Enze Ge[1], Qihang Jin[2], Yuhang Xie[2], Hongying Luo[2], Tianyang Wang[2], Ziqian Bi[3], Benji Peng[4], Junfeng Hao[2], and Junhao Song[5]

[1]University of Bologna
[2]AI Agent Lab, Vokram Group
[3]Purdue University
[4]Georgia Institute of Technology
[5]Imperial College London

October 21, 2025

Abstract

This paper presents a comprehensive review of Memory-Augmented Large Language Models (LLMs), with a particular emphasis on optimization strategies and alignment with human preferences. We synthesize recent advances in memory-efficient training, model compression (quantization, pruning, low-rank approximations, and differentiable SVD), and architectural innovations that integrate explicit memory modules to enhance long-context reasoning and interpretability. The paper also examines alignment methodologies, ranging from supervised fine-tuning to reinforcement learning with human or AI feedback, as well as emerging approaches for inference-time personalization. Beyond technical optimization, we highlight challenges in evaluation, data efficiency, and ethical considerations such as fairness, privacy, and memorization risks. Representative applications in domains including business, healthcare, education, and scientific discovery illustrate the practical value and limitations of optimized and memory-augmented LLMs. Our contributions are threefold: (i) providing a structured overview of the state of the art in LLM optimization, memory augmentation, and alignment, (ii) identifying open challenges in efficiency, interpretability, and responsible deployment, and (iii) outlining future research directions toward more scalable, trustworthy, and user-centric LLMs. This review offers researchers and practitioners a timely roadmap for advancing the next generation of large-scale language models.

From Memory to Alignment: A Comprehensive Review of Large Language Model Optimization

Enze Ge[1], Qihang Jin[2], Yuhang Xie[2], Hongying Luo[2], Tianyang Wang[2], Ziqian Bi[3], Benji Peng[4], Junfeng Hao[2], Junhao Song[5†],

[1]University of Bologna, Italy, enze.ge@studio.unibo.it
[2]AI Agent Lab, Vokram Group, United Kingdom, ai-agent-lab@vokram.com
[3]Purdue University, United States, bi32@purdue.edu
[4]Georgia Institute of Technology, United States, benji@appcubic.com
[5]Imperial College London, United Kingdom, junhao.song23@imperial.ac.uk

***Abstract*—This paper presents a comprehensive review of Memory-Augmented Large Language Models (LLMs), with a particular emphasis on optimization strategies and alignment with human preferences. We synthesize recent advances in memory-efficient training, model compression (quantization, pruning, low-rank approximations, and differentiable SVD), and architectural innovations that integrate explicit memory modules to enhance long-context reasoning and interpretability. The paper also examines alignment methodologies, ranging from supervised fine-tuning to reinforcement learning with human or AI feedback, as well as emerging approaches for inference-time personalization. Beyond technical optimization, we highlight challenges in evaluation, data efficiency, and ethical considerations such as fairness, privacy, and memorization risks. Representative applications in domains including business, healthcare, education, and scientific discovery illustrate the practical value and limitations of optimized and memory-augmented LLMs. Our contributions are threefold: (i) providing a structured overview of the state of the art in LLM optimization, memory augmentation, and alignment, (ii) identifying open challenges in efficiency, interpretability, and responsible deployment, and (iii) outlining future research directions toward more scalable, trustworthy, and user-centric LLMs. This review offers researchers and practitioners a timely roadmap for advancing the next generation of large-scale language models.**

***Index Terms*—large language models, multimodal learning, natural language processing, machine learning, artificial intelligence**

I. Introduction and Background

Large Language Models (LLMs) have revolutionized the field of natural language processing, demonstrating remarkable capabilities in a wide range of tasks, from traditional natural language understanding to knowledge-intensive and generative tasks [57], [60]. Their success has spurred extensive research into architectural innovations, improved training strategies, and multi-modal integration [60]. A comprehensive overview of the existing literature on LLM-related concepts is provided in [60], which highlights the importance of understanding both the strengths and limitations of these models.

The papers analyzed provide detailed perspectives on the mechanics, applications, and research directions of LLMs [37], [66]. Key findings include the ability of LLMs to perform diverse NLP tasks, including knowledge-intensive tasks, traditional natural language understanding, and natural language generation [104]. However, limitations such as spurious biases, efficiency issues, high cost, and latency remain major concerns [44], [66]. The importance of data is also emphasized, with studies highlighting the role of pre-training corpora, training data, and test data in shaping LLM performance [17].

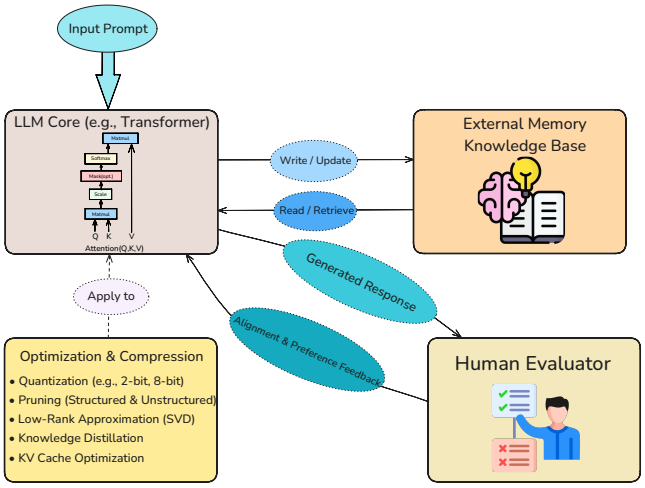

Fig. 1: Memory-augmented LLM architecture. The figure depicts the LLM core interacting with external memory modules, human feedback, and optimization techniques. External knowledge bases enable read–write operations, while human evaluators provide alignment signals. Optimization and compression methods, including quantization, pruning, low-rank approximation, knowledge distillation, and KV cache optimization, apply across components to improve efficiency.

As illustrated in Figure 1, a memory-augmented architecture extends the conventional LLM core by integrating external knowledge bases for persistent memory [56], feedback loops from human evaluators for alignment [40], and optimization techniques for efficiency [20]. This holistic view underscores

†: Corresponding author.

that LLM research must simultaneously address architectural design, memory integration, and resource constraints.

Recent surveys [57], [60] highlight that optimization for efficiency remains a central challenge. Memory efficiency during training is particularly critical, with studies emphasizing distributed training, mixed precision training, and gradient checkpointing as effective strategies to reduce resource consumption [3], [85]. The total memory footprint during backpropagation can be decomposed as:

$$M_{\text{total}} = M_{\text{model}} + M_{\text{optimizer}} + M_{\text{gradients}} + M_{\text{activations}}, \quad (1)$$

where M_{model} stores parameters, $M_{\text{optimizer}}$ holds states such as Adam moments, $M_{\text{gradients}}$ caches derivatives, and $M_{\text{activations}}$ keeps intermediate tensors. When mixed precision is adopted, the theoretical memory saving is estimated as:

$$M_{\text{saved}} = \tfrac{1}{2}\big(M_{\text{model}} + M_{\text{gradients}}\big). \quad (2)$$

This provides a practical rule-of-thumb for memory profiling in large-scale training settings.

Model compression is another critical strategy, with pruning and quantization widely evaluated to reduce inference costs [9], [20]. Surveys categorize optimization methods by their focus on computational, memory, energy, financial, and network resources [3]. Beyond training efficiency, compression and distillation are important for deployment on edge devices and mobile systems [9].

As shown in Figure 2, memory-augmented agents can be understood from two perspectives. On the left, the diagram depicts the interaction loop between the environment, AI agents, and LLMs equipped with explicit memory modules [56], emphasizing how contextual information can be stored and retrieved across interactions. On the right, representative application scenarios illustrate the breadth of domains where memory augmentation can improve personalization, reasoning, and continuity, ranging from healthcare and education to software development and daily-life assistance [37], [57].

Recent theoretical work further underscores the significance of memory: augmenting an LLM with a read–write external memory has been proven to elevate its computational expressiveness to that of a universal Turing machine, enabling the simulation of arbitrary algorithms without modifying model weights [70]. This result highlights the transformative potential of memory augmentation in extending LLM capabilities beyond fixed context windows.

Overall, the literature reveals intertwined progress in architecture, training, optimization, and memory integration. While substantial challenges persist, the frameworks illustrated in Figures. 1 and 2 capture the dual emphasis on efficiency and applicability, laying the groundwork for deeper exploration in the following sections [3], [9], [37], [57], [60], [85].

II. Memory-Augmented Architecture and Training

The integration of explicit memory modules is another approach to enhancing LLMs and improving their interpretability [56]. The MemLLM paper proposes a novel method of fine-tuning LLMs to use an explicit read-and-write memory module, which improves the model's performance and interpretability in language modeling and knowledge-intensive tasks [56]. This approach highlights the potential benefits of incorporating structured memory mechanisms into LLM architectures. Furthermore, [69] provides a comprehensive overview of the literature on memorization in LLMs, analyzing it across five key dimensions: intentionality, degree, retrievability, abstraction, and transparency.

Attention mechanisms play a significant role in memorization. In standard transformer architectures, the self-attention operation is defined as:

$$\text{Attn}(Q, K, V) = \text{softmax}\Big(\tfrac{QK^\top}{\sqrt{d_k}}\Big)V, \quad (3)$$

where Q, K, and V represent the query, key, and value matrices, respectively, and d_k is the dimensionality of the key vectors. This mechanism enables LLMs to dynamically retrieve relevant information from previous tokens, a key process that contributes to both generalization and memorization.

Empirical findings suggest that deeper transformer blocks are primarily responsible for memorization, while earlier layers are more crucial for generalization and reasoning capabilities [53]. The paper "Analyzing Memorization in Large Language Models through the Lens of Model Attribution" investigates memorization from an architectural perspective by analyzing the impact of different attention layers. A layer-wise saliency score often used to assess the importance of individual layers in memorization is defined as:

$$M_{\text{score}} = \max_l \big\|\partial \mathcal{L}/\partial W_l\big\|_2, \quad (4)$$

where $\mathcal{L}$ denotes the loss function and W_l is the weight matrix at layer l. This score reflects the sensitivity of the output to each layer, providing insights into which components may be memorizing training data.

Algorithm 1 Memory-Augmented Forward Pass

Input: Input sequence $x_{1:T}$, initial memory $M(0)$
for $t = 1$ to T **do**
  $h_t \leftarrow \text{LLM}(x_t, M_{\text{read}}(t-1))$
  $M(t) \leftarrow \text{Write}(M(t-1), h_t, w_t)$
  $M_{\text{read}}(t) \leftarrow \text{Read}(M(t), r_t)$
end for
Output: Final hidden state h_T, updated memory $M(T)$

A typical memory-augmented LLM architecture includes read and write operations into an external memory module across time steps. At time t, the output representation is:

$$\begin{aligned} h_t &= \text{LLM}\big(x_t,\ M_{\text{read}}(t{-}1)\big), \\ M(t) &= \text{Write}\big(M(t{-}1),\ h_t,\ w_t\big), \\ M_{\text{read}}(t) &= \text{Read}\big(M(t),\ r_t\big). \end{aligned} \quad (5)$$

A common theme among these papers is the importance of memory augmentation in improving LLM performance. For instance, [20] proposed a unified framework for long-context large language models, reformulating existing methods and introducing a new approach (UniMix) that achieves superior performance in handling long contexts. Similarly, [93] presented

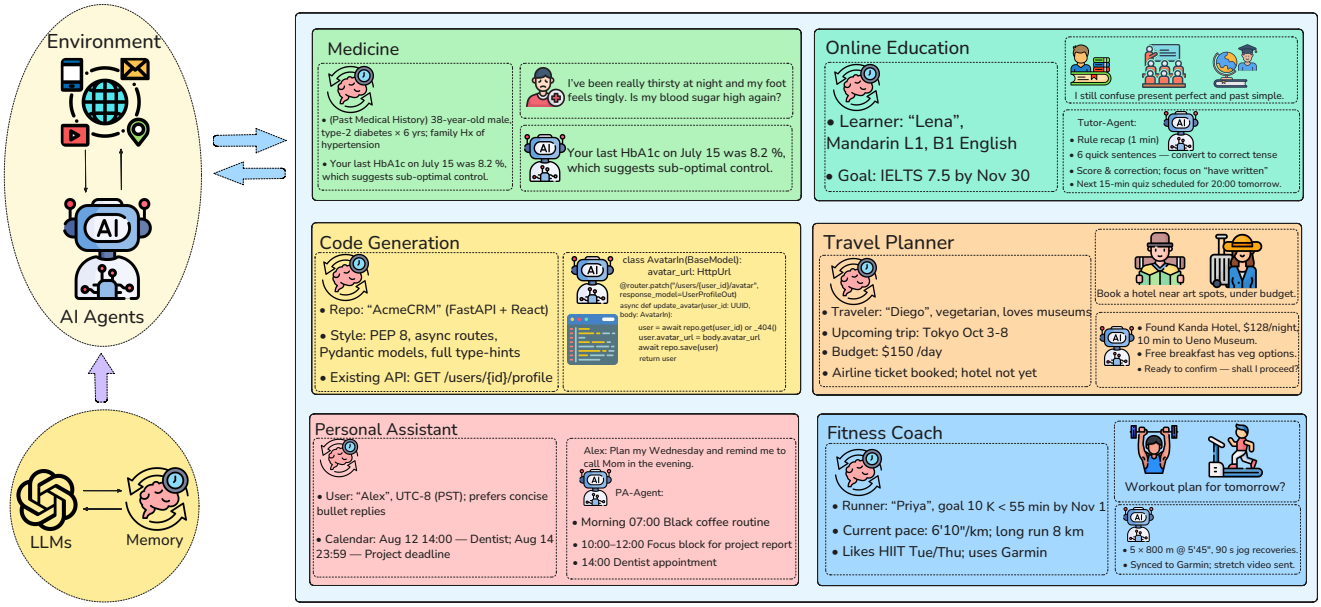

Fig. 2: Framework of memory-augmented LLM agents. The left side shows the interaction loop between environment, AI agents, and memory-enabled LLMs; the right side illustrates representative applications across domains such as medicine, education, code generation, personal assistance, travel, and fitness.

a framework that enables LLMs to memorize a long history, achieving remarkable improvements on memory-augmented in-context learning over LLMs. These studies demonstrate the effectiveness of memory augmentation in enhancing LLM capabilities.

A. Retrieval-Augmented Generation (RAG)

Recent research has increasingly drawn inspiration from human cognitive science to design more sophisticated memory systems for LLMs, particularly in the context of autonomous agents. A seminal work in this area introduced "Generative Agents," which are computational software agents that simulate believable human behavior by maintaining and reflecting on a persistent memory stream of their experiences [62]. This paradigm allows agents to plan, react, and form relationships in a dynamic environment. Building on this, the Reflexion framework equips agents with dynamic memory and self-reflection capabilities, enabling them to learn from past failures and refine their future actions without requiring updates to the model's weights [77]. Surveys on memory mechanisms highlight a shift from simple context windows to structured, human-like memory architectures, categorizing them into working memory, semantic memory, and episodic memory [98]. This cognitive inspiration is evident in frameworks like NEMORI, which proposes a self-organizing agent memory system [59], and A-MEM, which introduces an "agentic memory" framework designed for complex, long-horizon tasks [101]. To facilitate research and development, interactive environments like Memory Sandbox have been created to allow for transparent and interactive management of conversational agent memory [59].

Retrieval-Augmented Generation (RAG) has become a cornerstone for enhancing LLMs with external, up-to-date knowledge, thereby mitigating hallucinations and improving factual accuracy. The standard RAG pipeline involves retrieving relevant documents from a knowledge base and providing them as context to the LLM for generation [22]. Comprehensive surveys on RAG detail its evolution, from foundational models to advanced techniques involving sophisticated retrieval, generation, and augmentation strategies [81]. Innovations in this domain often focus on improving the efficiency and biological plausibility of the retrieval process. For instance, HippoRAG draws inspiration from the hippocampus to create a long-term memory system for LLMs, improving their ability to recall and reason over extended contexts [28]. The underlying principle of augmenting language models by retrieving from vast token databases was demonstrated by models like RETRO, which showed that scaling retrieval data can be more effective than scaling model parameters alone [6]. Architecturally, new models like the Retentive Network have been proposed as successors to the Transformer, offering a mechanism for retention and recurrence that is well-suited for long-sequence modeling [34]. However, challenges remain, as studies show that LLMs often struggle to effectively use information located in the middle of long contexts, a phenomenon known as the "lost in the middle" problem [49].

Multiple recent works have further expanded LLMs' memory capacities through specialized architectures and mechanisms. For example, Cognitive Workspace proposes an active memory management paradigm enabling functional "infinite" context windows via deliberate memory curation and hierarchical buffers, yielding significantly higher memory reuse

and efficiency compared to traditional retrieval-based methods [1]. MemoryBank introduces a long-term memory module that allows LLM-based agents to continually retain and recall relevant past interactions, using an Ebbinghaus-inspired forgetting mechanism to balance memory reinforcement and decay over time [119]. Similarly, RET-LLM defines a general read–write memory interface for LLMs to explicitly store extracted knowledge as interpretable triplets and retrieve it for reasoning, improving performance on knowledge-intensive and temporal question-answering tasks [55]. Another framework, SCM (Self-Controlled Memory), augments a transformer with an external memory stream and a controller that decides when and how to update or utilize the memory; this plug-and-play design enables ultra-long text processing and yields more informative outputs in lengthy dialogues without fine-tuning the base model [90].

Building on the idea of self-updatable models, MemoryLLM embeds a substantial fixed-size memory pool within an LLM's latent space, allowing the model to integrate new knowledge through memory writing operations and thus remain up-to-date without retraining [95]. Its extension, M+, significantly enhances long-term retention by introducing a scalable memory mechanism with a co-trained retriever, enabling knowledge recall over extremely long contexts with minimal additional GPU memory overhead [96]. Larimar offers a brain-inspired episodic memory module for one-shot knowledge updates and selective forgetting; it attains accuracy on sequential fact-editing benchmarks comparable to state-of-the-art methods but with 8–10× faster update speeds, thanks to its distributed memory that updates facts without expensive retraining [15]. In a different vein, CAMELoT attaches a training-free associative memory to a frozen LLM, consolidating new token representations on the fly by balancing novelty and recency; this approach enables arbitrarily long inputs and achieves up to 30% perplexity reduction on long-context modeling benchmarks without any model parameter updates [31].

For autonomous LLM based agents, specialized memory systems have also been proposed. MemInsight introduces an agent-specific memory augmentation framework that autonomously structures and updates an LLM agent's interaction history, leading to more contextually accurate responses [67]. RecallM provides an adaptable long-term memory architecture with a strong temporal understanding: it allows an LLM to update its stored beliefs over time and was shown to be four times more effective than a vector-database baseline at incorporating new facts, all while maintaining competitive general QA performance [43]. Additionally, MemTree organizes conversational memory into a dynamic tree of hierarchical summaries, enabling an LLM to manage long dialogues by abstracting and retrieving information at multiple levels of detail; this structured approach significantly improves multi-turn dialogue comprehension and long-document QA by linking new inputs to a schema of past contexts [65].

B. Advanced Paradigms in Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) has become a foundational technique for addressing the inherent limitations of static, parametric knowledge in LLMs. By conditioning generation on externally retrieved, up-to-date evidence, RAG systems can significantly improve factual consistency, enhance transparency, and reduce the incidence of model hallucinations [73]. This capability is particularly crucial for tasks requiring verifiable, domain-specific, or real-time information. However, the initial, straightforward paradigm of RAG has evolved into a sophisticated and modular field of study, with recent advancements focusing on enhancing robustness, efficiency, and adaptability to complex, real-world queries [13].

The progression of RAG architectures can be broadly categorized into three stages: Naive RAG, Advanced RAG, and Modular RAG [33]. This evolution reflects a growing understanding of the pipeline's complexities and failure points. Modern analyses deconstruct the RAG process into distinct, optimizable stages: pre-retrieval, retrieval, post-retrieval, and generation [116]. This modular perspective allows for targeted interventions to address specific challenges, such as retrieval noise, redundancy, and the potential misalignment between retrieved evidence and the final generated text.

III. Compression and Efficiency Optimization

The efficient deployment of Large Language Models (LLMs) is a crucial aspect of their practical application, as these models are often computationally intensive and require significant memory resources. To address this challenge, various compression techniques have been proposed, including quantization, pruning, and knowledge distillation [9], [114], [121]. These methods aim to reduce the computational and memory requirements of LLMs while maintaining their performance and accuracy. Aggressive post-training compression methods, for instance, have been shown to achieve significant reductions in model size with relatively small accuracy losses [114].

As illustrated in Figure 3, achieving simultaneous optimization across performance, inference efficiency, and memory footprint remains a core challenge for LLM research. Techniques such as quantization, pruning, and KV cache optimization substantially reduce memory usage and improve inference speed, but often at the expense of model accuracy and reasoning ability. In contrast, methods like knowledge distillation provide a more balanced trade-off, aiming to preserve performance while still reducing computational cost. This trilemma underscores why compression and optimization strategies must be carefully designed to balance these competing objectives rather than over-optimizing a single dimension.

A. Quantization Strategies

Quantization, in particular, is a widely adopted approach that effectively reduces model size and inference cost by lowering the precision of weights and activations. A formal

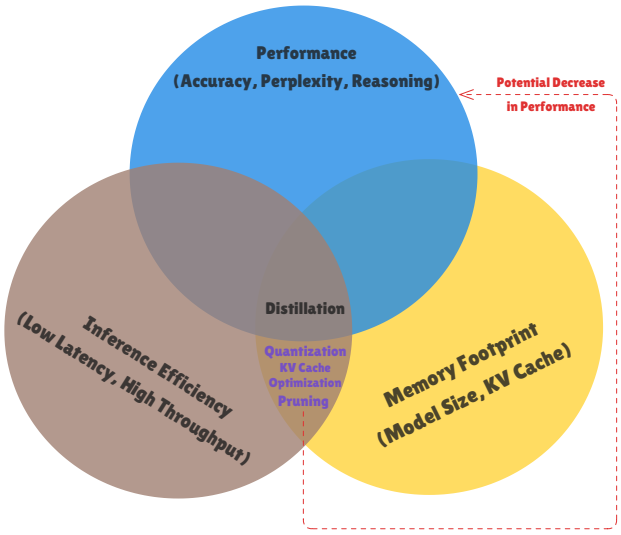

Fig. 3: Conceptual diagram of the LLM Efficiency Trilemma, showing trade-offs between performance, inference efficiency, and memory footprint.

description of b-bit symmetric quantization for a full-precision weight matrix $W \in \mathbb{R}^{m\times n}$ is:

$$W_q = s\,\text{clip}\Big(\text{round}\big(W/s\big), -2^{b-1}, 2^{b-1}-1\Big), \tag{6}$$

where s is a learnable scale factor, applied either per-tensor or per-channel, and clip$(\cdot)$ ensures that values remain within the quantization range. The quantization distortion is typically measured as:

$$E_q = \|W - W_q\|_F^2. \tag{7}$$

Despite its efficiency, even quantized LLMs may still be too large for edge or mobile devices, necessitating the use of data compression techniques to minimize I/O latency [50].

Algorithm 2 Symmetric b-bit Quantization Procedure

1: **Input:** Full-precision weight matrix $W \in \mathbb{R}^{m\times n}$, bit-width b
2: Compute scale: $s \leftarrow \max(|W|)/(2^{b-1}-1)$
3: Normalize: $\tilde{W} \leftarrow W/s$
4: Quantize: $W_q \leftarrow \text{clip}(\text{round}(\tilde{W}), -2^{b-1}, 2^{b-1}-1)$
5: Rescale: $W_q \leftarrow s \cdot W_q$
6: **Output:** Quantized matrix W_q

The development of new evaluation protocols is also essential to holistically assess the capabilities of compressed LLMs. Current state-of-the-art (SoTA) compression methods have limitations, such as significant performance degradation at moderate sparsity ratios and failure in knowledge-intensive tasks [35]. Moreover, the trade-off between model size reduction and accuracy loss is a common theme across various studies, highlighting the need for careful consideration of these factors when deploying compressed LLMs [9], [121]. The importance of evaluating compressed LLMs using diverse tasks and metrics is also emphasized in several papers, as this helps to ensure their effectiveness in real-world applications [35], [92].

Various technical approaches have been explored to achieve efficient deployment of LLMs, including the use of low-precision data types, such as int8, to reduce model size through quantization [36], [42]. Pruning methods, including network pruning and knowledge distillation, have also been discussed as effective techniques for reducing model complexity [16], [118]. Furthermore, the introduction of benchmarks like LLMC [24] and evaluation protocols such as LLM-KICK [35] provides a comprehensive framework for assessing the performance of compressed LLMs across various tasks, including language understanding, reasoning, generation, and in-context retrieval.

Recent advancements in quantization have pushed the boundaries of low-bit compression. AQLM (Activation-Quantized and Latency-aware Model) introduces a 2-bit quantization scheme that is optimized for hardware efficiency, demonstrating that extreme compression is feasible without catastrophic performance degradation [18]. Another approach improves quantization by focusing on Hadamard incoherence and information preservation, ensuring that the quantization process retains as much of the original model's representational capacity as possible [87].

In addition to these techniques, researchers have proposed several novel quantization and pruning methods to push LLM compression further. GPTVQ leverages the "blessing of dimensionality" to improve ultra-low bit quantization, showing that higher-dimensional weight spaces can be quantized more effectively and with less accuracy loss than expected by conventional wisdom [88]. ASER (Activation Smoothing and Error Reconstruction) introduces a quantization procedure that smooths the activation distribution and then reconstructs critical lost information, substantially reducing the performance degradation from aggressive weight quantization [117]. HWPQ (Hessian-free Weight Pruning-Quantization) presents a unified framework that prunes and quantizes weights jointly, guided by curvature information to minimize the increase in loss; this approach achieves high compression rates with minimal impact on LLM accuracy [38]. Meanwhile, SLiM combines one-shot quantization with low-rank sparsity induction in a single pipeline, using a low-rank approximation of weights to identify important components and prune the rest. SLiM is able to compress LLM weights in one pass without iterative retraining, attaining significant size reduction with only minor accuracy drops [58].

B. Low-Rank Approximations and KV Cache Optimization

Low-Rank Approximations and Progressive Compression for KV Cache Optimization have emerged as pivotal strategies in the quest to optimize memory usage in large language models (LLMs). Recent studies, including [72], [120], [30], [112], and [75], have delved into various aspects of KV cache optimization, underscoring the critical need for efficient memory utilization without compromising model performance.

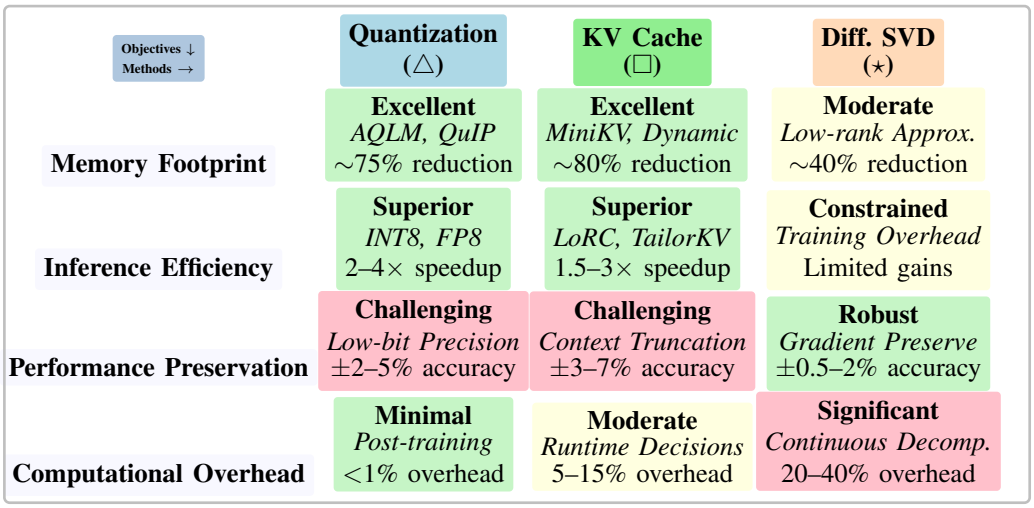

Fig. 4: **Evaluation matrix of LLM compression methods across optimization dimensions.** Pastel colors indicate effectiveness levels: green (strong), yellow (moderate), pink (challenging). Symbols distinguish method families: △ quantization, □ KV cache, ⋆ differentiable SVD.

As summarized in Figure 4, existing compression methodologies can be systematically compared across four major optimization dimensions: memory footprint, inference efficiency, performance preservation, and computational overhead. The matrix highlights that quantization and KV cache optimization are particularly effective for reducing memory and improving runtime efficiency, albeit with potential performance degradation at lower precision or under aggressive truncation. In contrast, differentiable SVD methods provide stronger robustness in preserving model accuracy, though often at the expense of higher computational overhead. This comparative view emphasizes that no single method universally dominates across all dimensions, underscoring the need for task- and resource-aware strategies when deploying large-scale LLMs.

A key finding in this domain is the introduction of novel compression techniques such as low-rank approximations and dynamic token retention. For instance, MiniKV [72] achieves an impressive 86% compression ratio with minimal accuracy loss by employing a 2-bit layer-discriminative KV cache, while DynamicKV [120] dynamically adjusts token retention based on task demands, retaining only a fraction of the KV cache size without significant performance degradation. These approaches not only highlight the potential for substantial memory savings but also demonstrate the feasibility of maintaining high model accuracy under compression.

One core strategy for KV cache compression is low-rank approximation of the key and value matrices using truncated singular value decomposition (SVD). Given a key matrix $K \in \mathbb{R}^{n\times d}$, it can be decomposed and truncated to rank r as:

$$K \approx U_K \Sigma_K V_K^\top, \tag{8}$$

$$K_{\text{compressed}} = U_K[:, 1{:}r]\, \Sigma_K[1{:}r, 1{:}r]\, V_K[:, 1{:}r]^\top, \tag{9}$$

where U_K, Σ_K, and V_K are the SVD components. The compression ratio (CR) for this scheme is:

$$\text{CR} = \frac{nd}{r(n+d+1)}, \tag{10}$$

offering a compact yet accurate approximation suitable for memory-constrained environments.

The concept of progressive compression, as introduced by LoRC [112], allows for a flexible and layer-wise adjustment of compression levels based on sensitivity analysis, offering a balanced approach between memory efficiency and performance preservation. This strategy, along with others like KCache [30] which alleviates memory bottlenecks during LLM inference without the need for KV Cache, points to a trend towards developing model-agnostic solutions that can integrate seamlessly with existing architectures.

The necessity for task-aware approaches is another theme that emerges prominently in this area of research. DynamicKV's [120] ability to adapt compression based on task-specific demands showcases the importance of understanding the nuanced requirements of different tasks and adjusting optimization strategies accordingly. In this context, token-level importance is measured to guide cache retention. The importance score of token i is defined as:

$$I_i = \sum_{j=1}^{h} \alpha_{ij} \|v_i\|_2, \tag{11}$$

where α_{ij} denotes the attention weight from head j to token i, and v_i is the value vector at position i. Tokens with low I_i scores are pruned first, enabling dynamic token retention while preserving critical information flow.

This is further reinforced by reviews such as [75], which emphasize the need for comprehensive evaluation metrics that capture both efficiency and long-text handling capabilities.

Technical innovations, including the development of specialized CUDA kernels for enhanced compatibility with architectures like FlashAttention [72], and methodologies such as low-rank approximations of KV weight matrices [112] and dynamic token retention strategies [120], have been instrumental in pushing the boundaries of what is achievable in KV cache optimization. These advancements not only contribute to the efficiency and scalability of LLMs but also pave the way for more sophisticated models that can handle complex tasks without prohibitive memory requirements.

The Key-Value (KV) cache is a major memory bottleneck during LLM inference. StreamingLLM addresses this by introducing "attention sinks"—initial tokens that anchor the attention mechanism—allowing models to handle infinite-length streams with a fixed-size cache [100]. The H2O (Heavy-Hitter Oracle) method further optimizes this by selectively evicting less important KV pairs from the cache, retaining only the heavy hitters that contribute most to the output [113]. Architecturally, LaCache proposes a ladder-shaped KV caching pattern that stores KV pairs both sequentially within layers and across layers, enhancing long-range dependency modeling under a fixed memory budget [74]. TailorKV introduces a hybrid framework that combines different optimization strategies based on layer characteristics for more efficient long-context inference [106].

C. Differentiable SVD for Model Compression

Differentiable SVD for LLM compression has emerged as a crucial technique in reducing memory requirements while preserving model performance, with several recent studies contributing to its development [92], [94], [47]. The core idea behind this approach is to leverage Singular Value Decomposition (SVD) to factorize weight matrices into smaller components, thereby facilitating compression.

Given a full-precision weight matrix $W \in \mathbb{R}^{m\times n}$, the standard SVD decomposes it as:

$$W = U\Sigma V^\top = \sum_i \sigma_i u_i v_i^\top, \tag{12}$$

where σ_i are the singular values, and u_i, v_i are the corresponding left and right singular vectors.

To enable compression, a truncated variant retains only the top-k singular modes:

$$W_k = \sum_{i=1}^{k} \sigma_i u_i v_i^\top, \tag{13}$$

introducing an approximation error bounded by the residual singular values: $\sum_{i>k} \sigma_i^2$.

However, SVD-based methods also introduce challenges such as information loss due to truncation and the need for efficient reconstruction strategies. A key finding in this area is the proposal of new approaches to address these challenges, including the use of activation truncation instead of optimization distance [92], and the introduction of truncation-aware data whitening and layer-wise closed-form model parameter updates [94]. These innovations have demonstrated superiority over state-of-the-art methods at high compression ratios, highlighting the potential for SVD-based techniques in LLM compression.

Recent approaches aim to make the SVD process differentiable and trainable, integrating low-rank factorization into model training or fine-tuning. A typical differentiable SVD formulation is:

$$\min_{U,V} \|W - UV^\top\|_F^2 + \lambda\|U\|_F^2 + \mu\|V\|_F^2, \tag{14}$$

where $U \in \mathbb{R}^{m\times r}$ and $V \in \mathbb{R}^{n\times r}$ are learnable low-rank factors, and λ, μ are regularization coefficients that promote stability and prevent overfitting.

Furthermore, adaptive SVD-based approaches that update singular matrices alternately and assign layer-specific compression ratios have shown promising results, outperforming existing methods with significantly reduced memory requirements [47]. A common theme among these studies is the emphasis on addressing limitations of existing SVD-based methods, such as information loss and uniform compression ratios. Techniques like adaptive compression ratio assignment [47] and low-rank delta approaches [54] demonstrate the importance of adaptivity and optimization in LLM compression.

As shown in Figure 5, the evolution of LLM compression techniques follows a layered trajectory. The initial challenge stems from the prohibitive computational and memory demands of large models, which motivated first-order solutions such as pruning, quantization, and knowledge distillation. Among these, low-rank approximation methods like SVD and LoRA provide a principled avenue for reducing parameter complexity, but also suffer from practical drawbacks, including truncation-induced information loss and lack of end-to-end trainability. These shortcomings have driven the emergence of refined approaches, such as differentiable SVD and hybrid compression pipelines, which aim to balance efficiency with accuracy preservation. This progression illustrates how the community incrementally addresses both theoretical and practical bottlenecks in LLM compression.

Beyond static SVD, recent works have explored more adaptive low-rank factorization methods. SVD-LLM proposes a truncation-aware SVD method that directly relates singular values to compression loss, enabling more precise control over the accuracy-compression trade-off [63], while LoRAP differentiates compression strategies across transformer sublayers, applying structured compression more aggressively where it is less impactful [45].

D. Technical Comparison and Trade-offs

The technical comparison of Large Language Model (LLM) compression and optimization methods reveals diverse strategies and trade-offs. Existing works explore techniques ranging from model-based compression to token-level input compression, with varying emphasis on efficiency, accuracy preservation, and hardware deployability.

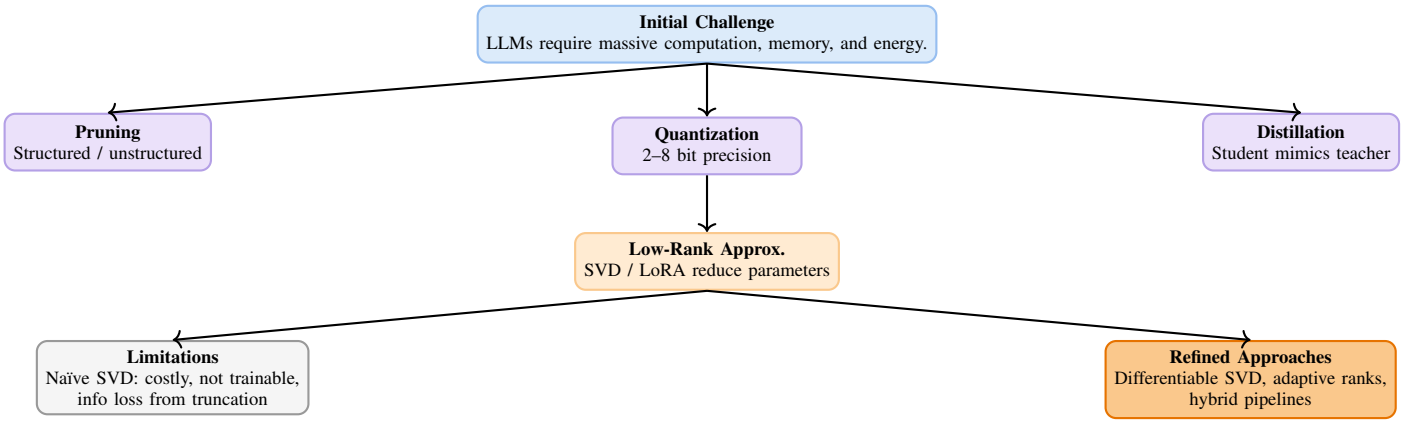

Fig. 5: Landscape of LLM compression. The figure traces the path from the **initial challenge** of computational cost to **first-order solutions** (pruning, quantization, distillation), then highlights the role of **low-rank approximation** (e.g., SVD, LoRA), and finally contrasts its **limitations** with **refined approaches** such as differentiable SVD and hybrid pipelines.

1) Compression Efficiency and Memory Savings: DeltaLLM [54] demonstrates that post-training compression can achieve a 12% reduction in memory footprint while retaining approximately 90% of original benchmark performance. In contrast, token-level approaches such as Compressed Tokens [115] achieve up to 15× compression by introducing uniformly spread position identifiers and a tailored compression loss.

$$\text{CR} = \frac{|W|_{\text{orig}}}{|W|_{\text{comp}}}, \tag{15}$$

where $|W|_{\text{orig}}$ denotes the original parameter size and $|W|_{\text{comp}}$ the compressed size.

Beyond general-purpose compression, some methods target specific deployment scenarios and introduce innovative algorithms. EDGE-LLM combines unified compression (blending pruning, quantization, and distillation) with an adaptive layer-voting mechanism to enable efficient on-device adaptation of LLMs [108]. This approach allows large models to run on edge hardware with minimal performance loss. CompactifAI exploits quantum inspired tensor network decompositions to achieve extreme compression of LLMs, restructuring model parameters into compact tensor cores and attaining compression ratios far beyond those of conventional methods [86]. EvoPress formulates dynamic model compression as an evolutionary search problem: it uses genetic algorithms to automatically find near-optimal compression policies that balance size and accuracy [78]. Lillama, compresses a model via low-rank feature distillation—it trains a much smaller student LLM to mimic the internal representations of a large model, yielding a highly compact model that preserves the original's performance on downstream tasks to a remarkable degree [82].

2) Benchmarking and Evaluation Protocols: Evaluation studies such as Zero-Shot Long-Context LLM Compression [91] and LLMCBench [103] emphasize rigorous benchmarking. Long-context reasoning benchmarks highlight error accumulation at scale, while LLMCBench provides standardized multi-task evaluation.

3) Trade-off Characterization: Compression introduces a fundamental balance between compactness and predictive performance. This trade-off can be expressed via the Pareto frontier:

$$\mathcal{P} = \{(c, p) \mid \nexists (c', p') : c' < c \wedge p' > p\}. \tag{16}$$

Beyond algorithmic innovations, researchers have begun to evaluate compressed LLMs from broader perspectives and provide theoretical insights. For instance, the Beyond Perplexity framework advocates for multi-dimensional safety and reliability evaluation of compressed models, examining how aggressive compression impacts factors like factuality, bias, and robustness in addition to traditional metrics [102]. Theoretical analysis has also shed light on compression's effect on model capabilities: one study proved that applying SVD-based weight pruning can actually enhance an LLM's in-context learning performance by preserving critical singular vectors, thus maintaining ability to adapt to new prompts after compression [107]. Finally, comprehensive surveys of transformer compression techniques have emerged to synthesize these developments—mapping out quantization, pruning, distillation, and low-rank methods and to discuss their trade-off in accuracy, memory footprint, and computational cost [84]. Such surveys provide a structured overview of the state of the art and highlight open challenges for deploying compressed LLMs at scale.

4) Deployment and Infrastructure Challenges: At moderate sparsity ratios, accuracy degradation remains significant [114]. Moreover, practical deployment requires hardware-software co-design [46], [89], complicating use on edge devices. Broad evaluation across tasks is also needed [35], [92].

IV. Alignment and Personalization

Personalized evaluation and alignment with human preferences is a crucial aspect of Memory-Augmented Large Language Models (LLMs), as it enables these models to adapt to

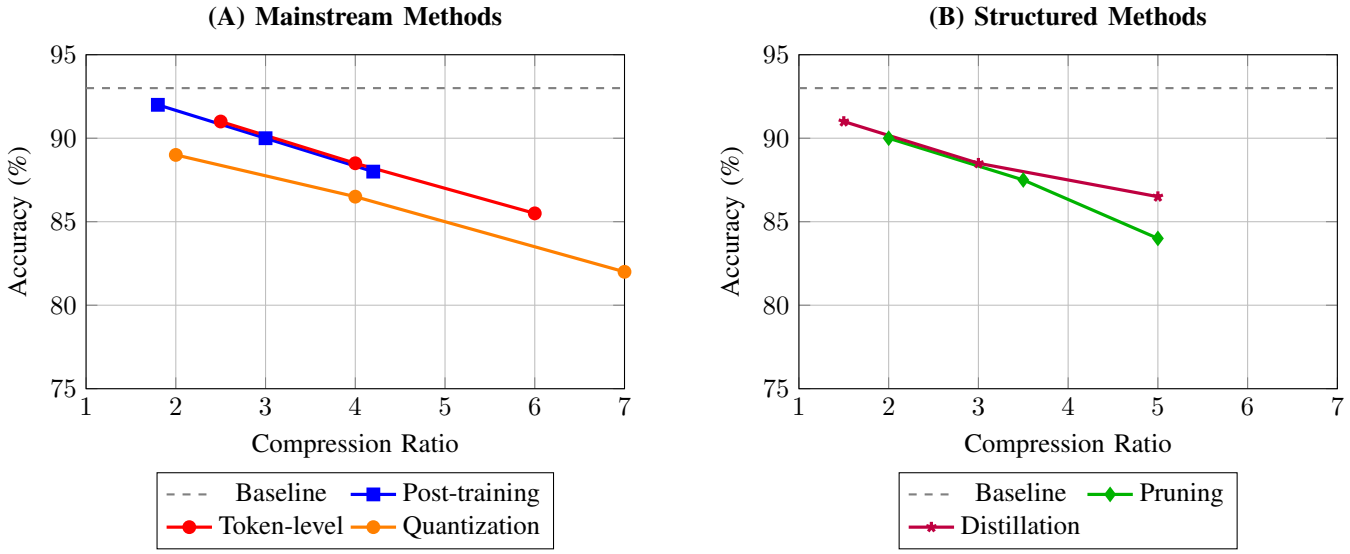

Fig. 6: Comparison of LLM compression strategies. (A) Mainstream methods (post-training, token-level, quantization). (B) Structured methods (pruning, distillation).

individual user needs and provide more accurate and relevant responses. Recent studies have proposed diverse strategies, ranging from supervised fine-tuning to reinforcement learning and inference-time control [5], [10], [71], [110], [122]. The central challenge lies in efficiently capturing the heterogeneity of human values and incorporating them into LLMs while balancing computational cost and generalization.

Personalization of LLMs to individual users has gained attention as an extension of alignment. Recent works explore on-device fine-tuning and profile-based customization to adapt a outputs to a specific user's preferences and context. For example, an adaptive on device learning approach continually updates an LLM using self-supervised signals from the user's interactions, enabling dynamic personalization without relying on extensive labeled data or server side training [51]. Other methods focus on representing user traits explicitly: one framework generates a synthetic user profile from past dialogues and then guides the LLM's responses with this profile, significantly improving personalization in style and content relevance [111]. In a similar vein, the Persona-Plug technique conditions generation on a learned persona embedding, allowing a base LLM to produce responses tailored to a given personality or role through lightweight prompt tuning [48]. These profile-driven strategies make it possible for a single model to serve many users while respecting their individual preferences.

Domain-specific personalization strategies have also emerged. LLM-Personalize introduces a reinforced self-training loop for aligning an LLM-based planner with human preferences in robotics; by simulating human feedback and iteratively refining plans, it aligns a household robot's instructions with end-user expectations without requiring costly human-in-the-loop data for each round [29]. In healthcare, researchers demonstrate a pipeline for tailoring a general LLM to a specific patient or clinical dataset. This general-to-specific adaptation yields more contextually appropriate medical advice and better diagnostic accuracy by incorporating personal health records into knowledge [76]. Another cutting-edge approach treats personalization as a contextual bandit problem: an LLM can be equipped with a neural bandit mechanism that continuously learns from a user's feedback on the fly [11]. This online personalizer updates the generation policy in real time, enabling rapid and fine-grained personalization in interactive applications.

A. Core Training Paradigms for Alignment

Early work relies on *Supervised Fine-Tuning* (SFT), where models are trained directly on labeled input–output pairs. More advanced strategies such as Reinforcement Learning with Human Feedback (RLHF), Direct Preference Optimization (DPO), and Reinforcement Learning from AI Feedback (RLAIF) aim to move beyond static labels toward preference-based optimization [21], [110]. These methods introduce a learned reward model r_ϕ that encodes human preference signals, offering stronger adaptability but at the cost of increased training and inference complexity.

To better illustrate the trade-offs among different compression strategies, Figure 6 compares mainstream approaches (post-training, token-level, and quantization) with structured methods (pruning and distillation). Mainstream methods generally achieve higher compression ratios but suffer from accuracy degradation at larger scales, while structured techniques provide more stable accuracy retention with moderate compression. Table I further summarizes these methods, highlighting their typical compression ratios, accuracy retention, computational complexity, and hardware requirements. Together, the figure and table reveal that no single approach dominates across all dimensions: quantization and token-level compression maximize efficiency, whereas pruning and distil-

TABLE I: Summary of LLM compression methods: trade-offs between compression ratio, accuracy retention, and practical considerations.

Method	Typical CR	Accuracy Retention	Complexity	Hardware Need	Remarks
Post-training (DeltaLLM)	2×–4×	High (∼90%)	Low	None	Good balance of simplicity and retention
Token-level Compression	10×–15×	Moderate–High	Medium	None	Strong for input-level efficiency
Quantization	4×–8×	Medium	Low–Medium	Often	Sensitive to precision loss
Pruning / Sparsity	2×–6×	Medium	High	Specialized	Accuracy loss at moderate ratios
Knowledge Distillation	2×–5×	Medium–High	Medium	None	Produces compact student models

lation emphasize balanced performance at the cost of higher complexity.

Formally, given a prompt x and two candidate responses (y^+, y^-), the reward model is trained with the pairwise ranking loss:

$$\mathcal{L}_{\text{align}} = -\log \sigma\big(r_\phi(x, y^+) - r_\phi(x, y^-)\big), \tag{17}$$

where σ is the sigmoid function. The aligned reward then guides policy updates of the LLM parameters θ via policy optimization such as PPO or DPO.

Algorithm 3 Preference Alignment Training Loop

1: **Input:** Prompt x, preferred y^+, dispreferred y^-
2: Train reward model r_ϕ with pairwise loss $\mathcal{L}_{\text{align}}$
3: Use r_ϕ to guide policy update (e.g., PPO [21], DPO [110])
4: Update LLM parameters θ accordingly
5: **Output:** Aligned model θ

This loop (Algorithm 3) captures the essence of preference-based alignment. In practice, stability and sample efficiency are major concerns: RLHF often suffers from reward hacking, while DPO trades off explicit exploration for simplicity. An important research direction is therefore the design of more robust objectives, e.g., adding KL regularization or constraint penalties:

$$\mathcal{L}_{\text{total}} = \mathcal{L}_{\text{align}} + \beta\, \text{KL}(\pi_\theta \parallel \pi_{\text{ref}}), \tag{18}$$

where π_{ref} is a reference model.

The field of preference alignment has rapidly evolved beyond the initial RLHF and DPO paradigms. SimPO (Simple Preference Optimization) simplifies the objective by introducing a reference-free reward mechanism, making the training process more stable and efficient [52]. The concept of Constitutional AI provides a framework for instilling principles a constitution into the model, allowing it to self-correct and align its behavior without continuous human feedback, primarily by using AI-generated feedback for harmlessness [4]. RLCD (Reinforcement Learning from Contrastive Distillation) generates preference pairs by contrasting model outputs from positive and negative prompts, creating cleaner training signals for the reward model [105].

Furthermore, the reliance on static, offline preference datasets is being challenged. Online AI Feedback (OAIF) methods use a powerful LLM as a live annotator during the alignment process, providing fresh, on-policy feedback that adapts as the model learns [26]. Other novel approaches incorporate principles from behavioral economics, such as Kahneman-Tversky Optimization, which models human risk aversion and loss aversion in the preference function [19]. Variants of DPO, such as β -DPO, introduce dynamic calibration of the trade-off parameter to improve stability and performance [97]. Self-Play Preference Optimization (SPPO) enables the model to generate its own training data by playing against itself, refining its capabilities without requiring external, stronger models [99].

B. Data Augmentation for Human Evaluator Alignment

Human-annotated preference datasets are expensive and limited. Data augmentation expands a small seed dataset into larger corpora by applying transformations $\{\mathcal{T}_i\}$ such as paraphrasing, back-translation, or synthetic instruction generation [25], [41], [71]. Filtering mechanisms $\mathcal{F}$ ensure quality by removing toxic or redundant samples.

Mathematically, the augmented dataset is:

$$\mathcal{D}_{\text{aug}} = \mathcal{D}_{\text{seed}} \cup \bigcup_{i=1}^{k} \{x' \mid x' = \mathcal{T}_i(x), \mathcal{F}(x') = 1\}. \tag{19}$$

To quantify quality improvements, many works evaluate Pearson correlation r between model scores and human judgments:

$$r = \frac{\sum_i (y_i - \bar{y})(\hat{y}_i - \bar{\hat{y}})}{\sqrt{\sum_i (y_i - \bar{y})^2} \sqrt{\sum_i (\hat{y}_i - \bar{\hat{y}})^2}}. \tag{20}$$

Algorithm 4 Data Augmentation Pipeline for Alignment

1: **Input:** Seed dataset $\mathcal{D}_{\text{seed}}$, transformations $\{\mathcal{T}_i\}$, filtering $\mathcal{F}$
2: Initialize $\mathcal{D}_{\text{aug}} \leftarrow \mathcal{D}_{\text{seed}}$
3: **for** each $x \in \mathcal{D}_{\text{seed}}$ **do**
4: **for** each $\mathcal{T}_i$ **do**
5: Generate candidate $x' \leftarrow \mathcal{T}_i(x)$
6: **if** $\mathcal{F}(x') = 1$ **then**
7: Add x' to $\mathcal{D}_{\text{aug}}$
8: **end if**
9: **end for**
10: **end for**
11: **Output:** Augmented dataset $\mathcal{D}_{\text{aug}}$

Augmented datasets have demonstrated up to 7% improvement in alignment correlation with human judges [71]. Advanced strategies like information gain maximization [41] or self-alignment [25] further reduce reliance on manual labels.

C. Inference-Time Personalization

Not all personalization requires retraining. Inference-time alignment enables large language models (LLMs) to dynamically adapt to user-specific preferences during decoding [5],

[10]. Given a preference function $r(\cdot)$ defined at token- or sequence-level, the logits are reweighted before sampling:

$$p'(y_t) \propto p_\theta(y_t|x, y_{<t}) \cdot \exp(\lambda\, r(y_t)), \tag{21}$$

where λ balances fluency and preference satisfaction.

This decoding-time mechanism enables *plug-and-play personalization* without retraining. For instance, abductive reasoning over user personas [5] allows implicit preference inference, while soft-constraint decoding [10] aligns outputs with diverse user preferences. Such methods are particularly promising in real-world deployment where computational budgets are tight.

To contextualize inference-time personalization, we compare five alignment strategies: *Supervised Fine-Tuning (SFT)*, *RLHF*, *DPO*, *RLAIF*, and *Inference-time Control*. They are evaluated across five key dimensions: efficiency, personalization, stability, data demand, and compute cost. Figures 7 and 8 present the comparison and workflow.

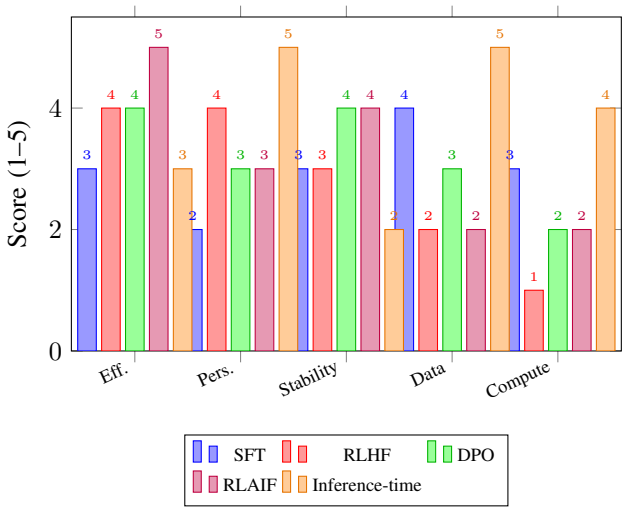

Fig. 7: Comparison of alignment methods across five dimensions. Scores range from 1 (low) to 5 (high).

As shown in Figure 7, five alignment strategies exhibit different trade-offs: SFT offers stability but weak personalization, RLHF and DPO achieve balanced performance at higher cost, while RLAIF improves scalability by leveraging AI feedback. Complementing these methods, Figure 8 illustrates inference-time personalization, which enables lightweight yet powerful adaptation during decoding, providing efficiency and flexibility though with varying stability.

V. Interpretability and Real-World Applications

A. Localization and Memorization

Understanding how LLMs store and retrieve memorized data is a central interpretability challenge, with direct ethical and legal implications [69]. Recent work proposes localization benchmarks such as INJ and DEL [8], designed to identify LLM components responsible for storing sensitive content. Results suggest that methods adapted from network pruning achieve competitive performance, yet highlight that evaluation

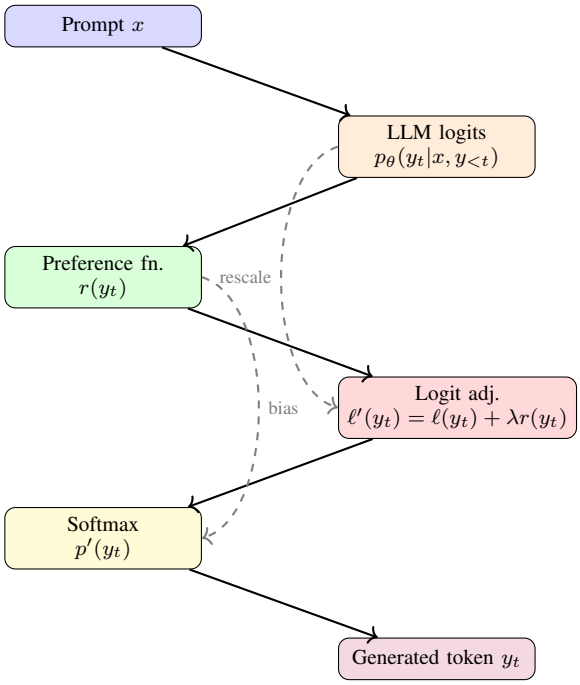

Fig. 8: Inference-time personalization workflow. Different colors highlight functional modules: input (blue), LLM computation (orange), preference modeling (green), logit adjustment (red), sampling (yellow), and output (purple). Dashed arrows indicate auxiliary influences such as bias or rescaling.

design critically influences outcomes. Beyond parametric localization, episodic memory benchmarks like SORT link memories to their temporal and contextual embeddings, offering new perspectives on sequence recall and context retention [64]. Together, these benchmarks emphasize that localization is both technically complex and evaluation-dependent.

B. Privacy, Security, and Ethical Risks

Despite progress, major limitations persist. Existing localization methods often fail to reliably pinpoint the precise components of memorization, partly due to the architectural complexity of LLMs [53]. This raises unresolved questions about interpretability, transparency, and compliance. Balancing performance with privacy remains an open challenge: models must retain utility while preventing unintended leakage of personal or proprietary data [69]. Moreover, domain-specific risks in conversational agents and multilingual models demand more nuanced interpretability frameworks [56], underscoring the socio-technical stakes of memorization research.

C. Applications and Integration

Interpretability advances also have practical consequences for real-world deployment. Explicit memory modules and attention-based mechanisms have been explored as tools to make memorization more transparent and controllable [53], [56]. Comprehensive benchmarks, such as SORT, INJ, and DEL [8], offer systematic evaluation pathways, enabling more trustworthy integration of LLMs into sensitive applications.

TABLE II: Comprehensive Analysis of Memory-Augmented LLM Applications (with Representative Models)

Domain	System / Method	Representative Models	Evaluation Metric	Latency Metric	Resource Usage	Limitations
Business Documents	Matrix	LLaMA (7B/13B), GPT-4, Claude, DocLLM	Accuracy, F1	Time to First Token	Memory Usage	Domain-specific training and limited transferability
Long-Context QA	UniMix	Claude-2/3, GPT-4 Turbo (128k), Gemini 1.5, MPT-30B	Perplexity, Recall/F1	Per-Token Latency	Memory Usage	Context length limitations and memory staleness
Conversational AI	MemLLM	GPT-3.5/4, Claude-2/3, LLaMA-2-Chat, Vicuna	Accuracy, Coherence Score	First Token Latency	Memory Usage	Limited updates and privacy concerns
Code Generation	Enhanced Copilot	CodeT5, Codex, StarCoder, Code LLaMA	BLEU, Execution Accuracy	Per-Token Latency	Memory Usage	Language-specific adaptation and limited reasoning
Scientific Literature	LongLLaMA+	Galactica, SciBERT, LongLLaMA+, PubMedGPT	ROUGE, Citation Recall	Time to First Token	GPU Memory Usage	High computational cost and hallucination
Customer Support	ChatMem	GPT-4, Claude-2, Gemini, LLaMA-2 Chat	Resolution Rate, Satisfaction Score	First Token Latency	Memory Usage	Dependency on training data and multilingual issues
Educational Tutoring	AdaptiveTeach	PaLM-2, Gemini, GPT-4, TutorLM	Learning Improvement, Engagement	Per-Token Latency	Memory Usage	Complex personalization and assessment challenges
Medical Diagnosis	MedMem-AI	BioGPT, ClinicalBERT, Med-PaLM, PubMedGPT	Diagnostic Accuracy, Recall	Time to First Token	Memory Usage	Regulatory compliance and liability risks
Financial Analysis	FinanceGPT+	BloombergGPT, FinGPT, GPT-4, LLaMA-2 (Finance)	Prediction Accuracy, Risk Assessment	Per-Token Latency	Memory Usage	Market volatility and real-time integration limits
Legal Research	LegalMind	GPT-4, Legal-BERT, LLaMA-2 70B, Lawformer	Relevance, Citation Accuracy	Time to First Token	Memory Usage	Jurisdiction variation and interpretation issues

These developments hold promise for high-stakes domains including healthcare, finance, and legal systems, where transparency and accountability are non-negotiable. Future research will need to further develop localization-aware training pipelines and integrate interpretability insights into system-level design [8], [69].

Addressing interpretability in specific application domains has become a priority. In finance, for example, researchers have proposed interpretable LLM frameworks for credit risk assessment, including a taxonomy of explanation techniques suited to banking requirements and regulatory compliance [23]. In the education domain, efforts to combine LLMs with symbolic reasoning have shown promise for greater transparency; notably, an XAI challenge explored how explaining answers in an educational QA system can be improved by integrating symbolic steps, yielding insights into effective strategies for making LLM reasoning more scrutable in tutoring scenarios [61]. More broadly, comprehensive surveys are beginning to map out the landscape of explainability and trustworthiness for LLM-powered decision support in sensitive fields, highlighting best practices to ensure fairness, accountability, and transparency [2]. As LLM technology extends beyond text into multi-modal settings, recent work has surveyed techniques for making multimodal LLMs more interpretable, noting that the fusion of visual and textual information poses additional challenges for explanation consistency and user understanding [14]. These studies underscore that interpretability solutions often need to be tailored to the context and stakes of the deployment domain.

On the methodology front, new techniques aim to imbue LLMs with greater transparency without significantly sacrificing performance. One line of work focuses on grounded explanation generation: for instance, the XplainLLM dataset was constructed to fine-tune LLMs to produce faithful, knowledge-grounded explanations, which improved the reliability of model-provided justifications in experiments [12]. Another approach introduces concept bottlenecks into LLMs, where the model is trained to predict a set of human-interpretable concepts as an intermediate step before the final answer; this design forces reasoning to pass through an explainable bottleneck and has been shown to maintain accuracy while providing insight into decision factors [79]. There are also techniques for inference time intervention: a sparsity guided explanation framework prunes less important neurons or attention heads and then observes the impact on outputs, yielding holistic explanations for model decisions and even allowing users to interactively test "what-if" scenarios on the reasoning process [83]. Model agnostic tools are emerging as well. For example, DLBacktrace can be applied to any deep model (including LLMs) to trace an output prediction back to the most influential parameters and training examples, thereby offering a post-hoc explanation of why the model produced a given output [68]. Finally, researchers are formulating general design principles for interpretability. Such as architectural modifications or training regimes that inherently yield more transparent behavior summarized in recent work on crafting LLMs for interpretability [80]. By incorporating these best practices, future LLMs can be built with interpretability as a first-class target alongside task performance.

VI. Open Challenges and Emerging Opportunities

The landscape of Large Language Models (LLMs) is rapidly evolving, with significant advancements in foundational principles, applications, training processes, and ethical considerations [57], [60]. Despite these developments, several open challenges and future directions emerge as crucial for the continued progress of LLM research. A key area of focus is the integration of explicit memory mechanisms into LLMs, such as proposed by MemLLM [56], which aims to improve performance on knowledge-related tasks, interpretability, and the ability to memorize rare events and update facts over time. This concept of memory augmentation is identified

as a promising direction for enhancing model performance, particularly in knowledge-intensive tasks.

Furthermore, incorporating human feedback into LLM training and ensuring that these models align with human values and preferences are recognized as critical aspects of future research [40]. The role of data augmentation in enhancing model performance is also explored, with strategies such as controllable and multi-modal data augmentation presenting both opportunities and challenges [17].

The socio-technical impacts, constraints, and emerging questions in LLM development are systematically investigated in [40], highlighting the importance of responsible development, algorithmic improvements, ethical challenges, and societal implications. This emphasis on socio-technical aspects contrasts with more technically focused papers, such as [57] and [60], which delve into technical advancements. The introduction of novel approaches like MemLLM [56] and the exploration of retrieval augmented generation [17] also showcase the diversity of methodological approaches in addressing LLM challenges.

Technical details and methodologies discussed across these papers include in-context learning and fine-tuning approaches, reinforcement learning frameworks that incorporate human feedback, and explicit read-write memory modules [40], [56]. Despite these advancements, limitations and challenges facing LLM research are evident, including ethical considerations and responsible deployment, interpretability and transparency, efficiency and scalability, data quality and availability, and adaptability to new information [40], [57], [60].

Another emerging opportunity is to rethink the transformer architecture itself in order to improve efficiency and reasoning capabilities. Several works propose novel attention mechanisms that break the scaling bottlenecks of standard self-attention. For instance, LevAttention introduces a time and space efficient algorithm for handling extremely long sequences, using a leveled approach that significantly lowers attention complexity while preserving performance on long-context tasks [39]. Other research aims to decouple knowledge and reasoning within LLMs: one modular architecture separates the transformer into a knowledge store and a reasoning module connected by generalized cross-attention, which improves interpretability and allows independent scaling of knowledge capacity versus reasoning power [27]. Similarly, graph aware attention mechanisms have been proposed to handle structured or time varying input data; a recent isomorphic graph attention model enables the transformer to dynamically adjust to evolving relational structures in the data, enhancing its adaptability to complex sequential dynamics [7]. On the efficient inference front, techniques like ZETA use Z-order curve hashing to quickly identify top-k relevant keys for each query, effectively implementing a sparse attention that avoids the quadratic cost of vanilla attention and accelerates inference on long inputs [109]. In fact, some approaches seek to break the attention bottleneck entirely by designing new architectures that do not rely on pairwise token attention; for example, one such architecture reimagines the attention mechanism to grow sublinearly with sequence length, enabling LLMs to process very long texts with orders-of-magnitude lower computational cost [32]. These architectural innovations are still in early stages, but they represent promising paths toward more scalable and efficient LLMs that can handle long horizon reasoning and knowledge integration beyond the current limits of transformer-based models.

VII. Conclusion

The future of LLM research is marked by both significant opportunities and challenges. Emerging trends such as memory augmentation, efficiency optimization, and socio-technical considerations will play crucial roles in shaping the development of LLMs. Addressing the identified limitations and challenges through innovative methodologies and a comprehensive understanding of LLMs' technical, ethical, and societal implications will be essential for realizing the full potential of these models [40], [57], [60]. As research in this area continues to evolve, it is critical to maintain a balanced approach that considers both the technical advancements and the broader socio-technical context in which LLMs are developed and deployed.

References

[1] Tao An. Cognitive workspace: Active memory management for llms–an empirical study of functional infinite context. *arXiv preprint arXiv:2508.13171*, 2025.

[2] Ayman Asaad, AM Azizul Hassan Chy, Anzam Shahriar Kabir, Amrun Nakib, and Nazifa Tabassum. Navigating the labyrinth: A review of explainability and trustworthiness in large language model-powered systems for sensitive decision-making. *Scientia. Technology, Science and Society*, 2(7):5–19, 2025.

[3] Guangji Bai, Zheng Chai, Chen Ling, Shiyu Wang, Jiaying Lu, Nan Zhang, Tingwei Shi, Ziyang Yu, Mengdan Zhu, Yifei Zhang, Xinyuan Song, Carl Yang, Yue Cheng, and Liang Zhao. Beyond efficiency: A systematic survey of resource-efficient large language models, 2024.

[4] Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, et al. Constitutional ai: Harmlessness from ai feedback. *arXiv preprint arXiv:2212.08073*, 2022.

[5] Nishant Balepur, Vishakh Padmakumar, Fumeng Yang, Shi Feng, Rachel Rudinger, and Jordan Lee Boyd-Graber. Whose boat does it float? improving personalization in preference tuning via inferred user personas, 2025.

[6] Sebastian Borgeaud, Arthur Mensch, Jordan Hoffmann, Trevor Cai, Eliza Rutherford, Katie Millican, George van den Driessche, Jean-Baptiste Lespiau, Bogdan Damoc, Aidan Clark, Diego de Las Casas, Aurelia Guy, Jacob Menick, Roman Ring, Tom Hennigan, Saffron Huang, Loren Maggiore, Chris Jones, Albin Cassirer, Andy Brock, Michela Paganini, Geoffrey Irving, Oriol Vinyals, Simon Osindero, Karen Simonyan, Jack W. Rae, Erich Elsen, and Laurent Sifre. Improving language models by retrieving from trillions of tokens, 2022.

[7] Markus J Buehler. Graph-aware isomorphic attention for adaptive dynamics in transformers. *APL Machine Learning*, 3(2), 2025.

[8] Ting-Yun Chang, Jesse Thomason, and Robin Jia. Do localization methods actually localize memorized data in llms? a tale of two benchmarks. 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 3190–3211, Mexico City, Mexico, 2024. Association for Computational Linguistics.

[9] Arnav Chavan, Raghav Magazine, Shubham Kushwaha, Mérouane Debbah, and Deepak Gupta. Faster and lighter llms: A survey on current challenges and way forward. In *Proceedings of the Thirty-Third International Joint Conference on Artificial Intelligence (IJCAI)*, pages 7980–7988, 2024. Survey Track.

[10] Ruizhe Chen, Xiaotian Zhang, Meng Luo, Wenhao Chai, and Zuozhu Liu. Pad: Personalized alignment of llms at decoding-time, 2024.

[11] Zekai Chen, Po-Yu Chen, and Francois Buet-Golfouse. Online personalizing white-box llms generation with neural bandits. In *Proceedings of the 5th ACM International Conference on AI in Finance*, pages 711–718, 2024.

[12] Zichen Chen, Jianda Chen, Ambuj Singh, and Misha Sra. Xplainllm: A knowledge-augmented dataset for reliable grounded explanations in llms. *arXiv preprint arXiv:2311.08614*, 2023.

[13] Mingyue Cheng, Yucong Luo, Jie Ouyang, Qi Liu, Huijie Liu, Li Li, Shuo Yu, Bohou Zhang, Jiawei Cao, Jie Ma, Daoyu Wang, and Enhong Chen. A survey on knowledge-oriented retrieval-augmented generation, 2025.

[14] Yunkai Dang, Kaichen Huang, Jiahao Huo, Yibo Yan, Sirui Huang, Dongrui Liu, Mengxi Gao, Jie Zhang, Chen Qian, Kun Wang, et al. Explainable and interpretable multimodal large language models: A comprehensive survey. *arXiv preprint arXiv:2412.02104*, 2024.

[15] Payel Das, Subhajit Chaudhury, Elliot Nelson, Igor Melnyk, Sarath Swaminathan, Sihui Dai, Aurélie Lozano, Georgios Kollias, Vijil Chenthamarakshan, Soham Dan, et al. Larimar: Large language models with episodic memory control. *arXiv preprint arXiv:2403.11901*, 2024.

[16] Lucio Dery, Steven Kolawole, Jean-François Kagy, Virginia Smith, Graham Neubig, and Ameet Talwalkar. Everybody prune now: Structured pruning of llms with only forward passes, 2024.

[17] Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. Data augmentation using llms: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 97–115, Bangkok, Thailand, 2024. Association for Computational Linguistics.

[18] Vage Egiazarian, Andrei Panferov, Denis Kuznedelev, Elias Frantar, Artem Babenko, and Dan Alistarh. Extreme compression of large language models via additive quantization, 2024. *URL https://arxiv.org/abs/2401.06118*, 63, 2024.

[19] Kawin Ethayarajh, Winnie Xu, Niklas Muennighoff, Dan Jurafsky, and Douwe Kiela. Kto: Model alignment as prospect theoretic optimization. *arXiv preprint arXiv:2402.01306*, 2024.

[20] Junjie Fang, Likai Tang, Hongzhe Bi, Yujia Qin, Si Sun, Zhenyu Li, Haolun Li, Yongjian Li, Xin Cong, Yankai Lin, Yukun Yan, Xiaodong Shi, Sen Song, Zhiyuan Liu, and Maosong Sun. Unimem: Towards a unified view of long-context large language models. In *Proceedings of the First Conference on Language Modeling (COLM)*, Online, 2024. Poster.

[21] Zhangyin Feng, Weitao Ma, Weijiang Yu, Lei Huang, Haotian Wang, Qianglong Chen, Weihua Peng, Xiaocheng Feng, Bing Qin, and Ting liu. Trends in integration of knowledge and large language models: A survey and taxonomy of methods, benchmarks, and applications, 2023.

[22] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. Retrieval-augmented generation for large language models: A survey, 2024.

[23] Muhammed Golec and Maha AlabdulJalil. Interpretable llms for credit risk: A systematic review and taxonomy. *arXiv preprint arXiv:2506.04290*, 2025.

[24] Ruihao Gong, Yang Yong, Shiqiao Gu, Yushi Huang, Chengtao Lv, Yunchen Zhang, Xianglong Liu, and Dacheng Tao. Llmc: Benchmarking large language model quantization with a versatile compression toolkit. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 132–152, Miami, Florida, USA, 2024. Association for Computational Linguistics.

[25] Hongyi Guo, Yuanshun Yao, Wei Shen, Jiaheng Wei, Xiaoying Zhang, Zhaoran Wang, and Yang Liu. Human-instruction-free llm self-alignment with limited samples, 2024.

[26] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. Direct language model alignment from online ai feedback, 2024. *URL https://arxiv. org/abs/2402.04792*, 2024.

[27] Zhenyu Guo and Wenguang Chen. Decoupling knowledge and reasoning in transformers: A modular architecture with generalized cross-attention. *arXiv preprint arXiv:2501.00823*, 2025.

[28] Bernal Jiménez Gutiérrez, Yiheng Shu, Yu Gu, Michihiro Yasunaga, and Yu Su. Hipporag: Neurobiologically inspired long-term memory for large language models, 2025.

[29] Dongge Han, Trevor McInroe, Adam Jelley, Stefano V Albrecht, Peter Bell, and Amos Storkey. Llm-personalize: Aligning llm planners with human preferences via reinforced self-training for housekeeping robots. *arXiv preprint arXiv:2404.14285*, 2024.

[30] Qiaozhi He and Zhihua Wu. Efficient llm inference with kcache, 2024.

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

[32] Kalle Hilsenbek. Breaking the attention bottleneck. *arXiv preprint arXiv:2406.10906*, 2024.

[33] Yizheng Huang and Jimmy Huang. A survey on retrieval-augmented text generation for large language models, 2024.

[34] Ziheng Huang, Sebastian Gutierrez, Hemanth Kamana, and Stephen MacNeil. Memory sandbox: Transparent and interactive memory management for conversational agents, 2023.

[35] Ajay Jaiswal, Zhe Gan, Xianzhi Du, Bowen Zhang, Zhangyang Wang, and Yinfei Yang. Compressing llms: The truth is rarely pure and never simple. In *Proceedings of the 12th International Conference on Learning Representations (ICLR)*, 2024. Poster.

[36] Renren Jin, Jiangcun Du, Wuwei Huang, Wei Liu, Jian Luan, Bin Wang, and Deyi Xiong. A comprehensive evaluation of quantization strategies for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 12186–12215, Bangkok, Thailand, 2024. Association for Computational Linguistics.

[37] Sandra Johnson and David Hyland-Wood. A primer on large language models and their limitations, 2024.

[38] Yuhan Kang, Zhongdi Luo, Mei Wen, Yang Shi, Jun He, Jianchao Yang, Zeyu Xue, Jing Feng, and Xinwang Liu. Hwpq: Hessian-free weight pruning-quantization for llm compression and acceleration. *arXiv e-prints*, pages arXiv–2501, 2025.

[39] Ravindran Kannan, Chiranjib Bhattacharyya, Praneeth Kacham, and David P Woodruff. Levattention: Time, space, and streaming efficient algorithm for heavy attentions. *arXiv preprint arXiv:2410.05462*, 2024.

[40] Zeyneb N. Kaya and Souvick Ghosh. Decoding large-language models: A systematic overview of socio-technical impacts, constraints, and emerging questions, 2024.

[41] Amrit Khera, Rajat Ghosh, and Debojyoti Dutta. Efficient alignment of large language models via data sampling. In *Proceedings of The 4th NeurIPS Efficient Natural Language and Speech Processing Workshop*, volume 262 of *PMLR*, 2024.

[42] Jeonghoon Kim, Jung Hyun Lee, Sungdong Kim, Joonsuk Park, Kang Min Yoo, Se Jung Kwon, and Dongsoo Lee. Memory-efficient fine-tuning of compressed large language models via sub-4-bit integer quantization. In *Proceedings of the 37th Conference on Neural Information Processing Systems (NeurIPS)*, 2023. Poster.

[43] Brandon Kynoch, Hugo Latapie, and Dwane van der Sluis. Recallm: An adaptable memory mechanism with temporal understanding for large language models. *arXiv preprint arXiv:2307.02738*, 2023.

[44] Baolin Li, Yankai Jiang, Vijay Gadepally, and Devesh Tiwari. Llm inference serving: Survey of recent advances and opportunities, 2024.

[45] Guangyan Li, Yongqiang Tang, and Wensheng Zhang. Lorap: Transformer sub-layers deserve differentiated structured compression for large language models. *arXiv preprint arXiv:2404.09695*, 2024.

[46] Shengrui Li, Junzhe Chen, Xueting Han, and Jing Bai. Nuteprune: Efficient progressive pruning with numerous teachers for large language models, 2024.

[47] Zhiteng Li, Mingyuan Xia, Jingyuan Zhang, Zheng Hui, Linghe Kong, Yulun Zhang, and Xiaokang Yang. Adasvd: Adaptive singular value decomposition for large language models. In *Proceedings of the 26th European Conference on Information Retrieval (ECIR)*, 2025.

[48] Jiongnan Liu, Yutao Zhu, Shuting Wang, Xiaochi Wei, Erxue Min, Yu Lu, Shuaiqiang Wang, Dawei Yin, and Zhicheng Dou. Llms+ persona-plug= personalized llms. *arXiv preprint arXiv:2409.11901*, 2024.

[49] Nelson F. Liu, Kevin Lin, John Hewitt, Ashwin Paranjape, Michele Bevilacqua, Fabio Petroni, and Percy Liang. Lost in the middle: How language models use long contexts, 2023.

[50] Yu Mao, Weilan Wang, Hongchao Du, Nan Guan, and Chun Jason Xue. On the compressibility of quantized large language models, 2024.

[51] Rafael Mendoza, Isabella Cruz, Richard Liu, Aarav Deshmukh, David Williams, Jesscia Peng, and Rohan Iyer. Adaptive self-supervised learning strategies for dynamic on-device llm personalization. *arXiv preprint arXiv:2409.16973*, 2024.

[52] Yu Meng, Mengzhou Xia, and Danqi Chen. Simpo: Simple preference optimization with a reference-free reward. *Advances in Neural Information Processing Systems*, 37:124198–124235, 2024.

[53] Tarun Ram Menta, Susmit Agrawal, and Chirag Agarwal. Analyzing memorization in large language models through the lens of model attribution, 2025.

[54] Liana Mikaelyan, Ayyoob Imani, Mathew Salvaris, Parth Pathak, and Mohsen Fayyaz. Deltallm: Compress llms with low-rank deltas between shared weights, 2025.

[55] Ali Modarressi, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. Ret-llm: Towards a general read-write memory for large language models. *arXiv preprint arXiv:2305.14322*, 2023.

[56] Ali Modarressi, Abdullatif Köksal, Ayyoob Imani, Mohsen Fayyaz, and Hinrich Schütze. Memllm: Finetuning llms to use an explicit read-write memory. In *New Frontiers in Associative Memories Workshop, in Proceedings of the International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2025. Oral.

[57] Milad Moradi, Ke Yan, David Colwell, Matthias Samwald, and Rhona Asgari. Exploring the landscape of large language models: Foundations, techniques, and challenges, 2024.

[58] Mohammad Mozaffari, Amir Yazdanbakhsh, and Maryam Mehri Dehnavi. Slim: One-shot quantization and sparsity with low-rank approximation for llm weight compression. *arXiv preprint arXiv:2410.09615*, 2024.

[59] Jiayan Nan, Wenquan Ma, Wenlong Wu, and Yize Chen. Nemori: Self-organizing agent memory inspired by cognitive science, 2025.

[60] Humza Naveed, Asad Ullah Khan, Shi Qiu, Muhammad Saqib, Saeed Anwar, Muhammad Usman, Naveed Akhtar, Nick Barnes, and Ajmal Mian. A comprehensive overview of large language models, 2023.

[61] Long ST Nguyen, Khang HN Vo, Thu HA Nguyen, Tuan C Bui, Duc Q Nguyen, Thanh-Tung Tran, Anh D Nguyen, Minh L Nguyen, Fabien Baldacci, Thang H Bui, et al. Bridging llms and symbolic reasoning in educational qa systems: Insights from the xai challenge at ijcnn 2025. *arXiv preprint arXiv:2508.01263*, 2025.

[62] Joon Sung Park, Joseph C. O'Brien, Carrie J. Cai, Meredith Ringel Morris, Percy Liang, and Michael S. Bernstein. Generative agents: Interactive simulacra of human behavior, 2023.

[63] Patricia Pauli, Dennis Gramlich, and Frank Allgöwer. Lipschitz constant estimation for general neural network architectures using control tools. *arXiv preprint arXiv:2405.01125*, 2024.

[64] Mathis Pink, Vy A. Vo, Qinyuan Wu, Jianing Mu, Javier S. Turek, Uri Hasson, Kenneth A. Norman, Sebastian Michelmann, Alexander Huth, and Mariya Toneva. Assessing episodic memory in llms with sequence order recall tasks, 2024.

[65] Alireza Rezazadeh, Zichao Li, Wei Wei, and Yujia Bao. From isolated conversations to hierarchical schemas: Dynamic tree memory representation for llms. *arXiv preprint arXiv:2410.14052*, 2024.

[66] Anna Rogers and Alexandra Sasha Luccioni. Position: Key claims in llm research have a long tail of footnotes. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, volume 235, pages 42647–42665, Vienna, Austria, 2024. PMLR.

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

[68] Vinay Kumar Sankarapu, Chintan Chitroda, Yashwardhan Rathore, Neeraj Kumar Singh, and Pratinav Seth. Dlbacktrace: A model agnostic explainability for any deep learning models. *arXiv preprint arXiv:2411.12643*, 2024.

[69] Ali Satvaty, Suzan Verberne, and Fatih Turkmen. Undesirable memorization in large language models: A survey, 2024.

[70] Dale Schuurmans. Memory augmented large language models are computationally universal. *arXiv preprint arXiv:2301.04589*, 2023.

[71] Javad Seraj, Mohammad Mahdi Mohajeri, Mohammad Javad Dousti, and Majid Nili Ahmadabadi. Optimizing alignment with less: Leveraging data augmentation for personalized evaluation, 2024.

[72] Akshat Sharma, Hangliang Ding, Jianping Li, Neel Dani, and Minjia Zhang. Minikv: Pushing the limits of llm inference via 2-bit layer-discriminative kv cache, 2024.

[73] Chaitanya Sharma. Retrieval-augmented generation: A comprehensive survey of architectures, enhancements, and robustness frontiers, 2025.

[74] Dachuan Shi, Yonggan Fu, Xiangchi Yuan, Zhongzhi Yu, Haoran You, Sixu Li, Xin Dong, Jan Kautz, Pavlo Molchanov, et al. Lacache: Ladder-shaped kv caching for efficient long-context modeling of large language models. *arXiv preprint arXiv:2507.14204*, 2025.

[75] Luohe Shi, Hongyi Zhang, Yao Yao, Zuchao Li, and Hai Zhao. Keep the cost down: A review on methods to optimize llm' s kv-cache consumption, 2024.

[76] Ruize Shi, Hong Huang, Wei Zhou, Kehan Yin, Kai Zhao, and Yun Zhao. From general to specific: Tailoring large language models for personalized healthcare. *arXiv preprint arXiv:2412.15957*, 2024.

[77] Noah Shinn, Federico Cassano, Edward Berman, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning, 2023.

[78] Oliver Sieberling, Denis Kuznedelev, Eldar Kurtic, and Dan Alistarh. Evopress: Accurate dynamic model compression via evolutionary search. In *Forty-second International Conference on Machine Learning*, 2024.

[79] Chung-En Sun, Tuomas Oikarinen, Berk Ustun, and Tsui-Wei Weng. Concept bottleneck large language models. *arXiv preprint arXiv:2412.07992*, 2024.

[80] Chung-En Sun, Tuomas Oikarinen, and Tsui-Wei Weng. Crafting large language models for enhanced interpretability. *arXiv preprint arXiv:2407.04307*, 2024.

[81] Yutao Sun, Li Dong, Shaohan Huang, Shuming Ma, Yuqing Xia, Jilong Xue, Jianyong Wang, and Furu Wei. Retentive network: A successor to transformer for large language models, 2023.

[82] Yaya Sy, Christophe Cerisara, and Irina Illina. Lillama: Large language models compression via low-rank feature distillation. *arXiv preprint arXiv:2412.16719*, 2024.

[83] Zhen Tan, Tianlong Chen, Zhenyu Zhang, and Huan Liu. Sparsity-guided holistic explanation for llms with interpretable inference-time intervention. In *Proceedings of the AAAI conference on artificial intelligence*, volume 38, pages 21619–21627, 2024.

[84] Yehui Tang, Yunhe Wang, Jianyuan Guo, Zhijun Tu, Kai Han, Hailin Hu, and Dacheng Tao. A survey on transformer compression. *arXiv preprint arXiv:2402.05964*, 2024.

[85] Kaiyuan Tian, Linbo Qiao, Baihui Liu, Gongqingjian Jiang, and Dongsheng Li. A survey on memory-efficient large-scale model training in ai for science, 2025.

[86] Andrei Tomut, Saeed S Jahromi, Abhijoy Sarkar, Uygar Kurt, Sukhbinder Singh, Faysal Ishtiaq, Cesar Muñoz, Prabdeep Singh Bajaj, Ali Elborady, Gianni del Bimbo, et al. Compactifai: extreme compression of large language models using quantum-inspired tensor networks. *arXiv preprint arXiv:2401.14109*, 2024.

[87] Albert Tseng, Jerry Chee, Qingyao Sun, Volodymyr Kuleshov, and Christopher De Sa. Quip#: Even better llm quantization with hadamard incoherence and lattice codebooks. *arXiv preprint arXiv:2402.04396*, 2024.

[88] Mart Van Baalen, Andrey Kuzmin, Ivan Koryakovskiy, Markus Nagel, Peter Couperus, Cedric Bastoul, Eric Mahurin, Tijmen Blankevoort, and Paul Whatmough. Gptvq: The blessing of dimensionality for llm quantization. *arXiv preprint arXiv:2402.15319*, 2024.

[89] Tom Wallace, Naser Ezzati-Jivan, and Beatrice Ombuki-Berman. Optimization strategies for enhancing resource efficiency in transformers and large language models, 2025.

[90] Bing Wang, Xinnian Liang, Jian Yang, Hui Huang, Shuangzhi Wu, Peihao Wu, Lu Lu, Zejun Ma, and Zhoujun Li. Scm: Enhancing large language model with self-controlled memory framework. *arXiv e-prints*, pages arXiv–2304, 2023.

[91] Chenyu Wang, Yihan Wang, and Kai Li. Evaluating zero-shot long-context llm compression, 2024.

[92] Qinsi Wang, Jinghan Ke, Masayoshi Tomizuka, Yiran Chen, Kurt Keutzer, and Chenfeng Xu. Dobi-svd: Differentiable SVD for LLM compression and some new perspectives. In *The Thirteenth International Conference on Learning Representations (ICLR)*, 2025.

[93] Weizhi Wang, Li Dong, Hao Cheng, Xiaodong Liu, Xifeng Yan, Jianfeng Gao, and Furu Wei. Augmenting language models with long-term memory. In *Proceedings of the 37th International Conference on Neural Information Processing Systems (NeurIPS)*, 2023. Poster.

[94] Xin Wang, Yu Zheng, Zhongwei Wan, and Mi Zhang. Svd-llm: Truncation-aware singular value decomposition for large language model compression. In *Proceedings of the 13th International Conference on Learning Representations (ICLR)*, Addis Ababa, Ethiopia, 2025. Poster.

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

[96] Yu Wang, Dmitry Krotov, Yuanzhe Hu, Yifan Gao, Wangchunshu Zhou, Julian McAuley, Dan Gutfreund, Rogerio Feris, and Zexue He. M+: Extending memoryllm with scalable long-term memory. *arXiv preprint arXiv:2502.00592*, 2025.

[97] Junkang Wu, Yuexiang Xie, Zhengyi Yang, Jiancan Wu, Jinyang Gao, Bolin Ding, Xiang Wang, and Xiangnan He. β-dpo: Direct preference optimization with dynamic β, 2024.

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

[99] Yue Wu, Zhiqing Sun, Huizhuo Yuan, Kaixuan Ji, Yiming Yang, and Quanquan Gu. Self-play preference optimization for language model alignment. *arXiv preprint arXiv:2405.00675*, 2024.

[100] Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks, 2024.

[101] Wujiang Xu, Kai Mei, Hang Gao, Juntao Tan, Zujie Liang, and Yongfeng Zhang. A-mem: Agentic memory for llm agents, 2025.

[102] Zhichao Xu, Ashim Gupta, Tao Li, Oliver Bentham, and Vivek Srikumar. Beyond perplexity: Multi-dimensional safety evaluation of llm compression. *arXiv preprint arXiv:2407.04965*, 2024.

[103] Ge Yang, Changyi He, Jinyang Guo, Jianyu Wu, Yifu Ding, Aishan Liu, Haotong Qin, Pengliang Ji, and Xianglong Liu. Llmcbench: Benchmarking large language model compression for efficient deployment. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2025.

[104] Jingfeng Yang, Hongye Jin, Ruixiang Tang, Xiaotian Han, Qizhang Feng, Haoming Jiang, Bing Yin, and Xia Hu. Harnessing the power of llms in practice: A survey on chatgpt and beyond, 2023.

[105] Kevin Yang, Dan Klein, Asli Celikyilmaz, Nanyun Peng, and Yuandong Tian. Rlcd: Reinforcement learning from contrastive distillation for language model alignment. *arXiv preprint arXiv:2307.12950*, 2023.

[106] Dingyu Yao, Bowen Shen, Zheng Lin, Wei Liu, Jian Luan, Bin Wang, and Weiping Wang. Tailorkv: A hybrid framework for long-context inference via tailored kv cache optimization. *arXiv preprint arXiv:2505.19586*, 2025.

[107] Xinhao Yao, Xiaolin Hu, Shenzhi Yang, and Yong Liu. Enhancing in-context learning performance with just svd-based weight pruning: A theoretical perspective. *Advances in Neural Information Processing Systems*, 37:38820–38851, 2024.

[108] Zhongzhi Yu, Zheng Wang, Yuhan Li, Ruijie Gao, Xiaoya Zhou, Sreenidhi Reddy Bommu, Yang Zhao, and Yingyan Lin. Edge-llm: Enabling efficient large language model adaptation on edge devices via unified compression and adaptive layer voting. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pages 1–6, 2024.

[109] Qiuhao Zeng, Jerry Huang, Peng Lu, Gezheng Xu, Boxing Chen, Charles Ling, and Boyu Wang. Zeta: Leveraging z-order curves for efficient top-k attention. *arXiv preprint arXiv:2501.14577*, 2025.

[110] Jianfei Zhang, Jun Bai, Bei Li, Yanmeng Wang, Rumei Li, Chenghua Lin, and Wenge Rong. Disentangling preference representation and text generation for efficient individual preference alignment, 2024.

[111] Jiarui Zhang. Guided profile generation improves personalization with llms. *arXiv preprint arXiv:2409.13093*, 2024.

[112] Rongzhi Zhang, Kuang Wang, Liyuan Liu, Shuohang Wang, Hao Cheng, Chao Zhang, and Yelong Shen. Lorc: Low-rank compression for llms kv cache with a progressive compression strategy, 2024.

[113] Zhenyu Zhang, Ying Sheng, Tianyi Zhou, Tianlong Chen, Lianmin Zheng, Ruisi Cai, Zhao Song, Yuandong Tian, Christopher Ré, Clark Barrett, Zhangyang Wang, and Beidi Chen. H_2o: Heavy-hitter oracle for efficient generative inference of large language models, 2023.

[114] Zining Zhang, Yao Chen, Bingsheng He, and Zhenjie Zhang. Aggressive post-training compression on extremely large language models, 2024.

[115] Runsong Zhao, Pengcheng Huang, Xinyu Liu, Chunyang Xiao, Tong Xiao, and Jingbo Zhu. More effective llm compressed tokens with uniformly spread position identifiers and compression loss, 2024.

[116] Siyun Zhao, Yuqing Yang, Zilong Wang, Zhiyuan He, Luna K. Qiu, and Lili Qiu. Retrieval augmented generation (rag) and beyond: A comprehensive survey on how to make your llms use external data more wisely, 2024.

[117] Weibo Zhao, Yubin Shi, Xinyu Lyu, Wanchen Sui, Shen Li, and Yong Li. Aser: activation smoothing and error reconstruction for large language model quantization. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 22822–22830, 2025.

[118] Zhen Zheng, Xiaonan Song, and Chuanjie Liu. Mixllm: Llm quantization with global mixed-precision between output-features and highly-efficient system design, 2024.

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

[120] Xiabin Zhou, Wenbin Wang, Minyan Zeng, Jiaxian Guo, Xuebo Liu, Li Shen, Min Zhang, and Liang Ding. Dynamickv: Task-aware adaptive kv cache compression for long context llms, 2024.

[121] Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577, 2024.

[122] Thomas P. Zollo, Andrew Wei Tung Siah, Naimeng Ye, Ang Li, and Hongseok Namkoong. Personalllm: Tailoring llms to individual preferences, 2024.

APPENDIX

In this appendix, we provide a comprehensive taxonomy of *Memory-Augmented Large Language Models (LLMs)* to complement the discussions in the main text. Figure 9 systematically organizes the diverse research directions into seven key categories:

- **Memory architectures** – define how models read, write, and maintain memory states.
- **Training strategies** – focus on efficiency and long-context adaptation.
- **Compression techniques** – enable deployment under strict resource constraints.
- **Memory optimization** – including KV cache and low-rank methods.
- **Alignment methods** – incorporating human preferences and personalization.
- **Evaluation and benchmarks** – measuring efficiency, compression, and reasoning capabilities.
- **Applications and case studies** – spanning real-world deployment scenarios and future challenges.

This taxonomy highlights both the breadth and interconnectedness of current approaches, serving as a roadmap for future exploration of memory-augmented reasoning in LLMs.

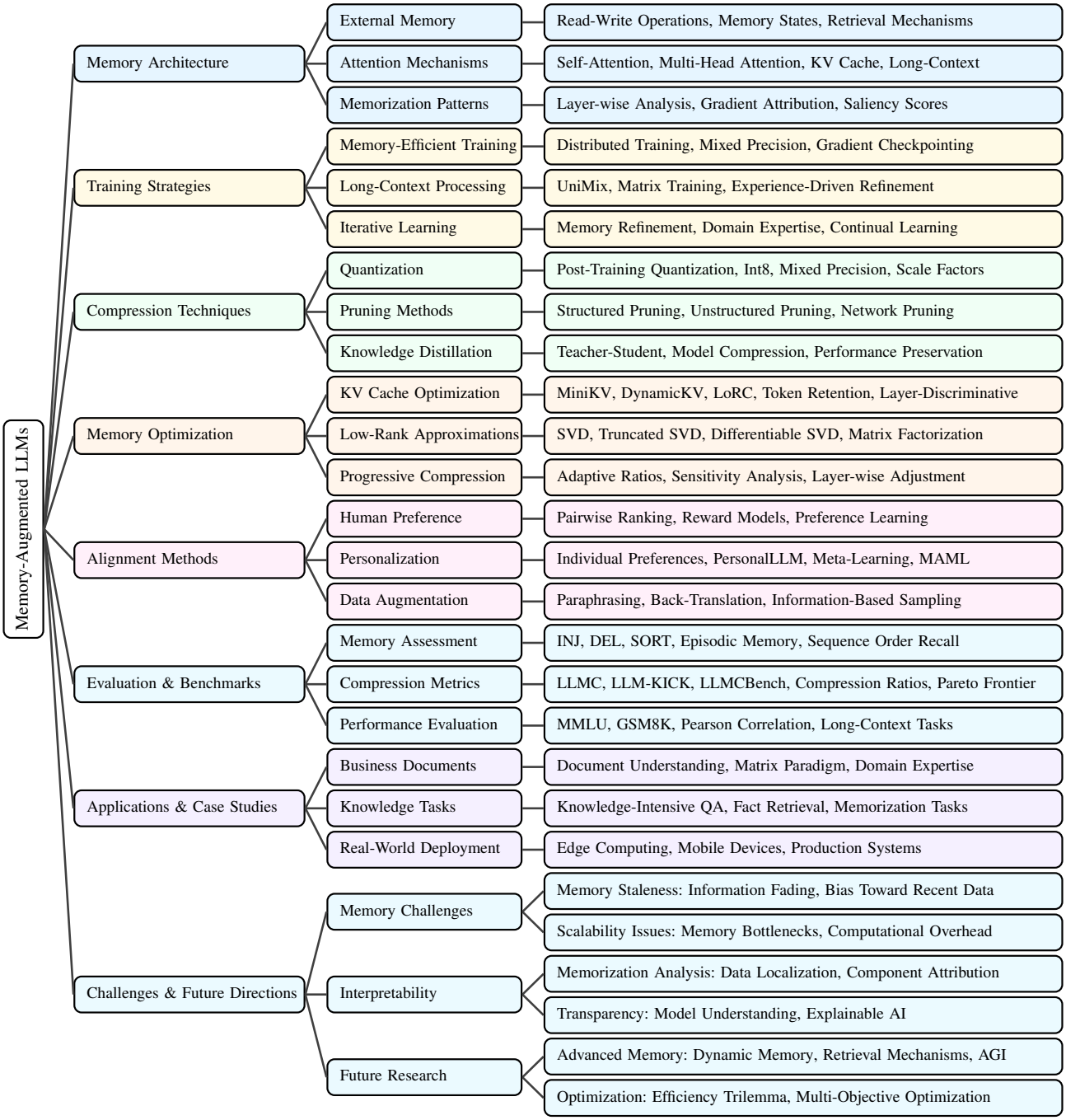

Fig. 9: A comprehensive taxonomy of Memory-Augmented Large Language Models, organizing the field into seven key categories: memory architecture, training strategies, compression techniques, memory optimization, alignment methods, evaluation & benchmarks, and applications & case studies with future directions.