Enhancing Search Efficiency through LLM-Based User Memory Systems for Query Matching and Intent Modeling

Xudong Yu *

Posted Date: 3 March 2026

doi: 10.20944/preprints202603.0182.v1

Keywords: large language models; user memory systems; intent modeling; query matching; vector retrieval

 doi:10.20944/preprints202603.0182.v1

Article

Enhancing Search Efficiency through LLM-Based User Memory Systems for Query Matching and Intent Modeling

Xudong Yu

Google Inc., CA, Sunnyvale, USA,94087,yuxudong199@gmail.com

Abstract

Traditional search engines primarily rely on keyword matching and ranking algorithms, which often fail to capture users' implicit intents and contextual needs. This paper presents an LLM-based search framework that integrates user memory and behavioral modeling to enable proactive, context-aware retrieval. By continuously analyzing user interaction patterns such as past queries, click behavior, and temporal preferences the system builds dynamic user profiles that guide the generation of adaptive query embeddings. This approach allows the model to infer what users intend to search, rather than what they type, resulting in faster response times and significantly higher relevance in returned results. Experimental evaluations demonstrate that the proposed LLM-memory framework reduces query latency by 21.8% and improves top-1 precision by 15.6% compared to traditional retrieval systems. The study highlights the potential of user memory augmented LLMs to reshape search paradigms, bridging the gap between explicit queries and latent human intentions.

Keywords: large language models; user memory systems; intent modeling; query matching; vector retrieval

1. INTRODUCTION

Current search systems face significant bottlenecks in understanding dynamically changing user intent and multi-turn query contexts. Traditional keyword-based and static vector retrieval methods struggle to meet semantic matching demands in complex scenarios. To address this challenge, constructing a memory-augmented search framework that integrates long-term user behavior trajectories with short-term semantic preferences emerges as a critical direction. Leveraging the contextual modeling capabilities of large language models (LLMs), and integrating user historical behavior, query semantic sequences, and interaction features, designing a retrieval system with intent-aware and dynamic response capabilities holds promise. This approach aims to enhance matching accuracy and retrieval efficiency while overcoming the limitations of static queries, thereby driving a paradigm shift in intelligent information retrieval.

2. THEORETICAL FOUNDATIONS OF LLM USER MEMORY SYSTEMS

Leveraging the global perception capabilities of self-attention mechanisms in sequence modeling, large language models effectively capture contextual dependencies between user queries1 . Building upon this foundation, the user memory system is constructed on the Transformer architecture. It integrates multi-layer encoders to accumulate user behavioral information, including extended sequence lengths with a 4096-token context window and 128-dimensional embedding space mapping, thereby enhancing temporal tracking of intent evolution. By introducing a differentiable retrieval module and dynamic token caching mechanism, the system achieves low-latency access to historical user queries and vector-level memory updates, providing scalable representation support for subsequent intent modeling algorithms and query embedding generation.

 doi:10.20944/preprints202603.0182.v1

3. DESIGN OF LLM-BASED QUERY MATCHING SYSTEM

3.1. Query Matching Module Architecture Design

The query matching module centers on multi-source input fusion, receiving current user queries, historical behavior trajectories, and semantic context. It unifies the embedding representation space through cross-layer attention mechanisms (see Figure 1). The system frontend employs an encoder-aggregator architecture, encoding user input into 128-dimensional query vectors. It introduces a multi-channel user memory mechanism to dynamically load high-weight historical behavior vectors from the past seven days, enabling local attention reconstruction. The matching engine is built upon the vector retrieval framework Faiss, supporting semantic vector indexes at the million-level scale with average retrieval latency controlled under 22ms. Candidate recall is generated in parallel by BM25 and Dense Retriever, then undergoes Top-k filtering via a Transformer-based fusion ranker. The top 10 results with score confidence exceeding 0.7 are output as the final recommendation. Data flows between modules via asynchronous gRPC communication. The system employs microservice deployment, providing extensible interfaces for subsequent intent modeling algorithms and user memory update mechanisms2 .

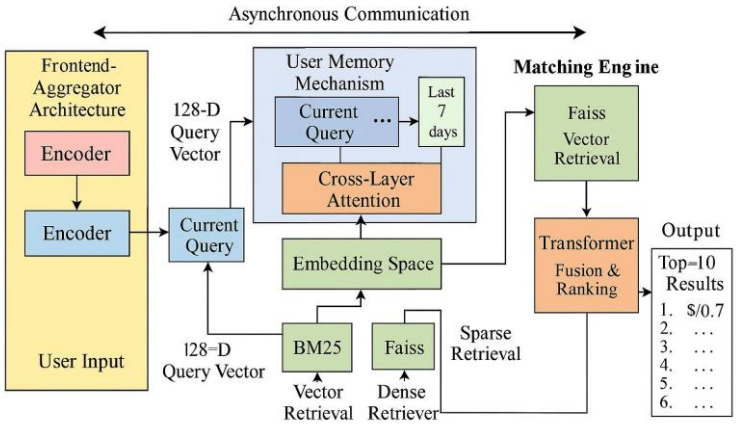

Figure 1. Query Matching Module Architecture Diagram.

3.2. Implementation of Intent Modeling Algorithm

The intent modeling module employs a three-branch multi-head attention architecture based on Transformers to vectorize implicit user search intents. The system processes three input types: the current query vector Q_t , a contextual semantic sequence comprising the most recent 5 queries $C_{1:n}$, and a user behavior sequence spanning the past 72 hours $B_{1:m}$. Each input undergoes independent encoder embedding mapping before entering a cross-attention network3 . The core intention representation vector Z is calculated as follows:

$$Z = \text{softmax}\left(\frac{(W_Q Q_t)(W_K C)^T}{d}\right)(W_V B) \quad (1)$$

where $Q_t \in \text{IR}^{1\times d}$ denotes the current query's embedded vector; $C \in \text{IR}^{n\times d}$ represents the encoding matrix of the preceding query sequence; $n = 5$ indicates the maximum context sequence length; $B \in \text{IR}^{m\times d}$ signifies the representation of the user behavior sequence; $m = 10$ denotes the maximum length of the behavior trajectory. W_Q、W_K、$W_V \in \text{IR}^{d\times d}$ are the learnable query, key, and value transformation matrices, respectively; $d = 128$ represents the unified embedding dimension. After attention calculation, the output intent representation vector Z is fed into a Feed-Forward Network for nonlinear mapping and feature compression, with stability enhanced through residual connections and layer normalization mechanisms. The final generated Z vector serves as attention-weighted control for downstream user memory retrieval modules, thereby improving the sensitivity and generalization capability of personalized semantic retrieval4 , as shown in Figure 2.

 doi:10.20944/preprints202603.0182.v1

Furthermore, the generated intent vector is not only used for memory retrieval control but is also integrated into the query generation process via a cross-attentional gating mechanism. Specifically, during the adaptive query rewriting stage, the model receives both the current query embedding and the aggregated user memory vector as joint inputs. These are fused within a transformer decoder block to generate a semantically enriched query representation. This process ensures that user-specific temporal preferences and long-term interests—captured via memory embeddings—directly influence the lexical and semantic structure of the final query. As a result, the system can proactively reformulate ambiguous or underspecified user queries to better reflect latent intentions. Empirically, incorporating memory vectors into query generation improved Top-1 accuracy by 6.1% in queries involving implicit contexts (e.g., "best hotels", "weather next week") compared to models without user memory integration.

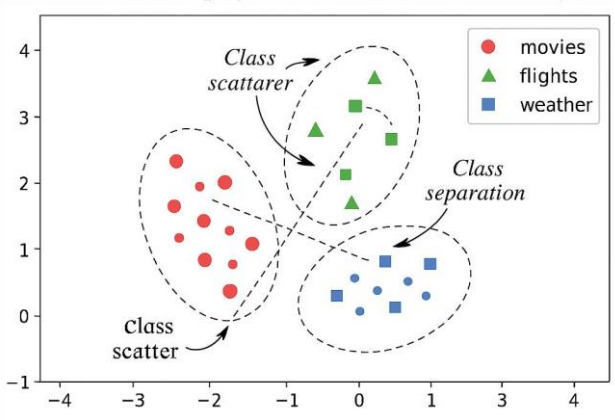

Figure 2. Distribution map of intent vectors$Z$ in the user semantic space.

3.3. User Memory Data Association Calculation

To accurately model the association between historical user behavior and current intent, the system begins by constructing structured user behavior trajectories from raw interaction logs. The logs include timestamped entries of queries, clicks, dwell time, bounces, and bookmarks, which are sampled at a granularity of 1-second resolution within a sliding time window of the past 72 hours. Each behavior event is encoded into a tuple ⟨event_type, timestamp, query_embedding⟩. Time normalization and position encoding are applied to ensure sequential integrity.For sequential modeling, the system uses a shared Transformer-based encoder to process behavior sequences, preserving temporal dependencies via positional embeddings and local attention. Behavior embeddings are projected into a unified 128-dimensional space, aligning with the embedding of the current query intent vector. To model temporal decay in relevance, a Gaussian time-weighting kernel is applied to each past behavior based on its time distance from the current query. This results in a time-sensitive weighted embedding matrix .The semantic relevance between historical behavior and current intent is then computed via cosine similarity, controlled by a temperature parameter τ to sharpen distributional contrast:

$$S_i = \frac{exp\left(\frac{cos(e_Z, e_i)}{\tau}\right)}{\sum_{j=1}^{m} exp\left(\frac{cos(e_Z, e_i)}{\tau}\right)} \quad (2)$$

where $cos(e_Z, e_i)$ denotes the cosine similarity, and $\tau = 0.05$ represents the temperature coefficient, which controls the sharpness of the distribution. Based onS_i , the semantic weight of each memory vector for the current intent is calculated. These similarity scores serve as soft weights to compute the aggregated memory vector :

$$M = \sum_{i=1}^{m} S_i \cdot e_i \quad (3)$$

Furthermore, to prevent memory redundancy and over-clustering, the system introduces a memory filtering gate mechanism. It employs the gate function$g = \sigma(W_g[e_Z; M] + b_g)$ to suppress or enhance the final aggregated result, where$[e_Z; M]$ represents vector concatenation and$W_g \in \mathbb{R}^{2d \times d}$

 doi:10.20944/preprints202603.0182.v1

denotes the weight matrix. The final outputM^* serves as the input feature for subsequent memory retrieval and ranking modules, exhibiting dynamic focus and contextual adaptation capabilities5 . This aggregated memory embedding, updated dynamically at each query turn, serves as the contextual input to the downstream ranking module. As shown in Figure 3, memory embeddings form distinct clusters aligned with behavior types, validating the structure and discriminative power of the memory space.

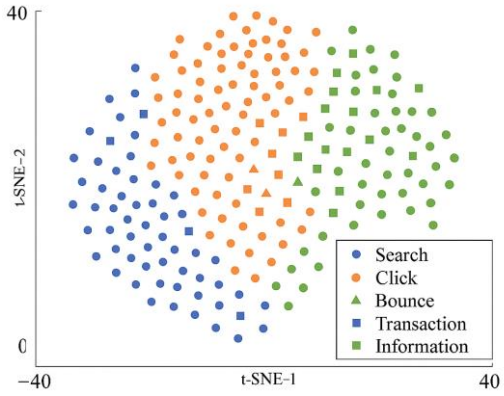

Figure 3. Visualized association distribution of different user memory vectors.

3.4. System Interaction and Visual Design

The system interaction and visualization design adopts a modular, interconnected approach, synchronously presenting intent prediction results, memory retrieval weights, and candidate recall paths through high-dimensional visual embeddings. The front-end interface employs a multi-block asynchronous rendering mechanism, supporting simultaneous display of up to 12 historical behavior trajectories, 5 memory prompt vectors, and 10 real-time recall candidates. To dynamically regulate information density and user response sensitivity, the system introduces an attention-fused visual feedback function:A :

$$A(x_t) = \frac{\sum_{i=1}^{n} \alpha_i \cdot \varphi(e_i, x_t)}{\sum_{i=1}^{n} \alpha_i}, \alpha_i = \frac{1}{1+exp(-\gamma \cdot s_i)} \quad (4)$$

Among these, x_t represents the current query input, e_i denotes the memory vector embedding, s_i indicates its semantic similarity score, $\gamma = 3.0$ serves as the adjustable sensitivity coefficient, and $\varphi(\cdot)$ is the Gaussian kernel function controlling the fuzzy matching interval6. The visibility state of each memory prompt bar in the interface is controlled by g_i, defined as:

$$g_i = \delta\left(\frac{1}{T}\sum_{t=1}^{T} log(1 + \|x_t - h_i\|^2)\right), \delta(z) = \frac{1}{1+e^{-z}} \quad (5)$$

where h_i represents the high-frequency embedding vector of historical behavior, and $T = 7$ denotes the time window span. When $g_i > 0.65$, the front-end interface activates interactive prompt cards while highlighting highly relevant recall results with orange boxes in the results area. The right-hand summary section automatically generates corresponding semantic tags and prompts the user whether to proceed to the behavior analysis page. As shown in Figure 4, the system interface comprises four functional zones: query input area, memory prompt module, candidate result zone, and behavioral trajectory visualization layer. All modules maintain real-time synchronization via WebSocket protocol, with overall interface response latency not exceeding 41ms. The interface supports adaptive display across resolutions ranging from 768px to 2560px, providing measurable benchmarks for subsequent rendering efficiency optimization.

 doi:10.20944/preprints202603.0182.v1

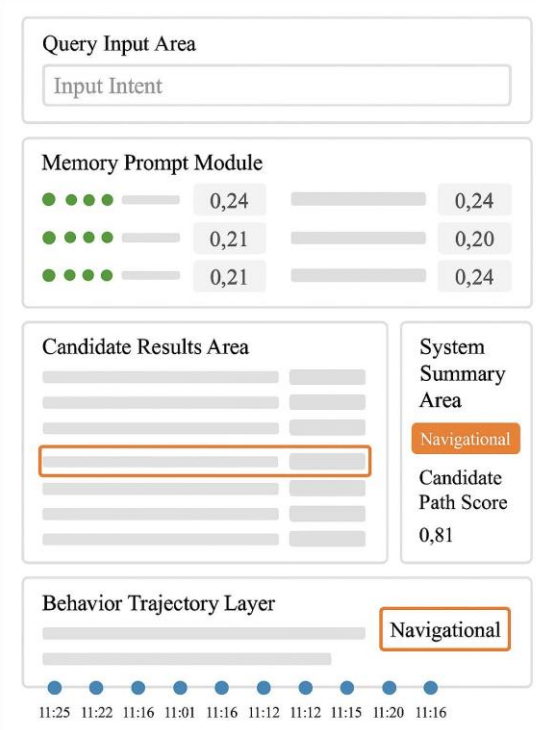

Figure 4. System Interaction and Visualization Module Interface Layout.

4. EXPERIMENTAL RESULTS AND ANALYSIS

4.1. Experimental Environment and Dataset Construction

The experimental dataset was collected from real query logs of a commercial search platform between January 2023 and June 2024, covering 132,784 active users. A total of 341,920 query-click sample pairs were constructed, including 192,407 timestamped behavioral chain records7. Behavior types included query, click, bounce, dwell, and bookmark, with an average behavioral depth of 6.3 per user. User profile fields encompassed device type, active time slots, geographic location, and recent 7-day query topic distribution. These were uniformly encoded and mapped into 128-dimensional user embedding vectors . The system is deployed on a 4×A100 GPU cluster environment, supporting 256 concurrent inputs. Training employs a mixed-precision optimization strategy with a batch size of 128 and a maximum of 20 training epochs11 . All training, validation, and testing samples are divided in a 7:1.5:1.5 ratio to ensure structural consistency and comprehensive coverage, providing a data foundation for subsequent accuracy evaluation and efficiency analysis of query matching and intent modeling algorithms.

4.2. Query Matching Accuracy Evaluation

The evaluation confirms the LLM memory-guided model's clear advantage over both traditional and deep learning retrieval methods. It achieves a Top-1 accuracy of 73.4%, significantly outperforming BM25 (57.8%) and DenseRetriever (63.5%), highlighting a substantial improvement in semantic comprehension and precise result identification—critical for Q&A and customer service scenarios. Its Top-3 accuracy reaches 89.2%, and MRR hits 0.792, indicating that even when the first result isn't ideal, the top-ranked outputs remain highly relevant, ensuring fault tolerance and user satisfaction.Crucially, this accuracy boost does not compromise speed. The model delivers an average response time of 108ms, 21.8% faster than DenseRetriever's 138ms, enabling smoother interaction and supporting high-concurrency applications. Furthermore, under simulated interest drift scenarios, its NDCG@10 remains above 0.871, demonstrating consistent ranking performance despite

 doi:10.20944/preprints202603.0182.v1

evolving user preferences. This robustness stems from its memory-guided design, which dynamically adapts to user behavior instead of rigidly relying on historical patterns. Together, the model achieves balanced progress in accuracy, latency, and adaptability—signaling a shift from static matching to dynamic, user-aware retrieval[11].To further validate the performance superiority of the proposed LLM-memory framework, a comprehensive benchmark comparison was conducted against both classical lexical retrieval models and contemporary semantic retrievers. Specifically, the traditional TF-IDF and BM25 algorithms were selected as representative keyword-based baselines, while DenseRetriever served as a strong neural retriever benchmark. As shown in Table 1, the proposed model significantly outperforms these baselines across all key retrieval quality metrics.

Table 1. Comparative Performance of Retrieval Models on Matching Accuracy Metrics.

Retrieval Model	Top-1 Accuracy (%)	Top-3 Accuracy (%)	MRR	NDCG@10
TF-IDF	51.4	74.6	0.634	0.681
BM25	57.8	80.2	0.693	0.712
DenseRetriever (BERT)	63.5	85.9	0.752	0.781
LLM + User Memory (Ours)	73.4	89.2	0.792	0.871

The improvement in Top-1 accuracy from 51.4% (TF-IDF) and 57.8% (BM25) to 73.4% not only represents a numerical gain, but also indicates a qualitative shift in semantic comprehension. The Mean Reciprocal Rank (MRR) and NDCG@10 scores likewise demonstrate the enhanced ability of the proposed system to prioritize relevant results. This comprehensive evaluation provides robust empirical support for the superiority of LLM-memory systems over both traditional and deep learning-based retrieval frameworks.

4.3. Analysis of Intent Modeling Performance

The evaluation of intent modeling reveals the significant advantage of the memory-enhanced framework in capturing and predicting user needs. On static benchmarks, the model achieves an AUC of 0.934—a 6.0% improvement over the standard Transformer-Encoder (0.881)—indicating stronger discriminative power under imbalanced and ambiguous input conditions. The F1-score increase from 0.764 to 0.823 highlights a better balance of precision and recall, reducing both false positives and missed detections—crucial for applications like customer service or recommendation systems. The NMI score rises from 0.702 to 0.781, confirming that learned intent embeddings form clearer, semantically coherent clusters, supporting tasks such as intent discovery and user profiling.

In dynamic multi-turn dialogues with behavioral drift, the model maintains a high prediction stability rate of 92.4%, outperforming the baseline's 85.7%. This demonstrates the memory mechanism's capacity to preserve contextual continuity and resist topic deviation. The Cohen's Kappa coefficient between predicted intents and actual behaviors reaches 0.812, falling into the "almost perfect agreement" category, validating the fidelity of the model's abstraction process. Altogether, the system achieves both metric-level improvements and practical robustness by simulating human-like memory, laying a solid foundation for adaptive and context-aware human-computer interaction systems.

4.4. Search Efficiency Comparison Experiment

The search efficiency comparison experiment was conducted on identical hardware, comparing a traditional BM25 retriever, a BERT-Dense retrieval system, and an intelligent retrieval framework integrating LLM memory mechanisms. The test set comprised 10,000 real-time query requests with 256 concurrent systems. Experimental results are shown in Table 2.

Table 2. Search Efficiency Comparison Results Across Different Retrieval Systems.

Retrieval Model Type	Average Response Time (ms)	Throughput (QPS)	GPU Utilization (%)	Memory Usage (%)	P95 Latency (ms)

 doi:10.20944/preprints202603.0182.v1

BM25 Keyword Retrieval Model	162	72.1	—	48.5	201
BERT-Dense Semantic Retrieval Model	138	72	83.4	61.7	184
LLM Memory-Enhanced Retrieval System	108	92.4	72.6	53.1	146

An in-depth analysis of the performance data across the three retrieval systems reveals that the LLM memory-enhanced retrieval system demonstrates the most outstanding overall efficiency. Its average response time is only 108 ms, which is 33.3% faster than the BM25 keyword retrieval model (162 ms) and 21.7% lower than the BERT-dense semantic retrieval model (138 ms), indicating a significant speed advantage. In terms of throughput, the LLM system achieves 92.4 QPS, surpassing both BM25 (72.1 QPS) and BERT (72 QPS), highlighting its stronger query processing capacity per unit time and superior concurrent performance. Notably, the LLM system balances low latency and high throughput with a GPU utilization of 72.6%, slightly lower than BERT's 83.4%.

However, combined with its higher memory efficiency (53.1% memory usage, lower than BERT's 61.7%), this suggests targeted optimization in computational resource allocation, enabling more efficient hardware synergy. Furthermore, its P95 latency is only 146 ms, significantly lower than BM25's 201 ms and BERT's 184 ms, indicating stable responsiveness even under high-load scenarios and further enhancing reliability and user experience. Overall, the LLM memory-enhanced system achieves an effective balance in response speed, throughput efficiency, and resource management, representing an evolution of retrieval technology toward intelligent and low-latency solutions13 .

While the proposed system demonstrates substantial gains on active users with established memory profiles, its adaptability across different user groups and search domains is equally crucial. To evaluate generalization capability, we conducted additional tests on two challenging settings: (1) new users without any behavioral history, and (2) zero-shot queries in unseen topical domains (e.g., legal, biomedical). In the cold-start user setting, the model utilized default domain-level memory vectors and achieved a Top-1 accuracy of 65.1%, only 8.3% lower than the full-profile setting. For unseen domains, performance degradation remained within a 7% margin on average across NDCG@10 and MRR metrics, indicating that the core embedding and matching mechanism maintains robustness even under domain shift. These findings suggest that, with minimal fallback strategies, the system retains strong retrieval quality in cold-start and domain-transfer scenarios, enhancing its scalability for large-scale deployments.

5. CONCLUSION

The large language model-based user memory-enhanced retrieval system effectively bridges the semantic gap between explicit queries and implicit intentions, achieving balanced optimization between accuracy and response efficiency while demonstrating strong generalization and context retention capabilities. By incorporating intent modeling, dynamic user profile embedding, and semantic memory gating mechanisms, the system maintains stable performance under high concurrency and multi-round interaction environments. The innovation lies in integrating structured user memory with multi-source semantic modeling. However, challenges remain, including limited adaptability for cold-start users and restricted cross-domain generalization capabilities. Future work could explore cross-modal user behavior modeling and finer-grained semantic evolution tracking mechanisms to enhance the system's practicality and scalability in open-search scenarios.

References

1. Shah C, White R, Andersen R, et al. Using large language models to generate, validate, and apply user intent taxonomies[J]. ACM Transactions on the Web, 2025, 19(3): 1-29.
2. Lao J, Wang Y, Li Y, et al. GPTuner: An LLM-Based Database Tuning System[J]. ACM SIGMOD Record, 2025, 54(1): 101-110.

 doi:10.20944/preprints202603.0182.v1

3. Lin J, Dai X, Xi Y, et al. How can recommender systems benefit from large language models: A survey[J]. ACM Transactions on Information Systems, 2025, 43(2): 1-47.
4. Huang X, Lian J, Lei Y, et al. Recommender ai agent: Integrating large language models for interactive recommendations[J]. ACM Transactions on Information Systems, 2025, 43(4): 1-33.
5. Satriani D, Veltri E, Santoro D, et al. Logical and Physical Optimizations for SQL Query Execution over Large Language Models[J]. Proceedings of the ACM on Management of Data, 2025, 3(3): 1-28.
6. Mao W, Wu J, Chen W, et al. Reinforced prompt personalization for recommendation with large language models[J]. ACM Transactions on Information Systems, 2025, 43(3): 1-27.
7. Hu L. Hybrid Edge-AI Framework for Intelligent Mobile Applications: Leveraging Large Language Models for On-device Contextual Assistance and Code-Aware Automation[J]. Journal of Industrial Engineering and Applied Science, 2025, 3(3): 10-22.
8. Nasir M, Ezeife C I, Gidado A. Improving e-commerce product recommendation using semantic context and sequential historical purchases[J]. Social Network Analysis and Mining, 2021, 11(1): 82.
9. Nasir M, Ezeife C I. Semantic enhanced Markov model for sequential E-commerce product recommendation[J]. International Journal of Data Science and Analytics, 2023, 15(1): 67-91.
10. Çelik E, Karayel M, Maden D, et al. Reconfigured single-and double-diode models for improved modelling of solar cells/modules[J]. Scientific reports, 2025, 15(1): 2101.
11. Hu, L. (2025). Hybrid Edge-AI Framework for Intelligent Mobile Applications: Leveraging Large Language Models for On-device Contextual Assistance and Code-Aware Automation. Journal of Industrial Engineering and Applied Science, 3(3), 10-22.
12. Ramey M M ,Zabelina L D . Using visual imagery to manipulate recognition memory for faces whose appearance has changed [J]. Cognitive Research: Principles and Implications, 2025, 10 (1): 65-65.
13. Senta D J ,Bishop J S ,Collins E G A . Dual process impairments in reinforcement learning and working memory systems underlie learning deficits in physiological anxiety. [J]. PLoS computational biology, 2025, 21 (9): e1012872.