MemAura: Persistent Personalized Context Memory for LLM Services in Smart Environments

Siyuan Liu, Huangxun Chen
Hong Kong University of Science and Technology (Guangzhou), Guangdong, China
sliu268@connect.hkust-gz.edu.cn,huangxunchen@hkust-gz.edu.cn

Abstract

In this poster, we present our efforts to enable personalized LLM services in smart environments through persistent context memory. By systematically organizing long-term sensor logs into a structured format that captures user's behavioral patterns/preference in the environment, our solution MemAura empowers LLMs to better understand user intents and deliver tailored services. With its effective memory management, MemAura holds promise to make smart living spaces more efficient, context-aware and user-centric.

CCS Concepts

• **Human-centered computing → Ubiquitous and mobile computing systems and tools**.

Keywords

Large Language Model, Personalized Memory Management

1 Introduction

Recent advances in LLMs have sparked growing research interest in extending their application from chat systems to serving as a reasoning engine for smart environments [1, 3]. This enables more flexible understanding of user intent and delivery of appropriate services, marking a promising step toward realizing truly ubiquitous intelligence.

However, prior works primarily reason about user intents and plan services based on a single-moment snapshot, e.g., the current environmental layout, while overlooking the integration of users' historical behaviors and latent preference patterns. Without precise and personalized context, even advanced LLMs struggle to deliver satisfying services. For example, given the user command "I'm going to bed," a state-of-the-art method [1] might respond with "All lights and TV in the living room and entry have been turned off for your bedtime." While this response reflects common sense, it lacks adaptation to user-specific living habits, unlike the personalized response shown in Figure 1.

Recent studies [5] have explored managing personalized context memory for personal electronic devices, e.g., smartphones, tablets, enabling LLMs to retrieve relevant information for more tailored services. However, these approaches cannot be directly applied to support LLM services in smart environments due to the following challenges.

CH1: Spatio-temporal Complexity. In a smart environment, residents move across various functional areas and interact with diverse devices over time. As a result, compared to context memory for smartphone/PC agents, personalized context in smart environments inherently involves both temporal and spatial dimensions, making it significantly more complex to manage.

CH2: Multi-user Sharing Nature. Smart environments and the devices within them are often shared by multiple residents, e.g., members of a household. Unlike personal devices that are tied to a single individual, the historical profiles of different users in a shared space may interleave within logs. This makes it challenging to accurately isolate the appropriate context when LLM is expected to deliver services to a specific target user.

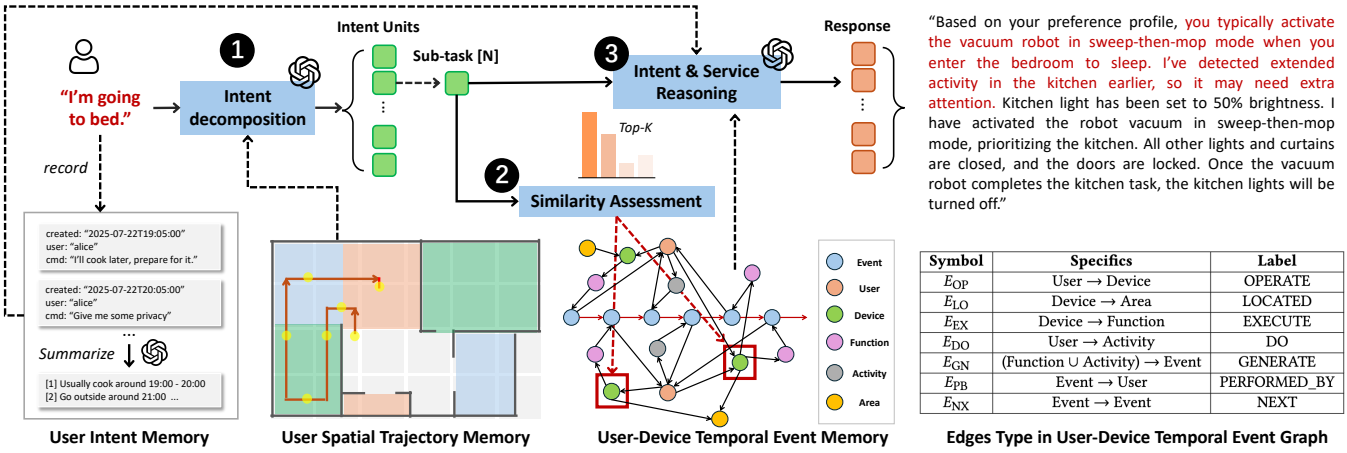
To address the above challenges, we propose MemAura, a system that enables persistent and personalized context memory for LLM services in smart environments. It systematically organizes long-term sensor logs collected from spatially distributed devices, supporting multi-user and multi-modal memory management. Technically, it features three key memory management modules: (1) user-device temporal event memory to capture user activity profiles over time, (2) user spatial trajectory memory to characterize user's movement patterns across the environment, and (3) user intent memory to model user preferences and needs. Our preliminary evaluation demonstrates that MemAura achieves 100% completion in satisfying user intents while significantly reducing total token consumption compared to baselines.

2 System Design

We propose MemAura, a persistent and personalized context memory management system designed to facilitate accurate LLM services in smart environments. Figure 1 presents our system overview, which comprises three main modules: (1) user-device temporal event memory; (2) user spatial trajectory memory; (3) user intent memory. Each module is equipped with efficient APIs that allow the LLM to query, add, and update memory as needed.

User-Device Temporal Event Memory. In a smart environment enriched with multiple devices, residents interact

Corresponding Author: Huangxun Chen.



Symbol	Specifics	Label
E_{OP}	User → Device	OPERATE
E_{LO}	Device → Area	LOCATED
E_{EX}	Device → Function	EXECUTE
E_{DO}	User → Activity	DO
E_{GN}	(Function ∪ Activity) → Event	GENERATE
E_{PB}	Event → User	PERFORMED_BY
E_{NX}	Event → Event	NEXT

Figure 1: MemAura System Overview

with them daily, revealing their living habits and personal preferences. Thus, we aim to develop an efficient structure to organize such dynamic and personalized interaction patterns over time, enabling upper-layer services to retrieve and distill highly relevant personalized context for improved service delivery.

To this end, we propose a directed and labeled heterogeneous graph G, which can be constructed from logs collected by smart devices and sensors. It includes six vertex classes $V = \bigcup_X V_X$, where $X \in \{U, D, F, AC, E, AR\}$ denote users (e.g., Alice), devices (e.g., coffee machine), functions (e.g., turn_on), activities (e.g., cooking), events (e.g., Alice cooks, Alice turns on the coffee machine), and areas (e.g., kitchen), respectively. Thus, we define node attribute function $\ell_V(V_X) = X$. For events, the attribute functions $\tau : V_E \to \mathbb{T}$ and $\sigma : V_E \to \mathbb{N}$ map each event to its timestamp in $\mathbb{T}$ and its sequence number in $\mathbb{N}$, respectively. The graph edges E have six classes and associated labels ℓ_E as shown in Figure 1. The resulting graph $G = (V, \ell_V, \tau, \sigma, E, \ell_E)$ integrates both relatively static entities, such as devices, devices' functions, residents, and routine activities, and dynamic event node expansion along with their interplay. To better support LLMs in efficiently retrieving personalized context, we further incorporate the following design:

(1) *Fine-grained Temporal Index*: Over the graph G, we incorporate a two-level index $I : \mathcal{D} \times \mathcal{H} \to V_E \cup \perp$ to support efficient daily/hourly event retrieval. Specifically, $I(d, h) = \arg\min_{v_e}\{\tau(e) \mid \tau(e) \in [T_{d,h}, T_{d,h+1})\}$, if no such event exists, $I(d, h) = \perp$, which maps a given date d and hour h to the first event in this duration, where $T_{d,h}$ denotes the starting timestamp of hour h on date d.

(2) *DevOp-Activity Transition Preference Graph*: Each user's event profile can be viewed as a sequence of device operations and activity executions. To make such patterns more accessible for LLM retrieval, we extract a user-specific event graph G_T^u from the global graph G. This graph primarily includes device vertices V_D and activity vertices V_{AC}. We examine each edge $\{e_m, e_n\} \in E_{NX}$ and identify the associated device or activity nodes connected to the endpoints, i.e., $\pi(e_m) = v_i, ; \pi(e_n) = v_j$, where $v_i, v_j \in V_D \cup V_{AC}$. We then establish a directed edge $(v_i, v_j) \in E_T^u$ and increment the associated edge attribute w by 1 to reflect the transition frequency. The resultant graph $G_T^u = (V_D \cup V_C, E_T^u, w_T^u)$ makes it easier for the LLM to access each resident's event patterns, e.g., how frequently a user performs a specific activity after interacting with a device.

(3) *Device Function Usage Graph*: Different users may exhibit distinct preferences over various device functions, even for the same device. To make such patterns more pronounced for LLM reasoning, we extract a user-specific device usage weighted graph G_F^u from the global graph G. This graph primarily consists of device vertices V_D and function vertices V_F. We consolidate all edges $\{v_d, v_f\} \in E_{EX}$ with the same endpoints and assign the edge attribute w as the total number of such edges, reflecting how many times user u has invoked function f on device d.

User Spatial Trajectory Memory. Besides temporal events, devices in a smart environment are distributed across various functional areas. The spatial movement of residents reflects their preferences and intentions, providing valuable context for LLM reasoning. To capture such spatial information, one approach is to encode the room layout in a structured format [1], accompanied by sensors/actuators logs. However, this fragmented representation embeds user trajectories implicitly, making it difficult for LLMs to interpret. Inspired by [4], we explicitly represent user movement by overlaying their trajectories onto the room layout in a grid-based manner, as shown in Figure 1. LLMs are allowed to retrieve trajectory maps for specified time intervals, enabling them to incorporate precise spatial information to better assist in user intent analysis.

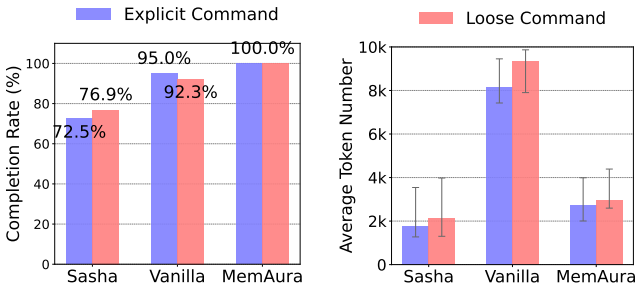
User Intent Memory. We record historical commands explicitly issued by users to model their daily routines and common needs. Each history entry comprises the user command text along with metadata (e.g., timestamp, user ID). We maintain a sliding window of length k, storing at most k recent entries for each user. To reduce substantial LLM token overhead due to potential verbosity, we employ a background periodicity service that summarizes and reflects on the current window, distilling it into a concise representation that effectively captures the user's preferences. An example is shown in Figure 1. This summary serves as a precise reference for LLM services.

Putting It All Together: Spatio-temporal Personalized Context-aware Reasoning. MemAura enables context-aware reasoning in smart environments by retrieving and integrating temporal, spatial, and personalized context. As illustrated in Figure 1, a resident may issue a command like *"I am going to bed."*, which expresses an implicit intent without mentioning specific smart devices. Our system leverages its persistent context memory to decipher the underlying need, based on the user's current situation, historical behavior and preferences, and generates a service plan specifying which devices should be invoked and how to fulfill the request. Specifically, MemAura ❶ retrieves the user's recent spatial trajectory to help decipher the intent behind the command, and potentially decomposes it into more concrete sub-tasks. For each sub-task, MemAura ❷ retrieves the associated device functions and usage patterns, selecting devices that are semantically aligned with the user's request. ❸ LLM then performs in-depth reasoning based on the identified sub-tasks and selected devices, while also retrieving user preferences as needed (e.g., use vacuum robot during sleeping). Finally, the outputs of all sub-tasks are comprehensively considered and synthesized into a coherent response.

3 Preliminary Evaluation

Setup. We adopt the command dataset from prior work [1] and utilize the corresponding environment layout (environment *h3* in [1]) to ensure fair comparison. The dataset includes 40 explicit commands with well-defined intents and 13 loosely defined commands exhibiting ambiguous intents. To populate the context memory in MemAura, we construct multi-user event logs based on real-world dataset [2], yielding 360 distinct 24-hour records. We adopt GPT-4o as the LLM backbone and compare our system with two baselines: (1) **Sasha** [1], (2) **vanilla**: Sasha [1] augmented with raw event logs. We use two evaluation metrics: (1) **completion rate**: the probability that the system fully satisfies the user's command; (2) **token consumption**.

Results. As shown in Figure 2, without user-specific context memory, Sasha achieves only a 72.5% completion rate



(a) Completion Rate **(b) Token Consumption**

Figure 2: Performance Against Baselines.

on explicit commands and 76.9% on loose commands, often due to omitted requirements or incorrect device function usage. Incorporating raw event logs significantly improves completion rates by approximately 20%, but also leads to a substantial increase in token consumption, from around $2k$ to $8k$–$9.5k$ tokens. In contrast, our design, MemAura, not only achieves a 100% completion rate on both command sets but also maintains reasonable token usage at around $3k$ tokens. This is because MemAura efficiently organizes context memory, enabling the LLM to accurately retrieve precise and relevant information for delivering correct services.

4 Conclusion and Future Work

In this work, we presented MemAura, a personalized context memory system to enhance LLM services in smart environments. For future work, we plan to explore robust memory update and reflection mechanisms to support long-term deployment, evaluate LLM of varying sizes on larger-scale command sets, and investigate potential privacy-preserving schemes, pushing MemAura toward real-world deployment.

References

[1] Evan King, Haoxiang Yu, Sangsu Lee, and Christine Julien. 2024. Sasha: Creative Goal-oriented Reasoning in Smart Homes with Large Language Models. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies* 8, 1 (2024), 1–38.

[2] Maithili Patel and Sonia Chernova. 2023. Proactive Robot Assistance via Spatio-Temporal Object Modeling. In *Conference on Robot Learning*. PMLR, 881–891.

[3] Dmitriy Rivkin, Francois Hogan, Amal Feriani, Abhisek Konar, Adam Sigal, Xue Liu, and Gregory Dudek. 2024. AIoT Smart Home via Autonomous LLM Agents. *IEEE Internet of Things Journal* (2024).

[4] Jihan Yang, Shusheng Yang, Anjali W Gupta, Rilyn Han, Li Fei-Fei, and Saining Xie. 2025. Thinking in Space: How Multimodal Large Language Models See, Remember, and Recall spaces. In *Proceedings of the Computer Vision and Pattern Recognition Conference*. 10632–10643.

[5] Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025. A Survey on the Memory Mechanism of Large Language Model based Agents. *ACM Trans. Inf. Syst.* (July 2025). doi:10.1145/3748302