TSUBASA: Improving Long-Horizon Personalization via Evolving Memory and Self-Learning with Context Distillation

Anonymous ACL submission

Abstract

Personalized large language models (PLLMs) have garnered significant attention for their ability to align outputs with individuals' needs and preferences. However, they still struggle with long-horizon tasks, such as tracking a user's extensive history of conversations or activities. Existing memory mechanisms often fail to capture evolving behaviors, and RAG paradigms are trapped by a *quality-efficiency tradeoff*. Meanwhile, parametric adaptation is bottlenecked by *train-inference gap* due to the scarcity of labeled data. To enhance the long-horizon capabilities of PLLMs, we introduce TSUBASA, a two-pronged approach designed to improve *memory writing* via dynamic memory evolution, and *memory reading* via self-learning with a context distillation objective to internalize user experiences. Extensive evaluations on long-horizon benchmarks using the QWEN model family (4B to 32B) validate the effectiveness of TSUBASA, surpassing competitive memory-augmented systems that rely primarily on memory writing, such as Mem0 and Memory-R1. Our analyses further confirm that TSUBASA breaks the quality-efficiency barrier to achieve Pareto improvements, delivering robust, high-fidelity personalization with a reduced token budget.[1]

1 Introduction

In recent years, large language models (LLMs) have evolved from passive text generators into versatile reasoning engines capable of analyzing diverse data modalities and performing deep reasoning (Chen et al., 2025; Comanici et al., 2025). However, despite their vast parametric knowledge, generic LLMs often follow a *one-size-fits-all* approach that fails to cater to individuals' nuanced, idiosyncratic requirements (Liu et al., 2025; Qiu et al., 2025; Duran and Aytekin, 2025). To bridge this gap, personalized LLMs (PLLMs) utilize user context and parametric adaptation to align outputs with individualized needs and specific user personas (Salemi et al., 2024; Zhang et al., 2025).

Recent advancements in PLLMs span prompt engineering (Park et al., 2023; Richardson et al., 2023), retrieval-augmented generation (RAG; Salemi et al., 2024; Mysore et al., 2024), lightweight parameterization (Zhang et al., 2024b; Magister et al., 2025), and hybrid *dual-memory* models that mirror human episodic and semantic memory (Zhang et al., 2025). Despite these efforts, existing methods fundamentally struggle with **long-horizon personalization** due to their inability to *grasp continuously evolving user preferences and a critical lack of labeled user data for parametric memory updates*. Concretely, standard episodic memory systems often act as linear-growing vector databases (Lewis et al., 2020; Madaan et al., 2022), failing on "reflective" tasks that require implicit temporal reasoning across interactions (Tan et al., 2025). While recent work explores evolving memory to address this (Nan et al., 2025; Wang et al., 2025b; Luo et al., 2026), they remain flawed: some compromise production privacy through cross-user training (Yan et al., 2025), while others rely on destructive DELETE operations (Chhikara et al., 2025) that erase historical context and obscure the cause-and-effect behind user evolution. Furthermore, internalizing memory via parameterization (Zhang et al., 2024b; Magister et al., 2024) requires abundant, hard-to-acquire labeled data.

To advance the memory mechanisms and unlock deep personalization for long-horizon tasks, PLLMs must achieve both **native personalized reasoning** and **access to episodic memory for factual grounding**. However, attempting to solve these high-level objectives with current architectures exposes two technical bottlenecks (Section 3). First, a *train-inference gap*: standard fine-tuning on raw conversations fails to prepare parametric memory for the complex personalization tasks at

[1] Codes will be released upon acceptance.

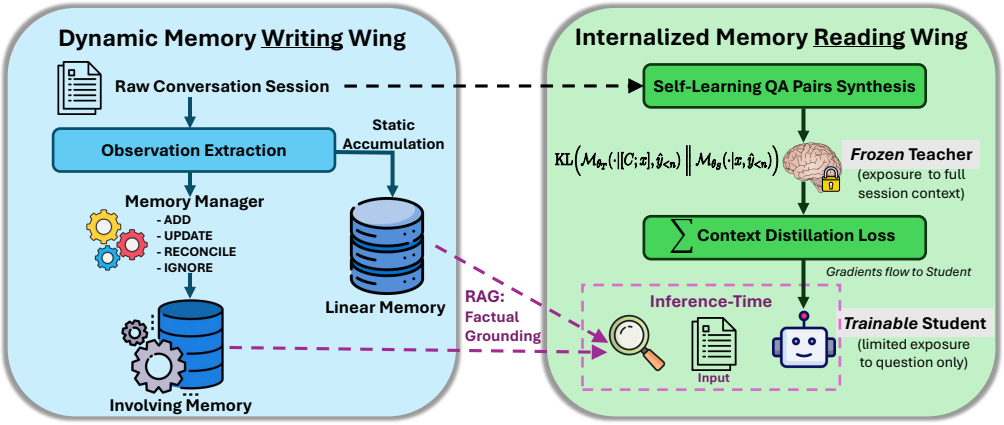

Figure 1: Overview of TSUBASA framework, built on two synergistic wings. **Dynamic memory writing** applies structured algorithmic evolution based on high-density observations distilled from raw utterances (Section 4.1). **Internalized memory reading** adopts the self-learning pipeline and applies context distillation objective on synthetic data (Section 4.2). During Inference, observations are retrieved for factual grounding.

evaluation. Second, a *quality-efficiency tradeoff*: standard RAG paradigms correlate better response quality with larger volumes of retrieved memories, incurring prohibitive computational costs.

To this end, we introduce TSUBASA framework (*lit.* *"Wing"* in Japanese; Figure 1), standing for **T**wo-winged per **S**onalization **U**nifying **B**i-Directional **A**utonomous memory **S**torage and parametric **A**ssimilation. We posit that an effective memory system comprises two synergistic wings (**writing** and **reading**), both essential for genuine personalization. Specifically, during *dynamic memory writing* (Figure A7), core, factual **observations** are extracted from raw conversations, and then a memory manager *autonomously* updates the memory store to prevent redundancy and resolve conflicts. During *internalized memory reading*, we employ a self-learning pipeline to tackle the lack of labeled user-specific data and the knowledge gap inherent in naive autoregressive training. We augment this with a teacher-student context distillation objective (eq. (1); Snell et al., 2022) on *synthetic user data*, enabling the teacher to distill rich "dark knowledge" about the user directly into the student.

We conduct extensive empirical evaluations on LoCoMo (Maharana et al., 2024) and LongMemEval (Wu et al., 2025a) to assess PLLMs in long-horizon conversational settings. Our results validate the high information density of extracted observations, the efficacy of internalized memory reading, the necessity of explicit RAG-backed grounding, and the power of our evolving memory algorithm especially when the model size scales. Remarkably, in terms of F1 results, TSUBASA-PRO demonstrates substantial gains: it outperforms widely used systems like Mem0 (Chhikara et al., 2025) by 37% (53.3 $\rightarrow$ 73.08) using LLM-as-a-judge, and surpasses the prior SOTA, Memory-R1 (Yan et al., 2025), by 19%, without training on cross-user data. Furthermore, our analysis confirms that TSUBASA breaks the quality-efficiency barrier to achieve strict *Pareto improvements*.

In summary, our contributions are three-fold:

- We identify two critical bottlenecks trapping existing PLLMs: the train-inference gap in semantic parameterization and the quality-efficiency tradeoff for episodic retrieval.
- We propose TSUBASA, a novel framework for long-horizon tasks featuring two synergistic wings: dynamic memory writing and internalized memory reading.
- Extensive experiments showcase the effectiveness of TSUBASA, highlighting its ability to achieve strict Pareto improvements.

2 Related Work

2.1 LLM Personalization

The pursuit of adapting models to human nuances has driven extensive research (Purificato et al., 2024). Early efforts evolved from explicit modeling of user profiles and demographic attributes (e.g., age, gender; Amato and Straccia, 1999; Fischer, 2001; Kim et al., 2013; Gao et al., 2013; Gou et al., 2014) to latent-factor models and collaborative filtering that map user-item interactions into low-dimensional embeddings (Koren et al., 2009; Jiang et al., 2014; He et al., 2017; Fareed et al., 2023). While Transformer (Vaswani et al., 2017) enabled learnable user embeddings (Qiu et al., 2021; Deng et al., 2023), these approaches struggle to leverage *unstructured* text or generalize across diverse tasks, necessitating frameworks capable of deeper semantic alignment with raw textual histories.

With the advent of LLMs, initial efforts relied on prompt engineering for static personas (Park

et al., 2023; Tang et al., 2024; Tseng et al., 2024) or profile summaries (Richardson et al., 2023), often yielding superficial, surface-level stylistic mimicry. To improve robustness, systems began incorporating entire interaction histories (Liu et al., 2023; Petrov and Macdonald, 2023; Kang et al., 2023). To overcome context window limits and "lost-in-the-middle" issue (Liu et al., 2024), retrievers like BM25 (Robertson and Zaragoza, 2009) and FAISS (Douze et al., 2024) were adopted to only keep relevant items (Madaan et al., 2022; Salemi et al., 2024; Mysore et al., 2024). Yet, noisy retrieval still restricts fine-grained personalization. As such, recent studies internalize historical engagement directly into model weights via lightweight adapters (Tan et al., 2024a,b; Zhang et al., 2024b; Magister et al., 2025). Inspired by cognitive theories (Tulving et al., 1972; Tulving, 1985), unified frameworks like PRIME combine RAG-based discrete episodic memory with parameterized semantic memory to further enhance capabilities (Zhang et al., 2025).

Despite this progress, existing unified, dual-memory architectures face two critical limitations. First, discrete episodic memory struggles to capture the *evolving* nature of user preferences, which is integral to long-horizon tasks (Wei et al., 2025; Chhikara et al., 2025). Second, parametric memory updating typically demand *labeled* user data (Zhang et al., 2024a; Salemi et al., 2024), which is frequently unavailable in practice. Thus, our goal is to advance the dual-memory architecture to address these specific challenges in long-horizon setting.

2.2 Memory Mechanisms for Personalization

Decades of psychological research have established core human memory components, ranging from short-term to durable long-term memory, which is further divided into episodic and semantic memory structures (Atkinson and Shiffrin, 1968; Tulving et al., 1972; Tulving, 1985). Imitating these mechanisms is central to LLM personalization, acting as an architectural cornerstone to overcome the inherent *statelessness* of plain LLMs (Zhang et al., 2024e; Wu et al., 2025b; Du et al., 2025). Traditional methods rely on brute-force caching of raw conversational logs, but this *static accumulation* fails catastrophically on "reflective" tasks requiring implicit reasoning across interactions (Tan et al., 2025). Consequently, the field has witnessed a rapid shift towards autonomously evolving memory systems that utilize reflection and abstraction to capture dynamic user preferences (Wang et al., 2024; Luo et al., 2026). These techniques range from static summarization with decay of older, less relevant memories (Zhong et al., 2024; Salama et al., 2025) to truly dynamic approaches executing autonomous, incremental updates (Kim et al., 2025; Nan et al., 2025; Xu et al., 2025; Rasmussen et al., 2025; Wang et al., 2025b).

A critical divergence in memory evolution algorithms is *how* memory updates (e.g., ADD, UPDATE, DELETE, NOOP) are executed. To perform these actions, frameworks like Memory-R1 (Yan et al., 2025) and Mem-α (Wang et al., 2025b) train dedicated managers via PPO (Schulman et al., 2017) or GRPO (Shao et al., 2024), while Mem0 (Chhikara et al., 2025) prompts proprietary models like GPT-4 (Achiam et al., 2023). However, these approaches introduce severe bottlenecks: RL pipelines are notoriously unstable and demand domain-specific reward models, whereas proprietary APIs incur prohibitive costs and latency. This makes them unsuitable for portable, on-device deployments with strict computational limits (Qin et al., 2024; Wang et al., 2025a). Furthermore, existing frameworks mishandle evolving preferences by using a destructive DELETE operation (Yan et al., 2025; Chhikara et al., 2025), which erases historical context and obscures the cause-and-effect of users' mindset shifts. To address this, we introduce a novel RECONCILE action and incorporate temporal signals to build coherent narratives, preserving the absolute integrity of the user timeline.

3 Prelim: Challenges of Memory Usage

To motivate our solution, we first rigorously examine the performance bottlenecks inherent in prevailing approaches. Specifically, we evaluate the LOCOMO dataset (Maharana et al., 2024) using the best-performing, portable LLMs (QWEN3; Yang et al., 2025), to highlight two fundamental challenges in *how answer agents utilize memory*.

Memory Structure and the Train-Inference Gap. The first fundamental challenge concerns how to *instantiate memory*. An intuitive baseline to internalize user context is to naively fine-tune the LLM using an autoregressive next-token prediction objective over the raw conversational history (i.e., *semantic memory usage*). However, empirical results (Table 1) demonstrate a counter-intuitive reality: this baseline actually underperforms an untrained LLM augmented with RAG (i.e., *episodic memory usage*). This discrepancy stems from a non-trivial

Method	QWEN2.5-7B			QWEN3-8B			Input Len. (tokens)
	F1	BLEU-1	Rouge-L	F1	BLEU-1	Rouge-L	
RAG (k=1)	21.97	17.87	21.46	21.26	17.06	20.82	29.4
RAG (k=3)	27.75	22.55	27.06	28.10	22.53	27.30	75.4
Train	6.48	4.55	6.14	6.07	4.32	5.73	5.6
Session	41.49	34.13	40.42	41.22	33.64	39.61	2,158.5

Table 1: Preliminary study reveals *train-inference gap* and *quality-efficiency tradeoff* as discussed in Section 3. Similar trend is observed on QWEN2.5-14B and QWEN3-14B models, shown in Table A1.

train-inference gap. During naive autoregressive training, the model optimizes for the rote memorization of conversations, including irrelevant chatter. Yet, at inference time, complex personalization tasks (e.g., multi-hop QA; Maharana et al., 2024) demand deeper reasoning to piece together temporally scattered facts. Meanwhile, relying solely on RAG to retrieve raw utterances is equally *flawed*. Raw dialogues contain massive semantic noise and repeated information, are temporally fragmented, and act merely as external prompts that do not contribute to genuine, internalized personalization.

The Quality-Efficiency Tradeoff. The second major challenge arises from the inherent tradeoff between inference efficiency and response quality. In the standard RAG paradigm, efficiency is strictly bottlenecked by the volume of retrieved memories (Madaan et al., 2022; Salemi et al., 2024; Mysore et al., 2024; Zhong et al., 2024). As shown in Table 1, both QWEN3-8B and QWEN3-14B exhibit the same pattern: full *session*[2] context exposure consistently yields the best results but consumes an average of $2,158.5$ tokens per query. Conversely, restricting the context window to only the single most relevant memory piece dramatically improves inference speed (29.4 tokens) but significantly degrades accuracy. In fact, real-world applications demand PLLMs that are both quality and efficient, this tradeoff highlights an imperative need for architectural advancement: the length of the historical context must be explicitly decoupled from inference-time compute, while the LLM should still retain the full access to the historical knowledge without the associated latency costs.

Next, we describe our proposed solution, TSUBASA, where the dynamic memory writing module addresses the *raw utterance usage* issue (i.e., *low information density, high redundancy, and lack of temporal awareness*), and the internalized memory reading module tackles both *train-inference gap* and *quality-efficiency tradeoff*.

[2]A session is an ordered sequence of interactions between two participants within a short, temporally bounded interval.

4 Methodology: TSUBASA Framework

TSUBASA (Figure 1) is constructed upon two synergistic wings: *dynamic memory writing* (Section 4.1) via structured algorithmic evolution, and *internalized memory reading* (Section 4.2) through self-learning with context distillation objective.

4.1 Dynamic Memory Writing: From Raw Storage to Evolving Observation

To address the aforementioned *raw utterance* issue, our memory writing mechanism first extracts core, factual **observations** from raw conversations before storage, analogous to the hierarchical event extraction processes (Zhang et al., 2024d). Figure A5 in appendix details the instruction used to extract these salient observations from a session. This process **improves the information density**, **reduces intra-session redundancy** and significantly cuts the memory usage for each incoming session (e.g., store 20 raw utterances vs. 7 salient observations on LOCOMO dataset (Maharana et al., 2024)).

Moving beyond static accumulation, which struggles with "reflective memory" tasks requiring implicit reasoning across temporally-related interactions (Tan et al., 2025), we implement a continuous, evolutionary memory manager. To mitigate **inter-session redundancy**, new observations are not indiscriminately appended. Instead, the management agent evaluates the semantic and temporal relationship between incoming observations and the existing memory profile. Guided by evolution rules defined in a system prompt (Figure A6), the manager autonomously executes one of four actions to combat linear memory growth:

- **ADD:** Creates a new entry for entirely novel information without duplicating existing ones.
- **UPDATE:** Modifies the existing memory to reflect granular details or a more recent status, avoiding unnecessary expansion.
- **RECONCILE:** Integrates *conflicting* information into a coherent temporal narrative, preserving complete historical context while preventing retrieval confusion. This operation explicitly accounts for the inherently non-stationary nature of human preferences and behaviors (Hoch and Loewenstein, 1991; Hughes et al., 2020).
- **IGNORE:** Discards trivial or redundant information already captured in memory.

Notably, our RECONCILE action improves upon the destructive DELETE found in existing frameworks (i.a., Yan et al., 2025; Chhikara et al.,

2025). When confronted with conflicting preferences, purging "outdated" facts irreparably compromises the user's engagement timeline and the cause-and-effect behind the user's evolving mindset. Instead, RECONCILE weaves conflicting information into a coherent, self-contained narrative. To ensure these narratives remain robust, we further augment with explicit **temporal grounding** (Tan et al., 2023; Zhang et al., 2024c; Piryani et al., 2025). In raw conversations, temporal references are primarily relative (e.g., "yesterday"), which quickly lose context in vector databases and corrupt retrieval quality afterwards. Thus, our dynamic writing module autonomously decodes them into *absolute* calendar dates. By implementing *RECONCILE* and *temporal awareness*, each memory piece now functions as a temporally grounded, conflict-free narrative (e.g., "The user previously preferred X, but as of [Date], prefers Y"). This preserves the evolutionary trajectory of the user's behavior and provides the downstream LLM with an interpretable timeline optimized for precise retrieval.

Practically, the same underlying model performs both memory writing and reading (Section 4.2). By avoiding costly proprietary LLMs (Maharana et al., 2024; Chhikara et al., 2025) and refraining from training on other users' data (Yan et al., 2025; Wang et al., 2025b), this approach paves the way for portable, on-device PLLM deployment.

4.2 Internalized Memory Reading

While the dynamic memory writing produces a dense, redundancy-free, and temporally grounded external memory, off-the-shelf LLMs treat retrieved observations as disconnected *"foreign texts"* instead of intrinsic personalization priors. To realize genuine personalization, we must *internalize the user's context into the model's parameters* while utilizing episodic observations for precise factual grounding. This mirrors cognitive offloading (Risko and Gilbert, 2016; Sweller, 1988), akin to a student internalizing core schemas while referencing a cheat sheet for specific formulas.

To achieve this without the prohibitive inference costs of full-context exposure or the train-inference gap of naive autoregressive training (Section 3), TSUBASA employs a self-learning pipeline that curates synthetic user data to compress extensive user experiences directly into LLM's parameters (Section 4.2.1). We augment this with a teacher-student context distillation strategy (Section 4.2.2; Snell et al., 2022; Choi et al., 2023) to deeply instill user's context. While context distillation was originally designed to internalize detailed prompt instructions for reduced latency, to our knowledge, we are the first to repurpose it to overcome the *data scarcity* and computational bottlenecks inherent in deep personalization (Zhang et al., 2025).

4.2.1 Self-Learning on Synthetic Data

Because user-specific labeled data is inherently scarce, existing models often rely on cross-user data (Yan et al., 2025; Wang et al., 2025b), raising significant privacy concerns. We bypass this by employing a self-learning paradigm (Zhang et al., 2019; Pham et al., 2022; Lee et al., 2025), where TSUBASA is trained on synthetic question-answer (QA) pairs $\{(x, y)\}$, exclusively generated from the target user's context by itself.

Specifically, we prompt the LLM to generate 5W1H-based QA pairs for each conversational session, ensuring questions are meaningful and answers remain concise and specific. Next, we follow QA literature and apply a standard data filtering pipeline to improve the synthetic data quality by removing: 1) trivial questions where the answer explicitly appears in the question; 2) non-appropriate QA pairs (Section A); and 3) non-cycle-consistent QA pairs (Alberti et al., 2019) where the synthesized answer y deviates from the teacher's output $\hat{y}$.[3] This process produces a user-specific QA dataset $\mathcal{D}$ on-the-fly for subsequent context distillation (Section 4.2.2). Notably, self-learning on our curated $\mathcal{D}$ **bridges the train–inference gap by** enabling the benefit of $\mathcal{M}_{\theta_S}$ internalizing personalization signals through *"learn to answer"* instead of merely rote memorization of past conversations.

4.2.2 Context Distillation

During distillation, the *teacher* $\mathcal{M}_{\theta_T}$ processes the full, uncompressed conversation history C alongside the target question x, to generates a highly personalized response $\hat{y}$. The *student* $\mathcal{M}_{\theta_S}$, **utilizing exactly the same architecture**, is strictly constrained to see the question x only during training without any retrieved memory. Formally, keeping the teacher's parameters θ_T *frozen*, we optimize the student's parameters θ_S to match the teacher's output distribution $\mathcal{M}_{\theta_T}(\cdot|[C; x])$ via a per-token KL divergence objective (Cover, 1999) as follows:

$$\mathcal{L}_{\mathcal{D},\theta_T}(\theta_S) = \mathbb{E}_{(x,y)\sim\mathcal{D}}\Bigg[\mathbb{E}_{\hat{y}\sim\mathcal{M}_{\theta_T}(\cdot|x)}\sum_{n=1}^{|\hat{y}|} \mathrm{KL}\Big(\mathcal{M}_{\theta_T}(\cdot|[C; x], \hat{y}_{<n}) \,\Big\|\, \mathcal{M}_{\theta_S}(\cdot|x, \hat{y}_{<n})\Big)\Bigg] \tag{1}$$

[3] We discard pairs if the semantic similarity is below 0.5.

Method	Qwen3-4B			Qwen3-8B			Qwen3-14B			Qwen3-32B		
	F1	BLEU-1	ROUGE-L	F1	BLEU-1	ROUGE-L	F1	BLEU-1	ROUGE-L	F1	BLEU-1	ROUGE-L
RAG (utterance)	28.53	23.20	27.66	28.10	22.53	27.30	31.38	26.02	30.59	28.64	23.19	27.83
RAG (observation)	31.17	25.50	30.31	31.77	25.91	30.63	34.83	28.79	33.43	34.00	28.00	32.56
Vanilla Training	5.76	4.02	5.53	6.07	4.32	5.73	6.59	4.96	6.24	8.28	6.22	7.91
TSUBASA (no grounding)	12.26	9.50	11.84	11.99	9.33	11.67	14.71	11.90	14.41	14.81	11.27	14.37
TSUBASA (static accumulation)	**32.82**	**27.21**	**31.96**	**34.50**	**29.25**	**33.40**	37.07	31.25	35.65	32.75	26.18	31.46
TSUBASA (evolving memory)	31.38	26.23	30.40	33.99	28.92	32.79	**37.63**	**32.16**	**36.16**	**34.25**	**28.12**	**32.85**

Table 2: Main results of TSUBASA and baselines on LoCoMo dataset (average of 3 runs), isolating module contributions. For each base model, the best results are in **bold**, while global optimal performances are underlined. The results demonstrate information density of observations, usefulness of internalized memory reading, necessity for factual grounding, and efficacy of evolving memory. Complete results are shown in Table A3.

where $|\hat{y}|$ is the teacher's response length, $[\cdot;\cdot]$ is the textual concatenation operator, and $\hat{y}_{<n}$ is the tokens generated up to step n. Since computing the exact full-vocabulary per-token KL divergence is intractable, we approximate it on the *top-d tokens*, and we analyze the effect of d in Section 6.3.

By aligning the student's probability space with the teacher's informed distribution, the student systematically learns to internalize the user experiences and preferences embodied in $\mathcal{D}$ into its own weights, even though *it never explicitly receives the target user's exact conversation history during training*. This elegantly **resolves the quality-efficiency tradeoff** by decoupling the historical context length from the inference-time compute.

5 Experiments

Datasets and Tasks. We conduct a holistic evaluation of PLLMs using the LoCoMo dataset (Maharana et al., 2024). LoCoMo comprises long-term conversations spanning several months across 10 participant pairs. On average, each pair engages in 19.3 sessions and 304.9 turns. Following Chhikara et al. (2025); Yan et al. (2025), we restrict our evaluation to the last eight participant pairs to ensure comparisons. This evaluation set contains $1,294$ QA items.[4] We further evaluate against SOTA models on LongMemEval (Wu et al., 2025a), a benchmark of extremely long conversations between user and AI assistant, with an average of 47.7 sessions and 493.4 turns per user-assistant pair. This evaluation set contains 500 QA items for 500 users.

Data creators confirm their datasets are free from personally identifiable info or offensive contents.

Setup. We primarily instantiate TSUBASA using recent open-weight LLMs, QWEN3 (Yang et al., 2025) *without thinking*. To ensure a comprehensive analysis, we experiment with models from 4B to 32B parameters. Our main results utilize $k = 3$ retrieved memory entries, and the retrieval is conducted by QWEN3-EMBEDDING-8B . During inference, we perform greedy decoding for reproducibility. Training details are listed in Section B.

We benchmark against *existing baselines* using both QWEN2.5-7B (Hui et al., 2024) and QWEN3-8B, since the SOTA—Memory-R1 (Yan et al., 2025)—also evaluates on QWEN2.5-7B. Notably, we report results for TSUBASA-PRO, which uses a more robust inference prompt and optimized hyper-parameters (detailed in Section C).

Practically, the same underlying model performs both memory writing and reading. Memory writing operation distribution is included in Table A2.

Baselines and TSUBASA Variants. We evaluate TSUBASA against two vanilla RAG baselines (retrieving raw utterances vs. extracted observations) and two training-based baselines (standard fine-tuning on raw conversations vs. a RAG-free variant of TSUBASA). Within the TSUBASA framework, we compare *static accumulation* with *evolving memory*, both built upon observations.

Further, we compare TSUBASA with several established baselines for long-horizon reasoning: LoCoMo (Maharana et al., 2024), A-Mem (Xu et al., 2025), Mem0 (Chhikara et al., 2025), and Memory-R1 (Yan et al., 2025).

Evaluation Metrics. We employ six quality metrics: token-level Precision (P), Recall (R), and F1-score, alongside BLEU-1, ROUGE-L, and LLM-as-a-Judge (J). Except for J, these metrics quantify the lexical overlap between the generated and ground-truth answers. Following Yan et al. (2025), we utilize GPT-5-nano (Singh et al., 2025) for the J metric to assess semantic correctness and contextual appropriateness. *J is only used for SOTA comparison due to budget*. We also report inference efficiency measured by input token length.

6 Results

6.1 Main Results and Systematic Evaluation

We systematically evaluate TSUBASA across multiple scales to isolate module contributions. Major results are included in Table 2, and complete results

[4] While the initial set contains $1,307$ QA pairs, we filter out questions whose answers cannot be grounded in conversations.

Method	Single Hop	Multi-Hop	Open Domain	Temporal	Overall
LoCoMo (Maharana et al.)	15.06	19.28	12.79	5.43	12.17
A-Mem (Xu et al.)	40.78	31.32	46.90	28.68	40.78
Mem0 (Chhikara et al.)	61.92	48.19	65.20	38.76	53.30
MemoryOS (Kang et al.)	48.12	38.55	63.14	24.81	51.26
Memory-R1-PPO (Yan et al.)	57.74	53.01	66.99	42.25	59.53
Memory-R1-GRPO (Yan et al.)	62.34	40.96	**67.81**	**49.61**	61.51
TSUBASA-PRO (QWEN2.5-7B)	**82.64±0.49**	80.93±0.92	60.09±1.64	43.36±0.55	**73.08±0.30**
TSUBASA-PRO (QWEN3-8B)	81.68±0.11	**82.20±0.35**	62.72±1.24	42.19±0.32	72.46±0.04

Table 3: LLM-as-a-judge evaluation of TSUBASA-PRO (average of 3 runs) compared to prior SOTA baselines on the LoCoMo benchmark. Baseline results are directly sourced from Yan et al. (2025). Best numbers are in **bold**. Notably, TSUBASA-PRO consistently surpasses all baselines in the *overall* metric, including the highly competitive Memory-R1. Complete results including lexical-match metrics are shown in Table A4.

(including all quality metrics and inference-time efficiency) are in Table A3. Our major findings are:

- **Information Density of Observations:** `RAG (observation)` consistently outperforms `RAG (utterance)`. This validates that compressing raw, noisy utterances into core factual observations creates higher-density retrieval targets, significantly enhancing response quality while reducing token budgets.
- **Efficacy of Internalized Memory Reading:** Our *self-learning* pipeline effectively bridges the train-inference gap observed in `Vanilla Training` (Section 3). Further, KL-divergence-based *context distillation* provides richer supervision to capture nuances than sparse cross-entropy, enabling more genuine personalization.
- **Necessity of Explicit Grounding:** Removing access to external episodic memory causes severe performance collapse. The performance decline in both `Vanilla Training` and TSUBASA `(no grounding)` confirms that parameterization alone to store user context is insufficient—explicit grounding is imperative for *high-fidelity* personalization.
- **Impact of Evolving Memory:** Employing a memory manager to autonomously govern evolution yields peak quality scores (e.g., 37.63 F1)[5] while minimizing input lengths. However, this manager requires sufficient model capacity, realizing its full potential *only beyond 8B params*.

Scaling Saturation at 32B. While TSUBASA significantly outperforms baselines at 4B-14B scales, we observe a saturation effect at 32B. To our knowledge, this scale-dependent saturation, consistent across all evaluated metrics, is a *new empirical finding* in personalization work. We attribute this to the stronger intrinsic reasoning and in-context learning of larger models, which allows them to robustly process retrieved observations without treating them as *"foreign texts"*. Consequently, the necessity of internalized memory decreases, revealing a principle of *diminishing returns for memory internalization as base model capacity increases*.

[5]While not universally best, evolving memory resolves temporally conflicting information and shifting preferences.

6.2 Comparison with SOTA Baselines

As shown in Table 3, TSUBASA-PRO outperforms existing baselines on LoCoMo benchmark, particularly in explicit *fact-retrieval* tasks like single/multi-hop reasoning. However, we observe a performance degradation in the *Open Domain* task (e.g., connecting a preference for classical music to a specific composer *not mentioned* in the context). This bottleneck is a direct consequence of our synthetic data generation pipeline, which is optimized for *factoid QA reasoning* rather than *abstract reasoning*. We acknowledge this limitation and advocate for future synthetic data generation pipeline to include more complex commonsense reasoning QA pairs to further bridge the gap.

Importantly, TSUBASA reaches the best overall J of 73.08, significantly surpassing the competitive baseline Memory-R1 as well as industry standards like Mem0. Notably, unlike Memory-R1, which suffer from a serious flaw by training on other users' data, TSUBASA operates strictly within the target user's portfolio. This ensures robust performance while maintaining strict per-user data isolation.

Table A5 presents the LongMemEval results. TSUBASA-PRO attains performance comparable to Memory-R1 while using only 3 retrieved memory pieces compared to Memory-R1's 60. Meanwhile, it also surpasses the popular industry system Mem0 by an average of over 20% improvements across three quality metrics, further demonstrating the strong competitiveness of TSUBASA.

6.3 Further Analyses

Pareto improvement: Breaking the Quality-Efficiency Tradeoff. Traditional retrieval methods often force a compromise between response quality and latency costs of long contexts. To evaluate how TSUBASA addresses this, we analyzed response quality—F1 (Figure 2), BLEU-1 (Fig-

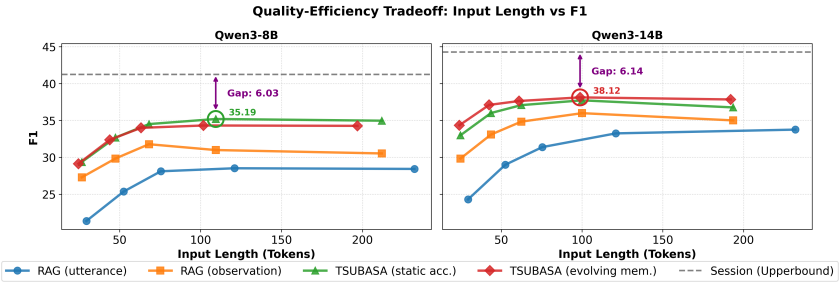

Figure 2: Quality-Efficiency tradeoff between input length and F1 metric on baselines and TSUBASA. Best-performing configuration is highlighted in circle with its gap to the established ceiling (`Session`) indicated. Importantly, our TSUBASA achieves *Pareto improvement*: attaining higher peak performance while utilizing only a fraction of the context budget. Tradeoffs w.r.t. ROUGE-L and BLEU-1 are presented in Figure A1 and Figure A2.

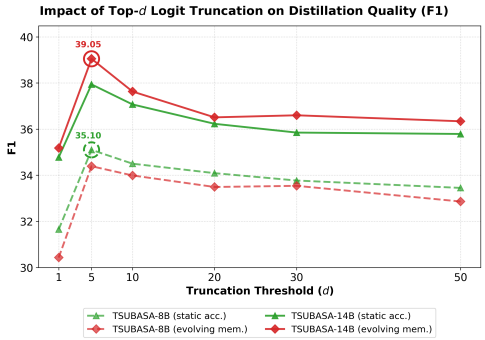

Figure 3: Impact of top-d truncation for approximating KL divergence on distillation quality (F1) with TSUBASA variants. Best-performing config is highlighted in circle for each model size. Importantly, we find $d = 1$ inadequate, while increasing d beyond 5 leads to performance degradation due to noises in long-tail distribution, establishing $d = 5$ at the sweet spot. See Figure A3 and Figure A4 for impacts on ROUGE-L and BLEU-1.

ure A2), and ROUGE-L (Figure A1)—against input lengths for $k \in \{1, 2, 3, 5, 10\}$ retrieved memory pieces. As shown in Figure 2, TSUBASA breaks this tradeoff and achieves a *Pareto improvement*, reaching higher peak performance while utilizing only a fraction of the context budget. For example, with QWEN3-14B, `TSUBASA (evolving)` at $k = 2$ achieves an F1 score of 37.10 with only 42.5 tokens. This easily surpasses the $k = 10$ configuration of `RAG (utterance)`, which requires 232.08 tokens to reach a 33.75 F1 score, echoing the importance of *information density* in retrieval targets as discussed in Section 6.1.

While reasoning with the full session history (`Session`) represents a performance ceiling, it incurs a prohibitive cost of $2,158$ tokens. TSUBASA effectively closes this gap: by $k = 5$ (~ 100 tokens), it captures the majority of relevant context, approaching the ceiling with $20\times$ fewer tokens.

Sweet-Spot of Top-d Truncation for Context Distillation Computing exact KL divergence over the entire vocabulary[6] for our context distillation objective is intractable. We approximate this by restricting the calculation to the top-d tokens, sorted by probability mass. Evaluating truncation thresholds $d \in \{1, 5, 10, 20, 30, 50\}$, we observe a non-monotonic relationship with response quality (Figure 3). This establishes $d = 5$ as the sweet spot and reveals two insights:

- **Inadequacy of Hard Labels ($d = 1$):** Truncating at $d = 1$ yields the worst performance, as KL divergence collapses into cross-entropy over the argmax token, *a standard fine-tuning baseline*. This highlights the necessity of soft labels, as the probability distribution among top candidate tokens contains rich semantic nuances that better guide the student's learning.
- **Long-Tail Noise Corrupts Learning ($d \geq 10$):** Counter-intuitively, performance steadily declines as $d > 5$. While a wider distribution seemingly captures more knowledge, the vocabulary's long tail is usually populated with noisy or irrelevant tokens (Li et al., 2025). Thus, aggressive truncation at $d = 5$ acts as a vital regularizer, filtering out teachers' noise while preserving essential dark knowledge.

7 Conclusion

In this work, we present TSUBASA to tackle the long-horizon personalization task. The framework of TSUBASA is motivated by limitations of current memory-augmented PLLMs: low-information-density retrieval targets in memory, train-inference gap, and quality-efficiency tradeoff. Specifically, TSUBASA features two *wings*: *dynamic memory writing* via structured algorithmic memory evolution on the basis of factual, information-dense observations, and *internalized memory reading* through self-learning pipeline with context distillation objective. We conduct extensive experiments to validate the design choice and quantify the contribution by each module. Benchmarking shows that TSUBASA surpasses significantly prior SOTA models in both quality and efficiency on LoCoMo.

[6] For `Qwen3`, the vocab size exceeds $150k$.

Limitations

Evaluation benchmarks. In this work, we have included two evaluation benchmarks, aiming to cover a diverse array of long-horizon personalization tasks. Yet, these two benchmarks cannot comprehensively represent the entire spectrum. For future research, we plan to extend TSUBASA to more applications, and examine its true generalizability in the wild.

Model Scales. In this study, we evaluate a diverse set of models ranging from $4B$ to $32B$ parameters. Owing to budget constraints, we do not extend our experiments to larger-scale models. Nonetheless, we expect our findings to hold on such models, and we have also discussed potential saturation effect post-$32B$ (Section 6.1). We leave the direct application of TSUBASA to larger models in the future work.

GPU resources. The base LLMs used in this work are of 4 to 32 billions parameters. It is thus more time-consuming than traditionally small models like BERT (Devlin et al., 2019) at both training and inference time, which in turn results in a higher carbon footprint. Specifically, for training, we run each base model of size $4B$ to $14B$ on 2 NVIDIA L40, and the $32B$ model on 2 NVIDIA RTX PRO 6000. For inference, we run TSUBASA on 1 single NVIDIA L40. All come with significant CPU and memory resources. The training time is generally 1-2 minutes per epoch. The inference time for each LLM on each benchmark ranges from several minutes to several hours, depending on the configurations.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, and 1 others. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Chris Alberti, Daniel Andor, Emily Pitler, Jacob Devlin, and Michael Collins. 2019. Synthetic QA corpora generation with roundtrip consistency. In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 6168–6173, Florence, Italy. Association for Computational Linguistics.

Giuseppe Amato and Umberto Straccia. 1999. User profile modeling and applications to digital libraries. In *Research and Advanced Technology for Digital Libraries, Third European Conference, ECDL'99, Paris, France, September 22-24, 1999, Proceedings*, volume 1696 of *Lecture Notes in Computer Science*, pages 184–197. Springer.

R. C. Atkinson and R. M. Shiffrin. 1968. Human memory: A proposed system and its control processes. In K. W. Spence and J. T. Spence, editors, *The Psychology of Learning and Motivation*, volume 2, pages 89–195. Academic Press, New York.

Qiguang Chen, Libo Qin, Jinhao Liu, Dengyun Peng, Jiannan Guan, Peng Wang, Mengkang Hu, Yuhang Zhou, Te Gao, and Wanxiang Che. 2025. Towards reasoning era: A survey of long chain-of-thought for reasoning large language models. *arXiv preprint arXiv:2503.09567*.

Prateek Chhikara, Dev Khant, Saket Aryan, Taranjeet Singh, and Deshraj Yadav. 2025. Mem0: Building production-ready ai agents with scalable long-term memory. *arXiv preprint arXiv:2504.19413*.

Eunbi Choi, Yongrae Jo, Joel Jang, Joonwon Jang, and Minjoon Seo. 2023. Fixed input parameterization for efficient prompting. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8428–8441, Toronto, Canada. Association for Computational Linguistics.

Gheorghe Comanici, Eric Bieber, Mike Schaekermann, Ice Pasupat, Noveen Sachdeva, Inderjit Dhillon, Marcel Blistein, Ori Ram, Dan Zhang, Evan Rosen, and 1 others. 2025. Gemini 2.5: Pushing the frontier with advanced reasoning, multimodality, long context, and next generation agentic capabilities. *arXiv preprint arXiv:2507.06261*.

Thomas M Cover. 1999. *Elements of information theory*. John Wiley & Sons.

Naihao Deng, Xinliang Zhang, Siyang Liu, Winston Wu, Lu Wang, and Rada Mihalcea. 2023. You are what you annotate: Towards better models through annotator representations. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 12475–12498, Singapore. Association for Computational Linguistics.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Matthijs Douze, Alexandr Guzhva, Chengqi Deng, Jeff Johnson, Gergely Szilvasy, Pierre-Emmanuel Mazaré, Maria Lomeli, Lucas Hosseini, and Hervé Jégou. 2024. The faiss library.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z Pan. 2025. Rethinking memory in

ai: Taxonomy, operations, topics, and future directions. *arXiv e-prints*, pages arXiv–2505.

Mehmet Samet Duran and Tevfik Aytekin. 2025. Beyond one-size-fits-all summarization: Customizing summaries for diverse users. *arXiv preprint arXiv:2503.10675*.

Aamir Fareed, Saima Hassan, Samir Brahim Belhaouari, and Zahid Halim. 2023. A collaborative filtering recommendation framework utilizing social networks. *Machine Learning with Applications*, 14:100495.

Gerhard Fischer. 2001. User modeling in human–computer interaction. *User modeling and user-adapted interaction*, 11(1):65–86.

Rui Gao, Bibo Hao, Shuotian Bai, Lin Li, Ang Li, and Tingshao Zhu. 2013. Improving user profile with personality traits predicted from social media content. In *Seventh ACM Conference on Recommender Systems, RecSys '13, Hong Kong, China, October 12-16, 2013*, pages 355–358. ACM.

Liang Gou, Michelle X. Zhou, and Huahai Yang. 2014. Knowme and shareme: understanding automatically discovered personality traits from social media and user sharing preferences. In *CHI Conference on Human Factors in Computing Systems, CHI'14, Toronto, ON, Canada - April 26 - May 01, 2014*, pages 955–964. ACM.

Xiangnan He, Lizi Liao, Hanwang Zhang, Liqiang Nie, Xia Hu, and Tat-Seng Chua. 2017. Neural collaborative filtering. In *Proceedings of the 26th international conference on world wide web*, pages 173–182.

Stephen J Hoch and George F Loewenstein. 1991. Time-inconsistent preferences and consumer self-control. *Journal of consumer research*, 17(4):492–507.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, and 1 others. 2022. Lora: Low-rank adaptation of large language models. *ICLR*, 1(2):3.

Dana Hughes, Akshat Agarwal, Yue Guo, and Katia Sycara. 2020. Inferring non-stationary human preferences for human-agent teams. In *2020 29th IEEE International Conference on Robot and Human Interactive Communication (RO-MAN)*, pages 1178–1185. IEEE.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, and 1 others. 2024. Qwen2. 5-coder technical report. *arXiv preprint arXiv:2409.12186*.

Meng Jiang, Peng Cui, Fei Wang, Wenwu Zhu, and Shiqiang Yang. 2014. Scalable recommendation with social contextual information. *IEEE Trans. Knowl. Data Eng.*, 26(11):2789–2802.

Jiazheng Kang, Mingming Ji, Zhe Zhao, and Ting Bai. 2025. Memory OS of AI agent. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 25961–25970, Suzhou, China. Association for Computational Linguistics.

Wang-Cheng Kang, Jianmo Ni, Nikhil Mehta, Maheswaran Sathiamoorthy, Lichan Hong, Ed H. Chi, and Derek Zhiyuan Cheng. 2023. Do llms understand user preferences? evaluating llms on user rating prediction. *CoRR*, abs/2305.06474.

Jieun Kim, Ahreum Lee, and Hokyoung Ryu. 2013. Personality and its effects on learning performance: Design guidelines for an adaptive e-learning system based on a user model. *International Journal of Industrial Ergonomics*, 43(5):450–461.

Sangyeop Kim, Yohan Lee, Sanghwa Kim, Hyunjong Kim, and Sungzoon Cho. 2025. Pre-storage reasoning for episodic memory: Shifting inference burden to memory for personalized dialogue. *arXiv preprint arXiv:2509.10852*.

Yehuda Koren, Robert M. Bell, and Chris Volinsky. 2009. Matrix factorization techniques for recommender systems. *Computer*, 42(8):30–37.

Jaehyeok Lee, Keisuke Sakaguchi, and JinYeong Bak. 2025. Self-training meets consistency: Improving LLMs' reasoning with consistency-driven rationale evaluation. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 10519–10539, Albuquerque, New Mexico. Association for Computational Linguistics.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, and 1 others. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems*, 33:9459–9474.

Minchong Li, Feng Zhou, and Xiaohui Song. 2025. BiLD: Bi-directional logits difference loss for large language model distillation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pages 1168–1182, Abu Dhabi, UAE. Association for Computational Linguistics.

Jiahong Liu, Zexuan Qiu, Zhongyang Li, Quanyu Dai, Wenhao Yu, Jieming Zhu, Minda Hu, Menglin Yang, Tat-Seng Chua, and Irwin King. 2025. A survey of personalized large language models: Progress and future directions. *arXiv preprint arXiv:2502.11528*.

Junling Liu, Chao Liu, Renjie Lv, Kang Zhou, and Yan Zhang. 2023. Is chatgpt a good recommender? A preliminary study. *CoRR*, abs/2304.10149.

Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy

Liang. 2024. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

Jinghao Luo, Yuchen Tian, Chuxue Cao, Ziyang Luo, Hongzhan Lin, Kaixin Li, Chuyi Kong, Ruichao Yang, and Jing Ma. 2026. From storage to experience: A survey on the evolution of llm agent memory mechanisms.

Aman Madaan, Niket Tandon, Peter Clark, and Yiming Yang. 2022. Memory-assisted prompt editing to improve GPT-3 after deployment. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2833–2861, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Lucie Charlotte Magister, Katherine Metcalf, Yizhe Zhang, and Maartje ter Hoeve. 2024. On the way to LLM personalization: Learning to remember user conversations. *CoRR*, abs/2411.13405.

Lucie Charlotte Magister, Katherine Metcalf, Yizhe Zhang, and Maartje Ter Hoeve. 2025. On the way to LLM personalization: Learning to remember user conversations. In *Proceedings of the First Workshop on Large Language Model Memorization (L2M2)*, pages 61–77, Vienna, Austria. Association for Computational Linguistics.

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of LLM agents. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13851–13870, Bangkok, Thailand. Association for Computational Linguistics.

Sheshera Mysore, Zhuoran Lu, Mengting Wan, Longqi Yang, Bahareh Sarrafzadeh, Steve Menezes, Tina Baghaee, Emmanuel Barajas Gonzalez, Jennifer Neville, and Tara Safavi. 2024. Pearl: Personalizing large language model writing assistants with generation-calibrated retrievers. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 198–219, Miami, Florida, USA. Association for Computational Linguistics.

Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. 2025. Nemori: Self-organizing agent memory inspired by cognitive science. *arXiv preprint arXiv:2508.03341*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th annual acm symposium on user interface software and technology*, pages 1–22.

Aleksandr V. Petrov and Craig Macdonald. 2023. Generative sequential recommendation with gptrec. *CoRR*, abs/2306.11114.

Minh Pham, Minsu Cho, Ameya Joshi, and Chinmay Hegde. 2022. Revisiting self-distillation. *arXiv preprint arXiv:2206.08491*.

Bhawna Piryani, Abdelrahman Abdullah, Jamshid Mozafari, Avishek Anand, and Adam Jatowt. 2025. It's high time: A survey of temporal information retrieval and question answering. *arXiv e-prints*, pages arXiv–2505.

Erasmo Purificato, Ludovico Boratto, and Ernesto William De Luca. 2024. User modeling and user profiling: A comprehensive survey. *CoRR*, abs/2402.09660.

Ruiyang Qin, Jun Xia, Zhenge Jia, Meng Jiang, Ahmed Abbasi, Peipei Zhou, Jingtong Hu, and Yiyu Shi. 2024. Enabling on-device large language model personalization with self-supervised data selection and synthesis. In *Proceedings of the 61st ACM/IEEE design automation conference*, pages 1–6.

Yilun Qiu, Xiaoyan Zhao, Yang Zhang, Yimeng Bai, Wenjie Wang, Hong Cheng, Fuli Feng, and Tat-Seng Chua. 2025. Measuring what makes you unique: Difference-aware user modeling for enhancing LLM personalization. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 21258–21277, Vienna, Austria. Association for Computational Linguistics.

Zhaopeng Qiu, Xian Wu, Jingyue Gao, and Wei Fan. 2021. U-BERT: pre-training user representations for improved recommendation. In *Thirty-Fifth AAAI Conference on Artificial Intelligence, AAAI 2021, Thirty-Third Conference on Innovative Applications of Artificial Intelligence, IAAI 2021, The Eleventh Symposium on Educational Advances in Artificial Intelligence, EAAI 2021, Virtual Event, February 2-9, 2021*, pages 4320–4327. AAAI Press.

Preston Rasmussen, Pavlo Paliychuk, Travis Beauvais, Jack Ryan, and Daniel Chalef. 2025. Zep: a temporal knowledge graph architecture for agent memory. *arXiv preprint arXiv:2501.13956*.

Christopher Richardson, Yao Zhang, Kellen Gillespie, Sudipta Kar, Arshdeep Singh, Zeynab Raeesy, Omar Zia Khan, and Abhinav Sethy. 2023. Integrating summarization and retrieval for enhanced personalization via large language models. *CoRR*, abs/2310.20081.

Evan F Risko and Sam J Gilbert. 2016. Cognitive offloading. *Trends in cognitive sciences*, 20(9):676–688.

Stephen E. Robertson and Hugo Zaragoza. 2009. The probabilistic relevance framework: BM25 and beyond. *Found. Trends Inf. Retr.*, 3(4):333–389.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. MemInsight: Autonomous memory augmentation for LLM agents. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language*

Processing, pages 33136–33152, Suzhou, China. Association for Computational Linguistics.

Alireza Salemi, Sheshera Mysore, Michael Bendersky, and Hamed Zamani. 2024. LaMP: When large language models meet personalization. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7370–7392, Bangkok, Thailand. Association for Computational Linguistics.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Aaditya Singh, Adam Fry, Adam Perelman, Adam Tart, Adi Ganesh, Ahmed El-Kishky, Aidan McLaughlin, Aiden Low, AJ Ostrow, Akhila Ananthram, and 1 others. 2025. Openai gpt-5 system card. *arXiv preprint arXiv:2601.03267*.

Charlie Snell, Dan Klein, and Ruiqi Zhong. 2022. Learning by distilling context. *arXiv preprint arXiv:2209.15189*.

John Sweller. 1988. Cognitive load during problem solving: Effects on learning. *Cognitive science*, 12(2):257–285.

Haoran Tan, Zeyu Zhang, Chen Ma, Xu Chen, Quanyu Dai, and Zhenhua Dong. 2025. MemBench: Towards more comprehensive evaluation on the memory of LLM-based agents. In *Findings of the Association for Computational Linguistics: ACL 2025*, pages 19336–19352, Vienna, Austria. Association for Computational Linguistics.

Qingyu Tan, Hwee Tou Ng, and Lidong Bing. 2023. Towards benchmarking and improving the temporal reasoning capability of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 14820–14835, Toronto, Canada. Association for Computational Linguistics.

Zhaoxuan Tan, Zheyuan Liu, and Meng Jiang. 2024a. Personalized pieces: Efficient personalized large language models through collaborative efforts. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6459–6475, Miami, Florida, USA. Association for Computational Linguistics.

Zhaoxuan Tan, Qingkai Zeng, Yijun Tian, Zheyuan Liu, Bing Yin, and Meng Jiang. 2024b. Democratizing large language models via personalized parameter-efficient fine-tuning. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 6476–6491, Miami, Florida, USA. Association for Computational Linguistics.

Xiangru Tang, Anni Zou, Zhuosheng Zhang, Ziming Li, Yilun Zhao, Xingyao Zhang, Arman Cohan, and Mark Gerstein. 2024. MedAgents: Large language models as collaborators for zero-shot medical reasoning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 599–621, Bangkok, Thailand. Association for Computational Linguistics.

Yu-Min Tseng, Yu-Chao Huang, Teng-Yun Hsiao, Wei-Lin Chen, Chao-Wei Huang, Yu Meng, and Yun-Nung Chen. 2024. Two tales of persona in LLMs: A survey of role-playing and personalization. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16612–16631, Miami, Florida, USA. Association for Computational Linguistics.

Endel Tulving. 1985. How many memory systems are there? *American psychologist*, 40(4):385.

Endel Tulving and 1 others. 1972. Episodic and semantic memory. *Organization of memory*, 1(381-403):1.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 5998–6008.

Haoming Wang, Boyuan Yang, Xiangyu Yin, and Wei Gao. 2025a. Never start from scratch: Expediting on-device llm personalization via explainable model selection. In *Proceedings of the 23rd Annual International Conference on Mobile Systems, Applications and Services*, pages 154–168.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and 1 others. 2024. A survey on large language model based autonomous agents. *Frontiers of Computer Science*, 18(6):186345.

Yu Wang, Ryuichi Takanobu, Zhiqi Liang, Yuzhen Mao, Yuanzhe Hu, Julian McAuley, and Xiaojian Wu. 2025b. Mem-{\alpha}: Learning memory construction via reinforcement learning. *arXiv preprint arXiv:2509.25911*.

Tianxin Wei, Noveen Sachdeva, Benjamin Coleman, Zhankui He, Yuanchen Bei, Xuying Ning, Mengting Ai, Yunzhe Li, Jingrui He, Ed H Chi, and 1 others. 2025. Evo-memory: Benchmarking llm agent test-time learning with self-evolving memory. *arXiv preprint arXiv:2511.20857*.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2025a. Longmemeval: Benchmarking chat assistants on long-term interactive memory. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Yaxiong Wu, Sheng Liang, Chen Zhang, Yichao Wang, Yongyue Zhang, Huifeng Guo, Ruiming Tang, and Yong Liu. 2025b. From human memory to ai memory: A survey on memory mechanisms in the era of llms. *arXiv preprint arXiv:2504.15965*.

Wujiang Xu, Zujie Liang, Kai Mei, Hang Gao, Juntao Tan, and Yongfeng Zhang. 2025. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110*.

Sikuan Yan, Xiufeng Yang, Zuchao Huang, Ercong Nie, Zifeng Ding, Zonggen Li, Xiaowen Ma, Jinhe Bi, Kristian Kersting, Jeff Z. Pan, Hinrich Schuetze, Volker Tresp, and Yunpu Ma. 2025. Memory-R1: Enhancing large language model agents to manage and utilize memories via reinforcement learning. *arXiv preprint arXiv:2508.19828*.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, and 1 others. 2025. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*.

Kai Zhang, Yangyang Kang, Fubang Zhao, and Xiaozhong Liu. 2024a. LLM-based medical assistant personalization with short- and long-term memory coordination. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 2386–2398, Mexico City, Mexico. Association for Computational Linguistics.

Kai Zhang, Lizhi Qing, Yangyang Kang, and Xiaozhong Liu. 2024b. Personalized LLM response generation with parameterized memory injection. *CoRR*, abs/2404.03565.

Linfeng Zhang, Jiebo Song, Anni Gao, Jingwei Chen, Chenglong Bao, and Kaisheng Ma. 2019. Be your own teacher: Improve the performance of convolutional neural networks via self distillation. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 3713–3722.

Xinliang Frederick Zhang, Nicholas Beauchamp, and Lu Wang. 2025. PRIME: Large language model personalization with cognitive dual-memory and personalized thought process. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 33707–33736, Suzhou, China. Association for Computational Linguistics.

Xinliang Frederick Zhang, Nick Beauchamp, and Lu Wang. 2024c. Narrative-of-thought: Improving temporal reasoning of large language models via recounted narratives. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 16507–16530, Miami, Florida, USA. Association for Computational Linguistics.

Xinliang Frederick Zhang, Carter Blum, Temma Choji, Shalin Shah, and Alakananda Vempala. 2024d. ULTRA: Unleash LLMs' potential for event argument extraction through hierarchical modeling and pairwise self-refinement. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 8172–8185, Bangkok, Thailand. Association for Computational Linguistics.

Zeyu Zhang, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Quanyu Dai, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2024e. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501*.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024. Memorybank: Enhancing large language models with long-term memory. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19724–19731.

A Non-appropriate QA pairs

Taking LoCoMo for instance, during our data filtering process within the self-learning pipeline, we consider the following three categories as non-appropriate QA pairs.

- Uninformative temporal questions (e.g., asking for the timestamp of a chat message rather than when an actual activity occurred)
- Unanswerable questions (e.g., the synthetic reference answer contains phrases like "i do not know", "not mentioned", "unspecified", etc.)
- Overly long answers (i.e., here, we heuristically drop QA pairs whose answer contains more than 30 tokens.)

B Training Details of Internalized Memory Reading

When performing parameter updating (Section 4.2) using LoRA technique (Hu et al., 2022), we manually set 5 epochs with warm-up ratio of 0.05, batch size at 4, gradient accumulation steps at 2, learning rate at 5×10^{-4} and rank $r = 16$, and we systematically study the impact of d (default: 10) for approximating KL divergence in Section 6.3, while leaving other hyper-parameters as default.

it is also worth mentioning that, as shown in Equation (1), the synthetic answer y is used primarily for *data construction and filtering*. The distillation objective relies on the teacher-generated distribution $\hat{y}$, following the standard formulation of context distillation.

C Comparison between TSUBASA and TSUBASA-PRO

Exactly the same framework and underlying models are used for both versions. The training details

Model	Method	P	R	F1	BLEU-1	Rouge-L	Input Len. (tokens)
QWEN2.5-7B	RAG (k=1)	19.90	35.90	21.97	17.87	21.46	29.4
	RAG (k=3)	24.47	48.41	27.75	22.55	27.06	75.4
	Train	4.94	14.82	6.48	4.55	6.14	5.6
	Session	36.81	66.62	41.49	34.13	40.42	2,158.5
QWEN3-8B	RAG (k=1)	18.30	35.91	21.26	17.06	20.82	29.4
	RAG (k=3)	27.74	48.81	28.10	22.53	27.30	75.4
	Train	4.57	13.22	6.07	4.32	5.73	5.6
	Session	34.66	69.27	41.22	33.64	39.61	2,158.5
QWEN2.5-14B	RAG (k=1)	21.61	34.06	22.99	18.97	22.58	29.4
	RAG (k=3)	29.71	48.44	32.51	27.47	31.71	75.4
	Train	2.67	10.46	3.70	2.56	3.50	5.6
	Session	41.65	69.54	46.80	39.66	45.61	2,158.5
QWEN3-14B	RAG (k=1)	21.97	36.18	24.29	20.00	23.78	29.4
	RAG (k=3)	27.69	49.05	31.38	26.02	30.59	75.4
	Train	5.29	12.57	6.59	4.96	6.24	5.6
	Session	38.61	70.10	44.26	36.40	42.87	2,158.5

Table A1: Complete preliminary study reveals *train-inference gap* and *quality-efficiency tradeoff* as discussed in Section 3. QWEN2.5-14B *(Train)* is showing extremely inferior results since it always generates "there is no specific mention of XYZ".

Memory Operation	**Count**	**Proportion**
ADD	2,594	78.8%
UPDATE	570	17.3%
RECONCILE	108	3.3%
IGNORE	21	0.6%
Total	**3,293**	**100.0%**

Table A2: Distribution of dynamic memory writing operations (ADD, UPDATE, RECONCILE, and IGNORE). Detailed explanations of each action are provided in Section 4.1. With our dynamic memory writing approach, we effectively save 22% on storing the embedding-based index. Furthermore, our analysis shows that performance remains largely unaffected even if the "IGNORE" action is disabled (i.e., enforcing all memory pieces to be stored in one way or another).

are identical for both, as described in Section 5, except for d in KL divergence estimation. $d = 10$ and $d = 20$ are used by TSUBASA and TSUBASA-PRO, respectively. The major differences reside in inference-time: 1) Compared with the system prompt designed for TSUBASA, TSUBASA-PRO additionally emphasizes on *concise generation*, as inspired by Yan et al. (2025); 2) the number of retrieved items (k) are different: TSUBASA always sets $k = 3$ by default, while TSUBASA-PRO uses the default value when tested on LongMemEval but increases to $k = 10$ for LoCoMo due to slighlty improved performance, in line with Figure 2.

Model	Method	P	R	F1	BLEU-1	ROUGE-L	Input Len.
Qwen3-4B	RAG (utterance)	24.58	49.87	28.53	23.20	27.66	75.4
	RAG (observation)	27.25	**51.20**	31.17	25.50	30.31	57.1
	Vanilla Training	4.34	13.44	5.76	4.02	5.53	5.6
	TSUBASA (no grounding)	10.43	21.54	12.26	9.50	11.84	5.6
	TSUBASA (static accumulation)	**29.14**	50.42	**32.82**	**27.21**	**31.96**	57.1
	TSUBASA (evolving memory)	27.95	47.72	31.38	26.23	30.40	51.7
Qwen3-8B	RAG (utterance)	23.74	48.81	28.10	22.53	27.30	75.4
	RAG (observation)	27.17	**53.52**	31.77	25.91	30.63	68.1
	Vanilla Training	4.57	13.22	6.07	4.32	5.73	5.6
	TSUBASA (no grounding)	10.31	20.37	11.99	9.33	11.67	5.6
	TSUBASA (static accumulation)	**30.99**	52.01	**34.50**	**29.25**	**33.40**	68.1
	TSUBASA (evolving memory)	30.95	50.03	33.99	28.92	32.79	63.2
Qwen3-14B	RAG (utterance)	27.69	49.05	31.38	26.02	30.59	75.4
	RAG (observation)	30.39	54.66	34.83	28.79	33.43	62.4
	Vanilla Training	5.29	12.57	6.59	4.96	6.24	5.6
	TSUBASA (no grounding)	13.19	22.96	14.71	11.90	14.41	5.6
	TSUBASA (static accumulation)	33.23	**54.98**	37.07	31.25	35.65	62.4
	TSUBASA (evolving memory)	**34.63**	53.82	**37.63**	**32.16**	**36.16**	61.1
Qwen3-32B	RAG (utterance)	24.31	50.35	28.64	23.19	27.83	75.4
	RAG (observation)	29.20	56.70	34.00	28.00	32.56	60.1
	Vanilla Training	6.86	15.63	8.28	6.22	7.91	5.6
	TSUBASA (no grounding)	12.15	26.35	14.81	11.27	14.37	5.6
	TSUBASA (static accumulation)	27.20	**57.30**	32.75	26.18	31.46	60.1
	TSUBASA (evolving memory)	**29.58**	56.09	**34.25**	**28.12**	**32.85**	59.0

Table A3: Complete results of TSUBASA and baselines on LoCoMo dataset (average of three runs), allowing for a holistic understanding of each module's contribution. Here, we use *Input Length* as proxies of inference-time efficiency, since the generation length to factoid questions in LoCoMo are generally short and largely unchanged (i mean, the average length) acorss methods. Therefore, the input length directly correlates with latency. Here, we do not consider offline operations (e.g., memory-writing operations, synthetic QA creation and filtering) as they do not impact latency. For each base model, the best results are highlighted in **bold**, while global optimal performances are underlined. The results demonstrate superior information density of observations, usefulness of internalized memory reading, necessity for factual grounding, and efficacy of evolving memory.

Method	Single Hop			Multi-Hop			Open Domain			Temporal			Overall		
	F1	BLEU-1	J	F1	BLEU-1	J	F1	BLEU-1	J	F1	BLEU-1	J	F1	BLEU-1	J
LoCoMo (Maharana et al.)	9.57	7.00	15.06	11.84	10.02	19.28	8.67	6.52	12.79	8.35	8.74	5.43	8.97	7.27	12.17
A-Mem (Xu et al.)	18.96	12.86	40.78	14.73	12.66	31.32	30.58	26.14	46.90	23.67	20.67	28.68	26.08	21.78	40.78
Mem0 (Chhikara et al.)	24.96	18.05	61.92	20.31	15.82	48.19	32.74	25.27	65.20	33.16	26.28	38.76	30.61	23.55	53.30
MemoryOS (Kang et al.)	29.55	22.59	48.12	21.03	18.41	38.55	40.85	36.26	63.14	26.26	19.70	24.81	34.64	29.36	51.26
Memory-R1-PPO (Yan et al.)	34.22	23.61	57.74	32.87	29.48	53.01	44.78	38.72	66.99	42.88	30.30	42.25	41.72	33.70	59.53
Memory-R1-GRPO (Yan et al.)	33.64	26.06	62.34	23.55	20.71	40.96	**46.86**	**40.92**	**67.81**	**47.75**	38.49	**49.61**	43.14	36.44	61.51
TSUBASA-PRO (static accmulation, QWEN2.5-7B)	39.76±0.60	32.57±0.83	**85.50±0.68**	31.56±0.44	25.58±0.41	78.71±1.28	19.73±2.06	16.59±1.91	60.20±1.43	31.07±0.59	25.89±0.74	40.62±1.30	35.37±0.39	29.03±0.49	**73.90±0.54**
TSUBASA-PRO (evolving memory, QWEN2.5-7B)	37.37±0.13	30.50±0.14	82.64±0.49	32.26±0.78	26.19±0.63	80.93±0.92	19.28±1.12	16.16±0.96	60.09±1.64	37.06±0.49	32.21±0.36	43.36±0.55	35.23±0.23	29.14±0.19	73.08±0.30
TSUBASA-PRO (static accmulation, QWEN3-8B)	**50.87±0.38**	**45.61±0.40**	84.92±0.32	37.39±0.36	30.94±0.31	80.61±0.46	26.16±1.15	23.43±1.26	61.51±2.85	42.14±0.23	37.49±0.30	33.69±1.01	**45.24±0.13**	**40.02±0.16**	72.59±1.92
TSUBASA-PRO (evolving memory, QWEN3-8B)	48.26±0.64	43.21±0.68	81.68±0.11	**38.39±0.46**	**32.71±0.28**	**82.20±0.35**	27.81±0.66	24.85±0.62	62.72±1.24	45.82±0.66	**42.07±0.60**	42.19±0.32	44.65±0.22	39.87±0.26	72.46±0.04

Table A4: Comparison of TSUBASA-PRO (average of 3 runs) with prior SOTA baselines on LoCoMo benchmark. Results of non-TSUBASA baselines are taken from Yan et al. (2025). Best numbers are in **bold**.

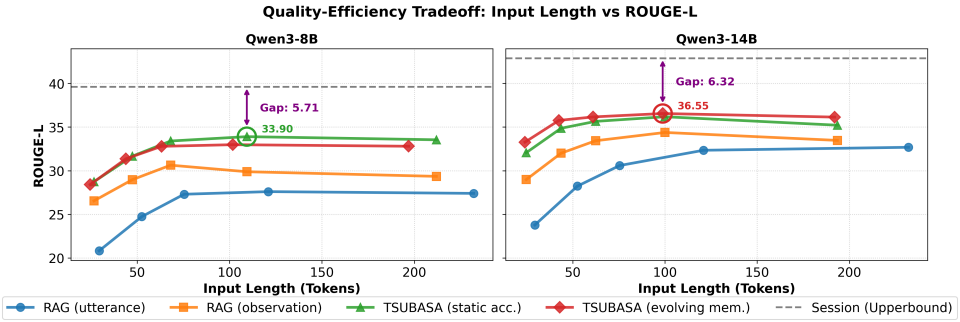

Figure A1: Quality-Efficiency tradeoff between input length and ROUGE-L metric on baselines and TSUBASA. Best-performing configuration is highlighted in circle with its gap to the established ceiling (Session) indicated. Importantly, our TSUBASA achieves *Pareto improvement*: attaining higher peak performance while utilizing only a fraction of the context budget.

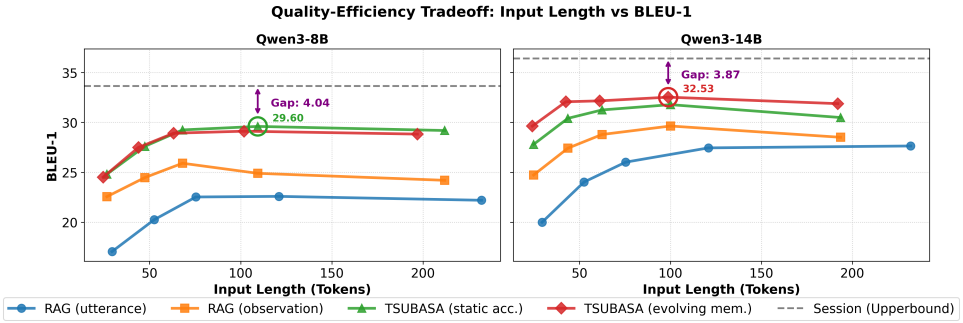

Figure A2: Quality-Efficiency tradeoff between input length and BLEU-1 metric on baselines and TSUBASA. Best-performing configuration is highlighted in circle with its gap to the established ceiling (Session) indicated. Importantly, our TSUBASA achieves *Pareto improvement*: attaining higher peak performance while utilizing only a fraction of the context budget.

Method	F1	BLEU-1	J
LoCoMo (Maharana et al.)	18.27	14.57	22.20
A-Mem (Xu et al.)	41.55	36.58	54.80
Mem0 (Chhikara et al.)	38.44	34.53	46.80
Memory-R1-PPO (Yan et al.)	40.30	35.50	47.40
Memory-R1-GRPO (Yan et al.)	**46.70**	41.10	**57.80**
TSUBASA-PRO (static acc.)	45.75	**42.10**	57.40
TSUBASA-PRO (evolving mem.)	43.43	39.85	53.00

Table A5: Comparison of TSUBASA-PRO (single run) with prior SOTA baselines on LongMemEval (Wu et al., 2025a). Results of non-TSUBASA baselines are taken from Yan et al. (2025). Best numbers are in **bold**. Overall, TSUBASA-PRO performs on par with Memory-R1.

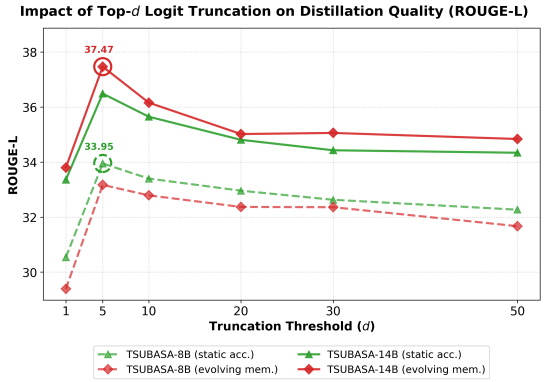

Figure A3: Impact of top-d truncation for approximating KL divergence on distillation quality (ROUGE-L) with TSUBASA variants. Best-performing configuration is highlighted in circle for each model size. Importantly, we find $d = 1$ inadequate, while increasing d beyond 5 leads to performance degradation due to noises in long-tail distribution, establishing $d = 5$ at the sweet spot.

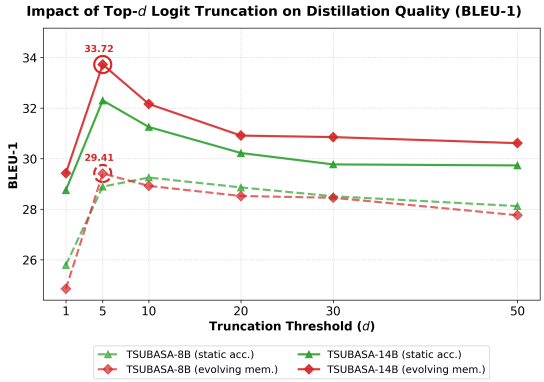

Figure A4: Impact of top-d truncation for approximating KL divergence on distillation quality (BLEU-1) with TSUBASA variants. Best-performing configuration is highlighted in circle for each model size. Importantly, we find $d = 1$ inadequate, while increasing d beyond 5 leads to performance degradation due to noises in long-tail distribution, establishing $d = 5$ at the sweet spot.

Observation Extraction Template

Analyze the conversation between {speaker_a} and {speaker_b}, and extract the major OBSERVATIONS for {speaker_target}.

Core Requirement: The OBSERVATIONS are defined as verifiable facts about {speaker_target}'s life, background, and habits. Avoid abstract "meta-observations" about the communication dynamics such as '{speaker_target} is supportive', '{speaker_target} appreciates' etc; Instead, focus on concrete facts such as past events, recurring habits, and stated preferences of {speaker_target}.

Organic 5W1H Integration: Each OBSERVATION should be a standalone sentence that provides as much salient context as possible while adhering to the 5W1H principle. While not every OBSERVATION requires all six points, strive to include:

- **Who/What:** The specific entity, role, or action.
- **Where:** The location or setting associated with the fact.
- **When:** The timeframe, frequency, or specific date.
- **Why/How:** The stated reason or method behind an action or preference.

Temporal Awareness: The timestamp provided ({time}) reflects only when the conversation took place. Do not assume events discussed happened on this date unless explicitly stated (e.g., "I did this yesterday"). Distinguish between the date of the report and the date of the event.

———————-
{conversation}
———————-

Return a JSON object of the form {{ "OBSERVATIONS": ["<string>", "<string>", ...] }}. The value should be a list of strings, each string being a major OBSERVATION about {speaker_target}. Do not include any explanations, commentary, or text outside the JSON object. Important: Escape all double-quote characters within string output with backslash. End your response with "#END".

Figure A5: Prompt used for extracting salient observations from a conversation session.

Memory Evolution Prompt

You are a Memory Manager. Your task is to evolve a person's "Memory Store" based on a list of "New Observations" extracted from a recent conversation session. Your goal is to maintain a "Memory Store" that is optimized for Vector Database Retrieval.

RETRIEVAL PRINCIPLE:
A retriever must be able to understand a "refined_observation" without seeing any other context.

ACTIONS:

1. **"ADD"**: Brand new information.
2. **"UPDATE"**: New info adds specific detail or a more recent status to an existing index.
3. **"RECONCILE"**: New info contradicts or changes an existing state. Combine them into a "Temporal Narrative" (e.g., "Subject previously X, but as of [Date] now Y").
4. **"IGNORE"**: The info is already perfectly captured or is trivial chatter.

CURRENT MEMORY STORE for {speaker}:
{current_memory}

NEW OBSERVATIONS LIST for {speaker} [session date : {timestamp} (context: today)]:
{new_obs_list}

RULES FOR "refined_observation":

- **Sequential Processing:** You must process every observation in the "NEW OBSERVATIONS LIST" in order.
- **Temporal Grounding:** You MUST include both the absolute date and the relative timeframe (e.g., "last week", "yesterday"), if either is mentioned in the observation. Perform calculations relative to the session date. This is crucial for retrieval relevance.
  - *Example*: If the session date is 25 February, 2026, and {speaker} says "I bought a car two days ago," the refined observation must include "On 23 February, 2026 (context: two days ago), {speaker} bought a car."
- **Stand-alone Density:** Each refined observation must be a complete, keyword-rich sentence, that defines the Subject, Action, and Context.
- **Index Integrity:** If "UPDATE" or "RECONCILE", the "index" must be the exact integer from the current store.

Return your response in STRICT JSON format. The output must be a list of objects:

```
[
  {
    "original_obs": "The raw observation from the NEW OBSERVATIONS LIST",
    "action": "ADD" | "UPDATE" | "RECONCILE" | "IGNORE",
    "index": index_to_change_or_null,
    "refined_observation": "The updated or new string to store. It must be self-contained
                            and standalone."
  },
  ...
]
```

#END

Figure A6: Prompt used by our memory manager to autonomously evolve the episodic memory.

Dynamic Memory Writing Process: Observation Extraction and Memory Evolution

RAW UTTERANCES (Session 6)
Timestamp: 2:33 pm on 5 February, 2023

...
John: Yeah, Maria, we learn a lot from our own struggles. I just started helping out with a food drive for folks who lost their jobs. Here's a picture of me at the food bank.
Maria: Wow, John, that's incredible! What inspired you to get involved with something like this?
John: Seeing the effect unemployment has on our neighbors made me decide to act. I wanted to help out in these tough times by doing a community food drive. We can all make a difference!
Maria: That's really great of you. What sparked your decision to start this initiative?
John: Thanks, Maria. Unemployment in our community was the reason behind it.
Maria: You did awesome! How's the response been to that?
John: Thanks, Maria! We've been overwhelmed by the response and the volunteers. Here's a photo of them at a recent event.
...

EXTRACTED OBSERVATIONS (Session 6)
Timestamp: 2:33 pm on 5 February, 2023

- ...
- John is involved in political activities as he mentioned running for office to make an impact.
- John initiated a community food drive due to the impact of unemployment on his neighbors.
- John has been overwhelmed by the positive response and volunteer participation in his food drive initiative.
- ...

EXTRACTED OBSERVATIONS (Session 28)
Timestamp: 5:19 pm on 5 August, 2023

- ...
- John lost his job at a mechanical engineering company on 5 August, 2023, due to the company tanking, which he did not foresee, and it has been a rough period for him.
- John has not been able to volunteer much lately, but he still cares about volunteering and values its impact on the community.
- ...

EVOLVING MEMORY STORE *(Action: RECONCILE)*

- ...
- [22] On 5 February, 2023, John initiated a community food drive due to the impact of unemployment on his neighbors. On 5 August, 2023, John has not been able to volunteer much lately, but he still cares about volunteering and values its impact on the community.
- ...

Figure A7: A sample example illustrating the TSUBASA memory writing process. Raw utterances are first compressed into dense observations, which are subsequently merged by the memory manager (RECONCILE action in this particular example) to form a coherent, temporally grounded narrative. **The colored text indicates the flow.** For example, there are two observations that can be extracted from the truncated raw conversations, one in red and another in green. The sampled entry (index: 22) from our evolving memory is derived from observations in session 6 and session 28. Note, this evolving memory piece is temporally grounded with absolute time markers.