Unifying Knowledge in Agentic LLMs: Concepts, Methods, and Recent Advancements

Lihui Liu
Wayne State University
hw6926@wayne.edu

Kai Shu
Emory University
kai.shu@emory.edu

Abstract

Large language models have demonstrated remarkable capabilities in text generation and problem solving, yet they continue to face fundamental challenges such as hallucination, lack of factual grounding, and limited reasoning reliability. The core of these issues lies the question of how LLMs acquire and use *knowledge*. While internal knowledge embedded in model parameters enables impressive generalization, it is often insufficient for up-to-date or domain-specific tasks. External knowledge integration, such as retrieval-augmented generation (RAG), provides grounding and factuality but introduces challenges of retrieval quality, latency, and reliability. Beyond these paradigms, recent advances in *agentic LLMs* extend models from passive generators to active problem solvers that can reason, plan, and interact with external tools. This survey provides a unified, knowledge-centric perspective on LLMs, organized along three complementary dimensions: (i) reactive: internal knowledge, (ii) lightactive: external knowledge, and (iii) proactive: agentic knowledge utilization for reasoning and tool interaction. We provide a taxonomy of knowledge usage in LLMs, analyze their respective strengths and limitations, and highlight how these paradigms interact in real-world systems. Finally, we identify open challenges to facilitate future research.

1 Introduction

Knowledge lies at the core of how large language models reason (Wei et al., 2022b; Yao et al., 2023a; Imani et al., 2023), generate response (Touvron et al., 2023; Achiam et al., 2023), and support decision-making (Yao et al., 2023b; Bran et al., 2023). While the vast parameters of modern LLMs encode an impressive amount of internalized world knowledge, such knowledge is often incomplete, outdated, or unreliable when applied to specialized domains. To address these shortcomings, recent researches (Lewis et al., 2021; Shi et al., 2023; Edge et al., 2025) have focused on augmenting LLMs with external sources of knowledge—from document corpora to structured knowledge graphs—so that generated outputs can be grounded in verifiable evidence. At the same time, the emergence of agentic LLMs (Yao et al., 2023b; Bran et al., 2023; Yang et al., 2023), which can autonomously plan, reason, and act through interaction with tools and environments, reflects a new paradigm of knowledge use in practice. Together, these three perspectives highlight the fundamental role of knowledge in enabling LLMs to move beyond static text generation toward trustworthy and effective reasoning systems.

Despite their importance, most existing work focus on only one of these directions. For example, work on in-context learning (ICL) (Wei et al., 2022b; Dong et al., 2024; Brown et al., 2020) primarily examine how LLMs exploit internal knowledge through prompting and contextual adaptation. Other work (Lewis et al., 2021; Shi et al., 2023; Edge et al., 2025) emphasize retrieval-augmented generation as a means of grounding outputs with external knowledge. A more recent line of work (Bran et al., 2023; Yang et al., 2023) has begun to review LLM-based agents, but typically from a systems or application-driven viewpoint rather than a unifying theory of knowledge use. What remains missing is a paradigm that treats these perspectives not as isolated techniques but as complementary ways of organizing, accessing, and applying knowledge. This work aims to fill that gap by presenting a knowledge-centric framework that unifies three key problems:

1. **Reactive use of internal knowledge**, where LLMs leverage their parametric memory and contextual reasoning.

2. **Lightly-active use of external knowledge**, where LLMs ground their outputs in verifiable external sources.

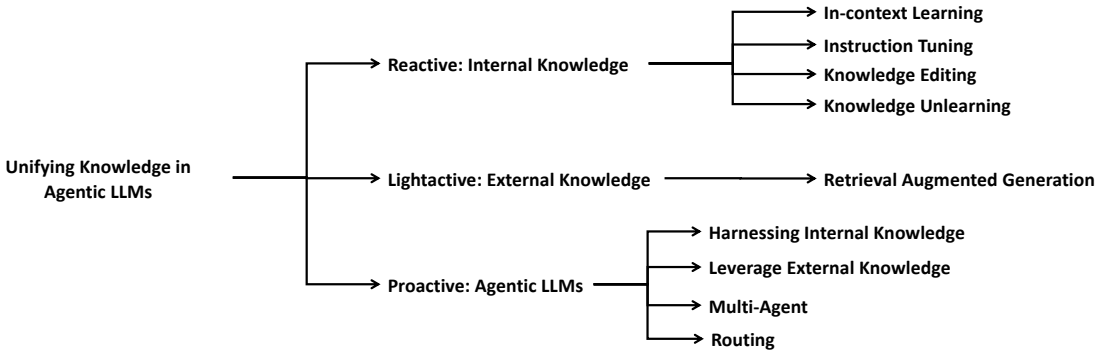

Figure 1: Overview of the survey.

3. **Proactive use of knowledge in an agentic manner**, where models integrate internal and external knowledge to plan, act, and adapt in dynamic environments.

By organizing the work around these three pillars, we aim to provide researchers with a comprehensive view of how knowledge is represented, accessed in LLMs. This unified perspective not only clarifies the current landscape but also illuminates open challenges and future directions for building more reliable, explainable, and knowledge-driven AI systems.

2 Foundations of Knowledge in LLMs

2.1 Background

Large Language Models (LLMs) are powerful deep neural networks built upon the Transformer architecture and trained on massive text corpora to model the probability distribution of natural language. Representative models such as GPT (Achiam et al., 2023), PaLM (Chowdhery et al., 2022), and LLaMA (Touvron et al., 2023) are typically trained using an autoregressive objective, where the model learns to predict the next token in a sequence given its preceding context:

$$\max \sum_{t} \log P(x_t \mid x_{<t}),$$

with x_t representing the current token and $x_{<t}$ denoting all tokens before it. This self-supervised pretraining enables the model to implicitly learn grammar, semantics, world knowledge, and basic reasoning patterns from large-scale unlabeled data.

After pretraining, LLMs are typically adapted to downstream tasks through *supervised fine-tuning* (Ouyang et al., 2022). In this stage, the model is trained on human-annotated input-output pairs (x, y), where x is a task-specific instruction or prompt, and $y = (y_1, \ldots, y_L)$ is the corresponding desired response. The model is optimized to maximize the conditional likelihood of the output sequence given the input:

$$\mathcal{L}_{\text{SFT}}(\theta) = -\sum_{t=1}^{L} \log P_\theta(y_t \mid x, y_{<t}),$$

where $P_\theta(y_t \mid x, y_{<t})$ is the probability of generating token y_t based on the input and previous output tokens, and θ denotes the model parameters. This step helps the model follow instructions and perform specific tasks more reliably.

To further align model behavior with human preferences, values, and safety goals, LLMs are often refined via *Reinforcement Learning from Human Feedback (RLHF)* (Ouyang et al., 2022). The RLHF pipeline begins with the collection of human preference data, where annotators rank several model-generated outputs for the same prompt. These rankings are used to train a *reward model* $r_\phi(x, y)$, which estimates the quality of a response y given a prompt x. Finally, the base model is fine-tuned using a reinforcement learning algorithm, commonly *Proximal Policy Optimization (PPO)* (Schulman et al., 2017), to maximize the expected reward under the learned reward model:

$$\mathcal{L}_{\text{RLHF}}(\theta) = \mathbb{E}_{y \sim \pi_\theta(\cdot|x)} \left[r_\phi(x, y) \right],$$

where π_θ is the current policy (the language model), and ϕ represents the parameters of the reward model. PPO ensures stable optimization by penalizing large deviations from the original policy during updates.

Together, these stages—pretraining, supervised fine-tuning, and RLHF—form a standard training framework that enables LLMs to exhibit strong language understanding, instruction-following behavior, and alignment with human intent. Despite

their generative capabilities, LLMs still face challenges in factual consistency, verifiable reasoning, and robustness, as their outputs are shaped by statistical patterns in data rather than explicit logical or grounded inference mechanisms.

2.2 Internal vs. External Knowledge

Large language models (LLMs) can acquire and utilize knowledge through two complementary channels: *internal knowledge* and *external knowledge*. Internal knowledge (Dong et al., 2024; Zhang et al., 2025; Wei et al., 2022b) refers to the information implicitly encoded within the model parameters during large-scale pretraining. This enables models to recall facts, linguistic structures, and commonsense reasoning without explicit access to external resources. By contrast, external knowledge (Lewis et al., 2021; Edge et al., 2025; Shi et al., 2023) refers to information retrieved or accessed from outside sources at inference time, such as unstructured corpora, structured knowledge bases, or multimodal databases. External knowledge provides grounding, verifiability, and adaptability, especially in dynamic or specialized domains where internal memory may be insufficient or outdated.

2.3 A Taxonomy of Knowledge Usage

We propose a taxonomy of knowledge usage in LLMs, organized into four dimensions:

1. **Knowledge Elicitation** — Drawing upon what the model already encodes, including long-term parametric memory (knowledge in model weights) and short-term contextual memory (context windows, external caches, or memory modules).

2. **Knowledge Updating** — Modifying, updating, or refining the model's internal knowledge. This includes instruction tuning, fine-tuning, or targeted parameter editing to incorporate new facts or correct existing knowledge.

3. **Knowledge Augmentation** — Incorporating external knowledge to enhance reasoning and grounding. This includes retrieval from documents, knowledge graphs, databases, or APIs, enabling the model to access information beyond its parametric memory.

4. **Knowledge Reasoning** — Operationalizing knowledge through agentic behaviors, such as interacting with tools and APIs, planning multi-step tasks, or verifying and refining outputs through feedback loops.

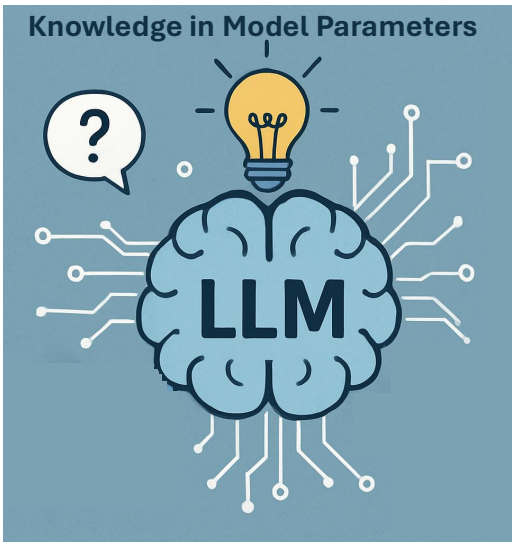

Figure 2: Internal Knowledge.

This taxonomy reflects a layered perspective of knowledge use in LLMs. *Knowledge elicitation* and *knowledge updating* describe access to and updates of internal knowledge, while *knowledge augmentation* captures external knowledge integration, and *knowledge reasoning* represents agentic application. Together, these dimensions provide a unified framework for analyzing how LLMs leverage both internal and external knowledge and how they act as knowledge-driven agents.

3 Internal Knowledge: Reactive

3.1 In-Context Learning

In-Context Learning (ICL) is one of the most distinctive capabilities of large language models (LLMs), providing a way to dynamically use and integrate knowledge. Rather than updating model parameters, ICL allows models to adapt to new tasks at inference time by conditioning on examples, instructions, or reasoning traces embedded directly in the input (Dong et al., 2024). This approach leverages the knowledge already encoded in pretrained parameters and enriches it with knowledge presented in the immediate context, effectively blending static and contextual information in a single reasoning process. ICL highlights how LLMs can exploit knowledge in flexible ways. Prompting strategies, few-shot demonstrations, and structured reasoning traces such as chain-of-thought (Wei

et al., 2022b) exemplars serve as vehicles for injecting relevant knowledge into the model's decision process. These techniques show how users can curate and deliver domain-specific or task-specific knowledge on the fly, without retraining.

This paradigm offers several advantages from a knowledge perspective. By using natural language instructions (zero-shot) or a handful of examples (few-shot), ICL allows LLMs to quickly generalize to new domains and make use of unfamiliar knowledge. Its adaptability makes it possible to integrate knowledge that was absent or only partially represented during pretraining, enabling rapid deployment across diverse application areas.

However, ICL has important limitations in managing and grounding knowledge. The finite context window restricts how much knowledge can be provided at once, which limits performance on knowledge-intensive tasks. Furthermore, because ICL does not explicitly verify or ground its outputs, it often generates hallucinations—fabricated information presented as fact. The absence of traceability means that the knowledge behind a prediction is opaque, reducing transparency and reliability. These issues highlight the need to extend ICL with external knowledge mechanisms, such as retrieval-augmented generation, to improve factuality and verifiability.

3.2 Instruction Tuning

Instruction Tuning is another central paradigm for adapting large language models (LLMs) to follow user-specified tasks more faithfully. Unlike In-Context Learning (ICL), which operates entirely at inference time, instruction tuning fine-tunes pretrained LLMs on a collection of curated datasets consisting of input-output pairs framed as natural language instructions (Zhang et al., 2025). Through this process, the model internalizes the mapping between instructions and desired behaviors, improving its ability to generalize to unseen tasks that share similar formats or intent. In effect, instruction tuning modifies the parameters of the LLM so that following instructions becomes an intrinsic behavior rather than an emergent property of prompting. Well-known examples include T5 (Raffel et al., 2023), FLAN (Wei et al., 2022a), and InstructGPT (Ouyang et al., 2022), which demonstrate that instruction tuning significantly enhances a model's usability by making it more responsive to natural instructions.

The benefits of instruction tuning are substantial. It improves robustness across domains, reduces the reliance on carefully engineered prompts, and provides a systematic approach to aligning LLM behavior with human preferences. Moreover, instruction-tuned models are more capable of handling task variation with minimal additional examples, narrowing the gap between artificial and human-like adaptability.

3.3 Knowledge Editing

Knowledge editing focuses on updating or modifying specific pieces of factual knowledge stored within large language models without retraining them from scratch. Unlike instruction tuning, which globally reshapes model behavior across tasks, knowledge editing seeks localized interventions: changing how the model responds to queries involving particular facts, entities, or relations, while preserving its overall performance and previously acquired knowledge (Meng et al., 2023a; De Cao et al., 2021). This makes knowledge editing especially valuable for time-sensitive or domain-specific updates, such as correcting outdated biomedical facts or incorporating new geopolitical events.

Formally, given a pretrained model f_θ, an editing algorithm aims to produce updated parameters $f_{\theta'}$ such that for a target query x^*, the output $f_{\theta'}(x^*)$ reflects the revised knowledge y^*. At the same time, for non-target queries $x \notin \mathcal{X}^*$, the outputs should remain close to their original predictions, minimizing unintended side effects. Many methods instantiate this principle through optimization objectives of the form:

$$\min_{\Delta\theta} \ \mathcal{L}(f_{\theta+\Delta\theta}(x^*), y^*) + \lambda \, \mathbb{E}_{x \notin \mathcal{X}^*} \left[\mathrm{Dist}(f_{\theta+\Delta\theta}(x), f_\theta(x)) \right] \tag{1}$$

, where the first term enforces correctness of the edit and the second penalizes deviation from the model's prior knowledge.

In practice, knowledge editing ranges from parameter-based updates (e.g., ROME (Meng et al., 2023a), MEMIT (Meng et al., 2023b)) to interventions on hidden representations, balancing precision in updating target facts with generalization to diverse contexts. Nevertheless, edits can propagate undesired changes, fail to generalize beyond surface-level rephrasings, or struggle with multi-hop knowledge, motivating research into more robust, interpretable, and hybrid approaches that combine editing with retrieval or external knowledge

Figure 3: External Knowledge.

for verifiable grounding.

3.4 LLM Knowledge Unlearning

Knowledge unlearning in LLMs aims to remove undesired or outdated knowledge while preserving unrelated information, which is critical for privacy, harm mitigation, and maintaining factual consistency in evolving KGs.

Model-based approaches typically modify parameters to erase specific knowledge, such as using Gradient Ascent (GA) (Neel et al., 2020) to invert gradients of undesired facts or variants that stabilize optimization via relabeled data. While effective at targeted removal, these methods can degrade retained knowledge and struggle to balance forgetting with preservation, especially given the interdependencies among entities and relations.

Evaluating unlearning remains challenging. Benchmarks like WHP (Eldan and Russinovich, 2023), TOFU (Maini et al., 2024), and WMDP (Li et al., 2024) measure fact removal using token-level or entity-level metrics, but most treat knowledge as independent and overlook relational structure. Multi-fact interactions have been explored, though current methods often rely on deterministic or rule-based evaluation, limiting scalability.

In KG–LLMs, unlearning is particularly delicate: removing one fact can inadvertently disrupt reasoning over related entities. Future work may focus on graph-aware algorithms, structural evaluation, and explainable methods to erase knowledge without compromising overall model reasoning and consistency.

4 External Knowledge: Lightly-active

While in-context learning (ICL) enables LLMs to exploit their internal parametric knowledge, it often suffers from limitations such as hallucination, factual errors, and lack of verifiability. Retrieval-Augmented Generation (RAG) (Edge et al., 2025; Lewis et al., 2021; Shi et al., 2023) has emerged as a complementary paradigm designed to address these issues. The central motivation behind RAG is to ground language model outputs in verifiable external sources, thereby improving factuality, reducing hallucinations, and providing transparency into the generation process. By incorporating retrieval mechanisms at inference time, RAG systems ensure that models are not solely dependent on internalized information, but can dynamically access up-to-date and domain-specific knowledge.

RAG systems are typically organized around several architectural patterns. The *classic pipeline* (Shi et al., 2023) follows a two-stage process: a retriever identifies relevant documents from an external corpus, which are then passed to the LLM for generation. More modular variants (Edge et al., 2025) separate retrieval and generation more explicitly, allowing fine-grained control over the knowledge integration process. Recent advances (Makino et al., 2024) also include *end-to-end retrieval-augmented training*, where both retrieval and generation components are optimized jointly, enabling the system to learn to retrieve the most useful context for the task at hand.

The effectiveness of RAG depends on the availability and quality of external knowledge sources. Commonly used sources include large-scale unstructured text corpora, structured repositories such as knowledge graphs, and multimodal databases that integrate text, images, or other data modalities. Text corpora provide breadth and coverage, knowledge graphs offer structured and interpretable representations, and multimodal databases extend LLMs' reasoning beyond text alone. The choice of source often depends on the application, with hybrid approaches combining multiple knowledge types to improve robustness.

Research in RAG has advanced in several directions. Adaptive retrieval methods dynamically select the most relevant content based on context, reducing noise and improving precision. Multi-hop retrieval (Tang and Yang, 2024) enables the chaining of retrieval steps, supporting more complex reasoning across multiple documents. Another trend involves mixture-of-expert retrievers (Jiang et al., 2024), where different retrieval modules specialize in distinct domains or modalities, and the system learns to route queries adaptively. Together, these advances push RAG beyond simple document

lookup toward more sophisticated, context-aware grounding strategies.

Despite its promise, RAG faces several challenges. Retrieval quality remains a critical bottleneck, as noisy or irrelevant documents can mislead the generator. Latency is another issue, as retrieval introduces additional computation that may hinder real-time applications. Finally, ensuring reliable grounding is non-trivial: models may ignore retrieved evidence or selectively use it in ways that do not guarantee factual correctness. Addressing these challenges is essential to make RAG both scalable and trustworthy.

5 Agentic LLMs: Proactive

Reactive vs Proactive: Traditional paradigms for leveraging knowledge in large language models, such as in-context learning and retrieval-augmented generation, primarily focus on how models passively access and integrate information. While effective in many scenarios, these approaches treat the model largely as a reactive system: it generates outputs based on prompts or retrieved context without actively initiating exploration, planning, or intervention. Agentic LLMs, in contrast, adopt a *proactive* stance. Rather than waiting for explicit queries, these models can autonomously identify relevant knowledge, anticipate information gaps, and plan multi-step actions to achieve objectives. This proactive behavior enables LLMs to actively operationalize knowledge, bridging the gap between static information retrieval and dynamic problem-solving. By reasoning about potential outcomes and taking initiative, agentic LLMs can efficiently navigate complex tasks, integrate diverse sources of knowledge, and adapt to changing environments.

Furthermore, proactive agentic behavior allows LLMs to better utilize knowledge in both parametric and non-parametric forms. Internally stored knowledge in model parameters can be applied strategically for reasoning and planning, while external knowledge sources—such as databases, APIs, or knowledge graphs—can be selectively queried to fill information gaps or verify hypotheses. This combination enhances both the effectiveness and reliability of the model, allowing it to act as an autonomous knowledge processor rather than a passive text generator. By enabling models to reason, plan, act, and interact with external tools or environments, agentic LLMs extend the scope of what AI systems can achieve. They are capable of sequential decision-making, iterative refinement, and adaptive behavior, all of which are crucial for real-world applications that demand more than single-step responses. In this sense, agentic LLMs represent a natural evolution from ICL and RAG toward models that can operationalize knowledge in a goal-directed and context-aware manner.

5.1 Harnessing Internal Knowledge

Agentic LLMs rely heavily on internal knowledge stored in their parameters, memory mechanisms, and reasoning capabilities:

Memory and Profile: In LLM agents, memory refers to the ability to retain and utilize past interactions, contextual information, and long-term knowledge about users or tasks, while profile represents structured information that defines the agent's role, attributes, preferences, and operational competencies. Together, they enable the agent to maintain continuity across sessions, adapt to user-specific needs, and perform domain-specific functions more effectively. Since both memory and profile are stored, organized, and accessed internally by the LLM (rather than retrieved externally), they are considered part of the LLM's internal knowledge, shaping how it reasons, plans, and interacts in dynamic environments.

Reasoning: In LLM agents, reasoning refers to the process of drawing logical inferences and making consistent conclusions based on available information, whether from the prompt, prior memory, or the model's internal knowledge. It allows the agent to connect facts, resolve ambiguities, and justify decisions beyond surface-level pattern matching. Since this ability emerges from the model's internal representations and learned structures—rather than depending on external retrieval—it is considered part of the LLM's internal knowledge.

Planning: In LLM agents, planning refers to the ability to generate and organize a sequence of coherent actions or reasoning steps that lead from the current state to a desired goal. Unlike simple response generation, planning enables the agent to anticipate future requirements, break down complex tasks into manageable sub-tasks, and adapt strategies based on constraints or feedback. As this process relies on the model's internal reasoning capabilities—using its knowledge and learned patterns to decide how to act rather than retrieving instructions from outside—it is regarded as part of

the LLM's internal knowledge and a key component of goal-directed autonomous behavior.

Frameworks like *ReAct* (Yao et al., 2023b) interleave reasoning and planning, generating intermediate reasoning traces while leveraging internal knowledge for decision-making. *Reflexion* (Shinn et al., 2023) adds feedback loops, enabling agents to evaluate and refine strategies based on past experiences.

5.2 Leverage External Knowledge

Agentic behavior is further enhanced through access to external knowledge sources:

- **Tool Use:** Agents can call APIs, search engines, or other software tools to retrieve up-to-date information.
- **Knowledge Graphs and Databases:** Structured data sources provide factual grounding and help reduce hallucinations.
- **Environment Interaction:** Agents can act within simulated or real-world environments to execute tasks or gather information.

Systems like *AutoGPT* (Yang et al., 2023) demonstrate automated planning and execution across longer horizons, integrating external tools, web interactions, and multi-step workflows to achieve complex goals. External grounding through retrieval-augmented generation (RAG) ensures that agentic actions are informed by accurate and current knowledge.

5.3 Multi-agent Collaboration

Agentic LLMs can also operate in multi-agent settings, where multiple models coordinate, negotiate, and share knowledge to achieve complex objectives. Let $\mathcal{A} = \{A_1, A_2, \ldots, A_m\}$ denote a set of agents, each with internal state $\mathbf{s}_t^i$ and access to a subset of knowledge $\mathcal{K}_i$. Multi-agent frameworks enable:

- **Task Decomposition:** Complex problems can be broken down into sub-tasks and distributed among specialized agents, allowing each agent to focus on what it does best. This division of labor improves efficiency and makes large-scale objectives more tractable.
- **Knowledge Sharing:** Agents exchange intermediate results, reasoning traces, or learned insights to build a richer shared context. This reduces redundancy, enables cross-validation of outputs, and improves the accuracy and robustness of the overall system.
- **Coordination and Negotiation:** Agents dynamically adjust their strategies, resolve conflicts, and balance trade-offs to align with shared objectives. Through negotiation and coordination, the system can adapt to evolving environments and optimize collective decision-making.

Multi-agent collaboration is particularly useful for scenarios that require parallel exploration, multi-perspective reasoning, or combining heterogeneous expertise.

5.4 Routing

Efficiently routing tasks and queries to the appropriate agent or knowledge source is crucial for scalability and reliability:

- **Dynamic Routing** (Ong et al., 2025): Tasks can be dynamically assigned based on agent capabilities, knowledge coverage, or past performance.
- **Hierarchical Routing** (Ding et al., 2024): Multi-level controllers can delegate subtasks to specialized agents or modules.
- **Load Balancing and Redundancy** (Shnitzer et al., 2023): Proper routing prevents bottlenecks and ensures robustness by allowing multiple agents to handle critical tasks.

Routing strategies can be formalized as a function $R : \mathcal{T} \times \mathcal{A} \to \mathcal{A}'$, mapping tasks to suitable agents $\mathcal{A}' \subseteq \mathcal{A}$ to maximize overall system utility while minimizing latency and redundancy.

By integrating internal knowledge, external grounding, multi-agent collaboration, and task routing, agentic LLMs advance toward truly autonomous and proactive AI systems capable of handling complex, real-world challenges.

6 Applications and Implications

- **Scientific Discovery:** Agents can autonomously search literature, design hypotheses, and propose experiments, accelerating research cycles and uncovering patterns that might be missed by human researchers. They can also integrate multi-modal data sources

(e.g., text, images, and structured datasets) to support more comprehensive scientific reasoning.

- **Decision Support:** Multi-step planning and tool integration allow agents to assist with travel booking, healthcare recommendations, or business strategy analysis. In complex scenarios, they can combine internal reasoning with external knowledge retrieval to support transparent and explainable recommendations.

- **Interactive Assistants:** Agentic LLMs manage dialogues, query databases, and integrate APIs to provide context-aware and useful responses. These systems can adapt to user preferences over time, offering personalized experiences across domains such as education, legal consultation, and customer support.

- **Autonomous Systems:** When combined with sensors and real-time feedback, agentic LLMs can drive autonomous decision-making in robotics, logistics, and infrastructure management, offering scalable solutions for dynamic environments.

However, agentic LLMs also introduce challenges: alignment with human values, safety in high-stakes domains, computational efficiency, and robustness to adversarial inputs. These concerns highlight the need for assurance frameworks to ensure trustworthy agentic AI. Moreover, ethical considerations, regulatory compliance, and sustainability must be addressed as these systems are deployed at scale. Balancing innovation with accountability will be critical to maximize the societal benefits of agentic LLMs while minimizing potential risks.

7 Comparative Analysis

The three paradigms of knowledge use in LLMs—internal, external, and agentic—offer complementary strengths and weaknesses. Internal knowledge is advantageous when rapid adaptability is needed, especially in domains where the model's parametric memory suffices. However, it suffers from lack of grounding and verifiability. External knowledge, by contrast, provides transparency and factuality, but its effectiveness is constrained by retrieval quality and latency. Agentic approaches expand the horizon of what LLMs can achieve, enabling complex planning and tool use, yet introduce new challenges of alignment, efficiency, and safety.

Synergies between these paradigms are increasingly important in real-world systems. For instance, agentic LLMs often rely on ICL for reasoning and RAG for grounding, combining the flexibility of internal memory with the factuality of external retrieval. Hybrid systems that dynamically balance between parametric and non-parametric knowledge sources are particularly promising for applications that demand both creativity and reliability.

The trade-offs between these approaches can be characterized along several dimensions. Cost is a major consideration, as retrieval and agentic actions introduce overhead relative to pure ICL. Reliability varies depending on the availability of external resources and the ability to ground outputs in evidence. Scalability is affected by both context length in ICL and retrieval efficiency in RAG. Finally, explainability is often higher in RAG and agentic systems, where outputs can be traced to sources or intermediate steps, than in pure ICL.

8 Open Challenges and Future Directions

Despite significant progress, several open challenges remain in applying LLMs to real-world applications. Addressing these challenges is critical for building systems that are scalable, reliable, and trustworthy. Key directions include:

- **Scaling Context vs. Retrieval:** A fundamental design question is whether to prioritize expanding context windows or enhancing retrieval mechanisms. Larger context windows improve ICL capabilities by allowing models to directly condition on more examples, but they are memory-intensive and less structured. In contrast, retrieval offers a more scalable way to incorporate external knowledge but introduces challenges in retrieval quality, latency, and integration.

- **Hybrid Symbolic-Neural Reasoning:** Integrating symbolic reasoning with neural models could combine the flexibility of deep learning with the precision of formal logic. Developing architectures that support structured reasoning, constraint satisfaction, and explainability while retaining the adaptability of neural methods remains an open problem.

- **Dynamic and Persistent Memory:** Current systems largely rely on ephemeral context windows or static databases. Building agents with long-term, verifiable memory—capable of accumulating experiences, updating knowledge, and adapting over time—remains a significant frontier. This raises questions around consistency, version control, and trust.
- **Evaluation Beyond Accuracy:** Existing benchmarks often focus on surface-level accuracy, which fails to capture important qualities such as factual grounding, reasoning depth, adaptability, efficiency, safety, and trustworthiness. There is a need for richer evaluation frameworks and stress tests to assess agent performance in open-world and high-stakes settings.
- **Alignment and Trustworthiness:** As agents gain autonomy, ensuring alignment with human values, ethical principles, and regulatory requirements becomes critical. Robustness against adversarial inputs, transparency in decision-making, and mechanisms for accountability will be essential components of future systems.

Ultimately, progress toward knowledge-centric and trustworthy AI will require unifying these threads. Future research must not only scale model capacity but also design systems that integrate memory, retrieval, reasoning, and action in ways that are reliable, interpretable, and aligned with human goals.

9 Conclusion

This work has reviewed the foundations and frontiers of knowledge use in large language models, focusing on three complementary paradigms: *internal knowledge* through in-context learning, *external knowledge* through retrieval-augmented generation, and *agentic knowledge use* through LLM-based agents. Each paradigm brings distinct strengths—adaptability, factual grounding, and operational autonomy—while also facing unique limitations.

By framing these approaches under a unified knowledge-centric perspective, we highlight their synergies as well as their trade-offs. Looking forward, we envision LLMs not merely as passive text generators, but as *knowledge processors* that internalize, ground, and operationalize information. Bridging these paradigms will be key to developing the next generation of trustworthy, explainable, and effective AI systems.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Andres M Bran, Sam Cox, Oliver Schilter, Carlo Baldassari, Andrew D White, and Philippe Schwaller. 2023. Chemcrow: Augmenting large-language models with chemistry tools.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Ben Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier Garcia, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayana Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Diaz, Orhan Firat, Michele Catasta, Jason Wei, Kathy Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways.

Nicola De Cao, Wilker Aziz, and Ivan Titov. 2021. Editing factual knowledge in language models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 6491–6506, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks V. S. Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hy-

brid llm: Cost-efficient and quality-aware query routing.

Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, Baobao Chang, Xu Sun, Lei Li, and Zhifang Sui. 2024. A survey on in-context learning.

Darren Edge, Ha Trinh, Newman Cheng, Joshua Bradley, Alex Chao, Apurva Mody, Steven Truitt, Dasha Metropolitansky, Robert Osazuwa Ness, and Jonathan Larson. 2025. From local to global: A graph rag approach to query-focused summarization.

Ronen Eldan and Mark Russinovich. 2023. Who's harry potter? approximate unlearning in llms.

Shima Imani, Liang Du, and Harsh Shrivastava. 2023. Mathprompter: Mathematical reasoning using large language models.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, Szymon Antoniak, Teven Le Scao, Théophile Gervet, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2024. Mixtral of experts.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2021. Retrieval-augmented generation for knowledge-intensive nlp tasks.

Nathaniel Li, Alexander Pan, Anjali Gopal, Summer Yue, Daniel Berrios, Alice Gatti, Justin D. Li, Ann-Kathrin Dombrowski, Shashwat Goel, Long Phan, Gabriel Mukobi, Nathan Helm-Burger, Rassin Lababidi, Lennart Justen, Andrew B. Liu, Michael Chen, Isabelle Barrass, Oliver Zhang, Xiaoyuan Zhu, Rishub Tamirisa, Bhrugu Bharathi, Adam Khoja, Zhenqi Zhao, Ariel Herbert-Voss, Cort B. Breuer, Samuel Marks, Oam Patel, Andy Zou, Mantas Mazeika, Zifan Wang, Palash Oswal, Weiran Lin, Adam A. Hunt, Justin Tienken-Harder, Kevin Y. Shih, Kemper Talley, John Guan, Russell Kaplan, Ian Steneker, David Campbell, Brad Jokubaitis, Alex Levinson, Jean Wang, William Qian, Kallol Krishna Karmakar, Steven Basart, Stephen Fitz, Mindy Levine, Ponnurangam Kumaraguru, Uday Tupakula, Vijay Varadharajan, Ruoyu Wang, Yan Shoshitaishvili, Jimmy Ba, Kevin M. Esvelt, Alexandr Wang, and Dan Hendrycks. 2024. The wmdp benchmark: Measuring and reducing malicious use with unlearning.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C. Lipton, and J. Zico Kolter. 2024. Tofu: A task of fictitious unlearning for llms.

Kohei Makino, Makoto Miwa, and Yutaka Sasaki. 2024. End-to-end trainable retrieval-augmented generation for relation extraction.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in gpt.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer.

Seth Neel, Aaron Roth, and Saeed Sharifi-Malvajerdi. 2020. Descent-to-delete: Gradient-based methods for machine unlearning.

Isaac Ong, Amjad Almahairi, Vincent Wu, Wei-Lin Chiang, Tianhao Wu, Joseph E. Gonzalez, M Waleed Kadous, and Ion Stoica. 2025. Routellm: Learning to route llms with preference data.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback.

Colin Raffel, Noam Shazeer, Adam Roberts, Katherine Lee, Sharan Narang, Michael Matena, Yanqi Zhou, Wei Li, and Peter J. Liu. 2023. Exploring the limits of transfer learning with a unified text-to-text transformer.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. 2017. Proximal policy optimization algorithms.

Weijia Shi, Sewon Min, Michihiro Yasunaga, Minjoon Seo, Rich James, Mike Lewis, Luke Zettlemoyer, and Wen-tau Yih. 2023. Replug: Retrieval-augmented black-box language models. *arXiv preprint arXiv:2301.12652*.

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

Tal Shnitzer, Anthony Ou, Mírian Silva, Kate Soule, Yuekai Sun, Justin Solomon, Neil Thompson, and Mikhail Yurochkin. 2023. Large language model routing with benchmark datasets.

Yixuan Tang and Yi Yang. 2024. Multihop-rag: Benchmarking retrieval-augmented generation for multi-hop queries.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. 2023. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*.

Jason Wei, Maarten Bosma, Vincent Y. Zhao, Kelvin Guu, Adams Wei Yu, Brian Lester, Nan Du, Andrew M. Dai, and Quoc V. Le. 2022a. Finetuned language models are zero-shot learners.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022b. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Hui Yang, Sifu Yue, and Yunzhong He. 2023. Auto-gpt for online decision making: Benchmarks and additional opinions.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik Narasimhan. 2023a. Tree of thoughts: Deliberate problem solving with large language models.

Shunyu Yao, Jeffrey Zhao, Dian Yu, Nan Du, Izhak Shafran, Karthik Narasimhan, and Yuan Cao. 2023b. React: Synergizing reasoning and acting in language models.

Shengyu Zhang, Linfeng Dong, Xiaoya Li, Sen Zhang, Xiaofei Sun, Shuhe Wang, Jiwei Li, Runyi Hu, Tianwei Zhang, Fei Wu, and Guoyin Wang. 2025. Instruction tuning for large language models: A survey.