JOURNAL OF INTERNATIONAL CRISIS AND RISK COMMUNICATION RESEARCH
ISSN: 2576-0017
2026, VOL 9, NO 2

Augmenting Large Language Models

Shameer Erakkath Saidumuhammed

Independent Researcher, USA

Abstract

While Large Language Models demonstrate capabilities in reasoning, creativity, and task automation, they remain unable to reliably execute high-precision enterprise tasks due to inherent constraints. This article explores strategies for overcoming these constraints through tool integration, retrieval systems, and structured workflows specifically addressing issues related to static training data, computational costs, limited context windows, and hallucinations inherent to the modeling approach. Experiments show that Retrieval-Augmented Generation yields 10-percentage-point improvements in accuracy on knowledge-intensive datasets, that multi-stage prompting yields 83.5 percentage point improvements in compositional reasoning datasets, and that scaling the number of parameters from 62 billion to 540 billion yields 7.6- to 12.2 percentage point improvements on different metrics of complex reasoning. Human-AI collaboration frameworks show 20-35% productivity gains in software engineering tasks and 40-60% data efficiency gains in interactive machine learning methods. By combining retrieval methods, experimental agent architectures, fine-tuning methods, and human-in-the-loop strategies, systems have been built that use language models closely as components of a larger pipeline. This augmented intelligence model can flexibly meet enterprise-grade requirements for precision, latency, and reliability and can also accommodate continued advances across a wide variety of operational contexts.

Keywords: Retrieval-Augmented Generation, Multi-Agent Systems, Human-AI Collaboration, Foundation Model Scaling, Augmented Intelligence.

1. Introduction

Large Language Models have led to substantial progress in artificial intelligence as a result of their reasoning, creativity, and automation capabilities, though these capabilities are restricted to situations where high precision and reliability are not required. The number of tokens the LLM observes is called the context window. Early LLMs had context windows limited to 8K tokens, causing issues with both datasets containing long documents and with long-term reasoning. Additionally, the stochastic nature of LLMs makes it difficult to produce expected deterministic outputs. Reproducible and verifiable outputs are necessary in compliance-heavy domains such as law, medicine, and financial services. The accelerating use cases for generative AI have shifted the focus away from using LLMs in isolation to using LLMs in conjunction with external tools and retrieval systems and in the orchestration of advanced workflows. This is critical in overcoming the limitations of using generative AI for mission-critical use cases such as knowledge staleness, hallucination, and lack of verifiable source attribution [2]. Also, the intrinsic high compute cost of the state-of-the-art generative AI models makes it difficult to deploy in a low-latency setting. Effectively, this limits the scalability of attentional approaches, which are typically quadratic in compute, but with retrieval-augmented generation, multi-agent architectures, and human-in-the-loop processes, augmented intelligence pipelines can yield more accurate, reliable, and goal-oriented AI. These hybrid orchestration architectures recognize that foundation models are composable building blocks,

mixing and matching generative capabilities of LLMs with production-grade precision. They are a natural evolution beyond monolithic models because human-centered AI systems in the real world inevitably require a combination of statistical ML capabilities with symbolic reasoning. external knowledge and explicit control that meet enterprise-level requirements of auditability and operational safety.

2. Fundamental Limitations of Large Language Models

2.1 Knowledge and Information Constraints

Given that LLMs train on a fixed corpus, they will have persistent gaps in knowledge specific to a domain. This is because datasets are curated to span broad knowledge domains rather than specializing deeply in domains with fewer and less-distributed data points. Specialist domains include, for example, regulatory compliance, clinical medicine, advanced materials science, or other knowledge-intensive fields. [3] Another challenge is the inability to access real-time information after training and adaptation. It causes the models to become increasingly stale as the world changes and the data distribution diverges, as well as conflicts with enterprise applications. Given the internet-scale data used to train the models, they often know only inaccurate and inconsistent terms and contextual knowledge for working and living in a working environment with specific conventions and institutional knowledge.

2.2 Computational and Behavioral Constraints

However, the transformer architecture has an important computational cost when implemented at scale, leading to trade-offs in production deployment for inference latency, compute costs, and model capacity and throughput. In practice, scaling LLMs in production environments requires large amounts of graphics processing unit (GPU) memory and compute, which leads to economic and operational challenges for real-time applications or applications that require high concurrency distributed across multiple servers. The stochastic nature of autoregressive language generation limits its deployability in production settings where determinism is required. Likewise, techniques like temperature sampling and top-k decoding introduce non-reproducibility, which obstructs auditing and process validation. Additionally, autoregressive models suffer from a well-studied tradeoff between the model's ability to scale on one hand and its controllability on the other. As autoregressive models are scaled up, few-shot and reasoning capabilities increase, but interpretability based on prompt sensitivity or freedom of output based on instruction following or safety guardrails becomes more difficult.

2.3 Architectural and Reliability Constraints

The pre-trained models limit the maximum sequence length available during inference, introducing an architectural trade-off between sequence length and efficiency, as self-attention has computational and memory complexity quadratic in the number of tokens. Longer documents are either lossy-compressed, sacrificing potential information, or separately chunked, breaking semantic coherence and cross-document relationships. Second, models fail to maintain long-term reasoning consistency across long inference chains, with performance not being durable for multi-hop inference tasks and long-term logical consistency beyond the model's attention span [3]. The most severe reliability problem is hallucination, where models generate outputs that are plausible but ultimately inaccurate and presented with high confidence, leading to low trust in mission-critical applications due to broken source attribution and the lack of means to evaluate the veracity of outputs [4]. These trust deficits require human validation workflows, which can undo most of the automation efficiency gains.

Table 1: Core Limitation Categories in Large Language Models [3, 4]

Limitation Category	Primary Constraints	Key Challenges
Knowledge & Information	Static training corpora, temporal staleness	Domain expertise gaps, inability to access current information, and general vs. specialized knowledge misalignment

Computational & Behavioral	Resource intensity, probabilistic generation	GPU memory demands, non-deterministic outputs, controllability vs. capability trade-offs
Architectural & Reliability	Context window bounds, reasoning coherence	Quadratic attention complexity, multi-hop inference degradation, hallucination, and source attribution failures

3. External Augmentation: Tools and Structured Workflows

3.1 Retrieval-Augmented Generation (RAG)

Retrieval-augmented generation architectures address these shortcomings of autoregressive language models through a multi-stage pipeline of embedding generation, vector storage, similarity-based retrieval, and evidence-grounded generation [1]. In the RAG framework, dense representations for documents give each document a semantic representation in terms of a variable-length vector, such that similar documents will have embedded representations likewise. These embeddings are further stored in retrieval-optimized dense vector databases to enable approximate nearest neighbor search (ANN). At inference time, the most relevant passages are retrieved using query similarity, and the language model is prompted with this retrieved evidence to produce grounded outputs. Experimental results show RAG architectures outperform parametric-only baselines on knowledge-intensive tasks. In the Natural Questions dataset, RAG models achieve 44.5% accuracy, while parametric-only baselines achieve 34.5% accuracy. In the TriviaQA dataset, RAG models achieve 45.2% accuracy, while parametric-only baselines achieve 38.6% accuracy. RAG directly addresses hallucination, verifiability, and knowledge obtention issues compared to parametric-only methods by grounding model outputs in verifiably consumed source documents via passage attribution and further avoids retraining or fine-tuning. RAG has become the de facto standard for enterprise LLM deployment due to the ability to directly and verifiably link prompts and token usage to an organization's or domain's knowledge bases, regulation documents, and domain-specific corpora, while remaining auditable and under acceptable levels of internal control.

3.2 Prompt Engineering and Multi-Stage Reasoning

Prompt chaining techniques decompose cognitive tasks into several intermediate steps and use the intermediate outputs to generate inputs for the next cognitive step, providing explicit reasoning chains and reducing error propagation [6]. Instead of monolithic prompt formats that encode the specifications of a task into a single instruction, prompt chaining enables the use of modular pipelines that break down the reasoning process, validate intermediate outputs, and refine the outputs generated at each step. Multi-stage reasoning approaches outperform end-to-end generation in terms of hallucination reduction, reducing the search space at each step while limiting error propagation from initial reasoning mistakes. For example, least-to-most prompting beats chain-of-thought prompting 99.7% to 16.2% on the SCAN benchmark under length split, and achieves 82.45% compared to 74.77% for chain-of-thought prompting on the non-football subset of DROP containing numerical problems [6]. The trade-off comes from the cost in latency and compute from making multiple calls to the model, versus the benefits of accuracy, tighter domain constraints, and human supervision of critical reasoning chain decisions.

3.3 Agentic AI and Multi-Agent Architectures

Agentic systems leverage LLMs' capabilities in more autonomous processes, programmatic tool use, and role assignment. LLMs instantiate the principles of human organizational design, where specialized cognitive agents are assigned roles. Multimodal agentic systems instantiate pattern language templates defining roles such as planner agents (breaking down high-level goals into smaller, executable subtasks), researcher agents (collating and summarizing information from external knowledge sources), critics (factually and logically evaluating intermediate results; relevant reasoning paths and steps), and executor agents (taking action, such as executing API calls or code). Within the context of language model capabilities, multi-agent debate has been shown to lead to improvements in reasoning accuracy and performance on mathematical and commonsense reasoning tasks via iterative refinement and cross-

validation between agent outputs [5]. A multi-agent system can also yield performance improvements compared to a single model by allowing agents to specialize in particular reasoning modalities or areas of knowledge instead of a single model handling all aspects of a complex process. While introducing coordination complexity and communication overhead between agents, the multi-agent model enables scaling solutions to more diverse tasks than the effective capacity of monolithic model architectures [5].

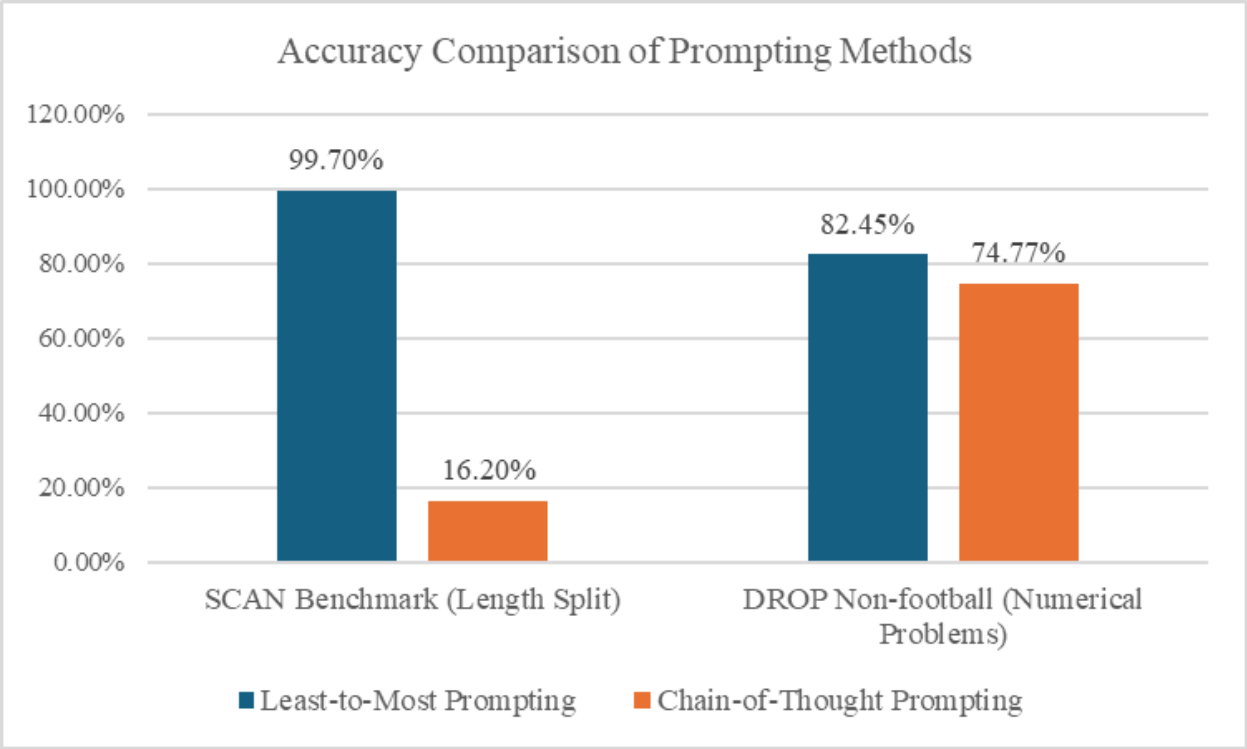

Fig 1: Accuracy (%) of Prompting Methods on SCAN and DROP Benchmarks [5, 6]

4. Internal Augmentation: Foundation Model Advancement

4.1 Evolution Across Model Generations

The reasoning capabilities of generative foundation models increase as the number of model parameters is increased. The relative improvement varies across different benchmarks and evaluation settings. For example, the PaLM 540B parameter model achieves 69.3% accuracy when evaluated on the 5-shot version of the MMLU benchmark. This represents a 15.6 percentage point gain over the 62B parameter model, which achieves 53.7% accuracy. [7] Scaling from 62B to 540B parameters provides larger improvements on one-fourth of BIG-bench tasks than scaling from 8B to 62B parameters, suggesting that further capabilities are only present beyond a certain scale [7]. Architectural modifications that seek to reduce computation, such as mixture-of-experts routing, attention mechanisms, and parameter sharing, are not consistently successful across implementations or tasks [8]. Comparisons of a wide variety of modified Transformer networks indicate that most modifications have little effect on overall performance on tasks such as transfer learning, supervised training and language modeling, with performance improvements being largely limited to small architectural changes (e.g. changing activation functions or number of attention heads) or changes that increase model parameter count with no additional processing time (e.g. sparse activation patterns) [8]. These observations suggest that architectural improvements require multiple implementations and applications to verify their generality, as many improvements in one experimental setting do not transfer to novel codebases or task distributions.

4.2 Reliability and Safety Improvements

Foundation model performance is measured on a range of axes including factuality, calibration and constraint of model outputs, and the increasing size of the foundation model leads to improved performance on difficult reasoning and other benchmarks. The PaLM 540B model achieves state-of-the-art few-shot performance on hundreds of language understanding and generation benchmarks, including greatly improved performance on multi-step reasoning tasks via chain-of-thought prompting [7]. In reasoning benchmarks requiring arithmetic or commonsense inference, 540B matches or outperforms fine-tuned state-of-the-art results on 8-shot evaluation. For example, on the GSM8K dataset for arithmetic problems, 540B achieves 58% accuracy whereas the 62B model achieves 33% accuracy [7]. Scaling the model size from 62B to 540B parameters corrects many semantic understanding and reasoning chain errors the smaller models produce [7]. However, fine-tuning certain architectural features to improve specific reliability estimates did not generalize to other tasks or implementations. Few methods consistently outperformed baseline architectures across 29 English NLP tasks or multilingual translation tasks [8]. For many natural language understanding and generation tasks, there is a strong correlation between pre-training perplexity and downstream performance (Spearman's $\rho = 0.87$ and 0.80), whereas knowledge-intensive tasks have lower correlation ($\rho = 0.69$). This would indicate that the reliability improvements with scale are due to the characteristics of the task [8].

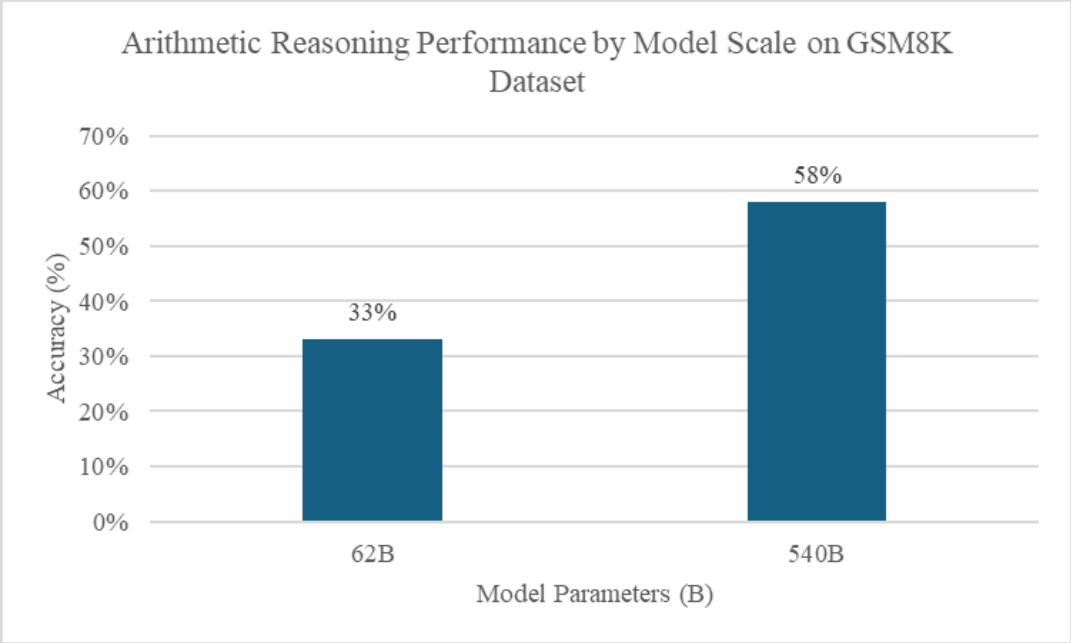

Fig 2: Pre-training Perplexity Correlation with Downstream Task Performance [7, 8]

5. Human-AI Collaboration Models

5.1 The Centaur Model

Centaur systems are a modularized approach to task allocation, in which automation and automated perception are performed by AI, while ethical reasoning and deliberative decision-making are done by humans. Due to this specialization, the modules which require more computations and intensive processing (like AI) are separated from others (like humans). Complementary team performance occurs when humans and AIs perform better together than apart when their predictions each make different types of mistakes. In sentiment classification, human-AI teams performed 2.2% to 6% better than unassisted humans on data of different conditions when the AI's accuracy was 84% [9]. This complementarity effect follows from the independence of the error distributions across components: the model allows for systematic upscaling of tasks by eliminating the need to reformulate organizational accountability frameworks, but it requires retraining workers and monitoring tasks between them.

5.2 The Cyborg Model

Tightly coupled cyborg architectures can steer all components of a system through simultaneous steering, iterative feedback and co-creative dialogue. A system of this type can change on all levels, while AI components can adapt to changing human preferences through partial dependence on a constant cycle of human-AI interaction. People have been shown to use AI predictions as priors or for backup verification, without blindly trusting them. However, users of AI systems process information differently depending on the order of information presentation. Prior (vs posterior) presentation reduces cognitive costs but introduces anchor effects and reduces cognitive independence [9]. Where performance is less than deterministic, such as in Cyborgs, additional issues arise around accountability because human input is now part of the decision-making process and most customary causal based accountability theories do not apply where human and computer contributions cannot be separated.

5.3 Practical Deployment Considerations

Enterprise AI is generally focused in low-risk, high-impact use cases to enable efficiency while ensuring proper product quality and regulatory compliance. An analysis of 10 product categories by 49 HCI researchers has led to 18 design rules across four phases: promoting capabilities and limitations, providing context-appropriate information, explaining and fixing errors, and cautiously adapting to user behavior. An important factor is whether expert domains are complementary. Collaboration improves if AI predictions reduce human task difficulty, but degrades if both make similar prediction errors [9]. Another challenge is calibration of reliance, rather than uncalibrated trust: for example, AI explanations may paradoxically increase human agreement with erroneous AI advice [9]. Obstacles to implementation include inter-system integration, data quality, worker pushback, response to regulatory requirements in judgment-intensive domains, and designing interfaces for the appropriate degrees of trust.

Table 2: Centaur vs. Cyborg: Human-AI Collaboration Models Compared [9, 10]

Characteristic	Centaur Model	Cyborg Model
Task Allocation	AI handles routine automation and pattern recognition; Humans handle ethical decisions and oversight	Tightly coupled workflows with simultaneous steering and co-creation
Coupling Type	Modular coupling with prescribed interfaces and handoffs	Dependent coupling with iterative feedback and refinement
Decision Making	Sequential: AI processes then human approves	Simultaneous: Iterative revision and dynamic adjustment
Organizational Impact	Worker retraining; New monitoring roles; Maintains existing structures	Fundamentally alters structures; Unclear responsibility assignment
Accountability	Clear lines of accountability maintained	Theories of causal attribution do not apply
Human Expertise Dependency	Performance does not drop dramatically without human expertise	Performance drops significantly without human expertise
Best Use Cases	Systematic task augmentation; Computationally intensive operations; High throughput tasks	Co-creation; Iterative refinement; Tasks requiring continuous human-AI exchange
AI Presentation Timing	Can present predictions before or after human decision	Timing critical: Before can cause anchor effects; After preserves independence

Error Pattern Consideration	Works best when human and AI errors minimally overlap	Requires careful design to avoid over-reliance on AI

Conclusion: Toward Augmented Intelligence Systems

Recent advances in AI architecture have increasingly pointed toward composite architectures for LLMs as a specialized sub-component of a deep multi-layered pipeline seeking to address the theoretical limitations of current-generation generative models. Retrieval-augmented generation systems, agentic workflows of models using tools, models fine-tuned for specific domains of services, and orchestrated human-in-the-loop workflows all contribute towards a model of orchestrated intelligence, which, instead of relying on monolithic model capability, leverages component specialization and covers wide swaths of systemic gaps more efficiently. The benefits to organizations of using this composite techniques include access to external knowledge bases for ground-truth data, distribution of different reasoning types for agent specialization, optimized model performance through closed-loop feedback and continuous retraining, and maintaining some degree of human agency by retaining control at the critical decision points for regulation, ethical oversight, accountability, and expert intervention. It is through augmentation that mission-critical, safe, dependable, and enterprise-ready AI systems can be created. Composite architectures are the answer, with the precision necessary to comply with regulations, the latency and speed needed for real-time performance, the auditability required by regulatory oversight and governance, and the generality to span multiple use cases. Viewing foundation models as efficient engines in a larger control architecture allows us to address hallucination (via retrieval-augmented generation), knowledge staleness (via retrievability), computational cost (via routing and caching), and capability scaling (via modular growth). This is an architectural perspective that goes beyond engineering optimization and reimagines how to integrate AI systems with enterprise-level workflows and investments. The most capable systems do not come from building larger, general-purpose architectures but from deliberately integrating statistical learning, symbolic reasoning, access to external knowledge and tools, and interaction with human experts across diverse tasks.

References

[1] Patrick Lewis et al., "Retrieval-Augmented Generation for Knowledge-Intensive NLP Tasks," Advances in Neural Information Processing Systems 33 (NeurIPS 2020), 2020. Available: https://proceedings.neurips.cc/paper/2020/hash/6b493230205f780e1bc26945df7481e5-Abstract.html

[2] Shehzaad Dhuliawala et al., "Chain-of-Verification Reduces Hallucination in Large Language Models," arXiv, 2023. Available: https://arxiv.org/pdf/2309.11495

[3] Zhengbao Jiang et al., "Active Retrieval Augmented Generation," Association for Computational Linguistics, 2023. Available: https://aclanthology.org/2023.emnlp-main.495/

[4] Ziwei Ji et al., "Survey of Hallucination in Natural Language Generation," ACM Computing Surveys, vol. 55, no. 12, 2023. Available: https://dl.acm.org/doi/10.1145/3571730

[5] Yilun Du et al., "Learning Iterative Reasoning through Energy Diffusion," Proceedings of the 41st International Conference on Machine Learning, 2024. Available: https://proceedings.mlr.press/v235/du24f.html

[6] Denny Zhou et al., "Least-to-Most Prompting Enables Complex Reasoning in Large Language Models," ICLR 2023, 2023. Available: https://openreview.net/pdf?id=WZH7099tgfM

[7] Aakanksha Chowdhery et al., "PaLM: Scaling Language Modeling with Pathways," Journal of Machine Learning Research, 2023. Available: https://jmlr.org/papers/volume24/22-1144/22-1144.pdf

[8] Sharan Narang et al., "Do Transformer Modifications Transfer Across Implementations and Applications?", Association for Computational Linguistics, 2021. Available: https://aclanthology.org/2021.emnlp-main.465/

[9] Gagan Bansal et al., "Does the Whole Exceed its Parts? The Effect of AI Explanations on Complementary Team Performance," ACM, 2021. Available: https://dl.acm.org/doi/10.1145/3411764.3445717

[10] Saleema Amershi et al., "Guidelines for Human-AI Interaction," ACM, 2019. Available: https://dl.acm.org/doi/10.1145/3290605.3300233