LLM Agent Memory: A Survey from a Unified Representation–Management Perspective

Zhenheng Tang †, Xin He †, Tiancheng Zhao †, Fanjunduo Wei †, Xiang Liu, Peijie Dong, Qian Wang, Qi Li, Huacan Wang, Ronghao Chen, Sen Hu, Weidong Guo, Yu Xu, Haolan Chen, Kunfeng Lai, Kaiyong Zhao, Keyan Ding, Ivor W. Tsang, Yew-Soon Ong, Bo Li, Xiaowen Chu *

Posted Date: 23 March 2026

doi: 10.20944/preprints202603.0359.v2

Keywords: LLM agent; memory; survey

Article

LLM Agent Memory: A Survey from a Unified Representation–Management Perspective

Zhenheng Tang [1,†], Xin He [2,†], Tiancheng Zhao [2,3,†], Fanjunduo Wei [4,5,6,†], Xiang Liu [7], Peijie Dong [7], Qian Wang [8], Qi Li [7,9], Huacan Wang [10], Ronghao Chen [11], Sen Hu [11], Weidong Guo [12], Yu Xu [12], Haolan Chen [12], Kunfeng Lai [7,13], Kaiyong Zhao [14], Keyan Ding [5], Ivor W. Tsang [2], Yew-Soon Ong [2,15], Bo Li [1] and Xiaowen Chu [1,16,*]

[1] CSE Department, HKUST, Hong Kong
[2] CFAR, Agency for Science, Technology and Research (A*STAR), Singapore
[3] Georgia Institute of Technology, Singapore
[4] College of Computer Science and Technology, Zhejiang University, China
[5] ZJU-Hangzhou Global Scientific and Technological Innovation Center, Zhejiang University, China
[6] CSE Department, Northeastern University, China
[7] DSA Thrust, HKUST (Guangzhou), China
[8] National University of Singapore, Singapore
[9] Tsinghua University, China
[10] UCAS, China
[11] Peking University, China
[12] Platform and Content Group, Tencent, China
[13] gomore.ai, China
[14] XGRIDS, Hong Kong
[15] Nanyang Technological University, Singapore
[16] AI Thrust, HKUST (Guangzhou), Hong Kong
* Correspondence: xwchu@hkust-gz.edu.cn
† These authors contributed equally to this work.

Abstract

Large language models (LLMs) face significant challenges in sustaining long-term memory for agentic applications due to limited context windows. To address this limitation, many work has proposed diverse memory mechanisms to support long-term, multi-turn interactions, leveraging different approaches tailored to distinct memory storage objects, such as KV caches. In this survey, we present a unified taxonomy that organizes memory systems for long-context scenarios by decoupling memory abstractions from model-specific inference and training methods. We categorize LLM memory into three primary paradigms: natural language tokens, intermediate representations and parameters. For each paradigm, we organize existing methods by three management stages, including memory construction, update, and query, so that long-context memory mechanisms can be described in a consistent way across system designs, with their implementation choices and constraints made explicit. Finally, we outline key research directions for long-context memory system design.

Keywords: LLM agent; memory; survey

1. Introduction

Large Language Models (LLMs) are deployed in long-context and interactive settings, such as multi-turn dialogue, task-oriented assistants, and agent-based systems, where task completion requires information to be retained and reused across extended temporal spans rather than within a single prompt (Qian et al., 2023; Gao et al., 2023; Wang et al., 2023). As input sequences grow to thousands or even millions of tokens, performance often degrades on tasks that require entity tracking, logical consistency, or recall of task-relevant facts across long interaction histories (Gao et al., 2024; Zhang et al., 2024; Wu et al., 2025). These observations indicate that the core difficulty in long-context and multi-turn scenarios lies not only in *how much information the model can observe*, but in *how information*

 doi:10.20944/preprints202603.0359.v2

is selectively retained, retrieved, and integrated during inference (Shinwari and Usama, 2025; Maharana et al., 2024; Wan and Ma, 2025). This has motivated the introduction of *memory* as a key abstraction in LLM-based systems, enabling task-relevant information to persist beyond the immediate context window and to be reused for future reasoning and decision-making.

Many existing studies have explored diverse LLM memory mechanisms to support long-context reasoning and multi-turn interaction (Shinn et al., 2023; Zhong et al., 2023; Modarressi et al., 2023; Zhong et al., 2024; Qian et al., 2024). Broadly, memory enables task-relevant information to persist beyond the immediate context window and be reused across interactions, often drawing loose analogies to human cognition, where short-term working memory supports immediate reasoning and long-term memory enables recall over time (Baddeley, 2007; Budson and Kensinger, 2023). The analogy serves as a useful intuition that effective memory depends not only on capacity, but also on selective access and integration (Gao et al., 2024; Wu et al., 2025). Corresponding surveys have reviewed this literature from multiple perspectives, including LLM-based agents and long-term interaction (Zhang et al., 2025), long-context modeling and long-term memory (Wu et al., 2025; Huang et al., 2023; Jiang et al., 2024), personalization (Liu et al., 2025), and system-level efficiency such as inference-time memory management (Pan and Li, 2025; LI et al., 2025; Luohe et al.). These works underscore the central role of memory across application, modeling, and system dimensions.

In this survey, we take a *system-level, operation-centric* view of LLM memory. Instead of cataloging methods by tasks or modules, we organize the literature around a unified abstraction that connects **task knowledge requirements** to **memory representations** through shared **management interfaces**. Figure 1 sketches this logic from applications (top) to recurring representation choices (middle) and a common management interface over long interactions (bottom). Under this view, diverse methods can be compared by three questions: *what is stored, where it is stored, and how it is operated*. We structure the survey along two recurring dimensions:

- **Memory representation** describes *where and in what form* information resides: *token-level memory* in the input context, *intermediate latent memory* as inference-time states (e.g., Key–Value caches), and *parameter-level memory* in model weights via adaptation or editing.
- **Memory management** describes *how memory is operated over time* to satisfy task requirements under practical constraints. Across representations, we observe a shared interface of three core operations: *memory construction* (what to store and how to structure it), *memory update* (how to maintain, consolidate, or remove stored content), and *memory query* (how to select and integrate relevant information during inference).

Following this organization, Sections 2–4 review memory mechanisms by aligning each representation with the same construction–update–query interface and its key trade-offs. Section 5 synthesizes insights across memory types and outlines open challenges for reliable long-context and multi-turn LLM memory.

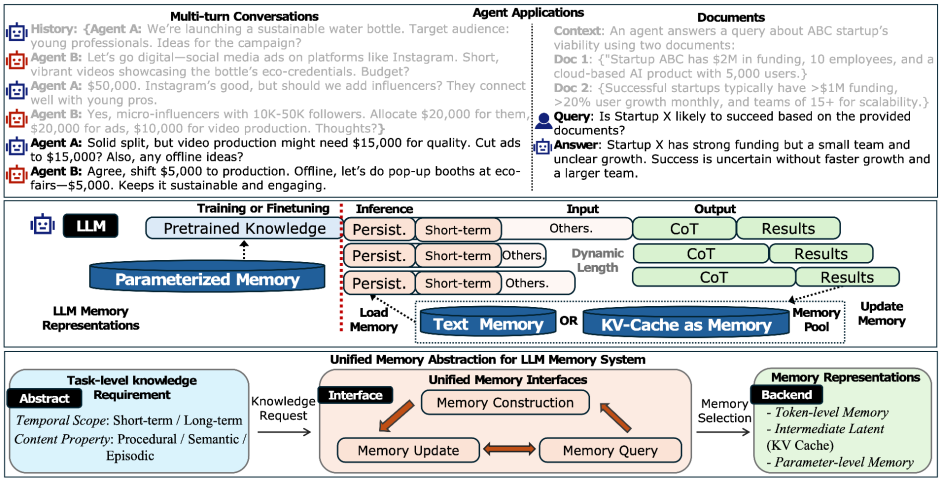


Figure 1. Overview of LLM memory applications, representation, and unified management abstraction.

 doi:10.20944/preprints202603.0359.v2

2. Natural Language Tokens as Memory

Natural language tokens are the most *explicit* memory format that leverages the context window for non-invasive information reuse. However, it is limited by $\mathcal{O}(n^2)$ computational costs and the tendency to overlook information buried in the middle of long sequences (Liu et al., 2024). To address these bottlenecks, **Retrieval-Augmented Generation** and **Agentic Memory** transform fragmented histories into structured, retrievable knowledge.

Goal: Keep the *right* evidence in context for reliable reasoning.
Challenge: Relevance under a context budget. Token-level memory should be treated as an *auditable working set*. Construction and update must compress and organize information so query returns a small, high-signal set rather than long history.

2.1. Retrieval-Augmented Generation (RAG)

RAG here refers to retrieval-based memory systems centered around an external a vector database, rather than full agent frameworks with explicit environment interaction. RAG offers a cost-effective way to extend long-context behavior by retrieving evidence and inserting it into the input.

Memory Construction.

Construction in RAG specifies *what evidence is stored* and *how it is indexed for retrieval*. For *what to store*, early RAG systems mainly ingest *unstructured* corpora and domain datasets (Li et al., 2023; Yan et al., 2024). More recent settings extend to *semi-structured* documents such as PDFs, where table-aware processing is often required (e.g., table-to-text normalization or Text-to-SQL-style querying (Zha et al., 2023; Luo et al., 2023)). RAG can also build on *structured* sources such as knowledge graphs, where KnowledGPT (Wang et al., 2023) and G-Retriever (He et al., 2024) improve graph grounding and evidence selection via soft prompting and PCST-style subgraph optimization. Some approaches leverage the *model's internal knowledge* to reduce retrieval overhead or bootstrap contexts, for example by selectively invoking retrieval (Wang et al., 2023), generating aligned contexts (Yu et al., 2022), or building unbounded memory pools (Cheng et al., 2023).

For *how to index*, most RAG systems first define the index unit, which determines the *retrieval granularity*: coarser units provide broader context but introduce redundancy, while finer units (e.g., tokens or phrases) improve precision at the risk of losing supporting context (Shi et al., 2023; Yu et al., 2023; Chen et al., 2023; Jin et al., 2023; Wang et al., 2023). A common approach splits documents into fixed-length spans (typically 100-512 tokens) and builds an index over these units (Teja, 2023). Variants such as recursive splitting, sliding windows, and Small2Big expansion preserve local coherence by attaching surrounding context (Langchain, 2023; Yang, 2023). Beyond unit definition, many systems enrich indexed entries with auxiliary fields (e.g., page numbers or generated cues) to improve matching (Gao et al., 2022). Indexes can further adopt *hierarchical* or *graph-based* structures to support multi-step evidence aggregation across documents (Wang et al., 2023). Finally, *embeddings* govern retrieval, encoding units as sparse or dense representations and ranking candidates via lexical matching (e.g., BM25) or embedding similarity, often using dense retrievers such as AngIE, Voyage, or BGE (Li and Li, 2023; VoyageAI, 2023; BAAI, 2023). Hybrid sparse-dense retrieval is commonly used to improve robustness, especially for zero-shot queries or rare entities (Zhang et al., 2025).

Memory Update.

RAG updates are primarily realized by modifying the external store and its index, including adding new documents, re-chunking and re-embedding content, refreshing metadata, and restructuring the index (e.g., moving from flat chunking to hierarchical or KG-based organization) to better match evolving domains and query patterns (Langchain, 2023; Yang, 2023; Gao et al., 2022; Wang et al., 2023). Some methods also adjust *retrieval invocation*, for example, deciding when retrieval is necessary

 doi:10.20944/preprints202603.0359.v2

or generating retrieval-aligned contexts to stabilize generation (Wang et al., 2023; Yu et al., 2022; Cheng et al., 2023).

Memory Query.

To improve robustness under underspecified or noisy inputs, existing methods enhance query quality from several complementary angles. One line focuses on *query reformulation*, including expansion (multi-query), decomposition (e.g., least-to-most prompting), validation (CoVe), and transformations (Gao et al., 2022; Zhou et al., 2023; Dhuliawala et al., 2023; Ma et al., 2023; Peng et al., 2023), where step-back prompting (Zheng et al., 2024) further abstracts the query to retrieve complementary evidence. Another line emphasizes *query routing*, selecting different retrieval pipelines based on metadata or semantic routers to enable hybrid strategies (Wang et al., 2025). A third line develops *multi-step and controllable querying*, where retrieval and generation are composed into modular pipelines (e.g., RRR, GenRead, RECITE (Yu et al., 2022; Ma et al., 2023; Sun et al., 2022)) and governed by dynamic controllers such as DSP, FLARE, and Self-RAG, sometimes combined with fine-tuning or reinforcement learning (Khattab et al., 2022; Jiang et al., 2023; Asai et al., 2023; Ke et al., 2024; Lin et al., 2023).

2.2. *Agentic Memory*

In contrast to RAG, which retrieves evidence from an external corpus, agentic memory targets *stateful, multi-turn interaction* and accumulates textual records of observations, actions, and outcomes across steps to support long-horizon reasoning.

Memory Construction.

Memory construction converts raw interaction traces into compact, retrievable textual units. A common baseline is to summarize dialogue histories, key events, and stable facts (e.g., user preferences or task states), as in MemoryBank and RET-LLM (Modarressi et al., 2023; Zhong et al., 2024). To enable efficient access, constructed memories are organized using explicit structures, including key-value slots (Modarressi et al., 2023; Salama et al., 2025; Xi et al., 2024), semantic vector representations (Zhong et al., 2024; Pan et al., 2025; mem0ai, 2024), and relation-aware graphs capturing dependencies among memory fragments, e.g., CGSN, GraphReader, HippoRAG (Nie et al., 2022; Li et al., 2024; Gutiérrez et al., 2024). Construction is strengthened by auxiliary signals such as timestamps, summaries, or factual tags to improve retrievability, e.g., LongMemEval (Wu et al., 2024), or by organizing memories along temporal and causal axes for context-sensitive navigation (Theanine (iunn Ong et al., 2025)). To control storage and inference cost, some systems apply token-level pruning, summarization, or soft compression at construction time, and reuse frequent contexts via prompt caching (Jiang et al., 2023; Chevalier et al., 2023; Liu et al., 2023; Gim et al., 2024).

Memory Update.

Memory update refines, consolidates, and revises stored content as interactions proceed. Early work controls growth through periodic summarization or restructuring, using explicit summarizers such as MemoryBank and ChatGPT-RSum (Zhong et al., 2024; Wang et al., 2025) or prompt-based extraction of salient topics as in MemoChat (Lu et al., 2023). Beyond compression, many methods treat updating as a *reasoning-driven* step: agents reflect on past actions and outcomes and write back reusable artifacts, including action-thought traces (Yao et al., 2022), self-critique and revision notes (Shinn et al., 2024), distilled reasoning templates (Yang et al., 2024), and workflow-level records (Wang et al., 2024). Experience-based agents refine memory through trial-and-error interaction and feedback, revising what to store and how to use it (Liu et al., 2023; Zhu et al., 2023; Wang et al., 2023; Yao et al., 2023; Zhao et al., 2024; Li et al., 2024). More recent systems emphasize *memory evolution*, allowing memories to be edited, linked, or reorganized over time. Examples include A-MEM's interconnected note-style growth (Xu et al., 2025; Kadavy, 2021), temporally adaptive structures such as Synapse, R2I, and SCM (Zheng et al., 2024; Samsami et al., 2024; Wang et al., 2024), as well as selective editing,

 doi:10.20944/preprints202603.0359.v2

recursive summarization, memory blending, and self-reflective verification to maintain relevance and consistency (Bae et al., 2022; Wang et al., 2025; Kim et al., 2024; Sun et al., 2024).

Memory Query.

Memory query determines how relevant entries are selected and integrated to support ongoing reasoning. Existing methods improve query effectiveness from complementary perspectives. *Query-centered* approaches reformulate or refine the query itself, for example via forward-looking rewriting or iterative refinement (Jiang et al., 2023; Jang et al., 2024). *Memory-centered* approaches enhance ranking and selection through richer indexing signals and reranking strategies, as explored in LongMemEval and personalized long-term memory retrieval (Wu et al., 2024; Du et al., 2024). Finally, *event- and structure-aware* retrieval leverages temporal, causal, or relational structure to traverse memory graphs or timelines, enabling coherent recall across long interaction histories (e.g., LoCoMo, CC, MSC, and graph-based multi-hop retrieval (Maharana et al., 2024; Qian et al., 2024; Gutiérrez et al., 2024; Jang et al., 2023; Xu et al., 2021)). Together, these strategies highlight that effective agentic memory query relies not only on semantic similarity, but also on adaptive, context-aware access over evolving memory states.

Takaways & Critical Insights

Token-level memory treats natural language itself as the memory interface, enabling transparent and easily editable storage. Several system-level insights emerge:

- **Selectivity dominates capacity.** The key challenge is not storing more text, but retrieving a small set of highly relevant evidence under strict context budgets.
- **Structure determines usability.** Chunking strategy, metadata, and relational organization directly shape retrieval effectiveness and downstream reasoning quality.
- **Memory as a working set.** Token memory should be actively curated through summarization, pruning, and restructuring so that the context window contains only high-signal evidence rather than raw history.

3. Intermediate Latent as Memory

Intermediate latent (IL) refer to inference-time internal representations in LLMs, such as attention activations or other continuous vectors, that can be cached and reused as memory. This section focuses on two forms of intermediate latent memory: the Key-Value (KV) cache, and other vector-based memory mechanisms.

Goal: Enable low-latency continuity within a session.
Challenge: State management under fixed capacity. IL memory is best viewed as *runtime state*. Construction, update (merge / compress / evict), and query must be explicitly scheduled to avoid churn, thrashing, and behavior drift.

3.1. KV Cache as Memory

Memory Construction.

The KV cache stores key–value vectors produced by the attention mechanism to accelerate autoregressive decoding. During prefilling, KV pairs for all prompt tokens are computed and cached; during decoding, only KV pairs for newly generated tokens are appended while attention reuses cached states. This reduces per-token complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ and is a core component of modern LLMs (Touvron et al., 2023a,2; Grattafiori et al., 2024; DeepSeek-AI et al., 2025a,2). Unlike token-based memory, KV cache construction involves no explicit selection of stored content: all tokens initially contribute full representations, leaving memory management to later stages. As a result, the KV cache serves as a transient, high-fidelity record of recent context tightly coupled to attention computation.

 doi:10.20944/preprints202603.0359.v2

Memory Update.

Memory update for KV cache focuses on controlling storage cost while preserving attention quality. Existing methods fall into several recurring strategies.

Eviction and dropping selectively discard KV entries using static patterns or dynamic importance signals. Representative approaches include fixed sparsity schemes (Xiao et al., 2024; Han et al., 2024), layer-wise retention (Wu and Tu, 2024), and attention- or query-aware dropping such as H_2O, FastGen, Radar, and NACL (Zhang et al., 2023; Ge et al., 2024; Hao et al., 2025; Chen et al., 2024). Variants further exploit attention statistics across heads, layers, or tasks (Liu et al., 2023; Devoto et al., 2024; Yao et al., 2024; Jiang et al., 2024; Zhong et al., 2024; Zhou et al., 2025).

Merging and semantic compression reduce redundancy by consolidating similar KV entries rather than removing them outright. This includes similarity-based merging (Liu et al., 2024; Kim et al., 2024; Agarwal et al., 2024) and semantic-preserving compression at token, chunk, or sentence granularity (Zhang et al., 2025; Liu et al., 2024,2; Zhu et al., 2025).

Quantization and low-rank approximation lower per-entry storage cost by reducing numerical precision or exploiting low-rank structure. Representative methods apply low-bit or asymmetric quantization (Liu et al., 2024; Duanmu et al., 2024; Dong et al., 2024; Hooper et al., 2024; Zhang et al., 2024; Li et al., 2025), attention- or layer-aware scaling (Lin et al., 2025; Yang et al., 2024), and low-rank compression with residual preservation (Dong et al., 2024; Saxena et al., 2024; Kang et al., 2024). Dynamic precision schemes further adapt quantization to runtime conditions (Sheng et al., 2023; Zhao et al., 2024; He et al., 2024).

System- and task-aware allocation adapts KV storage to deployment constraints and task characteristics. Examples include disaggregated and multi-GPU KV storage (Chen et al., 2024; Li et al., 2025), layer- or chunk-level budget assignment (Liu et al., 2025; Yang et al., 2024), and preference- or workload-aware allocation strategies (Zhu et al., 2025).

Memory Query.

Memory query determines how cached KV states are accessed during attention when attending to all entries is inefficient or unnecessary. 1) *KV selection* restricts attention to a subset of relevant entries using query-dependent signals: QUEST and TokenSelect estimate token importance via attention statistics or learned predictors (Tang et al., 2024; Wu et al., 2025), Selective Attention prunes targets across heads and layers (Leviathan et al., 2025), and RetrievalAttention treats KV states as retrievable items using approximate nearest neighbor search (Liu et al., 2024). 2) *KV reuse* avoids redundant computation by sharing cached states across overlapping contexts or requests. Prefix-based reuse organizes caches with tree structures, as in RadixAttention and ChunkAttention (Zheng et al., 2024; Ye et al., 2024), while cross-request reuse shares KV states based on semantic similarity (Yang et al., 2025; Agarwal et al., 2025; Tan et al., 2025). More recent systems extend reuse across retrieval and reranking stages in RAG pipelines (Yang et al., 2025; Yao et al., 2025; Hu et al., 2025; Zhu et al., 2025; An et al., 2025; Jiang et al., 2025), or externalize KV cache to support long-context inference beyond a single device (Wu et al., 2022; Tworkowski et al., 2023; Di et al., 2025).

3.2. Other Vectors as Memory

External Vectors.

External vector memory augments LLMs with a separate vector store that retains intermediate latents for retrieval and reuse, alleviating the quadratic cost of long-context attention (Al Adel and Burtsev, 2021). Early work explored sentence-level memory slots for sequence modeling, while later systems such as kNN-LM and the Memorizing Transformer leveraged pretrained embedding spaces and internal representations to enable scalable retrieval over large memory banks (Wu et al., 2022; Khandelwal et al., 2019). Subsequent designs, including MemGPT and Neurocache, maintain vector caches supporting dynamic retrieval and update for long-context or multi-session tasks (Packer et al., 2023; Safaya and Yuret, 2024). More recent architectures introduce structured or associative

 doi:10.20944/preprints202603.0359.v2

memory modules, such as CAMELoT, consolidating token representations while balancing novelty and recency (He et al., 2024), and MemOS or Memory3, which externalize knowledge via vector memory with metadata or sparsification, without modifying core model parameters (Li et al., 2025; Yang et al., 2024).

Steering Vectors.

Steering vectors function as intermediate memory for behavioral control rather than factual storage. Unlike interaction-heavy KV caches or external memories, they encode persistent biases as directions in activation space, originating from PPLM (Dathathri et al.). These vectors modulate hidden states to achieve alignment and interpretability via either contrastive or optimization-based approaches. *Contrastive methods* derive steering vectors from activation differences between datasets exhibiting desired versus undesired behaviors, encoding behavioral preferences as stable directions in activation space. Representative studies demonstrate control over sentiment, toxicity, refusal, or factuality using single or multiple contrastive prompts (Turner et al., 2023; Liu et al., 2023; Zou et al., 2023; Arditi et al., 2024). While effective, these approaches often require carefully constructed contrastive data and may capture spurious correlations that limit robustness and generalization (Chugtai and Bushnaq, 2025). *Optimization-based methods* instead learn steering vectors by optimizing simple objectives, such as maximizing target sequence likelihood or applying lightweight affine transformations to hidden states, sometimes with only a single example (Subramani et al., 2022; Hernandez et al., 2023; Dunefsky and Cohan, 2025; Mack and Turner, 2024). More recent work explores probe-based or low-shot steering to induce truthfulness, suppress refusal, or support personalization, though empirical effectiveness varies across models and tasks (Li et al., 2024; Turner et al., 2025; Cao et al., 2024).

Takaways & Critical Insights

Intermediate latent memory treats internal model states as runtime memory for maintaining short-horizon continuity during inference. Several system-level insights emerge:

- **Capacity management is the central challenge.** Since latent memory grows with sequence length, effective systems must actively control memory through eviction, compression, or quantization to remain scalable.
- **Runtime policies shape performance.** Cache scheduling, reuse, and allocation policies strongly influence latency and throughput, making memory management a systems-level optimization problem.
- **Session state rather than persistent knowledge.** Intermediate latent memory is best viewed as transient runtime state that preserves local reasoning context but is not suitable for durable knowledge storage.

4. Parameter as Memory

Goal: Persist knowledge across sessions and deployments via weight updates.
Challenge: Safe and localized writing.
Parameter-level memory represents *consolidated knowledge*. Updates must add new information while limiting interference with existing capabilities.

Memory Construction.

Parametric memory encodes knowledge directly into model weights through pretraining or fine-tuning, enabling long-term and context-independent storage. Its capacity and structure are largely shaped by training-time factors, including *training data composition*, *sequence length*, and *model scale*. Data augmentation strategies such as rephrasing or stylistic transformation can strengthen memorization (Allen-Zhu and Li, 2024), while data duplication produces superlinear memorization effects and raises privacy risks (Carlini et al., 2021; Lee et al., 2022; Kandpal et al., 2022). Longer training sequences expose models to richer dependencies, increasing the likelihood of verbatim recall (Carlini et al., 2023; Wang et al., 2024), and larger models further amplify parametric memory capacity (Carlini

 doi:10.20944/preprints202603.0359.v2

et al., 2023; Tirumala et al., 2022; Freeman et al., 2024). Mechanistic analyses suggest that such memory is not uniformly distributed: MLP layers can act as key-value stores (Geva et al., 2021), and factual knowledge may localize to specific neurons (Dai et al., 2021), providing a structural basis for later updates.

Memory Update.

Parameter-level memory update concerns modifying or extending knowledge embedded in model weights without reconstructing parametric memory from scratch. Unlike construction via large-scale training, update mechanisms incorporate new information or personalization while mitigating interference with existing parameters.

Continual learning addresses catastrophic forgetting by constraining or reinforcing parameter updates (Wang et al., 2024). Regularization-based methods protect parameters important to prior knowledge (e.g., EWC, TaSL, SELF-PARAM, POCL) (Kirkpatrick et al., 2017; Feng et al., 2024; Wang et al.; Wu et al., 2024), while replay-based approaches maintain memory by revisiting past or synthetic data, such as generative pseudo-query replay in DSI++ (Mehta et al., 2022). Extensions like LSCS further adapt continual learning to interactive agents that incrementally encode experiences into parameters (Wang et al., 2024).

Parameter-efficient fine-tuning (PEFT) updates parametric memory through lightweight adapters while keeping the backbone frozen (Han et al., 2024). This enables low-cost personalization and task adaptation, as seen in systems encoding character traits, personal histories, or episodic dialogue memory (e.g., Character-LLM, AI-Native Memory, MemoRAG, Echo) (Qian et al., 2024; Shao et al.; Shang et al., 2024; Liu et al., 2025).

Model merging combines multiple pretrained or fine-tuned models without original training data. Simple averaging (e.g., FedAvg) is efficient but prone to parameter conflicts (McMahan et al., 2017; Marczak et al., 2024). Recent work mitigates this through weighted merging based on parameter importance, subspace-based pruning or masking (e.g., TIES, DARE, Model Breadcrumbs), or routing mechanisms that dynamically select experts at inference time (Lee et al., 2019; Qu et al., 2022; Matena and Raffel, 2022; Yadav et al., 2023; Yu et al., 2024; Davari and Belilovsky, 2023; Shazeer et al., 2017; Muqeeth et al., 2024).

Task arithmetic interprets parameter updates as vector operations in weight space, enabling composition of task-specific knowledge through addition or subtraction (Ilharco et al., 2022). Methods such as TIES, AdaMerging, and TwinMerge explicitly resolve conflicts via parameter trimming or adaptive merging coefficients (Yadav et al., 2023; Yang et al., 2024; Lu et al., 2024).

Model editing performs targeted updates to parametric memory by inserting, modifying, or removing specific knowledge (Wang et al., 2024a). Systems such as MemoryLLM and WISE enable self-updating models and dual-memory routing between pretrained and edited knowledge, offering fine-grained control but relying on assumptions about knowledge localization (Wang et al., 2024b,2).

Memory Query.

Querying parametric memory differs fundamentally from querying token-level or KV-cache memory. Rather than retrieving stored entries, parametric memory is accessed implicitly through forward computation and analyzed via *memorization phenomena*. **Exact memorization** refers to verbatim reproduction of training sequences under suitable prompts (Carlini et al., 2021,2; Nasr et al., 2023), while **approximate memorization** captures semantic or structural similarity without exact copying (Ippolito et al., 2023). **Prompt-based memorization** further shows that carefully designed prompts can elicit stored content from partial prefixes, revealing the conditional nature of parametric recall (Biderman et al., 2023).

 doi:10.20944/preprints202603.0359.v2

Takaways & Critical Insights

Parametric memory stores knowledge directly in model weights, providing persistent memory that survives across sessions. This paradigm introduces a distinct set of design considerations:

- **Writing is expensive and risky.** Updating parametric memory requires careful mechanisms to incorporate new knowledge while avoiding interference with existing capabilities.
- **Persistence trades off with flexibility.** While weight-based memory offers strong durability, it is difficult to modify, inspect, or selectively retrieve compared to token-based memory.
- **Best suited for consolidated knowledge.** Parametric memory should encode stable, reusable knowledge, while frequently changing or context-specific information is better handled by external memory layers.

5. Discussion

From Knowledge Requests to Memory Backends.

Table 1 links task knowledge requirements to representation choice by highlighting the dominant management bottleneck of each memory type. We characterize requirements using two human memory-inspired axes (Tulving and Donaldson, 1972; Begg, 1984; Squire, 2009): *retention* (short-term vs. long-term) and *functional form* (episodic, semantic, procedural). Token-level memory supports short-term explicit content but is *query-limited* under long contexts, where selecting relevant evidence becomes the main challenge (Liu et al., 2024). Intermediate latent memory maintains short-horizon continuity but is *update-limited* due to fixed cache capacity and the need for efficient budgeting (Xiao et al., 2023). Parametric memory enables long-term consolidation but is *write-limited*, since updates are costly and must avoid interference or forgetting (Kirkpatrick et al., 2017; Meng et al., 2023).

Table 1. Decision-support matrix for LLM memory management. Knowledge requirements summarize the task regimes each representation most naturally supports. Management challenge indicates where engineering effort is typically concentrated under construction-update-query.

Memory Representation	Preferred Knowledge Requirements		Management Challenge			Strategic Gain (Return)
	Retention	Functional	Construction	Update	Query	
Token-level	Short-term	Episodic + Semantic	Medium	Medium	**High**	**Editability**
Intermediate latent	Short-term	Episodic	Low	**High**	Low	**Efficiency**
Parameter-level	Long-term	Procedural + Semantic	**High**	**High**	Low	**Persistence**

Unified Interfaces as System Glue.

These observations suggest a system-level principle: representation choice mainly shifts the bottleneck across construction, update, and query operations. Designing memory through a unified interface therefore enables systems to combine multiple backends while maintaining a consistent control plane for storing, maintaining, and retrieving information. Such decoupling improves portability across tasks and deployments and helps diagnose failures by tracing them to query selectivity (Liu et al., 2024), cache policies (Xiao et al., 2023; Zhang et al., 2023), or unsafe parameter updates (Meng et al., 2023).

Future Directions.

LLM memory design raises several open research directions, including specialized memory structures for heterogeneous workloads, unified training–inference systems for continual adaptation, and cross-domain methodology transfer from operating systems, databases, and distributed systems. These directions highlight the need for scalable, adaptive, and system-aware memory architectures for long-horizon agents. Detailed discussions are provided in Appendix C.

 doi:10.20944/preprints202603.0359.v2

Conclusion & Practical Design Guidelines

The three memory paradigms reflect complementary trade-offs across transparency, efficiency, and persistence, forming a layered design space where each type serves a distinct system role.

- **Token-level memory.** Natural-language memory provides the highest transparency, making it suited for grounding models on external knowledge, dynamic documents, or user-specific context. Its challenge lies in retrieval selectivity and context budget constraints as memory collections grow.
- **Intermediate memory.** Latent memory enables efficient session-level continuity by preserving internal representations during inference. It offers low-latency reasoning over recent context but remains transient and capacity-limited, making it unsuitable for durable or cross-session knowledge.
- **Parametric memory.** Weight-based memory provides the strongest persistence and global accessibility, enabling stable knowledge consolidation across tasks and deployments. However, writing to parametric memory is costly and potentially disruptive, requiring careful mechanisms to avoid interference or forgetting.

Practical guideline. System designers should select memory mechanisms based on knowledge volatility and usage scope: use token memory for *frequently changing or externally grounded information*, latent memory for *short-term reasoning state within a session*, and parametric memory for *stable knowledge that benefits from long-term consolidation*. In practice, robust LLM systems often combine these layers into a memory hierarchy that balances flexibility, efficiency, and persistence.

6. Conclusions

This survey presents a unified management view of LLM memory that links task knowledge requirements to memory representations through shared interfaces of construction, update, and query. By framing memory as a system-level capability rather than task-specific techniques, it enables systematic comparison of designs, analysis of effectiveness–efficiency trade-offs, and composition of hybrid memory backends for long-horizon agents. We hope this perspective helps standardize how future work specifies requirements, evaluates memory over extended interactions.

Limitations

This survey proposes a unified representation–management abstraction to organize LLM memory, but the fast pace of the field means our coverage may lag behind the newest systems and some industrial practices are discussed only at a high level. Our taxonomy is also a simplification: many methods span multiple representations and the boundaries between construction, update, and query can blur in long-horizon agents. Finally, quantitative comparisons across papers remain limited due to inconsistent tasks, models, evaluation protocols, and deployment settings, so our synthesis emphasizes recurring design trade-offs and failure modes rather than a unified benchmark ranking.

Appendix A. Related Surveys

The rapid development of LLMs has triggered many surveys on LLM-based agents, which study how an LLM can perceive, act, accumulate knowledge, and adapt over time. Early reviews (e.g., Wang et al. (2023)) organize agent research by how agents are built, where they are used, and how they are evaluated. Later surveys expand the scope with different taxonomies and emphases Xi et al. (2023); Zhao et al. (2023); Cheng et al. (2024); Ge et al. (2023). There are also focused reviews on key capabilities and settings, such as multimodal agents Durante et al. (2024), planning Huang et al. (2024), multi-agent interaction Guo et al. (2024), and personal assistant applications Li et al. (2024). These works provide useful summaries of agent pipelines, but memory is usually treated as one module among many, and is rarely analyzed as a first-class system component with a unified interface and lifecycle.

A separate line of surveys summarizes how LLMs are applied to specific domains. In information retrieval and extraction, surveys cover LLM-based query processing Zhu et al. (2023) and taxonomies for information extraction Xu et al. (2023). In recommender systems, several reviews discuss how LLMs and agent-style components are used for data generation and recommendation Li et al. (2023);

 doi:10.20944/preprints202603.0359.v2

Lin et al. (2023); Wang et al. (2023). In software engineering, surveys summarize the use of LLMs across design, development, and testing Fan et al. (2023); Wang et al. (2024); Zheng et al. (2023). Other domain surveys cover robotics Zeng et al. (2023), autonomous driving Cui et al. (2024); Yang et al. (2023), medicine He et al. (2023); Zhou et al. (2023); Wang et al. (2023), finance Li et al. (2023), and psychology He et al. (2023). While these surveys are valuable for understanding domain adaptation, they typically treat memory as domain-specific prompting or retrieval practice, rather than a general representation and management problem.

Surveys that target memory in LLMs and agent systems are more closely related to our work, but the current picture is still fragmented. Some surveys discuss operational aspects of memory Zhang et al. (2024), yet many narrow their scope to long-context modeling Huang et al. (2023), long-term memory Jiang et al. (2024); He et al. (2024), personalization Liu et al. (2025), or knowledge editing Wang et al. (2024a). This topical split makes it hard to compare methods across settings, and it often blurs the boundary between (i) what is stored (the memory representation) and (ii) how it is used and maintained (the memory management). As a result, practical foundations such as consistent benchmarks, tools, and implementation constraints are not discussed in a unified way.

Several recent surveys propose alternative lenses for understanding memory. Some move beyond a pure time-based split (short-term vs. long-term) and categorize memory by the memory "object", such as personal memories for user interaction and system memories for internal state Zhang et al. (2024); Zhong et al. (2024); Jiang et al. (2024). Others focus on memory mechanisms inside LLM-based agents and review their design, evaluation, and applications for self-evolving behaviors Zhang et al. (2025). Another direction decomposes memory into smaller operations and separates parametric and contextual forms, listing operations such as updating, indexing, retrieval, and compression Du et al. (2025). Human-memory-inspired surveys further relate human memory categories to AI memory designs and propose multi-dimensional categorizations Wu et al. (2025). Empirical studies evaluate how memory structures and retrieval strategies affect agent performance, including how memory addition and deletion influence long-horizon behaviors Zeng et al. (2024). These perspectives are informative, but they often treat the operation list as the primary organizing principle, which does not directly expose the system constraints behind different memory backends.

A complementary line of surveys studies memory from the deployment and efficiency angle. Inference system surveys summarize how to deliver high throughput and quality under large workloads Pan and Li (2025), and KV-cache management is reviewed as a key technique for reducing redundant computation and improving memory use during decoding LI et al. (2025); Luohe et al.; Hatalis et al. (2024). Broader inference optimization surveys also analyze sources of inefficiency and summarize techniques at the data, model, and system levels Zhou et al. (2024). These works are mainly organized around performance techniques. They provide less discussion on how runtime memory (e.g., KV cache) relates to other memory forms under a single representation-and-management view, and how different backends can be composed in one agent system. A related survey Shan et al. (2025) uses a taxonomy close to ours, but it does not provide a systematic discussion of implementation details across memory backends.

In contrast, our survey treats memory as a separable system component and organizes prior work by two orthogonal axes: the representation used to store memory and the management process that constructs, updates, and queries that memory. This decouples memory mechanisms from specific learning modes such as in-context learning or weight updates, and clarifies how the same management goals can be realized through different backends (token memory, intermediate latent memory, and parametric memory) under different cost and reliability constraints.

Appendix B. Overview of Human and LLM Memory and Taxonomy

This section provides a high-level overview of human and LLM memory through a concise taxonomy of both. By decoupling human memory from LLM memory, we analyze how different categories of human memory can be instantiated using distinct LLM memory mechanisms. Building

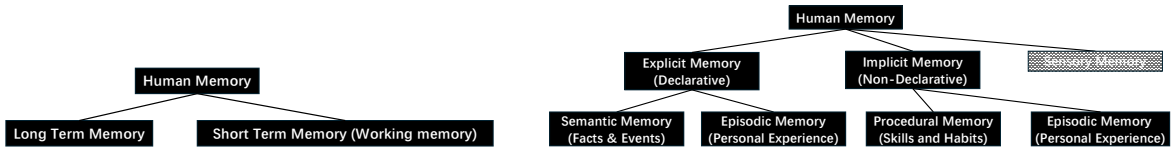
 doi:10.20944/preprints202603.0359.v2

on this perspective, we present a holistic framework that demystifies LLM memory design and offers a unified intuition for understanding approaches.

Appendix B.1. Human Memory

Human memory is a complex and multifaceted phenomenon, recognized in cognitive neuroscience as a collection of interconnected processes, including encoding, consolidation, storage, and retrieval Baddeley and Hitch (1974); Sridhar et al. (2023). It represents the brain's remarkable capacity to store, retain, and recall information, serving as the foundation for learning, adapting to environments, and shaping personal identity Sherwood et al. (2004); Weng (2023). Memory underpins higher-order cognitive functions such as reasoning, problem-solving, and language comprehension, profoundly influencing behavior and decision-making Budson and Kensinger (2023); Shan et al. (2025).

Based on the *duration of information retention*, human memory is classified into **short-term** and **long-term memory**, as illustrated in Figure A1(a) Baddeley (2007). Short-term memory temporarily holds and processes information for seconds to minutes as working memory, which actively manipulates information for immediate tasks like reasoning and comprehension Budson and Kensinger (2023); Baddeley and Hitch (1974). In contrast, long-term memory stores information for extended periods, ranging from minutes to years, forming a repository for enduring knowledge and experiences Budson and Kensinger (2023).



(a) Memory Categories according to duration. **(b)** Memory Categories according to semantics.

Figure A1. Human Memory Overview.

Based on functional roles, human memory is commonly categorized into **explicit** (declarative), **implicit** (non-declarative), and **sensory memory** [1], as shown in Figure A1(b) Budson and Kensinger (2023). Explicit memory involves conscious recall of facts and events that can be readily articulated. It includes episodic memory, which captures personal experiences tied to specific times and contexts (e.g., recalling what one ate for lunch) Tulving and Donaldson (1972), and semantic memory, which stores factual knowledge independent of personal experience (e.g., knowing the Earth is round) Begg (1984). Implicit memory operates unconsciously and is harder to verbalize, including procedural memory that governs skills and habits acquired through repetition, such as riding a bicycle or playing a musical instrument Squire (2009). These memory systems interact to support information processing and storage.

From a cognitive psychology perspective, memory is a fundamental mental process critical to learning and behavior Solso and Kagan (1979). It enables the accumulation of knowledge, abstraction of high-level concepts, and formation of social norms through the retention of cultural values and personal experiences Craik and Lockhart (1972); Leydesdorff (2017). Memory also supports decision-making by allowing individuals to anticipate potential consequences Johnson-Laird (1983). These insights are invaluable for designing LLM-based agents, as memory modules that mirror human cognitive processes enhance their ability to perform complex tasks and exhibit human-like behavior Laird (2019); Sun (2001).

Memory is essential for the self-evolution of LLM-based agents in dynamic environments Sutton and Barto (2018). It facilitates experience accumulation, enabling agents to retain past errors, inappropriate behaviors, or failed attempts to improve future performance and learning efficiency Zheng et al. (2023). Memory also supports environment exploration by guiding agents to prioritize less-explored

[1] For LLMs, we do not discuss sensory memory in detail, as LLMs primarily operate on text.

 doi:10.20944/preprints202603.0359.v2

actions or revisit previously unsuccessful trials, enhancing adaptability Montazeralghaem et al. (2020); Zhu et al. (2023). Additionally, memory enables knowledge abstraction, allowing agents to summarize raw observations into high-level insights, which is crucial for generalizing to new environments Zhao et al. (2023).

Table A1. Characteristics of Human Memory.

Memory Type	Key Function/Characteristics	Duration/Capacity
Sensory Memory	Brief buffer for incoming sensory information (visual, auditory, etc.)	Milliseconds to a few seconds
Working Memory (WM)	Transient active store for manipulating information; supports complex cognitive operations (reasoning, language)	Tens of seconds to minutes; limited items
Short-Term Memory (STM)	Temporary holding of information before transfer to LTM or forgetting	Tens of seconds to minutes; limited items
Long-Term Memory (LTM)	Stores information for extended periods; large capacity and durability	Minutes to decades; Vast capacity
Declarative (Explicit)	Consciously recalled facts and events	Minutes to decades; Vast capacity
Episodic Memory	Personal experiences, specific events with contextual details	Minutes to decades
Semantic Memory	General world knowledge, facts, concepts, language	Minutes to decades
Non-Declarative (Implicit)	Unconscious learning: skills, habits, priming, conditioning	Acquired slowly, long-lasting

Appendix B.2. LLM Memory

LLM Inference. LLMs such as GPT Brown et al. (2020), LLaMA Touvron et al. (2023a,2); Grattafiori et al. (2024), Qwen Qwen et al. (2025); Bai et al. (2023), and DeepSeek DeepSeek-AI et al. (2025a,2) operate under the autoregressive generation paradigm. In this approach, the model predicts the next token based on all previously generated tokens. Given an input sequence of tokens $(x_1, \ldots, x_n)$, the model computes a probability distribution over the vocabulary for the next token at each time step t, typically using the final token's representation. This process is governed by the joint probability:

$$\begin{aligned} P(x_1, \ldots, x_n) = P(x_1) \times P(x_2 \mid x_1) \\ \times \ldots \times P(x_n \mid x_1, \ldots, x_{n-1}). \end{aligned} \tag{A1}$$

The inference process (Equation (A1)) relies on the self-attention mechanism within the transformer architecture. For each token i, self-attention computes a weighted sum over the representations of all previous tokens $\{1, 2, \ldots, i\}$, resulting in a time complexity of $\mathcal{O}(n^2)$ for a sequence of length n. This quadratic complexity becomes computationally expensive for long sequences, as attention is recalculated over all preceding tokens at every step.

To mitigate this inefficiency, the KV cache is widely used as an optimization technique. The KV cache divides inference into two phases: **prefill** and **decoding**. In the prefill phase, the model processes the entire prompt with full-sequence attention, computing and storing the key and value vectors for all tokens. During the decoding phase, as the model generates one token at a time, only the key and value vectors for the new token are computed and appended to the cache. Attention is then calculated solely between the current query and the cached KV pairs, eliminating redundant computations and significantly improving the efficiency of autoregressive generation.

Transformer Architecture. The remarkable performance of transformer-based LLMs across diverse tasks is primarily driven by the self-attention mechanism Leviathan et al. (2025); Meng et al. (2025); Vaswani et al. (2017). In this mechanism, a sequence of input hidden states $(h_1, \ldots, h_n)$ is transformed through a linear projection layer to produce the Query ($\boldsymbol{Q}$), Key ($\boldsymbol{K}$), and Value ($\boldsymbol{V}$) vectors, as defined in Equation (A2).

 doi:10.20944/preprints202603.0359.v2

$$\text{concat}(q_i, k_i, v_i) = \text{concat}(W_q, W_k, W_v) \cdot h_i \tag{A2}$$

Subsequently, attention scores a_{ij} are computed by taking the dot product between a Query vector (q_i) and a Key vector (k_j), scaled by the square root of the dimension d. These scores are normalized and used as weights to sum the Value vectors (v_j), producing the output (o_i) as shown in:

$$a_{ij} = \frac{\exp(q_i^\top k_j/\sqrt{d})}{\sum_{t=1}^{i}\exp(q_i^\top k_t/\sqrt{d})}, \quad o_i = \sum_{j=1}^{i} a_{ij} v_j. \tag{A3}$$

The output of the self-attention mechanism is then processed by a Feed Forward Network (FFN), which typically consists of multiple linear layers interleaved with activation functions, further enhancing the model's ability to capture complex patterns.

In Context Learning. In-context learning enables LLMs to retrieve and utilize memory by incorporating relevant information directly within the input prompt as natural language, without modifying model parameters or storing intermediate representations explicitly. The model relies on the provided context—such as examples, instructions, or retrieved data from a pool or database—to guide its predictions. For a prompt with a sequence of tokens $(x_1, \ldots, x_n)$, the self-attention mechanism, as defined in Equation (A3), weighs the relevance of each token in the context when predicting the next token, effectively mimicking memory retrieval. Research has shown that in-context learning can be viewed as implicit structure induction or meta-optimization, where the model learns task-specific patterns during inference by performing a form of gradient descent within the attention mechanism Dai et al. (2023); Hahn and Goyal (2023); Garg et al. (2022).

In practice, in-context learning is implemented by constructing prompts with relevant examples or retrieved documents, as seen in models like GPT Brown et al. (2020) or Qwen Qwen et al. (2025); Bai et al. (2023). For instance, few-shot learning scenarios provide question-answer pairs to guide responses, with the self-attention mechanism computing scores $a_{ij} = \frac{\exp(q_i^\top k_j/\sqrt{d})}{\sum_{t=1}^{i}\exp(q_i^\top k_t/\sqrt{d})}$ to focus on relevant context tokens. Studies suggest that in-context learning excels in task recognition and adaptation, but its effectiveness depends on the quality and structure of the provided demonstrations Min et al. (2022); Pan et al. (2023). Additionally, emergent in-context learning capabilities may be transient and tied to pretraining task diversity, highlighting the role of training data in enabling this mechanism Singh et al. (2023); Raventos et al. (2023); Shen et al. (2024).

KV Cache or Intermediate Representations. The KV cache stores intermediate Key ($\boldsymbol{K}$) and Value ($\boldsymbol{V}$) vectors computed during the self-attention process, as defined in Equation (A2), to enhance efficiency in autoregressive generation. During the **prefill phase**, the model processes the entire prompt, generating and caching $\boldsymbol{K}$ and $\boldsymbol{V}$ vectors for all tokens. In the **decoding phase**, only the $\boldsymbol{K}$ and $\boldsymbol{V}$ vectors for each newly generated token are computed and appended to the cache, with attention calculated solely between the current Query and cached KV pairs, producing the output $o_i = \sum_{j=1}^{i} a_{ij} v_j$. This reduces the time complexity from $\mathcal{O}(n^2)$ to $\mathcal{O}(n)$ per token during decoding, as used in models like LLaMA Touvron et al. (2023a,2); Grattafiori et al. (2024) and DeepSeek DeepSeek-AI et al. (2025a,2).

Recent advancements have focused on optimizing the KV cache to handle long-context scenarios and reduce memory overhead. Techniques like CacheGen and ChunkKV employ semantic-preserving compression to reduce the cache size while maintaining performance Liu et al. (2024,2); Zhu et al. (2025); Liu et al. (2025). KIVI introduces asymmetric 2-bit quantization for KV cache entries, further reducing memory footprint Liu et al. (2024). Methods like SnapKV and H2O selectively retain important KV pairs based on attention patterns, improving efficiency for long-context inference Li et al. (2024); Zhang et al. (2023). Additionally, StreamingLLM uses attention sinks to stabilize attention distributions, enabling efficient streaming inference Xiao et al. (2023). Distributed approaches, such as KVDirect and FlowKV, optimize KV cache storage and transfer across multi-GPU systems Chen et al. (2024); Li et al. (2025). These optimizations make the KV cache a critical memory mechanism for scalable and efficient LLM inference.

 doi:10.20944/preprints202603.0359.v2

Training-Based Knowledge Integration. Training-based memorization embeds knowledge directly into the model's parameters during training, effectively transforming data into a compressed, implicit memory. LLMs like GPT, LLaMA, Qwen, and DeepSeek are trained on vast datasets, encoding patterns, facts, and relationships within weights, particularly in the linear projection layers (W_q, W_k, W_v) and the Feed Forward Network (FFN). The training process optimizes the joint probability $P(x_1, \ldots, x_n) = P(x_1) \cdot P(x_2 \mid x_1) \cdot \ldots \cdot P(x_n \mid x_1, \ldots, x_{n-1})$, adjusting parameters via backpropagation to capture statistical and semantic patterns. This enables the model to recall knowledge during inference without external storage, though the memory is static unless fine-tuned or retrained.

Summary. As shown in Figure A2, In-context learning provides flexible memory through natural language prompts, leveraging self-attention to adaptively retrieve task-specific information, with its efficacy tied to demonstration quality and pretraining diversity Min et al. (2022); Raventos et al. (2023). Examples of implementing memory with different kinds of natural language tokens are shown in Table A2. The KV cache optimizes inference by storing intermediate representations, with recent advancements like compression and selective retention enhancing efficiency for long contexts Liu et al. (2024); Xiao et al. (2023); Li et al. (2024). Training-based knowledge integration embeds static memory in model parameters, enabling generalization across tasks. Together, these mechanisms enable LLMs to balance flexibility, efficiency, and generalization in diverse applications.

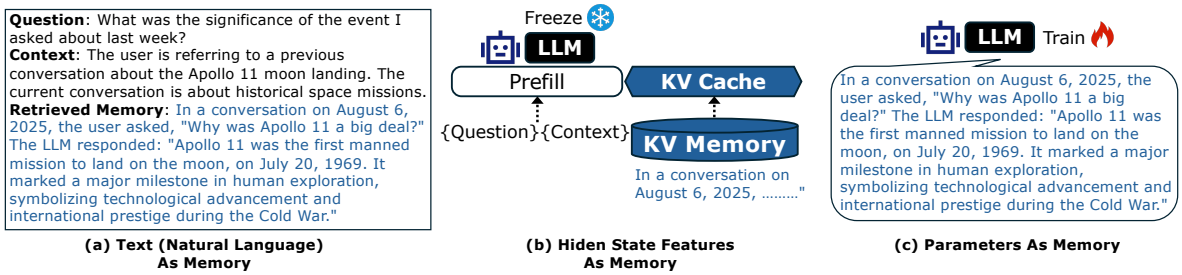


Figure A2. LLM Memory Classification.

Table A2. Examples of using different text memory (or can be KV cache).

Question: Can you remind me about the trip I mentioned planning to Paris?
Memory (Vector Database): Embedding match - Conversation: "User plans a trip to Paris in September 2025, interested in visiting the Louvre and Eiffel Tower." Vector Database: Enables semantic retrieval by matching the query's meaning to stored embeddings, useful for broad or vague queries.
Memory (Time Index): On July 15, 2025, at 14:30, the user said, "I'm planning a trip to Paris next month and want to see the Louvre." Time Index: Organizes memories chronologically, ideal for queries referencing recent or specific dates.
Memory (Username Index): User "JaneDoe123" discussed a Paris trip, mentioning a preference for art museums. Username Index: Ensures personalization by linking memories to a specific user, enhancing relevance.
Memory (Event Name Index): Event "Paris Trip 2025": User plans to visit Paris, focusing on cultural landmarks. Event Name Index: Tags memories with specific events, allowing precise retrieval for event-related queries.
Memory (Story Index): Story "Jane's European Adventure": Includes a chapter on planning a Paris trip, with details about booking a hotel near the Seine. Story Index: Structures memories as narratives, preserving context across related interactions.
Memory (Place Index): Place "Paris, France": User mentioned visiting the Eiffel Tower and dining at a café in Montmartre. Place Index: Associates memories with locations, enabling spatial queries about specific places.

Appendix B.3. Taxonomy of Memory Implementations

Different from existing surveys that connect different LLM memory types with human memory with one-to-one mapping, we observe that all different LLM memory mechanisms can be used to implement human-like memory with LLM. However, different LLM memory mechanisms have different characteristics, advantages and limitations, which are summarized in Table A3 and A4.

In-Context Learning. In-context learning supports both *short-term memory (processing immediate context)* and *long-term memory*. It relies on the self-attention mechanism to focus on relevant tokens, with no memory scale limitation for short-term use but high inference costs for long-term memory due to processing large contexts. In-context learning excels in episodic (context-specific events) and semantic

 doi:10.20944/preprints202603.0359.v2

(factual knowledge) memory, as prompts can encode task-specific examples or facts. Procedural memory (task processes) is limited due to reliance on explicit instructions.

Since in-context learning does not modify parameters, it avoids catastrophic forgetting, preserving pretrained knowledge. Short-term memory has no limitation, as it depends on the prompt size. Long-term memory faces high inference costs due to the $\mathcal{O}(n^2)$ complexity of self-attention for long sequences, and the lost-in-the-middle problem. The memory is explicit in the prompt, making it interpretable as the model's output directly reflects the provided context. No additional training or storage is required, though inference costs increase with context length. Malicious or biased prompts can influence outputs, raising concerns about misuse. High for episodic and semantic memory, as prompts can encode specific events or facts. Limited for procedural memory, as it requires explicit task instructions, which may not generalize well.

Table A3. Reconstructed Memory Implementation Characteristics.

Implementation Ways	Memory Type	Forgetting Pretrained Knowledge	Memory Scalability	Explainability	Serving Costs
In-context Learning	Short Term	No	High	High	Low
	Long Term	No	Weak	High	Low
Parameter by Training	Short Term	Weak	Hight	Low	Medium
	Long Term	Severe	Hight	Low	High

Table A4. Reconstructed Characteristics of Memory Types by Implementation.

Implementation Ways	Memory Type	Forgetting Pretrained Knowledge	Explainability	Costs	Knowledge Match Degree
In-context Learning	Procedural	No	High	Low	Low
	Episodic	No	High	Low	High
	Semantic	No	High	Low	High
Parameter by Training	Procedural	Possible	Low	High	High
	Episodic	Possible	Low	High	Moderate
	Semantic	Possible	Low	High	Moderate

KV Cache or Intermediate Representations. The KV cache primarily functions as short-term memory, storing $\boldsymbol{K}$ and $\boldsymbol{V}$ vectors during the prefill and decoding phases of inference. It retains contextual information for the current sequence, enabling efficient token generation. Limited applicability, as the cache is typically cleared between sessions.

The KV cache is a runtime mechanism that does not alter model parameters, preserving pretrained knowledge. Limited by memory constraints, as storing $\boldsymbol{K}$ and $\boldsymbol{V}$ vectors for long sequences can be memory-intensive. While the KV cache stores intermediate representations, interpreting their content is less straightforward than in-context learning, as it involves analyzing attention weights and vectors. The KV cache requires additional memory to store vectors, but optimizations like compression reduce this cost. Inference efficiency is improved compared to recomputing attention. Low. The KV cache is a technical optimization with minimal impact on output content, though improper cache management could affect output coherence. High for episodic memory, as the cache retains sequence-specific context. Limited for semantic or procedural memory, as it does not store generalized knowledge or task processes independently of the input sequence.

Parameter by Training. Training embeds both short-term (immediate patterns) and long-term (generalized knowledge) memory into parameters. Short-term memory has weak forgetting, as recent patterns are retained, but long-term memory suffers from severe forgetting due to catastrophic forgetting during fine-tuning. Training excels in procedural memory, as models learn task-specific patterns during pretraining. Episodic and semantic memory are moderately supported, as specific

 doi:10.20944/preprints202603.0359.v2

events or facts are compressed into parameters but may be less precise. Fine-tuning can overwrite pretrained knowledge, especially for long-term memory, leading to catastrophic forgetting. The model's parameters can encode vast amounts of knowledge, limited only by model size and training data. Knowledge embedded in parameters is opaque, making it difficult to trace specific outputs to learned patterns. Training requires significant computational resources, especially for large models and datasets. None. Knowledge is fixed in parameters, reducing risks from external inputs, though biases in training data can persist. High for procedural memory, as training optimizes task-specific patterns. Moderate for episodic and semantic memory, as specific events or facts are generalized but may lose granularity.

Appendix B.4. How Human Memory Benefits LLM Agentic Applications

Memory is an indispensable component in various practical LLM-based agent applications. For instance, in a conversational agent, memory stores information about historical conversations, providing the necessary context for generating coherent and relevant responses; without it, the agent cannot maintain a continuous conversation Lu et al. (2023). Similarly, in a simulation agent, memory is crucial for maintaining consistent role profiles, preventing the agent from deviating from its assigned character during a simulation Wang et al. (2025,2). These examples underscore that memory is not an optional feature but a necessary component for LLM-based agents to effectively accomplish their given tasks. Thus, the cognitive basis of human memory, coupled with its necessity for agent self-evolution and practical applications, provides critical insights for designing sophisticated memory mechanisms in LLM-based systems.

Information Retrieval and Processing. Long-context LLMs like Longformer and LongT5 enhance response relevance and document summarization by processing larger text segments, reducing reliance on external RAG tools Jin et al. (2024); Shi et al. (2024); Beltagy et al. (2020); Guo et al. (2022); Jin et al. (2024). Advanced semantic vector models, such as text-embedding-3-large, jina-embeddings-v2, and BGE-M3, overcome window size limitations, improving usability in tasks like translating complex documents and entire novels Zhu et al. (2023); Wang et al. (2024); OpenAI (2024); Günther et al. (2023); Chen et al. (2024); Zhu et al. (2024); Saad-Falcon et al. (2024); Herold and Ney (2023); Wang et al. (2024); Lyu et al. (2024).

Chatbots. Long-context processing enhances chatbots by enabling extended memory and contextual coherence, as seen in platforms like ChatGPT, Pi, Character AI, and Talkie, which use persistent memory and techniques like prompt-based memorization, memory-augmented architectures, and context extension for style-consistent, engaging dialogues OpenAI (2024); Inflection (2023); Character AI (2023); Ai (2024); Lee et al. (2023); Zhong et al. (2024); Wang et al. (2023,2).

Code Development. LLMs leverage memory to store development knowledge and conversational context, with models like StarCoder2, Qwen2.5-Coder, and Granite Code Models enabling scalable code completion and predictive debugging in tools like GitHub Copilot and Anysphere Cursor Qian et al. (2023); Tsai et al. (2023); Chen et al. (2023); Li et al. (2023); Zhang et al. (2024); Lozhkov et al. (2024); Hui et al. (2024); Mishra et al. (2024); GitHub (2022); Anysphere (2025).

Social Simulation. Memory defines character traits for realistic role-playing and supports multi-agent social simulations by improving self-monitoring, maintaining economic environments, and simulating dynamic behaviors Gao et al. (2023); Wang et al. (2025); Li et al. (2023); Shao et al. (2023); Kaiya et al. (2023); Li et al. (2023); Hua et al. (2023).

Personal Assistant. LLM-based personal assistants rely on memory for consistent, personalized dialogues, using textual retrieval, conversation summarization, and external tools to maintain conversational flow Lu et al. (2023); Lee et al. (2023); Pan et al. (2023); Wu et al. (2023).

Application in Specific Domains. Long-context LLMs improve coherence in news summaries, simplify legal document interpretation, enhance healthcare and financial decision-making, and advance drug discovery and scientific problem-solving by leveraging external knowledge and memory Gao et al. (2019); Kapoor et al. (2024); Fan et al. (2024); Reddy et al. (2024); Masry and Hajian (2024); Nie

 doi:10.20944/preprints202603.0359.v2

et al. (2024); Hilgert et al. (2024); Shao and Yan (2024); Wang et al. (2023); Xiong et al. (2023); Liu et al. (2023); Wang et al. (2023); Yunxiang et al. (2023); Chen et al. (2024); Zhao et al. (2024); Chen et al. (2023); Wang et al. (2023); Qiang et al. (2023).

Appendix C. Future Directions

Specialized Memory Structures. The growing scale and diversity of LLM memory workloads expose limitations of general-purpose memory hierarchies for long-context and agentic applications Yao et al. (2022). Memory mechanisms such as KV caches Pope et al. (2022), vector databases Lewis et al. (2020), and graph-structured memories Park et al. (2023) exhibit heterogeneous access patterns and update behaviors, motivating specialized memory structures for different abstractions. Meanwhile, the high cost of data movement highlights software–hardware co-design, where memory-aware algorithms and hardware support jointly reduce latency and energy Wulf and McKee (1995); Jouppi et al. (2017); Dao et al. (2022). Future work should explore dedicated memory layouts and tighter software–hardware integration to support memory construction, update, and query at scale Yu et al. (2022); Zhong et al. (2024).

Unified Training–Inference Systems. Current LLM systems largely separate training from inference, limiting adaptation to evolving users and environments Brown et al. (2020); Bommasani et al. (2022). While this survey focuses on inference memory, long-term agentic settings increasingly blur this boundary through continual learning and personalization during deployment Shinn et al. (2023); Park et al. (2023). This trend motivates unified training–inference designs that integrate memory management with lightweight updates and inference-time adaptation while mitigating catastrophic forgetting Hu et al. (2021). Future work should explore integrated architectures to enable continuous learning and personalized behavior.

Cross-Domain Methodology Transfer. LLM memory challenges often parallel classic problems in operating systems (OS) and databases. For example, KV cache management adopts OS-level paradigms like paging, eviction, and tiered hierarchies to handle limited memory Zhang et al. (2023); Kwon et al. (2023); Xiao et al. (2024). RAG systems leverage database techniques for indexing, query optimization, and execution planning Asai et al. (2023); Karpukhin et al. (2020); Khattab and Zaharia (2020); Izacard et al. (2023). Furthermore, distributed systems and cloud computing inspire solutions for scaling long-context workloads through partitioning and remote memory Zhong et al. (2024); Fu et al. (2024); Jin and Wu (2025). These established paradigms provide a foundational blueprint for scalable and efficient LLM memory orchestration.

Appendix D. Taxonomy of Different Memory

Table A5. Taxonomy of RAG.

Retrieval Augmented Generation Section 2.1		
Memory Construction	Unstructured Text Sources	CREA-ICL (Li et al., 2023), CRAG (Yan et al., 2024),
	Knowledge Graph	TableGPT (Zha et al., 2023), PKG (Luo et al., 2023), KnowledGPT (Wang et al., 2023), G-Retriever (He et al., 2024)
	Internal Knowledge	SKR (Wang et al., 2023), GenRead (Yu et al., 2022), Selfmem (Cheng et al., 2023)
Data Indexing	Granularity-based	(Shi et al., 2023), CoN (Yu et al., 2023), DenseX (Chen et al., 2023), LLMIndexer (Jin et al., 2023), LLM-R (Wang et al., 2023)
	Chunk-based	Chunk (Teja, 2023), LangChain (Langchain, 2023), Small2Big (Yang, 2023)
	Metadata-enriched	Reverse HyDE (Gao et al., 2022)
	Graph-based	KGP (Wang et al., 2023)
	Sparse Embedding	BM25
	Dense Embedding	AngIE (Li and Li, 2023), Voyage (VoyageAI, 2023), BGE (BAAI, 2023)
	Hybrid	LevelRAG (Zhang et al., 2025)
Memory Update	KG-based	KGP (Wang et al., 2023), Reverse HyDE (Gao et al., 2022), LangChain (Langchain, 2023), Small2Big (Yang, 2023)
	Retrieval Invocation	SKR (Wang et al., 2023), GenRead (Yu et al., 2022), Selfmem(Cheng et al., 2023)
Memory Query	Query Reformulation	(Zhou et al., 2023), CoVe (Dhuliawala et al., 2023), RRR (Ma et al., 2023), BEQUE (Peng et al., 2023), HyDE (Gao et al., 2022), Take a step back (Zheng et al., 2024)
	Query Routing	SemanticRouter (Wang et al., 2025)
	Multi-step Querying	DSP (Khattab et al., 2022), FLARE (Jiang et al., 2023), Self-RAG (Asai et al., 2023), BGM (Ke et al., 2024), RA-DIT (Lin et al., 2023)

 doi:10.20944/preprints202603.0359.v2

Table A6. Taxonomy of Agent Memory.

Agentic Memory Section 2.2		
Memory Construction	Summarization	MemoryBank (Zhong et al., 2024), RET-LLM (Modarressi et al., 2023)
	Key-value	RET-LLM (Murre and Dros, 2015), Meminsight (Salama et al., 2025), Memocrs (Xi et al., 2024)
	Semantic Representations	Memorybank (Zhong et al., 2024), (Pan et al., 2025), Mem0 (mem0ai, 2024)
	Relation Graph	CGSN (Nie et al., 2022), GraphReader (Li et al., 2024), HippoRAG (Gutiérrez et al., 2024)
	Auxiliary Signals	LongMemEval (Wu et al., 2024), THEANINE (iunn Ong et al., 2025)
	Storage and Inference Efficiency	Llmlingua (Jiang et al., 2023), AutoCompressor (Chevalier et al., 2023), TCRA-LLM (Liu et al., 2023), Promptcache (Gim et al., 2024)
Memory Updating	Summarization and Restructuring	MemoryBank (Zhong et al., 2024), ChatGPT-RSum (Wang et al., 2025), MemoChat (Lu et al., 2023)
	Reasoning and Self-Reflection	ReAct (Yao et al., 2022), Reflexion (Shinn et al., 2024), BoT (Yang et al., 2024), Agent Workflow Memory (Wang et al., 2024)
	Reasoning with Experience	TiM (Liu et al., 2023), GITM (Zhu et al., 2023), Voyager (Wang et al., 2023), Retroformer (Yao et al., 2023), ExpeL (Zhao et al., 2024), LD-Agent (Li et al., 2024)
	Memory Evolution	A-MEM (Xu et al., 2025), Synapse (Zheng et al., 2024), R2I (Samsami et al., 2024), SCM (Wang et al., 2024), Selective Editing (Bae et al., 2022), Blending (Kim et al., 2024), Recursive Summarization (Wang et al., 2025), Self-Reflection (Sun et al., 2024)
Memory Query	Query-centered	FLARE (Jiang et al., 2023), IterCQR (Jang et al., 2024)
	Memory-centered	LongMemEval (Wu et al., 2024), Perltqa (Du et al., 2024)
	Event and Structure-aware	LoCoMo (Maharana et al., 2024), CC (Jang et al., 2023), MSC (Xu et al., 2021), HippoRAG (Gutiérrez et al., 2024), Memorag (Qian et al., 2024)

Table A7. Taxonomy of KV Cache as Memory.

KV cache as Memory Section 3.1		
Memory Construction	KV Construction	Llama Series (Touvron et al., 2023a,2; Grattafiori et al., 2024), DeepSeek (DeepSeek-AI et al., 2025a,2)
Memory Update	Eviction and dropping	StreamingLLM (Xiao et al., 2024), LM-Infinite (Han et al., 2024), H_2O (Zhang et al., 2023), FastGen (Ge et al., 2024), Radar (Hao et al., 2025), NACL (Chen et al., 2024)
	Attention guided elimination	Scissorhands (Liu et al., 2023), L_2 Norm (Devoto et al., 2024), SirLLM (Yao et al., 2024), D-LLM (Jiang et al., 2024), ZigZagKV (Zhong et al., 2024), DynamicKV (Zhou et al., 2025)
	Merging and Semantic compression	MiniCache (Liu et al., 2024), InfiniPot (Kim et al., 2024), CHAI (Agarwal et al., 2024), Activation Beacon (Zhang et al., 2025), CacheGen (Liu et al., 2024), ChunkKV (Liu et al., 2025), SentenceKV (Zhu et al., 2025)
	Quantization and Low-rank Approximation	KIVI (Liu et al., 2024), SKVQ (Duanmu et al., 2024), QAQ (Dong et al., 2024), KVQuant (Hooper et al., 2024), CQ (Zhang et al., 2024), AnTKV (Li et al., 2025), SmoothAttention (Lin et al., 2025), MiKV (Yang et al., 2024), LESS (Dong et al., 2024), Eigen (Saxena et al., 2024), GEAR (Kang et al., 2024), FlexGen (Sheng et al., 2023), Atom (Zhao et al., 2024), ZipCache (He et al., 2024)
	System and Task-aware Allocation	KVDirect (Chen et al., 2024), FlowKV (Li et al., 2025), PyramidInfer (Yang et al., 2024), ChunkKV (Liu et al., 2025), OracleKV (Zhu et al., 2025)
Memory Query	KV selection	QUEST (Tang et al., 2024), TokenSelect (Wu et al., 2025), Selective Attention (Leviathan et al., 2025), RetrievalAttention (Liu et al., 2024)
	KV Reuse	RadixAttention (Zheng et al., 2024), ChunkAttention (Ye et al., 2024), KVShare (Yang et al., 2025), Cache-craft (Agarwal et al., 2025), Teola (Tan et al., 2025), KVLink (Yang et al., 2025), CacheBlend (Yao et al., 2025), EPIC (Hu et al., 2025), SubGCache (Zhu et al., 2025), HyperRAG (An et al., 2025), RAGO (Jiang et al., 2025)

 doi:10.20944/preprints202603.0359.v2

Table A8. Taxonomy of Other Vectors as Memory.

Other Vectors as Memory Section 3.2		
External Vector Memory	Sentence-level Encoding	Slot-based encoding (Al Adel and Burtsev, 2021)
	Key-value Stores	kNN-LM (Khandelwal et al., 2019), Memorizing Transformer (Wu et al., 2022)
	Vector cache	MemGPT (Packer et al., 2023), Neurocache (Packer et al., 2023)
	Structured and Associative Modules	CAMELoT (He et al., 2024), MemOS (Li et al., 2025), Memory3 (Yang et al., 2024)
Steering Vectors	Foundational method	PPLM (Dathathri et al.)
	Contrastive-based	Turner et al. (Turner et al., 2023), Liu et al. (Liu et al., 2023), Zou et al. (Zou et al., 2023), Arditi et al. (Arditi et al., 2024)
	Optimization-based	Subramani et al. (Subramani et al., 2022), Hernandez et al. (Hernandez et al., 2023), Dunefsky et al. (Dunefsky and Cohan, 2025), Mack et al. (Mack and Turner, 2024), Li et al. (Li et al., 2024), Turner et al. (Turner et al., 2025), Cao et al. (Cao et al., 2024)

Table A9. Taxonomy of Parameter as Memory.

Parameter as Memory Section 4		
Memory Construction	Data Composition	Data augmentation (Allen-Zhu and Li, 2024), Extracting (Carlini et al., 2021), Lee et al. (Lee et al., 2022), Kandpal et al. (Kandpal et al., 2022)
	Sequence length	Carlin et al. (Carlini et al., 2023), Wang et al. (Wang et al., 2024)
	Model Scale	Mem0 (Tirumala et al., 2022), Carlin et al. (Carlini et al., 2023), Freeman et al. (Freeman et al., 2024), Geva et al. (Geva et al., 2021), Dai et al. (Dai et al., 2021)
Memory Update	Continual Learning	SCM (Wang et al., 2024), EWC (Kirkpatrick et al., 2017), TaSL (Feng et al., 2024), SELF-PARAM (Wang et al.), POCL (Wu et al., 2024), DSI++ (Mehta et al., 2022), LSCS (Wang et al., 2024)
	PEFT	PEFT (Han et al., 2024), Character-LLM (Shao et al.), AI-Native Memory (Shang et al., 2024), MemoRAG (Qian et al., 2024), Echo (Liu et al., 2025)
	Model Merging	FedAvg (McMahan et al., 2017), MagMax (Marczak et al., 2024), SNIP (Lee et al., 2019), FisherMerging (Matena and Raffel, 2022), FedSAM (Qu et al., 2022), FedFisher (Jhunjhunwala et al., 2024), Gamegpt (Daheim et al., 2024), TIES (Yadav et al., 2023), DARE (Yu et al., 2024,2), Model Breadcrumbs (Davari and Belilovsky, 2023), TALL-masks (Wang et al., 2024), SMEAR (Muqeeth et al., 2024), Twin-Merging (Lu et al., 2024), Weight-Ensembling MoE (Tang et al., 2024)
	Task Arithmetic	Task Arithmetic (Ilharco et al., 2022), TIES (Yadav et al., 2023), AdaMerging (Yang et al., 2024), TwinMerge (Lu et al., 2024)
	Model Editing	Wang et al. (Wang et al., 2024a), MemoryLLM (Wang et al., 2024b), WISE (Wang et al., 2024c)
Memory Query	Exact Memorization	Carlini et al. (Carlini et al., 2021), Carlin et al. (Carlini et al., 2023), Nasr et al. (Nasr et al., 2023)
	Approximate Memorization	Ippolito et al. (Ippolito et al., 2023)
	Prompt-based Memorization	Biderman et al. (Biderman et al., 2023)

References

Qian, C.; Cong, X.; Yang, C.; Chen, W.; Su, Y.; Xu, J.; Liu, Z.; Sun, M. Communicative agents for software development. *arXiv preprint arXiv:2307.07924* **2023**.

Gao, C.; Lan, X.; Lu, Z.; Mao, J.; Piao, J.; Wang, H.; Jin, D.; Li, Y. S^3: Social-network simulation system with large language model-empowered agents. *arxiv* **2023**, [arXiv:cs.AI/arXiv:2307.14984].

Wang, L.; Ma, C.; Feng, X.; Zhang, Z.; Yang, H.; Zhang, J.; Chen, Z.; Tang, J.; Chen, X.; Lin, Y.; et al. A survey on large language model based autonomous agents. *arxiv* **2023**, [arXiv:cs.AI/arXiv:2308.11432].

 doi:10.20944/preprints202603.0359.v2

Gao, M.; Lu, T.; Yu, K.; Byerly, A.; Khashabi, D. Insights into LLM Long-Context Failures: When Transformers Know but Don't Tell. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024; Al-Onaizan, Y.; Bansal, M.; Chen, Y.N., Eds., Miami, Florida, USA, 2024.

Zhang, Z.; Bo, X.; Ma, C.; Li, R.; Chen, X.; Dai, Q.; Zhu, J.; Dong, Z.; Wen, J.R. A survey on the memory mechanism of large language model based agents. *arXiv preprint arXiv:2404.13501* **2024**.

Wu, Y.; Liang, S.; Zhang, C.; Wang, Y.; Zhang, Y.; Guo, H.; Tang, R.; Liu, Y. From Human Memory to AI Memory: A Survey on Memory Mechanisms in the Era of LLMs, 2025, [arXiv:cs.IR/2504.15965].

Shinwari, H.U.K.; Usama, M. Memory-Augmented Architecture for Long-Term Context Handling in Large Language Models. *arXiv preprint arXiv:2506.18271* **2025**.

Maharana, A.; Lee, D.H.; Tulyakov, S.; Bansal, M.; Barbieri, F.; Fang, Y. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753* **2024**.

Wan, L.; Ma, W. StoryBench: A Dynamic Benchmark for Evaluating Long-Term Memory with Multi Turns. *arXiv preprint arXiv:2506.13356* **2025**.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.R.; Yao, S. Reflexion: Language agents with verbal reinforcement learning. In Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, 2023.

Zhong, W.; Guo, L.; Gao, Q.; Wang, Y. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250* **2023**.

Modarressi, A.; Imani, A.; Fayyaz, M.; Schütze, H. Ret-llm: Towards a general read-write memory for large language models. *arxiv* **2023**, [arXiv:cs.AI/arXiv:2305.14322].

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; Wang, Y. Memorybank: Enhancing large language models with long-term memory. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 19724–19731.

Qian, H.; Zhang, P.; Liu, Z.; Mao, K.; Dou, Z. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591* **2024**.

Baddeley, A. *Working memory, thought, and action*; Vol. 45, OuP Oxford, 2007.

Budson, A.E.; Kensinger, E.A. *Why we forget and how to remember better: the science behind memory*; Oxford University Press, 2023.

Zhang, Z.; Dai, Q.; Bo, X.; Ma, C.; Li, R.; Chen, X.; Zhu, J.; Dong, Z.; Wen, J.R. A Survey on the Memory Mechanism of Large Language Model based Agents. *ACM Trans. Inf. Syst.* **2025**. Just Accepted, https://doi.org/10.1145/3748302.

Huang, Y.; Xu, J.; Lai, J.; Jiang, Z.; Chen, T.; Li, Z.; Yao, Y.; Ma, X.; Yang, L.; Chen, H.; et al. Advancing transformer architecture in long-context large language models: A comprehensive survey. *arXiv preprint arXiv:2311.12351* **2023**.

Jiang, X.; Li, F.; Zhao, H.; Wang, J.; Shao, J.; Xu, S.; Zhang, S.; Chen, W.; Tang, X.; Chen, Y.; et al. Long term memory: The foundation of ai self-evolution. *arXiv* **2024**, [2410.15665].

Liu, J.; Qiu, Z.; Li, Z.; Dai, Q.; Zhu, J.; Hu, M.; Yang, M.; King, I. A Survey of Personalized Large Language Models: Progress and Future Directions. *arXiv preprint arXiv:2502.11528* **2025**.

Pan, J.; Li, G. A Survey of LLM Inference Systems, 2025, [arXiv:cs.DB/2506.21901].

LI, H.; Li, Y.; Tian, A.; Tang, T.; Xu, Z.; Chen, X.; HU, N.; Dong, W.; Qing, L.; Chen, L. A Survey on Large Language Model Acceleration based on KV Cache Management. *Transactions on Machine Learning Research* **2025**.

Luohe, S.; Zhang, H.; Yao, Y.; Li, Z.; et al. Keep the Cost Down: A Review on Methods to Optimize LLM's KV-Cache Consumption. In Proceedings of the First Conference on Language Modeling.

Liu, N.F.; Lin, K.; Hewitt, J.; Paranjape, A.; Bevilacqua, M.; Petroni, F.; Liang, P. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics* **2024**, *12*, 157–173.

Li, X.; Nie, E.; Liang, S. From Classification to Generation: Insights into Crosslingual Retrieval Augmented ICL. *arXiv preprint arXiv:2311.06595* **2023**.

Yan, S.Q.; Gu, J.C.; Zhu, Y.; Ling, Z.H. Corrective Retrieval Augmented Generation. *arXiv preprint arXiv:2401.15884* **2024**.

Zha, L.; Zhou, J.; Li, L.; Wang, R.; Huang, Q.; Yang, S.; Yuan, J.; Su, C.; Li, X.; Su, A.; et al. Tablegpt: Towards unifying tables, nature language and commands into one gpt. *arXiv preprint arXiv:2307.08674* **2023**.

Luo, Z.; Xu, C.; Zhao, P.; Geng, X.; Tao, C.; Ma, J.; Lin, Q.; Jiang, D. Augmented Large Language Models with Parametric Knowledge Guiding. *arXiv preprint arXiv:2305.04757* **2023**.

Wang, X.; Yang, Q.; Qiu, Y.; Liang, J.; He, Q.; Gu, Z.; Xiao, Y.; Wang, W. KnowledGPT: Enhancing Large Language Models with Retrieval and Storage Access on Knowledge Bases. *arXiv preprint arXiv:2308.11761* **2023**.

 doi:10.20944/preprints202603.0359.v2

He, X.; Tian, Y.; Sun, Y.; Chawla, N.V.; Laurent, T.; LeCun, Y.; Bresson, X.; Hooi, B. G-Retriever: Retrieval-Augmented Generation for Textual Graph Understanding and Question Answering. *arXiv preprint arXiv:2402.07630* **2024**.

Wang, Y.; Li, P.; Sun, M.; Liu, Y. Self-Knowledge Guided Retrieval Augmentation for Large Language Models. *arXiv preprint arXiv:2310.05002* **2023**.

Yu, W.; Iter, D.; Wang, S.; Xu, Y.; Ju, M.; Sanyal, S.; Zhu, C.; Zeng, M.; Jiang, M. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063* **2022**.

Cheng, X.; Luo, D.; Chen, X.; Liu, L.; Zhao, D.; Yan, R. Lift Yourself Up: Retrieval-augmented Text Generation with Self Memory. *arXiv preprint arXiv:2305.02437* **2023**.

Shi, F.; Chen, X.; Misra, K.; Scales, N.; Dohan, D.; Chi, E.H.; Schärli, N.; Zhou, D. Large language models can be easily distracted by irrelevant context. In Proceedings of the International Conference on Machine Learning. PMLR, 2023, pp. 31210–31227.

Yu, W.; Zhang, H.; Pan, X.; Ma, K.; Wang, H.; Yu, D. Chain-of-Note: Enhancing Robustness in Retrieval-Augmented Language Models. *arXiv preprint arXiv:2311.09210* **2023**.

Chen, T.; Wang, H.; Chen, S.; Yu, W.; Ma, K.; Zhao, X.; Yu, D.; Zhang, H. Dense X Retrieval: What Retrieval Granularity Should We Use? *arXiv preprint arXiv:2312.06648* **2023**.

Jin, B.; Zeng, H.; Wang, G.; Chen, X.; Wei, T.; Li, R.; Wang, Z.; Li, Z.; Li, Y.; Lu, H.; et al. Language Models As Semantic Indexers. *arXiv preprint arXiv:2310.07815* **2023**.

Wang, L.; Yang, N.; Wei, F. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164* **2023**.

Teja, R. Evaluating the Ideal Chunk Size for a RAG System using LlamaIndex. https://www.llamaindex.ai/blog/evaluating-the-ideal-chunk-size-for-a-rag-system-using-llamaindex-6207e5d3fec5, 2023.

Langchain. Recursively split by character. https://python.langchain.com/docs/modules/data_connection/document_transformers/recursive_text_splitter, 2023.

Yang, S. Advanced RAG 01: Small-to-Big Retrieval. https://towardsdatascience.com/advanced-rag-01-small-to-big-retrieval-172181b396d4, 2023.

Gao, L.; Ma, X.; Lin, J.; Callan, J. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496* **2022**.

Wang, Y.; Lipka, N.; Rossi, R.A.; Siu, A.; Zhang, R.; Derr, T. Knowledge graph prompting for multi-document question answering. *arXiv preprint arXiv:2308.11730* **2023**.

Li, X.; Li, J. AnglE-optimized Text Embeddings. *arXiv preprint arXiv:2309.12871* **2023**.

VoyageAI. Voyage's embedding models. https://docs.voyageai.com/embeddings/, 2023.

BAAI. FlagEmbedding. https://github.com/FlagOpen/FlagEmbedding, 2023.

Zhang, Z.; Feng, Y.; Zhang, M. LevelRAG: Enhancing Retrieval-Augmented Generation with Multi-hop Logic Planning over Rewriting Augmented Searchers, 2025, [arXiv:cs.CL/2502.18139].

Zhou, D.; Schärli, N.; Hou, L.; Wei, J.; Scales, N.; Wang, X.; Schuurmans, D.; Cui, C.; Bousquet, O.; Le, Q.; et al. Least-to-Most Prompting Enables Complex Reasoning in Large Language Models, 2023, [arXiv:cs.AI/2205.10625].

Dhuliawala, S.; Komeili, M.; Xu, J.; Raileanu, R.; Li, X.; Celikyilmaz, A.; Weston, J. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495* **2023**.

Ma, X.; Gong, Y.; He, P.; Zhao, H.; Duan, N. Query Rewriting for Retrieval-Augmented Large Language Models. *arXiv preprint arXiv:2305.14283* **2023**.

Peng, W.; Li, G.; Jiang, Y.; Wang, Z.; Ou, D.; Zeng, X.; Chen, E.; et al. Large language model based long-tail query rewriting in taobao search. *arXiv preprint arXiv:2311.03758* **2023**.

Zheng, H.S.; Mishra, S.; Chen, X.; Cheng, H.T.; Chi, E.H.; Le, Q.V.; Zhou, D. Take a Step Back: Evoking Reasoning via Abstraction in Large Language Models, 2024, [arXiv:cs.LG/2310.06117].

Wang, C.; Liu, X.; Liu, Y.; Zhu, Y.; Mo, X.; Jiang, J.; Chen, H. When to Reason: Semantic Router for vLLM. *arXiv preprint arXiv:2510.08731* **2025**.

Sun, Z.; Wang, X.; Tay, Y.; Yang, Y.; Zhou, D. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296* **2022**.

Khattab, O.; Santhanam, K.; Li, X.L.; Hall, D.; Liang, P.; Potts, C.; Zaharia, M. Demonstrate-Search-Predict: Composing retrieval and language models for knowledge-intensive NLP. *arXiv preprint arXiv:2212.14024* **2022**.

Jiang, Z.; Xu, F.F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; Neubig, G. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983* **2023**.

 doi:10.20944/preprints202603.0359.v2

Asai, A.; Wu, Z.; Wang, Y.; Sil, A.; Hajishirzi, H. Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection. *arXiv preprint arXiv:2310.11511* **2023**.

Ke, Z.; Kong, W.; Li, C.; Zhang, M.; Mei, Q.; Bendersky, M. Bridging the Preference Gap between Retrievers and LLMs. *arXiv preprint arXiv:2401.06954* **2024**.

Lin, X.V.; Chen, X.; Chen, M.; Shi, W.; Lomeli, M.; James, R.; Rodriguez, P.; Kahn, J.; Szilvasy, G.; Lewis, M.; et al. RA-DIT: Retrieval-Augmented Dual Instruction Tuning. *arXiv preprint arXiv:2310.01352* **2023**.

Salama, R.; Cai, J.; Yuan, M.; Currey, A.; Sunkara, M.; Zhang, Y.; Benajiba, Y. Meminsight: Autonomous memory augmentation for llm agents. *arXiv preprint arXiv:2503.21760* **2025**.

Xi, Y.; Liu, W.; Lin, J.; Chen, B.; Tang, R.; Zhang, W.; Yu, Y. Memocrs: Memory-enhanced sequential conversational recommender systems with large language models. In Proceedings of the Proceedings of the 33rd ACM International Conference on Information and Knowledge Management, 2024, pp. 2585–2595.

Pan, Z.; Wu, Q.; Jiang, H.; Luo, X.; Cheng, H.; Li, D.; Yang, Y.; Lin, C.Y.; Zhao, H.V.; Qiu, L.; et al. On memory construction and retrieval for personalized conversational agents. *arXiv preprint arXiv:2502.05589* **2025**.

mem0ai. mem0: The memory layer for personalized ai. mem0.ai, 2024.

Nie, Y.; Huang, H.; Wei, W.; Mao, X.L. Capturing Global Structural Information in Long Document Question Answering with Compressive Graph Selector Network. In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing; Goldberg, Y.; Kozareva, Z.; Zhang, Y., Eds., Abu Dhabi, United Arab Emirates, 2022; pp. 5036–5047. https://doi.org/10.18653/v1/2022.emnlp-main.336.

Li, S.; He, Y.; Guo, H.; Bu, X.; Bai, G.; Liu, J.; Liu, J.; Qu, X.; Li, Y.; Ouyang, W.; et al. GraphReader: Building Graph-based Agent to Enhance Long-Context Abilities of Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024; Al-Onaizan, Y.; Bansal, M.; Chen, Y.N., Eds., Miami, Florida, USA, 2024; pp. 12758–12786. https://doi.org/10.18653/v1/2024.findings-emnlp.746.

Gutiérrez, B.J.; Shu, Y.; Gu, Y.; Yasunaga, M.; Su, Y. Hipporag: Neurobiologically inspired long-term memory for large language models. In Proceedings of the The Thirty-eighth Annual Conference on Neural Information Processing Systems, 2024.

Wu, D.; Wang, H.; Yu, W.; Zhang, Y.; Chang, K.W.; Yu, D. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813* **2024**.

iunn Ong, K.T.; Kim, N.; Gwak, M.; Chae, H.; Kwon, T.; Jo, Y.; won Hwang, S.; Lee, D.; Yeo, J. Towards Lifelong Dialogue Agents via Timeline-based Memory Management. In Proceedings of the Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Mexico City, Mexico, 2025.

Jiang, H.; Wu, Q.; Lin, C.Y.; Yang, Y.; Qiu, L. LLMLingua: Compressing Prompts for Accelerated Inference of Large Language Models. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing; Bouamor, H.; Pino, J.; Bali, K., Eds., Singapore, 2023; pp. 13358–13376. https://doi.org/10.18653/v1/2023.emnlp-main.825.

Chevalier, A.; Wettig, A.; Ajith, A.; Chen, D. Adapting Language Models to Compress Contexts. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing; Bouamor, H.; Pino, J.; Bali, K., Eds., Singapore, 2023; pp. 3829–3846. https://doi.org/10.18653/v1/2023.emnlp-main.232.

Liu, J.; Li, L.; Xiang, T.; Wang, B.; Qian, Y. TCRA-LLM: Token Compression Retrieval Augmented Large Language Model for Inference Cost Reduction. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2023; Bouamor, H.; Pino, J.; Bali, K., Eds., Singapore, 2023; pp. 9796–9810. https://doi.org/10.18653/v1/2023.findings-emnlp.655.

Gim, I.; Chen, G.; Lee, S.s.; Sarda, N.; Khandelwal, A.; Zhong, L. Prompt cache: Modular attention reuse for low-latency inference. *Proceedings of Machine Learning and Systems* **2024**, *6*, 325–338.

Wang, Q.; Fu, Y.; Cao, Y.; Wang, S.; Tian, Z.; Ding, L. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing* **2025**, *639*, 130193. https://doi.org/https://doi.org/10.1016/j.neucom.2025.130193.

Lu, J.; An, S.; Lin, M.; Pergola, G.; He, Y.; Yin, D.; Sun, X.; Wu, Y. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239* **2023**.

Yao, S.; Zhao, J.; Yu, D.; Du, N.; Shafran, I.; Narasimhan, K.; Cao, Y. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629* **2022**.

Shinn, N.; Cassano, F.; Gopinath, A.; Narasimhan, K.; Yao, S. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems* **2024**, *36*.

 doi:10.20944/preprints202603.0359.v2

Yang, L.; Yu, Z.; Zhang, T.; Cao, S.; Xu, M.; Zhang, W.; Gonzalez, J.E.; Cui, B. Buffer of thoughts: Thought-augmented reasoning with large language models. *arXiv preprint arXiv:2406.04271* **2024**.

Wang, Z.Z.; Mao, J.; Fried, D.; Neubig, G. Agent Workflow Memory, 2024, [arXiv:cs.CL/2409.07429].

Liu, L.; Yang, X.; Shen, Y.; Hu, B.; Zhang, Z.; Gu, J.; Zhang, G. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719* **2023**.

Zhu, X.; Chen, Y.; Tian, H.; Tao, C.; Su, W.; Yang, C.; Huang, G.; Li, B.; Lu, L.; Wang, X.; et al. Ghost in the minecraft: Generally capable agents for open-world environments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144* **2023**.

Wang, G.; Xie, Y.; Jiang, Y.; Mandlekar, A.; Xiao, C.; Zhu, Y.; Fan, L.; Anandkumar, A. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291* **2023**.

Yao, W.; Heinecke, S.; Niebles, J.C.; Liu, Z.; Feng, Y.; Xue, L.; Murthy, R.; Chen, Z.; Zhang, J.; Arpit, D.; et al. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151* **2023**.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.J.; Huang, G. Expel: Llm agents are experiential learners. In Proceedings of the Proceedings of the AAAI Conference on Artificial Intelligence, 2024, Vol. 38, pp. 19632–19642.

Li, H.; Yang, C.; Zhang, A.; Deng, Y.; Wang, X.; Chua, T.S. Hello again! llm-powered personalized agent for long-term dialogue. *arXiv preprint arXiv:2406.05925* **2024**.

Xu, W.; Liang, Z.; Mei, K.; Gao, H.; Tan, J.; Zhang, Y. A-mem: Agentic memory for llm agents. *arXiv preprint arXiv:2502.12110* **2025**.

Kadavy, D. *Digital Zettelkasten: Principles, Methods, and Examples*; Kadavy, Inc., 2021.

Zheng, L.; Wang, R.; Wang, X.; An, B. Synapse: Trajectory-as-Exemplar Prompting with Memory for Computer Control. In Proceedings of the Proceedings of the International Conference on Learning Representations (ICLR), 2024.

Samsami, M.R.; Zholus, A.; Rajendran, J.; Chandar, S. Mastering Memory Tasks with World Models. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.

Wang, B.; Liang, X.; Yang, J.; Huang, H.; Wu, S.; Wu, P.; Lu, L.; Ma, Z.; Li, Z. Enhancing Large Language Model with Self-Controlled Memory Framework, 2024, [arXiv:cs.CL/2304.13343].

Bae, S.; Kwak, D.; Kang, S.; Lee, M.Y.; Kim, S.; Jeong, Y.; Kim, H.; Lee, S.W.; Park, W.; Sung, N. Keep Me Updated! Memory Management in Long-term Conversations. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2022; Goldberg, Y.; Kozareva, Z.; Zhang, Y., Eds., Abu Dhabi, United Arab Emirates, 2022; pp. 3769–3787. https://doi.org/10.18653/v1/2022.findings-emnlp.276.

Wang, Q.; Fu, Y.; Cao, Y.; Wang, S.; Tian, Z.; Ding, L. Recursively Summarizing Enables Long-Term Dialogue Memory in Large Language Models. *Neurocomputing* **2025**, p. 130193. https://doi.org/10.1016/j.neucom.2025.130193.

Kim, S.H.; Ka, K.; Jo, Y.; Hwang, S.w.; Lee, D.; Yeo, J. Ever-Evolving Memory by Blending and Refining the Past. *arXiv preprint arXiv:2403.04787* **2024**.

Sun, H.; Cai, H.; Wang, B.; Hou, Y.; Wei, X.; Wang, S.; Zhang, Y.; Yin, D. Towards Verifiable Text Generation with Evolving Memory and Self-Reflection. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; Al-Onaizan, Y.; Bansal, M.; Chen, Y.N., Eds., Miami, Florida, USA, 2024; pp. 8211–8227.

Jiang, Z.; Xu, F.; Gao, L.; Sun, Z.; Liu, Q.; Dwivedi-Yu, J.; Yang, Y.; Callan, J.; Neubig, G. Active Retrieval Augmented Generation. In Proceedings of the Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing; Bouamor, H.; Pino, J.; Bali, K., Eds., Singapore, 2023; pp. 7969–7992. https://doi.org/10.18653/v1/2023.emnlp-main.495.

Jang, Y.; Lee, K.i.; Bae, H.; Lee, H.; Jung, K. IterCQR: Iterative Conversational Query Reformulation with Retrieval Guidance. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers); Duh, K.; Gomez, H.; Bethard, S., Eds., Mexico City, Mexico, 2024; pp. 8121–8138. https://doi.org/10.18653/v1/2024.naacl-long.449.

Du, Y.; Wang, H.; Zhao, Z.; Liang, B.; Wang, B.; Zhong, W.; Wang, Z.; Wong, K.F. PerLTQA: A Personal Long-Term Memory Dataset for Memory Classification, Retrieval, and Synthesis in Question Answering, 2024, [arXiv:cs.CL/2402.16288].

Jang, J.; Boo, M.; Kim, H. Conversation chronicles: Towards diverse temporal and relational dynamics in multi-session conversations. *arXiv preprint arXiv:2310.13420* **2023**.

 doi:10.20944/preprints202603.0359.v2

Xu, J.; Szlam, A.; Weston, J. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567* **2021**.

Touvron, H.; Lavril, T.; Izacard, G.; Martinet, X.; Lachaux, M.A.; Lacroix, T.; Rozière, B.; Goyal, N.; Hambro, E.; Azhar, F.; et al. LLaMA: Open and Efficient Foundation Language Models, 2023, [arXiv:cs.CL/2302.13971].

Touvron, H.; Martin, L.; Stone, K.; et al. Llama 2: Open Foundation and Fine-Tuned Chat Models, 2023, [arXiv:cs.CL/2307.09288].

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; et al. The Llama 3 Herd of Models, 2024, [arXiv:cs.AI/2407.21783].

DeepSeek-AI.; Guo, D.; Yang, D.; Zhang, H.; Song, J.; et al. DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning, 2025, [arXiv:cs.CL/2501.12948].

DeepSeek-AI.; Liu, A.; Feng, B.; Xue, B.; Wang, B.; Wu, B.; et al. DeepSeek-V3 Technical Report, 2025, [arXiv:cs.CL/2412.19437].

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; Lewis, M. Efficient Streaming Language Models with Attention Sinks. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.

Han, C.; Wang, Q.; Peng, H.; Xiong, W.; Chen, Y.; Ji, H.; Wang, S. LM-Infinite: Zero-Shot Extreme Length Generalization for Large Language Models. In Proceedings of the Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers); Duh, K.; Gomez, H.; Bethard, S., Eds., Mexico City, Mexico, 2024; pp. 3991–4008. https://doi.org/10.18653/v1/2024.naacl-long.222.

Wu, H.; Tu, K. Layer-Condensed KV Cache for Efficient Inference of Large Language Models. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Ku, L.W.; Martins, A.; Srikumar, V., Eds., Bangkok, Thailand, 2024; pp. 11175–11188. https://doi.org/10.18653/v1/2024.acl-long.602.

Zhang, Z.; Sheng, Y.; Zhou, T.; Chen, T.; Zheng, L.; Cai, R.; Song, Z.; Tian, Y.; Re, C.; Barrett, C.; et al. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, 2023.

Ge, S.; Zhang, Y.; Liu, L.; Zhang, M.; Han, J.; Gao, J. Model Tells You What to Discard: Adaptive KV Cache Compression for LLMs. In Proceedings of the The Twelfth International Conference on Learning Representations, 2024.

Hao, Y.; Zhai, M.; Hajimirsadeghi, H.; Hosseini, S.; Tung, F. Radar: Fast Long-Context Decoding for Any Transformer. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.

Chen, Y.; Wang, G.; Shang, J.; Cui, S.; Zhang, Z.; Liu, T.; Wang, S.; Sun, Y.; Yu, D.; Wu, H. NACL: A General and Effective KV Cache Eviction Framework for LLM at Inference Time. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Ku, L.W.; Martins, A.; Srikumar, V., Eds., Bangkok, Thailand, 2024; pp. 7913–7926. https://doi.org/10.18653/v1/2024.acl-long.428.

Liu, Z.; Desai, A.; Liao, F.; Wang, W.; Xie, V.; Xu, Z.; Kyrillidis, A.; Shrivastava, A. Scissorhands: Exploiting the Persistence of Importance Hypothesis for LLM KV Cache Compression at Test Time. In Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, 2023.

Devoto, A.; Zhao, Y.; Scardapane, S.; Minervini, P. A Simple and Effective L_2 Norm-Based Strategy for KV Cache Compression. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, Miami, Florida, USA, 2024; pp. 18476–18499. https://doi.org/10.18653/v1/2024.emnlp-main.1027.

Yao, Y.; Li, Z.; Zhao, H. SirLLM: Streaming Infinite Retentive LLM, 2024, [arXiv:cs.CL/2405.12528].

Jiang, Y.; Wang, H.; Xie, L.; Zhao, H.; Zhang, C.; Qian, H.; Lui, J.C. D-LLM: A Token Adaptive Computing Resource Allocation Strategy for Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems; Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; Zhang, C., Eds. Curran Associates, Inc., 2024, Vol. 37, pp. 1725–1749.

Zhong, M.; Liu, X.; Zhang, C.; Lei, Y.; Gao, Y.; Hu, Y.; Chen, K.; Zhang, M. ZigZagkv: Dynamic KV Cache Compression for Long-context Modeling based on Layer Uncertainty, 2024, [arXiv:cs.CL/2412.09036].

Zhou, X.; Wang, W.; Zeng, M.; Guo, J.; Liu, X.; Shen, L.; Zhang, M.; Ding, L. DynamicKV: Task-Aware Adaptive KV Cache Compression for Long Context LLMs, 2025, [arXiv:cs.CL/2412.14838].

Liu, A.; Liu, J.; Pan, Z.; He, Y.; Haffari, G.; Zhuang, B. MiniCache: KV Cache Compression in Depth Dimension for Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems;

 doi:10.20944/preprints202603.0359.v2

Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; Zhang, C., Eds. Curran Associates, Inc., 2024, Vol. 37, pp. 139997–140031.

Kim, M.; Shim, K.; Choi, J.; Chang, S. InfiniPot: Infinite Context Processing on Memory-Constrained LLMs. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; Al-Onaizan, Y.; Bansal, M.; Chen, Y.N., Eds., Miami, Florida, USA, 2024; pp. 16046–16060. https://doi.org/10.18653/v1/2024.emnlp-main.897.

Agarwal, S.; Acun, B.; Hosmer, B.; Elhoushi, M.; Lee, Y.; Venkataraman, S.; Papailiopoulos, D.; Wu, C.J. CHAI: Clustered Head Attention for Efficient LLM Inference. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning; Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; Berkenkamp, F., Eds. PMLR, 21–27 Jul 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 291–312.

Zhang, P.; Liu, Z.; Xiao, S.; Shao, N.; Ye, Q.; Dou, Z. Long Context Compression with Activation Beacon. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.

Liu, Y.; Li, H.; Cheng, Y.; Ray, S.; Huang, Y.; Zhang, Q.; Du, K.; Yao, J.; Lu, S.; Ananthanarayanan, G.; et al. CacheGen: KV Cache Compression and Streaming for Fast Large Language Model Serving, 2024, [arXiv:cs.NI/2310.07240].

Liu, X.; Tang, Z.; Dong, P.; Li, Z.; Liu, Y.; Li, B.; Hu, X.; Chu, X. ChunkKV: Semantic-Preserving KV Cache Compression for Efficient Long-Context LLM Inference, 2025, [arXiv:cs.CL/2502.00299].

Zhu, Y.; Falahati, A.; Yang, D.H.; Amiri, M.M. SentenceKV: Efficient LLM Inference via Sentence-Level Semantic KV Caching, 2025, [arXiv:cs.CL/2504.00970].

Liu, Z.; Yuan, J.; Jin, H.; Zhong, S.; Xu, Z.; Braverman, V.; Chen, B.; Hu, X. KIVI: A Tuning-Free Asymmetric 2bit Quantization for KV Cache. In Proceedings of the International Conference on Machine Learning, ICML 2024. PMLR, 2024, pp. 32332–32344.

Duanmu, H.; Yuan, Z.; Li, X.; Duan, J.; Zhang, X.; Lin, D. SKVQ: Sliding-window Key and Value Cache Quantization for Large Language Models, 2024, [arXiv:cs.LG/2405.06219].

Dong, S.; Cheng, W.; Qin, J.; Wang, W. QAQ: Quality Adaptive Quantization for LLM KV Cache, 2024, [arXiv:cs.CL/2403.04643].

Hooper, C.; Kim, S.; Mohammadzadeh, H.; Mahoney, M.W.; Shao, S.; Keutzer, K.; Gholami, A. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. *Advances in Neural Information Processing Systems, NeurIPS 2024* **2024**, *37*, 1270–1303.

Zhang, T.; Yi, J.; Xu, Z.; Shrivastava, A. KV cache is 1 bit per channel: Efficient large language model inference with coupled quantization. *Advances in Neural Information Processing Systems, NeurIPS 2024* **2024**, *37*, 3304–3331.

Li, Z.; Xiao, C.; Wang, Y.; Liu, X.; Tang, Z.; Lu, B.; Yang, M.; Chen, X.; Chu, X. AnTKV: Anchor Token-Aware Sub-Bit Vector Quantization for KV Cache in Large Language Models, 2025, [arXiv:cs.CL/2506.19505].

Lin, Y.; Tang, H.; Yang, S.; Zhang, Z.; Xiao, G.; Gan, C.; Han, S. QServe: W4A8KV4 Quantization and System Co-design for Efficient LLM Serving, 2025, [arXiv:cs.CL/2405.04532].

Yang, J.Y.; Kim, B.; Bae, J.; Kwon, B.; Park, G.; Yang, E.; Kwon, S.J.; Lee, D. No Token Left Behind: Reliable KV Cache Compression via Importance-Aware Mixed Precision Quantization, 2024, [arXiv:cs.LG/2402.18096].

Dong, H.; Yang, X.; Zhang, Z.; Wang, Z.; Chi, Y.; Chen, B. Get More with LESS: Synthesizing Recurrence with KV Cache Compression for Efficient LLM Inference. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning; Salakhutdinov, R.; Kolter, Z.; Heller, K.; Weller, A.; Oliver, N.; Scarlett, J.; Berkenkamp, F., Eds. PMLR, 21–27 Jul 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 11437–11452.

Saxena, U.; Saha, G.; Choudhary, S.; Roy, K. Eigen Attention: Attention in Low-Rank Space for KV Cache Compression. In Proceedings of the Findings of the Association for Computational Linguistics: EMNLP 2024; Al-Onaizan, Y.; Bansal, M.; Chen, Y.N., Eds., Miami, Florida, USA, 2024; pp. 15332–15344. https://doi.org/10.18653/v1/2024.findings-emnlp.899.

Kang, H.; Zhang, Q.; Kundu, S.; Jeong, G.; Liu, Z.; Krishna, T.; Zhao, T. GEAR: An Efficient KV Cache Compression Recipe for Near-Lossless Generative Inference of LLM, 2024, [arXiv:cs.LG/2403.05527].

Sheng, Y.; Zheng, L.; Yuan, B.; Li, Z.; Ryabinin, M.; Chen, B.; Liang, P.; Ré, C.; Stoica, I.; Zhang, C. FlexGen: high-throughput generative inference of large language models with a single GPU. In Proceedings of the Proceedings of the 40th International Conference on Machine Learning. JMLR.org, 2023, ICML'23.

Zhao, Y.; Lin, C.Y.; Zhu, K.; Ye, Z.; Chen, L.; Zheng, S.; Ceze, L.; Krishnamurthy, A.; Chen, T.; Kasikci, B. Atom: Low-Bit Quantization for Efficient and Accurate LLM Serving. In Proceedings of the MLSys, 2024.

 doi:10.20944/preprints202603.0359.v2

He, Y.; Zhang, L.; Wu, W.; Liu, J.; Zhou, H.; Zhuang, B. ZipCache: Accurate and Efficient KV Cache Quantization with Salient Token Identification. In Proceedings of the Advances in Neural Information Processing Systems; Globerson, A.; Mackey, L.; Belgrave, D.; Fan, A.; Paquet, U.; Tomczak, J.; Zhang, C., Eds. Curran Associates, Inc., 2024, Vol. 37, pp. 68287–68307.

Chen, S.; Jiang, R.; Yu, D.; Xu, J.; Chao, M.; Meng, F.; Jiang, C.; Xu, W.; Liu, H. KVDirect: Distributed Disaggregated LLM Inference, 2024, [arXiv:cs.DC/2501.14743].

Li, W.; Jiang, G.; Ding, X.; Tao, Z.; Hao, C.; Xu, C.; Zhang, Y.; Wang, H. FlowKV: A Disaggregated Inference Framework with Low-Latency KV Cache Transfer and Load-Aware Scheduling, 2025, [arXiv:cs.DC/2504.03775].

Yang, D.; Han, X.; Gao, Y.; Hu, Y.; Zhang, S.; Zhao, H. PyramidInfer: Pyramid KV Cache Compression for High-throughput LLM Inference. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2024, Bangkok, Thailand, 2024; pp. 3258–3270. https://doi.org/10.18653/v1/2024.findings-acl.195.

Zhu, Y.; Tang, Z.; Liu, X.; Li, A.; Li, B.; Chu, X.; Han, B. OracleKV: Oracle Guidance for Question-Independent KV Cache Compression. In Proceedings of the ICML 2025 Workshop on Long-Context Foundation Models, 2025.

Tang, J.; Zhao, Y.; Zhu, K.; Xiao, G.; Kasikci, B.; Han, S. QUEST: Query-Aware Sparsity for Efficient Long-Context LLM Inference. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning. PMLR, 21–27 Jul 2024, Vol. 235, *Proceedings of Machine Learning Research*, pp. 47901–47911.

Wu, W.; Pan, Z.; Wang, C.; Chen, L.; Bai, Y.; Wang, T.; Fu, K.; Wang, Z.; Xiong, H. TokenSelect: Efficient Long-Context Inference and Length Extrapolation for LLMs via Dynamic Token-Level KV Cache Selection, 2025, [arXiv:cs.CL/2411.02886].

Leviathan, Y.; Kalman, M.; Matias, Y. Selective Attention Improves Transformer. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.

Liu, D.; Chen, M.; Lu, B.; Jiang, H.; Han, Z.; Zhang, Q.; Chen, Q.; Zhang, C.; Ding, B.; Zhang, K.; et al. RetrievalAttention: Accelerating Long-Context LLM Inference via Vector Retrieval, 2024, [arXiv:cs.LG/2409.10516].

Zheng, L.; Yin, L.; Xie, Z.; Sun, C.; Huang, J.; Yu, C.H.; Cao, S.; Kozyrakis, C.; Stoica, I.; Gonzalez, J.E.; et al. SGLang: Efficient Execution of Structured Language Model Programs, 2024, [arXiv:cs.AI/2312.07104].

Ye, L.; Tao, Z.; Huang, Y.; Li, Y. ChunkAttention: Efficient Self-Attention with Prefix-Aware KV Cache and Two-Phase Partition, 2024, [arXiv:cs.LG/2402.15220].

Yang, H.; Zhang, R.; Huang, M.; Wang, W.; Tang, Y.; Li, Y.; Liu, Y.; Zhang, D. KVShare: An LLM Service System with Efficient and Effective Multi-Tenant KV Cache Reuse, 2025, [arXiv:cs.CL/2503.16525].

Agarwal, S.; Sundaresan, S.; Mitra, S.; Mahapatra, D.; Gupta, A.; Sharma, R.; Kapu, N.J.; Yu, T.; Saini, S. Cache-Craft: Managing Chunk-Caches for Efficient Retrieval-Augmented Generation, 2025, [arXiv:cs.DC/2502.15734].

Tan, X.; Jiang, Y.; Yang, Y.; Xu, H. Teola: Towards End-to-End Optimization of LLM-based Applications, 2025, [arXiv:cs.DC/2407.00326].

Yang, J.; Hou, B.; Wei, W.; Bao, Y.; Chang, S. KVLink: Accelerating Large Language Models via Efficient KV Cache Reuse, 2025, [arXiv:cs.CL/2502.16002].

Yao, J.; Li, H.; Liu, Y.; Ray, S.; Cheng, Y.; Zhang, Q.; Du, K.; Lu, S.; Jiang, J. CacheBlend: Fast Large Language Model Serving for RAG with Cached Knowledge Fusion, 2025, [arXiv:cs.LG/2405.16444].

Hu, J.; Huang, W.; Wang, W.; Wang, H.; Hu, T.; Zhang, Q.; Feng, H.; Chen, X.; Shan, Y.; Xie, T. EPIC: Efficient Position-Independent Caching for Serving Large Language Models, 2025, [arXiv:cs.LG/2410.15332].

Zhu, Q.; Zhang, L.; Xu, Q.; Long, C.; Zhang, J. SubGCache: Accelerating Graph-based RAG with Subgraph-level KV Cache, 2025, [arXiv:cs.LG/2505.10951].

An, Y.; Cheng, Y.; Park, S.J.; Jiang, J. HyperRAG: Enhancing Quality-Efficiency Tradeoffs in Retrieval-Augmented Generation with Reranker KV-Cache Reuse, 2025, [arXiv:cs.CL/2504.02921].

Jiang, W.; Subramanian, S.; Graves, C.; Alonso, G.; Yazdanbakhsh, A.; Dadu, V. RAGO: Systematic Performance Optimization for Retrieval-Augmented Generation Serving, 2025, [arXiv:cs.IR/2503.14649].

Wu, Y.; Rabe, M.N.; Hutchins, D.; Szegedy, C. Memorizing transformers. *arXiv preprint arXiv:2203.08913* **2022**.

Tworkowski, S.; Staniszewski, K.; Pacek, M.a.; Wu, Y.; Michalewski, H.; Mił oś, P. Focused Transformer: Contrastive Training for Context Scaling. In Proceedings of the Advances in Neural Information Processing Systems; Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; Levine, S., Eds. Curran Associates, Inc., 2023, Vol. 36, pp. 42661–42688.

Di, S.; Yu, Z.; Zhang, G.; Li, H.; TaoZhong.; Cheng, H.; Li, B.; He, W.; Shu, F.; Jiang, H. Streaming Video Question-Answering with In-context Video KV-Cache Retrieval. In Proceedings of the The Thirteenth International Conference on Learning Representations, 2025.

 doi:10.20944/preprints202603.0359.v2

Al Adel, A.; Burtsev, M.S. Memory transformer with hierarchical attention for long document processing. In Proceedings of the 2021 International Conference Engineering and Telecommunication, 2021, pp. 1–7. https://doi.org/10.1109/EnT50460.2021.9681776.

Khandelwal, U.; Levy, O.; Jurafsky, D.; Zettlemoyer, L.; Lewis, M. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172* **2019**.

Packer, C.; Wooders, S.; Lin, K.; Fang, V.; Patil, S.G.; Stoica, I.; Gonzalez, J.E. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560* **2023**.

Safaya, A.; Yuret, D. Neurocache: Efficient vector retrieval for long-range language modeling. *arXiv preprint arXiv:2407.02486* **2024**.

He, Z.; Karlinsky, L.; Kim, D.; McAuley, J.; Krotov, D.; Feris, R. Camelot: Towards large language models with training-free consolidated associative memory. *arXiv preprint arXiv:2402.13449* **2024**.

Li, Z.; Song, S.; Xi, C.; Wang, H.; Tang, C.; Niu, S.; Chen, D.; Yang, J.; Li, C.; Yu, Q.; et al. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724* **2025**.

Yang, H.; Lin, Z.; Wang, W.; Wu, H.; Li, Z.; Tang, B.; Wei, W.; Wang, J.; Tang, Z.; Song, S.; et al. *Memory*^{3}: Language Modeling with Explicit Memory. *arXiv preprint arXiv:2407.01178* **2024**.

Dathathri, S.; Madotto, A.; Lan, J.; Hung, J.; Frank, E.; Molino, P.; Yosinski, J.; Liu, R. Plug and Play Language Models: A Simple Approach to Controlled Text Generation. In Proceedings of the International Conference on Learning Representations.

Turner, A.M.; Thiergart, L.; Leech, G.; Udell, D.; Vazquez, J.J.; Mini, U.; MacDiarmid, M. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248* **2023**.

Liu, S.; Ye, H.; Xing, L.; Zou, J. In-context vectors: Making in context learning more effective and controllable through latent space steering. *arXiv* **2023**, [2311.06668].

Zou, A.; Phan, L.; Chen, S.; Campbell, J.; Guo, P.; Ren, R.; Pan, A.; Yin, X.; Mazeika, M.; Dombrowski, A.K.; et al. Representation engineering: A top-down approach to ai transparency. *arXiv* **2023**, [2310.01405].

Arditi, A.; Obeso, O.; Syed, A.; Paleka, D.; Panickssery, N.; Gurnee, W.; Nanda, N. Refusal in language models is mediated by a single direction. *arXiv* **2024**, [2406.11717].

Chugtai, B.; Bushnaq, L. Activation space interpretability may be doomed, 2025.

Subramani, N.; Suresh, N.; Peters, M.E. Extracting latent steering vectors from pretrained language models. *arXiv* **2022**, [2205.05124].

Hernandez, E.; Li, B.Z.; Andreas, J. Inspecting and editing knowledge representations in language models. *arXiv* **2023**, [2304.00740].

Dunefsky, J.; Cohan, A. Investigating generalization of one-shot LLM steering vectors. *arXiv preprint arXiv:2502.18862* **2025**.

Mack, A.; Turner, A. Mechanistically eliciting latent behaviors in language models, 2024.

Li, K.; Patel, O.; Viégas, F.; Pfister, H.; Wattenberg, M. Inference-time intervention: Eliciting truthful answers from a language model. In Proceedings of the Advances in Neural Information Processing Systems, 2024, Vol. 36.

Turner, A.; Kurzeja, M.; Orr, D.; Elson, D. Steering gemini using bidpo vectors, 2025.

Cao, Y.; Zhang, T.; Cao, B.; Yin, Z.; Lin, L.; Ma, F.; Chen, J. Personalized steering of large language models: Versatile steering vectors through bi-directional preference optimization. *arXiv* **2024**, [2406.00045].

Allen-Zhu, Z.; Li, Y. Physics of language models: part 3.1, knowledge storage and extraction. In Proceedings of the Proceedings of the 41st International Conference on Machine Learning, 2024, pp. 1067–1077.

Carlini, N.; Tramer, F.; Wallace, E.; Jagielski, M.; Herbert-Voss, A.; Lee, K.; Roberts, A.; Brown, T.; Song, D.; Erlingsson, U.; et al. Extracting Training Data from Large Language Models, 2021, [arXiv:cs.CR/2012.07805].

Lee, K.; Ippolito, D.; Nystrom, A.; Zhang, C.; Eck, D.; Callison-Burch, C.; Carlini, N. Deduplicating Training Data Makes Language Models Better. In Proceedings of the Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics. Association for Computational Linguistics, 2022.

Kandpal, N.; Wallace, E.; Raffel, C. Deduplicating Training Data Mitigates Privacy Risks in Language Models, 2022, [arXiv:cs.CR/2202.06539].

Carlini, N.; Ippolito, D.; Jagielski, M.; Lee, K.; Tramer, F.; Zhang, C. Quantifying Memorization Across Neural Language Models, 2023, [arXiv:cs.LG/2202.07646].

Wang, Z.; Bao, R.; Wu, Y.; Taylor, J.; Xiao, C.; Zheng, F.; Jiang, W.; Gao, S.; Zhang, Y. Unlocking Memorization in Large Language Models with Dynamic Soft Prompting. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing; Al-Onaizan, Y.; Bansal, M.; Chen, Y.N., Eds., Miami, Florida, USA, 2024; pp. 9782–9796. https://doi.org/10.18653/v1/2024.emnlp-main.546.

 doi:10.20944/preprints202603.0359.v2

Tirumala, K.; Markosyan, A.H.; Zettlemoyer, L.; Aghajanyan, A. Memorization Without Overfitting: Analyzing the Training Dynamics of Large Language Models, 2022, [arXiv:cs.CL/2205.10770].

Freeman, J.; Rippe, C.; Debenedetti, E.; Andriushchenko, M. Exploring Memorization and Copyright Violation in Frontier LLMs: A Study of the New York Times v. OpenAI 2023 Lawsuit, 2024, [arXiv:cs.LG/2412.06370].

Geva, M.; Schuster, R.; Berant, J.; Levy, O. Transformer feed-forward layers are key-value memories. *arXiv* **2021**, [2109.04554].

Dai, D.; Dong, L.; Hao, Y.; Sui, Z.; Chang, B.; Wei, F. Knowledge neurons in pretrained transformers. *arXiv* **2021**, [2104.08696].

Wang, L.; Zhang, X.; Su, H.; Zhu, J. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence* **2024**.

Kirkpatrick, J.; Pascanu, R.; Rabinowitz, N.; Veness, J.; Desjardins, G.; Rusu, A.A.; Milan, K.; Quan, J.; Ramalho, T.; Grabska-Barwinska, A.; et al. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences* **2017**, *114*, 3521–3526.

Feng, Y.; Chu, X.; Xu, Y.; Shi, G.; Liu, B.; Wu, X.M. TaSL: Continual Dialog State Tracking via Task Skill Localization and Consolidation. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers); Ku, L.W.; Martins, A.; Srikumar, V., Eds., Bangkok, Thailand, 2024; pp. 1266–1279. https://doi.org/10.18653/v1/2024.acl-long.69.

Wang, Y.; Liu, X.; Chen, X.; O'Brien, S.; Wu, J.; McAuley, J. Self-Updatable Large Language Models by Integrating Context into Model Parameters. In Proceedings of the The Thirteenth International Conference on Learning Representations.

Wu, Y.; Wang, H.; Zhao, P.; Zheng, Y.; Wei, Y.; Huang, L.K. Mitigating catastrophic forgetting in online continual learning by modeling previous task interrelations via pareto optimization. In Proceedings of the Forty-first International Conference on Machine Learning, 2024.

Mehta, S.V.; Gupta, J.; Tay, Y.; Dehghani, M.; Tran, V.Q.; Rao, J.; Najork, M.; Strubell, E.; Metzler, D. DSI++: Updating transformer memory with new documents. *arXiv preprint arXiv:2212.09744* **2022**.

Wang, Y.; Han, C.; Wu, T.; He, X.; Zhou, W.; Sadeq, N.; Chen, X.; He, Z.; Wang, W.; Haffari, G.; et al. Towards lifespan cognitive systems. *arXiv preprint arXiv:2409.13265* **2024**.

Han, Z.; Gao, C.; Liu, J.; Zhang, J.; Zhang, S.Q. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608* **2024**.

Shao, Y.; Li, L.; Dai, J.; Qiu, X. Character-llm: A trainable agent for role-playing.

Shang, J.; Zheng, Z.; Wei, J.; Ying, X.; Tao, F.; Team, M. Ai-native memory: A pathway from llms towards agi. *arXiv preprint arXiv:2406.18312* **2024**.

Liu, W.; Zhang, R.; Zhou, A.; Gao, F.; Liu, J. Echo: A large language model with temporal episodic memory. *arXiv preprint arXiv:2502.16090* **2025**.

McMahan, B.; Moore, E.; Ramage, D.; Hampson, S.; y Arcas, B.A. Communication-efficient learning of deep networks from decentralized data. In Proceedings of the AISTATS. PMLR, 2017, pp. 1273–1282.

Marczak, D.; Twardowski, B.; Trzciński, T.; Cygert, S. MagMax: Leveraging Model Merging for Seamless Continual Learning. In Proceedings of the ECCV, 2024.

Lee, N.; Ajanthan, T.; Torr, P. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In Proceedings of the International Conference on Learning Representations, 2019.

Qu, Z.; Li, X.; Duan, R.; Liu, Y.; Tang, B.; Lu, Z. Generalized federated learning via sharpness aware minimization. In Proceedings of the International Conference on Machine Learning. PMLR, 2022, pp. 18250–18280.

Matena, M.S.; Raffel, C.A. Merging models with fisher-weighted averaging. *NeurIPS* **2022**, *35*, 17703–17716.

Yadav, P.; Tam, D.; Choshen, L.; Raffel, C.; Bansal, M. TIES-Merging: Resolving Interference When Merging Models **2023**. [2306.01708].

Yu, L.; Yu, B.; Yu, H.; Huang, F.; Li, Y. Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. *ICML* **2024**, [2311.03099].

Davari, M.; Belilovsky, E. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *arXiv preprint arXiv:2312.06795* **2023**.

Shazeer, N.; Mirhoseini, A.; Maziarz, K.; Davis, A.; Le, Q.; Hinton, G.; Dean, J. Outrageously Large Neural Networks: The Sparsely-Gated Mixture-of-Experts Layer **2017**. [1701.06538].

Muqeeth, M.; Liu, H.; Raffel, C. Soft merging of experts with adaptive routing. *TMLR* **2024**.

Ilharco, G.; Ribeiro, M.T.; Wortsman, M.; Gururangan, S.; Schmidt, L.; Hajishirzi, H.; Farhadi, A. Editing Models with Task Arithmetic **2022**. [2212.04089].

 doi:10.20944/preprints202603.0359.v2

Yang, E.; Wang, Z.; Shen, L.; Liu, S.; Guo, G.; Wang, X.; Tao, D. AdaMerging: Adaptive Model Merging for Multi-Task Learning. *ICLR* **2024**, [2310.02575].

Lu, Z.; Fan, C.; Wei, W.; Qu, X.; Chen, D.; Cheng, Y. Twin-Merging: Dynamic Integration of Modular Expertise in Model Merging. *NIPS* **2024**, [2406.15479].

Wang, S.; Zhu, Y.; Liu, H.; Zheng, Z.; Chen, C.; Li, J. Knowledge editing for large language models: A survey. *ACM Computing Surveys* **2024**, *57*, 1–37.

Wang, Y.; Gao, Y.; Chen, X.; Jiang, H.; Li, S.; Yang, J.; Yin, Q.; Li, Z.; Li, X.; Yin, B.; et al. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624* **2024**.

Wang, P.; Li, Z.; Zhang, N.; Xu, Z.; Yao, Y.; Jiang, Y.; Xie, P.; Huang, F.; Chen, H. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems* **2024**, *37*, 53764–53797.

Nasr, M.; Carlini, N.; Hayase, J.; Jagielski, M.; Cooper, A.F.; Ippolito, D.; Choquette-Choo, C.A.; Wallace, E.; Tramèr, F.; Lee, K. Scalable Extraction of Training Data from (Production) Language Models, 2023, [arXiv:cs.LG/2311.17035].

Ippolito, D.; Tramer, F.; Nasr, M.; Zhang, C.; Jagielski, M.; Lee, K.; Choquette Choo, C.; Carlini, N. Preventing Generation of Verbatim Memorization in Language Models Gives a False Sense of Privacy. In Proceedings of the Proceedings of the 16th International Natural Language Generation Conference; Keet, C.M.; Lee, H.Y.; Zarrieß, S., Eds., Prague, Czechia, 2023; pp. 28–53. https://doi.org/10.18653/v1/2023.inlg-main.3.

Biderman, S.; Prashanth, U.S.; Sutawika, L.; Schoelkopf, H.; Anthony, Q.; Purohit, S.; Raff, E. Emergent and Predictable Memorization in Large Language Models, 2023, [arXiv:cs.CL/2304.11158].

Tulving, E.; Donaldson, W. *Episodic and semantic memory*; Academic Press, 1972.

Begg, I. Tulving's memory [Review of the book Elements of episodic memory, by E. Tulving]. *Canadian Journal of Psychology / Revue canadienne de psychologie* **1984**, *38*, 144–147.

Squire, L.R. Memory and brain systems: 1969-2009. J Neurosci. 2009 Oct 14;29(41):12711-6. doi: 10.1523/JNEUROSCI.3575-09.2009. PMID: 19828780; PMCID: PMC2791502. *J Neurosci* **2009**, *29*, 12711–12716.

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; Lewis, M. Efficient Streaming Language Models with Attention Sinks. *arXiv* **2023**.

Meng, K.; Sharma, A.S.; Andonian, A.J.; Belinkov, Y.; Bau, D. Mass-Editing Memory in a Transformer. In Proceedings of the The Eleventh International Conference on Learning Representations, 2023.

Zhang, Z.; Sheng, Y.; Zhou, T.; Chen, T.; Zheng, L.; Cai, R.; Song, Z.; Tian, Y.; Ré, C.; Barrett, C.; et al. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems* **2023**, *36*, 34661–34710.

Xi, Z.; Chen, W.; Guo, X.; He, W.; Ding, Y.; Hong, B.; Zhang, M.; Wang, J.; Jin, S.; Zhou, E.; et al. The rise and potential of large language model based agents: A survey. *arxiv* **2023**, [arXiv:cs.AI/arXiv:2309.07864].

Zhao, P.; Jin, Z.; Cheng, N. An in-depth survey of large language model-based artificial intelligence agents. *arXiv* **2023**, [2309.14365].

Cheng, Y.; Zhang, C.; Zhang, Z.; Meng, X.; Hong, S.; Li, W.; Wang, Z.; Wang, Z.; Yin, F.; Zhao, J.; et al. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv* **2024**, [2401.03428].

Ge, Y.; Ren, Y.; Hua, W.; Xu, S.; Tan, J.; Zhang, Y. Llm as os (llmao), agents as apps: Envisioning aios, agents and the aios-agent ecosystem. *arXiv* **2023**, [2312.03815].

Durante, Z.; Huang, Q.; Wake, N.; Gong, R.; Park, J.S.; Sarkar, B.; Taori, R.; Noda, Y.; Terzopoulos, D.; Choi, Y.; et al. Agent ai: Surveying the horizons of multimodal interaction. *arXiv* **2024**, [2401.03568].

Huang, X.; Liu, W.; Chen, X.; Wang, X.; Wang, H.; Lian, D.; Wang, Y.; Tang, R.; Chen, E. Understanding the planning of llm agents: A survey. *arXiv* **2024**, [2402.02716].

Guo, T.; Chen, X.; Wang, Y.; Chang, R.; Pei, S.; Chawla, N.V.; Wiest, O.; Zhang, X. Large language model based multi-agents: A survey of progress and challenges. *arXiv* **2024**, [2402.01680].

Li, Y.; Wen, H.; Wang, W.; Li, X.; Yuan, Y.; Liu, G.; Liu, J.; Xu, W.; Wang, X.; Sun, Y.; et al. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv* **2024**, [2401.05459].

Zhu, Y.; Yuan, H.; Wang, S.; Liu, J.; Liu, W.; Deng, C.; Dou, Z.; Wen, J.R. Large language models for information retrieval: A survey. *arXiv* **2023**, [2308.07107].

Xu, D.; Chen, W.; Peng, W.; Zhang, C.; Xu, T.; Zhao, X.; Wu, X.; Zheng, Y.; Chen, E. Large language models for generative information extraction: A survey. *arXiv* **2023**, [2312.17617].

Li, L.; Zhang, Y.; Liu, D.; Chen, L. Large language models for generative recommendation: A survey and visionary discussions. *arXiv* **2023**, [2309.01157].

 doi:10.20944/preprints202603.0359.v2

Lin, J.; Dai, X.; Xi, Y.; Liu, W.; Chen, B.; Li, X.; Zhu, C.; Guo, H.; Yu, Y.; Tang, R.; et al. How can recommender systems benefit from large language models: A survey. *arXiv* **2023**, [2306.05817].

Wang, W.; Lin, X.; Feng, F.; He, X.; Chua, T.S. Generative recommendation: Towards next-generation recommender paradigm. *arXiv* **2023**, [2304.03516].

Fan, A.; Gokkaya, B.; Harman, M.; Lyubarskiy, M.; Sengupta, S.; Yoo, S.; Zhang, J.M. Large language models for software engineering: Survey and open problems. *arXiv* **2023**, [2310.03533].

Wang, J.; Huang, Y.; Chen, C.; Liu, Z.; Wang, S.; Wang, Q. Software testing with large language models: Survey, landscape, and vision. *IEEE Transactions on Software Engineering* **2024**.

Zheng, Z.; Ning, K.; Wang, Y.; Zhang, J.; Zheng, D.; Ye, M.; Chen, J. A survey of large language models for code: Evolution, benchmarking, and future trends. *arXiv* **2023**, [2311.10372].

Zeng, F.; Gan, W.; Wang, Y.; Liu, N.; Yu, P.S. Large language models for robotics: A survey. *arXiv* **2023**, [2311.07226].

Cui, C.; Ma, Y.; Cao, X.; Ye, W.; Zhou, Y.; Liang, K.; Chen, J.; Lu, J.; Yang, Z.; Liao, K.D.; et al. A survey on multimodal large language models for autonomous driving. In Proceedings of the Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision, 2024, pp. 958–979.

Yang, Z.; Jia, X.; Li, H.; Yan, J. A survey of large language models for autonomous driving. *arXiv* **2023**, [2311.01043].

He, K.; Mao, R.; Lin, Q.; Ruan, Y.; Lan, X.; Feng, M.; Cambria, E. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics. *arXiv* **2023**, [2310.05694].

Zhou, H.; Gu, B.; Zou, X.; Li, Y.; Chen, S.S.; Zhou, P.; Liu, J.; Hua, Y.; Mao, C.; Wu, X.; et al. A survey of large language models in medicine: Progress, application, and challenge. *arXiv* **2023**, [2311.05112].

Wang, B.; Xie, Q.; Pei, J.; Chen, Z.; Tiwari, P.; Li, Z.; Fu, J. Pre-trained language models in biomedical domain: A systematic survey. *ACM Computing Surveys* **2023**, *56*, 1–52.

Li, Y.; Wang, S.; Ding, H.; Chen, H. Large language models in finance: A survey. In Proceedings of the Proceedings of the Fourth ACM International Conference on AI in Finance, 2023, pp. 374–382.

He, T.; Fu, G.; Yu, Y.; Wang, F.; Li, J.; Zhao, Q.; Song, C.; Qi, H.; Luo, D.; Zou, H.; et al. Towards a psychological generalist ai: A survey of current applications of large language models and future prospects. *arXiv* **2023**, [2312.04578].

He, Z.; Lin, W.; Zheng, H.; Zhang, F.; Jones, M.W.; Aitchison, L.; Xu, X.; Liu, M.; Kristensson, P.O.; Shen, J. Human-inspired Perspectives: A Survey on AI Long-term Memory. *arXiv preprint arXiv:2411.00489* **2024**.

Du, Y.; Huang, W.; Zheng, D.; Wang, Z.; Montella, S.; Lapata, M.; Wong, K.F.; Pan, J.Z. Rethinking Memory in AI: Taxonomy, Operations, Topics, and Future Directions, 2025, [arXiv:cs.CL/2505.00675].

Zeng, R.; Fang, J.; Liu, S.; Meng, Z. On the structural memory of llm agents. *arXiv preprint arXiv:2412.15266* **2024**.

Hatalis, K.; Christou, D.; Myers, J.; Jones, S.; Lambert, K.; Amos-Binks, A.; Dannenhauer, Z.; Dannenhauer, D. Memory Matters: The Need to Improve Long-Term Memory in LLM-Agents. *Proceedings of the AAAI Symposium Series* **2024**, *2*.

Zhou, Z.; Ning, X.; Hong, K.; Fu, T.; Xu, J.; Li, S.; Lou, Y.; Wang, L.; Yuan, Z.; Li, X.; et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294* **2024**.

Shan, L.; Luo, S.; Zhu, Z.; Yuan, Y.; Wu, Y. Cognitive memory in large language models. *arXiv preprint arXiv:2504.02441* **2025**.

Baddeley, A.D.; Hitch, G. Working Memory; Academic Press, 1974; Vol. 8, *Psychology of Learning and Motivation*, pp. 47–89. https://doi.org/https://doi.org/10.1016/S0079-7421(08)60452-1.

Sridhar, S.; Khamaj, A.; Asthana, M. Cognitive neuroscience perspective on memory: overview and summary. *Frontiers in human neuroscience* **2023**, *17*, 1217093.

Sherwood, L.; Kell, R.T.; Ward, C. *Human physiology: from cells to systems*; Thomson/Brooks/Cole, 2004.

Weng, L. Llm-powered autonomous agents. lilianweng.github.io, 2023.

Solso, R.L.; Kagan, J. *Cognitive psychology*; Houghton Mifflin Harcourt P, 1979.

Craik, F.I.; Lockhart, R.S. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior* **1972**, *11*, 671–684.

Leydesdorff, S. *Memory cultures: Memory, subjectivity and recognition*; Routledge, 2017.

Johnson-Laird, P.N. *Mental models: Towards a cognitive science of language, inference, and consciousness*; Vol. 6, Harvard University Press, 1983.

Laird, J.E. *The Soar cognitive architecture*; MIT press, 2019.

Sun, R. *Duality of the mind: A bottom-up approach toward cognition*; Psychology Press, 2001.

Sutton, R.S.; Barto, A.G. *Reinforcement learning: An introduction*; MIT press, 2018.

Zheng, L.; Wang, R.; Wang, X.; An, B. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In Proceedings of the NeurIPS 2023 Foundation Models for Decision Making Workshop, 2023.

 doi:10.20944/preprints202603.0359.v2

Montazeralghaem, A.; Zamani, H.; Allan, J. A reinforcement learning framework for relevance feedback. In Proceedings of the Proceedings of the 43rd international acm sigir conference on research and development in information retrieval, 2020, pp. 59–68.

Zhu, X.; Chen, Y.; Tian, H.; Tao, C.; Su, W.; Yang, C.; Huang, G.; Li, B.; Lu, L.; Wang, X.; et al. Ghost in the minecraft: Generally capable agents for open-world enviroments via large language models with text-based knowledge and memory. *arXiv preprint arXiv:2305.17144* **2023**.

Zhao, A.; Huang, D.; Xu, Q.; Lin, M.; Liu, Y.J.; Huang, G. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144* **2023**.

Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language Models are Few-Shot Learners, 2020, [arXiv:cs.CL/2005.14165].

Qwen.; :.; Yang, A.; Yang, B.; Zhang, B.; Hui, B.; Zheng, B.; et al. Qwen2.5 Technical Report, 2025, [arXiv:cs.CL/2412.15115].

Bai, J.; Bai, S.; Chu, Y.; Cui, Z.; Dang, K.; Deng, X.; et al. Qwen Technical Report, 2023, [arXiv:cs.CL/2309.16609].

Meng, F.; Tang, P.; Tang, X.; Yao, Z.; Sun, X.; Zhang, M. TransMLA: Multi-Head Latent Attention Is All You Need, 2025, [arXiv:cs.LG/2502.07864].

Vaswani, A.; Shazeer, N.; Parmar, N.; Uszkoreit, J.; Jones, L.; Gomez, A.N.; Kaiser, L.; Polosukhin, I. Attention is all you need. In Proceedings of the Proceedings of the 31st International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 2017; NIPS'17, p. 6000–6010.

Dai, D.; Sun, Y.; Dong, L.; Hao, Y.; Ma, S.; Sui, Z.; Wei, F. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, 2023; pp. 4005–4019.

Hahn, M.; Goyal, N. A theory of emergent in-context learning as implicit structure induction. *arxiv* **2023**, *arXiv:2303.07971*.

Garg, S.; Tsipras, D.; Liang, P.S.; Valiant, G. What can transformers learn in-context? A case study of simple function classes. In Proceedings of the Advances in Neural Information Processing Systems, 2022, Vol. 35, pp. 30583–30598.

Min, S.; Lyu, X.; Holtzman, A.; Artetxe, M.; Lewis, M.; Hajishirzi, H.; Zettlemoyer, L. Rethinking the role of demonstrations: What makes in-context learning work? In Proceedings of the Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing, Abu Dhabi, United Arab Emirates, 2022; pp. 11048–11064.

Pan, J.; Gao, T.; Chen, H.; Chen, D. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In Proceedings of the Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, 2023; pp. 8298–8319.

Singh, A.K.; Chan, S.C.; Moskovitz, T.; Grant, E.; Saxe, A.M.; Hill, F. The transient nature of emergent in-context learning in transformers. In Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, 2023.

Raventos, A.; Paul, M.; Chen, F.; Ganguli, S. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In Proceedings of the Thirty-seventh Conference on Neural Information Processing Systems, 2023.

Shen, L.; Mishra, A.; Khashabi, D. Do pretrained transformers learn in-context by gradient descent? *arxiv* **2024**, *arXiv:2310.08540*.

Liu, X.; Chen, H.; Hu, X.; Chu, X. FlowKV: Enhancing Multi-Turn Conversational Coherence in LLMs via Isolated Key-Value Cache Management. In Proceedings of the First Workshop on Multi-Turn Interactions in Large Language Models, 2025.

Li, Y.; Huang, Y.; Yang, B.; Venkitesh, B.; Locatelli, A.; Ye, H.; Cai, T.; Lewis, P.; Chen, D. SnapKV: LLM Knows What You are Looking for Before Generation. *arXiv preprint arXiv:2404.14469* **2024**.

Zhang, Z.; Sheng, Y.; Zhou, T.; Chen, T.; Zheng, L.; Cai, R.; Song, Z.; Tian, Y.; Ré, C.; Barrett, C.; et al. H2O: Heavy-Hitter Oracle for Efficient Generative Inference of Large Language Models. In Proceedings of the Advances in Neural Information Processing Systems; Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; Levine, S., Eds. Curran Associates, Inc., 2023, Vol. 36, pp. 34661–34710.

Lu, J.; An, S.; Lin, M.; Pergola, G.; He, Y.; Yin, D.; Sun, X.; Wu, Y. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239* **2023**.

Wang, Q.; Tang, Z.; He, B. Can LLM Simulations Truly Reflect Humanity? A Deep Dive. In Proceedings of the The Fourth Blogpost Track at ICLR 2025, 2025.

 doi:10.20944/preprints202603.0359.v2

Wang, L.; Zhang, J.; Yang, H.; Chen, Z.; Tang, J.; Zhang, Z.; Chen, X.; Lin, Y.; Song, R.; Zhao, W.X.; et al. When large language model based agent meets user behavior analysis: A novel user simulation paradigm **2023**.

Jin, B.; Yoon, J.; Han, J.; Arik, S.Ö. Long-Context LLMs Meet RAG: Overcoming Challenges for Long Inputs in RAG. *CoRR* **2024**, *abs/2410.05983*, [2410.05983]. https://doi.org/10.48550/ARXIV.2410.05983.

Shi, K.; Sun, X.; Li, Q.; Xu, G. Compressing Long Context for Enhancing RAG with AMR-based Concept Distillation. *CoRR* **2024**, *abs/2405.03085*, [2405.03085]. https://doi.org/10.48550/ARXIV.2405.03085.

Beltagy, I.; Peters, M.E.; Cohan, A. Longformer: The Long-Document Transformer. *CoRR* **2020**, *abs/2004.05150*, [2004.05150].

Guo, M.; Ainslie, J.; Uthus, D.C.; Ontañón, S.; Ni, J.; Sung, Y.; Yang, Y. LongT5: Efficient Text-To-Text Transformer for Long Sequences. In Proceedings of the Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022; Carpuat, M.; de Marneffe, M.; Ruíz, I.V.M., Eds. Association for Computational Linguistics, 2022, pp. 724–736. https://doi.org/10.18653/V1/2022.FINDINGS-NAACL.55.

Jin, H.; Zhang, Y.; Meng, D.; Wang, J.; Tan, J. A Comprehensive Survey on Process-Oriented Automatic Text Summarization with Exploration of LLM-Based Methods. *CoRR* **2024**, *abs/2403.02901*, [2403.02901]. https://doi.org/10.48550/ARXIV.2403.02901.

Zhu, Y.; Yuan, H.; Wang, S.; Liu, J.; Liu, W.; Deng, C.; Dou, Z.; Wen, J. Large Language Models for Information Retrieval: A Survey. *CoRR* **2023**, *abs/2308.07107*, [2308.07107]. https://doi.org/10.48550/ARXIV.2308.07107.

Wang, L.; Yang, N.; Huang, X.; Yang, L.; Majumder, R.; Wei, F. Improving Text Embeddings with Large Language Models. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024; Ku, L.; Martins, A.; Srikumar, V., Eds. Association for Computational Linguistics, 2024, pp. 11897–11916. https://doi.org/10.18653/V1/2024.ACL-LONG.642.

OpenAI. New Embedding Models and API Updates, 2024. Accessed: 2024-01-25.

Günther, M.; Ong, J.; Mohr, I.; Abdessalem, A.; Abel, T.; Akram, M.K.; Guzman, S.; Mastrapas, G.; Sturua, S.; Wang, B.; et al. Jina Embeddings 2: 8192-Token General-Purpose Text Embeddings for Long Documents. *CoRR* **2023**, *abs/2310.19923*, [2310.19923]. https://doi.org/10.48550/ARXIV.2310.19923.

Chen, J.; Xiao, S.; Zhang, P.; Luo, K.; Lian, D.; Liu, Z. BGE M3-Embedding: Multi-Lingual, Multi-Functionality, Multi-Granularity Text Embeddings Through Self-Knowledge Distillation. *CoRR* **2024**, *abs/2402.03216*, [2402.03216]. https://doi.org/10.48550/ARXIV.2402.03216.

Zhu, D.; Wang, L.; Yang, N.; Song, Y.; Wu, W.; Wei, F.; Li, S. LongEmbed: Extending Embedding Models for Long Context Retrieval. In Proceedings of the Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024; Al-Onaizan, Y.; Bansal, M.; Chen, Y., Eds. Association for Computational Linguistics, 2024, pp. 802–816.

Saad-Falcon, J.; Fu, D.Y.; Arora, S.; Guha, N.; Ré, C. Benchmarking and Building Long-Context Retrieval Models with LoCo and M2-BERT. In Proceedings of the Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024. OpenReview.net, 2024.

Herold, C.; Ney, H. Improving Long Context Document-Level Machine Translation. *CoRR* **2023**, *abs/2306.05183*, [2306.05183]. https://doi.org/10.48550/ARXIV.2306.05183.

Wang, L.; Du, Z.; Jiao, W.; Lyu, C.; Pang, J.; Cui, L.; Song, K.; Wong, D.F.; Shi, S.; Tu, Z. Benchmarking and Improving Long-Text Translation with Large Language Models. In Proceedings of the Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024; Ku, L.; Martins, A.; Srikumar, V., Eds. Association for Computational Linguistics, 2024, pp. 7175–7187. https://doi.org/10.18653/V1/2024.FINDINGS-ACL.428.

Lyu, C.; Du, Z.; Xu, J.; Duan, Y.; Wu, M.; Lynn, T.; Aji, A.F.; Wong, D.F.; Wang, L. A Paradigm Shift: The Future of Machine Translation Lies with Large Language Models. In Proceedings of the Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy; Calzolari, N.; Kan, M.; Hoste, V.; Lenci, A.; Sakti, S.; Xue, N., Eds. ELRA and ICCL, 2024, pp. 1339–1352.

OpenAI. Memory and New Controls for ChatGPT, 2024. Accessed: 2024-02-13.

Inflection. I'm Pi, Your personal AI. https://inflection.ai/, 2023.

Character AI. Character AI. Retrieved September 14, 2023 from https://character.ai/, 2023.

Ai, T. Talkie | AI-Native Character Community, 2024.

Lee, G.; Hartmann, V.; Park, J.; Papailiopoulos, D.; Lee, K. Prompted LLMs as Chatbot Modules for Long Open-domain Conversation. In Proceedings of the Findings of the Association for Computational Linguistics:

 doi:10.20944/preprints202603.0359.v2

ACL 2023, Toronto, Canada, July 9-14, 2023; Rogers, A.; Boyd-Graber, J.L.; Okazaki, N., Eds. Association for Computational Linguistics, 2023, pp. 4536–4554. https://doi.org/10.18653/V1/2023.FINDINGS-ACL.277.

Zhong, W.; Guo, L.; Gao, Q.; Ye, H.; Wang, Y. MemoryBank: Enhancing Large Language Models with Long-Term Memory. In Proceedings of the Thirty-Eighth AAAI Conference on Artificial Intelligence, AAAI 2024, Thirty-Sixth Conference on Innovative Applications of Artificial Intelligence, IAAI 2024, Fourteenth Symposium on Educational Advances in Artificial Intelligence, EAAI 2014, February 20-27, 2024, Vancouver, Canada; Wooldridge, M.J.; Dy, J.G.; Natarajan, S., Eds. AAAI Press, 2024, pp. 19724–19731. https://doi.org/10.1609/AAAI.V38I17.29946.

Wang, W.; Dong, L.; Cheng, H.; Liu, X.; Yan, X.; Gao, J.; Wei, F. Augmenting Language Models with Long-Term Memory. In Proceedings of the Advances in Neural Information Processing Systems 36: Annual Conference on Neural Information Processing Systems 2023, NeurIPS 2023, New Orleans, LA, USA, December 10 - 16, 2023; Oh, A.; Naumann, T.; Globerson, A.; Saenko, K.; Hardt, M.; Levine, S., Eds., 2023.

Wang, X.; Salmani, M.; Omidi, P.; Ren, X.; Rezagholizadeh, M.; Eshaghi, A. Beyond the Limits: A Survey of Techniques to Extend the Context Length in Large Language Models. In Proceedings of the Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024. ijcai.org, 2024, pp. 8299–8307.

Tsai, Y.; Liu, M.; Ren, H. Rtlfixer: Automatically fixing rtl syntax errors with large language models. *arXiv preprint arXiv:2311.16543* **2023**.

Chen, D.; Wang, H.; Huo, Y.; Li, Y.; Zhang, H. Gamegpt: Multi-agent collaborative framework for game development. *arXiv preprint arXiv:2310.08067* **2023**.

Li, Y.; Zhang, Y.; Sun, L. Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents. *arXiv preprint arXiv:2310.06500* **2023**.

Zhang, K.; Li, J.; Li, G.; Shi, X.; Jin, Z. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. *arXiv preprint arXiv:2401.07339* **2024**.

Lozhkov, A.; Li, R.; Allal, L.B.; Cassano, F.; Lamy-Poirier, J.; Tazi, N.; Tang, A.; Pykhtar, D.; Liu, J.; Wei, Y.; et al. StarCoder 2 and The Stack v2: The Next Generation. *CoRR* **2024**, *abs/2402.19173*, [2402.19173]. https://doi.org/10.48550/ARXIV.2402.19173.

Hui, B.; Yang, J.; Cui, Z.; Yang, J.; Liu, D.; Zhang, L.; Liu, T.; Zhang, J.; Yu, B.; Dang, K.; et al. Qwen2.5-Coder Technical Report. *CoRR* **2024**, *abs/2409.12186*, [2409.12186]. https://doi.org/10.48550/ARXIV.2409.12186.

Mishra, M.; Stallone, M.; Zhang, G.; Shen, Y.; Prasad, A.; Soria, A.M.; Merler, M.; Selvam, P.; Surendran, S.; Singh, S.; et al. Granite Code Models: A Family of Open Foundation Models for Code Intelligence. *CoRR* **2024**, *abs/2405.04324*, [2405.04324]. https://doi.org/10.48550/ARXIV.2405.04324.

GitHub. GitHub Copilot, 2022.

Anysphere. Cursor - The AI Code Editor. https://www.cursor.com/en, 2025.

Shao, Y.; Li, L.; Dai, J.; Qiu, X. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158* **2023**.

Kaiya, Z.; Naim, M.; Kondic, J.; Cortes, M.; Ge, J.; Luo, S.; Yang, G.R.; Ahn, A. Lyfe agents: Generative agents for low-cost real-time social interactions. *arXiv* **2023**, [2310.02172].

Li, N.; Gao, C.; Li, Y.; Liao, Q. Large language model-empowered agents for simulating macroeconomic activities. *arXiv* **2023**, [2310.10436].

Hua, W.; Fan, L.; Li, L.; Mei, K.; Ji, J.; Ge, Y.; Hemphill, L.; Zhang, Y. War and peace (waragent): Large language model-based multi-agent simulation of world wars. *arXiv preprint arXiv:2311.17227* **2023**.

Lee, G.; Hartmann, V.; Park, J.; Papailiopoulos, D.; Lee, K. Prompted llms as chatbot modules for long open-domain conversation. *arXiv preprint arXiv:2305.04533* **2023**.

Pan, H.; Zhai, Z.; Yuan, H.; Lv, Y.; Fu, R.; Liu, M.; Wang, Z.; Qin, B. Kwaiagents: Generalized information-seeking agent system with large language models. *arXiv* **2023**, [2312.04889].

Wu, Q.; Bansal, G.; Zhang, J.; Wu, Y.; Zhang, S.; Zhu, E.; Li, B.; Jiang, L.; Zhang, X.; Wang, C. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155* **2023**.

Gao, S.; Chen, X.; Li, P.; Ren, Z.; Bing, L.; Zhao, D.; Yan, R. Abstractive Text Summarization by Incorporating Reader Comments. In Proceedings of the The Thirty-Third AAAI Conference on Artificial Intelligence, AAAI 2019, The Thirty-First Innovative Applications of Artificial Intelligence Conference, IAAI 2019, The Ninth AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019, Honolulu, Hawaii, USA, January 27 - February 1, 2019. AAAI Press, 2019, pp. 6399–6406. https://doi.org/10.1609/AAAI.V33I01.33016399.

Kapoor, S.; Henderson, P.; Narayanan, A. Promises and pitfalls of artificial intelligence for legal applications. *CoRR* **2024**, *abs/2402.01656*, [2402.01656]. https://doi.org/10.48550/ARXIV.2402.01656.

 doi:10.20944/preprints202603.0359.v2

Fan, Y.; Sun, H.; Xue, K.; Zhang, X.; Zhang, S.; Ruan, T. MedOdyssey: A Medical Domain Benchmark for Long Context Evaluation Up to 200K Tokens. *CoRR* **2024**, *abs/2406.15019*, [2406.15019]. https://doi.org/10.48550/ARXIV.2406.15019.

Reddy, V.; Koncel-Kedziorski, R.; Lai, V.D.; Krumdick, M.; Lovering, C.; Tanner, C. DocFinQA: A Long-Context Financial Reasoning Dataset. In Proceedings of the Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024; Ku, L.; Martins, A.; Srikumar, V., Eds. Association for Computational Linguistics, 2024, pp. 445–458.

Masry, A.; Hajian, A. LongFin: A Multimodal Document Understanding Model for Long Financial Domain Documents. *CoRR* **2024**, *abs/2401.15050*, [2401.15050]. https://doi.org/10.48550/ARXIV.2401.15050.

Nie, Y.; Kong, Y.; Dong, X.; Mulvey, J.M.; Poor, H.V.; Wen, Q.; Zohren, S. A Survey of Large Language Models for Financial Applications: Progress, Prospects and Challenges. *CoRR* **2024**, *abs/2406.11903*, [2406.11903]. https://doi.org/10.48550/ARXIV.2406.11903.

Hilgert, L.; Liu, D.; Niehues, J. Evaluating and Training Long-Context Large Language Models for Question Answering on Scientific Papers. In Proceedings of the Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U); Kumar, S.; Balachandran, V.; Park, C.Y.; Shi, W.; Hayati, S.A.; Tsvetkov, Y.; Smith, N.; Hajishirzi, H.; Kang, D.; Jurgens, D., Eds., Miami, Florida, USA, 2024; pp. 220–236. https://doi.org/10.18653/v1/2024.customnlp4u-1.17.

Shao, B.; Yan, J. A long-context language model for deciphering and generating bacteriophage genomes. *Nature Communications* **2024**, *15*, 9392.

Wang, H.; Liu, C.; Xi, N.; Qiang, Z.; Zhao, S.; Qin, B.; Liu, T. Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint arXiv:2304.06975* **2023**.

Xiong, H.; Wang, S.; Zhu, Y.; Zhao, Z.; Liu, Y.; Wang, Q.; Shen, D. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097* **2023**.

Liu, Z.; Zhong, A.; Li, Y.; Yang, L.; Ju, C.; Wu, Z.; Ma, C.; Shu, P.; Chen, C.; Kim, S.; et al. Radiology-gpt: A large language model for radiology. *arXiv preprint arXiv:2306.08666* **2023**.

Wang, H.; Zhao, S.; Qiang, Z.; Li, Z.; Xi, N.; Du, Y.; Cai, M.; Guo, H.; Chen, Y.; Xu, H.; et al. Knowledge-tuning large language models with structured medical knowledge bases for reliable response generation in chinese. *arXiv preprint arXiv:2309.04175* **2023**.

Yunxiang, L.; Zihan, L.; Kai, Z.; Ruilong, D.; You, Z. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv preprint arXiv:2303.14070* **2023**.

Chen, K.; Li, J.; Wang, K.; Du, Y.; Yu, J.; Lu, J.; Li, L.; Qiu, J.; Pan, J.; Huang, Y.; et al. Chemist-x: Large language model-empowered agent for reaction condition recommendation in chemical synthesis **2024**.

Zhao, Z.; Ma, D.; Chen, L.; Sun, L.; Li, Z.; Xu, H.; Zhu, Z.; Zhu, S.; Fan, S.; Shen, G.; et al. Chemdfm: Dialogue foundation model for chemistry. *arXiv preprint arXiv:2401.14818* **2024**.

Chen, Z.Y.; Xie, F.K.; Wan, M.; Yuan, Y.; Liu, M.; Wang, Z.G.; Meng, S.; Wang, Y.G. Matchat: A large language model and application service platform for materials science. *Chinese Physics B* **2023**, *32*, 118104.

Wang, Z.; Liu, Z.; Zhang, Y.; Zhong, A.; Fan, L.; Wu, L.; Wen, Q. Rcagent: Cloud root cause analysis by autonomous agents with tool-augmented large language models. *arXiv* **2023**, [2310.16340].

Qiang, Z.; Wang, W.; Taylor, K. Agent-om: Leveraging large language models for ontology matching. *arXiv* **2023**, [2312.00326].

Pope, R.; Douglas, S.; Chowdhery, A.; Devlin, J.; Bradbury, J.; Levskaya, A.; Heek, J.; Xiao, K.; Agrawal, S.; Dean, J. Efficiently Scaling Transformer Inference, 2022, [arXiv:cs.LG/2211.05102].

Lewis, P.; Perez, E.; Piktus, A.; Petroni, F.; Karpukhin, V.; Goyal, N.; Küttler, H.; Lewis, M.; Yih, W.t.; Rocktäschel, T.; et al. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in neural information processing systems* **2020**, *33*, 9459–9474.

Park, J.S.; O'Brien, J.; Cai, C.J.; Morris, M.R.; Liang, P.; Bernstein, M.S. Generative agents: Interactive simulacra of human behavior. In Proceedings of the Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology, 2023, pp. 1–22.

Wulf, W.A.; McKee, S.A. Hitting the memory wall: implications of the obvious. *SIGARCH Comput. Archit. News* **1995**, *23*, 20–24. https://doi.org/10.1145/216585.216588.

Jouppi, N.P.; Young, C.; Patil, N.; Patterson, D.; Agrawal, G.; Bajwa, R.; Bates, S.; Bhatia, S.; Boden, N.; Borchers, A.; et al. In-Datacenter Performance Analysis of a Tensor Processing Unit. *SIGARCH Comput. Archit. News* **2017**, *45*, 1–12. https://doi.org/10.1145/3140659.3080246.

 doi:10.20944/preprints202603.0359.v2

Dao, T.; Fu, D.Y.; Ermon, S.; Rudra, A.; Ré, C. FLASHATTENTION: fast and memory-efficient exact attention with IO-awareness. In Proceedings of the Proceedings of the 36th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 2022; NIPS '22.

Yu, G.I.; Jeong, J.S.; Kim, G.W.; Kim, S.; Chun, B.G. Orca: A Distributed Serving System for Transformer-Based Generative Models. In Proceedings of the 16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22), Carlsbad, CA, 2022; pp. 521–538.

Zhong, Y.; Liu, S.; Chen, J.; Hu, J.; Zhu, Y.; Liu, X.; Jin, X.; Zhang, H. DistServe: disaggregating prefill and decoding for goodput-optimized large language model serving. In Proceedings of the Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation, USA, 2024; OSDI'24.

Brown, T.B.; Mann, B.; Ryder, N.; Subbiah, M.; Kaplan, J.; Dhariwal, P.; Neelakantan, A.; Shyam, P.; Sastry, G.; Askell, A.; et al. Language models are few-shot learners. In Proceedings of the Proceedings of the 34th International Conference on Neural Information Processing Systems, Red Hook, NY, USA, 2020; NIPS '20.

Bommasani, R.; Hudson, D.A.; Adeli, E.; Altman, R.; Arora, S.; von Arx, S.; Bernstein, M.S.; Bohg, J.; Bosselut, A.; Brunskill, E.; et al. On the Opportunities and Risks of Foundation Models, 2022, [arXiv:cs.LG/2108.07258].

Hu, E.J.; Shen, Y.; Wallis, P.; Allen-Zhu, Z.; Li, Y.; Wang, S.; Wang, L.; Chen, W. LoRA: Low-Rank Adaptation of Large Language Models, 2021, [arXiv:cs.CL/2106.09685].

Kwon, W.; Li, Z.; Zhuang, S.; Sheng, Y.; Zheng, L.; Yu, C.H.; Gonzalez, J.; Zhang, H.; Stoica, I. Efficient Memory Management for Large Language Model Serving with PagedAttention. In Proceedings of the Proceedings of the 29th Symposium on Operating Systems Principles, New York, NY, USA, 2023; SOSP '23, p. 611–626. https://doi.org/10.1145/3600006.3613165.

Xiao, G.; Tian, Y.; Chen, B.; Han, S.; Lewis, M. Efficient Streaming Language Models with Attention Sinks, 2024, [arXiv:cs.CL/2309.17453].

Karpukhin, V.; Oguz, B.; Min, S.; Lewis, P.S.; Wu, L.; Edunov, S.; Chen, D.; Yih, W.t. Dense passage retrieval for open-domain question answering. In Proceedings of the EMNLP (1), 2020, pp. 6769–6781.

Khattab, O.; Zaharia, M. ColBERT: Efficient and Effective Passage Search via Contextualized Late Interaction over BERT. In Proceedings of the Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval, New York, NY, USA, 2020; SIGIR '20, p. 39–48. https://doi.org/10.1145/3397271.3401075.

Izacard, G.; Lewis, P.; Lomeli, M.; Hosseini, L.; Petroni, F.; Schick, T.; Dwivedi-Yu, J.; Joulin, A.; Riedel, S.; Grave, E. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.* **2023**, *24*.

Fu, Y.; Xue, L.; Huang, Y.; Brabete, A.O.; Ustiugov, D.; Patel, Y.; Mai, L. ServerlessLLM: low-latency serverless inference for large language models. In Proceedings of the Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation, USA, 2024; OSDI'24.

Jin, H.; Wu, Y. CE-CoLLM: Efficient and Adaptive Large Language Models Through Cloud-Edge Collaboration. In Proceedings of the 2025 IEEE International Conference on Web Services (ICWS), 2025, pp. 316–323. https://doi.org/10.1109/ICWS67624.2025.00046.

Murre, J.M.; Dros, J. Replication and analysis of ebbinghaus' forgetting curve. *PloS one* **2015**, *10*, e0120644.

Wang, Z.Z.; Mao, J.; Fried, D.; Neubig, G. Agent workflow memory. *arXiv preprint arXiv:2409.07429* **2024**.

Jhunjhunwala, D.; Wang, S.; Joshi, G. FedFisher: Leveraging Fisher Information for One-Shot Federated Learning. In Proceedings of the AISTATS. PMLR, 2024, pp. 1612–1620.

Daheim, N.; Möllenhoff, T.; Ponti, E.; Gurevych, I.; Khan, M.E. Model Merging by Uncertainty-Based Gradient Matching. In Proceedings of the ICLR, 2024.

Yu, L.; Yu, B.; Yu, H.; Huang, F.; Li, Y. Language Models are Super Mario: Absorbing Abilities from Homologous Models as a Free Lunch. *ICML* **2024**.

Wang, K.; Dimitriadis, N.; Ortiz-Jimenez, G.; Fleuret, F.; Frossard, P. Localizing Task Information for Improved Model Merging and Compression. *ICML* **2024**.

Lu, Z.; Fan, C.; Wei, W.; Qu, X.; Chen, D.; Cheng, Y. Twin-Merging: Dynamic Integration of Modular Expertise in Model Merging. *arXiv preprint arXiv:2406.15479* **2024**.

Tang, A.; Shen, L.; Luo, Y.; Yin, N.; Zhang, L.; Tao, D. Merging Multi-Task Models via Weight-Ensembling Mixture of Experts. *ICML* **2024**.