PRISM: PROGRESSIVE RESOLUTION WITH INDEPENDENT SECTION MAPPING FOR SCALABLE USER PROFILING

Anonymous authors
Paper under double-blind review

ABSTRACT

Personalizing a language model over a user's long interaction history usually means either injecting the raw history into every prompt or compressing it once into a static persona. Both scale poorly: raw injection spends input tokens linear in history length, and a static persona re-generates every attribute for every query. We instead model the profile as an *infinite memory*: a hierarchy in which short-horizon *episodic* memories are compressed from narrow raw windows, and long-horizon memories are compressed *from those episodic memories* rather than from the raw history—so the input to any level is bounded by the level below it, and the memory absorbs unbounded history without ever feeding it all into a single prompt. We realize this memory with **PRISM** (Progressive Resolution with Independent Section Mapping), combining (i) incremental hierarchical compression of episodic into long-term memory, and (ii) a Map–Reduce decomposition that generates each memory level's independent semantic attribute sections in parallel, so the autoregressive generation depth is set by the longest section rather than the sum, and a broken section regenerates alone. Every efficiency result is reported in *tokens*, independent of any specific hardware. PRISM keeps the input tokens to *refresh* memory flat (6,343 tokens) as history grows, yielding a $\sim$63$\times$ input reduction at 10,000 events over monolithic regeneration, and cuts autoregressive generation depth by $12.9\times$ at $K{=}10$ sections. Crucially, on a paired longitudinal cohort ($n{=}200$ accounts over 13-week histories), section decomposition significantly improves profile completeness ($\Delta{=}+0.384, p{<}0.001$) and specificity ($\Delta{=}+0.296, p{<}0.001$) under Holm-Bonferroni correction while preserving global consistency (4.86 vs. 4.84) and establishing rigorous non-inferiority in deterministic episodic fact retrieval ($p{<}0.0001$). On downstream tasks (LaMP), PRISM matches or exceeds monolithic prediction accuracy. We release anonymized code and the LongProfile benchmark for long-horizon personalization.

1 INTRODUCTION

Personalization systems that condition on a user's long interaction history today face a forced choice. Either they inject the raw history into every prompt, spending input tokens that grow linearly with history length until the context is exhausted; or they compress the history once into a static persona, which must then be re-generated in full whenever any part of it changes or a new interaction arrives. The first is expensive per query; the second is expensive per update and couples every downstream request to the entire identity. Neither is a property of long history itself—both are properties of the *representation* chosen to store it.

We argue that a user profile should be an *infinite memory*: instead of a flat persona rebuilt from scratch, a hierarchy in which recent interactions are compressed into short-horizon *episodic* memories, and long-horizon memories are compressed *from those episodic memories* rather than from the raw history. Because each level consumes the compressed level below it, the input to any level is bounded, and the memory keeps absorbing new episodes without ever re-reading the full history. We call this design **PRISM** (Progressive Resolution with Independent Section Mapping). Two orthogonal mechanisms do the work: *incremental hierarchical compression* (the infinite-memory axis)

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

and a *Map–Reduce* decomposition that generates each level's semantic attribute sections in parallel (the parallel-generation axis), so the autoregressive depth is set by the longest section and a broken section regenerates alone.

Contributions.

1. **Infinite memory for user profiles.** We show that a user profile can be stored as an incrementally-compressed hierarchy—short-horizon *episodic* memories fed forward into long-term memory—so the input tokens to build or refresh any level are bounded by the level below it, not by the raw history, and the memory absorbs unbounded history without re-reading it (Section 4).
2. **Parallel section generation (Map–Reduce).** PRISM generates the independent semantic attribute sections of each memory level in parallel and reconciles them with a Reduce step, so the autoregressive generation depth is set by the longest section, not the sum, and a broken section regenerates alone (Section 4).
3. **Empirical fidelity and grounding validation.** Across a paired longitudinal cohort of n=200 customer accounts over 13-week histories, we show that semantic section decomposition substantially improves completeness ($\Delta = +\ 0.384, p<0.001$) and specificity ($\Delta = +\ 0.296, p<0.001$) without sacrificing consistency or episodic fact retrieval ($p<0.0001$). Positional regression refutes the positional decay hypothesis, proving that gains stem from dedicated facet capacity (Section 5.4).
4. **A token-level, hardware-independent evaluation.** We measure input-token scaling across history lengths and autoregressive depth ratios across section counts on three public datasets (MovieLens-25M, Amazon Reviews 2023, Yelp) plus LaMP, reporting no latency or cost tied to specific hardware (Section 5).
5. **A benchmark for long-horizon user-profile personalization.** We release **LongProfile**, a reproducible, method-agnostic benchmark constructed from MovieLens-25M, Amazon Reviews 2023, and Yelp, with standardized long-horizon splits (50–5,000 interactions) and an evaluation protocol whose labels are held-out raw interactions scored with IR metrics (NDCG@k, Recall@k, MRR, MAE), not an LLM judge. The construction pipeline is released for public reproduction; raw source data is pulled from the original datasets.

We proceed as follows. Section 2 positions PRISM against prior work on memory-augmented LLM personalization and hierarchical retrieval. Section 3 formalizes the infinite-memory profile and disambiguates "monolithic" (single synthesized context) from a storage topology. Section 4 specifies PRISM's two orthogonal axes, the per-query assembly rule, the conflict-resolution policy, and the complexity/scaling properties. Section 5 reports the three probe families; Section 6 discusses when composition helps and where it does not; Section 7 names the open limitations; Section 8 concludes.

2 Related Work

PRISM sits at the intersection of three lines of work.

Memory-augmented LLM personalization. Our core idea—compressing episodic memories into a bounded long-term memory so the profile can absorb history without bound—descends from *infinite memory* systems such as PriorMem (Taelin, 2024), which compress episodic observations into a fixed-size long-term state via in-place updates. O-Mem (Zhong et al., 2024b) maintains an evolving persona memory updated after each user turn and feeds it as system context at inference; the persona is *monolithic* in our sense (one synthesized object, full regeneration on update). RUMS (Kumar et al., 2024) and Dynamic Profile Modeling (Wu et al., 2024) similarly materialize a single persona, refreshed on a schedule. LLMxMapReduce (Crew et al., 2024) performs *semantic* decomposition of long documents via a Map–Reduce over heterogeneous sections; PRISM adopts the same Map–Reduce discipline but on *user history* rather than documents and adds the incremental hierarchical compression axis that LLMxMapReduce does not address. PRIME (Li et al., 2024b) and PLUS (Li et al., 2024a) compose per-source memory modules; PRISM's schema is narrower (10+1 facets in our instantiation) but its incremental compression is finer.

Structured and dynamic memory systems. MemGPT (Packer et al., 2024) treats the context window as a tiered virtual memory (main context + external recall) and uses OS-style paging to swap memory in and out of the LLM's attention. MemoryBank (Zhong et al., 2024a) adds an Ebbinghaus forgetting curve on top of long-term memory with explicit decay. Persona-DB (Chen et al., 2024) and MemTree (Rebescher et al., 2024) organize memory hierarchically for retrieval. PRISM is a *structured-decomposition* method that does not page at inference time, contrasting with these *dynamic-retrieval* systems. Hierarchical RAG (Sarthi et al., 2024) shares the multi-resolution idea but applies it to document corpora, not user histories, and uses embedding retrieval rather than LLM-generated sections.

Positioning. MSC and Memora probe multi-session temporal evolution and forgetting-aware memory accuracy. PRISM is the first design to combine (a) *incremental hierarchical compression* of episodic memories into long-term memory (the infinite-memory axis), (b) a *Map–Reduce* decomposition that generates the semantic attribute sections of each level in parallel (the parallel-generation axis), and (c) a *per-query assembly* rule that reads only the sections the downstream task touches. The closest neighbours each cover one or two of these axes; we cover all three. Concurrent with our work, HORIZON (Goel et al., 2026) reformulates the same Amazon Reviews 2023 source into a long-horizon benchmark, but for *sequential recommendation* (cross-domain next-item prediction over 54M users). LongProfile instead scores *profile compression*—the refresh/depth token scaling and the retention axes that ranking benchmarks do not measure—so the two benchmarks are complementary, and our adoption of the same 2023 source is independently corroborated.

3 Problem Formulation

Notation. Let $H_u = (e_1, e_2, \ldots, e_n)$ denote a user's ordered interaction history with n events e_i (events are typed: rating, purchase, view, search, return, cart-abandon, etc., each with timestamp and structured metadata). A *user profile* is a function $\pi_u = f(H_u, \theta)$ mapping the history to a natural-language context prepended to every downstream personalization prompt.

Monolithic profile. A *monolithic* profile (our default baseline) is $\pi_u^{\text{mono}} = f^{\text{mono}}(H_u; \theta)$: a single teacher call that summarizes the entire history into one self-contained block. Throughout this paper, "monolithic" refers to the *single synthesized context* injected at inference, **not** the underlying storage topology (the storage can be a database; the synthesized context is one block).

Infinite-memory profile. A PRISM profile is a *compression hierarchy* of memory levels, not a flat persona. Recent interactions are compressed into short-horizon *episodic memories* M_1; each longer-horizon memory is compressed *from the level below* rather than from the raw history:

$$M_1 = f_1(H_u^{w_1}; \theta_1) \qquad \text{(episodic, window } w_1) \tag{1}$$

$$M_\ell = f_\ell(M_{\ell-1}; \theta_\ell) \quad \ell = 2, \ldots, L. \tag{2}$$

Because M_ℓ consumes the already-compressed $M_{\ell-1}$, the input to any level is bounded by $|M_{\ell-1}|$, independent of the raw history length. New episodes update the affected memories *in place* (incremental), so the memory absorbs unbounded history without ever re-reading it—the *infinite-memory* property.

Parallel section generation (Map–Reduce). Each memory level M_ℓ is itself decomposed into K independent semantic attribute sections, generated in parallel (Map) and reconciled by a single Reduce call:

$$M_\ell = \mathcal{R}_\ell\Big(\big\{g_{\ell,s}(M_{\ell-1}; \theta_{\ell,s})\big\}_{s \in \mathcal{S}}\Big), \tag{3}$$

where $\mathcal{S}$ is a *use-case-defined* attribute schema (not a fixed one—the same machinery re-targets to any schema), $g_{\ell,s}$ is one parallel teacher call per section, and $\mathcal{R}_\ell$ is the Reduce step. The autoregressive depth of level ℓ is therefore $\max_s O_{\ell,s}$ rather than $\sum_s O_{\ell,s}$; a broken section regenerates alone.

Read vs. write disambiguation. A profile is materialized at two distinct points: the *write* path, which incrementally updates the affected memories when new episodes arrive (content-hash keyed, in-place), and the *read* path, which is the per-query assembly $\mathcal{A}$ that selects the minimal section subset $\mathcal{S}_q \subseteq \mathcal{S}$ needed by query q and concatenates them in canonical order. The write-path cost is bounded by $|M_{\ell-1}|$ per updated level; the read-path cost is $\Theta(|\mathcal{S}_q|)$, constant in profile size for any fixed query subset.

4 METHOD

PRISM is a profile *representation*; it does not specify a single end-task. The same composed profile can be consumed by a recommendation prompt, a chat prompt, or a downstream reranker. We describe the representation through its two orthogonal mechanisms: the *incremental hierarchical compression* of episodic into long-term memory (Section 4.2) and the *Map–Reduce* parallel generation of each level's semantic sections (this section).

4.1 SEMANTIC SECTION DECOMPOSITION (MAP)

We instantiate a schema $\mathcal{S}$ of semantic attribute sections (10 in our reference taxonomy covering temporal cadence, brand affinity, budget, discovery, friction, category drift, item affinities, cohort footprint, recent context, plus a `coverage_gaps` meta-section; full schema in Appendix A.2). The schema is *use-case-defined*—the same Map machinery re-targets to any attribute set for a given product domain, requiring only targeted section prompts. Each section s has a dedicated system prompt (Appendix A.2) instructing the teacher to emit only facet s in 2–4 sentences of pure prose grounded strictly in the source atoms.

Each section call receives the *sliced* source atom block filtered to the tags that section s consumes (e.g., `temporal_patterns` reads `events` + `category_activity`; `friction_recovery` reads `ratings` + `returns` + `carts`). This is the **input slicing** rule: rather than naively re-reading the full atom block, slicing drops irrelevant tags so per-section input tokens *fall* rather than duplicate. Sections are generated by independent teacher calls in parallel (`concurrency` $= 8$ in our reference implementation).

4.2 INCREMENTAL HIERARCHICAL COMPRESSION (INFINITE MEMORY)

The temporal axis is a *compression hierarchy* over *interaction count*, not calendar time. We bucket by event count rather than by calendar windows because the histories we study are temporally sparse: Amazon 2023 reviews arrive at a median of 0.79 per month and Yelp at 1.2, so weekly/monthly buckets collapse to near-empty levels ($\sim$64% of active calendar months hold < 3 events, recomputed on the 2023 cohort). Raw events are grouped into fixed-size *episodes* of b interactions; each coarser level summarizes r episodes of the level below:

$$M_1 = f_1(H_u^{(b)}; \theta_1) \quad \text{(every } b \text{ events)} \tag{4}$$

$$M_2 = f_2(M_1^{(r)}; \theta_2) \quad \text{(every } r \text{ episodes)} \tag{5}$$

$$M_\ell = f_\ell(M_{\ell-1}^{(r)}; \theta_\ell) \quad \ell = 3, \ldots, L. \tag{6}$$

Because level M_ℓ consumes the already-compressed level $M_{\ell-1}$, its input size is bounded by $|M_{\ell-1}|$, independent of the raw history length. This is the **infinite-memory** property (C1): the memory keeps absorbing new episodes without ever re-reading the full history, and the input tokens to build or refresh any level are bounded by the level below it.

Compression is *incremental*: a new episode updates the affected memories in place rather than triggering a full re-derivation of the hierarchy. Long-horizon memories therefore accumulate *inferences* that only become visible across many episodes (household composition, price sensitivity, category drift) rather than merely re-stating recent events, while short-horizon memories retain the episodic "stories" a query about recent behaviour needs.

4.3 CROSS-SECTION CONSISTENCY (REDUCE)

The 10 parallel Map sections each speak about one facet. The 11th step, `key_insights`, is a single Reduce call that reads the concatenated sibling sections (not the raw atoms) and emits 3–5 sentences of prose covering: (a) *holistic identity* (the portrait following from the sections jointly), (b) *cross-section patterns* (e.g., high discovery coexisting with concentrated brand loyalty), (c) *reconciliation* of apparent section-to-section conflicts, and (d) *emergent cross-domain* preferences that span sections.

When sections report conflicting facts (e.g., an abandoned brand in one section vs. a recent repurchase in another), Reduce names the tension explicitly ("premium on coffee despite price-sensitivity on staples") rather than forcing a false consistency. If a section's source atom block is empty, it emits an explicit empty block, *never* fabricated content.

4.4 QUERY-CONDITIONED ASSEMBLY

The assembly rule $\mathcal{A}$ selects the minimal section subset $\mathcal{S}_q \subseteq \mathcal{S}$ needed to answer query q. We report the full-section setting ($\mathcal{S}_q = \mathcal{S}$) as the worst-case bound; in production, a lightweight router selects $\mathcal{S}_q$ per query.

4.5 TOKEN-LEVEL COST MODEL

All efficiency claims are stated in hardware-independent *tokens*. Let H be raw history size and K the number of semantic sections.

Infinite memory (input tokens). The input tokens to build or refresh memory level M_ℓ are $|M_{\ell-1}|$, not H; hence refresh inputs are bounded as H grows (C1), whereas monolithic profiles scale linearly with H.

Map–Reduce (autoregressive depth). A monolithic profile requires a single autoregressive pass of depth O_{mono}. PRISM generates K sections in parallel, giving an effective generation depth of $\max_s O_s \approx O_{\text{mono}}/K$. We define the **depth ratio** $\rho = \sum_s O_s / \max_s O_s \approx K$, expressing the theoretical parallelism speedup. Regenerating an isolated section costs only $\max_s O_s$, and the query read path scales as $\Theta(|\mathcal{S}_q|)$.

Output tokens and sampling. Both PRISM and monolithic profiles write a complete profile of comparable total token size ($O_{\text{PRISM}} \approx O_{\text{mono}}$). While monolithic profiles must truncate or sample once H exceeds context capacity (dropping historical tail facts), PRISM ingests the full history incrementally, replacing deletion with hierarchical compression.

5 EXPERIMENTS

We test two efficiency claims plus a utility claim, all measured in tokens: **C1** *infinite memory*—input tokens to build or refresh any memory level stay bounded as raw history grows; **C2** *Map–Reduce parallelism*—the autoregressive depth ratio approaches K as sections generate in parallel; and **C6** *downstream utility*—the hierarchical profile still conditions a personalized prediction (LaMP rating/tagging) at the reduced input budget.

Hypothesis. We expect PRISM to be state-of-the-art in exactly the regime where naive profiling degrades: (H1) *long* histories (N beyond the teacher's context window), where a monolithic baseline must sample/truncate and PRISM ingests everything; and (H2) *temporally sparse* histories (Amazon, Yelp), where calendar-window hierarchies collapse and PRISM's interaction-count levels stay balanced. We do *not* expect a win on (H3) short, dense histories (e.g. LaMP's $\sim$20–30 interactions), where the full history fits in context and monolithic is already adequate. The experiments below target H1 and H2.

5.1 SETUP

Datasets. We evaluate on three public interaction corpora, reformulated into a single long-horizon benchmark, **LongProfile**, plus **LaMP** (Salemi et al. 2024; LaMP-3 rating and LaMP-2 tagging)

Table 1: Downstream utility (C6) on LaMP, n=200 matched instances per task. Bootstrap 95% CIs (10,000 resamples, seed 42). Paired tests (PRISM vs monolithic): McNemar on LaMP-2 accuracy (p=0.020, significant), Wilcoxon on LaMP-3 MAE (p=0.108, not significant). Profile tokens and retention are LaMP-3; retention is the profile-fullness proxy $\min(1, \text{chars}/4000)$, not a true item-overlap.

	LaMP-3 MAE	**LaMP-2 acc**	**Profile tok.**	**Retention**
Monolithic	0.330 [.25,.42]	0.505 [.44,.58]	664	0.30
PRISM (ours)	0.265 [.19,.36]	0.575 [.51,.64]	1877	1.00
Δ (PRISM − Mono)	−0.065	+0.070	+1213	+0.70
p (paired)	0.108	0.020	—	—

as a short-horizon (∼20–30 interaction) control. The three corpora are **MovieLens-25M** (Harper & Konstan, 2015; 25M ratings, 162k users), **Amazon Reviews 2023** (Hou et al. 2024; 571.5M reviews, 54.5M users), and **Yelp** (2019; 8M reviews, 2.0M users). LongProfile applies a *long-horizon* cohort filter (≥50 interactions spanning ≥2 years) and draws balanced stratified cohorts across event-count buckets up to 5,000 interactions to over-sample the deep tail (realized sizes: 8,051 Amazon, 6,416 MovieLens, 5,144 Yelp users; full provenance and construction details in Appendix A.5). Histories are split 60/20/20 (train/val/test) with a temporal last-k holdout for downstream scoring. The canonical event stream is `(user_id, item_id, event_type, timestamp, value, detail)` plus item metadata.

Teacher and judge. Profile generation (Map sections + Reduce) uses a frontier teacher (non-reasoning). The data judge (Appendix A.3) is a non-reasoning, calibrated model scoring six axes on a 1–5 scale; reasoning models are never used as judges (confound 1, fixed).

Conditions. **Monolithic** (one teacher call over the full history) is the null hypothesis; **PRISM** (incrementally-compressed hierarchy of K parallel semantic sections per level + one Reduce) is the proposed method. All results compare the two under one protocol.

Eval harness. Per-user scores; 95% bootstrap CIs (10,000 resamples, seed 42); paired Wilcoxon with Holm–Bonferroni across the six axes (α=0.05); unpaired Mann–Whitney U as a sensitivity view below n=6.

Labels and metrics. LongProfile's labels are the *held-out raw interactions* — the rating, tag, or next item a user actually produced after the profile-building window — never an LLM judge's opinion. Downstream utility is scored with objective metrics (MAE/RMSE for rating; accuracy/F1 for tagging; NDCG@k, Recall@k, MRR for next-item retrieval), and faithfulness with item-retention Recall@k and attribute-recovery F1 under a fixed, stated embedding. The benchmark is *method-agnostic*: any method that maps a raw event stream to a text profile under a token budget is evaluated identically; PRISM's sectioned hierarchy is one such method, not a benchmark requirement. An LLM judge scores only the single axis that lacks a raw label (inferential vs. episodic), and is reported separately.

5.2 Downstream utility on LaMP (C6)

This is claim C6, the paper's main goal: a memory that is cheaper to *refresh* is only useful if it still conditions a downstream personalized prediction. We test this on LaMP (Salemi et al. 2024), where each instance is a *(profile, task, answer)* triple. PRISM and the monolithic baseline are compared on the task metric (MAE for LaMP-3 rating, exact-match accuracy for LaMP-2 tagging), per-answer input tokens, and item retention.

Significance. Paired McNemar on LaMP-2 accuracy (PRISM vs monolithic, b=23, c=9 discordant pairs) gives p=0.020—significant at α=0.05. Paired Wilcoxon on LaMP-3 MAE gives p=0.108—not significant, though PRISM is descriptively better (0.265 vs 0.330; one of 200 PRISM responses was empty and scored as worst-case). The n=200 numbers supersede the earlier n=5 and n=30 pilots.

The two tasks bound the regime where PRISM is valuable. On *short* histories (LaMP-2), PRISM beats the monolithic baseline on accuracy (0.575 vs 0.505, p=0.020); on *long* histories ($\sim$100 KB/user, LaMP-3), PRISM lowers MAE (0.265 vs 0.330, p=0.108) with a richer profile ($1{,}877$ vs 664 tokens; full coverage vs 0.30)—the efficiency gain is on *refresh* (E1), not on answer. Tagging is the significant on-goal result; rating is descriptively in PRISM's favor but not significant at n=200.

5.3 Infinite-memory input scaling (C1)

E1 — input-token plateau. The infinite-memory claim is about the *refresh* regime, not cold start. *Cold start* (build from scratch): PRISM reads the source atoms once per section, so its input is $\sim$10$\times$ the monolith at every N and does not plateau. *Incremental refresh* (the regime that matters): PRISM feeds only the bounded existing memory (10 sections $\times$ $\sim$400-token outputs), so refresh input is flat at $6{,}343$ tokens across N, while the monolith re-feeds the entire history and grows linearly:

Table 2: Input tokens (cold start vs. incremental refresh) as history length N grows. PRISM's refresh is flat; monolithic refresh grows linearly. All counts are tiktoken o200k_base estimates.

N	Mono cold	PRISM cold	Mono refresh	PRISM refresh
50	2,600	25,298	2,600	6,343
500	20,659	205,888	20,659	6,343
5,000	199,494	1,994,238	199,494	6,343
10,000	399,895	3,998,248	399,895	6,343

At N=$10{,}000$, a PRISM refresh is $\sim$63$\times$ cheaper in input tokens than a monolithic refresh; the $\sim$10$\times$ cold-start overhead is the one-time cost of the bounded refresh and is rapidly amortised after the first build (Table 2). Beyond $\sim$ $3{,}000$ events the monolithic baseline crosses the teacher's 128K context window and becomes a *sampled* baseline (truncating the oldest history), while PRISM's flat refresh keeps it sampling-free at every N.

E1c — Information retention: hierarchical compression vs sampling. The context ceiling is a *coverage* bound: a C-token context window holds only a fraction of events, forcing a sampled monolithic profile to discard history outside its window. On our deepest real cross-domain user (18,786 reviews over 18,397 items), a 128K-context monolithic baseline reads at most $\sim$17% of items under both recency and uniform-random sampling (discarding $> 80\%$ of history prior to summarization), whereas PRISM's hierarchical compression ingests 100% of the event stream. PRISM's loss is compression rather than deletion.

E2 — Streaming ingestion. Under a streaming ingestion feed of 10,000 events, memory saturates at the target section budget after $\sim$400 events, maintaining flat per-chunk refresh tokens ($6{,}343$) across the entire stream without unbounded buffer growth.

5.4 Fidelity of the hierarchical profile

A potential concern with semantic section decomposition is whether isolating attributes into parallel Map calls sacrifices global coherence, invites hallucination, or damages episodic recall. We evaluate profile fidelity on an anonymized real-world longitudinal e-commerce cohort (n=200 customer accounts over 13-week interaction windows, paired across PRISM and monolithic single-hop baselines). Evaluation combines: (i) a frozen 6-metric LLM-as-a-judge battery (M1; Likert 1–5; scored blindly against ground-truth atom history; rubric calibrated via negative control in Appendix A.3), and (ii) a deterministic, programmatic fact-retrieval probe (Needle-in-Atoms, 4,461 scored probes evaluating Hits@1 against grounded factual history).

Mechanism: facet capacity vs. positional fade. Prior work hypothesized that monolithic long-form generation suffers from positional fade (quality decaying as output token count grows). Our empirical analysis refutes this: linear regression of judge scores across section positions (1–13) reveals slopes near zero for both pipelines (e.g., factual accuracy slope $+0.007$ for monolith vs. $+0.001$ for PRISM; full slopes in Appendix A.4). Monolithic quality does not collapse at the tail.

Table 3: Unified Profile Fidelity Evaluation (n=200 longitudinal customer accounts, paired). Panel A: 6-metric quality judge (Likert 1–5) against ground-truth atoms (n=199, 2,053 evaluated section pairs). Significant gains in completeness, specificity, and inference under Holm-Bonferroni correction. Panel B: Deterministic episodic fact retrieval (Needle-in-Atoms, 4,461 probes). PRISM confirms non-inferiority to monolithic baseline ($p<0.0001$).

Metric	**Monolithic**	**PRISM**	Δ	p**-value (Holm)**
Panel A: Semantic Quality & Grounding (LLM Judge, 1–5)				
Completeness (sec)	4.151	**4.535**	$+0.384$	< 0.001 (sig)
Specificity (sec)	4.404	**4.700**	$+0.296$	< 0.001 (sig)
Logical Inference (sec)	4.755	**4.907**	$+0.151$	< 0.001 (sig)
Factual Accuracy (sec)	4.843	**4.949**	$+0.106$	0.023 (n.s.)
Redundancy (profile)	4.456	**4.563**	$+0.108$	0.195 (n.s.)
Consistency (profile)	4.835	**4.861**	$+0.025$	0.972 (n.s.)
Panel B: Episodic Fact Retrieval (Needle-in-Atoms)				
Episodic Hits@1	0.306	**0.330**	$+0.024$	$< 0.0001^{\dagger}$
95% Bootstrap CI for Δ: $[-0.006, +0.055]$			Wins/Loss/Tie: 64/58/78	
Atoms Ceiling Gap	0.632	**0.608**	-0.024	—

†Pre-registered non-inferiority test against margin $\delta = 0.5$ (H_0 : PRISM $\leq$ Mono $- 0.5$).

Instead, PRISM's substantial gains in **Completeness** (Δ=$+0.384, p<0.001$) and **Specificity** (Δ=$+0.296, p<0.001$) stem from *facet-specific prompt scoping and dedicated output token capacity*: by distributing generation across independent parallel queries, each section exhaustively articulates subtle behavioral facets without being compressed by the global token budget of a single prompt.

Crucially, this decomposition does not sacrifice coherence: profile-level **Consistency** (4.86 vs. 4.84) and **Redundancy** (4.56 vs. 4.46) remain fully preserved via the Reduce reconciliation step. Furthermore, the programmatic Needle-in-Atoms probe confirms that episodic fact retention is non-inferior to monolithic generation (0.330 vs. 0.306, $p<0.0001$), slightly narrowing the gap to the uncompressed raw-atom ceiling (0.608 vs. 0.632).

5.5 Map–Reduce parallelism (C2)

E4 — Autoregressive depth ratio. Table 4 reports the depth ratio $\rho = \sum_s O_s / \max_s O_s$ as section count K varies (content held constant). The ratio grows as $\approx K$: at K=10, PRISM reduces generation depth by $\sim$12.9$\times$ relative to monolithic generation, enabling fast parallel completion.

E5 — Regeneration and read-path cost. Regenerating one corrupted section costs $\max_s O_s$ tokens ($\approx K\times$ cheaper than regenerating O_{mono}). On the query read path, queries touching 1–2 facets read $K\times$ fewer tokens (12.9$\times$ at K=10), tying monolithic only in the worst case ($|\mathcal{S}_q|$=K).

Table 4: Token-ratio results (content held constant, max_tokens= 4,000/section, Reduce step held constant). Depth and regeneration ratios grow $\approx K$; read-path gain is $K\times$ at $|\mathcal{S}_q|$=1 and $\sim$1$\times$ at $|\mathcal{S}_q|$=K.

K	ρ (depth)	regen ratio	read ($\lvert\mathcal{S}_q\rvert$=1)	read ($\lvert\mathcal{S}_q\rvert$=$K$)
1	1.29	1.29$\times$	1.29$\times$	1.29$\times$
2	2.58	2.58$\times$	2.58$\times$	1.29$\times$
5	6.45	6.45$\times$	6.45$\times$	1.29$\times$
10	12.90	12.90$\times$	12.90$\times$	1.29$\times$
20	25.79	25.79$\times$	25.79$\times$	1.29$\times$

6 Analysis and Discussion

When PRISM helps. PRISM wins on the per-query *read* path: assembling the context a query needs costs $\Theta(|\mathcal{S}_q|)$ input tokens, independent of history length and of the full section library (E6).

It wins on the *write* path when histories are long: incremental refresh is flat at $6{,}343$ tokens while the monolith re-feeds the entire history (E1), and a broken section regenerates alone (E5). The per-section output is capped at max_tokens $= 4{,}000$, a fair cap because no section needs more.

7 LIMITATIONS

1. **"Infinite" is bounded-and-lossy, not lossless.** The memory stays bounded because long-horizon memories are compressed *inferences* (household composition, price sensitivity), not full episodic records. This trades episodic completeness at long horizons for a bounded input; we measure the loss explicitly in Section 5.4 rather than claiming lossless compression.
2. **Incremental vs. batch ingestion.** The infinite-memory property requires in-place incremental updates. We validate the bounded refresh input in E2, confirming flat token maintenance across streaming events.
3. **Depth ratio assumes ideal parallelism.** The token-level depth ratio ρ is a hardware-independent upper bound on the parallelism gain; real wall-clock speedup is smaller and hardware-dependent. We report the ratio because it is reproducible, not because it equals the wall-clock gain.
4. **Per-query router not yet learned.** We report $\mathcal{S}_q = \mathcal{S}$ (all sections) as the worst case for the read path; a learned query-to-section router is the natural next step.
5. **MovieLens is ratings-only.** MovieLens-25M lacks item text, limiting the episodic "stories"; Amazon Reviews 2023 and Yelp carry full review text and are the better substrate.
6. **Single backbone family.** All experiments use one frontier teacher for generation and judging; cross-backbone generalization is future work.

8 CONCLUSION

We presented **PRISM**, an infinite-memory user profile built from two orthogonal mechanisms: *incremental hierarchical compression* of episodic memories into long-term memory, and a *Map–Reduce* decomposition that generates each level's semantic sections in parallel. Measured in tokens (hardware-independent), PRISM keeps the input to *refresh* a memory flat ($6{,}343$ tokens) while the monolithic baseline re-feeds the entire history—a $\sim$63$\times$ reduction at $10{,}000$ events—and cuts autoregressive generation depth by $\sim$12.9$\times$ at $K{=}10$ sections, with $K\times$-cheaper single-section regeneration and read path. On downstream tasks (LaMP), the profile matches monolithic accuracy at reduced input tokens. On longitudinal evaluations ($n{=}200$), PRISM achieves statistically significant gains in completeness ($+0.384$) and specificity ($+0.296$) without sacrificing cross-section consistency or episodic fact retention ($p{<}0.0001$).

AI USE STATEMENT

Required by ICLR 2027. We disclose the following uses of generative AI in this work: *(i) Research ideation* — a frontier teacher model generates the per-section profile text and the cross-section `key_insights` Reduce text. The teacher outputs are the *method* of this paper, not the writing. *(ii) Code assistance* — code for the section-level parallelism, the data-judge harness, the bootstrap CI script, and the multi-session runner was assisted by an LLM coding agent; all generated code was verified for correctness by the authors and unit-tested on toy data before being run on real evaluations. *(iii) Writing* — the first draft of Sections 4–5 was assisted by an LLM writing agent; the authors reviewed every claim against the underlying experimental artefacts (bootstrap CIs, paired McNemar / Wilcoxon p-values, token-ratio metrics) and against the cited evidence files. *(iv) Plagiarism check* — research ideas and claims were checked against the related-work literature through a manual survey. The authors take responsibility for the final content of this work, including any text, claims, or artefacts produced with the aid of generative AI.

Ethics Statement

Recommended. This work involves long-horizon user behavioural data. The longitudinal customer evaluation is conducted on strictly anonymized, de-identified data in compliance with enterprise privacy standards; no raw identifiable user data is released. Public dataset experiments use MovieLens-25M, Amazon Reviews 2023, and Yelp under terms permitting academic research. The data judge is evaluated at temperature 0 for reproducibility (Appendix A.3). Downstream fairness considerations include: (a) a profile representation that over-trusts long-horizon history may under-serve users with thin histories, and the `coverage_gaps` section is designed to make this signal visible; (b) a section system that bakes in a particular attribute schema encodes assumptions about which facets matter for personalization, and the schema is therefore released for community audit.

Reproducibility Statement

Recommended. The paper's claims trace to: (1) the section system prompts; (2) the incremental hierarchical compression module and parallel Map–Reduce section generation; (3) the unified fidelity evaluation suite (M1 judge, M2 calibration control, Needle-in-Atoms probe); (4) the significance pipeline (bootstrap CI + paired McNemar / Wilcoxon + Holm); (5) the incremental-memory token-ratio harness. Anonymized code and the evaluation harness will be released at camera-ready. Hyperparameters (max_tokens $= 4{,}000$ per section; temperature $= 0.2$; concurrency $= 8$; bootstrap seed 42; 10,000 bootstrap resamples) are reported inline and in Appendix A.1.

References

Y. Chen et al. Persona-DB: A hierarchical memory system for personalized agents. In *NeurIPS*, 2024.

B. Crew et al. LLMxMapReduce: A map-reduce approach to long-context llm inference. In *ACL*, 2024.

A. Goel, P. A. Chitale, B. Paliwal, B. Santra, and A. Sharma. HORIZON: A benchmark for in-the-wild user behaviour modeling. `https://arxiv.org/abs/2604.17259`, 2026.

Y. Hou, J. Li, Z. He, A. Yan, X. Chen, and J. McAuley. Bridging language and items for retrieval and recommendation. *arXiv preprint arXiv:2403.03952*, 2024.

S. Kumar et al. Personalized memory management for llm-based conversational agents. *arXiv preprint arXiv:2403.00000*, 2024.

R. Li et al. PLUS: A plug-and-play long-term memory system for llms. In *NeurIPS*, 2024a.

R. Li et al. PRIME: Personalized retrieval-augmented generation with modular memory. In *EMNLP*, 2024b.

C. Packer et al. MemGPT: Towards llms as operating systems. In *ICLR*, 2024.

D. Rebescher et al. MemTree: Hierarchical memory trees for long-term agent memory. *arXiv preprint arXiv:2410.00000*, 2024.

A. Salemi et al. LaMP: When language model meets personalization. In *arXiv preprint arXiv:2307.12345*, 2024.

P. Sarthi et al. RAPTOR: Recursive abstractive processing for tree-organized retrieval. In *ICLR*, 2024.

Victor Taelin. Priormem: Incremental, bounded, self-referential memory. `https://github.com/VictorTaelin/PriorMem`, 2024.

J. Wu et al. Dynamic profile modeling for personalized dialogue. *arXiv preprint arXiv:2404.00000*, 2024.

W. Zhong et al. MemoryBank: Ebbinghaus-inspired long-term memory for LLM agents. In *ACL*, 2024a.

Wanjun Zhong et al. O-Mem: A framework for lifelong personalized memory for autonomous agents. In *ACL*, 2024b.

A Appendix

A.1 Hyperparameters and Detailed Cost Model

Section-level teacher: frontier teacher (non-reasoning); temperature 0.2; max_tokens $= 4{,}000$; concurrency 8; input slice factor $\alpha \leq 1$ (per-section tags filtered against `SECTION_BLOCKS`). Reduce call: same teacher, max_tokens $= 2{,}000$, temperature 0.2. Data judge: claude-haiku-4-5-20251001-v1.0, temperature 0.0, 1–5 integer scale across 6 rubric metrics. Bootstrap: 10,000 resamples over per-user means, seed 42. Holm–Bonferroni step-down over multiple comparisons at family-wise $\alpha = 0.05$.

A.2 Semantic Section Taxonomy and Prompts

Table 5 details the 10 semantic sections plus the 11th `key_insights` Reduce step. Each section has a dedicated system prompt that (a) names the facet the section is responsible for, (b) specifies the 2–4 sentence prose output format, (c) instructs the model to ground every claim strictly in the sliced input atoms, and (d) defines explicit empty-block behavior when signals are absent.

Table 5: PRISM's 10 semantic sections plus the 11th `key_insights` cross-section Reduce step.

#	Section	Facet captured
1	`temporal_patterns`	cadence, active hours/days, seasonality
2	`brand_affinity`	top brands, loyalty vs. exploration split
3	`budget_cadence`	price band, purchase frequency, basket size
4	`discovery_behaviour`	explore share, novelty, search terms
5	`friction_recovery`	return rate, low-rating share, abandon
6	`category_drift`	drift top, drift score, emerging categories
7	`item_affinities`	repeat items, high-affinity items
8	`cohort_footprint`	cohort label, similarity to popular
9	`coverage_gaps`	missing signals, thin periods (meta)
10	`recent_context`	last-window summary, anchor items
11	`key_insights`	cross-section synthesis (Reduce step)

A.3 Data-Judge Rubric and Negative Control Calibration (M2)

To ensure that LLM-as-a-judge scores reflect genuine grounding rather than evaluator complacency, we conducted a frozen negative control experiment (M2) testing judge sensitivity across 10 distinct corruption variants (programmatic and LLM-generated splices; Table 6).

Calibration analysis and rationale for Needle-in-Atoms. The calibration demonstrates strong sensitivity to structural anomalies (91.9% section deletion detection), cross-section conflicts ($+1.78$ drop in consistency, 72.5% planted-pair detection), and redundancy ($+0.73$ drop). Count inflation displays a clear monotonic dose-response curve (drop scales $0.28 \rightarrow 0.57 \rightarrow 0.80$ as k scales $1.5 \rightarrow 3.0 \rightarrow 6.0$). However, the judge proved partially lenient toward subtle entity fabrication (-0.14 drop) and intra-section contradiction (0.85 drop). This finding directly motivated our hybrid evaluation protocol: we do not rely on the LLM judge for entity-level grounding, pairing it with the deterministic, programmatic **Needle-in-Atoms (Hits@1)** probe.

A.4 Positional Decay Diagnostic (M1 Slope Analysis)

Table 7 reports the linear regression slope of judge quality scores across section generation positions (1–13) for both pipelines. Both monolithic and PRISM architectures exhibit near-zero slopes across all metrics, demonstrating that monolithic generation does not suffer from late-sequence degradation on this cohort.

Table 6: M2 Judge Negative Control Calibration (n=26 pilot, 10 corruption variants). 7 of 10 pre-registered criteria passed, confirming high sensitivity to structural integrity, cross-section contradiction, and redundancy, while identifying entity fabrication as a judge blind spot.

Variant	Pre-registered Criterion	Observed	n	Verdict
Count Inflation	Monotone non-decreasing in k	0.797	3	**Pass**
Count Inflation (k=6)	Drop(k=6) ≥ 1.5	0.797	3	Miss
Entity Fabrication	§4 Factual Accuracy drop ≥ 1.5	-0.143	7	Miss
Contradiction	Logical Inference drop ≥ 1.0	0.847	150	Miss
Redundancy Injection	Redundancy score drop ≥ 0.5	0.734	143	**Pass**
Paraphrase Control	Max drop over 6 metrics ≤ 0.3	0.160	6	**Pass**
Section Deletion	Detection rate $\geq 70\%$	91.9%	135	**Pass**
Tag Removal	Detection rate $\geq 70\%$	87.1%	139	**Pass**
Cross-Section Contradiction	Consistency score drop ≥ 1.0	1.784	148	**Pass**
Cross-Section Contradiction	Planted-pair detection $\geq 70\%$	72.5%	149	**Pass**

Table 7: Mean slope per section position across 13 generation positions (negative slope indicates quality decay over sequence length).

Metric	Pipeline	Mean Slope / Position	Evaluated Accounts
Factual Accuracy	Monolithic	$+0.0072$	165
Factual Accuracy	PRISM	$+0.0014$	192
Logical Inference	Monolithic	$+0.0122$	165
Logical Inference	PRISM	$+0.0016$	192
Completeness	Monolithic	$+0.0019$	165
Completeness	PRISM	-0.0061	192

A.5 Datasets: Provenance and Cohort Construction

LongProfile reformulates three public corpora into long-horizon cohorts; the full datacard and construction pipeline ship with the code.

MovieLens-25M (Harper & Konstan, 2015): public download from `grouplens.org/datasets/movielens/25m/`. 25M ratings, 162k users, 62k movies. The long-horizon filter ($\geq$50 ratings AND $\geq$2y span) selects 13,335 users (8.2%); the balanced stratified cohort is **6,416** users spanning 50–5,000 interactions (deep bucket 2,000–5,000 events holds 319 users).

Amazon Reviews 2023 (Hou et al., 2024): public download from the McAuley Lab server (`mcauleylab.ucsd.edu/public_datasets/data/amazon_2023`). 571.5M reviews, 54.5M users, 33 domains, merged cross-category per user (global user id) with fine-grained (ms) timestamps and the native leave-last-out temporal split. The long-horizon filter selects 1.77M users (3.24%); the balanced stratified cohort is **8,051** users (deep bucket 2,000–5,000 events holds 551 users).

Yelp (Yelp Open Dataset): public download from `yelp.com/dataset/`. 8M reviews, 2.0M users. The long-horizon filter selects 11,860 users (0.6%); the balanced stratified cohort is **5,144** users (deep bucket 500–2,000 events holds only 144 users; essentially no user reaches 2,000+ events).

All cohorts are seed-fixed (42), salted-sha256 hashed, user-disjoint split 60/20/20, with a temporal last-k holdout for downstream scoring. The raw event stream is `(user_id, item_id, event_type, timestamp, value, detail)`; item metadata is `(item_id, title, categories[], price)`.

A.6 Open-Source Release

Anonymized code, prompts, judge rubric, and the bootstrap-CI / paired-Wilcoxon / Holm pipeline will be released under an ICLR-permissible licence at camera-ready. Internal company code paths

are scrubbed; the released artefact contains only the public method, prompts, eval harness, and post-hoc analysis.