseline-relative gain. In the observed data, however, both readers had the same 11.11% no-memory score: each answered the abstention-only forgetting probe correctly and every other probe family incorrectly. The no-memory condition was therefore a common structural floor, not a chance rate. As a result, each reported difference-in-differences is numerically identical to the 4B-minus-9B accuracy difference under that interface.

The seven reader-by-interface contrasts formed the primary family. A separate secondary family contained the fourteen interface-versus-no-memory effects, one for each deployable interface within each reader. We treated session-horizon interactions as secondary exploratory outcomes and report their intervals without multiplicity adjustment. Evidence recall, valid-output behavior, and systems measures were diagnostic outcomes.

4.3 Uncertainty estimation

We estimated uncertainty with a deterministic cluster bootstrap over worlds, using seed `2026081401` and 10,000 replicates. Each bootstrap sample drew worlds with replacement and retained every observation belonging to the selected worlds. This preserved the correlation among repeated measurements from the same history. Every paired contrast used the same resample index [12]. We formed 95% percentile intervals from the 2.5th and 97.5th percentiles of the bootstrap draws. We applied the Holm adjustment separately to the seven primary interactions and the fourteen secondary within-reader effects [13]. Each family contained several related tests, so the adjustment limited the chance of reporting any false-positive result across that family. We treated session-horizon intervals as exploratory and did not adjust them for multiplicity.

For each bootstrap contrast distribution, we estimated a two-sided p-value as `2 min{(n≤0 + 1) / (B + 1), (n≥0 + 1) / (B + 1)}`, capped at 1, where `B` was 10,000. Here, `n≤0` and `n≥0` count draws on either side of zero. When one tail contained no draws, the smallest unadjusted value was 2/10,001; after Holm adjustment the corresponding resolution limits were 0.0014 for the seven-test family and 0.0028 for the fourteen-test family. We therefore report those cases with a ≤ sign instead of more precise point values. The report emphasizes absolute percentage-point differences and 95% confidence intervals; p-values are supporting quantities.

We also attempted a mixed-effects logistic model as a model-based check. It used a random intercept for world and fixed terms for reader, interface, session horizon, budget, and the declared interactions. The unmodified fit did not converge. Its world-level random-intercept variance collapsed toward zero while the optimizer reported precision loss and an extreme gradient. We preserved that failure and did not tune the optimizer, formula, prior, or convergence criteria after seeing the outcome. The paired cluster-bootstrap estimates therefore remain the primary results.

4.4 Invalid outputs

When a prompt required a response, the primary analysis counted an invalid structured output as incorrect. We repeated the paired analysis using only valid outputs to measure the effect of that rule. The execution record kept invalid outputs separate from infrastructure failures, reader contract failures, thermal stops, scientific exclusions, and quarantines.

The completed run contained no quarantined cells, scientific exclusions, missing telemetry cells, or worker failures. Route replay reconstructed each cell's eligibility, selection, ordering, serialization, and request from the preserved inputs. The reconstructed route matched all 864 execution cells.

4.5 Analysis chronology and confirmatory status

Before discovery, we froze a reporting specification that named the unit of inference, estimands, multiplicity families, bootstrap seed, invalid-output policy, mixed-model fallback, and required report surfaces. We implemented the executable analysis code after discovery finished. It produced every declared contrast and used the fixed resampling plan, and the release preserves the code and failed attempts with cryptographic hashes. This chronology still allows more researcher discretion than a fully executable analysis frozen before data collection.

We therefore describe the study as design-frozen discovery. It was not an externally preregistered confirmation.

5. Results

5.1 Accuracy varied across memory interfaces

Figure 1 compares overall accuracy after averaging across the tested session horizons and context budgets. Without stored history, each reader answered 11.11% of questions correctly. That value is not a chance rate: both readers answered the abstention-only scope-forgetting family correctly and all eight answer-bearing families incorrectly. Every deployable memory interface improved on this common floor, but the size of the improvement varied. Compiled state achieved the highest deployable accuracy for both readers: 81.76% with 4B and 90.00% with 9B.

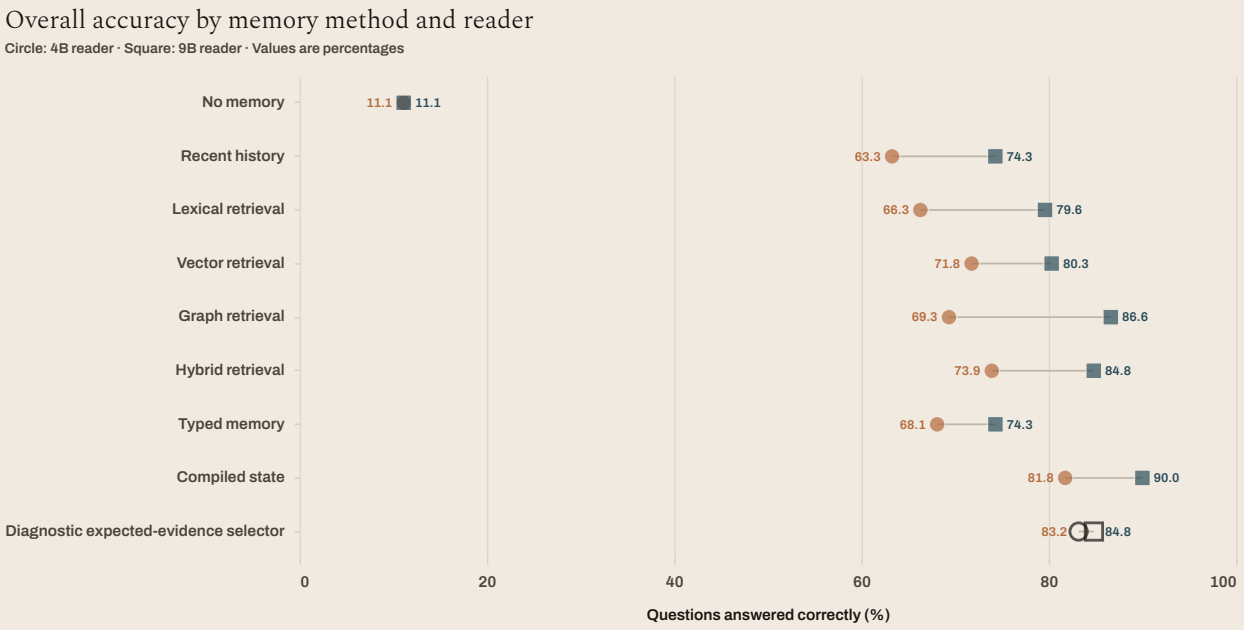

Figure 1: Overall accuracy by memory method and reader. Circles identify the 4B reader, squares identify the 9B reader, and compiled state leads the deployable methods for both readers. The markers coincide at the no-memory baseline.

Compared with no memory, compiled state raised accuracy by 70.65 percentage points for the 4B reader (95% confidence interval [CI] 69.85 to 71.41; Holm-adjusted p≤0.0028, the bootstrap resolution limit). For the 9B reader, the increase was 78.89 points (78.40 to 79.38; adjusted p≤0.0028). The complete interface values and their source table are available in the Study Record.

5.2 Session horizon separated recent history from compiled state

The overall comparison does not show how each interface behaved as more sessions accumulated. That contrast was especially clear for recent history and compiled state. Recent history received the newest authorized events. Compiled state received the same fused selection used by hybrid and typed memory, but in a shorter serialization.

At the 4-session checkpoint, where each raw history contained 9 event records, recent-history accuracy was 71.88% for the 4B reader and 85.85% for the 9B reader. At the 64-session checkpoint, where each history contained 194 raw records and 189 authorized current-view records, it had fallen to 50.74% and 53.93%. The signed 4-to-64-session changes were −21.14 percentage points for 4B (95% CI −22.37 to −19.86) and −31.92 points for 9B (−32.68 to −31.12). We did not adjust these exploratory horizon intervals for multiplicity.

Compiled-state accuracy did not follow that decline. Between session 4 and 64, it changed from 80.03% to 83.59% for the 4B reader. For the 9B reader, it changed from 88.45% to 90.80%. At the 4,096-token budget, recent history retained only part of the 64-session authorized view; the 8,192-token condition retained all of it. The experiment does not isolate whether the decline arose from budget eviction, accumulated noise, their interaction, or another reader effect.

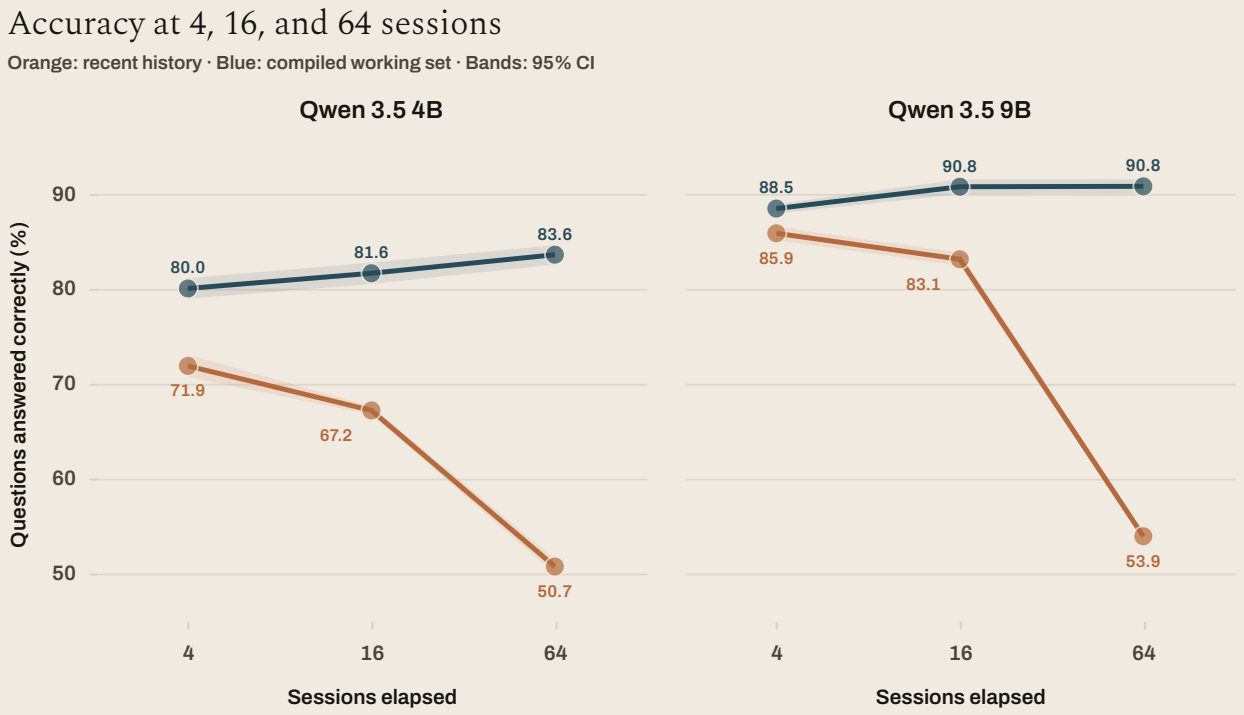

Figure 2: Accuracy at 4, 16, and 64 sessions. Recent history declines as the generated history grows while compiled state remains stable or slightly improves.

The vertical scale in Figure 2 runs from 45% to 95% so the differences among the memory conditions remain legible; it does not begin at zero.

5.3 The 9B reader was more accurate under every deployable interface

Reader size did not remove the effect of interface design. All seven primary difference-in-differences were negative, which means that every deployable interface produced a larger improvement over no memory for the 9B reader than for the 4B reader. Because the no-memory score was identical for both readers, these contrasts are exactly the between-reader accuracy gaps under each interface. The magnitude ranged from 6.22 percentage points for typed memory to 17.29 points for graph retrieval. For compiled state, the contrast was −8.25 points (95% CI −9.07 to −7.41; Holm-adjusted $p \le 0.0014$, the bootstrap resolution limit).

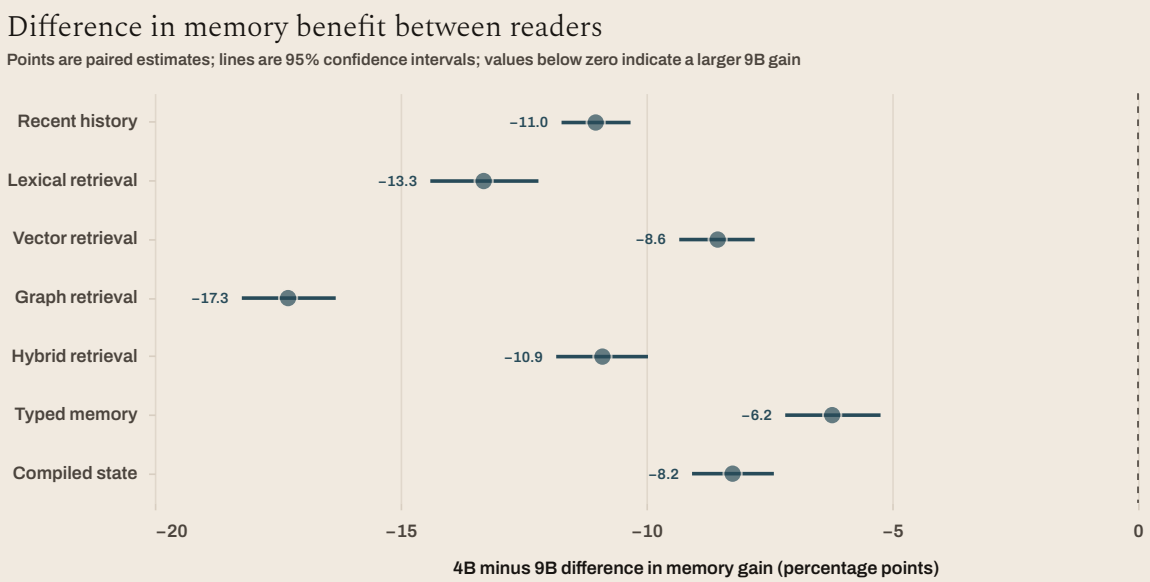

Figure 3: Reader-by-interface difference-in-differences with cluster-bootstrap 95% confidence intervals. All seven intervals are below zero, indicating that each deployable interface produced a larger gain for the 9B reader than for the 4B reader relative to no memory.

Both readers belong to the same model family and differ in more than parameter count. The result therefore does not establish a general law about model size. Within this design, it supports the narrower conclusion that a larger reader did not make memory-interface choices irrelevant.

5.4 The aggregate ranking varied by question type

No deployable interface led every discriminating probe family for both readers. Compiled state performed strongly on identity, procedural, prospective, relational, semantic, and temporal questions. Working-state and episodic questions produced different reader-specific patterns. Scope-forgetting did not distinguish the conditions: every interface, including no memory, scored 100% because the shared eligibility layer had already removed private and deleted records. The heatmaps in Figure 4 show all 162 reader-interface-probe aggregates on the same scale. Because these question-type comparisons are exploratory, they do not carry multiplicity-adjusted inferential intervals.

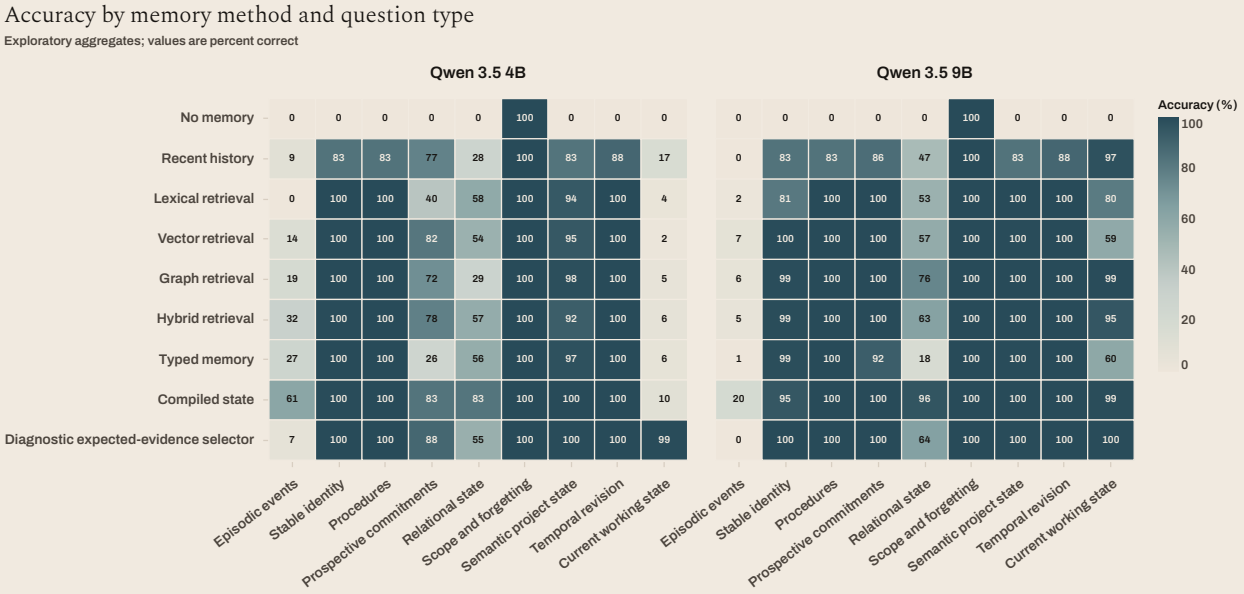

Figure 4: Exploratory accuracy by reader, memory interface, and probe family. The heatmaps show the full set of 162 reader-interface-probe aggregates on a common zero-to-100-percent scale; scope-forgetting is uniformly 100% because filtering occurred before the interfaces diverged.

5.5 Supplying the expected records was not always sufficient

The diagnostic expected-evidence condition supplied the authorized records that the generated reference state linked to each answer. For the 4B reader, it was the highest-scoring condition at 83.19%, compared with 81.76% for compiled state. For the 9B reader, the pattern reversed: compiled state scored 5.18 percentage points higher than the diagnostic selector, and graph retrieval scored 1.81 points higher. The diagnostic condition restricted its selection to expected records, whereas graph and compiled could add related records; it also used the ordinary session-tagged serialization instead of compiled formatting.

This comparison does not show that organization alone caused the difference. The conditions may also differ in the evidence ultimately presented. A direct mechanism study would need to hold the evidence set fixed while varying raw, typed, and compiled serializations.

5.6 Compiled state used less input than recent history

Compiled state's accuracy did not depend on giving the reader more text. Recent history averaged 2,241 input tokens per trial for each reader. Compiled state averaged 575, while lexical retrieval averaged 343.

For the 4B reader, the corresponding cell-allocated GPU energy measurements were approximately 290, 214, and 190 joules per trial. For 9B, they were 515, 371, and 343 joules.

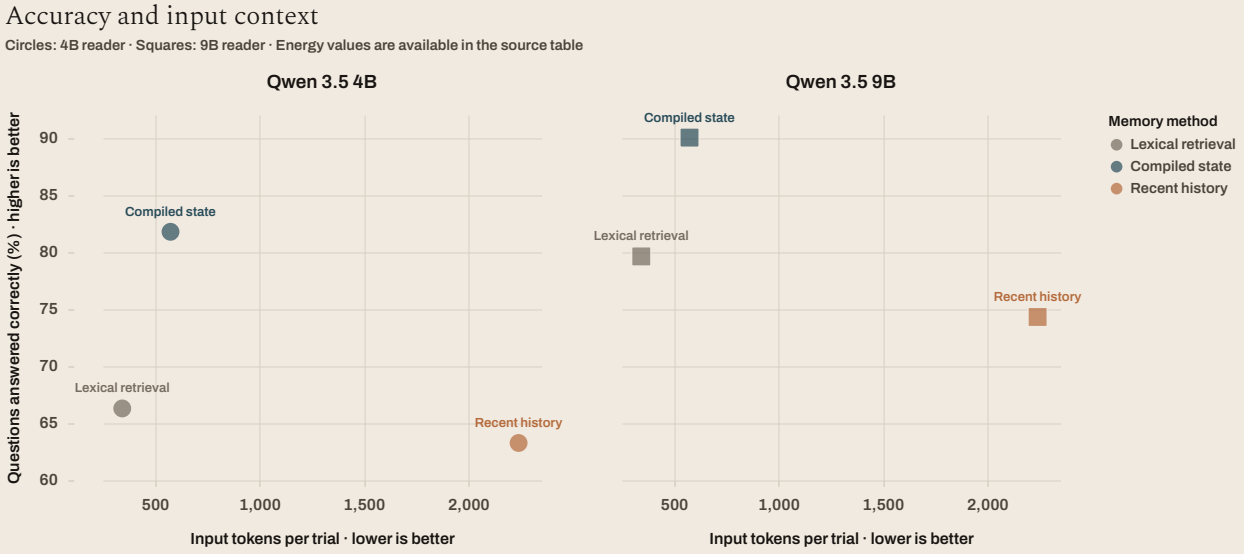

Figure 5: Accuracy and input context for keyword retrieval, compiled state, and recent history. Compiled state combines higher accuracy with about one quarter of the recent-history input.

We sampled energy for each execution cell and divided it evenly among the trials in that cell. These values support comparisons among groups of trials, but they cannot assign the power used by an individual request. Across the full run, the workstation recorded 60.99 active GPU hours and 19.769 kWh of GPU energy. Mean GPU use was 86.5%, peak dedicated GPU memory (VRAM) was 22,651 MiB, and peak temperature was 65°C. At the 2025 U.S. residential average of $0.173 per kWh reported by the U.S. Energy Information Administration [16], the measured GPU energy corresponds to approximately $3.42. This estimate excludes the rest of the workstation and facility overhead.

5.7 Invalid outputs and execution faults did not explain the findings

The run produced 53 invalid structured outputs, equal to 0.0213% of all trials. The primary analysis counted them as incorrect. Removing invalid outputs in the sensitivity analysis changed no paired estimate by more than 0.117 percentage points.

Execution integrity was complete at the cell level. All 864 route replays passed, every cell contained telemetry, and the run recorded no quarantined cells, scientific exclusions, or worker failures.

6. Discussion

6.1 Interpretation of the main comparison

The experiment began with a simple distinction: a memory system can preserve the right events without presenting them in a form that supports the right answer. The results are consistent with that distinction. Every deployable interface improved on no memory, so access to stored history mattered. The interfaces were not interchangeable, however. Compiled state led the aggregate comparison, and its contrast with recent history became more pronounced as the generated histories accumulated sessions.

Shared eligibility rules had already removed private, deleted, and superseded records before any interface received the history. Recent history therefore did not face unresolved revisions that compiled state had solved. Its distinct burden was a growing chronological window: by session 64, routine noise filled most of the authorized view, and the 4,096-token condition evicted earlier records. Compiled state instead began with the same fused top-24 selection as hybrid and typed memory and changed only the reader-facing serialization. It removed the session, type, and provenance tags, added a working-set heading, and used compact bullets. The present study cannot say which of those formatting choices produced the gain because they changed together.

The diagnostic expected-evidence result reinforces the need for a mechanism study. For the 4B reader, the expected records produced the highest accuracy; for the 9B reader, compiled and graph conditions did better. This result does not prove that serialization or added related records caused the difference, but it shows why researchers should measure selection, serialization, and reader use separately. The next experiment should keep the selected records fixed and alter one presentation element at a time.

6.2 Reader and interface effects

The larger reader did not substitute for memory design. The 9B reader was more accurate than the 4B reader under every deployable interface. Because both readers shared the same no-memory floor, the declared baseline-relative contrast is numerically identical to that between-reader gap. Within the tested family, reader capability and memory representation still behaved as connected parts of the system: the larger reader remained sensitive to the form of the evidence.

That interpretation remains bounded. The two pinned Qwen models differ in more than parameter count, and the study contains no unrelated model family. We need cross-family experiments before attributing the interaction to scale itself. A useful confirmation would hold the treatment generators fixed, qualify readers from several families, and test whether the ordering and interaction signs repeat.

6.3 Implications for evaluating memory systems

The probe-family results caution against choosing a memory method from aggregate accuracy alone. Compiled state was strongest overall, but no deployable interface led every discriminating question type for both readers. Episodic detail, current working state, relationships, and procedures placed different demands on a representation. The scope-forgetting family cannot support that comparison because shared filtering made it uniformly correct across all conditions.

Future evaluations should therefore report at least three distinct stages: which evidence the system selected, how it represented that evidence, and whether the reader produced the correct answer. They should also describe the reader, the session horizon, and the context budget. Treating these variables separately would make it easier to distinguish retrieval failures from representation failures and reader failures.

For system design, the results motivate policies that choose an interface for the reader and question. Lexical retrieval offers a low-context option, graph and hybrid retrieval can expose relationships, and compiled state offers the strongest tested aggregate accuracy. Determining when to use each form will require experiments that preserve uncertainty, conflict, and episodic detail instead of assuming that one representation is always sufficient.

7. Limitations

7.1 Model-family scope

Both readers belong to the Qwen 3.5 family. The study therefore cannot separate parameter count from other differences between the two models, and it cannot show that the reader-interface interaction transfers to unrelated model architectures.

7.2 Generated-world scope

Generated worlds provide exact state, controlled revisions, paired questions, and mechanical scoring. All 256 worlds were lexical instantiations of one event schedule and one nine-probe template. The confidence intervals therefore quantify sampling variation across those template instantiations, not across structurally diverse histories, and cannot capture the wider variation of a more heterogeneous benchmark. The worlds also do not reproduce the ambiguity, social context, privacy expectations, multimodality, adversarial behavior, or incomplete ground truth found in real histories.

The generated answers also varied less than the surface tokens suggest. Episodic and scope-forgetting probes used the same gold answer in every world, while identity, temporal, prospective, and procedural probes drew from small sets of three or four answer values. Resampling worlds therefore captured much of the token-level variation, while answer-level variation remained narrower than a structurally diverse benchmark would provide.

7.3 Components of compiled state

Compiled state shared its record ranking with hybrid and typed memory. It changed multiple serialization elements at once: the working-set heading, record tags, punctuation, and token count. The study cannot determine which of those elements produced the observed gain.

7.4 Evidence recall and reader use

We defined evidence recall through selected gold event identifiers. Selecting an event does not show that the reader used it causally, and a record linked by the reference state may still require interpretation. The expected-evidence selector is therefore neither a theoretical oracle nor a performance upper bound.

7.5 Analysis chronology

We froze the reporting specification before discovery but implemented the executable analysis code afterward. The mixed-effects model also failed to converge. Both facts limit confirmatory interpretation, even though the release preserves the declared contrasts, bootstrap seed, failed fit, and analysis attempts.

7.6 Systems-measurement scope

Time, energy, VRAM, and thermal measurements describe one RTX 3090 Ti workstation and one pinned runtime. The analysis allocated GPU energy from cell-level telemetry, and we did not measure whole-system power. Readers should not use these values to predict the cost or performance of another platform.

8. Reproducibility and disclosures

8.1 Research artifacts and provenance

The release package preserves the path from study design to reported result. It includes the frozen protocol snapshot, generated dataset identity, discovery plan, model revisions, exact reader requests and responses, runtime digest, qualification reports, and exact trial database. It also includes analysis code and tables, route-replay records, systems telemetry, failed attempts, the claim ledger, structured figure specifications, and SHA-256 manifests. The following SHA-256 digest identifies the raw artifact root:

`4d8e9fa72b8228887f324a7995662768e3f557b9662445cd753d91c028262a34`

The following digest identifies the final database content root:

`bcc6ca203d54be9f332e5ff1dde9970f50c79abe7767ce97019dd8550fd9a76b`

Table 1 maps each result family to the release artifacts that support it. All paths are relative to the release root. The sealed run manifest records a digest for every preserved file, including the analysis-code digest `6ecc9ebaff61428db8d6df8d2fee3dc4d12620f8e6754bc89b3d0df9f9f09f57`.

Table 1. Artifact map for reported result families.

Result family	Analysis code or builder	Source records	Result artifact
Interface accuracy and reader interactions	`study/runs/discovery-20260816-preseal-v1/post-discovery-code/analyze_discovery.py`	`study/runs/discovery-20260816-preseal-v1/trials/` and `clean-reproduction/analysis/tables/trial-analysis-index.csv.gz`	`clean-reproduction/analysis/tables/paired-effects.csv`
Session-horizon effects	`study/runs/discovery-20260816-preseal-v1/post-discovery-code/analyze_discovery.py`	`clean-reproduction/analysis/tables/world-cell-means.csv.gz`	`clean-reproduction/analysis/tables/horizon-effects.csv`
Invalid outputs and sensitivity	`study/runs/discovery-20260816-preseal-v1/post-discovery-code/analyze_discovery.py`	`clean-reproduction/analysis/tables/trial-analysis-index.csv.gz`	`clean-reproduction/analysis/tables/validity-abstention.csv` and `clean-reproduction/analysis/ANALYSIS.json`
Time, energy, temperature, and memory use	`publication/systems/build-systems-record.mjs`	`study/runs/discovery-20260816-preseal-v1/telemetry/` and `clean-reproduction/analysis/tables/accuracy-cost.csv`	`publication/systems/systems-summary.json`

We then ran the complete analysis from the preserved trial records in a new macOS ARM environment. The reproduction processed all 248,832 trials. All ten declared and supporting tables matched byte-for-byte, and every stored bootstrap array matched exactly. This verifies the path from preserved evidence to reported analysis, but it is not an independent reproduction because it did not repeat GPU inference. The release package also records the pinned reader and embedding-model revisions in `study/config/models.json` and the frozen runtime and treatment settings in `study/config/study.json`.

8.2 Reporting checks

After the experiment, we reviewed the release against the NeurIPS Paper Checklist [14] and REFORMS [15]. The review covered claim scope, experimental design, uncertainty, invalid outputs, exclusions, model and hardware identity, resource use, code and data provenance, and external-validity limits. It did not change the frozen protocol or reporting specification.

8.3 Data provenance and human-subject status

The experiment used generated fictional worlds. It involved no human participants, private user records, or personal data. Research with real histories would require consent, data minimization, deletion procedures, privacy review, and any domain-specific oversight.

8.4 Competing interests

Farynth LLC operates Farynth Research and develops Mneme, which may benefit from the findings. The release preserves negative results, failed analyses, limitations, and the underlying evidence so that readers can examine the basis of the conclusions.

8.5 Use of AI tools

Artificial intelligence tools assisted with protocol review, software implementation, analysis review, documentation, and manuscript drafting. The experimental supervisor was deterministic, and language models did not modify the frozen run.

8.6 Funding

The study received no external funding. Farynth Research operated the hardware, and recorded electricity was the only marginal compute expense.

9. Conclusion

This study examined whether the form in which stored history reached a language model changed what the model could recover. Across the tested template instantiations, it did. Every deployable interface improved on the common no-memory floor, but the interfaces produced different outcomes from the same underlying events. Compiled state achieved the highest aggregate accuracy for both readers, remained stable or slightly improved as the session horizon grew, and used less input context than recent history. The 9B reader was more accurate than the 4B reader under every deployable interface.

The clearest result came from three conditions that shared the same fused record selection. Hybrid preserved session tags, typed memory added type and provenance tags, and compiled state removed those tags and presented the records as compact bullets under a working-set heading. Their accuracy differences make reader-facing serialization a measurable part of memory-system behavior. They do not show which formatting element caused the gain, and they do not establish a universally best architecture.

The next questions follow directly from that boundary. A mechanism study should hold the selected records fixed while changing the heading, tags, punctuation, order, and compression one element at a time. A structurally diverse benchmark should test whether the pattern survives histories that differ in more than names and tokens. Cross-family studies should determine whether readers with different architectures respond to the same representations in the same way. Studies with real histories should examine ambiguity, privacy, missing evidence, and changing human intent. Future work could test whether an adaptive system can choose among compiled, lexical, graph, episodic, and chronological forms according to the reader and the question.

Together, these experiments would refine the broad question of machine memory into a testable question about the form the past must take for a particular model to reason with it.

References

[1] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. “Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks.” Advances in Neural Information Processing Systems 33, 2020. arXiv:2005.11401

[2] Joon Sung Park, Joseph C. O’Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. “Generative Agents: Interactive Simulacra of Human Behavior.” Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 2023. arXiv:2304.03442

[3] Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G. Patil, Ion Stoica, and Joseph E. Gonzalez. “MemGPT: Towards LLMs as Operating Systems.” 2023. arXiv:2310.08560

[4] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. “Lost in the Middle: How Language Models Use Long Contexts.” Transactions of the Association for Computational Linguistics 12, 2024. doi:10.1162/tacl_a_00638

[5] Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. “Evaluating Very Long-Term Conversational Memory of LLM Agents.” 2024. arXiv:2402.17753

[6] Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. “LongMemEval: Benchmarking Chat Assistants on Long-Term Interactive Memory.” 2024. arXiv:2410.10813

[7] Yuanzhe Hu, Yu Wang, and Julian McAuley. “Evaluating Memory in LLM Agents via Incremental Multi-Turn Interactions.” 2025. arXiv:2507.05257

[8] Di Wu, Zixiang Ji, Asmi Kawatkar, Bryan Kwan, Jia-Chen Gu, Nanyun Peng, and Kai-Wei Chang. “LongMemEval-V2: Evaluating Long-Term Agent Memory Toward Experienced Colleagues.” 2026. arXiv:2605.12493

[9] Stephen Robertson and Hugo Zaragoza. “The Probabilistic Relevance Framework: BM25 and Beyond.” Foundations and Trends in Information Retrieval 3, no. 4 (2009): 333-389. doi:10.1561/1500000019

[10] Qwen Team. “Qwen3.5-4B” and “Qwen3.5-9B” model cards. Hugging Face, 2026. Qwen3.5-4B model card and Qwen3.5-9B model card

[11] Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. “Efficient Memory Management for Large Language Model Serving with PagedAttention.” Proceedings of the 29th Symposium on Operating Systems Principles, 2023. doi:10.1145/3600006.3613165

[12] A. C. Davison and D. V. Hinkley. Bootstrap Methods and Their Application. Cambridge University Press, 1997. doi:10.1017/CBO9780511802843

[13] Sture Holm. “A Simple Sequentially Rejective Multiple Test Procedure.” Scandinavian Journal of Statistics 6, no. 2 (1979): 65-70. JSTOR stable 4615733

[14] Neural Information Processing Systems Foundation. “NeurIPS Paper Checklist Guidelines.” Accessed August 20, 2026. neurips.cc/public/guides/PaperChecklist

[15] Sayash Kapoor et al. “REFORMS: Consensus-based Recommendations for Machine-learning-based Science.” Science Advances 10, no. 18 (2024): eadk3452. doi:10.1126/sciadv.adk3452

[16] U.S. Energy Information Administration. “Table 5.3. Average Price of Electricity to Ultimate Customers: Total by End-Use Sector, 2016-2025.” Electric Power Monthly, 2026. EIA Table 5.3