Surveying Modern AI Systems: LLMs, RAG, and Intelligent Agent Architectures

Sebastian Balderrama[1], Paul Matsialko[2], Harsh Shekhar Koli[3], Haitham Timimi[4], Maheen Zehra[5]

Abstract— **The introduction of Large Language Models (LLMs) has given a new twist to NLP, with high-end applications such as text generation, summarization, translation, and question-answering systems. Most recently, limitations such as factual inaccuracy and static knowledge have led to the development of Retrieval-Augmented Generation (RAG), whereby researchers augment LLMs with external information retrieval systems to further improve the accuracy of results based on relevance.**

This paper thus briefs on four major available LLM platforms: OpenAI's GPT-4, Meta's LLaMA 3, Google's PaLM2, and Deepseek's R1. This paper also looks at implementations such as those provided by LangChain and Llamaindex, which support the RAG-based solution approach. The architecture, capabilities, and trade-offs of each model are examined, as well as pertinent challenges such as computational cost, ethical issues, and fine-tuning complexity. This study is a foundational overview, through the use of academic papers and related works, on understanding and evaluating the growing ecosystems of LLM and RAG frameworks.

I. INTRODUCTION

AI models, since their inception, have experienced a myriad of technological advancements, many of which have been implemented for mainstream use. The rise of LLMs specifically has helped to trailblaze in the field of natural language processing. Some prominent NLP models include Open AI's GPT-4, Meta's LLaMA 3, Google's PaLM2, and Deepseek's R1. All of these models have proved themselves to be capable of performing a wide range of tasks and specifically being able to break down large datasets to generate human-like text. As these models evolve, their architectures, training methodologies, and applications diversify, making comparative studies essential to understand their strengths and limitations. In the following, we provide an overview of these models, highlighting their innovations, performance, and contributions to the landscape of artificial intelligence.

LLaMA (Large Language Model Meta AI) challenges the conventional wisdom that larger models inherently perform better [1]. Developed by Meta, LLaMA focuses on optimizing inference efficiency by training smaller models (7B to 65B parameters) on extensive datasets, achieving competitive performance with models like GPT-3 despite being significantly smaller. For instance, LLaMA-13B outperforms GPT-3 on most benchmarks while being ten times smaller [1]. A key advantage of LLaMA is its reliance on publicly available data, making it compatible with open-source initiatives. This democratizes access to high-performance LLMs, enabling research with limited computational resources [1]. The model also incorporates architectural modifications to the transformer and emphasizes responsible AI practices, including bias and toxicity evaluations [1]. Other models like Google's PaLM2 [2] prioritize multilingualism and compute-optimal scaling, reflecting a broader trend toward balancing performance and resource allocation.

PaLM2 [2], the successor to PaLM, introduces advancements in multilingual understanding and compute-optimal scaling. Unlike earlier models that prioritized model size, PaLM2 balances data and model scale for optimal performance, validating the scaling laws proposed by Hoffmann et al. (2022) [2]. Its pre-training corpus spans

[1]Sebastian Balderrama, 212072020
[2]Paul Matsialko, 169028235
[3]Harsh Shekhar Koli, 000006799
[4]Haitham Timimi, 000006805
[5]Maheen Zehra, 13116031

hundreds of languages and domains, including programming and mathematics, while maintaining English proficiency [2]. PaLM2 also integrates a mixture of pre-training objectives, improving versatility. Despite its smaller size compared to PaLM, it outperforms its predecessor in generation, translation, and reasoning tasks, even achieving teaching-level language proficiency [2]. PaLM2's multilingual and efficient design complements the rise of general-purpose multimodal models like GPT-4 [3], which expands capabilities beyond text to include visual inputs.

GPT-4 [3] represents a leap forward in multimodal processing, handling both text and image inputs to generate coherent outputs. Evaluated on human-designed exams, GPT-4 scores in the top 10% of test-takers, a significant improvement over GPT-3.5 [3]. It surpasses state-of-the-art models in 24 of 26 languages on translated benchmarks and excels in traditional NLP tasks [3]. Despite its achievements, GPT-4 retains limitations such as hallucinations and a finite context window, necessitating careful deployment in high-stakes scenarios [3].

While GPT-4 showcases generalist capabilities, specialized advancements in reasoning, such as those in DeepSeek-R1 [4], highlight the potential of reinforcement learning to unlock niche LLM abilities.

DeepSeek-R1 [4] pioneers pure reinforcement learning (RL) to enhance reasoning without supervised data. Starting with the DeepSeek-V3 base model, iterative RL training improves its pass@1 score on the AIME 2024 benchmark from 15.6% to 71.0% [4]. To address readability and language-mixing issues, the model combines cold-start data, supervised fine-tuning, and RL in a multi-stage pipeline [4]. Distilled versions of DeepSeek-R1, such as the 14B and 32B models, set new benchmarks for open-source reasoning performance, even rivaling proprietary systems like OpenAI's o1 series [4].

LLM's, despite their impressive feats, do face a variety of inherent drawbacks. They are limited to a static database, where once the models are trained, their knowledge of a subject doesn't evolve, unless it is specifically retrained to do so. These limitations can lead to factual inconsistencies, hallucination, and inability to perform as efficiently under time constraints and react in real time to real world problems.

Retrieval-Augmented Generation (RAG) approach by contrast, has incorporated external retrievers that can retrieve data in real time to solve the task at hand. This then forwarded to an LLM generator- which generates a more robust, and fully contextualized response. As a result, RAG has become the foundational cornerstone for developing more flexible and reliable AI.

II. Related Works

While high-level surveys provide useful syntheses of trends, technical reports and academic papers remain indispensable for rigorous LLM research. These primary sources offer detailed descriptions of model designs (e.g., LLaMA's rotary embeddings [1], PaLM2's mixture-of-experts routing [2]) that surveys tend to simplify and raw performance metrics (described in detail in the coming section) without the interpretation bias that surveys might introduce. These technical reports also provide insight into each LLM's individual architecture:

- **LLaMA:** Touvron et al. [1] demonstrated that smaller models (7B–65B parameters) trained on extensive datasets can match larger counterparts like GPT-3 in performance while optimizing inference efficiency—a key consideration for real-world deployment.
- **PaLM2:** Anil et al. [2] validated compute-optimal scaling laws [3], showing that multilingual training and balanced data-model scaling improve performance beyond sheer parameter count.
- **GPT-4:** OpenAI [4] introduced multimodal capabilities (text:image processing) and safety enhancements, though limitations like hallucinations persist.
- **DeepSeek-R1:** DeepSeek-AI [5] pioneered pure reinforcement learning for reasoning, achieving 71% pass@1 on AIME 2024 without supervised fine-tuning.

Related works in the form of recent surveys provide useful syntheses of trends and give perspective across different models. The following surveys were specially useful:

1) **LLM Survey:** Mùngce et al. [23] taxonomized language models into four waves (statistical → neural → pre-trained → LLMs), highlighting emergent abilities like in-context learning. While thorough, their coverage ends in 2023, omitting newer models like DeepSeek-R1.
2) **RAG Survey:** Gao et al. [17] systematically categorized RAG into *Naive*, *Advanced*, and *Modular* stages, analyzing retrieval/generation/augmentation techniques. Their benchmark of 50+ datasets is particularly relevant to our RAG implementation comparisons.
3) **Agentic RAG Survey:** Singh et al. [18] extended this by integrating autonomous agents for dynamic retrieval workflows. Their taxonomy of *single-agent*, *multi-agent*, and *graph-based* frameworks informs our analysis of agentic RAG systems.

Our work bridges these perspectives by updating [23] with 2024 models (DeepSeek-R1, GPT-4), extending [17]'s RAG benchmarks to include agentic adaptations from [18] and provides the side-by-side evaluation of how LLaMA, PaLM2, GPT-4, and DeepSeek-R1 perform in both traditional and agentic RAG pipelines (LangChain and Llamaindex).

III. Methodology

Although previous sections have illuminated the distinct strengths of modern LLMs (e.g., LLaMA's efficiency, GPT-4's multimodality) and touched upon RAG and Agentic RAG, these advances have yet to be compared within their own categories. Our methodology addresses this gap through two parallel analyses:

First, we conduct a comparison of the four LLMs—LLaMA, PaLM 2, GPT-4, and DeepSeek-R1 across five rigorously defined dimensions, using only metrics reported in their original technical reports and supporting academic papers.

Then, we separately evaluate RAG implementations (RECOMP, Self-RAG, and Agentic RAG) through five metrics, drawing on controlled benchmarks from ADD CITE. This approach ensures fair comparisons within each technology class.

A. LLM Comparison

The five metrics used for LLM comparison are the following:

1) **Model Architecture**: Transformer variants, parameter counts, and specialized components (e.g., mixture-of-experts (MoE) layers, reinforcement learning (RL) training).
2) **Training Data**: Dataset size, composition (multilingual vs. monolingual), and preprocessing techniques.
3) **Performance**: Standardized benchmarks including MMLU (massive multitask language understanding), GSM8K (mathematical reasoning), HumanEval (code generation), and specialized tasks like AIME 2024.
4) **Efficiency**: Inference speed, hardware requirements, and quantization support.
5) **Accessibility**: Licensing terms and availability for research or commercial use.

The data for this analysis was sourced from technical reports published by Meta, Google, OpenAI, and DeepSeek-AI and supporting papers, ensuring a focus on measurable differences rather than speculative claims.

1. Model Architectures

LLaMA [1] employs a standard Transformer architecture enhanced with RMSNorm for pre-normalization, SwiGLU activation functions, and rotary positional embeddings (RoPE). It ranges from 7B to 65B parameters and uses a dense architecture, prioritizing hardware-friendly deployment over specialized scaling techniques.

PaLM 2 [2] builds on the Transformer framework but introduces compute-optimal scaling, where model size and training tokens grow proportionally. It is smaller than its predecessor (PaLM 540B) but achieves superior performance through refined data quality and architectural adjustments, including a tuned mixture of pretraining objectives inspired by UL2.

GPT-4 [3]is a multimodal Transformer capable of processing both text and image inputs. While OpenAI has not disclosed full architectural details, the model reportedly employs MoE techniques to balance computational cost and performance. Its design emphasizes predictable scaling, enabling accurate loss prediction from smaller proxy models.

DeepSeek-R1 [4] distinguishes itself with an RL-first approach. Unlike the other models, DeepSeek-R1-Zero is trained purely via reinforcement learning without supervised fine-tuning (SFT), resulting in emergent self-verification and reflection capabilities. The hybrid pipeline combines RL with cold-start data and multi-stage training to address readability issues. Additionally, DeepSeek-R1 distills reasoning capabilities into smaller models (1.5B–70B), outperforming traditional RL-trained counterparts [4].

2. Training Data

LLaMA relies on 1.4 trillion tokens from public datasets, including CommonCrawl (67%), C4 (15%), GitHub (4.5%), and Wikipedia (4.5%). Its training corpus is predominantly English-focused, with limited multilingual or domain-specific pretraining.

PaLM 2 uses a more diverse dataset, with 20% non-English text spanning hundreds of languages. It incorporates parallel multilingual documents, code, and scientific texts (e.g., arXiv), alongside safety-focused features like "canary tokens" for memorization measurement.

GPT-4's training data composition is undisclosed, but it includes both publicly available and licensed third-party content. The model demonstrates strong multilingual performance, outperforming English-language benchmarks in 24 of 26 languages on MMLU. Its multimodal training (image + text) further differentiates it, though image capabilities remain incompletely documented.

DeepSeek-R1 leverages cold-start data consisting of thousands of long chain-of-thought (CoT) examples fine-tuned before RL training. The RL-optimized pipeline employs rule-based rewards (accuracy + format) and model-based rewards for alignment, enabling high sample efficiency in reasoning tasks.

3. Performance

The models were evaluated on standardized benchmarks:

- **MMLU (5-shot)**: GPT-4 leads with 86.4%, followed by DeepSeek-R1 (90.8%), PaLM 2-L (69.3%), and LLaMA-65B (63.4%).
- **GSM8K (CoT)**: DeepSeek-R1 achieves state-of-the-art performance (97.3%), surpassing GPT-4 (92.0%) and PaLM 2-L (91.0%).
- **HumanEval (Pass@1)**: GPT-4 scores highest (67.0%), with DeepSeek-R1 close behind (65.9%), while PaLM 2-S (code-tuned) and LLaMA-65B trail at 37.6% and 23.7%, respectively.
- **AIME 2024 (Pass@1)**: DeepSeek-R1 dominates (79.8%), far outperforming DeepSeek-V3 (39.2%), the only other model evaluated on this specialized reasoning task.

Key Insights:

- DeepSeek-R1 excels in reasoning-intensive tasks (e.g., AIME, GSM8K) but trails GPT-4 in general knowledge (MMLU).
- GPT-4 leads in broad benchmarks, while PaLM 2 performs strongly in multilingual tasks.
- LLaMA's performance is competitive given its smaller size and open-source nature.

4. Efficiency

LLaMA is optimized for single-GPU inference, with the 13B model running efficiently on consumer-grade hardware (e.g., a single V100). It supports 8-bit quantization without significant performance loss.

PaLM 2 emphasizes compute-efficient inference on Google's TPUv4 infrastructure, with smaller models outperforming larger predecessors. Quantization capabilities are undocumented.

GPT-4 incurs high inference costs due to its scale and multimodality, though specifics are undisclosed. Its hardware requirements are datacenter-scale. DeepSeek-R1's RL training is computationally expensive but achieves high sample efficiency (e.g., AIME pass@1 improves from 15.6% to 71.0% via RL). Distilled smaller models (e.g., 7B) outperform larger counterparts like Qwen-32B on reasoning tasks, mitigating some efficiency concerns.

5. Licensing and Accessibility

LLaMA is open-source under a non-commercial license, making it ideal for academic research. Its public training data and efficient inference further enhance accessibility.

PaLM 2 and GPT-4 are proprietary, available only via API endpoints. While PaLM 2's technical report provides detailed scaling laws, GPT-4's opacity contrasts with the openness of Meta and Google.

DeepSeek-R1 and DeepSeek-V3 are open-source, with model weights publicly available. This positions DeepSeek favorably for research and commercial applications requiring transparency.

Quick Discussion

LLaMA stands out for its accessibility and efficiency, making it a practical choice for many applications. However, it falls short in supporting multilingual and multimodal tasks. PaLM 2, on the other hand, strikes a balance between performance and efficiency, particularly excelling in multilingual capabilities, though its proprietary nature limits broader access. GPT-4 dominates general benchmarks and offers advanced multimodality, but its high costs and lack of transparency pose significant drawbacks. Meanwhile, DeepSeek-R1 achieves state-of-the-art performance in reasoning tasks but struggles with language mixing and incurs high training expenses.

For research purposes, LLaMA and DeepSeek-R1 are ideal due to their open-source availability and specialized functionalities. In enterprise settings, PaLM 2 provides scalability, while DeepSeek-V3 emerges as a cost-effective solution for multilingual needs. When it comes to cutting-edge applications, GPT-4 remains unparalleled in multimodality, whereas DeepSeek-R1 leads the pack in mathematical reasoning.

While architectural and training innovations define the foundational capabilities of LLMs, their real-world applicability is often enhanced through retrieval-augmented generation (RAG). Traditional RAG systems, while effective, often struggle with challenges such as computational inefficiency, static retrieval strategies, and limited adaptability to complex queries. Recent advancements, including RECOMP (Xu et al., 2023), Self-RAG (Asai et al., 2023), and Agentic RAG (Zhao et al., 2024), address these limitations through distinct yet complementary innovations. We now evaluate how three advanced RAG frameworks (RECOMP, Self-RAG, and Agentic RAG) optimize this synergy between parametric knowledge and retrieval.

B. RAG Implementation Comparison

To compare RECOMP [13], Self-RAG [12], and Agentic RAG [18][19], we utilized the following criteria:

1) **Retrieval Mechanism**: How each system retrieves and integrates external knowledge.
2) **Compression and Summarization**: Techniques for reducing document length and preserving relevance.
3) **Adaptability and Reflection**: Dynamic adjustments during retrieval and generation.
4) **Performance on Downstream Tasks**: Empirical results on benchmarks such as question-answering and long-form generation.
5) **Architectural Complexity**: Computational overhead and coordination requirements.

The analysis draws from methodologies and experimental results presented in the respective papers, ensuring an evidence-based comparison.

1. Retrieval Mechanism

RECOMP assumes a fixed set of retrieved documents and employs extractive or abstractive compressors to distill relevant information. It selectively prepends summaries or omits them entirely, prioritizing efficiency.

Self-RAG dynamically decides when to retrieve documents using reflection tokens (e.g., [ISREL], [ISSUP]), enabling adaptive retrieval and critique. This ensures retrieval occurs only when beneficial, reducing unnecessary computation.

Agentic RAG extends retrieval with autonomous agents capable of planning, tool use, and multi-step reasoning. Systems like Router models centralize decision-making, while multi-agent frameworks distribute tasks for complex workflows. Graph-based approaches (e.g., Agent-G) enhance multi-hop reasoning by integrating structured and unstructured data.

2. Compression and Summarization

RECOMP achieves high compression rates (5–10% of original document size) using extractive and abstractive techniques, minimizing performance loss.

Self-RAG does not explicitly compress documents but filters irrelevant passages via reflection tokens, implicitly reducing noise.

Agentic RAG leverages agents like Relevance Evaluators to iteratively refine retrieved content, balancing summarization with contextual depth.

3. Adaptability and Reflection

RECOMP adapts by omitting irrelevant sum-

maries but lacks dynamic self-assessment during generation.

Self-RAG excels in self-reflection, critiquing outputs and adjusting retrieval strategies in real-time.

Agentic RAG introduces corrective mechanisms (e.g., Query Refinement Agents) and hierarchical workflows, enabling granular adaptability for complex tasks.

4. Performance on Downstream Tasks

RECOMP excels in efficiency-centric tasks, achieving strong performance with minimal token usage (e.g., language modeling, QA).

Self-RAG outperforms in fact-intensive tasks (e.g., open-domain QA) due to its citation accuracy and self-critique capabilities.

Agentic RAG dominates in dynamic, multi-step scenarios (e.g., healthcare analytics, legal document processing), where iterative refinement and multi-agent collaboration are critical.

5. Architectural Complexity RECOMP and Self-RAG are computationally lighter but less suited for complex workflows.

Agentic RAG incurs higher overhead due to agent coordination but offers unparalleled flexibility for real-time, scalable applications.

Tools & Framework for LLM+RAG Integration

The implementation of Retrieval-Augmented Generation (RAG) systems—particularly those with agentic capabilities—relies heavily on modular frameworks that streamline data ingestion, retrieval, and reasoning workflows. Two tools have emerged to help build such pipelines: **LangChain**, a framework for orchestrating multi-agent RAG workflows, and **Llamaindex**, specialized for optimizing document indexing and retrieval. Below, we analyze their roles through real-world deployments documented in recent papers.

Langchain

Langchain [21] is currently the most popular RAG pipeline. It's a python framework that actually gives LLM's the tools and library to retrieve external data, abstract from documents, and maintain a memory retentive chatbot. In the Agentic RAG paper by Zhao et al. 2024, Langchain was employed to design Agent-G which adapted a modular framework. Different "sub-agents" were responsible for retrieval, planning, and actual solution generation. Each one had its own respective memory and rationalization process. The paper itself states, **"We use LangChain's high-level abstractions to integrate tools, memory, and agents... Each agent's memory is managed separately using LangChain's context handling modules"** (Zhao et al., p. 8).

Langchain [21]implementation has also transcended experimentation. In a paper published by the IEEE in 2023, a Langchain bot was used for a university help desk, where it would routinely draw answers from official documents all while retaining it's memory throughout the chat.

LLAMAINDEX

Llamaindex [22] is primarily used for data ingestion and indexing. It's model draws from various types of data, PDF's, websites, and Notion Pages, that it transforms into retrievable chunks of data that an LLM can later pull from. Back in 2024, Llamaindex was implemented into a medical question-answering system. They indexed radiology report which they then imbedded using Sentence-BERT and then stored them in FAISS (Facebook AI Similarity Search), the LLM then retrieves the clinical data during question answering.

"We first used Llamaindex to chunk and embed all radiology reports... These embeddings were stored in FAISS and retrieved via cosine similarity" (Ma et al., p. 3).

This implementation helps limit hallucinations as the model is drawing directly from the documents, which helps limit the possibility of error in the field of healthcare.

Whereas **Langchain's** primary purpose it to draw together different components to create a structured networks (i.e prompts, retrievers, etc.)—**Llamaindex** is more oriented to generating bite sized data for the retrieval to pull from memory whenever it is required. In conclusion **Langchain** tends to construct custom ai-agents whereas **Llamaindex** is used primarily for memory.

Future Directions

RECOMP, Self-RAG, and Agentic RAG represent progressive advancements in retrieval-augmented generation, each addressing distinct

challenges. RECOMP prioritizes efficiency, Self-RAG enhances factuality through self-reflection, and Agentic RAG enables dynamic, agent-driven workflows.

Future research should explore hybrid models, such as integrating RECOMP's compression with Agentic RAG's multi-agent frameworks, to optimize both efficiency and adaptability. Additionally, developing standardized benchmarks for Agentic RAG and addressing ethical considerations in autonomous decision-making will be critical for widespread adoption.

IV. Results and Discussion

1. Benchmark Performance and Model Capabilities

We evaluated state-of-the-art LLMs across standardized benchmarks to assess their reasoning, knowledge, and generative abilities (refer to Table 1 below):

The benchmark results reveal clear architectural strengths. Decoder-only models like GPT-4 dominate generative and commonsense reasoning tasks, evidenced by its 95.3% HellaSwag score, but require significant computational resources. In contrast, DeepSeek-R1 excels in specialized domains, achieving 97.3% on GSM8K (math) and 79.8% on AIME 2024, though it slightly trails GPT-4 in broad-knowledge benchmarks like MMLU (90.8% vs. 86.4%). This suggests a growing divide between generalist and specialist model capabilities.

2. Model-Specific Strengths and Limitations

Refer to Table 2 below.

GPT-4's decoder-only architecture enables superior multimodal reasoning but at datacenter-scale costs, while LLaMA 2's open-source design allows efficient single-GPU inference (13B model with 8-bit quantization) at the cost of requiring fine-tuning. PaLM 2 trades long-context capabilities for multilingual prowess.

3. RAG Variants: Performance and Trade-offs

RECOMP achieves 5–10% document compression via extractive summarization, ideal for latency-sensitive applications, while **Self-RAG** improves QA accuracy by 12% through reflection tokens at the cost of added computation. **Agentic RAG**'s multi-agent workflow increases latency (20%) but enables unparalleled precision in healthcare and legal analytics.

A. 4. Architectural Trade-offs and Future Directions

Decoder-only architectures, exemplified by GPT-4's auto-regressive design, remain the gold standard for text generation tasks but incur substantial computational costs due to their sequential token prediction mechanism. Hybrid approaches like RETRO are gaining traction by strategically combining the bidirectional context processing of encoders with the generative capabilities of decoders, offering a promising balance between efficiency and creativity. Simultaneously, the trend toward specialization is yielding compelling results, with compact models such as Phi-2 demonstrating superior performance to larger general-purpose counterparts in domain-specific applications, suggesting that targeted architecture optimization may outweigh sheer scale in certain use cases.

B. Future Work

Three critical avenues emerge for advancing LLM capabilities. First, Agentic RAG systems require optimization to mitigate their inherent latency penalties (currently slower than vanilla RAG) while maintaining the precision gains achieved through multi-agent collaboration, particularly in high-stakes domains like healthcare and legal analytics. Second, quantization techniques must evolve to enhance the practical deployability of open-source models, with LLaMA and T5 serving as key testbeds for achieving GPU-efficient inference without sacrificing performance. Finally, the field urgently needs standardized cross-model benchmarks tailored to RAG-enhanced LLMs to enable rigorous, apples-to-apples comparisons of retrieval-augmented systems, moving beyond the current fragmentation of evaluation methodologies. These developments would collectively address the pressing challenges of efficiency, scalability, and measurable progress in next-generation language models.

TABLE I
BENCHMARK PERFORMANCE OF STATE-OF-THE-ART LLMS

Benchmark	ARC-Challenge
Purpose	Scientific reasoning (MCQ)
Top Model (Score)	GPT-4 (85.3%)
Implication	Strong factual/logical inference
Source	GPT-4 Technical Report [3]
Benchmark	HellaSwag
Purpose	Commonsense reasoning
Top Model (Score)	GPT-4 (95.3%)
Implication	Human-like scenario prediction
Source	GPT-4 Technical Report [3]
Benchmark	GSM8K
Purpose	Multi-step math
Top Model (Score)	DeepSeek-R1 (97.3%)
Implication	Symbolic reasoning prowess
Source	DeepSeek-R1 Paper [4]
Benchmark	MMLU
Purpose	Broad knowledge (57 tasks)
Top Model (Score)	DeepSeek-R1 (90.8%)
Implication	Domain versatility
Source	DeepSeek-R1 Paper [4]
Benchmark	AIME 2024
Purpose	Complex reasoning
Top Model (Score)	DeepSeek-R1 (79.8%)
Implication	Specialized task superiority
Source	DeepSeek-R1 Paper [4]

TABLE II
MODEL-SPECIFIC STRENGTHS AND LIMITATIONS

Model	GPT-4
Architecture	Decoder-only
Best Benchmark (Score)	HellaSwag (95.3%)
Unique Advantage	Multimodal reasoning
Key Limitation	Closed-source; high cost
Source	GPT-4 Technical Report [3]
Model	PaLM 2
Architecture	Decoder-only
Best Benchmark (Score)	GSM8K (65.8%)
Unique Advantage	Multilingual fluency
Key Limitation	Weak long-context handling
Source	PaLM 2 Paper [2]
Model	LLaMA 2
Architecture	Decoder-only
Best Benchmark (Score)	MMLU (68.9%)
Unique Advantage	Open-source customization
Key Limitation	Requires fine-tuning
Source	LLaMA-2 Paper [5]

TABLE III
RAG VARIANTS: PERFORMANCE AND TRADE-OFFS

Approach	RECOMP
Compression	Extractive (5–10% doc size)
Adaptability	Static summarization
Best Use Case	Efficiency-centric QA
Source	RECOMP Paper [13]
Approach	Self-RAG
Compression	Noise reduction via reflection
Adaptability	Real-time self-critique
Best Use Case	Fact-intensive QA (citations)
Source	Self-RAG Paper [12]
Approach	Agentic RAG
Compression	Iterative refinement by agents
Adaptability	Multi-step workflows (e.g., healthcare)
Best Use Case	Dynamic, complex tasks
Source	Agentic RAG Survey [18]

V. CONCLUSION

To conclude this paper, Large Language Models (LLMs) and Retrieval-Augmented Generation (RAG) represent a significant advancement in AI-driven tools, balancing knowledge retention with real-time information retrieval. While LLMs excel at generating coherent and relevant text based on extensive pretraining, their limitations in handling dynamic or domain-specific information underscore the necessity of RAG. By integrating retrieval mechanisms, RAG enhances the accuracy, relevance, and adaptability of AI-generated responses, mitigating issues such as hallucination and outdated knowledge.

A notable evolution in this space is Agentic RAG, which extends traditional RAG by embedding autonomous agents capable of dynamic re-

trieval, multi-step reasoning, and self-correction. These systems—exemplified by frameworks like LangChain's Agent-G—leverage modular workflows where specialized agents (e.g., retrievers, planners, and validators) collaborate to refine outputs iteratively. While Agentic RAG achieves unparalleled precision in complex tasks like legal analytics or healthcare diagnostics, it introduces trade-offs in computational overhead and latency, demanding further optimization for widespread adoption.

However, implementing RAG (and its agentic variants) comes with challenges, including computational costs, retrieval efficiency, and the need for robust indexing strategies. As AI continues to evolve, refining RAG architectures—particularly through hybrid approaches that combine the efficiency of RECOMP with the adaptability of Agentic RAG—will be crucial in bridging the gap between static and dynamic information needs. In the end, the synergy between LLMs and RAG (from naive to agentic implementations) presents a promising path toward more reliable, context-aware, and factually grounded AI applications.

REFERENCES

[1] A. Dubey *et al.*, "The Llama 3 Herd of Models," *arXiv.org*, 2024. [Online]. Available: https://arxiv.org/abs/2407.21783

[2] R. Anil *et al.*, "PaLM 2 Technical Report," *arXiv.org*, May 17, 2023. [Online]. Available: https://arxiv.org/abs/2305.10403

[3] OpenAI, "GPT-4 Technical Report," *arXiv:2303.08774 [cs]*, Mar. 2023. doi: https://doi.org/10.48550/arXiv.2303.08774

[4] DeepSeek-AI *et al.*, "DeepSeek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2501.12948

[5] H. Touvron *et al.*, "LLaMA: Open and Efficient Foundation Language Models," *arXiv:2302.13971 [cs]*, Feb. 2023.

[6] "PaLM-E: An Embodied Multimodal Language Model," *palm-e.github.io*. [Online]. Available: https://palm-e.github.io/

[7] DeepSeek-AI *et al.*, "DeepSeek-V2: A Strong, Economical, and Efficient Mixture-of-Experts Language Model," *arXiv.org*, 2024. [Online]. Available: https://arxiv.org/abs/2405.04434

[8] DeepSeek-AI *et al.*, "DeepSeek-V3 Technical Report," *arXiv.org*, 2024. [Online]. Available: https://arxiv.org/abs/2412.19437v1 (accessed Apr. 03, 2025).

[9] M. Li, J. Sun, and X. Tan, "Evaluating the effectiveness of large language models in abstract screening: a comparative analysis," *Systematic Reviews*, vol. 13, no. 1, Aug. 2024. doi: https://doi.org/10.1186/s13643-024-02609-x

[10] V. Mavroudis, "LangChain," Nov. 2024. doi: https://doi.org/10.20944/preprints202411.0566.v1

[11] J. Li *et al.*, "The Dawn After the Dark: An Empirical Study on Factuality Hallucination in Large Language Models," *arXiv.org*, Jan. 06, 2024. [Online]. Available: https://arxiv.org/abs/2401.03205

[12] A. Asai, Z. Wu, Y. Wang, A. Sil, and H. Hajishirzi, "Self-RAG: Learning to Retrieve, Generate, and Critique through Self-Reflection," *arXiv.org*, Oct. 17, 2023. [Online]. Available: https://arxiv.org/abs/2310.11511

[13] F. Xu, W. Shi, and E. Choi, "RECOMP: Improving Retrieval-Augmented LMs with Compression and Selective Augmentation," *arXiv.org*, Oct. 06, 2023. [Online]. Available: https://arxiv.org/abs/2310.04408

[14] H.-C. Lee, K. Hung, G. M.-T. Man, R. Ho, and M. Leung, "Development of an RAG-Based LLM Chatbot for Enhancing Technical Support Service," *TENCON 2024 - 2024 IEEE Region 10 Conference (TENCON)*, pp. 1080–1083, Dec. 2024. doi: https://doi.org/10.1109/tencon61640.2024.10902801

[15] S. Simon, A. Mailach, J. Dorn, and N. Siegmund, "A Methodology for Evaluating RAG Systems: A Case Study On Configuration Dependency Validation," *arXiv.org*, 2024. [Online]. Available: https://arxiv.org/abs/2410.08801

[16] Y. Ji, H. Zhang, and Y. Wang, "Bias Evaluation and Mitigation in Retrieval-Augmented Medical Question-Answering Systems," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2503.15454 (accessed Apr. 03, 2025).

[17] Y. Gao *et al.*, "Retrieval-Augmented Generation for Large Language Models: A Survey," *arXiv.org*, Dec. 18, 2023. [Online]. Available: https://arxiv.org/abs/2312.10997

[18] A. Singh, A. Ehtesham, S. Kumar, and Khoei, Tala Talaei, "Agentic Retrieval-Augmented Generation: A Survey on Agentic RAG," *arXiv.org*, 2025. [Online]. Available: https://arxiv.org/abs/2501.09136

[19] S. Yao *et al.*, "ReAct: Synergizing Reasoning and Acting in Language Models," *arXiv.org*, Mar. 09, 2023. [Online]. Available: https://arxiv.org/abs/2210.03629

[20] T. B. Brown *et al.*, "Language Models Are Few-Shot Learners," *arxiv.org*, vol. 4, May 2020. [Online]. Available: https://arxiv.org/abs/2005.14165

[21] "LangChain," *www.langchain.com*. [Online]. Available: https://www.langchain.com/

[22] "LlamaIndex, Data Framework for LLM Applications," *www.llamaindex.ai*. [Online]. Available: https://www.llamaindex.ai/

[23] "S. Minace et al. "Large language models: A survey," ACM Comput. Surv., vol. 56, no. 3, pp. 1–38" 2024.