Attribute-Constrained Information Construction for Long-Term In-Car Preference Memory

Mingyang Guo[a]; **Zhipeng Hu**[b]**; Jun Ma**[a*]; **Yuxuan Wu**[a]; **Zhengjun Wang**[a]; **Guoxia Zhang**[c]; **Haizhou Li**[c]; **Chengjun Li**[d]**; Qiang Tang**[d]**; Liangmin Wu**[d]

[a] College of Automotive and Energy Engineering, Tongji University, Ningyuan Hall, No. 4800 Cao'an Highway, Jiading District, Shanghai 201804, China

[b] Shanghai CyberSpark Artificial Intelligence Technology Co., Ltd., Shanghai 200232, China

[c] Bosch Automotive Products Co., Ltd., 333 Fuquan North Road, Changning District, Shanghai 200335, China.

[d] ZEEKR Blue New Energy Technology CO., Ltd., Shanghai 200232, China

Abstract: Long-term personalization is essential for in-car conversational agents, where user preferences about navigation, parking, charging, media, and cabin settings must be remembered and reused across future interactions. Existing memory systems often compress conversations into flat facts or isolated memory entries, which can break the binding among preference intent, attribute values, entities, conditions, and supporting evidence. This paper proposes an attribute-constrained information construction method for long-term preference memory. The method identifies supporting dialogue evidence and constructs reusable preference relations with verified auxiliary candidates. It then organizes this information into evidence, semantic, and attribute-relation records linked to a single preference node. Retrieval uses record-level matches to identify candidates but scores, ranks, and returns complete nodes. Experiments on CarMem and manually annotated real-world in-vehicle data show that the proposed method has improved the attribute-level preference recall by 9.85% compared to the existing public memory systems. The results indicate that reliable preference memory depends on establishing attribute ownership during information construction and preserving it through storage and retrieval.

Key Words: Autonomous Vehicles; long-term preference memory; attribute-constrained information construction; function-specific record organization; node-level retrieval.

1. INTRODUCTION

With the development of intelligent cockpits and in-vehicle large language models, in-vehicle dialogue assistants are gradually expanding from single-turn question answering and immediate function control into personalized agents capable of continuously adapting to users across sessions (Murali et al., 2021; X. Huang et al., 2025). Navigation route selection, parking lot recommendations, charging facility filtering, media playback, air-conditioning settings, and cockpit function control all depend on stable preferences formed by users through long-term interactions. Effective long-term preference memory can reduce repeated inquiries, maintain continuity across sessions, and provide a personalized basis for subsequent recommendations and vehicle function execution (Joko et al., 2024; X. Huang et al., 2025). Therefore, constructing maintainable, retrievable, and traceable user preferences from natural-language interactions has become an important foundational capability of in-vehicle dialogue agents.

User preferences in in-vehicle dialogues exhibit pronounced attribute-based (Chen et al., 2023), conditional (Firdaus et al., 2023), and context-dependent characteristics (Hu et al., 2022). A reusable preference typically involves the task object, attribute type, attribute value, entity and applicability condition; this information may be distributed across multiple turns between the user and the assistant (Andreas et al., 2020). Assistant responses may also contain selected entities, execution results, and scenario information that do not appear explicitly in the user input (Balaraman & Magnini, 2021). Meanwhile, some short values lack clear semantics when detached from their associated attributes (Liao et al., 2021), and some preferences hold only at particular times, with particular companions, in particular driving states, or for particular task goals (Gao et al., 2024). Subsequent queries usually contain only task cues or partial attribute cues rather than repeating the complete historical expression (Weng et al., 2016). Therefore, the central problem is not simply to extract preference facts or retrieve topically relevant interactions. A memory system must construct evidence-supported preference information, organize it so that different query cues can access the appropriate content, and recover the complete attribute-value-condition relation as a logical unit.

Existing research on long-term memory for large language models has made important progress in cross-session fact retention, long-range consistency, and multi-session retrieval (Z. Zhang et al., 2025; Maharana et al., 2024; Chhikara et al., 2025). Beyond fact retention, existing methods have also begun to model and evaluate memory updating, selective forgetting, and test-time learning as core capabilities of long-term memory (Chen et al., 2026; Hou et al., 2024). In terms of system efficiency, long-term memory architectures have shifted from concatenating the full context to structured storage, staged retrieval, and low-cost invocation to reduce latency and token overhead (D. Zhang et al., 2025; Hatalis et al., 2024). Fact- and profile-based memory methods support long-term personalization and stable cross-session user modeling by extracting and maintaining salient user information (X. Huang et al., 2025; Li et al., 2024). Event-based and episodic memory methods reduce fine-grained information loss caused by summarization and fact extraction by retaining event fragments, conversational episodes, or complete episodic evidence (Wang et al., 2023; S. Wang et al., 2026); graph-structured and dynamic-note methods further support the organization of entity relations, memory connections, updating, and forgetting (Sarin et al., 2025; Chhikara et al., 2025); and lightweight memory architectures reduce runtime overhead by separating offline consolidation from online retrieval (Hatalis et al., 2024; J. Zhang et al., 2026). These methods have advanced long-term memory from a simple dialogue archive to an external memory system that can be extracted, organized, and updated.

However, existing methods generally use facts (Tian et al., 2025; Chhikara et al., 2025), events (Zhou, 2025), notes (N. Zhang et al., 2026), summaries (Q. Wang et al., 2023), or relational edges (S. Wang et al., 2026) as their basic memory units. Their central limitation for in-vehicle preference memory is that memory writing is usually treated as direct extraction without explicit attribute constraints. This can omit entities, conditions, units, raw values, and supporting evidence or detach them from the attributes and preferences they qualify (T. Chen et al., 2026). Serializing all content into one text can produce semantic interference, whereas storing local fields as independent records can weaken ownership and create competition for limited top-K positions (J. Zhang et al., 2026; Liu et al., 2026). The organization and retrieval problems therefore originate from a construction problem: the memory artifact is not required to preserve a complete attribute-centered preference relation.

Accordingly, this paper investigates one central question: can attribute-constrained information construction preserve the values, entities, and conditions needed for reusable preferences more reliably than direct extraction? To implement this construction in a retrievable memory system, the study further examines how the constructed information should be organized for different query cues and how record-level matches should be mapped back to complete preferences. Function-specific record and node-level retrieval are therefore treated as storage and access implementations of the construction method.

Cognitive memory theory provides support for both the construction constraints and their implementation. The episodic buffer and feature-binding theory emphasize that information from different sources or dimensions must be integrated into object-level representations with unified ownership (Baddeley, 2000), motivating explicit binding among attributes, values, entities, conditions,

and evidence. The distinction between episodic and semantic memory supports retaining both original interaction evidence and reusable preference relations (Tulving, 1984). The encoding specificity principle further indicates that different query cues require different access paths (Tulving & Thomson, 1973). These principles motivate attribute-constrained construction as the core method and function-specific records with node-level return as its retrieval implementation.

Based on these considerations, this paper proposes attribute-constrained information construction for long-term in-vehicle preference memory. The method first localizes the interaction evidence supporting a preference and constructs each reusable relation under explicit bindings among the task object, attribute, values, entities, and conditions, and evidence. It also generates auxiliary relations and retains only candidates that satisfy the same constraints. Figure.1 illustrates the difference between direct structured extraction and attribute-constrained preference construction. To store and access the resulting information, the implementation organizes it into evidence-preserving, reusable-semantic, and attribute-relation records linked by one preference_id. Record-level matches discover candidates, whereas complete preference nodes are scored, ranked, and returned. The implementation thereby preserves the attribute constraints from memory construction through final retrieval.

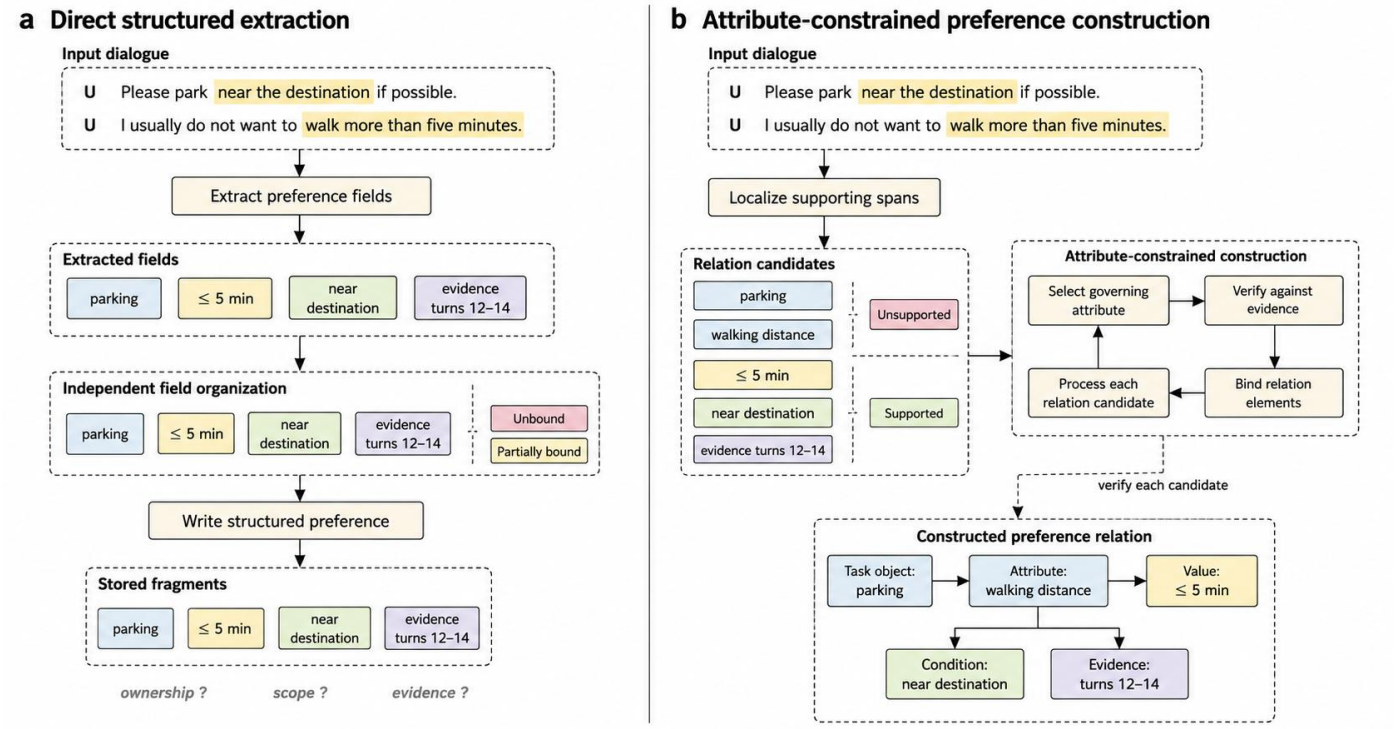

Figure. 1 Comparison between direct structured extraction and attribute-constrained preference construction

The main contributions of this paper are as follows. First, it identifies the lack of explicit attribute binding as a central limitation of direct preference extraction and formulates long-term in-vehicle preference memory as an attribute-constrained information-construction problem. Second, it introduces a construction method that localizes supporting interaction spans, forms reusable preference relations under explicit attribute constraints, and verifies auxiliary candidates against the same evidence. Third, it implements the method through function-specific records and node-level retrieval and evaluates both the core construction strategy and these implementation choices using preference-level and attribute-level metrics on CarMem and manually annotated real-world in-vehicle interactions.

The remainder of this paper is organized as follows. Section 2 reviews existing research; Section 3 introduces the proposed framework; Section 4 describes the datasets, comparison systems, and evaluation metrics; Section 5 reports the experimental results; Section 6 discusses the main findings, scope of applicability, and limitations; and Section 7 concludes the paper.

2. RELATED WORK

2.1 Long-Term Memory for Large Language Model Agents

Large language model agents typically rely on external memory mechanisms to maintain consistency across turns, sessions, and tasks (Park et al., 2023). Constrained by context-window length and inference costs, models cannot continuously load all historical information during every interaction and therefore require external storage to write, update, retrieve, and inject historical information.

Fact- and user-profile-based methods generally extract information with long-term value from dialogues and form compact user memories through deduplication, updating, and conflict resolution. Chhikara et al. (2025) identify salient facts from multi-turn interactions and use entity relations to enhance long-term information organization. J. Zhang et al. (2026) separate online memory access from offline memory consolidation and reduce online interaction costs through lightweight retrieval and background compression. Such methods can improve the efficiency of cross-session fact retention and personalized information reuse and are suitable for managing relatively independent user facts, preference statements, and historical states.

Event-based and episodic memory methods emphasize raw interactions and their context. Wang et al. (2026) reduce the information loss caused by fact extraction and summary compression by preserving specific interaction experiences and abstract user information. These methods retain temporal order, dialogue participants, event processes, and local context, which is conducive to handling subsequent queries that depend on raw expressions or event backgrounds. However, their memory units are generally still centered on complete events or dialogue segments, and their internal attribute relations are not necessarily represented explicitly.

Dynamic-note and graph-structured methods focus on the relationships among memories and their evolution. Xu et al. (2025) and Yu et al. (2026) incorporate memory writing, connection, updating, summarization, and forgetting into the agent decision-making process, allowing memory to evolve continuously according to task states and new evidence. Rasmussen et al. (2025) use a graph-memory method to organize long-term interaction content through entity nodes, relational edges, and temporal information, supporting cross-event connections, relational reasoning, and state-change modeling.

Existing long-term memory systems have advanced fact retention, episodic fidelity, dynamic updating, and relational organization, respectively, but their basic memory units are typically facts, profile fields, events, notes, graph nodes, or relational edges. For in-vehicle user preferences, a reusable memory often simultaneously contains supporting evidence, stable intent, attribute types, attribute values, entity objects, and applicability conditions. Existing general-purpose long-term memory methods seldom specifically discuss the ownership relationships among this information within the same preference, nor do they often ensure that retrieval results can be returned as complete preference units.

2.2 Long-Term User Preference Modeling and Representation

User preference modeling primarily serves recommender systems. Studies generally construct user representations from ratings, clicks, and historical choices and predict users' inclinations toward candidate items (Y. Zhang et al., 2018). Conversational recommender systems further acquire user-relevant attributes and constraints proactively through multi-turn question answering (Gao et al., 2021). Gao et al. (2021) summarize existing conversational recommendation research into such directions as preference elicitation, dialogue strategy, recommendation generation, and user simulation; K. Xu et al. (2021) improve preference elicitation efficiency in multi-turn dialogues through attribute selection and user-representation updating. These methods have advanced preference modeling from static behavioral inference toward interactive attribute elicitation, but their primary goal remains to improve recommendation results in the current session, with less attention paid to the long-term preservation and source tracing of cross-session preferences.

With the development of large language models, researchers have begun to use user profiles and historical content to directly control model outputs. Salemi et al. (2024) model personalization as retrieving relevant information from user histories and using it for text generation or classification, demonstrating the importance of historical information retrieval for personalized generation. Long-term dialogue research further focuses on cross-session user modeling. Li et al. (2025) divide long-term personalized dialogue into event awareness, persona extraction, and response generation and use long-term and short-term event memory to support continuous interaction. These studies show that long-term personalization requires the simultaneous management of specific historical events and abstract user characteristics.

In the in-vehicle domain, Kirmayr et al. (2025) extract, store, maintain, and retrieve in-vehicle user preferences around predefined categories and construct a multi-turn, multi-session dataset for in-vehicle voice assistants. Category constraints improve the transparency and maintainability of preference extraction and provide a task-specific foundation for long-term in-vehicle personalization. However, categorized preference entries still primarily emphasize preference classification, updating, and overall retrieval. They do not specifically address how raw evidence, stable intent, attribute values, entities, and applicability conditions within a preference can be preserved simultaneously, or how local attribute cues can participate in retrieval without forming independent memories.

2.3 Memory Retrieval and Logical Return Units

Long-term memory systems typically select historical information relevant to the current query through sparse retrieval, dense vector retrieval, hybrid retrieval, and reranking mechanisms (Yang et al., 2025). Sparse retrieval can use exact matching among keywords, entities, numerical values, and short texts, while dense retrieval can handle semantic similarity, synonymous expressions,

and query reformulation (Julianto et al., 2026). Hybrid retrieval improves candidate coverage by combining lexical and semantic signals, and cross-encoders, generative scoring, and task rules can further adjust candidate rankings (Liakhnovich et al., 2025).

In retrieval-augmented generation and long-term memory systems, combinations of BM25 and vector retrieval, rank-fusion methods such as Reciprocal Rank Fusion, and similarity-based score fusion are widely used to integrate multiple retrieval channels (Yang et al., 2025). Methods such as Maximal Marginal Relevance further balance relevance against candidate diversity (Carbonell & Goldstein, 1998). Multi-query expansion and query reformulation alleviate insufficient information in the original query by generating retrieval cues in different forms (Abirami et al., 2025). These methods primarily address candidate coverage and ranking integration across different retrieval signals.

Preference-memory retrieval has stricter return requirements. A user query may contain only a local cue. Attribute, entity, value, and condition records can respond effectively to such local cues, but an individual local fragment generally cannot independently express a complete user preference (P. Wang et al., 2025). The system ultimately needs to recover the preference intent to which the cue belongs, other attributes, applicability conditions, and their supporting evidence.

Existing research on hybrid retrieval, rank fusion, score fusion, and reranking primarily focuses on integrating results among information items (Liakhnovich et al., 2025), generally assuming that each candidate can participate independently in final ranking. When multiple physical records and auxiliary attributes jointly belong to one logical preference, existing methods seldom explicitly handle the node ownership of local records and its effect on top-K competition. For attribute-level preference recall, the retrieval mechanism needs to allow local cues to trigger candidate discovery while completing score fusion, ranking, and complete-content return at the level of logical preference nodes.

2.4 Functional Information Organization and Node Binding

Cognitive memory research provides functional references for assigning representational roles, enabling cue-based access, and binding information in artificial long-term memory systems. Tulving's distinction between episodic and semantic memory shows that specific experiences and the contexts in which they occur serve different functions from abstract knowledge reusable independently of particular events (Tulving, 1984). In long-term user preferences, raw interactions preserve sources, original wording, and local context, whereas stable preference relations support reuse across expressions and tasks.

The encoding specificity principle states that memory retrieval depends on the match between current retrieval cues and encoded traces (Tulving & Thomson, 1973). Subsequent queries from in-vehicle users may resemble the original expression, or they may contain only a task goal, attribute name, entity, value, or applicability condition. Different query cues have different matching relationships with different memory representations; therefore, the same preference needs multiple functionally explicit access points.

Research on the episodic buffer and feature binding in working memory emphasizes that information from different sources or dimensions must form an object-level representation with unified ownership (Baddeley, 2000). Attributes, values, entities, and conditions can support stable identification and subsequent use only when their mutual relationships are preserved. In preference memory, local attribute information needs to participate in retrieval, but its associated user preference, supporting evidence, and applicability context must not be lost during representation and return.

These cognitive theories motivate one construction requirement and two supporting implementation requirements. Feature binding requires attributes, values, entities, conditions, and evidence to be constructed under unified preference ownership. The distinction between episodic and semantic memory supports storing original evidence and reusable semantics in functionally differentiated records, while encoding specificity supports access paths for different query cues. Node binding is then required to ensure that these access paths return the complete preference constructed under the attribute constraints.

In summary, existing research has established an important foundation in cross-session information retention, long-term preference modeling, structured memory organization, and multi-channel retrieval, but a common gap remains: preference information is seldom constructed under explicit constraints that keep attributes, values, entities, conditions, and evidence within one reusable relation. Mixing this information in one representation or ranking its fragments independently are downstream implementation problems that further weaken access and ownership. This paper addresses the construction gap through attribute-constrained information construction and uses Function-specific record and node-level retrieval to preserve the constructed relations during storage, ranking, and return.

3. METHOD

3.1 Overview of the Proposed Framework

The proposed method addresses a construction failure in existing preference-memory pipelines. Direct structured extraction can omit task-critical fields or detach values, entities, conditions, and evidence from the attributes they qualify. If this ownership is not established during construction, later storage may either mix heterogeneous content in one representation or split local information into independently ranked fragments. The method therefore uses attributes as explicit construction constraints and preserves the resulting relations through supporting organization and retrieval implementations.

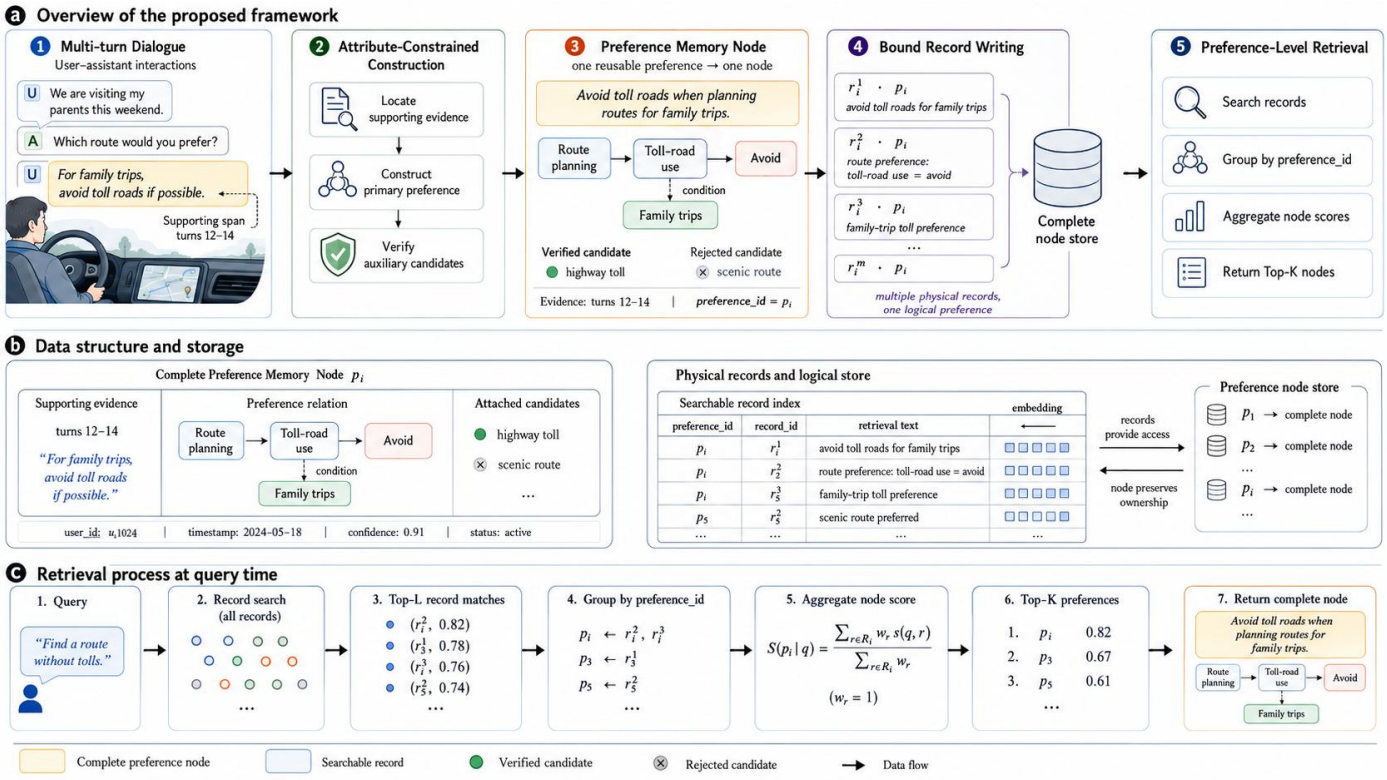

Figure. 2 Overview of the proposed preference-memory framework.

As shown in Figure.2, attribute-constrained information construction localizes supporting dialogue spans, constructs a primary preference relation under explicit attribute bindings, generates auxiliary relations, and verifies them against the same evidence. After construction, Function-specific record separates evidence-preserving, reusable-semantic, and attribute-relation content into physical access records that share one preference_id. Node-level retrieval searches these records but groups, scores, and returns results at the preference-node level. The latter two stages implement access to the constructed information without changing its logical ownership.

During the writing stage, Qwen2.5-14B-Instruct deployed through an OpenAI-compatible interface is used. Retrieval texts are encoded using sentence-transformers/all-MiniLM-L6-v2 and stored in a local Qdrant vector index. The specific runtime parameters and software environment are reported in the experimental setup.

The method places attribute-constrained construction, candidate verification, and record organization in the offline memory-writing stage while keeping the supporting online retrieval procedure lightweight. This division is suitable for in-vehicle dialogue assistants, where reusable preference information can be constructed after a session or in the background, but access during interaction must satisfy low-latency and stable-ranking requirements

3.2 Attribute-Constrained Information Construction

Figure.3 illustrates the detailed process of attribute-constrained preference information construction using an illustrative dialogue example. The method localizes preference-bearing evidence, constructs a primary relation under explicit attribute ownership, verifies auxiliary candidates through evidence-constrained arbitration, and retains the resulting information under the same attribute ownership.

Attribute-constrained information construction comprises four operations: evidence localization, primary-relation construction, auxiliary-relation generation, and evidence arbitration. The system first identifies the dialogue spans that support a reusable preference judgment. It then constructs the primary relation and supplementary relations under explicit links to those spans. Only information supported by the raw user-assistant dialogue and reusable in subsequent tasks of the same type is admitted to memory;

assistant responses may supply selected entities, execution results, or scenario information but cannot independently establish a user preference.

The construction schema uses the attribute relation as its binding anchor. Each relation records the domain, task object, attribute, raw value, normalized value, value type, condition and evidence pointer. A value is valid only when its attribute and object are identified, and entities, units, conditions remain attached to the relation they qualify. Evidence pointers connect the constructed relation to its factual basis. These constraints prevent short values or local conditions from becoming detached facts and preserve the information needed for later execution and source tracing.

In addition to the primary preference relation, the method generates open relations as auxiliary candidates to increase information coverage and identify omitted entities, values, attributes, or conditions. Candidates are retained only when they can be mapped to explicit evidence and satisfy the same attribute constraints as the primary relation. To control generative variability, the primary constructor, auxiliary generator, and evidence arbiter use fixed input schemas, output schemas, and prompt templates. Outputs with parsing failures, missing required fields, unsupported content, or no valid mapping to an evidence turn are rejected.

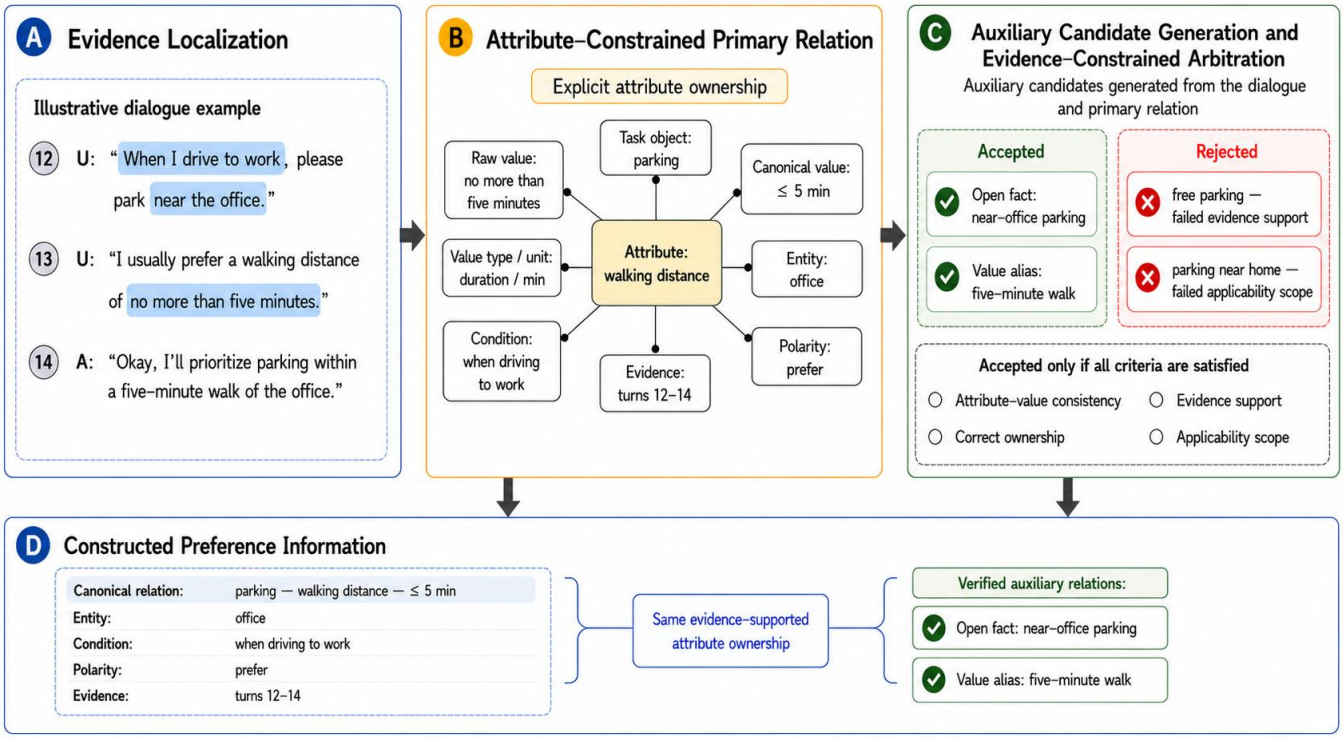

Figure. 3 Detailed process of attribute-constrained preference information construction using an illustrative dialogue example.

3.3 Function-specific record

To make the information produced by attribute-constrained construction retrievable, the implementation organizes it according to access function rather than serializing it into one overloaded text. Each reusable preference forms one logical Preference Memory Node supported by one or more dialogue segments. Within the node, function-specific physical records provide distinct access paths while retaining the constructed associations among evidence, reusable semantics, attributes, values, entities, and applicability conditions through a shared preference_id.

The evidence-preserving record stores the dialogue turns that support the preference, including original turn order, entity names, action results, user wording, and local context. Retaining assistant utterances can supplement selected entities, execution results, and contextual information absent from the user input. This record supports source tracing, raw-language matching, preference verification, and error analysis.

The reusable-semantic record expresses the stable meaning constructed from the evidence as a concise natural-language preference statement. It preserves the task object, attribute, selected value, entity, and condition while abstracting away incidental wording. When a subsequent query uses a different expression or states only the current task, this record provides a semantic access path to the owning node.

The attribute-relation record preserves entity, attribute, raw_value, canonical_value, value_type, and condition around the primary preference relation. It maintains explicit attribute-value binding and stores short values and entities together with the context that gives them meaning. Evidence-verified auxiliary candidates are attached to this record under the same preference_id and remain distinguishable from the canonical relation, but they are not indexed or ranked as separate memory records.

Node metadata stores management and verification information, including node identifier, user identifier, timestamp, evidence source, confidence, status, attribute fields, and condition constraints. Separating retrieval text from metadata preserves natural-language access while supporting result interpretation, attribute verification, node updating, and experimental evaluation. Functionalized organization is therefore an implementation choice for exposing the constructed information to retrieval, whereas the preference node retains unified logical ownership.

3.4 Node-Level Retrieval and Complete-Preference Return

To retrieve the attribute-constrained preference information, the system encodes the current query and searches all active function-specific records within the current user's memory space. Each physical record carries a preference_id and a record-type label. Record-level similarity is used only to discover candidate nodes: hits from different record types are merged by preference_id, and no physical record is treated as an independently returnable result. The online process therefore consists of query encoding, vector search, candidate-node construction, and node aggregation, without invoking the large language model used during memory construction.

For each candidate node, the system collects the similarity scores of records associated with its preference_id. If multiple records serve the same retrieval function, only the maximum similarity for that function is retained. The remaining function-level scores are combined as a normalized weighted average. This aggregation separates candidate discovery from final ranking and prevents the number of physical records from increasing a node's score by itself. Nodes are ranked by the fused score, and the Top-K preference_ids are used to reconstruct complete preference information from the node store. Auxiliary candidate content can make its owning node easier to discover, but it never receives an independent rank position. Physical records thus provide retrieval access paths, whereas the logical preference node remains the unit of scoring and return.

4. EXPERIMENTAL SETUP

4.1 Datasets

This paper uses a mixed preference-memory evaluation set composed of CarMem and manually annotated real-world in-vehicle data. The two types of data use a unified preference-representation format, user-level memory partitioning, and query-evaluation protocol and are jointly used for the overall comparison and ablation experiments.

CarMem contains 100 users, 1,000 user preferences, and 4,000 test queries. Each preference corresponds to four types of queries: retrieval, equal, negate, and different. The data provide target preference, attribute, and value annotations and can be used to evaluate preference-level and attribute-level recall.

The real-world in-vehicle data are derived from actual HMI interaction logs and contain 188 sessions, 14,057 turns, and 13,612 valid ASR text turns. The data cover interactions with drivers and passengers and include scenarios involving windows, volume, air conditioning, multimedia, navigation, ordering, and application control. In addition to natural-language inputs, the logs provide structured signals such as domain, intent, tool call, tool parameters, memory update, and outcome.

Based on the real-world in-vehicle logs, this paper manually annotates gold preferences. Each annotation record contains the raw evidence supporting the preference, preference intent, attribute, raw value, normalized value, applicability condition. The annotation process distinguishes long-term preferences from one-time operations, short-term contextual states, and system-execution records.

The experiments use independent user-level memory spaces. Each query performs retrieval only within the preference memory of its associated user. Overall results are calculated on the mixed query set, while stratified results for the CarMem subset and the real-world in-vehicle subset are retained to analyze differences in method performance between standardized data and real-world interactions.

4.2 Compared Systems and Experimental Conditions

The external-system comparison selects LightMem, A-Mem, MemMachine, Mem0, and Graphiti, covering lightweight fact memory, dynamic-note memory, event-based episodic memory, general-purpose memory services, and graph-structured memory, respectively. All systems use the same user-level memory boundaries and query set.

The internal experiments first test the core construction claim and then analyze two supporting implementation choices. The construction experiment compares direct structured extraction, post-extraction evidence arbitration, and attribute-constrained information construction. The record-organization analysis compares an evidence-centered single record, a relation template, a Graph-to-Text narrative, a unified mixed representation, and function-specific records. The candidate-access analysis compares no auxiliary candidates, candidates as independent memories, candidates appended to existing text, and candidates bound to the owning node. The evaluation settings are top-K=5 and candidate-K=20.

4.3 Implementation Environment

The experimental environment consists of Python 3.8.10, sentence-transformers 3.2.1, PyTorch 2.4.1, and qdrant-client 1.12.1; the host contains two Intel Xeon Platinum 8457C processors, 1.7 TiB RAM, and eight NVIDIA H20 GPUs. Encoding and retrieval are executed on the CPU, and Qwen is provided by an independent service.

The writing stage uses ModelScope Qwen2.5-14B-Instruct. The primary-relation and auxiliary candidate generation calls use temperature=0.0 and a concurrency of 4. The maximum outputs are three primary preferences for the primary-relation call and six candidates per candidate type for the auxiliary candidate call. The inputs contain turn-indexed user-assistant dialogues together with the corresponding fixed prompt templates.

The function-specific retrieval records are encoded by all-MiniLM L6-v2 as 384-dimensional normalized vectors. Encoding is executed on the CPU with a batch size of 64 and FP32. Vector storage uses local Qdrant through qdrant-client 1.12.1 with cosine distance, and an independent collection is established for each user and experimental variant.

During retrieval, record-level hits are merged by preference_id to form candidate nodes. For each candidate node, the raw cosine similarities of its associated function-specific records are aggregated using equal weights as a parameter-free setting. When duplicate records serve the same function, the maximum similarity is retained before aggregation. The system retains the top20 nodes and calculates the primary metrics using the top5. All internal experiments hold the language model, encoding, and retrieval configurations constant and change only the core construction strategy or one supporting implementation choice.

4.4 Metrics

This paper uses Hit@5, Attr@5, Recall@20, AttrRecall@20, MRR@5, AttrMRR@5, and Retrieval Latency. Hit@5 and Recall@20 measure whether the target preference node enters the top 5 results and top 20 candidates, respectively; Attr@5 and AttrRecall@20 further require the retrieved node to contain the correct gold attribute and value. MRR@5 evaluates the first ranking position of the target preference within the top 5 results, and AttrMRR@5 evaluates the ranking position of the first attribute-level correct result. Retrieval Latency represents the mean online time from the query entering vector retrieval to the return of ranked preference nodes.

5. RESULTS

5.1 Overall Comparison with Existing Memory Systems

Table 1 compares the complete implementation of the proposed attribute-constrained construction method with five external long-term memory systems.

Table 1 Overall Comparison of the Proposed Framework with Existing Long-Term Memory Systems

Method	Hit@5	Attr@5	Recall@20	AttrRecall@20	MRR@5	AttrMRR@5	Retrieval Latency
LightMem	76.75%	70.38%	76.75%	70.38%	63.42%	58.71%	**14.03 ms**
A-Mem	94.03%	83.65%	94.80%	85.00%	87.10%	73.30%	16.84 ms
MemMachine	77.08%	69.77%	85.55%	80.73%	62.63%	56.04%	863.57 ms
Mem0	73.75%	66.12%	76.60%	68.78%	63.30%	57.50%	124.51 ms
Graphiti	54.57%	50.65%	56.53%	52.53%	46.80%	43.60%	29.02 ms
Proposed method	**98.28%**	**93.50%**	**99.30%**	**95.60%**	**93.05%**	**86.41%**	21.05 ms

The proposed framework achieves the highest results on all recall and ranking metrics. Its Attr@5 reaches 93.50%, an improvement of 9.85% over the strongest-performing external baseline, indicating that attribute-level correct results appear more consistently near the top of the node-ranked retrieval results.

LightMem, Mem0, and MemMachine all use fact- or event-level memory units, and their Attr@5 values are 70.38%, 66.12%, and 69.77%, respectively. These results are lower than their corresponding Hit@5 values, indicating that retrieving relevant memories does not directly guarantee the complete preservation of attributes and values. Graphiti's Hit@5 and Attr@5 on the current task are 54.57% and 50.65%, respectively, indicating that graph-structured organization does not provide an effective advantage for attribute-level recall on this dataset.

The proposed method has a mean retrieval latency of 21.05 ms. Compared with A-Mem, the strongest-performing external baseline, the supporting node-level grouping and fusion procedure adds 4.21 ms of mean retrieval time while achieving better preference-level recall, attribute-level recall, and ranking quality.

5.2 Effect of Attribute-Constrained Information Construction

This experiment compares direct structured extraction, post-extraction evidence arbitration, and evidence-first writing to examine the role of evidence in preference abstraction. The results are shown in Table 2. This experiment tests the core claim that explicit attribute constraints improve the construction of reusable preference information over direct extraction. Direct structured extraction treats attributes, values, entities, and conditions produced from the full dialogue as the final memory artifact. Post-extraction evidence arbitration checks those outputs against raw evidence and auxiliary candidates but does not reconstruct missing ownership. The proposed construction method instead localizes supporting spans before relation formation and requires values, entities, conditions, units, polarity, and evidence to remain bound to the attribute relation.

Table 2 Comparison of Attribute-Constrained Preference Construction Strategies

Method	Hit@5	Attr@5	MRR@5	AttrMRR@5	Retrieval Latency
Early unified structured extraction	89.38%	72.10%	85.51%	65.33%	16.26 ms
Structured extraction + evidence-grounded arbiter	92.15%	76.45%	87.48%	68.78%	17.42 ms
Attribute-constrained information construction	95.78%	82.80%	92.12%	74.35%	18.05 ms

Adding evidence arbitration after direct structured extraction increases Hit@5 and Attr@5 by 2.77% and 4.35%, respectively. Verification against the raw dialogue therefore reduces erroneous, overgeneralized, and incomplete memory writes, but it still operates after the initial abstraction has already been produced.

Attribute-constrained information construction further increases Hit@5, Attr@5, MRR@5, and AttrMRR@5 to 95.78%, 82.80%, 92.12%, and 74.35%. Compared with direct structured extraction, Attr@5 and AttrMRR@5 increase by 10.70% and 9.02%. The larger improvement at the attribute level shows that the benefit is not merely extracting additional fields: establishing evidence-supported attribute ownership during construction preserves the relations among entities, values, conditions, and the preferences they qualify.

5.3 Implementation Analysis: Function-specific record

This implementation analysis examines how attribute-constrained preference information should be represented for retrieval. Relation template append adds structured fields to evidence text; Graph-to-Text converts relations into rule-based narratives; Unified Mixed Representation repeats evidence, preference semantics, and attribute information in each retrievable text; Function-Specific Records assign these access roles to separate physical records bound to one node. The results are shown in Table 3.

Relation template append and Graph-to-Text narrative both show performance degradation after the addition of more structured information. Compared with the attribute-constrained single-record condition, their Hit@5 values decrease by 14.55% and 20.75%, respectively, and their Attr@5 values decrease by 10.12% and 10.52%, respectively. This result indicates that increasing the number of structured fields cannot substitute for a retrieval representation suitable for dense retrieval, and fixed templates and rule-based narratives weaken the semantic distinctiveness of the raw evidence.

Under the same mean node-scoring rule, Unified Mixed Representation achieves higher Hit@5 and MRR@5, whereas function-specific organization improves Attr@5 and AttrMRR@5 by 1.95% and 2.74%. Repeating evidence, schema, and values across records can increase lenient target matches, but function-specific records more effectively recover the correct attribute relations.

Functionalized records are therefore an effective retrieval implementation for the constructed information, rather than an alternative information-construction method.

Table 3 Comparison of Different Preference-Memory Construction and Representation Strategies

Method	Hit@5	Attr@5	MRR@5	AttrMRR@5
Attribute-constrained single record	95.78%	82.80%	92.12%	74.35%
Relation template append	81.23%	72.68%	75.65%	66.08%
Graph-to-Text narrative	75.03%	72.28%	68.71%	65.96%
Unified Mixed Representation	**99.18%**	91.55%	**96.79%**	83.67%
Function-Specific Records	98.28%	**93.50%**	93.05%	**86.41%**

5.4 Comparison of Attribute Candidate Organization Strategies

This implementation analysis compares ways of exposing auxiliary attribute candidates to retrieval. The no-candidate condition retains only evidence-centered and canonical preference records. The results are shown in Table 4. Independent-memory conditions write open facts, or open facts plus Entity-Value-Condition candidates, as separately ranked vector records. Text-appending conditions add candidate content to one or both existing retrieval records. The node-bound condition stores normalized, evidence-verified candidates in the attribute-relation record and links that record to the same preference_id as the evidence-preserving and reusable-semantic records.

Table 4 Comparison of Different Attribute-Candidate Organization Methods

Experimental Condition	Hit@5	Attr@5	MRR@5	AttrMRR@5	Retrieval Latency
No attached candidates	95.78%	82.80%	92.12%	74.35%	18.05 ms
Open facts as independent memories	92.85%	83.05%	79.73%	72.36%	17.31 ms
Open facts and EVC as independent memories	90.70%	83.62%	78.29%	73.30%	17.80 ms
Candidates appended to evidence text	98.40%	87.85%	95.66%	81.54%	21.86 ms
Candidates appended to canonical and evidence text	**98.50%**	88.10%	**95.71%**	81.57%	21.53 ms
Candidates bound to the owning preference node	98.28%	**93.50%**	93.05%	**86.41%**	21.05 ms

When candidates are written as independent memories, Attr@5 achieves only limited gains, while Hit@5, MRR@5, and AttrMRR@5 decrease significantly. Local facts therefore compete with complete preference nodes for limited Top-K positions.

Appending candidates to existing texts can improve attribute-level results, but it also mixes evidence semantics with relational fields. Binding candidates to the attribute-relation record provides both local-cue access and explicit logical ownership and achieves the best attribute-level results with the construction method held fixed. Compared with no attached candidates, Attr@5 and AttrMRR@5 increase by 9.03% and 11.28%. Node-bound candidate access is therefore the more effective retrieval implementation for verified auxiliary relations.

6. DISCUSSION

6.1 Attribute Constraints as the Core of Information Construction

The core construction experiment shows that preference-memory quality is already affected by the writing process before vector retrieval. Direct structured extraction can produce manageable fields, but its one-step abstraction can omit raw entities, conditions, units, and the interaction relationships between users and assistants (Xu et al., 2024). Post-extraction arbitration partially repairs unsupported outputs, whereas attribute-constrained construction improves attribute-level recall and ranking more substantially because ownership is established while reusable relations are formed.

Attribute constraints define both admissibility and relational ownership. Each value, entity, condition, unit, and polarity must remain attached to a specified attribute and task object, while the evidence pointer establishes the factual basis of that relation. This distinction is especially important in in-vehicle dialogue, where one-time operations, assistant execution results, environmental conditions, and long-term user inclinations may coexist. Without these constraints, a system can generalize temporary instructions into long-term preferences or omit the conditions that delimit a preference's scope.

Evidence is therefore a component of attribute-constrained construction rather than a separate top-level method. Evidence localization controls which dialogue content can support a preference, and evidence arbitration rejects unsupported primary or auxiliary relations. Together with explicit attribute binding, these operations provide information fidelity, source tracing, and a stable basis for later updates.

6.2 Functionalized Organization as a Retrieval Implementation

The record-organization analysis shows that the information produced by attribute-constrained construction still requires an appropriate retrieval representation. Relation templates and Graph-to-Text narratives add structured content to raw evidence, but preference-level and attribute-level retrieval performance decline simultaneously. Repeated fields, generic schema tokens, and rule-based narratives change the semantic focus of dense representations and reduce the distinctiveness among different preferences.

The comparison between Unified Mixed Representation and function-specific records further shows that repeating evidence, preference semantics, attributes, and values across texts increases opportunities for local lexical or semantic matches and therefore produces higher Hit@5 and MRR@5. Function-specific records perform better on Attr@5 and AttrMRR@5, indicating that differentiated access roles are more conducive to retrieving the attribute relations created by the construction method.

Lenient relevance remains a necessary foundation, but correct attribute relations correspond more directly to downstream personalized decisions. In-vehicle tasks such as navigation, parking, charging, media selection, and cockpit control require explicit attributes, values, entities, units, and applicability conditions (Dong et al., 2025). Functionalized organization is therefore used as an implementation trade-off: it sacrifices a small amount of preference-level Hit@5 and MRR@5 while improving access to the attribute relations constructed for execution-oriented tasks.

6.3 Node Ownership in the Retrieval Implementation

The candidate-access analysis shows that verified auxiliary relations require simultaneous control of retrieval visibility and logical ownership. Open facts, entity-value-condition relations, value aliases, and composite conditions can supplement information omitted from the primary preference relation, but their retrieval behavior depends on how they are connected to the preference constructed under the attribute constraints.

When candidates are written as independent memories, local attribute information can respond directly to queries, but the Hit@5 and MRR@5 of complete preference nodes decrease. Independent candidates often contain entities, short values, or conditions highly similar to a query and can therefore occupy limited Top-K positions. Such fragments lack complete interaction evidence, stable preference semantics, and applicability scope; strong local matching does not mean that they can independently support personalized decisions.

Appending candidates to evidence-oriented or reusable-semantic text can improve attribute recall, confirming that auxiliary candidates contain useful retrieval signals. However, the observed results suggest that appending heterogeneous content to the same retrieval text can dilute locally discriminative semantic cues.

The node-bound attribute-relation record preserves the ownership established during construction through preference_id. It distinguishes the primary relation from auxiliary candidates while retaining their evidence and attribute links, giving local information a retrieval access point without granting it an independent rank position. Compared with no attached candidates, this implementation produces the largest improvements in Attr@5 and AttrMRR@5.

Node ownership is therefore a retrieval constraint that protects the output of attribute-constrained construction. Additional relations may expand local coverage while also causing fragmentation and ranking competition if they are returned independently. Execution-oriented preference systems should retain complete reusable preferences as logical memory units and expose entities, short values, aliases, and conditions as node-bound access points.

6.4 Attribute-Level Evaluation for Executable Preference Memory

The difference between preference-level and attribute-level metrics indicates that traditional memory-recall evaluation may overestimate a system's actual ability to support in-vehicle personalization tasks. Hit@K only requires the target memory to enter the result set and does not require the system to recover the correct attributes, values, entities, and conditions within it. A system

may retrieve a topically relevant historical experience but be unable to extract the specific preference relation required by the current task.

This difference is reflected in both external memory systems and internal organization strategies. Most systems have lower Attr@5 than Hit@5, indicating that retrieving a target experience does not guarantee recovery of its attribute relations. The different rankings of unified mixed representation and function-specific records on the two metric families further show that general preference-level metrics and attribute-level metrics evaluate distinct capabilities: the former measure accessibility of the target node, whereas the latter measure whether the relations within that node can support precise decisions.

For in-vehicle systems, this distinction has practical significance. Navigation tasks need to determine whether toll roads or highways should be avoided; parking tasks need to recover walking-distance, price, and location requirements; and cockpit-control tasks need to recover the target temperature, seat, airflow level, and applicability conditions. Topically relevant memories cannot directly generate these execution parameters; correct attribute relations are the direct inputs to personalized decisions.

6.5 Limitations and Future Work

6.5.1 Limitations

The current study focuses primarily on preference construction and recall and does not yet cover the complete preference lifecycle. User preferences may change with time, vehicle state, accompanying persons, and task conditions, while temporary exceptions, preference replacement, temporal decay, and memory invalidation still require further modeling.

The current study primarily validates the retrieval capability of preference memory and has not fully evaluated the actual impact of preference memory on in-vehicle decisions and execution results. Attribute-level recall corresponds more directly to execution-oriented tasks, but retrieving the correct attributes does not necessarily guarantee that response generation, tool-parameter construction, and vehicle control are all correct. Error propagation and decision coupling remain between memory retrieval and downstream agent reasoning.

The current method primarily targets user preferences that can be supported by dialogue evidence and expressed as relations among objects, attributes, values, and conditions. This modeling approach is suitable for attribute-oriented tasks such as navigation, parking, media selection, and cockpit control, but its coverage of more complex forms of preference requires further validation, including multi-objective trade-offs, interaction styles, behavioral habits, procedural strategies, and implicit preferences that can only be inferred through multiple interactions.

6.5.2 Future Work

Future research should expand dynamic preference-management mechanisms. The Preference Memory Node can further incorporate temporal validity, contextual scope, preference strength, conflict relations, and replacement status, allowing the system to distinguish among long-term preferences, temporary requests, and contextual exceptions and to reinforce, correct, decay, and invalidate preferences on the basis of new evidence.

Future research should also conduct end-to-end validation of in-vehicle tasks. Retrieved preference nodes should be integrated into response generation, tool-parameter construction, and vehicle-function execution, with comprehensive evaluation of attribute-parameter accuracy, task-completion rate, inappropriate preference-application rate, and users' subjective experience.

Future research also needs to investigate multi-objective preferences, implicit inclinations, interaction strategies, and cross-context behavioral patterns and validate the method in long-term, continuous, multi-user real-world cockpit interactions. Expanding the types of preferences and the scope of tasks can further test the cross-scenario applicability of attribute-constrained information construction and its organization and retrieval implementations.

7. CONCLUSION

This paper investigates how reusable long-term in-vehicle preferences should be constructed. Existing pipelines often reduce memory writing to direct information extraction without explicit constraints on how attributes, values, entities, conditions, and evidence remain related. The resulting artifact may lose task-critical details, mix heterogeneous content in one representation, or fragment local information into independently ranked candidates.

The paper proposes attribute-constrained information construction as the core method. It localizes raw interaction evidence, constructs reusable preference relations under explicit attribute bindings, and verifies auxiliary relations against the same evidence. Functionalized records and node-level retrieval implement storage and access: they expose different query cues while preserving one preference_id as the unit of scoring and return. Experiments show that attribute-constrained construction outperforms direct

structured extraction, while the implementation analyses show that function-specific records and node-bound candidates provide more effective access to the constructed attribute relations.

These findings show that the quality of long-term preference memory is determined primarily by how preference information is constructed. Explicit attribute constraints preserve the relations required for source tracing, reuse, and execution; functionalized organization and node-level retrieval preserve and expose those relations during storage and access. This hierarchy provides a focused methodological basis for traceable and execution-oriented preference memory in in-vehicle dialogue agents.

REFERENCES

Abirami, S., Joshma, M. J., Jayavarthini, A., Joshi, A., & Gajendran, M. K. (2025). Evaluating Lexical, Dense, and Hybrid Retrieval Pipelines for RAG. *2025 IEEE Pune Section International Conference (PuneCon)*, 1–6. https://doi.org/10.1109/punecon67554.2025.11379254

Andreas, J., Bufe, J., Burkett, D., Chen, C., Clausman, J., Crawford, J., Crim, K., DeLoach, J., Dorner, L., Eisner, J., Fang, H., Guo, A., Hall, D., Hayes, K., Hill, K., Ho, D., Iwaszuk, W., Jha, S., Klein, D., . . . Zotov, A. (2020). Task-Oriented dialogue as dataflow synthesis. *Transactions of the Association for Computational Linguistics*, *8*, 556–571. https://doi.org/10.1162/tacl_a_00333

Baddeley, A. (2000). The episodic buffer: a new component of working memory? *Trends in Cognitive Sciences*, *4*(11), 417–423. https://doi.org/10.1016/s1364-6613(00)01538-2

Balaraman, V., & Magnini, B. (2021). Domain-Aware Dialogue State Tracker for Multi-Domain dialogue systems. *IEEE/ACM Transactions on Audio Speech and Language Processing*, *29*, 866–873. https://doi.org/10.1109/taslp.2021.3054309

Carbonell, J., & Goldstein, J. (1998). The use of MMR, diversity-based reranking for reordering documents and producing summaries. *Proceedings of the 21st Annual International ACM SIGIR Conference on Research and Development in Information Retrieval*, 335–336. https://doi.org/10.1145/290941.291025

Chen, T., Lu, J., Shen, Y., & Zhang, L. (2026). ES-MemEval: Benchmarking Conversational Agents on Personalized Long-Term Emotional Support. *Proceedings of the ACM Web Conference 2026*, 5810–5821. https://doi.org/10.1145/3774904.3792143

Chen, Z., Liu, Y., Chen, L., Zhu, S., Wu, M., & Yu, K. (2023). OPAL: Ontology-Aware Pretrained Language Model for End-to-End Task-Oriented Dialogue. *Transactions of the Association for Computational Linguistics*, *11*, 68–84. https://doi.org/10.1162/tacl_a_00534

Chhikara, P., Khant, D., Aryan, S., Singh, T., & Yadav, D. (2025). Mem0: Building Production-Ready AI Agents with Scalable Long-Term Memory. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2504.19413

Dong, H., Wang, W., Sun, Z., Kang, Z., Ge, X., Gao, F., & Wang, J. (2025). Knowledge graph construction for intelligent cockpits based on large language models. *Scientific Reports*, *15*(1), 7635. https://doi.org/10.1038/s41598-025-92002-y

Fang, J., Deng, X., Xu, H., Jiang, Z., Tang, Y., Xu, Z., Deng, S., Yao, Y., Wang, M., Qiao, S., Chen, H., & Zhang, N. (2025, October 21). *LightMem: Lightweight and Efficient Memory-Augmented Generation*. arXiv.org. https://arxiv.org/abs/2510.18866

Firdaus, M., Madasu, A., & Ekbal, A. (2023). A Unified Framework for Slot based Response Generation in a Multimodal Dialogue System. *Multimedia Tools and Applications*, *83*(4), 11643–11667. https://doi.org/10.1007/s11042-023-15915-8

Gao, C., Lei, W., He, X., De Rijke, M., & Chua, T. (2021). Advances and challenges in conversational recommender systems: A survey. *AI Open*, *2*, 100–126. https://doi.org/10.1016/j.aiopen.2021.06.002

Gao, F., Ge, X., Li, J., Fan, Y., Li, Y., & Zhao, R. (2024). Intelligent Cockpits for connected vehicles: taxonomy, architecture, interaction technologies, and future directions. *Sensors*, *24*(16), 5172. https://doi.org/10.3390/s24165172

Hatalis, K., Christou, D., Myers, J., Jones, S., Lambert, K., Amos-Binks, A., Dannenhauer, Z., & Dannenhauer, D. (2024). Memory Matters: The need to improve Long-Term Memory in LLM-Agents. *Proceedings of the AAAI Symposium Series*, *2*(1), 277–280. https://doi.org/10.1609/aaaiss.v2i1.27688

Hou, Y., Tamoto, H., & Miyashita, H. (2024). “My agent understands me better”: Integrating Dynamic Human-like Memory Recall and Consolidation in LLM-Based Agents. *Extended Abstracts of the CHI Conference on Human Factors in Computing Systems*, 1–7. https://doi.org/10.1145/3613905.3650839

Hu, C., Huang, S., Zhang, Y., & Liu, Y. (2022). Learning to infer user implicit preference in conversational recommendation. *Proceedings of the 45th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 256–266. https://doi.org/10.1145/3477495.3531844

Huang, X., Lian, J., Lei, Y., Yao, J., Lian, D., & Xie, X. (2025). Recommender AI Agent: Integrating large language models for interactive recommendations. *ACM Transactions on Information Systems*, *43*(4), 1–33. https://doi.org/10.1145/3731446

Huang, Z., Tian, Z., Guo, Q., Zhang, F., Zhou, Y., Jiang, D., Xie, Z., & Zhou, X. (2025, November 3). *LiCoMemory: Lightweight and Cognitive Agentic Memory for Efficient Long-Term Reasoning*. arXiv.org. https://arxiv.org/abs/2511.01448

Joko, H., Chatterjee, S., Ramsay, A., De Vries, A. P., Dalton, J., & Hasibi, F. (2024). Doing personal LAPS: LLM-Augmented Dialogue Construction for Personalized Multi-Session Conversational Search. *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 796–806. https://doi.org/10.1145/3626772.3657815

Julianto, E., Evanza, R., Amir, A. D. H., Karim, M. I., Thohir, M., Magribi, W. P., Qusyairy, M. F., & Tjahjono, B. (2026). Adaptive Multi-Criteria Re-Ranking for Hybrid Semantic Information Retrieval in Metadata-Rich Collections. *2026 International Seminar on Intelligent Business and Edge-Computing Research (ISIBER)*, 950–955. https://doi.org/10.1109/isiber68248.2026.11470214

Kirmayr, J., Stappen, L., Schneider, P., Matthes, F., & André, E. (2025, January 16). *CarMem: Enhancing Long-Term Memory in LLM Voice Assistants through Category-Bounding*. arXiv.org. https://arxiv.org/abs/2501.09645

Li, H., Yang, C., Zhang, A., Deng, Y., Wang, X., & Chua, T. (2025). Hello Again! LLM-powered Personalized Agent for Long-term Dialogue. *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, 5259–5276. https://doi.org/10.18653/v1/2025.naacl-long.272

Liakhnovich, K., Lashinin, O., Babkin, A., Pechatov, M., & Ananyeva, M. (2025). SMMR: Sampling-Based MMR Reranking for Faster, More Diverse, and Balanced Recommendations and Retrieval. *Proceedings of the 48th International ACM SIGIR Conference on Research and Development in Information Retrieval*, 2754–2758. https://doi.org/10.1145/3726302.3730250

Liao, L., Long, L. H., Ma, Y., Lei, W., & Chua, T. (2021). Dialogue State Tracking with Incremental Reasoning. *Transactions of the Association for Computational Linguistics*, *9*, 557–569. https://doi.org/10.1162/tacl_a_00384

Liu, J., Su, Y., Xia, P., Han, S., Zheng, Z., Xie, C., Ding, M., & Yao, H. (2026). SimpleMem: Efficient Lifelong Memory for LLM Agents. arXiv preprint arXiv:2601.02553. https://doi.org/10.48550/arXiv.2601 .02553

Maharana, A., Lee, D., Tulyakov, S., Bansal, M., Barbieri, F., & Fang, Y. (2024). Evaluating very Long-Term Conversational Memory of LLM agents. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2402.17753

Murali, P. K., Kaboli, M., & Dahiya, R. (2021). Intelligent In-Vehicle Interaction Technologies. *Advanced Intelligent Systems*, *4*(2). https://doi.org/10.1002/aisy.202100122

Park, J. S., O’Brien, J., Cai, C. J., Morris, M. R., Liang, P., & Bernstein, M. S. (2023). Generative Agents: Interactive Simulacra of Human Behavior. *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, 1–22. https://doi.org/10.1145/3586183.3606763

Rasmussen, P., Paliychuk, P., Beauvais, T., Ryan, J., & Chalef, D. (2025, January 20). *Zep: A Temporal Knowledge Graph Architecture for Agent Memory*. arXiv.org. https://arxiv.org/abs/2501.13956

Salemi, A., Mysore, S., Bendersky, M., & Zamani, H. (2024). LaMP: When Large Language Models Meet Personalization. *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*, 7370–7392. https://doi.org/10.18653/v1/2024.acl-long.399

Sarin, S., Singh, L., Sarmah, B., & Mehta, D. (2025). Memoria: A Scalable Agentic Memory Framework for Personalized Conversational AI. *2025 5th International Conference on AI-ML-Systems (AIMLSystems)*, 32–39. https://doi.org/10.1109/aimlsystems67835.2025.11330332

Tian, A., Lu, Y., Fan, X., Wang, C., Zhou, L., Zhang, Y., & Liu, Y. (2025). RGMEM: Renormalization Group-inspired Memory Evolution for Language agents. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2510.16392

Tulving, E. (1984). Précis of Elements of episodic memory. *Behavioral and Brain Sciences*, *7*(2), 223–238. https://doi.org/10.1017/s0140525x0004440x

Tulving, E., & Thomson, D. M. (1973). Encoding specificity and retrieval processes in episodic memory. *Psychological Review*, *80*(5), 352–373. https://doi.org/10.1037/h0020071

Wang, P., Tian, M., Li, J., Liang, Y., Wang, Y., Chen, Q., Wang, T., Lu, Z., Ma, J., Jiang, Y. E., & Zhou, W. (2025). O-MEM: omni memory system for personalized, long horizon, Self-Evolving agents. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2511.13593

Wang, Q., Fu, Y., Cao, Y., Wang, S., Tian, Z., & Ding, L. (2023). Recursively summarizing enables Long-Term dialogue memory in large language models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2308.15022

Wang, S., Yu, E., Love, O., Zhang, T., Wong, T., Scargall, S., & Fan, C. (2026). MemMachine: a Ground-Truth-Preserving memory system for personalized AI agents. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2604.04853

Weng, F., Angkititrakul, P., Shriberg, E. E., Heck, L., Peters, S., & Hansen, J. H. (2016). Conversational In-Vehicle Dialog Systems: The past, present, and future. *IEEE Signal Processing Magazine*, *33*(6), 49–60. https://doi.org/10.1109/msp.2016.2599201

Xu, D., Chen, W., Peng, W., Zhang, C., Xu, T., Zhao, X., Wu, X., Zheng, Y., Wang, Y., & Chen, E. (2024). Large language models for generative information extraction: a survey. *Frontiers of Computer Science*, *18*(6). https://doi.org/10.1007/s11704-024-40555-y

Xu, K., Yang, J., Xu, J., Gao, S., Guo, J., & Wen, J. (2021). Adapting User Preference to Online Feedback in Multi-round Conversational Recommendation. *Proceedings of the 14th ACM International Conference on Web Search and Data Mining*, 364–372. https://doi.org/10.1145/3437963.3441791

Xu, W., Liang, Z., Mei, K., Gao, H., Tan, J., & Zhang, Y. (2025, February 17). *A-MEM: Agentic Memory for LLM Agents*. arXiv.org. https://arxiv.org/abs/2502.12110

Yang, Z., Chen, S., Gao, C., Li, Z., Hu, X., Liu, K., & Xia, X. (2025). An Empirical Study of Retrieval-Augmented Code Generation: Challenges and Opportunities. *ACM Transactions on Software Engineering and Methodology*. https://doi.org/10.1145/3717061

Yu, Y., Yao, L., Xie, Y., Tan, Q., Feng, J., Li, Y., & Wu, L. (2026, January 5). *Agentic Memory: Learning Unified Long-Term and Short-Term Memory Management for Large Language Model Agents*. arXiv.org. https://arxiv.org/abs/2601.01885

Zhang, D., Feng, Y., Liu, H., Sun, C., Luo, J., Chen, X., & Li, X. (2025). Conversational Agents: From RAG to LTM. *Proceedings of the 2025 Annual International ACM SIGIR Conference on Research and Development in Information Retrieval in the Asia Pacific Region.*, 447–452. https://doi.org/10.1145/3767695.3769671

Zhang, J., Zhang, C., Chen, S., Huang, Z., Zheng, P., Wang, Z., Guo, P., Mo, F., Bae, S., Zou, J., Wei, J., & Yang, Y. (2026). Lightweight LLM Agent Memory with Small Language Models. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2604.07798

Zhang, N., Yang, X., Tan, Z., Deng, W., & Wang, W. (2026). HIMEM: Hierarchical Long-Term Memory for LLM Long-Horizon Agents. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2601.06377

Zhang, Y., Chen, X., Ai, Q., Yang, L., & Croft, W. B. (2018). Towards Conversational Search and Recommendation. *Proceedings of the 27th ACM International Conference on Information and Knowledge Management*, 177–186. https://doi.org/10.1145/3269206.3271776

Zhang, Z., Dai, Q., Bo, X., Ma, C., Li, R., Chen, X., Zhu, J., Dong, Z., & Wen, J. (2025). A survey on the memory mechanism of large language model-based agents. *ACM Transactions on Information Systems*, *43*(6), 1–47. https://doi.org/10.1145/3748302

Zhou, S. (2025). A simple yet strong baseline for Long-Term conversational memory of LLM agents. *arXiv (Cornell University)*. https://doi.org/10.48550/arxiv.2511.17208