Sims: A Multi-Context Memory Architecture for Precision-Aware Personalization in Language Models

Chirag Shah
University of Washington
Seattle, USA
chirags@uw.edu

Abstract

Personalized language model systems overwhelmingly treat users as single, undifferentiated entities: a flat profile of facts, preferences, or conversation history fed into a prompt. This works poorly when the same person needs different things in different contexts, and fails entirely when those contexts have conflicting priorities. We introduce **Sims** (Simulation constructs), a structured multi-context memory architecture proposed by Shah and White [2025] as part of a broader agent ecosystem. Rather than merging user context into a flat undifferentiated block, Sims organizes personal information into semantically typed domain-specific blocks that give a language model explicit signal about the role and weight of each piece of context. This paper provides the first empirical evaluation of Sim-structured representations on two established personalization benchmarks. We compare three prompt construction strategies: a flat merged profile (Condition A), structured Sim blocks with explicit conflict-resolution instruction (Condition B), and structured Sim blocks without such instruction (Condition C). On LaMP-QA [Salemi and Zamani, 2025], both structured conditions outperform the flat baseline in ROUGE-L (0.1380 and 0.1389 vs. 0.1312). On LOCOMO [Maharana et al., 2024], structured representations nearly double span-containment accuracy on multi-hop reasoning queries (Cat-3: 10.42% vs. 5.21%), and Condition B outperforms Condition C on preference-sensitive queries (Cat-5) — the category where conflict-resolution reasoning matters most. These results establish that structured multi-context representation consistently outperforms flat profiles, and that explicit conflict reasoning provides targeted benefit on preference-sensitive tasks. We also identify a gap in existing evaluation frameworks: current metrics reward recall and surface overlap but do not penalize contextually inappropriate responses, limiting their ability to distinguish structured personalization from verbosity.

1 Introduction

The dominant approach to personalizing language model outputs is surprisingly crude. A user's preferences, history, or attributes get concatenated into a prompt block, and the model treats this flat representation as the complete picture of who the user is. For narrow tasks this is often fine. For anything that spans multiple life domains, it creates two distinct failure modes.

The first is *context interference*: irrelevant personal information contaminates the response. A model asked about meal planning might surface work scheduling constraints, or weight fitness goals against family dietary restrictions without any principled basis for the tradeoff. The second is *context collapse*: a single profile cannot simultaneously represent the professional who needs concise status

updates, the parent who needs to know about school pickup windows, and the endurance athlete tracking macros. Averaging these representations produces outputs that serve none of them well.

Real people do not experience themselves as a single unified agent with a fixed preference vector. Context shapes behavior, priorities, and constraints in ways that are orthogonal to each other. A business traveler's preferences for a client dinner look nothing like their preferences for a Saturday hike. Existing memory systems that do not model this structure necessarily underserve users whose lives cross multiple domains.

Sims, introduced by Shah and White [2025] as part of a proposed ecosystem of agents, Sims, and Assistants, are structured domain-specific representations of a user in a given life context. Each Sim captures preferences, behavioral patterns, constraints, and decision weights for one domain. Multiple Sims coexist for a single user, each carrying a defined semantic role. This paper asks a targeted empirical question: does structuring user context into typed Sim blocks improve personalized LLM outputs, and does adding an explicit conflict-resolution instruction on top of that structure provide further benefit?

We test three prompt strategies on two established benchmarks. Condition A is a flat merged profile — the standard approach. Condition B adds Sim block structure plus an explicit instruction to resolve conflicts between blocks. Condition C provides the same structure without the conflict instruction. Results across LaMP-QA and LOCOMO show that structure consistently improves over flat profiles, and that conflict-resolution instructions help specifically on preference-sensitive tasks.

2 Related Work

Agents, Sims, and the ecosystem framing. Shah and White [2025] argue that generative AI agents alone are insufficient for effective task execution and propose a three-component ecosystem: agents that execute tasks, Assistants that interact with users and coordinate execution, and Sims that represent user preferences and behaviors across life contexts. That paper introduces the Sims concept, the ecosystem framing, and the distinction between persistent user representations and ephemeral task-execution instances. The present paper provides the first empirical evaluation of the Sim context selection mechanism, testing whether structured multi-context representations with arbitration outperform flat profiles and unarbitrated alternatives.

Memory in language models. Memory-augmented language model systems span a wide design space. Retrieval-augmented generation (RAG) [Lewis et al., 2020] grounds responses in retrieved documents but treats retrieval and generation as largely independent steps. MemGPT [Packer et al., 2023] introduces tiered memory management for extended conversations. Neither system models the multi-context structure of a single user's life.

Personalized LLMs. Recent work on personalized generation includes persona-conditioned models [Zhang et al., 2018] and user-adaptive summarization [Salemi et al., 2023]. LaMP [Salemi et al., 2023] provides a benchmark for personalized NLP tasks using historical user data. These approaches treat personalization as a prompt engineering or retrieval problem rather than a structured representation problem.

Context filtering and multi-agent retrieval. MAIN-RAG [Tang et al., 2025] introduces multi-agent filtering for retrieved documents, using adaptive thresholds to reduce noise in RAG pipelines. The arbitration mechanism in Sims shares the intuition that not all retrieved context is equally relevant, but operates on structured user representations rather than document corpora.

Digital twins and user modeling. Digital twin frameworks model entities as persistent computational representations [Grieves, 2014]. Human digital twins for health applications [Björnsson et al., 2019] operate on biological process data. Sims differs from persistent digital twin paradigms in two ways: Sim representations are domain-segmented rather than unified, and task execution uses ephemeral instances that dissolve after completion rather than maintaining persistent state.

On-device personalization. Parameter-efficient fine-tuning methods including LoRA [Hu et al., 2021] enable adaptation with minimal compute overhead. The current experiments operate at the prompt level; Sim-guided on-device fine-tuning is a planned extension of this work.

Despite meaningful progress across these research threads, a critical gap persists. Memory-augmented systems treat user context as a monolithic retrieval corpus; personalization benchmarks evaluate response quality against static reference outputs; digital twin frameworks model entities as unified persistent representations; and retrieval-based personalization methods optimize for relevance without modeling the structural conflict that arises when a single user holds orthogonal preferences across life domains. No existing work addresses the compound problem: a user is not one person but many, each context carrying distinct priorities, communication norms, and constraints that may actively contradict one another. Flat profiles average across this variation and lose it. RAG pipelines retrieve within it without resolving it. The consequence is a class of personalization failures that current benchmarks do not measure and current architectures cannot fix – responses that are individually plausible but contextually wrong, technically accurate but contextually incoherent. What is needed is a representation layer that preserves the orthogonality of life contexts rather than collapsing it, a selection mechanism that identifies which contexts bear on a given task rather than passing everything, and an inference procedure that reasons explicitly about inter-context conflict rather than leaving that reasoning implicit. The Sims architecture addresses all three.

3 The Sims Architecture

3.1 Formal Sim Definition

Definition 1 (Sim)**.** A *Sim* S_k is a typed structured object representing user u in life context $k \in \mathcal{K}$, where $\mathcal{K}$ is the set of recognized life domains (e.g., WORK, FAMILY, HEALTH, HOBBY). Formally:

$$S_k = \langle \mathbf{p}_k,\ \mathbf{w}_k,\ \boldsymbol{\theta}_k,\ \alpha_k,\ \ell_k,\ t_k \rangle \tag{1}$$

where:

- $\mathbf{p}_k \in \mathbb{R}^d$ is the *preference vector*: a dense encoding of domain-specific likes, dislikes, constraints, and communication style.
- $\mathbf{w}_k \in \mathbb{R}^n$ is the *behavioral weight vector*: frequency-weighted encodings of past decisions and patterns observed in context k.
- $\boldsymbol{\theta}_k$ is the *context parameter set*: structured key-value attributes including role description, priority list, and constraint string.
- α_k is the *activation profile*: a sparse descriptor over domain indicators, temporal patterns, and entity references, used for relevance scoring at inference time.
- $\ell_k \in \{\text{NASCENT}, \text{ACTIVE}, \text{MATURE}, \text{DORMANT}\}$ is the *lifecycle state*.
- $t_k \in \mathbb{R}$ is the *last-updated timestamp* (Unix epoch).

The full Sim space for user u is $\mathcal{S}_u = \{S_k : k \in \mathcal{K}\}$.

Algorithm 1 Sim Lifecycle Transition

Require: Sim S_k, thresholds $\delta_{\text{nascent}}, \delta_{\text{dormant}}, f_{\min}$, window W
Ensure: Updated lifecycle state ℓ_k

if $f_k = 0$ **and** $\Delta t_k > \delta_{\text{nascent}}$ **then**
  $\ell_k \leftarrow$ NASCENT ▷ No interactions observed yet
else if $\Delta t_k > \delta_{\text{dormant}}$ **or** $f_k < f_{\min}$ **then**
  $\ell_k \leftarrow$ DORMANT ▷ Inactive beyond threshold
else if $f_k \geq f_{\min}$ **and** $\Delta t_k \leq \delta_{\text{dormant}}$ **then**
  if $\ell_k =$ MATURE **then**
    $\ell_k \leftarrow$ MATURE ▷ Stable high-use Sim; no downgrade
  else
    $\ell_k \leftarrow$ ACTIVE
  end if
end if
if $\ell_k =$ ACTIVE **and** $f_k \geq f_{\text{mature}}$ **and** $\Delta t_k \leq \delta_{\text{mature}}$ **then**
  $\ell_k \leftarrow$ MATURE ▷ Promote on sustained high engagement
end if
return ℓ_k

Algorithm 2 Context Fingerprinting

Require: Query string q, domain lexicon $\mathcal{L}$, entity gazetteer $\mathcal{G}$
Ensure: Context fingerprint $\phi(q) = \langle \mathbf{d}, \mathbf{e}, \mathbf{v} \rangle$

$\mathbf{d} \leftarrow \text{DOMAINSCAN}(q, \mathcal{L})$ ▷ Keyword and n-gram matches to domain indicators
$\mathbf{e} \leftarrow \text{ENTITYEXTRACT}(q, \mathcal{G})$ ▷ Named entities: people, places, organizations
$h \leftarrow \text{PARSETEMPORAL}(q)$ ▷ Extract time expressions, day-of-week, urgency markers
if $h \neq \emptyset$ **then**
  $\mathbf{v} \leftarrow \text{TEMPORALENCODE}(h)$ ▷ Map to domain-relevant temporal vector
else
  $\mathbf{v} \leftarrow \mathbf{0}$
end if
return $\phi(q) = \langle \mathbf{d}, \mathbf{e}, \mathbf{v} \rangle$

3.2 Sim Lifecycle Management

Sims are not static. A Sim transitions between lifecycle states based on interaction recency and frequency, which governs whether it enters the active candidate set during arbitration. Let $\Delta t_k = t_{\text{now}} - t_k$ denote the elapsed time since the Sim was last updated, and let f_k denote the interaction frequency over a trailing window W. Algorithm 1 defines the transition logic.

Only Sims with $\ell_k \in \{\text{ACTIVE}, \text{MATURE}\}$ enter the arbitration candidate set $\mathcal{S}_u^+ \subseteq \mathcal{S}_u$.

3.3 Context Fingerprinting

When query q arrives, the system extracts a context fingerprint $\phi(q)$ from three signal channels. Algorithm 2 defines this procedure.

In the current implementation, fingerprint extraction is approximated by a structured LLM prompt that maps q to a ranked list of candidate contexts along with a brief chain-of-thought rationale. The full Algorithm 2 represents the intended design with explicit feature extraction modules; the LLM

Algorithm 3 Sim Arbitration and Selection

Require: Query q, active Sims $\mathcal{S}_u^+$, budget m, threshold τ, weights $\boldsymbol{\lambda}$
Ensure: Selected Sim set $\mathcal{A}^* \subseteq \mathcal{S}_u^+$

$\phi(q) \leftarrow \textsc{Fingerprint}(q)$ ▷ Algorithm 2
for each $S_k \in \mathcal{S}_u^+$ **do**
  $r_k \leftarrow r(q, S_k; \boldsymbol{\lambda})$ ▷ Equation 2
end for
$\mathcal{R} \leftarrow \{S_k \in \mathcal{S}_u^+ : r_k \geq \tau\}$ ▷ Filter by relevance threshold
$\mathcal{A}^* \leftarrow \textsc{TopK}(\mathcal{R}, m)$ ▷ Select top-m by score; $m \in \{1, 2\}$
if $\mathcal{A}^* = \emptyset$ **then**
  $\mathcal{A}^* \leftarrow \textsc{TopK}(\mathcal{S}_u^+, 1)$ ▷ Fallback: highest-scoring active Sim
end if
return $\mathcal{A}^*$

approximation is used here to isolate the effect of the arbitration mechanism from implementation-specific feature engineering choices.

3.4 Sim Arbitration and Selection

Given fingerprint $\phi(q)$ and the active candidate set $\mathcal{S}_u^+$, the arbitration step scores each Sim and selects a budget-constrained subset. The relevance score for Sim S_k is:

$$r(q, S_k) = \lambda_1 \cdot \text{sim}_{\text{domain}}(\mathbf{d},\, \alpha_k) + \lambda_2 \cdot \text{sim}_{\text{entity}}(\mathbf{e},\, \alpha_k) + \lambda_3 \cdot \text{sim}_{\text{temporal}}(\mathbf{v},\, \alpha_k) \tag{2}$$

where $\lambda_1 + \lambda_2 + \lambda_3 = 1$ and each sim. function computes cosine similarity between the corresponding fingerprint component and the matching component of α_k. The λ weights are learned from user feedback over a held-out validation window. Algorithm 3 defines the full selection procedure.

3.5 Sim-Conditioned Inference

Given selected Sims $\mathcal{A}^*$, the inference objective is to construct a context-injected prompt $\pi(q, \mathcal{A}^*)$ that maximizes contextual precision while respecting the relevance budget:

$$\mathcal{A}^* = \underset{\mathcal{A} \subseteq \mathcal{S}_u^+,\, |\mathcal{A}| \leq m}{\arg\max} \sum_{k \in \mathcal{A}} r(q, S_k) \quad \text{subject to} \quad r(q, S_k) \geq \tau \;\; \forall\, S_k \in \mathcal{A} \tag{3}$$

This formulation is a constrained subset selection problem. Under the budget $m \leq 2$ used in our experiments, it reduces to a simple top-m operation over scores exceeding τ. Algorithm 4 shows how $\mathcal{A}^*$ maps to a structured system prompt.

Condition A substitutes π_{sys} with a flat unstructured profile string containing the same information merged into a single block. Conditions B and C both set $\mathcal{A}^* = \mathcal{S}_u^+$ — all active Sims — but differ in whether π_{sys} includes a conflict-resolution instruction. Condition B appends an explicit directive instructing the model to reason about and resolve disagreements between Sim blocks; Condition C does not. The three conditions therefore differ only in prompt construction: structure and conflict instruction in B, structure only in C, neither in A. This design isolates two questions independently: does Sim structure help over flat context, and does explicit conflict reasoning add further benefit?

Algorithm 4 Sim-Conditioned Prompt Construction

Require: Query q, selected Sims $\mathcal{A}^*$, serializer SERIALIZE
Ensure: Prompt π

$\pi_{\text{sys}} \leftarrow$ "`You are a multi-context persona with the following contexts:`"
for each $S_k \in \mathcal{A}^*$ **do**
  $b_k \leftarrow \text{SERIALIZE}(S_k)$ ▷ Render $\boldsymbol{\theta}_k$ as structured text block
  $\pi_{\text{sys}} \leftarrow \pi_{\text{sys}} \parallel b_k$ ▷ Concatenate context block
end for
$\pi \leftarrow \langle \pi_{\text{sys}},\ q \rangle$ ▷ Final prompt: system context + user query
return π

4 Experimental Setup

We evaluate three prompt construction strategies across both benchmarks, holding all other inference parameters constant (GPT-3.5-turbo, temperature 0.7, max tokens 500):

- **Condition A** (Flat Profile): All user context merged into a single unstructured block. No Sim separation, no semantic roles, no conflict handling. The model must implicitly infer structure from undifferentiated text.
- **Condition B** (Structured + Conflict Resolution): The same information split into four explicit Sim blocks, each with a defined semantic role. An explicit conflict-resolution instruction tells the model how to reason when blocks disagree.
- **Condition C** (Structured Only): Identical four-block Sim structure as Condition B, but no conflict-resolution instruction. The model receives structure but no guidance on arbitration.

Condition A tests whether structure matters at all. Condition B vs. Condition C isolates the effect of explicit conflict-resolution reasoning on top of structure. All three conditions pass the same total information to the model; the manipulation is purely in how that information is organized and what reasoning instruction, if any, accompanies it.

4.1 LaMP-QA Benchmark

LaMP-QA [Salemi and Zamani, 2025] is a recently introduced benchmark for personalized long-form question answering, covering three major categories: Arts & Entertainment, Lifestyle & Personal Development, and Society & Culture across 45+ subcategories. Each instance pairs a question with a user profile built from past interactions; the task is to generate a response that reflects the user's individual preferences and background. We evaluate on 100 randomly sampled instances from the test set and report ROUGE-L [Lin, 2004] computed using Google's `rouge-score` library. ROUGE-L measures longest common subsequence overlap between generated and reference responses, capturing fluency and content coverage without requiring verbatim string matching.

4.2 LOCOMO Benchmark

LOCOMO [Maharana et al., 2024] evaluates long-context memory use across 1,986 questions spanning five categories (Cat-1 through Cat-5) reflecting different reasoning and retrieval demands — from simple fact lookup to multi-hop cross-temporal reasoning and preference-sensitive queries. Evaluation uses span-containment accuracy: a response is scored correct if the gold answer string

appears verbatim in the model's output. This metric is strict and favors recall; it does not penalize contextually inappropriate responses that happen to contain the answer span. We evaluate all 1,986 questions and report both overall accuracy and per-category breakdown.

5 Results

5.1 LaMP-QA

Table 1 reports ROUGE-L scores across conditions. Condition C (structured only) achieves the highest score at 0.1389, followed by Condition B at 0.1380. The differences are small in absolute terms but consistent in direction: both structured conditions outperform the flat profile baseline, with Condition C marginally edging Condition B on this aggregate metric.

Table 1: LaMP-QA results (ROUGE-L, 100 samples). Both structured conditions outperform the flat profile baseline. Condition C marginally outperforms Condition B on aggregate scoring.

Condition	**ROUGE-L**
A (Flat Profile)	0.1312
B (Structured + Conflict Resolution)	0.1380
C (Structured Only)	**0.1389**

The gap between A and the structured conditions (0.006–0.008 ROUGE-L) reflects the value of Sim-based context organization: even without any change to the underlying information, structuring user context into semantically distinct blocks improves response quality on a personalized QA task. The narrow margin between B and C on ROUGE-L suggests that conflict-resolution instructions add modest aggregate benefit; their value is more apparent in settings where conflicting context actively degrades response quality.

5.2 LOCOMO

Table 2 reports span-containment accuracy by category. Overall, Condition C achieves the highest accuracy (3.17%), followed by Condition B (2.62%) and Condition A (2.37%). Absolute values are low across all conditions, consistent with LOCOMO's strict verbatim matching criterion applied to free-form generation.

Table 2: LOCOMO span-containment accuracy by category. Cat-3: multi-hop/cross-temporal reasoning. Cat-5: preference-based queries. Bold indicates best per column.

Condition	**Overall**	**Cat-1**	**Cat-2**	**Cat-3**	**Cat-4**	**Cat-5**
A (Flat Profile)	0.0237	0.0426	**0.0093**	0.0521	0.0190	0.0247
B (Structured + Conflict Res.)	0.0262	0.0355	0.0031	0.0833	0.0214	**0.0336**
C (Structured Only)	**0.0317**	**0.0319**	0.0031	**0.1042**	**0.0345**	0.0314

The category-level pattern is informative. Cat-3 (multi-hop reasoning) shows the largest absolute gains from structuring: Condition C reaches 10.42% vs. 5.21% for the flat profile, a near-doubling. Cat-5 (preference-based queries) is the one category where Condition B outperforms Condition C (3.36% vs. 3.14%), suggesting that conflict-resolution instructions specifically help when the query

is sensitive to competing personal preferences. For Cat-1 and Cat-2 (simpler fact retrieval), the flat profile performs competitively or better, consistent with the expectation that these tasks do not require multi-context reasoning.

6 Discussion

6.1 Structure Is the Primary Driver

Across both benchmarks, the consistent finding is that structured multi-context representation improves over flat profiles. The information content is identical across conditions; what changes is its organization. Separating user context into semantically defined Sim blocks — each carrying an explicit role description, priority list, communication style, and constraint set — gives the model a cleaner signal about what kind of information it is reading and how to weight it. On LaMP-QA, this produces a 5–6% relative ROUGE-L improvement. On LOCOMO, it nearly doubles accuracy on the most complex reasoning category.

This finding holds regardless of whether a conflict-resolution instruction is present. The representation design itself does most of the work.

6.2 When Conflict Resolution Helps

The B vs. C comparison isolates the effect of explicit conflict-resolution instruction. On aggregate metrics, the differences are small and inconsistent in direction: C edges B on LaMP-QA ROUGE-L and LOCOMO overall. But Cat-5 (preference-based queries) is a consistent exception — B outperforms C on LOCOMO Cat-5 across all runs.

This makes sense. Preference-based queries are exactly where Sim blocks are most likely to contain competing signals: a user's work preferences may suggest one answer while their health or family context suggests another. Telling the model explicitly how to reason about these conflicts — rather than leaving it to implicit inference — produces more coherent responses when the answer depends on correctly weighting competing preferences. For simpler retrieval tasks, the instruction adds no value and may introduce unnecessary hedging that slightly reduces span coverage.

The practical implication: conflict-resolution instructions should be applied selectively, not universally. This is precisely what the arbitration mechanism in the Sims architecture is designed to do — activate conflict reasoning when Sim activation profiles indicate competing relevance, suppress it when a single Sim clearly dominates.

6.3 The Metric Gap

Span-containment and ROUGE-L both have structural biases that partially obscure the value of structured personalization. Span-containment rewards any response that contains the answer string, regardless of whether it came from the contextually appropriate Sim or from incidental mention across unstructured context. ROUGE-L measures surface overlap with a reference response, which may itself reflect one particular personalization style rather than the user's actual preferences.

Neither metric penalizes responses that are contextually inappropriate — that surface the right answer through the wrong reasoning path. A system that correctly identifies which Sim is relevant and reasons from it faithfully will not necessarily outscore a system that dumps all context indiscriminately, if the latter happens to include the answer string somewhere in its output.

The community lacks evaluation infrastructure for this. Source attribution — tracking which Sim contributed to which part of a response — and interference scoring — penalizing responses that blend

irrelevant context — would give a more accurate picture of what structured personalization actually achieves. Sims makes the structure explicit enough that such evaluation is technically tractable; building the evaluation tooling is left for future work.

6.4 Applied Motivation and Privacy

The architecture described here was motivated by a concrete deployment constraint: users of personal AI systems should not have to surrender personal data to benefit from personalization. Sim construction happens on-device. The only information crossing a network boundary is the inference call to the base language model, which receives structured Sim context rather than raw personal data. The graduated disclosure property — where only task-relevant context reaches external services rather than the complete user profile — follows directly from the Sims representation design, independent of which inference condition is used.

On-device Sim-guided fine-tuning, where LoRA adapters are conditioned on active Sim contexts and trained locally without transmitting personal data, is the planned next component of the architecture and is reserved for future work.

7 Conclusion

Sims offers a structured alternative to flat user profiles for personalized language model systems. Separating user context into semantically defined, domain-specific blocks consistently improves response quality over undifferentiated profile concatenation — across two benchmarks, two evaluation metrics, and three query categories. The representation design is the primary driver; explicit conflict-resolution instructions provide targeted additional benefit on preference-sensitive tasks where competing Sim signals are most likely to arise.

These results also surface a measurement gap. Standard personalization benchmarks reward recall and surface overlap but do not penalize contextually inappropriate responses. Evaluation frameworks that incorporate source attribution and interference scoring would give a more accurate picture of what structured multi-context personalization actually achieves. Sims is a step toward making that evaluation tractable.

References

Bergthor Björnsson, Carl Borrebaeck, Nicolai Elander, Thomas Gasslander, Danuta R Gawel, Mika Gustafsson, Rebecka Jornsten, Eun Jung Lee, Xinxiu Li, Sandra Lilja, et al. Digital twins to personalize medicine. *Genome Medicine*, 12(1):1–4, 2019.

Michael Grieves. Digital twin: Manufacturing excellence through virtual factory replication. *White Paper*, 2014.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*, 2021.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems*, volume 33, pages 9459–9474, 2020.

Chin-Yew Lin. ROUGE: A package for automatic evaluation of summaries. In *Text Summarization Branches Out*, pages 74–81, 2004.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Luo. LoCoMo: A long context modular memory benchmark. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, pages 5375–5403, 2024.

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

Alireza Salemi and Hamed Zamani. LaMP-QA: A benchmark for personalized long-form question answering. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 1139–1159, Suzhou, China, 2025. Association for Computational Linguistics. doi: 10.18653/v1/2025.emnlp-main.60.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. LaMP: When large language models meet personalization. *arXiv preprint arXiv:2304.11406*, 2023.

Chirag Shah and Ryen W. White. Agents are not enough. *IEEE Computer*, 58(10):87–92, 2025. doi: 10.1109/MC.2025.3575829.

Chia-Yuan Tang, Kuei-Chun Chang, Ryan A Rossi, Sungchul Kim, Tong Yu Chen, Franck Dernoncourt, Ruiyi Wang, and Jiuxiang Yoon. MAIN-RAG: Multi-agent filtering retrieval-augmented generation. *arXiv preprint arXiv:2501.00332*, 2025.

Saizheng Zhang, Emily Dinan, Jack Urbanek, Arthur Szlam, Douwe Kiela, and Jason Weston. Personalizing dialogue agents: I have a bike. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics*, pages 2112–2121, 2018.

A Experimental Details

LaMP-QA. We sampled 100 instances uniformly at random from the LaMP-QA test set. For each instance, user profile information was organized into four Sim blocks (`work_self`, `family_self`, `hobby_self`, `health_self`) based on the semantic content of past interactions. For Condition A, profile text was merged into a single unstructured block. ROUGE-L was computed using Google's `rouge-score` Python library with default tokenization.

LOCOMO. We evaluated all 1,986 questions using the standard LOCOMO span-containment protocol. Responses were generated with GPT-3.5-turbo (temperature 0.7, max tokens 500). Condition A used the conversation history as a flat concatenated block. Conditions B and C organized the same history into structured Sim blocks corresponding to the semantic roles present in LOCOMO's multi-turn conversation structure. Evaluation used the benchmark's official `evaluate_predictions` function.