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

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

[1]CSE Department, HKUST, [2]CFAR, Agency for Science, Technology and Research (A*STAR),
[3]College of Computer Science and Technology, Zhejiang University,
[4]ZJU-Hangzhou Global Scientific and Technological Innovation Center, Zhejiang University,
[5]CSE Department, Northeastern University, [6]DSA Thrust, HKUST (Guangzhou),
[7]National University of Singapore, [8]Tsinghua University, [9]UCAS, [10]Peking University,
[11]Platform and Content Group, Tencent, [12]gomore.ai, [13]Georgia Institute of Technology,
[14]XGRIDS, [15]Nanyang Technological University, [16]AI Thrust, HKUST (Guangzhou)

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.

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., 2023e). 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., 2024c; Wu et al., 2025b). 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 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., 2024b; 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., 2025b). Corresponding surveys have reviewed this literature from multiple perspectives, including LLM-based agents and long-term interaction (Zhang et al., 2025b), long-context mod-

*Corresponding author.
†Equal contribution.

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 3750–3785
November 6-10, 2026 ©2026 Association for Computational Linguistics

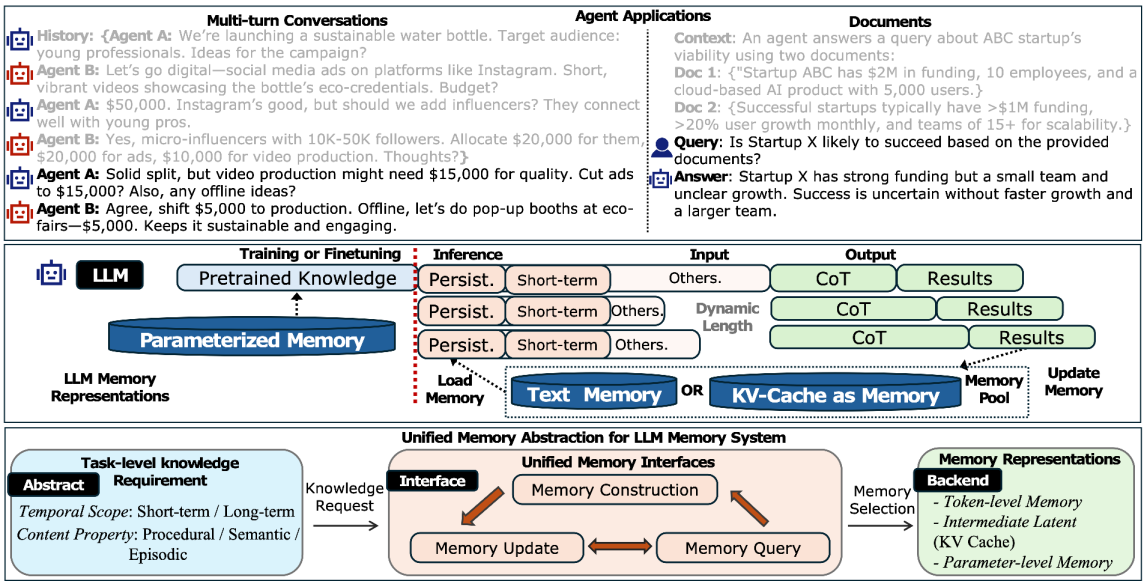


Figure 1: Overview of LLM memory applications, representation, and unified management abstraction.

eling and long-term memory (Huang et al., 2023; Jiang et al., 2024a; Wu et al., 2025b), personalization (Liu et al., 2025a), 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.

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., 2024c). 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., 2023c; 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., 2023j) and G-Retriever (He et al., 2024a) 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., 2023k), generating aligned contexts (Yu et al., 2022b), 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., 2023b; Jin et al., 2023; Wang et al., 2023g). 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., 2023l). 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., 2025c).

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 (Wang et al., 2023l; Gao et al., 2022; Langchain, 2023; Yang, 2023). Some methods also adjust *retrieval invocation*, for example, deciding when retrieval is necessary or generating retrieval-aligned contexts to stabilize generation (Wang et al., 2023k; Yu et al., 2022b; 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 (Zhou et al., 2023a; Dhuliawala et al., 2023; Ma et al., 2023; Peng et al., 2023; Gao et al., 2022), where step-back prompting (Zheng et al., 2024a) 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., 2025a). A third line develops *multi-step and controllable querying*, where retrieval and generation are composed into modular pipelines (e.g., RRR, GenRead, RECITE (Ma et al., 2023; Yu et al., 2022b; 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., 2023c; Asai et al., 2023; Ke et al., 2024; Lin et al., 2023b).

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 (Zhong et al., 2024b; Modarressi et al., 2023). 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., 2024b; 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., 2024c; 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., 2024a), 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., 2023a; Chevalier et al., 2023; Liu et al., 2023a; 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., 2024b; Wang et al., 2025d) or prompt-based extraction of salient topics as in MemoChat (Lu et al., 2023a). 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., 2024e), and workflow-level records (Wang et al., 2024m). Experience-based agents refine memory through trial-and-error interaction and feedback, revising what to store and how to use it (Liu et al., 2023b; Zhu et al., 2023b; Wang et al., 2023b; Yao et al., 2023; Zhao et al., 2024a; Li et al., 2024a). 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., 2024c; Samsami et al., 2024; Wang et al., 2024a), as well as selective editing, recursive summarization, memory blending, and self-reflective verification to maintain relevance and consistency (Bae et al., 2022; Wang et al., 2025c; Kim et al., 2024b; 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., 2023b; 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., 2024a; 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; Jang et al., 2023; Xu et al., 2021; Gutiérrez et al., 2024; Qian et al., 2024)). 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,b; Grattafiori et al., 2024; DeepSeek-AI et al., 2025a,b). 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.

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., 2024a; Han et al., 2024a), layer-wise retention (Wu and Tu, 2024), and attention- or query-aware dropping such as H_2O, FastGen, Radar, and NACL (Zhang et al., 2023a; Ge et al., 2024; Hao et al., 2025; Chen et al., 2024d). Variants further exploit attention statistics across heads, layers, or tasks (Liu et al., 2023e; Devoto et al., 2024; Yao et al., 2024; Jiang et al., 2024b; Zhong et al., 2024a; 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., 2024a; Kim et al., 2024a; Agarwal et al., 2024) and semantic-preserving compression at token, chunk, or sentence granularity (Zhang et al., 2025a; Liu et al., 2024d, 2025d; Zhu et al., 2025c).

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., 2024e; Duanmu et al., 2024; Dong et al., 2024b; Hooper et al., 2024; Zhang et al., 2024b; Li et al., 2025b), attention- or layer-aware scaling (Lin et al., 2025; Yang et al., 2024d), and low-rank compression with residual preservation (Dong et al., 2024a; Saxena et al., 2024; Kang et al., 2024). Dynamic precision schemes further adapt quantization to runtime conditions (Sheng et al., 2023; Zhao et al., 2024b; He et al., 2024b).

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., 2024c; Li et al., 2025a), layer- or chunk-level budget assignment (Yang et al., 2024a; Liu et al., 2025d), and preference- or workload-aware allocation strategies (Zhu et al., 2025b).

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., 2024b; Wu et al., 2025a), 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., 2024b). 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., 2024b; Ye et al., 2024), while cross-request reuse shares KV states based on semantic similarity (Yang et al., 2025a; Agarwal et al., 2025; Tan et al., 2025). More recent systems extend reuse across retrieval and reranking stages in RAG pipelines (Yang et al., 2025b; Yao et al., 2025; Hu et al., 2025; Zhu et al., 2025a; 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 (Khandelwal et al., 2019; Wu et al., 2022). 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 memory modules, such as CAMELoT, consolidating token representations while balancing novelty and recency (He et al., 2024c), and MemOS or Memory3, which externalize knowledge via vector memory with metadata or sparsification, without modifying core model parameters (Li et al., 2025c; Yang et al., 2024c).

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., 2023c; 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., 2024b; 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., 2024l), and larger models further amplify parametric memory capacity (Tirumala et al., 2022; Carlini et al., 2023; 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., 2024e). 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., 2024b), 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., 2024k).

Parameter-efficient fine-tuning (PEFT) updates parametric memory through lightweight adapters while keeping the backbone frozen (Han et al., 2024b). 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) (Shao et al.; Shang et al., 2024; Qian et al., 2024; Liu et al., 2025b).

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., 2024a; 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., 2024b; Lu et al., 2024a).

Model editing performs targeted updates to parametric memory by inserting, modifying, or removing specific knowledge (Wang et al., 2024h). 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., 2024j,g).

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, 2023; 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).

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

Memory Representation	Preferred Knowledge Requirements		Management Challenge			Strategic Gain
	Retention	Functional	Construction	Update	Query	(Return)
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**

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.

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., 2024c). 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).

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., 2024c), cache policies (Xiao et al., 2023; Zhang et al., 2023c), 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.

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 Conclusion

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.

Acknowledgments

We acknowledge the use of OpenAI GPT for checking grammar and assisting with the placement of figures and tables.

References

Saurabh Agarwal, Bilge Acun, Basil Hosmer, Mostafa Elhoushi, Yejin Lee, Shivaram Venkataraman, Dimitris Papailiopoulos, and Carole-Jean Wu. 2024. CHAI: Clustered head attention for efficient LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 291–312. PMLR.

Shubham Agarwal, Sai Sundaresan, Subrata Mitra, Debabrata Mahapatra, Archit Gupta, Rounak Sharma, Nirmal Joshua Kapu, Tong Yu, and Shiv Saini. 2025. Cache-craft: Managing chunk-caches for efficient retrieval-augmented generation. *Preprint*, arXiv:2502.15734.

Talkie Ai. 2024. Talkie | ai-native character community.

Arij Al Adel and Mikhail S. Burtsev. 2021. Memory transformer with hierarchical attention for long document processing. In *2021 International Conference Engineering and Telecommunication*, pages 1–7.

Zeyuan Allen-Zhu and Yuanzhi Li. 2024. Physics of language models: part 3.1, knowledge storage and extraction. In *Proceedings of the 41st International Conference on Machine Learning*, pages 1067–1077.

Yuwei An, Yihua Cheng, Seo Jin Park, and Junchen Jiang. 2025. Hyperrag: Enhancing quality-efficiency tradeoffs in retrieval-augmented generation with reranker kv-cache reuse. *Preprint*, arXiv:2504.02921.

Anysphere. 2025. Cursor - the ai code editor. https://www.cursor.com/en.

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

Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

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

Alan Baddeley. 2007. *Working memory, thought, and action*, volume 45. OuP Oxford.

Alan D. Baddeley and Graham Hitch. 1974. Working memory. volume 8 of *Psychology of Learning and Motivation*, pages 47–89. Academic Press.

Sanghwan Bae, Donghyun Kwak, Soyoung Kang, Min Young Lee, Sungdong Kim, Yuin Jeong, Hyeri Kim, Sang-Woo Lee, Woomyoung Park, and Nako Sung. 2022. Keep me updated! memory management in long-term conversations. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 3769–3787, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, and 1 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

I. Begg. 1984. Tulving's memory [review of the book elements of episodic memory, by e. tulving]. *Canadian Journal of Psychology / Revue canadienne de psychologie*, 38(1):144–147.

Iz Beltagy, Matthew E. Peters, and Arman Cohan. 2020. Longformer: The long-document transformer. *CoRR*, abs/2004.05150.

Stella Biderman, USVSN Sai Prashanth, Lintang Sutawika, Hailey Schoelkopf, Quentin Anthony, Shivanshu Purohit, and Edward Raff. 2023. Emergent and predictable memorization in large language models. *Preprint*, arXiv:2304.11158.

Rishi Bommasani, Drew A. Hudson, Ehsan Adeli, Russ Altman, Simran Arora, Sydney von Arx, Michael S. Bernstein, Jeannette Bohg, Antoine Bosselut, Emma Brunskill, Erik Brynjolfsson, Shyamal Buch, Dallas Card, Rodrigo Castellon, Niladri Chatterji, Annie Chen, Kathleen Creel, Jared Quincy Davis, Dora Demszky, and 95 others. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020a. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, and 12 others. 2020b. Language models are few-shot learners. In *Proceedings of the 34th International Conference on Neural Information Processing Systems*, NIPS '20, Red Hook, NY, USA. Curran Associates Inc.

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

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

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. 2023. Quantifying memorization across neural language models. *Preprint*, arXiv:2202.07646.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, Alina Oprea, and Colin Raffel. 2021. Extracting training data from large language models. *Preprint*, arXiv:2012.07805.

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

Dake Chen, Hanbin Wang, Yunhao Huo, Yuzhao Li, and Haoyang Zhang. 2023a. Gamegpt: Multi-agent collaborative framework for game development. *arXiv preprint arXiv:2310.08067*.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. 2024a. BGE m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *CoRR*, abs/2402.03216.

Kexin Chen, Junyou Li, Kunyi Wang, Yuyang Du, Jiahui Yu, Jiamin Lu, Lanqing Li, Jiezhong Qiu, Jianzhang Pan, Yi Huang, Qun Fang, Pheng Ann Heng, and Guangyong Chen. 2024b. Chemist-x: Large language model-empowered agent for reaction condition recommendation in chemical synthesis.

Shiyang Chen, Rain Jiang, Dezhi Yu, Jinlai Xu, Mengyuan Chao, Fanlong Meng, Chenyu Jiang, Wei Xu, and Hang Liu. 2024c. Kvdirect: Distributed disaggregated llm inference. *Preprint*, arXiv:2501.14743.

Tong Chen, Hongwei Wang, Sihao Chen, Wenhao Yu, Kaixin Ma, Xinran Zhao, Dong Yu, and Hongming Zhang. 2023b. Dense x retrieval: What retrieval granularity should we use? *arXiv preprint arXiv:2312.06648*.

Yilong Chen, Guoxia Wang, Junyuan Shang, Shiyao Cui, Zhenyu Zhang, Tingwen Liu, Shuohuan Wang, Yu Sun, Dianhai Yu, and Hua Wu. 2024d. NACL: A general and effective KV cache eviction framework for LLM at inference time. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7913–7926, Bangkok, Thailand. Association for Computational Linguistics.

Zi-Yi Chen, Fan-Kai Xie, Meng Wan, Yang Yuan, Miao Liu, Zong-Guo Wang, Sheng Meng, and Yan-Gang Wang. 2023c. Matchat: A large language model and application service platform for materials science. *Chinese Physics B*, 32(11):118104.

Xin Cheng, Di Luo, Xiuying Chen, Lemao Liu, Dongyan Zhao, and Rui Yan. 2023. Lift yourself up: Retrieval-augmented text generation with self memory. *arXiv preprint arXiv:2305.02437*.

Yuheng Cheng, Ceyao Zhang, Zhengwen Zhang, Xiangrui Meng, Sirui Hong, Wenhao Li, Zihao Wang, Zekai Wang, Feng Yin, Junhua Zhao, and et al. 2024. Exploring large language model based intelligent agents: Definitions, methods, and prospects. *arXiv*.

Alexis Chevalier, Alexander Wettig, Anirudh Ajith, and Danqi Chen. 2023. Adapting language models to compress contexts. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3829–3846, Singapore. Association for Computational Linguistics.

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

Fergus IM Craik and Robert S Lockhart. 1972. Levels of processing: A framework for memory research. *Journal of verbal learning and verbal behavior*, 11(6):671–684.

Can Cui, Yunsheng Ma, Xu Cao, Wenqian Ye, Yang Zhou, Kaizhao Liang, Jintai Chen, Juanwu Lu, Zichong Yang, Kuei-Da Liao, and et al. 2024. A survey on multimodal large language models for autonomous driving. In *Proceedings of the IEEE/CVF Winter Conference on Applications of Computer Vision*, pages 958–979.

Nico Daheim, Thomas Möllenhoff, Edoardo Ponti, Iryna Gurevych, and Mohammad Emtiyaz Khan. 2024. Model merging by uncertainty-based gradient matching. In *ICLR*.

Damai Dai, Li Dong, Yaru Hao, Zhifang Sui, Baobao Chang, and Furu Wei. 2021. Knowledge neurons in pretrained transformers. *arXiv*.

Damai Dai, Yutao Sun, Li Dong, Yaru Hao, Shuming Ma, Zhifang Sui, and Furu Wei. 2023. Why can GPT learn in-context? language models secretly perform gradient descent as meta-optimizers. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 4005–4019, Toronto, Canada. Association for Computational Linguistics.

Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. 2022. Flashattention: fast and memory-efficient exact attention with io-awareness. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Sumanth Dathathri, Andrea Madotto, Janice Lan, Jane Hung, Eric Frank, Piero Molino, Jason Yosinski, and Rosanne Liu. Plug and play language models: A simple approach to controlled text generation. In *International Conference on Learning Representations*.

MohammadReza Davari and Eugene Belilovsky. 2023. Model breadcrumbs: Scaling multi-task model merging with sparse masks. *arXiv preprint arXiv:2312.06795*.

DeepSeek-AI, Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, and 1 others. 2025a. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *Preprint*, arXiv:2501.12948.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, and 1 others. 2025b. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

Alessio Devoto, Yu Zhao, Simone Scardapane, and Pasquale Minervini. 2024. A simple and effective l_2 norm-based strategy for KV cache compression. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 18476–18499, Miami, Florida, USA. Association for Computational Linguistics.

Shehzaad Dhuliawala, Mojtaba Komeili, Jing Xu, Roberta Raileanu, Xian Li, Asli Celikyilmaz, and Jason Weston. 2023. Chain-of-verification reduces hallucination in large language models. *arXiv preprint arXiv:2309.11495*.

Shangzhe Di, Zhelun Yu, Guanghao Zhang, Haoyuan Li, TaoZhong, Hao Cheng, Bolin Li, Wanggui He, Fangxun Shu, and Hao Jiang. 2025. Streaming video question-answering with in-context video KV-cache retrieval. In *The Thirteenth International Conference on Learning Representations*.

Harry Dong, Xinyu Yang, Zhenyu Zhang, Zhangyang Wang, Yuejie Chi, and Beidi Chen. 2024a. Get more with LESS: Synthesizing recurrence with KV cache compression for efficient LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 11437–11452. PMLR.

Shichen Dong, Wen Cheng, Jiayu Qin, and Wei Wang. 2024b. Qaq: Quality adaptive quantization for llm kv cache. *Preprint*, arXiv:2403.04643.

Yiming Du, Wenyu Huang, Danna Zheng, Zhaowei Wang, Sebastien Montella, Mirella Lapata, Kam-Fai Wong, and Jeff Z. Pan. 2025. Rethinking memory in ai: Taxonomy, operations, topics, and future directions. *Preprint*, arXiv:2505.00675.

Yiming Du, Hongru Wang, Zhengyi Zhao, Bin Liang, Baojun Wang, Wanjun Zhong, Zezhong Wang, and Kam-Fai Wong. 2024. Perltqa: A personal long-term memory dataset for memory classification, retrieval, and synthesis in question answering. *Preprint*, arXiv:2402.16288.

Haojie Duanmu, Zhihang Yuan, Xiuhong Li, Jiangfei Duan, Xingcheng Zhang, and Dahua Lin. 2024. SKVQ: Sliding-window key and value cache quantization for large language models. *Preprint*, arXiv:2405.06219.

Jacob Dunefsky and Arman Cohan. 2025. Investigating generalization of one-shot llm steering vectors. *arXiv preprint arXiv:2502.18862*.

Zane Durante, Qiuyuan Huang, Naoki Wake, Ran Gong, Jae Sung Park, Bidipta Sarkar, Rohan Taori, Yusuke Noda, Demetri Terzopoulos, Yejin Choi, and et al. 2024. Agent ai: Surveying the horizons of multimodal interaction. *arXiv*.

Angela Fan, Beliz Gokkaya, Mark Harman, Mitya Lyubarskiy, Shubho Sengupta, Shin Yoo, and Jie M Zhang. 2023. Large language models for software engineering: Survey and open problems. *arXiv*.

Yongqi Fan, Hongli Sun, Kui Xue, Xiaofan Zhang, Shaoting Zhang, and Tong Ruan. 2024. Medodyssey: A medical domain benchmark for long context evaluation up to 200k tokens. *CoRR*, abs/2406.15019.

Yujie Feng, Xu Chu, Yongxin Xu, Guangyuan Shi, Bo Liu, and Xiao-Ming Wu. 2024. TaSL: Continual dialog state tracking via task skill localization and consolidation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1266–1279, Bangkok, Thailand. Association for Computational Linguistics.

Joshua Freeman, Chloe Rippe, Edoardo Debenedetti, and Maksym Andriushchenko. 2024. Exploring memorization and copyright violation in frontier LLMs: A study of the new york times v. openai 2023 lawsuit. *Preprint*, arXiv:2412.06370.

Yao Fu, Leyang Xue, Yeqi Huang, Andrei-Octavian Brabete, Dmitrii Ustiugov, Yuvraj Patel, and Luo Mai. 2024. Serverlessllm: low-latency serverless inference for large language models. In *Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation*, OSDI'24, USA. USENIX Association.

Chen Gao, Xiaochong Lan, Zhihong Lu, Jinzhu Mao, Jinghua Piao, Huandong Wang, Depeng Jin, and Yong Li. 2023. S^3: Social-network simulation system with large language model-empowered agents. *arxiv*.

Luyu Gao, Xueguang Ma, Jimmy Lin, and Jamie Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

Muhan Gao, TaiMing Lu, Kuai Yu, Adam Byerly, and Daniel Khashabi. 2024. Insights into LLM long-context failures: When transformers know but don't tell. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, Miami, Florida, USA. Association for Computational Linguistics.

Shen Gao, Xiuying Chen, Piji Li, Zhaochun Ren, Lidong Bing, Dongyan Zhao, and Rui Yan. 2019. Abstractive text summarization by incorporating reader comments. In *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*, pages 6399–6406. AAAI Press.

Shivam Garg, Dimitris Tsipras, Percy S Liang, and Gregory Valiant. 2022. What can transformers learn in-context? A case study of simple function classes. In *Advances in Neural Information Processing Systems*, volume 35, pages 30583–30598.

Suyu Ge, Yunan Zhang, Liyuan Liu, Minjia Zhang, Jiawei Han, and Jianfeng Gao. 2024. Model tells you what to discard: Adaptive KV cache compression for LLMs. In *The Twelfth International Conference on Learning Representations*.

Yingqiang Ge, Yujie Ren, Wenyue Hua, Shuyuan Xu, Juntao Tan, and Yongfeng Zhang. 2023. Llm as os (llmao), agents as apps: Envisioning aios, agents and the aios-agent ecosystem. *arXiv*.

Mor Geva, Roei Schuster, Jonathan Berant, and Omer Levy. 2021. Transformer feed-forward layers are key-value memories. *arXiv*.

In Gim, Guojun Chen, Seung-seob Lee, Nikhil Sarda, Anurag Khandelwal, and Lin Zhong. 2024. Prompt cache: Modular attention reuse for low-latency inference. *Proceedings of Machine Learning and Systems*, 6:325–338.

GitHub. 2022. Github copilot.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, and 1 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Michael Günther, Jackmin Ong, Isabelle Mohr, Alaeddine Abdessalem, Tanguy Abel, Mohammad Kalim Akram, Susana Guzman, Georgios Mastrapas, Saba Sturua, Bo Wang, Maximilian Werk, Nan Wang, and Han Xiao. 2023. Jina embeddings 2: 8192-token general-purpose text embeddings for long documents. *CoRR*, abs/2310.19923.

Mandy Guo, Joshua Ainslie, David C. Uthus, Santiago Ontañón, Jianmo Ni, Yun-Hsuan Sung, and Yinfei Yang. 2022. Longt5: Efficient text-to-text transformer for long sequences. In *Findings of the Association for Computational Linguistics: NAACL 2022, Seattle, WA, United States, July 10-15, 2022*, pages 724–736. Association for Computational Linguistics.

Taicheng Guo, Xiuying Chen, Yaqi Wang, Ruidi Chang, Shichao Pei, Nitesh V Chawla, Olaf Wiest, and Xiangliang Zhang. 2024. Large language model based multi-agents: A survey of progress and challenges. *arXiv*.

Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. 2024. Hipporag: Neurobiologically inspired long-term memory for large language models. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Michael Hahn and Navin Goyal. 2023. A theory of emergent in-context learning as implicit structure induction. *arxiv*, arXiv:2303.07971.

Chi Han, Qifan Wang, Hao Peng, Wenhan Xiong, Yu Chen, Heng Ji, and Sinong Wang. 2024a. LM-infinite: Zero-shot extreme length generalization for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3991–4008, Mexico City, Mexico. Association for Computational Linguistics.

Zeyu Han, Chao Gao, Jinyang Liu, Jeff Zhang, and Sai Qian Zhang. 2024b. Parameter-efficient fine-tuning for large models: A comprehensive survey. *arXiv preprint arXiv:2403.14608*.

Yongchang Hao, Mengyao Zhai, Hossein Hajimirsadeghi, Sepidehsadat Hosseini, and Frederick Tung. 2025. Radar: Fast long-context decoding for any transformer. In *The Thirteenth International Conference on Learning Representations*.

Kostas Hatalis, Despina Christou, Joshua Myers, Steven Jones, Keith Lambert, Adam Amos-Binks, Zohreh Dannenhauer, and Dustin Dannenhauer. 2024. Memory matters: The need to improve long-term memory in llm-agents. *Proceedings of the AAAI Symposium Series*, 2.

Kai He, Rui Mao, Qika Lin, Yucheng Ruan, Xiang Lan, Mengling Feng, and Erik Cambria. 2023a. A survey of large language models for healthcare: from data, technology, and applications to accountability and ethics. *arXiv*.

Tianyu He, Guanghui Fu, Yijing Yu, Fan Wang, Jianqiang Li, Qing Zhao, Changwei Song, Hongzhi Qi, Dan Luo, Huijing Zou, and et al. 2023b. Towards a psychological generalist ai: A survey of current applications of large language models and future prospects. *arXiv*.

Xiaoxin He, Yijun Tian, Yifei Sun, Nitesh V Chawla, Thomas Laurent, Yann LeCun, Xavier Bresson, and Bryan Hooi. 2024a. G-retriever: Retrieval-augmented generation for textual graph understanding and question answering. *arXiv preprint arXiv:2402.07630*.

Yefei He, Luoming Zhang, Weijia Wu, Jing Liu, Hong Zhou, and Bohan Zhuang. 2024b. Zipcache: Accurate and efficient kv cache quantization with salient token identification. In *Advances in Neural Information Processing Systems*, volume 37, pages 68287–68307. Curran Associates, Inc.

Zexue He, Leonid Karlinsky, Donghyun Kim, Julian McAuley, Dmitry Krotov, and Rogerio Feris. 2024c. Camelot: Towards large language models with training-free consolidated associative memory. *arXiv preprint arXiv:2402.13449*.

Zihong He, Weizhe Lin, Hao Zheng, Fan Zhang, Matt W. Jones, Laurence Aitchison, Xuhai Xu, Miao Liu, Per Ola Kristensson, and Junxiao Shen. 2024d. Human-inspired perspectives: A survey on ai long-term memory. *arXiv preprint arXiv:2411.00489*.

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

Christian Herold and Hermann Ney. 2023. Improving long context document-level machine translation. *CoRR*, abs/2306.05183.

Lukas Hilgert, Danni Liu, and Jan Niehues. 2024. Evaluating and training long-context large language models for question answering on scientific papers. In *Proceedings of the 1st Workshop on Customizable NLP: Progress and Challenges in Customizing NLP for a Domain, Application, Group, or Individual (CustomNLP4U)*, pages 220–236, Miami, Florida, USA. Association for Computational Linguistics.

Coleman Hooper, Sehoon Kim, Hiva Mohammadzadeh, Michael W Mahoney, Sophia Shao, Kurt Keutzer, and Amir Gholami. 2024. KVQuant: Towards 10 million context length LLM inference with KV cache quantization. *Advances in Neural Information Processing Systems, NeurIPS 2024*, 37:1270–1303.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Junhao Hu, Wenrui Huang, Weidong Wang, Haoyi Wang, Tiancheng Hu, Qin Zhang, Hao Feng, Xusheng Chen, Yizhou Shan, and Tao Xie. 2025. Epic: Efficient position-independent caching for serving large language models. *Preprint*, arXiv:2410.15332.

Wenyue Hua, Lizhou Fan, Lingyao Li, Kai Mei, Jianchao Ji, Yingqiang Ge, Libby Hemphill, and Yongfeng Zhang. 2023. War and peace (waragent): Large language model-based multi-agent simulation of world wars. *arXiv preprint arXiv:2311.17227*.

Xu Huang, Weiwen Liu, Xiaolong Chen, Xingmei Wang, Hao Wang, Defu Lian, Yasheng Wang, Ruiming Tang, and Enhong Chen. 2024. Understanding the planning of llm agents: A survey. *arXiv*.

Yunpeng Huang, Jingwei Xu, Junyu Lai, Zixu Jiang, Taolue Chen, Zenan Li, Yuan Yao, Xiaoxing Ma, Lijuan Yang, Hao Chen, and 1 others. 2023. Advancing transformer architecture in long-context large language models: A comprehensive survey. *arXiv preprint arXiv:2311.12351*.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Kai Dang, An Yang, Rui Men, Fei Huang, Xingzhang Ren, Xuancheng Ren, Jingren Zhou, and Junyang Lin. 2024. Qwen2.5-coder technical report. *CoRR*, abs/2409.12186.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic.

Inflection. 2023. I'm pi, your personal ai. `https://inflection.ai/`.

Daphne Ippolito, Florian Tramer, Milad Nasr, Chiyuan Zhang, Matthew Jagielski, Katherine Lee, Christopher Choquette Choo, and Nicholas Carlini. 2023. Preventing generation of verbatim memorization in language models gives a false sense of privacy. In *Proceedings of the 16th International Natural Language Generation Conference*, pages 28–53, Prague, Czechia. Association for Computational Linguistics.

Kai Tzu iunn Ong, Namyoung Kim, Minju Gwak, Hyungjoo Chae, Taeyoon Kwon, Yohan Jo, Seung won Hwang, Dongha Lee, and Jinyoung Yeo. 2025. Towards lifelong dialogue agents via timeline-based memory management. In *Proceedings of the 2025 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, Mexico City, Mexico. Association for Computational Linguistics.

Gautier Izacard, Patrick Lewis, Maria Lomeli, Lucas Hosseini, Fabio Petroni, Timo Schick, Jane Dwivedi-Yu, Armand Joulin, Sebastian Riedel, and Edouard Grave. 2023. Atlas: few-shot learning with retrieval augmented language models. *J. Mach. Learn. Res.*, 24(1).

Jihyoung Jang, Minseong Boo, and Hyounghun Kim. 2023. Conversation chronicles: Towards diverse temporal and relational dynamics in multi-session conversations. *arXiv preprint arXiv:2310.13420*.

Yunah Jang, Kang-il Lee, Hyunkyung Bae, Hwanhee Lee, and Kyomin Jung. 2024. IterCQR: Iterative conversational query reformulation with retrieval guidance. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 8121–8138, Mexico City, Mexico. Association for Computational Linguistics.

Divyansh Jhunjhunwala, Shiqiang Wang, and Gauri Joshi. 2024. Fedfisher: Leveraging fisher information for one-shot federated learning. In *AISTATS*, pages 1612–1620. PMLR.

Huiqiang Jiang, Qianhui Wu, Chin-Yew Lin, Yuqing Yang, and Lili Qiu. 2023a. LLMLingua: Compressing prompts for accelerated inference of large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 13358–13376, Singapore. Association for Computational Linguistics.

Wenqi Jiang, Suvinay Subramanian, Cat Graves, Gustavo Alonso, Amir Yazdanbakhsh, and Vidushi Dadu. 2025. Rago: Systematic performance optimization for retrieval-augmented generation serving. *Preprint*, arXiv:2503.14649.

Xun Jiang, Feng Li, Han Zhao, Jiaying Wang, Jun Shao, Shihao Xu, Shu Zhang, Weiling Chen, Xavier Tang, Yize Chen, and et al. 2024a. Long term memory: The foundation of ai self-evolution. *arXiv*.

Yikun Jiang, Huanyu Wang, Lei Xie, Hanbin Zhao, Chao Zhang, Hui Qian, and John C.S. Lui. 2024b. D-llm: A token adaptive computing resource allocation strategy for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 1725–1749. Curran Associates, Inc.

Zhengbao Jiang, Frank Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023b. Active retrieval augmented generation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 7969–7992, Singapore. Association for Computational Linguistics.

Zhengbao Jiang, Frank F Xu, Luyu Gao, Zhiqing Sun, Qian Liu, Jane Dwivedi-Yu, Yiming Yang, Jamie Callan, and Graham Neubig. 2023c. Active retrieval augmented generation. *arXiv preprint arXiv:2305.06983*.

Bowen Jin, Jinsung Yoon, Jiawei Han, and Sercan Ö. Arik. 2024a. Long-context llms meet RAG: overcoming challenges for long inputs in RAG. *CoRR*, abs/2410.05983.

Bowen Jin, Hansi Zeng, Guoyin Wang, Xiusi Chen, Tianxin Wei, Ruirui Li, Zhengyang Wang, Zheng Li, Yang Li, Hanqing Lu, and 1 others. 2023. Language models as semantic indexers. *arXiv preprint arXiv:2310.07815*.

Hanlei Jin, Yang Zhang, Dan Meng, Jun Wang, and Jinghua Tan. 2024b. A comprehensive survey on process-oriented automatic text summarization with exploration of llm-based methods. *CoRR*, abs/2403.02901.

Hongpeng Jin and Yanzhao Wu. 2025. Ce-collm: Efficient and adaptive large language models through cloud-edge collaboration. In *2025 IEEE International Conference on Web Services (ICWS)*, pages 316–323.

Philip Nicholas Johnson-Laird. 1983. *Mental models: Towards a cognitive science of language, inference, and consciousness*, volume 6. Harvard University Press.

Norman P. Jouppi, Cliff Young, Nishant Patil, David Patterson, Gaurav Agrawal, Raminder Bajwa, Sarah Bates, Suresh Bhatia, Nan Boden, Al Borchers, Rick Boyle, Pierre-luc Cantin, Clifford Chao, Chris Clark, Jeremy Coriell, Mike Daley, Matt Dau, Jeffrey Dean, Ben Gelb, and 57 others. 2017. In-datacenter performance analysis of a tensor processing unit. *SIGARCH Comput. Archit. News*, 45(2):1–12.

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

Zhao Kaiya, Michelangelo Naim, Jovana Kondic, Manuel Cortes, Jiaxin Ge, Shuying Luo, Guangyu Robert Yang, and Andrew Ahn. 2023. Lyfe agents: Generative agents for low-cost real-time social interactions. *arXiv*.

Nikhil Kandpal, Eric Wallace, and Colin Raffel. 2022. Deduplicating training data mitigates privacy risks in language models. *Preprint*, arXiv:2202.06539.

Hao Kang, Qingru Zhang, Souvik Kundu, Geonhwa Jeong, Zaoxing Liu, Tushar Krishna, and Tuo Zhao. 2024. Gear: An efficient kv cache compression recipe for near-lossless generative inference of llm. *Preprint*, arXiv:2403.05527.

Sayash Kapoor, Peter Henderson, and Arvind Narayanan. 2024. Promises and pitfalls of artificial intelligence for legal applications. *CoRR*, abs/2402.01656.

Vladimir Karpukhin, Barlas Oguz, Sewon Min, Patrick SH Lewis, Ledell Wu, Sergey Edunov, Danqi Chen, and Wen-tau Yih. 2020. Dense passage retrieval for open-domain question answering. In *EMNLP (1)*, pages 6769–6781.

Zixuan Ke, Weize Kong, Cheng Li, Mingyang Zhang, Qiaozhu Mei, and Michael Bendersky. 2024. Bridging the preference gap between retrievers and llms. *arXiv preprint arXiv:2401.06954*.

Urvashi Khandelwal, Omer Levy, Dan Jurafsky, Luke Zettlemoyer, and Mike Lewis. 2019. Generalization through memorization: Nearest neighbor language models. *arXiv preprint arXiv:1911.00172*.

Omar Khattab, Keshav Santhanam, Xiang Lisa Li, David Hall, Percy Liang, Christopher Potts, and Matei Zaharia. 2022. Demonstrate-search-predict: Composing retrieval and language models for knowledge-intensive nlp. *arXiv preprint arXiv:2212.14024*.

Omar Khattab and Matei Zaharia. 2020. Colbert: Efficient and effective passage search via contextualized late interaction over bert. In *Proceedings of the 43rd International ACM SIGIR Conference on Research and Development in Information Retrieval*, SIGIR '20, page 39–48, New York, NY, USA. Association for Computing Machinery.

Minsoo Kim, Kyuhong Shim, Jungwook Choi, and Simyung Chang. 2024a. InfiniPot: Infinite context processing on memory-constrained LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 16046–16060, Miami, Florida, USA. Association for Computational Linguistics.

Seo Hyun Kim, Keummin Ka, Yohan Jo, Seung-won Hwang, Dongha Lee, and Jinyoung Yeo. 2024b. Ever-evolving memory by blending and refining the past. *arXiv preprint arXiv:2403.04787*.

James Kirkpatrick, Razvan Pascanu, Neil Rabinowitz, Joel Veness, Guillaume Desjardins, Andrei A Rusu, Kieran Milan, John Quan, Tiago Ramalho, Agnieszka Grabska-Barwinska, and 1 others. 2017. Overcoming catastrophic forgetting in neural networks. *Proceedings of the national academy of sciences*, 114(13):3521–3526.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. 2023. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th Symposium on Operating Systems Principles*, SOSP '23, page 611–626, New York, NY, USA. Association for Computing Machinery.

John E Laird. 2019. *The Soar cognitive architecture*. MIT press.

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

Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, and Kangwook Lee. 2023a. Prompted llms as chatbot modules for long open-domain conversation. In *Findings of the Association for Computational Linguistics: ACL 2023, Toronto, Canada, July 9-14, 2023*, pages 4536–4554. Association for Computational Linguistics.

Gibbeum Lee, Volker Hartmann, Jongho Park, Dimitris Papailiopoulos, and Kangwook Lee. 2023b. Prompted llms as chatbot modules for long open-domain conversation. *arXiv preprint arXiv:2305.04533*.

Katherine Lee, Daphne Ippolito, Andrew Nystrom, Chiyuan Zhang, Douglas Eck, Chris Callison-Burch, and Nicholas Carlini. 2022. Deduplicating training data makes language models better. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. 2019. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2025. Selective attention improves transformer. In *The Thirteenth International Conference on Learning Representations*.

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

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

Hao Li, Chenghao Yang, An Zhang, Yang Deng, Xiang Wang, and Tat-Seng Chua. 2024a. Hello again! llm-powered personalized agent for long-term dialogue. *arXiv preprint arXiv:2406.05925*.

Haoyang LI, Yiming Li, Anxin Tian, Tianhao Tang, Zhanchao Xu, Xuejia Chen, Nicole HU, Wei Dong, Li Qing, and Lei Chen. 2025. A survey on large language model acceleration based on KV cache management. *Transactions on Machine Learning Research*.

K. Li, O. Patel, F. Viégas, H. Pfister, and M. Wattenberg. 2024b. Inference-time intervention: Eliciting truthful answers from a language model. In *Advances in Neural Information Processing Systems*, volume 36.

Lei Li, Yongfeng Zhang, Dugang Liu, and Li Chen. 2023a. Large language models for generative recommendation: A survey and visionary discussions. *arXiv*.

Nian Li, Chen Gao, Yong Li, and Qingmin Liao. 2023b. Large language model-empowered agents for simulating macroeconomic activities. *arXiv*.

Shilong Li, Yancheng He, Hangyu Guo, Xingyuan Bu, Ge Bai, Jie Liu, Jiaheng Liu, Xingwei Qu, Yangguang Li, Wanli Ouyang, Wenbo Su, and Bo Zheng. 2024c. GraphReader: Building graph-based agent to enhance long-context abilities of large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 12758–12786, Miami, Florida, USA. Association for Computational Linguistics.

Weiqing Li, Guochao Jiang, Xiangyong Ding, Zhangcheng Tao, Chuzhan Hao, Chenfeng Xu, Yuewei Zhang, and Hao Wang. 2025a. Flowkv: A disaggregated inference framework with low-latency kv cache transfer and load-aware scheduling. *Preprint*, arXiv:2504.03775.

Xianming Li and Jing Li. 2023. Angle-optimized text embeddings. *arXiv preprint arXiv:2309.12871*.

Xiaoqian Li, Ercong Nie, and Sheng Liang. 2023c. From classification to generation: Insights into crosslingual retrieval augmented icl. *arXiv preprint arXiv:2311.06595*.

Yinheng Li, Shaofei Wang, Han Ding, and Hang Chen. 2023d. Large language models in finance: A survey. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 374–382.

Yuan Li, Yixuan Zhang, and Lichao Sun. 2023e. Metaagents: Simulating interactions of human behaviors for llm-based task-oriented coordination via collaborative generative agents. *arXiv preprint arXiv:2310.06500*.

Yuanchun Li, Hao Wen, Weijun Wang, Xiangyu Li, Yizhen Yuan, Guohong Liu, Jiacheng Liu, Wenxing Xu, Xiang Wang, Yi Sun, and et al. 2024d. Personal llm agents: Insights and survey about the capability, efficiency and security. *arXiv*.

Yuhong Li, Yingbing Huang, Bowen Yang, Bharat Venkitesh, Acyr Locatelli, Hanchen Ye, Tianle Cai, Patrick Lewis, and Deming Chen. 2024e. Snapkv: Llm knows what you are looking for before generation. *arXiv preprint arXiv:2404.14469*.

Zeyu Li, Chuanfu Xiao, Yang Wang, Xiang Liu, Zhenheng Tang, Baotong Lu, Mao Yang, Xinyu Chen, and Xiaowen Chu. 2025b. Antkv: Anchor token-aware sub-bit vector quantization for kv cache in large language models. *Preprint*, arXiv:2506.19505.

Zhiyu Li, Shichao Song, Chenyang Xi, Hanyu Wang, Chen Tang, Simin Niu, Ding Chen, Jiawei Yang, Chunyu Li, Qingchen Yu, and 1 others. 2025c. Memos: A memory os for ai system. *arXiv preprint arXiv:2507.03724*.

Jianghao Lin, Xinyi Dai, Yunjia Xi, Weiwen Liu, Bo Chen, Xiangyang Li, Chenxu Zhu, Huifeng Guo, Yong Yu, Ruiming Tang, and et al. 2023a. How can recommender systems benefit from large language models: A survey. *arXiv*.

Xi Victoria Lin, Xilun Chen, Mingda Chen, Weijia Shi, Maria Lomeli, Rich James, Pedro Rodriguez, Jacob Kahn, Gergely Szilvasy, Mike Lewis, and 1 others. 2023b. Ra-dit: Retrieval-augmented dual instruction tuning. *arXiv preprint arXiv:2310.01352*.

Yujun Lin, Haotian Tang, Shang Yang, Zhekai Zhang, Guangxuan Xiao, Chuang Gan, and Song Han. 2025. Qserve: W4a8kv4 quantization and system co-design for efficient llm serving. *Preprint*, arXiv:2405.04532.

Akide Liu, Jing Liu, Zizheng Pan, Yefei He, Gholamreza Haffari, and Bohan Zhuang. 2024a. Minicache: Kv cache compression in depth dimension for large language models. In *Advances in Neural Information Processing Systems*, volume 37, pages 139997–140031. Curran Associates, Inc.

Di Liu, Meng Chen, Baotong Lu, Huiqiang Jiang, Zhenhua Han, Qianxi Zhang, Qi Chen, Chengruidong Zhang, Bailu Ding, Kai Zhang, Chen Chen, Fan Yang, Yuqing Yang, and Lili Qiu. 2024b. Retrievalattention: Accelerating long-context llm inference via vector retrieval. *Preprint*, arXiv:2409.10516.

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

Junyi Liu, Liangzhi Li, Tong Xiang, Bowen Wang, and Yiming Qian. 2023a. TCRA-LLM: Token compression retrieval augmented large language model for inference cost reduction. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 9796–9810, Singapore. Association for Computational Linguistics.

Lei Liu, Xiaoyan Yang, Yue Shen, Binbin Hu, Zhiqiang Zhang, Jinjie Gu, and Guannan Zhang. 2023b. Think-in-memory: Recalling and post-thinking enable llms with long-term memory. *arXiv preprint arXiv:2311.08719*.

Nelson F Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. 2024c. Lost in the middle: How language models use long contexts. *Transactions of the Association for Computational Linguistics*, 12:157–173.

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

WenTao Liu, Ruohua Zhang, Aimin Zhou, Feng Gao, and JiaLi Liu. 2025b. Echo: A large language model with temporal episodic memory. *arXiv preprint arXiv:2502.16090*.

Xiang Liu, Hong Chen, Xuming Hu, and Xiaowen Chu. 2025c. FlowKV: Enhancing multi-turn conversational coherence in LLMs via isolated key-value cache management. In *First Workshop on Multi-Turn Interactions in Large Language Models*.

Xiang Liu, Zhenheng Tang, Peijie Dong, Zeyu Li, Yue Liu, Bo Li, Xuming Hu, and Xiaowen Chu. 2025d. Chunkkv: Semantic-preserving kv cache compression for efficient long-context llm inference. *Preprint*, arXiv:2502.00299.

Yuhan Liu, Hanchen Li, Yihua Cheng, Siddhant Ray, Yuyang Huang, Qizheng Zhang, Kuntai Du, Jiayi Yao, Shan Lu, Ganesh Ananthanarayanan, Michael Maire, Henry Hoffmann, Ari Holtzman, and Junchen Jiang. 2024d. Cachegen: Kv cache compression and streaming for fast large language model serving. *Preprint*, arXiv:2310.07240.

Zhengliang Liu, Aoxiao Zhong, Yiwei Li, Longtao Yang, Chao Ju, Zihao Wu, Chong Ma, Peng Shu, Cheng Chen, Sekeun Kim, and et al. 2023d. Radiology-gpt: A large language model for radiology. *arXiv preprint arXiv:2306.08666*.

Zichang Liu, Aditya Desai, Fangshuo Liao, Weitao Wang, Victor Xie, Zhaozhuo Xu, Anastasios Kyrillidis, and Anshumali Shrivastava. 2023e. Scissorhands: Exploiting the persistence of importance hypothesis for LLM KV cache compression at test

time. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zirui Liu, Jiayi Yuan, Hongye Jin, Shaochen Zhong, Zhaozhuo Xu, Vladimir Braverman, Beidi Chen, and Xia Hu. 2024e. KIVI: A tuning-free asymmetric 2bit quantization for KV cache. In *International Conference on Machine Learning, ICML 2024*, pages 32332–32344. PMLR.

Anton Lozhkov, Raymond Li, Loubna Ben Allal, Federico Cassano, Joel Lamy-Poirier, Nouamane Tazi, Ao Tang, Dmytro Pykhtar, Jiawei Liu, Yuxiang Wei, Tianyang Liu, Max Tian, Denis Kocetkov, Arthur Zucker, Younes Belkada, Zijian Wang, Qian Liu, Dmitry Abulkhanov, Indraneil Paul, and 38 others. 2024. Starcoder 2 and the stack v2: The next generation. *CoRR*, abs/2402.19173.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023a. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Junru Lu, Siyu An, Mingbao Lin, Gabriele Pergola, Yulan He, Di Yin, Xing Sun, and Yunsheng Wu. 2023b. Memochat: Tuning llms to use memos for consistent long-range open-domain conversation. *arXiv preprint arXiv:2308.08239*.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024a. Twin-merging: Dynamic integration of modular expertise in model merging. *NIPS*.

Zhenyi Lu, Chenghao Fan, Wei Wei, Xiaoye Qu, Dangyang Chen, and Yu Cheng. 2024b. Twin-merging: Dynamic integration of modular expertise in model merging. *arXiv preprint arXiv:2406.15479*.

Ziyang Luo, Can Xu, Pu Zhao, Xiubo Geng, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. 2023. Augmented large language models with parametric knowledge guiding. *arXiv preprint arXiv:2305.04757*.

Shi Luohe, Hongyi Zhang, Yao Yao, Zuchao Li, and 1 others. Keep the cost down: A review on methods to optimize llm's kv-cache consumption. In *First Conference on Language Modeling*.

Chenyang Lyu, Zefeng Du, Jitao Xu, Yitao Duan, Minghao Wu, Teresa Lynn, Alham Fikri Aji, Derek F. Wong, and Longyue Wang. 2024. A paradigm shift: The future of machine translation lies with large language models. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation, LREC/COLING 2024, 20-25 May, 2024, Torino, Italy*, pages 1339–1352. ELRA and ICCL.

Xinbei Ma, Yeyun Gong, Pengcheng He, Hai Zhao, and Nan Duan. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

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

Adyasha Maharana, Dong-Ho Lee, Sergey Tulyakov, Mohit Bansal, Francesco Barbieri, and Yuwei Fang. 2024. Evaluating very long-term conversational memory of llm agents. *arXiv preprint arXiv:2402.17753*.

Daniel Marczak, Bartłomiej Twardowski, Tomasz Trzciński, and Sebastian Cygert. 2024. Magmax: Leveraging model merging for seamless continual learning. In *ECCV*.

Ahmed Masry and Amir Hajian. 2024. Longfin: A multimodal document understanding model for long financial domain documents. *CoRR*, abs/2401.15050.

Michael S Matena and Colin A Raffel. 2022. Merging models with fisher-weighted averaging. *NeurIPS*, 35:17703–17716.

Brendan McMahan, Eider Moore, Daniel Ramage, Seth Hampson, and Blaise Aguera y Arcas. 2017. Communication-efficient learning of deep networks from decentralized data. In *AISTATS*, pages 1273–1282. PMLR.

Sanket Vaibhav Mehta, Jai Gupta, Yi Tay, Mostafa Dehghani, Vinh Q Tran, Jinfeng Rao, Marc Najork, Emma Strubell, and Donald Metzler. 2022. Dsi++: Updating transformer memory with new documents. *arXiv preprint arXiv:2212.09744*.

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

Fanxu Meng, Pingzhi Tang, Xiaojuan Tang, Zengwei Yao, Xing Sun, and Muhan Zhang. 2025. Transmla: Multi-head latent attention is all you need. *Preprint*, arXiv:2502.07864.

Kevin Meng, Arnab Sen Sharma, Alex J Andonian, Yonatan Belinkov, and David Bau. 2023. Mass-editing memory in a transformer. In *The Eleventh International Conference on Learning Representations*.

Sewon Min, Xinxi Lyu, Ari Holtzman, Mikel Artetxe, Mike Lewis, Hannaneh Hajishirzi, and Luke Zettlemoyer. 2022. Rethinking the role of demonstrations: What makes in-context learning work? In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 11048–11064, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

Mayank Mishra, Matt Stallone, Gaoyuan Zhang, Yikang Shen, Aditya Prasad, Adriana Meza Soria, Michele Merler, Parameswaran Selvam, Saptha Surendran, Shivdeep Singh, Manish Sethi, Xuan-Hong Dang, Pengyuan Li, Kun-Lung Wu, Syed Zawad, Andrew Coleman, Matthew White, Mark Lewis, Raju Pavuluri, and 27 others. 2024. Granite code models: A family of open foundation models for code intelligence. *CoRR*, abs/2405.04324.

Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. 2023. Ret-llm: Towards a general read-write memory for large language models. *arxiv*.

Ali Montazeralghaem, Hamed Zamani, and James Allan. 2020. A reinforcement learning framework for relevance feedback. In *Proceedings of the 43rd international acm sigir conference on research and development in information retrieval*, pages 59–68.

Mohammed Muqeeth, Haokun Liu, and Colin Raffel. 2024. Soft merging of experts with adaptive routing. *TMLR*.

Jaap MJ Murre and Joeri Dros. 2015. Replication and analysis of ebbinghaus' forgetting curve. *PloS one*, 10(7):e0120644.

Milad Nasr, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Eric Wallace, Florian Tramèr, and Katherine Lee. 2023. Scalable extraction of training data from (production) language models. *Preprint*, arXiv:2311.17035.

Yuqi Nie, Yaxuan Kong, Xiaowen Dong, John M. Mulvey, H. Vincent Poor, Qingsong Wen, and Stefan Zohren. 2024. A survey of large language models for financial applications: Progress, prospects and challenges. *CoRR*, abs/2406.11903.

Yuxiang Nie, Heyan Huang, Wei Wei, and Xian-Ling Mao. 2022. Capturing global structural information in long document question answering with compressive graph selector network. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 5036–5047, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

OpenAI. 2024a. Memory and new controls for chatgpt. Accessed: 2024-02-13.

OpenAI. 2024b. New embedding models and api updates. Accessed: 2024-01-25.

Charles Packer, Sarah Wooders, Kevin Lin, Vivian Fang, Shishir G Patil, Ion Stoica, and Joseph E Gonzalez. 2023. Memgpt: Towards llms as operating systems. *arXiv preprint arXiv:2310.08560*.

Haojie Pan, Zepeng Zhai, Hao Yuan, Yaojia Lv, Ruiji Fu, Ming Liu, Zhongyuan Wang, and Bing Qin. 2023a. Kwaiagents: Generalized information-seeking agent system with large language models. *arXiv*.

James Pan and Guoliang Li. 2025. A survey of llm inference systems. *Preprint*, arXiv:2506.21901.

Jane Pan, Tianyu Gao, Howard Chen, and Danqi Chen. 2023b. What in-context learning "learns" in-context: Disentangling task recognition and task learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8298–8319, Toronto, Canada. Association for Computational Linguistics.

Zhuoshi Pan, Qianhui Wu, Huiqiang Jiang, Xufang Luo, Hao Cheng, Dongsheng Li, Yuqing Yang, Chin-Yew Lin, H Vicky Zhao, Lili Qiu, and et al. 2025. On memory construction and retrieval for personalized conversational agents. *arXiv preprint arXiv:2502.05589*.

Joon Sung Park, Joseph O'Brien, Carrie Jun Cai, Meredith Ringel Morris, Percy Liang, and Michael S Bernstein. 2023. Generative agents: Interactive simulacra of human behavior. In *Proceedings of the 36th Annual ACM Symposium on User Interface Software and Technology*, pages 1–22.

Wenjun Peng, Guiyang Li, Yue Jiang, Zilong Wang, Dan Ou, Xiaoyi Zeng, Enhong Chen, and 1 others. 2023. Large language model based long-tail query rewriting in taobao search. *arXiv preprint arXiv:2311.03758*.

Reiner Pope, Sholto Douglas, Aakanksha Chowdhery, Jacob Devlin, James Bradbury, Anselm Levskaya, Jonathan Heek, Kefan Xiao, Shivani Agrawal, and Jeff Dean. 2022. Efficiently scaling transformer inference. *Preprint*, arXiv:2211.05102.

Chen Qian, Xin Cong, Cheng Yang, Weize Chen, Yusheng Su, Juyuan Xu, Zhiyuan Liu, and Maosong Sun. 2023. Communicative agents for software development. *arXiv preprint arXiv:2307.07924*.

Hongjin Qian, Peitian Zhang, Zheng Liu, Kelong Mao, and Zhicheng Dou. 2024. Memorag: Moving towards next-gen rag via memory-inspired knowledge discovery. *arXiv preprint arXiv:2409.05591*.

Zhangcheng Qiang, Weiqing Wang, and Kerry Taylor. 2023. Agent-om: Leveraging large language models for ontology matching. *arXiv*.

Zhe Qu, Xingyu Li, Rui Duan, Yao Liu, Bo Tang, and Zhuo Lu. 2022. Generalized federated learning via sharpness aware minimization. In *International Conference on Machine Learning*, pages 18250–18280. PMLR.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, and 1 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Allan Raventos, Mansheej Paul, Feng Chen, and Surya Ganguli. 2023. Pretraining task diversity and the emergence of non-bayesian in-context learning for regression. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Varshini Reddy, Rik Koncel-Kedziorski, Viet Dac Lai, Michael Krumdick, Charles Lovering, and Chris Tanner. 2024. Docfinqa: A long-context financial reasoning dataset. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics, ACL 2024 - Short Papers, Bangkok, Thailand, August 11-16, 2024*, pages 445–458. Association for Computational Linguistics.

Jon Saad-Falcon, Daniel Y. Fu, Simran Arora, Neel Guha, and Christopher Ré. 2024. Benchmarking and building long-context retrieval models with loco and M2-BERT. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*. OpenReview.net.

Ali Safaya and Deniz Yuret. 2024. Neurocache: Efficient vector retrieval for long-range language modeling. *arXiv preprint arXiv:2407.02486*.

Rana Salama, Jason Cai, Michelle Yuan, Anna Currey, Monica Sunkara, Yi Zhang, and Yassine Benajiba. 2025. Meminsight: Autonomous memory augmentation for llm agents. *arXiv preprint arXiv:2503.21760*.

Mohammad Reza Samsami, Artem Zholus, Janarthanan Rajendran, and Sarath Chandar. 2024. Mastering memory tasks with world models. In *The Twelfth International Conference on Learning Representations*.

Utkarsh Saxena, Gobinda Saha, Sakshi Choudhary, and Kaushik Roy. 2024. Eigen attention: Attention in low-rank space for KV cache compression. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 15332–15344, Miami, Florida, USA. Association for Computational Linguistics.

Lianlei Shan, Shixian Luo, Zezhou Zhu, Yu Yuan, and Yong Wu. 2025. Cognitive memory in large language models. *arXiv preprint arXiv:2504.02441*.

Jingbo Shang, Zai Zheng, Jiale Wei, Xiang Ying, Felix Tao, and Mindverse Team. 2024. Ai-native memory: A pathway from llms towards agi. *arXiv preprint arXiv:2406.18312*.

Bin Shao and Jiawei Yan. 2024. A long-context language model for deciphering and generating bacteriophage genomes. *Nature Communications*, 15(1):9392.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. Character-llm: A trainable agent for role-playing.

Yunfan Shao, Linyang Li, Junqi Dai, and Xipeng Qiu. 2023. Character-llm: A trainable agent for role-playing. *arXiv preprint arXiv:2310.10158*.

Noam Shazeer, Azalia Mirhoseini, Krzysztof Maziarz, Andy Davis, Quoc Le, Geoffrey Hinton, and Jeff Dean. 2017. Outrageously large neural networks: The sparsely-gated mixture-of-experts layer.

Lingfeng Shen, Aayush Mishra, and Daniel Khashabi. 2024. Do pretrained transformers learn in-context by gradient descent? *arxiv*, arXiv:2310.08540.

Ying Sheng, Lianmin Zheng, Binhang Yuan, Zhuohan Li, Max Ryabinin, Beidi Chen, Percy Liang, Christopher Ré, Ion Stoica, and Ce Zhang. 2023. Flexgen: high-throughput generative inference of large language models with a single gpu. In *Proceedings of the 40th International Conference on Machine Learning*, ICML'23. JMLR.org.

Lauralee Sherwood, Robert Thomas Kell, and Christopher Ward. 2004. *Human physiology: from cells to systems*. Thomson/Brooks/Cole.

Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*, pages 31210–31227. PMLR.

Kaize Shi, Xueyao Sun, Qing Li, and Guandong Xu. 2024. Compressing long context for enhancing RAG with amr-based concept distillation. *CoRR*, abs/2405.03085.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. 2024. Reflexion: Language agents with verbal reinforcement learning. *Advances in Neural Information Processing Systems*, 36.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik R. Narasimhan, and Shunyu Yao. 2023. Reflexion: Language agents with verbal reinforcement learning. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Haseeb Ullah Khan Shinwari and Muhammad Usama. 2025. Memory-augmented architecture for long-term context handling in large language models. *arXiv preprint arXiv:2506.18271*.

Aaditya K Singh, Stephanie C.Y. Chan, Ted Moskovitz, Erin Grant, Andrew M Saxe, and Felix Hill. 2023. The transient nature of emergent in-context learning in transformers. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Robert L Solso and Jerome Kagan. 1979. *Cognitive psychology*. Houghton Mifflin Harcourt P.

L. R. Squire. 2009. 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*, 29(41):12711–12716.

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

N. Subramani, N. Suresh, and M. E. Peters. 2022. Extracting latent steering vectors from pretrained language models. *arXiv*.

Hao Sun, Hengyi Cai, Bo Wang, Yingyan Hou, Xiaochi Wei, Shuaiqiang Wang, Yan Zhang, and Dawei Yin. 2024. Towards verifiable text generation with evolving memory and self-reflection. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 8211–8227, Miami, Florida, USA. Association for Computational Linguistics.

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

Zhiqing Sun, Xuezhi Wang, Yi Tay, Yiming Yang, and Denny Zhou. 2022. Recitation-augmented language models. *arXiv preprint arXiv:2210.01296*.

Richard S Sutton and Andrew G Barto. 2018. *Reinforcement learning: An introduction*. MIT press.

Xin Tan, Yimin Jiang, Yitao Yang, and Hong Xu. 2025. Teola: Towards end-to-end optimization of llm-based applications. *Preprint*, arXiv:2407.00326.

Anke Tang, Li Shen, Yong Luo, Nan Yin, Lefei Zhang, and Dacheng Tao. 2024a. Merging multi-task models via weight-ensembling mixture of experts. *ICML*.

Jiaming Tang, Yilong Zhao, Kan Zhu, Guangxuan Xiao, Baris Kasikci, and Song Han. 2024b. QUEST: Query-aware sparsity for efficient long-context LLM inference. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 47901–47911. PMLR.

Ravi Teja. 2023. 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.

Kushal Tirumala, Aram H. Markosyan, Luke Zettlemoyer, and Armen Aghajanyan. 2022. Memorization without overfitting: Analyzing the training dynamics of large language models. *Preprint*, arXiv:2205.10770.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, and 1 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

YunDa Tsai, Mingjie Liu, and Haoxing Ren. 2023. Rtlfixer: Automatically fixing rtl syntax errors with large language models. *arXiv preprint arXiv:2311.16543*.

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

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

Alexander Matt Turner, Lisa Thiergart, Gavin Leech, David Udell, Juan J Vazquez, Ulisse Mini, and Monte MacDiarmid. 2023. Steering language models with activation engineering. *arXiv preprint arXiv:2308.10248*.

Szymon Tworkowski, Konrad Staniszewski, Mikoł aj Pacek, Yuhuai Wu, Henryk Michalewski, and Piotr Mił oś. 2023. Focused transformer: Contrastive training for context scaling. In *Advances in Neural Information Processing Systems*, volume 36, pages 42661–42688. Curran Associates, Inc.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Łukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6000–6010, Red Hook, NY, USA. Curran Associates Inc.

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

Luanbo Wan and Weizhi Ma. 2025. Storybench: A dynamic benchmark for evaluating long-term memory with multi turns. *arXiv preprint arXiv:2506.13356*.

Benyou Wang, Qianqian Xie, Jiahuan Pei, Zhihong Chen, Prayag Tiwari, Zhao Li, and Jie Fu. 2023a. Pretrained language models in biomedical domain: A systematic survey. *ACM Computing Surveys*, 56(3):1–52.

Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. 2024a. Enhancing large language model with self-controlled memory framework. *Preprint*, arXiv:2304.13343.

Chen Wang, Xunzhuo Liu, Yuhan Liu, Yue Zhu, Xiangxi Mo, Junchen Jiang, and Huamin Chen. 2025a. When to reason: Semantic router for vllm. *arXiv preprint arXiv:2510.08731*.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023b. Voyager: An open-ended embodied agent with large language models. *arXiv preprint arXiv:2305.16291*.

Haochun Wang, Chi Liu, Nuwa Xi, Zewen Qiang, Sendong Zhao, Bing Qin, and Ting Liu. 2023c. Huatuo: Tuning llama model with chinese medical knowledge. *arXiv preprint arXiv:2304.06975*.

Haochun Wang, Sendong Zhao, Zewen Qiang, Zijian Li, Nuwa Xi, Yanrui Du, MuZhen Cai, Haoqiang Guo, Yuhan Chen, Haoming Xu, and et al. 2023d. Knowledge-tuning large language models with structured medical knowledge bases for reliable response generation in chinese. *arXiv preprint arXiv:2309.04175*.

Junjie Wang, Yuchao Huang, Chunyang Chen, Zhe Liu, Song Wang, and Qing Wang. 2024b. Software testing with large language models: Survey, landscape, and vision. *IEEE Transactions on Software Engineering*.

Ke Wang, Nikolaos Dimitriadis, Guillermo Ortiz-Jimenez, François Fleuret, and Pascal Frossard. 2024c. Localizing task information for improved model merging and compression. *ICML*.

Lei Wang, Chen Ma, Xueyang Feng, Zeyu Zhang, Hao Yang, Jingsen Zhang, Zhiyuan Chen, Jiakai Tang, Xu Chen, Yankai Lin, and et al. 2023e. A survey on large language model based autonomous agents. *arxiv*.

Lei Wang, Jingsen Zhang, Hao Yang, Zhiyuan Chen, Jiakai Tang, Zeyu Zhang, Xu Chen, Yankai Lin, Ruihua Song, Wayne Xin Zhao, Jun Xu, Zhicheng Dou, Jun Wang, and Ji-Rong Wen. 2023f. When large language model based agent meets user behavior analysis: A novel user simulation paradigm.

Liang Wang, Nan Yang, Xiaolong Huang, Linjun Yang, Rangan Majumder, and Furu Wei. 2024d. Improving text embeddings with large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11897–11916. Association for Computational Linguistics.

Liang Wang, Nan Yang, and Furu Wei. 2023g. Learning to retrieve in-context examples for large language models. *arXiv preprint arXiv:2307.07164*.

Liyuan Wang, Xingxing Zhang, Hang Su, and Jun Zhu. 2024e. A comprehensive survey of continual learning: Theory, method and application. *IEEE Transactions on Pattern Analysis and Machine Intelligence*.

Longyue Wang, Zefeng Du, Wenxiang Jiao, Chenyang Lyu, Jianhui Pang, Leyang Cui, Kaiqiang Song, Derek F. Wong, Shuming Shi, and Zhaopeng Tu. 2024f. Benchmarking and improving long-text translation with large language models. In *Findings of the Association for Computational Linguistics, ACL 2024, Bangkok, Thailand and virtual meeting, August 11-16, 2024*, pages 7175–7187. Association for Computational Linguistics.

Peng Wang, Zexi Li, Ningyu Zhang, Ziwen Xu, Yunzhi Yao, Yong Jiang, Pengjun Xie, Fei Huang, and Huajun Chen. 2024g. Wise: Rethinking the knowledge memory for lifelong model editing of large language models. *Advances in Neural Information Processing Systems*, 37:53764–53797.

Qian Wang, Zhenheng Tang, and Bingsheng He. 2025b. Can LLM simulations truly reflect humanity? a deep dive. In *The Fourth Blogpost Track at ICLR 2025*.

Qingyue Wang, Yanan Fu, Yanan Cao, Shi Wang, Zhiliang Tian, and Liang Ding. 2025c. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, page 130193.

Qingyue Wang, Yanhe Fu, Yanan Cao, Shuai Wang, Zhiliang Tian, and Liang Ding. 2025d. Recursively summarizing enables long-term dialogue memory in large language models. *Neurocomputing*, 639:130193.

Song Wang, Yaochen Zhu, Haochen Liu, Zaiyi Zheng, Chen Chen, and Jundong Li. 2024h. Knowledge editing for large language models: A survey. *ACM Computing Surveys*, 57(3):1–37.

Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. 2023h. Augmenting language models with long-term memory. In *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*.

Wenjie Wang, Xinyu Lin, Fuli Feng, Xiangnan He, and Tat-Seng Chua. 2023i. Generative recommendation: Towards next-generation recommender paradigm. *arXiv*.

Xindi Wang, Mahsa Salmani, Parsa Omidi, Xiangyu Ren, Mehdi Rezagholizadeh, and Armaghan Eshaghi. 2024i. Beyond the limits: A survey of techniques to extend the context length in large language models. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence, IJCAI 2024, Jeju, South Korea, August 3-9, 2024*, pages 8299–8307. ijcai.org.

Xintao Wang, Qianwen Yang, Yongting Qiu, Jiaqing Liang, Qianyu He, Zhouhong Gu, Yanghua Xiao, and Wei Wang. 2023j. Knowledgpt: Enhancing large language models with retrieval and storage access on knowledge bases. *arXiv preprint arXiv:2308.11761*.

Yile Wang, Peng Li, Maosong Sun, and Yang Liu. 2023k. Self-knowledge guided retrieval augmentation for large language models. *arXiv preprint arXiv:2310.05002*.

Yu Wang, Yifan Gao, Xiusi Chen, Haoming Jiang, Shiyang Li, Jingfeng Yang, Qingyu Yin, Zheng Li, Xian Li, Bing Yin, and et al. 2024j. Memoryllm: Towards self-updatable large language models. *arXiv preprint arXiv:2402.04624*.

Yu Wang, Chi Han, Tongtong Wu, Xiaoxin He, Wangchunshu Zhou, Nafis Sadeq, Xiusi Chen, Zexue He, Wei Wang, Gholamreza Haffari, and 1 others. 2024k. Towards lifespan cognitive systems. *arXiv preprint arXiv:2409.13265*.

Yu Wang, Nedim Lipka, Ryan A Rossi, Alexa Siu, Ruiyi Zhang, and Tyler Derr. 2023l. Knowledge graph prompting for multi-document question answering. *arXiv preprint arXiv:2308.11730*.

Yu Wang, Xinshuang Liu, Xiusi Chen, Sean O'Brien, Junda Wu, and Julian McAuley. Self-updatable large language models by integrating context into model parameters. In *The Thirteenth International Conference on Learning Representations*.

Zefan Wang, Zichuan Liu, Yingying Zhang, Aoxiao Zhong, Lunting Fan, Lingfei Wu, and Qingsong Wen. 2023m. Rcagent: Cloud root cause analysis by autonomous agents with tool-augmented large language models. *arXiv*.

Zhepeng Wang, Runxue Bao, Yawen Wu, Jackson Taylor, Cao Xiao, Feng Zheng, Weiwen Jiang, Shangqian Gao, and Yanfu Zhang. 2024l. Unlocking memorization in large language models with dynamic soft

prompting. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 9782–9796, Miami, Florida, USA. Association for Computational Linguistics.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2024m. Agent workflow memory. *Preprint*, arXiv:2409.07429.

Zora Zhiruo Wang, Jiayuan Mao, Daniel Fried, and Graham Neubig. 2024n. Agent workflow memory. *arXiv preprint arXiv:2409.07429*.

Lilian Weng. 2023. Llm-powered autonomous agents. lilianweng.github.io.

Di Wu, Hongwei Wang, Wenhao Yu, Yuwei Zhang, Kai-Wei Chang, and Dong Yu. 2024a. Longmemeval: Benchmarking chat assistants on long-term interactive memory. *arXiv preprint arXiv:2410.10813*.

Haoyi Wu and Kewei Tu. 2024. Layer-condensed KV cache for efficient inference of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11175–11188, Bangkok, Thailand. Association for Computational Linguistics.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Wei Wu, Zhuoshi Pan, Chao Wang, Liyi Chen, Yunchu Bai, Tianfu Wang, Kun Fu, Zheng Wang, and Hui Xiong. 2025a. Tokenselect: Efficient long-context inference and length extrapolation for llms via dynamic token-level kv cache selection. *Preprint*, arXiv:2411.02886.

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

Yichen Wu, Hong Wang, Peilin Zhao, Yefeng Zheng, Ying Wei, and Long-Kai Huang. 2024b. Mitigating catastrophic forgetting in online continual learning by modeling previous task interrelations via pareto optimization. In *Forty-first International Conference on Machine Learning*.

Yuhuai Wu, Markus N Rabe, DeLesley Hutchins, and Christian Szegedy. 2022. Memorizing transformers. *arXiv preprint arXiv:2203.08913*.

Wm. A. Wulf and Sally A. McKee. 1995. Hitting the memory wall: implications of the obvious. *SIGARCH Comput. Archit. News*, 23(1):20–24.

Yunjia Xi, Weiwen Liu, Jianghao Lin, Bo Chen, Ruiming Tang, Weinan Zhang, and Yong Yu. 2024. Memocrs: Memory-enhanced sequential conversational recommender systems with large language models. In *Proceedings of the 33rd ACM International Conference on Information and Knowledge Management*, pages 2585–2595.

Zhiheng Xi, Wenxiang Chen, Xin Guo, Wei He, Yiwen Ding, Boyang Hong, Ming Zhang, Junzhe Wang, Senjie Jin, Enyu Zhou, and et al. 2023. The rise and potential of large language model based agents: A survey. *arxiv*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2023. Efficient streaming language models with attention sinks. *arXiv*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024a. Efficient streaming language models with attention sinks. In *The Twelfth International Conference on Learning Representations*.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. 2024b. Efficient streaming language models with attention sinks. *Preprint*, arXiv:2309.17453.

Honglin Xiong, Sheng Wang, Yitao Zhu, Zihao Zhao, Yuxiao Liu, Qian Wang, and Dinggang Shen. 2023. Doctorglm: Fine-tuning your chinese doctor is not a herculean task. *arXiv preprint arXiv:2304.01097*.

Derong Xu, Wei Chen, Wenjun Peng, Chao Zhang, Tong Xu, Xiangyu Zhao, Xian Wu, Yefeng Zheng, and Enhong Chen. 2023. Large language models for generative information extraction: A survey. *arXiv*.

Jing Xu, Arthur Szlam, and Jason Weston. 2021. Beyond goldfish memory: Long-term open-domain conversation. *arXiv preprint arXiv:2107.07567*.

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

Prateek Yadav, Derek Tam, Leshem Choshen, Colin Raffel, and Mohit Bansal. 2023. Ties-merging: Resolving interference when merging models.

Shi-Qi Yan, Jia-Chen Gu, Yun Zhu, and Zhen-Hua Ling. 2024. Corrective retrieval augmented generation. *arXiv preprint arXiv:2401.15884*.

Dongjie Yang, Xiaodong Han, Yan Gao, Yao Hu, Shilin Zhang, and Hai Zhao. 2024a. PyramidInfer: Pyramid KV cache compression for high-throughput LLM inference. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3258–3270, Bangkok, Thailand. Association for Computational Linguistics.

Enneng Yang, Zhenyi Wang, Li Shen, Shiwei Liu, Guibing Guo, Xingwei Wang, and Dacheng Tao. 2024b. Adamerging: Adaptive model merging for multi-task learning. *ICLR*.

Hongkang Yang, Zehao Lin, Wenjin Wang, Hao Wu, Zhiyu Li, Bo Tang, Wenqiang Wei, Jinbo Wang, Zeyun Tang, Shichao Song, and 1 others. 2024c. $memory^3$: Language modeling with explicit memory. *arXiv preprint arXiv:2407.01178*.

Huan Yang, Renji Zhang, Mingzhe Huang, Weijun Wang, Yin Tang, Yuanchun Li, Yunxin Liu, and Deyu Zhang. 2025a. Kvshare: An llm service system with efficient and effective multi-tenant kv cache reuse. *Preprint*, arXiv:2503.16525.

Jingbo Yang, Bairu Hou, Wei Wei, Yujia Bao, and Shiyu Chang. 2025b. Kvlink: Accelerating large language models via efficient kv cache reuse. *Preprint*, arXiv:2502.16002.

June Yong Yang, Byeongwook Kim, Jeongin Bae, Beomseok Kwon, Gunho Park, Eunho Yang, Se Jung Kwon, and Dongsoo Lee. 2024d. No token left behind: Reliable kv cache compression via importance-aware mixed precision quantization. *Preprint*, arXiv:2402.18096.

Ling Yang, Zhaochen Yu, Tianjun Zhang, Shiyi Cao, Minkai Xu, Wentao Zhang, Joseph E Gonzalez, and Bin Cui. 2024e. Buffer of thoughts: Thought-augmented reasoning with large language models. *arXiv preprint arXiv:2406.04271*.

Sophia Yang. 2023. Advanced rag 01: Small-to-big retrieval. https://towardsdatascience.com/advanced-rag-01-small-to-big-retrieval-172181b396d4.

Zhenjie Yang, Xiaosong Jia, Hongyang Li, and Junchi Yan. 2023. A survey of large language models for autonomous driving. *arXiv*.

Jiayi Yao, Hanchen Li, Yuhan Liu, Siddhant Ray, Yihua Cheng, Qizheng Zhang, Kuntai Du, Shan Lu, and Junchen Jiang. 2025. Cacheblend: Fast large language model serving for rag with cached knowledge fusion. *Preprint*, arXiv:2405.16444.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2022. React: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

Weiran Yao, Shelby Heinecke, Juan Carlos Niebles, Zhiwei Liu, Yihao Feng, Le Xue, Rithesh Murthy, Zeyuan Chen, Jianguo Zhang, Devansh Arpit, and et al. 2023. Retroformer: Retrospective large language agents with policy gradient optimization. *arXiv preprint arXiv:2308.02151*.

Yao Yao, Zuchao Li, and Hai Zhao. 2024. Sirllm: Streaming infinite retentive llm. *Preprint*, arXiv:2405.12528.

Lu Ye, Ze Tao, Yong Huang, and Yang Li. 2024. Chunkattention: Efficient self-attention with prefix-aware kv cache and two-phase partition. *Preprint*, arXiv:2402.15220.

Gyeong-In Yu, Joo Seong Jeong, Geon-Woo Kim, Soojeong Kim, and Byung-Gon Chun. 2022a. Orca: A distributed serving system for Transformer-Based generative models. In *16th USENIX Symposium on Operating Systems Design and Implementation (OSDI 22)*, pages 521–538, Carlsbad, CA. USENIX Association.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *ICML*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024b. Language models are super mario: Absorbing abilities from homologous models as a free lunch. *ICML*.

Wenhao Yu, Dan Iter, Shuohang Wang, Yichong Xu, Mingxuan Ju, Soumya Sanyal, Chenguang Zhu, Michael Zeng, and Meng Jiang. 2022b. Generate rather than retrieve: Large language models are strong context generators. *arXiv preprint arXiv:2209.10063*.

Wenhao Yu, Hongming Zhang, Xiaoman Pan, Kaixin Ma, Hongwei Wang, and Dong Yu. 2023. Chain-of-note: Enhancing robustness in retrieval-augmented language models. *arXiv preprint arXiv:2311.09210*.

Li Yunxiang, Li Zihan, Zhang Kai, Dan Ruilong, and Zhang You. 2023. Chatdoctor: A medical chat model fine-tuned on llama model using medical domain knowledge. *arXiv preprint arXiv:2303.14070*.

Fanlong Zeng, Wensheng Gan, Yongheng Wang, Ning Liu, and Philip S Yu. 2023. Large language models for robotics: A survey. *arXiv*.

Ruihong Zeng, Jinyuan Fang, Siwei Liu, and Zaiqiao Meng. 2024. On the structural memory of llm agents. *arXiv preprint arXiv:2412.15266*.

Liangyu Zha, Junlin Zhou, Liyao Li, Rui Wang, Qingyi Huang, Saisai Yang, Jing Yuan, Changbao Su, Xiang Li, Aofeng Su, and 1 others. 2023. Tablegpt: Towards unifying tables, nature language and commands into one gpt. *arXiv preprint arXiv:2307.08674*.

Kechi Zhang, Jia Li, Ge Li, Xianjie Shi, and Zhi Jin. 2024a. Codeagent: Enhancing code generation with tool-integrated agent systems for real-world repo-level coding challenges. *arXiv preprint arXiv:2401.07339*.

Peitian Zhang, Zheng Liu, Shitao Xiao, Ninglu Shao, Qiwei Ye, and Zhicheng Dou. 2025a. Long context compression with activation beacon. In *The Thirteenth International Conference on Learning Representations*.

Tianyi Zhang, Jonah Yi, Zhaozhuo Xu, and Anshumali Shrivastava. 2024b. KV cache is 1 bit per channel: Efficient large language model inference with coupled quantization. *Advances in Neural Information Processing Systems, NeurIPS 2024*, 37:3304–3331.

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

Zeyu Zhang, Quanyu Dai, Xiaohe Bo, Chen Ma, Rui Li, Xu Chen, Jieming Zhu, Zhenhua Dong, and Ji-Rong Wen. 2025b. A survey on the memory mechanism of large language model based agents. *ACM Trans. Inf. Syst.* Just Accepted.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Re, Clark Barrett, Zhangyang Wang, and Beidi Chen. 2023a. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang "Atlas" Wang, and Beidi Chen. 2023b. H2o: Heavy-hitter oracle for efficient generative inference of large language models. In *Advances in Neural Information Processing Systems*, volume 36, pages 34661–34710. Curran Associates, Inc.

Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, and 1 others. 2023c. H2o: Heavy-hitter oracle for efficient generative inference of large language models. *Advances in Neural Information Processing Systems*, 36:34661–34710.

Zhuocheng Zhang, Yang Feng, and Min Zhang. 2025c. Levelrag: Enhancing retrieval-augmented generation with multi-hop logic planning over rewriting augmented searchers. *Preprint*, arXiv:2502.18139.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2023a. Expel: Llm agents are experiential learners. *arXiv preprint arXiv:2308.10144*.

Andrew Zhao, Daniel Huang, Quentin Xu, Matthieu Lin, Yong-Jin Liu, and Gao Huang. 2024a. Expel: Llm agents are experiential learners. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19632–19642.

Pengyu Zhao, Zijian Jin, and Ning Cheng. 2023b. An in-depth survey of large language model-based artificial intelligence agents. *arXiv*.

Yilong Zhao, Chien-Yu Lin, Kan Zhu, Zihao Ye, Lequn Chen, Size Zheng, Luis Ceze, Arvind Krishnamurthy, Tianqi Chen, and Baris Kasikci. 2024b. Atom: Low-bit quantization for efficient and accurate llm serving. In *MLSys*.

Zihan Zhao, Da Ma, Lu Chen, Liangtai Sun, Zihao Li, Hongshen Xu, Zichen Zhu, Su Zhu, Shuai Fan, Guodong Shen, and et al. 2024c. Chemdfm: Dialogue foundation model for chemistry. *arXiv preprint arXiv:2401.14818*.

Huaixiu Steven Zheng, Swaroop Mishra, Xinyun Chen, Heng-Tze Cheng, Ed H. Chi, Quoc V Le, and Denny Zhou. 2024a. Take a step back: Evoking reasoning via abstraction in large language models. *Preprint*, arXiv:2310.06117.

Lianmin Zheng, Liangsheng Yin, Zhiqiang Xie, Chuyue Sun, Jeff Huang, Cody Hao Yu, Shiyi Cao, Christos Kozyrakis, Ion Stoica, Joseph E. Gonzalez, Clark Barrett, and Ying Sheng. 2024b. Sglang: Efficient execution of structured language model programs. *Preprint*, arXiv:2312.07104.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. 2023a. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Longtao Zheng, Rundong Wang, Xinrun Wang, and Bo An. 2024c. Synapse: Trajectory-as-exemplar prompting with memory for computer control. In *Proceedings of the International Conference on Learning Representations (ICLR)*.

Zibin Zheng, Kaiwen Ning, Yanlin Wang, Jingwen Zhang, Dewu Zheng, Mingxi Ye, and Jiachi Chen. 2023b. A survey of large language models for code: Evolution, benchmarking, and future trends. *arXiv*.

Meizhi Zhong, Xikai Liu, Chen Zhang, Yikun Lei, Yan Gao, Yao Hu, Kehai Chen, and Min Zhang. 2024a. Zigzagkv: Dynamic kv cache compression for long-context modeling based on layer uncertainty. *Preprint*, arXiv:2412.09036.

Wanjun Zhong, Lianghong Guo, Qiqi Gao, and Yanlin Wang. 2023. Memorybank: Enhancing large language models with long-term memory. *arXiv preprint arXiv:2305.10250*.

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

Wanjun Zhong, Lianghong Guo, Qiqi Gao, He Ye, and Yanlin Wang. 2024c. Memorybank: Enhancing large language models with long-term memory. In *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*, pages 19724–19731. AAAI Press.

Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang.

2024d. Distserve: disaggregating prefill and decoding for goodput-optimized large language model serving. In *Proceedings of the 18th USENIX Conference on Operating Systems Design and Implementation*, OSDI'24, USA. USENIX Association.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023a. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

Hongjian Zhou, Boyang Gu, Xinyu Zou, Yiru Li, Sam S Chen, Peilin Zhou, Junling Liu, Yining Hua, Chengfeng Mao, Xian Wu, and et al. 2023b. A survey of large language models in medicine: Progress, application, and challenge. *arXiv*.

Xiabin Zhou, Wenbin Wang, Minyan Zeng, Jiaxian Guo, Xuebo Liu, Li Shen, Min Zhang, and Liang Ding. 2025. Dynamickv: Task-aware adaptive kv cache compression for long context llms. *Preprint*, arXiv:2412.14838.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, and 1 others. 2024. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*.

Dawei Zhu, Liang Wang, Nan Yang, Yifan Song, Wenhao Wu, Furu Wei, and Sujian Li. 2024. Longembed: Extending embedding models for long context retrieval. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing, EMNLP 2024, Miami, FL, USA, November 12-16, 2024*, pages 802–816. Association for Computational Linguistics.

Qiuyu Zhu, Liang Zhang, Qianxiong Xu, Cheng Long, and Jie Zhang. 2025a. Subgcache: Accelerating graph-based rag with subgraph-level kv cache. *Preprint*, arXiv:2505.10951.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, and et al. 2023a. 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*.

Xizhou Zhu, Yuntao Chen, Hao Tian, Chenxin Tao, Weijie Su, Chenyu Yang, Gao Huang, Bin Li, Lewei Lu, Xiaogang Wang, and et al. 2023b. 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*.

Yuanbing Zhu, Zhenheng Tang, Xiang Liu, Ang Li, Bo Li, Xiaowen Chu, and Bo Han. 2025b. OracleKV: Oracle guidance for question-independent KV cache compression. In *ICML 2025 Workshop on Long-Context Foundation Models*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023c. Large language models for information retrieval: A survey. *arXiv*.

Yutao Zhu, Huaying Yuan, Shuting Wang, Jiongnan Liu, Wenhan Liu, Chenlong Deng, Zhicheng Dou, and Ji-Rong Wen. 2023d. Large language models for information retrieval: A survey. *CoRR*, abs/2308.07107.

Yuxuan Zhu, Ali Falahati, David H. Yang, and Mohammad Mohammadi Amiri. 2025c. Sentencekv: Efficient llm inference via sentence-level semantic kv caching. *Preprint*, arXiv:2504.00970.

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

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., 2023e)) 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., 2023b; 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., 2024d). 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., 2023c) 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., 2023a; Lin et al., 2023a; Wang et al., 2023i). In software engineering, surveys summarize the use of LLMs across design, development, and testing (Fan et al., 2023; Wang et al., 2024b; Zheng et al., 2023b). Other domain surveys cover robotics (Zeng et al., 2023), autonomous driving (Cui et al., 2024; Yang et al., 2023), medicine (He et al., 2023a; Zhou et al., 2023b; Wang et al., 2023a), finance (Li et al., 2023d), and psychology (He et al., 2023b). 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., 2024c), yet many narrow their scope to long-context modeling (Huang et al., 2023), long-term memory (He et al., 2024d; Jiang et al., 2024a), personalization (Liu et al., 2025a), or knowledge editing (Wang et al., 2024h). 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 (Jiang et al., 2024a; Zhang et al., 2024c; Zhong et al., 2024b). Others focus on memory mechanisms inside LLM-based agents and review their design, evaluation, and applications for self-evolving behaviors (Zhang et al., 2025b). 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., 2025b). 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 (Hatalis et al., 2024; LI et al., 2025; Luohe et al.). 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 sur-

vey (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.

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

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).

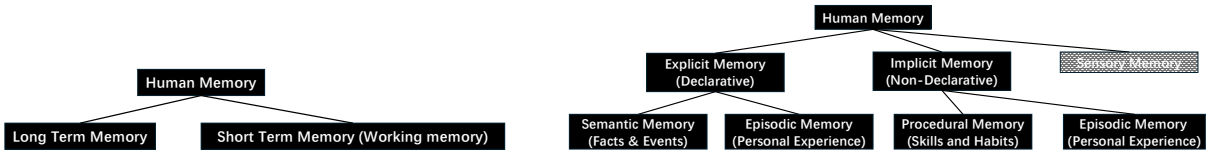
Based on the *duration of information retention*, human memory is classified into **short-term** and **long-term memory**, as illustrated in Figure 2(a) (Baddeley, 2007). Short-term memory temporarily holds and processes information for seconds to minutes as working memory, which actively manipulates 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).

Based on functional roles, human memory is commonly categorized into **explicit** (declarative), **implicit** (non-declarative), and **sensory memory** *, as shown in Figure 2(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., 2023a). Memory also supports environment exploration by guiding agents to prioritize less-explored actions or revisit previously unsuccessful trials, enhancing adaptability (Montazeral-

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



(a) Memory Categories according to duration.

(b) Memory Categories according to semantics.

Figure 2: Human Memory Overview.

ghaem et al., 2020; Zhu et al., 2023a). 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., 2023a).

B.2 LLM Memory

LLM Inference. LLMs such as GPT (Brown et al., 2020a), LLaMA (Touvron et al., 2023a,b; Grattafiori et al., 2024), Qwen (Qwen et al., 2025; Bai et al., 2023), and DeepSeek (DeepSeek-AI et al., 2025a,b) 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:

$$P(x_1, \ldots, x_n) = P(x_1) \times P(x_2 \mid x_1) \times \ldots \times P(x_n \mid x_1, \ldots, x_{n-1}). \quad (1)$$

The inference process (Equation 1) 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 2.

$$\text{concat}(q_i, k_i, v_i) = \text{concat}(W_q, W_k, W_v) \cdot h_i \quad (2)$$

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:

$$a_{ij} = \frac{\exp(q_i^\top k_j / \sqrt{d})}{\sum_{t=1}^{i} \exp(q_i^\top k_t / \sqrt{d})}, \quad o_i = \sum_{j=1}^{i} a_{ij} v_j. \quad (3)$$

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 (3), weighs the relevance

Table 2: 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

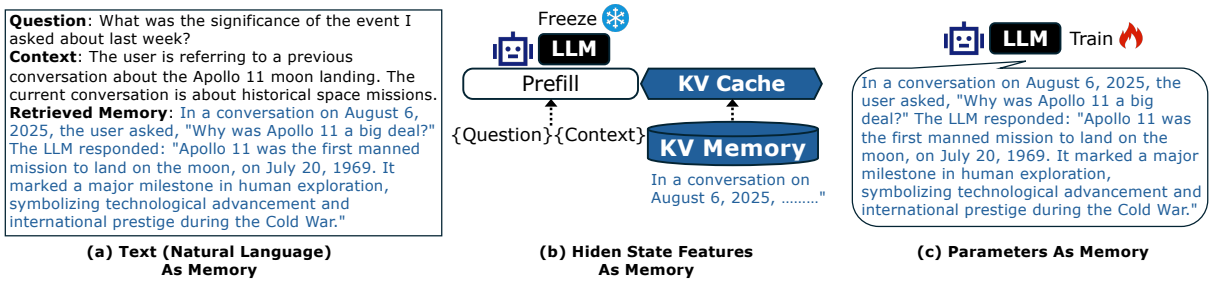


Figure 3: LLM Memory Classification.

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., 2020a) 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., 2023b). 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 (2), 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,b; Grattafiori et al., 2024) and DeepSeek (DeepSeek-AI et al., 2025a,b).

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., 2024d, 2025d; Zhu et al., 2025b; Liu et al., 2025c). KIVI introduces asymmetric 2-bit quantization for KV cache entries, further reducing memory footprint (Liu et al., 2024e). Methods like SnapKV and H2O selectively retain important KV pairs based on attention patterns, improving efficiency for long-context inference (Li et al., 2024e; Zhang et al., 2023b). Additionally, StreamingLLM uses atten-

tion 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., 2024c; Li et al., 2025a). These optimizations make the KV cache a critical memory mechanism for scalable and efficient LLM inference.

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 3, 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 3. 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., 2024e; Li et al., 2024e; Xiao et al., 2023). 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.

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 4 and 5.

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 (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.

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

Table 3: 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.

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 4: Reconstructed Memory Implementation Characteristics.

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 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.

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., 2023b). 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., 2025b, 2023f). 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.,

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

Table 5: Reconstructed Characteristics of Memory Types by Implementation.

2024a; Shi et al., 2024; Beltagy et al., 2020; Guo et al., 2022; Jin et al., 2024b). 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., 2023d; Wang et al., 2024d; OpenAI, 2024b; Günther et al., 2023; Chen et al., 2024a; Zhu et al., 2024; Saad-Falcon et al., 2024; Herold and Ney, 2023; Wang et al., 2024f; 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, 2024a; Inflection, 2023; Character AI, 2023; Ai, 2024; Lee et al., 2023a; Zhong et al., 2024c; Wang et al., 2023h, 2024i).

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 (Tsai et al., 2023; Chen et al., 2023a; Qian et al., 2023; Li et al., 2023e; Zhang et al., 2024a; 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 (Wang et al., 2025b; Shao et al., 2023; Kaiya et al., 2023; Gao et al., 2023; Li et al., 2023b,e; 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., 2023b; Lee et al., 2023b; Pan et al., 2023a; 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 et al., 2024; Hilgert et al., 2024; Shao and Yan, 2024; Wang et al., 2023c; Xiong et al., 2023; Liu et al., 2023d; Wang et al., 2023d; Yunxiang et al., 2023; Chen et al., 2024b; Zhao et al., 2024c; Chen et al., 2023c; Wang et al., 2023m; Qiang et al., 2023).

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., 2022a; Zhong et al., 2024d).

Unified Training–Inference Systems. Current LLM systems largely separate training from inference, limiting adaptation to evolving users and environments (Brown et al., 2020b; 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 (Park et al., 2023; Shinn 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 (Kwon et al., 2023; Xiao et al., 2024b; Zhang et al., 2023a). RAG systems leverage database techniques for indexing, query optimization, and execution planning (Karpukhin et al., 2020; Khattab and Zaharia, 2020; Izacard et al., 2023; Asai et al., 2023). Furthermore, distributed systems and cloud computing inspire solutions for scaling long-context workloads through partitioning and remote memory (Zhong et al., 2024d; Fu et al., 2024; Jin and Wu, 2025). These established paradigms provide a foundational blueprint for scalable and efficient LLM memory orchestration.

D Taxonomy of Different Memory

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

Table 6: Taxonomy of RAG.

Agentic Memory §2.2		
Memory Construction	Summarization	MemoryBank (Zhong et al., 2024b), 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., 2024b), (Pan et al., 2025), Mem0 (mem0ai, 2024)
	Relation Graph	CGSN (Nie et al., 2022), GraphReader (Li et al., 2024c), HippoRAG (Gutiérrez et al., 2024)
	Auxiliary Signals	LongMemEval (Wu et al., 2024a), THEANINE (iunn Ong et al., 2025)
	Storage and Inference Efficiency	Llmlingua (Jiang et al., 2023a), AutoCompressor (Chevalier et al., 2023), TCRA-LLM (Liu et al., 2023a), Promptcache (Gim et al., 2024)
Memory Updating	Summarization and Restructuring	MemoryBank (Zhong et al., 2024b), ChatGPT-RSum (Wang et al., 2025d), MemoChat (Lu et al., 2023a)
	Reasoning and Self-Reflection	ReAct (Yao et al., 2022), Reflexion (Shinn et al., 2024), BoT (Yang et al., 2024e), Agent Workflow Memory (Wang et al., 2024n)
	Reasoning with Experience	TiM (Liu et al., 2023b), GITM (Zhu et al., 2023b), Voyager (Wang et al., 2023b), Retroformer (Yao et al., 2023), ExpeL (Zhao et al., 2024a), LD-Agent (Li et al., 2024a)
	Memory Evolution	A-MEM (Xu et al., 2025), Synapse (Zheng et al., 2024c), R2I (Samsami et al., 2024), SCM (Wang et al., 2024a), Selective Editing (Bae et al., 2022), Blending (Kim et al., 2024b), Recursive Summarization (Wang et al., 2025c), Self-Reflection (Sun et al., 2024)
Memory Query	Query-centered	FLARE (Jiang et al., 2023b), IterCQR (Jang et al., 2024)
	Memory-centered	LongMemEval (Wu et al., 2024a), 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 7: Taxonomy of Agent Memory.

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

Table 8: Taxonomy of KV Cache as Memory.

Other Vectors as Memory §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., 2024c), MemOS (Li et al., 2025c), Memory3 (Yang et al., 2024c)
Steering Vectors	Foundational method	PPLM (Dathathri et al.)
	Contrastive-based	Turner et al. (Turner et al., 2023), Liu et al. (Liu et al., 2023c), 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., 2024b), Turner et al. (Turner et al., 2025), Cao et al. (Cao et al., 2024)

Table 9: Taxonomy of Other Vectors as Memory.

Parameter as Memory §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., 2024l)
	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., 2024e), EWC (Kirkpatrick et al., 2017), TaSL (Feng et al., 2024), SELF-PARAM (Wang et al.), POCL (Wu et al., 2024b), DSI++ (Mehta et al., 2022), LSCS (Wang et al., 2024k)
	PEFT	PEFT (Han et al., 2024b), Character-LLM (Shao et al.), AI-Native Memory (Shang et al., 2024), MemoRAG (Qian et al., 2024), Echo (Liu et al., 2025b)
	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., 2024a,b), Model Breadcrumbs (Davari and Belilovsky, 2023), TALL-masks (Wang et al., 2024c), SMEAR (Muqeeth et al., 2024), Twin-Merging (Lu et al., 2024b), Weight-Ensembling MoE (Tang et al., 2024a)
	Task Arithmetic	Task Arithmetic (Ilharco et al., 2022), TIES (Yadav et al., 2023), AdaMerging (Yang et al., 2024b), TwinMerge (Lu et al., 2024a)
	Model Editing	Wang et al. (Wang et al., 2024h), MemoryLLM (Wang et al., 2024j), WISE (Wang et al., 2024g)
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)

Table 10: Taxonomy of Parameter as Memory.