Towards Scalable Compound AI Systems: A Vision Paper

Richard Ding *
ChainOpera, Inc.
TensorOpera, Inc.
richard@chainopera.com

Yuhang Yao *
TensorOpera, Inc.
yuhang@tensoropera.com

Dimitris Stripelis
TensorOpera, Inc.
dimitris@tensoropera.com

Min Chang Jordan Ren†
ChainOpera, Inc.
jordan9ren8@gmail.com

Qile Wu
Nanyang Technological University
ChainOpera, Inc.
qile001@e.ntu.edu.sg

Salman Avestimehr
University of South California
TensorOpera, Inc.
ChainOpera, Inc.
avestimehr@tensoropera.com

Chaoyang He
ChainOpera, Inc.
TensorOpera, Inc.
ch@tensoropera.com

Abstract

Compound Artificial Intelligence systems have emerged as a promising approach that enables the integration of multiple AI components to improve the quality and reliability of AI-driven tasks to fit the complex requirements of modern applications. In this paper, we explore the trends, methodologies, and technical challenges associated with compound AI systems, highlighting their importance in achieving state-of-the-art performance, improving scalability, and driving innovation in next-generation AI agent systems. Specifically, we focus on the inference process, which we define as the orchestration of multiple AI components to produce reliable, task-specific outputs. In addition, compound AI systems introduce unique challenges, such as optimizing for latency, accuracy, and system-level coherence, which require new frameworks and methodologies for design, optimization, and monitoring. We also discuss future directions for infrastructure evolution to meet the demands of increasingly complex AI workflows and multi-agent systems. By examining real-world use cases, we aim to illustrate how this approach is transforming fields such as healthcare, finance, customer service, and autonomous systems, among others. We posit that the design and refinement of compound AI architectures will not only maximize the performance of current AI models but will also serve as a foundation for the next generation of adaptable high-performance AI applications.

*Equal Contributions
†Work done as research assistant at ChainOpera

1 Introduction

Rapid advances in large language models (LLMs) (Brown et al., 2020; Achiam et al., 2023; Team et al., 2023) over recent years have transformed artificial intelligence, enabling models to perform a wide range of tasks through simple prompting (Ding et al., 2023). Initially, much of the research and application development focused on the capabilities of individual models, aiming to expand and refine LLMs by scaling their architectures and training on larger datasets (Kaplan et al., 2020; Anil et al., 2023; Dubey et al., 2024). However, emerging evidence indicates a paradigm shift towards compound AI systems (Zaharia et al., 2024a), which strategically combine multiple components — such as LLMs, retrieval systems (Lewis et al., 2020b), and external tools (Schick et al., 2024) — to improve both the quality and reliability of AI-driven tasks. This evolution in system design is significant, as it reflects an essential need to achieve higher accuracy, adaptability, and control beyond what is possible through model scaling alone (Jain et al., 2024).

In this paper, we focus on the advances and open problems in compound AI systems, especially on the inference process, which we define as the orchestration of multiple AI components to produce reliable and task-specific output (Han et al., 2024b). Unlike model training, where optimization efforts focus on refining the predictive capabilities of a single model, inferencing in compound AI systems involves real-time decision making across several

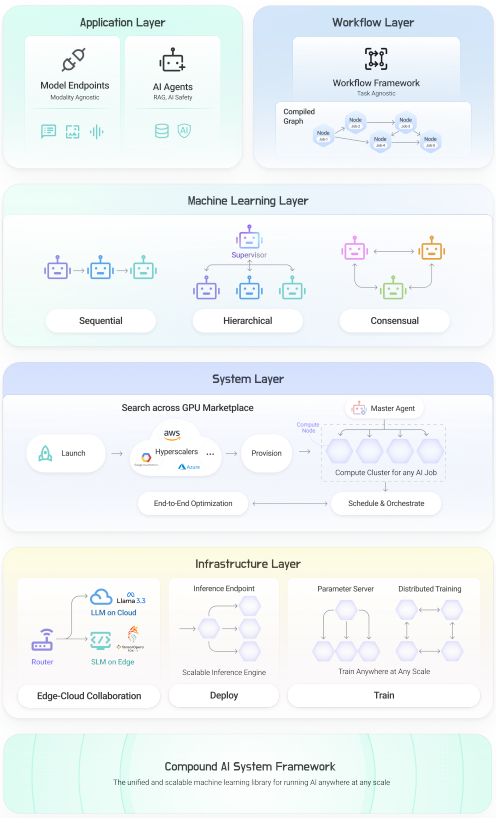

Figure 1: Five-layer-view of Compound AI System. The Application Layer provides user-facing tools like model endpoints and AI agents. The Workflow Layer supports task-agnostic orchestration frameworks. The Machine Learning Layer coordinates AI agents using sequential, hierarchical, and consensual modes. The System Layer manages compute resources via GPU marketplaces and master agents for optimization. The Infrastructure Layer enables edge-cloud collaboration, scalable inference deployment, and distributed training for AI at any scale.

components, including the retrieval and verification modules (Lewis et al., 2020a). Although the training phase of compound AI systems poses significant challenges, such as aligning diverse objectives, managing interdependencies between components, and dealing with feedback loops, our emphasis on inferencing reflects its central role in ensuring the operational effectiveness of these systems (Santhanam et al., 2024). Inferencing strategies, such as retrieval-augmented generation (RAG), tool-based reasoning, and iterative chain-of-thought sampling, underscore the need for adaptive and modular design to enhance accuracy, control, and efficiency (Snell et al., 2024). Moreover, compound AI inferencing introduces unique challenges, such as optimizing latency, accuracy, and system-level coherence, which require new frameworks and methodologies for design, optimization, and monitoring (Davis et al., 2024b). As shown in Figure 1, the compound AI system can be presented in five layers, where the challenges and components are systematically structured to support efficient, modular, and scalable AI deployments (Feng et al., 2024). Each layer addresses specific functionalities and integration requirements within the system:

1. Application Layer: This topmost layer hosts the end-user interfaces and tools, such as model endpoints that remain modality-agnostic and support diverse AI tasks. It also includes AI Agents for applications like Retrieval-Augmented Generation (RAG) and AI safety, providing robust workflows for complex problem-solving.

2. Workflow Layer: The workflow layer defines task-agnostic orchestration frameworks, enabling compiled graphs for various AI pipelines. By modularizing tasks and dependencies, it ensures flexibility and scalability across applications.

3. Machine Learning Layer: This layer features Supervisors that coordinate AI agents and workflows across three modes of operation:
   - Sequential, where agents operate in a step-by-step pipeline.
   - Hierarchical, involving layered, task-dependent control.
   - Consensual, where decisions are made collaboratively among multiple agents.

4. System Layer: At the system level, AI deployments leverage GPU marketplaces (e.g., AWS, Azure, and Hyperscalers) for launching, provisioning, and optimizing compute resources. A master agent handles scheduling, orchestration, and end-to-end optimization, ensuring smooth execution of large-scale AI workloads.

5. Infrastructure Layer: The infrastructure layer supports AI model execution and training across cloud and edge environments. It incorporates:
   - Edge-Cloud Collaboration for seamless integration between cloud-hosted and edge-based LLMs.

- Deployment mechanisms like scalable inference engines and endpoints.
- Training Infrastructure for distributed training, parameter servers, and large-scale AI training environments.

Most of the challenges associated with compound AI inferencing remain open questions, lacking clear definitions and systematic approaches for addressing them (Zaharia et al., 2024b). For instance, in retrieval-augmented generation (RAG) pipelines, a fundamental question arises: should more computational resources be allocated to improving the retriever's precision, enhancing the large language model's generation capabilities, or increasing the number of interactions between the retriever and generator? Similarly, when combining diverse models, such as integrating a text generation model with text-to-audio or text-to-video models to produce multimodal outputs, new challenges emerge. Developers must decide how to sequence these components effectively—should the generated text first undergo semantic refinement, or should it directly feed into audio and video generation pipelines with minimal preprocessing? These decisions impact both latency and coherence of the final output, yet the field offers little guidance on such integration strategies. Despite the growing prominence of compound systems in real-world applications, there is limited consensus on such critical design trade-offs, as well as on broader aspects like the design of control logic and the integration of diverse components for end-to-end optimization. These gaps highlight the need for a comprehensive vision paper that explicitly defines the inferencing phase in compound AI systems, identifies key challenges, and establishes a framework for guiding future research. Such a paper could provide the foundational concepts and methodologies required to advance this emerging area, fostering a more structured and coherent understanding of the domain.

By exploring the trends, methodologies, and technical challenges associated with compound AI inferencing, this paper aims to illuminate the growing importance of inferencing strategies in realizing the full potential of AI systems. We posit that the design and refinement of compound inference architectures will not only maximize the performance of current AI models but also serve as a foundation for the next generation of adaptable, high-performance AI applications.

This paper is structured to provide a comprehensive exploration of compound AI systems from multiple perspectives, each addressed in a dedicated Section. Section 2 defines the foundational concept of Compound AI System, establishing its scope and relevance within the context of modern AI systems. Section 3 delves into the infrastructure perspectives of compound AI system, focusing on the technical and architectural underpinnings that support the integration and operation of multiple AI components. Section 4 examines the system-level perspectives, highlighting challenges and design principles related to orchestration, modularity, and scalability. Section 5 shifts to the machine learning perspectives, discussing model-specific considerations and how training and inferencing strategies adapt within compound systems. Section 6 explores workflow perspectives, detailing the sequential and parallel interactions between components to achieve reliable and efficient outputs. Section 7 provides application-specific perspectives, showcasing how compound AI inference is tailored to domains like medical diagnostics, programming, and multimodal content generation. Section 8 discusses future directions, identifying open research questions and emerging trends in the field including agent training on compound AI systems. Finally, Section 9 reviews related work, situating this paper within the broader research landscape and emphasizing its contributions to advancing the understanding and development of compound AI systems.

2 Compound AI Systems

A Compound AI System refers to a sophisticated artificial intelligence framework that integrates multiple interacting components (Zaharia et al., 2024a), such as generative language models (LLMs) (Ding et al., 2024), retrieval systems (Cuconasu et al., 2024), symbolic engines(Sprague et al., 2024), or external tools(Qin et al., 2023), to address complex tasks through cooperative functionalities rather than relying solely on a monolithic model. This framework is essential for achieving state-of-the-art performance in scenarios where single-model approaches fall short, particularly in tasks requiring dynamic adaptability, precise control, or enhanced reliability (Shen et al., 2024). By enabling modular design and optimizing specific components for distinct functions, compound AI systems enhance performance, scalability, and user trust.

They exemplify a significant shift in AI development by demonstrating that system-level innovations, including dynamic orchestration, retrieval-augmented generation, and task-specific chains, can surpass the capabilities of even the most advanced standalone models. This paradigm is increasingly recognized as vital to maximizing AI's impact across diverse applications, offering a flexible and efficient pathway for leveraging evolving technologies (Lund and Ting, 2023).

While the concept of *Compound AI Systems* has gained traction for their ability to tackle complex tasks through a modular and cooperative framework, their success is deeply tied to *compound AI inference*—the process of orchestrating and executing the interactions between these diverse components in real-time to produce high-quality results. Compound AI inference involves not only running the individual components, such as LLMs or retrieval systems, but also managing how they communicate, exchange partial results, and adapt dynamically to task requirements. It embodies the operational essence of compound systems, transforming static architectures into dynamic and context-sensitive applications (Ehsan et al., 2021).

We focus on compound AI inference in this paper because it addresses the practical and technical challenges of deploying these systems at scale. Unlike traditional inference for standalone models, compound inference involves multiple layers of complexity, including:

- **Dynamic Workflow Execution**: Components may operate at different granularities, from token-level to paragraph-level outputs, requiring coordination across asynchronous or parallel processes (Erbel and Grabowski, 2023).

- **Adaptive Decision-Making**: Systems often involve runtime decisions, such as determining which tools to invoke or how many times to call a model, based on the input and intermediate results. This demands intelligent and context-aware orchestration (Liu et al., 2022).

- **Efficiency and Scalability**: Managing latency and computational costs is critical, especially when serving large-scale applications. Compound inference must optimize resource allocation across heterogeneous components to maintain real-time performance while minimizing overhead (Meng, 2024).

- **Quality Optimization**: Compound inference enables iterative refinement of results by chaining multiple models or steps, allowing for improved accuracy through techniques like self-consistency, ensembling, and contextual verification (Okuyelu and Adaji, 2024).

- **Trust and Transparency**: By explicitly structuring interactions between components, compound inference makes it easier to track the provenance of results, debug errors, and enhance user trust through techniques like fact-checking and evidence-based responses. (Olateju et al., 2024)

The emphasis on compound AI inference stems from its pivotal role in operationalizing the advantages of compound systems. While compound systems inherently offer flexibility, reliability, and enhanced functionality, their potential can only be fully realized through efficient and intelligent inference strategies. This paper delves into the emerging methodologies, challenges, and solutions for compound AI inference, highlighting its importance in achieving state-of-the-art performance, improving scalability, and driving innovation in next-generation AI systems.

3 Infrastructure Perspectives of Scalable Compound AI Systems

The design and deployment of scalable compound AI systems necessitate careful consideration of the underlying computational infrastructure. With the proliferation of diverse hardware architectures and network configurations, selecting an appropriate deployment strategy significantly influences system performance, latency, and cost-effectiveness. In this section, we examine three primary infrastructure paradigms: on-device, cloud-based, and edge-cloud hybrid systems, emphasizing their respective roles and challenges in supporting compound AI systems. We also discuss future directions for infrastructure evolution to meet the demands of increasingly complex AI workflows.

3.1 On-Device AI Systems

On-device AI systems leverage local computational resources, such as GPUs, TPUs, or dedicated AI accelerators integrated into smartphones, IoT devices, or edge gateways. These systems prioritize low-latency inference and data privacy by processing

data locally, eliminating the need for data transmission to external servers. On-device systems are particularly effective for applications with stringent real-time requirements, such as autonomous vehicles, wearable health monitoring, and natural user interfaces (Ham et al., 2021).

However, the constrained computational and memory resources in on-device systems pose challenges for deploying compound AI architectures. Training Small Language Models like Fox-1 (Hu et al., 2024b) is designed to reduce the size of LLMs with the same capability. Techniques such as model quantization, pruning, and distillation are commonly employed to adapt complex models for on-device execution. Additionally, resource-aware orchestration frameworks, such as TensorFlow Lite and PyTorch Mobile, are vital for optimizing multi-component pipelines in resource-constrained environments.

3.2 Cloud-Based AI Systems

Cloud-based AI systems utilize centralized data centers to perform computationally intensive tasks, offering virtually unlimited scalability and access to specialized hardware such as GPUs, TPUs, and FPGA clusters. Cloud infrastructure is ideal for training and inferencing large-scale compound AI models, as it facilitates seamless integration of multiple components, such as retrieval systems, large language models (LLMs), and external APIs (Witanto et al., 2022).

Modern cloud platforms provide flexible orchestration frameworks, such as Kubernetes and Apache Kafka, to manage distributed compound AI workflows. These platforms enable dynamic resource allocation and fault-tolerant execution of tasks, ensuring robust performance under varying workloads. However, reliance on cloud systems introduces latency due to data transmission and poses potential security and compliance challenges for sensitive data.

3.3 Edge-Cloud Hybrid AI Systems

Edge-cloud hybrid systems integrate the strengths of on-device and cloud-based paradigms by partitioning workloads across edge devices and centralized cloud resources (e.g. TensorOpera Router (Stripelis et al., 2024)). This approach is particularly well-suited for compound AI systems that require both low-latency processing and access to powerful computational resources. For instance, initial inference steps may be performed on edge devices, while more complex tasks, such as multi-step reasoning or large-scale data retrieval, are offloaded to the cloud (Williams et al., 2024).

Key challenges in edge-cloud systems include optimizing task partitioning, minimizing data transmission costs, and ensuring synchronization between edge and cloud components. Techniques such as federated learning, hierarchical model deployment, and adaptive data compression play pivotal roles in addressing these challenges. Emerging frameworks like Open Horizon and AWS IoT Greengrass provide robust support for orchestrating edge-cloud AI systems, enabling dynamic coordination across heterogeneous infrastructure layers.

3.4 Future Directions

The evolution of infrastructure for compound AI systems will be driven by the increasing complexity and diversity of AI applications. Future research and development efforts are likely to focus on:

- **Resource-Adaptive Orchestration:** Developing intelligent orchestration frameworks that dynamically allocate tasks across on-device, edge, and cloud resources based on real-time performance metrics, energy efficiency, and cost constraints.

- **Collaborative AI Workflows:** Enhancing support for collaborative AI workflows in distributed environments, enabling seamless integration of decentralized components in a privacy-preserving and secure manner.

- **Quantum and Neuromorphic Computing:** Exploring the integration of emerging computing paradigms, such as quantum computing and neuromorphic processors, to address the computational demands of future compound AI systems.

- **Autonomous Infrastructure Management:** Leveraging AI to automate infrastructure management tasks, including fault diagnosis, self-healing, and predictive scaling, for improved reliability and efficiency.

- **Standardization and Interoperability:** Establishing standardized protocols and APIs to facilitate interoperability between diverse infrastructure components, fostering modular and reusable system designs.

These advancements will be crucial for ensuring the scalability, robustness, and cost-efficiency of compound AI systems in an increasingly interconnected world.

4 System Perspectives of Scalable Compound AI Systems

4.1 Performance Dimensions of Scalable Inference

The design of scalable compound AI systems requires meticulous consideration of system-level metrics that directly impact their performance in real-world deployments, which needs further optimization instead of a single LLM serving system like ScaleLLM (Yao et al., 2024a). Key dimensions include throughput, latency, memory efficiency, and cost-effectiveness, each of which must be optimized to ensure seamless operation under varying workloads and constraints.

High Throughput. Scalable systems must efficiently handle a high volume of concurrent inference requests. This necessitates the optimization of computational pipelines, effective load balancing across distributed nodes, and parallel execution strategies. Techniques such as asynchronous processing and dynamic batching can further enhance throughput by maximizing resource utilization and minimizing idle time (Chen et al., 2022).

Low Latency. Ensuring low end-to-end response time is critical for user-facing applications and time-sensitive enterprise workflows. Achieving this requires the reduction of communication overhead, the use of hardware accelerators (e.g., GPUs, TPUs), and optimizations such as model quantization and pruning. Additionally, pre-fetching mechanisms and low-latency data retrieval techniques, including in-memory databases and efficient caching layers, play pivotal roles in minimizing delays (Konakanchi, 2024).

Memory Efficiency. Compound AI systems involve multiple interacting components, including large-scale language models, retrievers, and external tools. Memory efficiency can be achieved through techniques such as weight sharing, model distillation, and selective model activation. Furthermore, leveraging sparse representations and gradient checkpointing can significantly reduce memory overhead, especially in resource-constrained environments (Kim et al., 2024b).

Cost Effectiveness. Cost considerations necessitate judicious allocation of computational resources while maintaining system performance. Approaches such as multi-tier inference, where simpler models handle less complex queries, and on-demand scaling using serverless architectures can reduce operational costs. Hybrid execution models that utilize edge computing for preliminary processing and cloud resources for intensive tasks can further enhance cost-efficiency (Hill et al., 2024).

4.2 Challenges in Multi-Component Optimization

The integration of diverse components in compound AI systems introduces challenges in joint optimization. Unlike monolithic models, where end-to-end backpropagation is feasible, compound systems require specialized optimization techniques due to the presence of non-differentiable components. Strategies such as evolutionary algorithms, reinforcement learning for pipeline configuration, and gradient approximation methods can be explored to co-optimize these systems. Balancing resource allocation among components—such as deciding the FLOP budget for retrieval versus generation—requires adaptive strategies informed by workload characteristics and performance goals (Dessaux et al., 2024).

4.3 Future Directions

Scalable compound AI systems present a fertile ground for innovation, with several promising avenues for future research:

Adaptive Resource Allocation. Developing frameworks capable of dynamically adjusting resource allocation based on workload patterns and performance metrics can significantly improve efficiency. This includes runtime adaptation strategies that tailor inference pipelines to specific queries.

End-to-End Optimization Frameworks. There is a need for tools and frameworks that enable end-to-end optimization of compound systems. Techniques like differentiable programming extensions and hybrid optimization algorithms could bridge the gap between component-level tuning and system-level performance.

Green AI Considerations. With the increasing deployment of AI systems, environmental impact is a growing concern. Research into energy-efficient model architectures, carbon-aware scheduling, and

eco-friendly hardware design is essential to make compound AI systems sustainable.

Interdisciplinary Approaches. The complexity of compound systems calls for collaborative efforts across domains, including systems engineering, machine learning, and human-computer interaction. Designing user-centric tools for debugging, monitoring, and controlling these systems will be pivotal for their adoption in enterprise settings.

Security and Reliability. As compound AI systems handle critical tasks, ensuring their robustness against adversarial attacks and operational failures is paramount. Future research should focus on enhancing traceability, incorporating fail-safe mechanisms, and designing secure interfaces for system interactions.

By addressing these directions, scalable compound AI systems can achieve higher levels of efficiency, reliability, and adaptability, paving the way for their widespread deployment in diverse applications.

5 Machine Learning Perspectives of Scalable Compound AI Systems

The development of scalable compound AI systems introduces unique machine learning (ML) challenges, particularly in the areas of optimization, resource allocation, and coordination across diverse system components. Unlike traditional monolithic ML models, compound systems rely on multiple interacting agents, including task-specific modules, retrieval engines, and auxiliary components, to collaboratively achieve system-wide objectives. This necessitates novel approaches to optimize these heterogeneous systems under constraints such as latency, cost, and task-specific performance metrics.

5.1 Multi-Agent Optimization and Coordination

In compound AI systems, the problem of coordinating multiple autonomous agents emerges as a critical challenge. Each agent, representing a specialized computational unit (e.g., a language model, a retrieval engine, or a database query module), must operate cohesively to maximize the system's overall utility. To achieve this, compound systems often incorporate a *Manager Agent* or *Controller Agent*, which orchestrates tasks and ensures adherence to system-level constraints. These controllers leverage metadata from registries to assign subtasks dynamically and monitor execution against predefined quality and performance targets (Cao et al., 2012).

The coordination problem is further compounded by the need for real-time decision-making. For instance, task planners generate task execution graphs (DAGs) representing dependencies and sequencing among agents. The dynamic nature of these tasks necessitates adaptive replanning mechanisms, particularly in response to runtime contingencies such as data unavailability or agent failures. Recent advancements in reinforcement learning (RL) and bandit-based optimization techniques offer promising avenues to enable agents to learn optimal task allocations and execution strategies from feedback signals (Wang et al., 2022).

5.2 Judgment Agents and Self-Correcting Pipelines

A defining feature of compound AI systems is their reliance on *Judgment Agents* to evaluate intermediate and final outputs. These agents provide essential feedback loops by applying user-defined or learned quality metrics to assess whether a task's outputs meet the desired criteria. This capability is particularly valuable for error-prone components, such as generative language models, where hallucinations or inaccuracies can significantly impact downstream tasks (Li et al., 2024a).

Judgment Agents are often implemented as discriminative models, ensemble classifiers, or heuristic evaluators that operate alongside generative components. Their outputs inform re-execution decisions, allowing the system to iterate on subtasks until satisfactory results are achieved. This self-correcting behavior draws parallels with human-in-the-loop (HITL) frameworks but is increasingly automated through techniques such as self-consistency decoding and chain-of-thought (CoT) evaluation strategiesc (Kim et al., 2024a).

5.3 End-to-End Optimization in Non-Differentiable Systems

Unlike traditional ML systems, which are optimized end-to-end using differentiable loss functions, compound AI systems involve non-differentiable components such as search engines, symbolic solvers, and external APIs. This necessitates alternative optimization paradigms, such as *pipeline-level tuning* and *stochastic optimization*, to align the performance of individual agents with

system-wide objectives. Frameworks like DSPy introduce novel optimization strategies by leveraging language model capabilities to generate and tune prompts, few-shot examples, and hyperparameter configurations for each component (Cheng et al., 2024a).

End-to-end optimization also involves trade-offs between computational costs and task performance. For example, routing strategies such as those employed in FrugalGPT (Chen et al., 2023a) optimize the allocation of computational resources by dynamically selecting appropriate models or agents based on task complexity and cost constraints. Similarly, multi-agent reinforcement learning (MARL) methods are gaining traction for optimizing collaborative agent behavior in scenarios involving interdependent subtasks (Mondal et al., 2021).

5.4 Scalability and Resource Allocation

Scalable compound AI systems must effectively allocate limited computational resources across agents to meet latency and throughput requirements (Joshi and Kumar, 2021). Techniques such as hierarchical task decomposition and dynamic load balancing are employed to ensure that high-priority tasks are processed efficiently while minimizing bottlenecks (Marzari et al., 2021). Moreover, multi-agent systems increasingly utilize distributed architectures, where task execution is parallelized across clusters of agents to achieve scalability (Olfati-Saber et al., 2007).

Data planners also play a pivotal role in resource allocation by decomposing complex data retrieval operations into sub-tasks and optimizing their execution across available storage and processing units (Cheng et al., 2024b). Metadata-driven approaches, enabled by data registries, allow systems to select optimal data sources and retrieval strategies based on quality, cost, and availability metrics (Peng et al., 2023).

5.5 Future Directions

As compound AI systems continue to evolve, several promising directions for future research and development emerge. These include advancements in multi-agent learning, robust optimization under uncertainty, and the seamless integration of diverse system components.

Adaptive Learning and Transferable Policies. A key challenge in multi-agent optimization is ensuring agents can adapt to changing environments and task requirements. Research into transfer learning and meta-learning techniques may enable agents to generalize learned policies across diverse tasks and environments. This could significantly reduce the time and computational cost of optimizing compound systems for new use cases.

Explainability and Trust. The complexity of compound AI systems poses challenges in ensuring transparency and trustworthiness. Future systems could integrate explainable AI (XAI) methodologies to provide interpretable insights into agent decisions and task execution workflows. Combining XAI techniques with Judgment Agents could also enhance error detection and facilitate user understanding of system behavior.

Federated and Decentralized Architectures. To further improve scalability and privacy, future systems may adopt federated and decentralized architectures, where agents operate autonomously within a distributed network. This approach could mitigate latency issues and reduce reliance on centralized resources while enhancing data security by keeping sensitive data localized.

Enhanced Optimization Frameworks. Advances in optimization frameworks will be critical to managing the growing complexity of compound AI systems. Techniques such as differentiable programming for quasi-differentiable pipelines and hybrid optimization methods that combine gradient-based and heuristic approaches may unlock new levels of performance. Additionally, frameworks like DSPy could be extended to support more sophisticated metrics, including fairness, robustness, and energy efficiency.

Human-AI Collaboration. As these systems become more capable, there is a growing opportunity to redefine how humans and AI collaborate. Future research can explore interactive interfaces and agent architectures that support intuitive user interactions, adaptive feedback loops, and cooperative problem-solving, thus amplifying human creativity and decision-making.

By addressing these areas, compound AI systems can unlock their full potential to deliver reliable, scalable, and context-aware AI applications in enterprise and beyond.

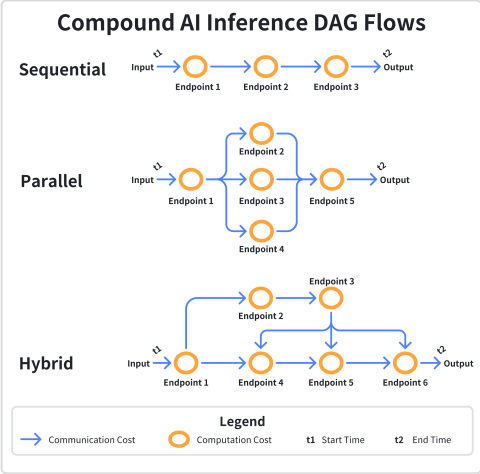

Figure 2: Workflow Architectures of Scalable Compound AI Systems.

6 Workflow Perspective of Scalable Compound AI Systems

From a workflow perspective, scalable Compound AI systems can be categorized into three distinct architectural paradigms: sequential, parallel, and hybrid workflows. Each paradigm provides unique benefits and caters to different application requirements. Below, we discuss these paradigms with illustrative examples and conclude with future directions.

6.1 Sequential Inference

Sequential inference represents workflows where the output of one component serves as the input for the subsequent component in a pipeline. This approach is optimal for tasks requiring structured and stepwise data transformation (Fu et al., 2024).

Examples:

- *Fact-Checking Pipelines*: Systems such as FacTool (Chern et al., 2023) sequentially process inputs by extracting claims, generating search queries, retrieving evidence, and verifying claims (Russo et al., 2024).
- *Retrieval-Augmented Generation (RAG)*: RAG workflows first retrieve relevant documents using retrieval models, which are then passed as context to LLMs for response generation (Pouplin et al., 2024).
- *Code Generation Pipelines (e.g., AlphaCode 2)*: These systems sequentially generate code, execute it for validation, and rank outputs based on predefined metrics like correctness and efficiency (Zhang et al., 2023b).

6.2 Parallel Inference

Parallel inference involves running multiple AI components concurrently to address diverse subtasks or explore complementary approaches. This paradigm is suitable for tasks that benefit from distributed execution or result aggregation (Li et al., 2024c).

Examples:

- *Search Engine Pipelines*: Systems employ parallel retrieval methods (e.g., sparse and dense vector retrieval) to optimize result diversity and relevance (Nie et al., 2024).
- *Self-Consistency in Chain-of-Thought (CoT) Prompting*: Systems like Gemini sample multiple reasoning paths simultaneously and aggregate responses for the most consistent output (Rodríguez et al., 2024).
- *Medical Diagnosis Systems (e.g., MedPrompt (Chen et al., 2023b))*: Parallel prompts are used to generate diverse diagnostic suggestions, which are ensembled to improve reliability and trustworthiness (Zavaleta-Monestel et al., 2024).

6.3 Hybrid Inference

Hybrid inference integrates sequential and parallel execution within the same workflow, combining the strengths of both paradigms for efficiency and robustness. This approach is particularly advantageous for complex tasks with interdependent and independent components (Hao et al., 2024).

Examples:

- *Chatbot Verification Systems*: Inspired by FacTool, claims are extracted sequentially but verified in parallel through independent queries or scoring functions (Vural and Kalaman, 2024; Li et al., 2024b).
- *Multi-Hop Question Answering*: Context for each reasoning hop is retrieved sequentially, while multiple reasoning chains for each hop can be processed in parallel (Cheng et al., 2024c).
- *Tool-Augmented LLMs (e.g., ChatGPT* [1]*)*: These systems execute plugin calls (e.g., web

[1] https://chatgpt.com/

browsing, code execution) in parallel while orchestrating their usage sequentially for logical flow [2].

6.4 Future Directions

The evolution of workflows in scalable Compound AI systems presents numerous opportunities for innovation:

- **Adaptive Workflow Design**: Future systems could dynamically switch between sequential and parallel paradigms based on task complexity and resource availability, optimizing performance and cost.
- **End-to-End Optimization**: Research is needed to develop frameworks that co-optimize components across sequential and parallel workflows, balancing latency, accuracy, and resource constraints.
- **Enhanced Monitoring and Debugging**: Novel MLOps tools tailored for hybrid workflows are crucial to track and visualize intermediate outputs, enabling robust performance tracking and anomaly detection.
- **Data-Driven Workflow Adaptation**: Leveraging feedback loops to refine workflow strategies, particularly for hybrid systems, could enhance adaptability to evolving tasks and datasets.
- **Interdisciplinary Integration**: Collaboration between AI, systems engineering, and HCI researchers could yield innovative orchestration mechanisms that seamlessly integrate existing infrastructure into compound systems.

In conclusion, the diverse architectural paradigms of sequential, parallel, and hybrid workflows provide a versatile foundation for Compound AI systems, addressing the complex requirements of modern applications. Continued advancements in these workflows will shape the next generation of scalable, efficient, and reliable AI systems.

7 Application Perspectives of Scalable Compound AI Inference

In this section, we will explore the diverse applications of compound AI inference across various industries and scenarios. By examining real-world use cases, we aim to illustrate how this approach is transforming fields like healthcare, finance, customer service, and autonomous systems, among others. These examples will demonstrate the flexibility, efficiency, and effectiveness of compound AI systems in meeting the unique demands of each application domain.

[2] https://www.cursor.com/

Type	Workflow Description	Examples
Sequential	Outputs flow linearly between components	Fact-checking (FacTool), RAG pipelines, code generation (AlphaCode 2)
Parallel	Independent components run concurrently	Search engines, self-consistency in CoT, medical diagnosis (MedPrompt)
Hybrid	Combines sequential and parallel workflows	Chatbot verification, multi-hop QA (HotpotQA), tool-augmented LLMs (ChatGPT Plus)

Table 1: Summary of Compound AI System Workflows: Compound AI system workflows are categorized into sequential, parallel, and hybrid designs based on how components interact and process tasks. Sequential workflows rely on a linear progression of outputs, parallel workflows enable simultaneous processing across independent components, and hybrid workflows combine the strengths of both by strategically employing sequential and parallel operations.

7.1 Healthcare

- **Medical Diagnostics and Question Answering:** Compound AI systems are revolutionizing medical diagnostics by combining generative language models (LLMs) with retrieval systems and domain-specific databases. For example, tools like MedPrompt(Chen et al., 2023b) integrate LLMs with nearest-neighbor searches to retrieve relevant medical records, combining them with chain-of-thought reasoning for accurate diagnosis or treatment rec-

ommendations. These systems surpass standalone models by ensuring that recommendations are grounded in verified clinical guidelines and patient-specific information.

- **Drug Discovery:** AI systems like AlphaFold(Abramson et al., 2024) leverage compound inference by integrating symbolic solvers with machine learning models for protein structure prediction. By combining different tools, researchers achieve higher efficiency in simulating and predicting molecular interactions, accelerating the drug discovery process.

- **Patient Monitoring:** In remote patient monitoring, compound AI systems fuse sensor data analysis with LLMs for natural language summarization of critical trends. This enables healthcare providers to make timely, data-driven decisions.(Zavaleta-Monestel et al., 2024)

7.2 Legal and Regulatory Compliance

- **Document Summarization and Legal Research:** Compound AI inference like Athena(Xiao and Chen, 2024) enables the processing and summarization of lengthy legal documents by combining LLMs with retrieval-augmented generation (RAG) techniques. These systems retrieve and verify relevant case laws and statutes, providing accurate summaries or tailored recommendations.(Guha et al., 2023)

- **Regulatory Adherence:** Enterprises use compound systems to analyze compliance documents and regulations dynamically(Hassani, 2024; Hassani et al., 2024). For instance, integrating LLMs with rule-based engines ensures adherence to evolving legal standards while enabling efficient cross-referencing.

7.3 Enterprise and Customer Service

- **Conversational AI and Chatbots:** Modern chatbots, like those in ChatGPT Plus, integrate LLMs with external tools such as web browsing[3] and code execution[4] to handle dynamic queries. Compound AI inference enables these bots to provide accurate, up-to-date, and actionable responses while maintaining conversational coherence.

- **Customer Insights and Feedback Analysis:** AI systems for customer feedback leverage retrieval mechanisms alongside generative models to identify patterns in customer sentiment. By processing vast datasets and correlating findings with user-specific data, these systems enhance decision-making and product development.(Zhang et al., 2023a)

7.4 Finance

- **Fraud Detection:** Compound AI systems in finance integrate predictive models with rule-based systems and anomaly detection engines to identify fraudulent transactions(Hu et al., 2024a; Dubois et al., 2024). This approach allows for real-time insights while reducing false positives.

- **Portfolio Optimization:** By combining LLMs for market trend analysis with symbolic optimizers for asset allocation, these systems enable dynamic portfolio adjustments based on both historical data and real-time inputs.(Wang et al., 2024)

7.5 Education and Knowledge Management

- **Adaptive Learning Platforms:** Compound AI inference powers personalized education by integrating LLMs for generating custom content with retrieval engines to source supplementary learning materials. These systems cater to individual learning styles and improve engagement.

- **Research Assistance:** Researchers benefit from AI systems that synthesize literature reviews by combining generative AI capabilities with retrieval from academic databases.[5] Tools like AlphaGeometry(Trinh et al., 2024) enhance problem-solving by integrating symbolic reasoning engines with generative models.

7.6 Technology and Software Development

- **Code Generation and Debugging:** Systems like AlphaCode(Siam et al., 2024) combine

[3] https://platform.openai.com/docs/guides/tools-web-search?api-mode=chat

[4] https://microsoft.github.io/autogen/stable/user-guide/core-user-guide/design-patterns/code-execution-groupchat.html

[5] https://github.com/binary-husky/gpt_academic

LLMs with code execution modules to generate and test millions of programming solutions. This approach ensures reliability and efficiency in software development.

- **AI-Powered DevOps:** In software operations, compound systems enhance task automation by integrating LLMs with monitoring tools and rule-based systems. This enables faster troubleshooting and continuous deployment optimization.

7.7 Creative Industries

- **Content Creation and Editing:** Tools like DALL-E[6] and GPT-4 integrate generative AI with user-defined style guides or editing tools, enabling seamless creation and refinement of written, visual, and multimedia content.

- **Interactive Storytelling:** Compound AI inference powers immersive storytelling experiences by combining LLMs with contextual retrieval and dialogue engines, ensuring coherent and dynamic narratives.(Kumaran et al., 2024)

7.8 Autonomous Systems and Robotics

- **Self-Driving Cars:** Compound systems in autonomous vehicles integrate perception modules (e.g., lidar and radar processing) with decision-making engines and generative reasoning. These systems work collaboratively to ensure safety and adaptability to real-world scenarios.

- **Smart Manufacturing:** Industrial robots employ compound AI systems to integrate LLM-driven task planning with sensor-based feedback loops, enabling precision and adaptability in manufacturing processes.(Lee and Su, 2023)

7.9 Search and Retrieval

- **Enhanced Search Engines:** Tools like Bing[7] and RAG-based systems enhance search capabilities by integrating LLMs with retrieval systems.(Gao et al., 2023) These systems provide contextual and accurate results for queries by dynamically generating search terms and refining outputs.

[6] https://openai.com/index/dall-e-3/
[7] https://www.bing.com/

- **Knowledge Graph Construction:** Compound AI inference helps construct and maintain knowledge graphs by combining natural language understanding with data extraction tools, ensuring updated and verified information.(Zhang and Soh, 2024)

8 Compound AI Training: Challenges and Future Directions

As compound AI systems continue to evolve, there are several promising avenues for future research and development. These directions aim to address key challenges in design, optimization, and operation while unlocking new capabilities to better integrate AI into diverse applications. Below, we outline some critical areas for exploration, including the training of compound AI systems, which is emerging as a cornerstone for their advancement.

8.1 Enhanced Training Techniques for Compound AI Systems

Training compound AI systems is inherently challenging due to their non-differentiable components and the interplay between diverse modules (Yao et al., 2024b). Unlike single-model systems, where end-to-end differentiability enables straightforward optimization, compound systems require innovative training paradigms. Future work could explore:

- **Differentiable Surrogates for Non-Differentiable Components:** Developing surrogate models or gradient approximations for non-differentiable components, such as search engines or external tools, to enable gradient-based optimization across the pipeline.

- **Co-Training and Fine-Tuning Across Modules:** Investigating techniques to jointly fine-tune language models and auxiliary tools (e.g., retrievers, calculators, and databases) to maximize their synergy within the compound system.

- **Reinforcement Learning for Dynamic Pipelines:** Applying reinforcement learning (RL) to optimize the selection and configuration of modules dynamically based on input data and target objectives.

- **Human-in-the-Loop Optimization:** Leveraging human feedback to fine-tune individual components or entire systems, particularly for

tasks where subjective quality metrics like interpretability and user satisfaction are critical.

- **Co-Optimization in Multi-Agent Systems:** Investigating strategies for optimizing multiple models or agents simultaneously in a coordinated manner. This includes techniques for joint fine-tuning where the parameters of various components—such as databases, retrieval systems, and computational models—are updated concurrently to improve overall system performance. Challenges like inter-agent communication, avoiding gradient conflicts, and ensuring stability during training need to be addressed. Future research could also explore decentralized optimization techniques, where each agent learns independently while sharing key updates with the system, akin to federated learning paradigms.

8.2 Scaling and Resource Optimization

The increasing complexity of compound AI systems requires novel strategies to manage computational resources while maintaining performance. Future work could focus on:

- **Dynamic Resource Allocation:** Developing adaptive strategies to allocate resources to different components based on real-time workload and target latencies, optimizing system efficiency under cost and performance constraints.
- **Fine-Grained Scheduling:** Enhancing pipeline-aware scheduling to minimize idle time across distributed components, leveraging techniques such as task preemption and predictive load balancing.
- **Low-Cost Deployment Strategies:** Extending frameworks like FrugalGPT to handle compound systems, enabling cost-efficient execution while preserving accuracy across multiple interconnected components.

8.3 Architectural Innovations

As the design space of compound systems expands, research could delve deeper into understanding and exploring optimal architectures:

- **Modular and Flexible Frameworks:** Building systems that are highly modular and easily reconfigurable, allowing developers to experiment with new combinations of tools and models with minimal overhead.
- **Adaptive Systems:** Designing architectures that adapt dynamically to different inputs or user contexts, reconfiguring module choices and interaction strategies on-the-fly.
- **Diverse Interaction Patterns:** Investigating strategies for optimizing interactions between modules, such as feedback loops where outputs from one module refine inputs to another.

8.4 Robustness, Security, and Explainability

With the complexity of compound AI systems comes an increased need for robustness, security, and interpretability:

- **Error Detection and Recovery:** Building mechanisms to identify and mitigate errors within individual modules or their interactions, ensuring reliable end-to-end performance.
- **Security Protocols:** Developing tools to secure compound systems against emerging threats, such as adversarial attacks that exploit vulnerabilities in tool-LLM interactions.
- **Explainable Compound AI Systems:** Creating techniques to provide interpretable explanations of how decisions are made across the pipeline, enhancing trust and accountability in critical applications.

8.5 Scalable MLOps and Monitoring

As compound AI systems become more prevalent, operational challenges will grow, necessitating advanced MLOps solutions:

- **Holistic Monitoring:** Designing systems that provide end-to-end traceability, enabling developers to analyze intermediate outputs and debug complex workflows effectively.
- **DataOps Integration:** Extending monitoring solutions to ensure high-quality data inputs for all components, aligning data pipelines with the behavior of downstream AI systems.
- **Feedback-Driven Improvement:** Leveraging operational insights, such as failure modes or inefficiencies, to refine system components or retrain modules dynamically.

8.6 Human-Centric Design and Applications

Finally, future work should focus on improving the usability and societal impact of compound AI systems:

- **User-Friendly Design Frameworks:** Simplifying the process of designing, deploying, and maintaining compound AI systems for developers, researchers, and domain experts.
- **Collaborative AI Systems:** Enabling systems that work alongside humans in a collaborative manner, adapting to user inputs and preferences in real time.
- **Ethical and Fair AI Pipelines:** Ensuring that compound AI systems prioritize fairness and minimize biases, especially in applications that impact diverse populations.

By addressing these future directions, the field of compound AI systems can achieve significant advancements, enabling more powerful, efficient, and trustworthy AI solutions. Training compound AI systems, in particular, holds the potential to redefine the boundaries of what such systems can achieve, bridging the gap between modular innovation and seamless integration.

9 Related Work

The development of Compound AI systems has been shaped by advancements in modular architectures, task-specific optimizations, and safety-focused innovations, paving the way for adaptable and scalable AI applications.

9.1 Modular Frameworks and Architectures

Frameworks such as *LangChain*[8], *AutoGPT*[9] , and *DSPy*(Khattab et al., 2023) have been instrumental in streamlining the design and integration of modular AI systems. These tools enable automated tuning, resource allocation, and workflow orchestration, ensuring efficient inter-component collaboration. Systems like *Databricks Mosaic AI* [10] demonstrate how integrating foundational models with tools like vector databases can address complex real-world applications, showcasing the scalability and versatility of compound architectures.

9.2 Task-Specific and Retrieval-Driven Systems

Compound AI systems excel in task-specific optimizations by integrating specialized tools with large language models (LLMs). Examples include *AlphaCode 2(Siam et al., 2024)*, which leverages clustering and scoring for superior programming solutions, and *MedPrompt (Nori et al., 2023)*, which combines retrieval-augmented generation with chain-of-thought reasoning for medical queries. Retrieval-based approaches, such as CoK (Li et al., 2024d), enhance contextual understanding and accuracy, highlighting the value of integrating external data sources to surpass standalone model limitations.

9.3 Safety and Reliability

Ensuring safety and reliability is critical for deploying AI in sensitive domains. Systems like *TorchOpera (Han et al., 2024a)* integrate safety detection, contextual grounding, and error repair modules to mitigate risks such as hallucinations and sensitive content leakage. Verifier-based architectures introduced by (Davis et al., 2024a) improve accuracy in reasoning tasks by separating generation and verification processes, demonstrating the importance of trustworthiness in Compound AI systems.

These advancements underline the potential of Compound AI to redefine AI capabilities by integrating modularity, task-specific tools, and safety protocols into cohesive systems.

10 Conclusion

The advancements and challenges presented in this paper emphasize the transformative potential of compound AI systems as a frontier in artificial intelligence. By integrating diverse AI components into cohesive architectures, compound AI systems offer unparalleled scalability, flexibility, and reliability across a wide range of applications. This modular approach addresses the limitations of monolithic AI models, enabling dynamic adaptability, enhanced performance, and efficient resource utilization.

Our exploration highlights the pivotal role of inferencing in compound AI systems, where the orchestration of heterogeneous components facilitates real-time decision-making, iterative refinement, and robust quality assurance. The layered architecture of these systems—spanning applications, workflows, machine learning models, system-level optimizations, and infrastructure—provides a structured framework for managing complexity and achieving end-to-end optimization.

Despite their potential, compound AI systems face significant challenges, including the need for

[8] https://www.langchain.com/

[9] https://github.com/Significant-Gravitas/AutoGPT

[10] https://www.databricks.com/product/machine-learning

novel optimization techniques, adaptive resource management, and enhanced interoperability. The lack of standardized methodologies for addressing trade-offs in latency, accuracy, and coherence underscores the necessity for continued research and innovation. Furthermore, ensuring the robustness, transparency, and ethical deployment of these systems remains critical, particularly as they are increasingly adopted in high-stakes domains like healthcare, finance, and autonomous systems.

Looking ahead, future research must focus on developing scalable training paradigms for compound AI systems, advancing multi-agent coordination, and exploring adaptive and explainable workflows. The integration of emerging technologies such as quantum computing and decentralized architectures could further enhance the capabilities of compound AI systems. Finally, fostering interdisciplinary collaboration and user-centric design will be essential to unlocking their full potential and driving innovation across industries.

In conclusion, compound AI systems represent a paradigm shift in artificial intelligence, offering a pathway to more adaptable, high-performance, and trustworthy AI applications. By addressing the open problems and leveraging the advances outlined in this paper, the research community can pave the way for the next generation of AI systems, capable of meeting the complex demands of modern applications and transforming diverse fields through intelligent, scalable solutions.

Limitations

ACL 2023 requires all submissions to have a section titled "Limitations", for discussing the limitations of the paper as a complement to the discussion of strengths in the main text. This section should occur after the conclusion, but before the references. It will not count towards the page limit. The discussion of limitations is mandatory. Papers without a limitation section will be desk-rejected without review.

While we are open to different types of limitations, just mentioning that a set of results have been shown for English only probably does not reflect what we expect. Mentioning that the method works mostly for languages with limited morphology, like English, is a much better alternative. In addition, limitations such as low scalability to long text, the requirement of large GPU resources, or other things that inspire crucial further investigation are welcome.

References

Josh Abramson, Jonas Adler, Jack Dunger, Richard Evans, Tim Green, Alexander Pritzel, Olaf Ronneberger, Lindsay Willmore, Andrew J. Ballard, Joshua Bambrick, Sebastian W. Bodenstein, David A. Evans, Chia-Chun Hung, Michael O'Neill, David Reiman, Kathryn Tunyasuvunakool, Zachary Wu, Akvilė Žemgulytė, Eirini Arvaniti, Charles Beattie, Ottavia Bertolli, Alex Bridgland, Alexey Cherepanov, Miles Congreve, Alexander I. Cowen-Rivers, Andrew Cowie, Michael Figurnov, Fabian B. Fuchs, Hannah Gladman, Rishub Jain, Yousuf A. Khan, Caroline M. R. Low, Kuba Perlin, Anna Potapenko, Pascal Savy, Sukhdeep Singh, Adrian Stecula, Ashok Thillaisundaram, Catherine Tong, Sergei Yakneen, Ellen D. Zhong, Michal Zielinski, Augustin Žídek, Victor Bapst, Pushmeet Kohli, Max Jaderberg, Demis Hassabis, and John M. Jumper. 2024. Accurate structure prediction of biomolecular interactions with alphafold 3. *Nature*, 630(8016):493—-500.

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*.

Rohan Anil, Andrew M Dai, Orhan Firat, Melvin Johnson, Dmitry Lepikhin, Alexandre Passos, Siamak Shakeri, Emanuel Taropa, Paige Bailey, Zhifeng Chen, et al. 2023. Palm 2 technical report. *arXiv preprint arXiv:2305.10403*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Yongcan Cao, Wenwu Yu, Wei Ren, and Guanrong Chen. 2012. An overview of recent progress in the study of distributed multi-agent coordination. *IEEE Transactions on Industrial Informatics*, 9:427–438.

Baiqi Chen, Jianpei Dong, Marina Ruelas, Xiangyi Ye, Jinxu He, Ruijie Yao, Yuqiu Fu, Ying Liu, Jingpeng Hu, Tianyu Wu, et al. 2022. Artificial intelligence-assisted high-throughput screening of printing conditions of hydrogel architectures for accelerated diabetic wound healing. *Advanced Functional Materials*, 32(38):2201843.

Lingjiao Chen, Matei A. Zaharia, and James Y. Zou. 2023a. Frugalgpt: How to use large language models while reducing cost and improving performance. *ArXiv*, abs/2305.05176.

Xuhang Chen, Chi-Man Pun, and Shuqiang Wang. 2023b. Medprompt: Cross-modal prompting for multi-task medical image translation. *ArXiv*, abs/2310.02663.

Ching-An Cheng, Allen Nie, and Adith Swaminathan. 2024a. Trace is the next autodiff: Generative optimization with rich feedback, execution traces, and llms. In *Neural Information Processing Systems*.

Jie Cheng, Yingbing Chen, Xiaodong Mei, Bowen Yang, Bo Li, and Ming Liu. 2024b. Rethinking imitation-based planners for autonomous driving. *2024 IEEE International Conference on Robotics and Automation (ICRA)*, pages 14123–14130.

Keyuan Cheng, Gang Lin, Haoyang Fei, Yuxuan Zhai, Lu Yu, Muhammad Asif Ali, Lijie Hu, and Di Wang. 2024c. Multi-hop question answering under temporal knowledge editing. *ArXiv*, abs/2404.00492.

Ethan Chern, Steffi Chern, Shiqi Chen, Weizhe Yuan, Kehua Feng, Chunting Zhou, Junxian He, Graham Neubig, and Pengfei Liu. 2023. Factool: Factuality detection in generative ai - a tool augmented framework for multi-task and multi-domain scenarios. *ArXiv*, abs/2307.13528.

Florin Cuconasu, Giovanni Trappolini, F. Siciliano, Simone Filice, Cesare Campagnano, Yoelle Maarek, Nicola Tonellotto, and Fabrizio Silvestri. 2024. The power of noise: Redefining retrieval for rag systems. *ArXiv*, abs/2401.14887.

Jared Quincy Davis, Boris Hanin, Lingjiao Chen, Peter Bailis, Ion Stoica, and Matei Zaharia. 2024a. Networks of networks: Complexity class principles applied to compound ai systems design. *arXiv preprint arXiv:2407.16831*.

Jared Quincy Davis, Boris Hanin, Lingjiao Chen, Peter D. Bailis, Ion Stoica, and Matei Zaharia. 2024b. Networks of networks: Complexity class principles applied to compound ai systems design. *ArXiv*, abs/2407.16831.

Delphine Dessaux, Samuel Buchet, Lucie Barthe, Marianne Defresne, Gianluca Cioci, Simon De Givry, Luis F Garcia-Alles, Thomas Schiex, and Sophie Barbe. 2024. Designing symmetrical multicomponent proteins using a hybrid generative ai approach. *bioRxiv*, pages 2024–06.

Bosheng Ding, Chengwei Qin, Linlin Liu, Yew Ken Chia, Boyang Li, Shafiq Joty, and Lidong Bing. 2023. Is GPT-3 a good data annotator? In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11173–11195, Toronto, Canada. Association for Computational Linguistics.

Bosheng Ding, Chengwei Qin, Ruochen Zhao, Tianze Luo, Xinze Li, Guizhen Chen, Wenhan Xia, Junjie Hu, Anh Tuan Luu, and Shafiq Joty. 2024. Data augmentation using LLMs: Data perspectives, learning paradigms and challenges. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1679–1705, Bangkok, Thailand. Association for Computational Linguistics.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475*.

Upol Ehsan, Qingzi Vera Liao, Michael J. Muller, Mark O. Riedl, and Justin D. Weisz. 2021. Expanding explainability: Towards social transparency in ai systems. *Proceedings of the 2021 CHI Conference on Human Factors in Computing Systems*.

Johannes Erbel and Jens Grabowski. 2023. Scientific workflow execution in the cloud using a dynamic runtime model. *Softw. Syst. Model.*, 23:163–193.

Yanlin Feng, Sajjadur Rahman, Aaron Feng, Vincent Chen, and Eser Kandogan. 2024. Cmdbench: A benchmark for coarse-to-fine multimodal data discovery in compound ai systems. *Proceedings of the Conference on Governance, Understanding and Integration of Data for Effective and Responsible AI*.

Yichao Fu, Peter D. Bailis, Ion Stoica, and Hao Zhang. 2024. Break the sequential dependency of llm inference using lookahead decoding. *ArXiv*, abs/2402.02057.

Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Haofen Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997*, 2.

Neel Guha, Julian Nyarko, Daniel E. Ho, Christopher Ré, Adam Chilton, Aditya Narayana, Alex Chohlas-Wood, Austin Peters, Brandon Waldon, Daniel N. Rockmore, Diego Zambrano, Dmitry Talisman, Enam Hoque, Faiz Surani, Frank Fagan, Galit Sarfaty, Gregory M. Dickinson, Haggai Porat, Jason Hegland, Jessica Wu, Joe Nudell, Joel Niklaus, John Nay, Jonathan H. Choi, Kevin Tobia, Margaret Hagan, Megan Ma, Michael Livermore, Nikon Rasumov-Rahe, Nils Holzenberger, Noam Kolt, Peter Henderson, Sean Rehaag, Sharad Goel, Shang Gao, Spencer Williams, Sunny Gandhi, Tom Zur, Varun Iyer, and Zehua Li. 2023. Legalbench: A collaboratively built benchmark for measuring legal reasoning in large language models.

MyungJoo Ham, Jijoong Moon, Geunsik Lim, Jaeyun Jung, Hyoungjoo Ahn, Wook Song, Sangjung Woo, Parichay Kapoor, Dongju Chae, Gichan Jang, Yong Min Ahn, and Jihoon Lee. 2021. Nnstreamer: Efficient and agile development of on-device ai systems. *2021 IEEE/ACM 43rd International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, pages 198–207.

Shanshan Han, Zijian Hu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Dimitris Stripelis, Zhaozhuo Xu, and Chaoyang He. 2024a. Torchopera: A compound ai system for llm safety. *arXiv preprint arXiv:2406.10847*.

Shanshan Han, Yuhang Yao, Zijian Hu, Dimitris Stripelis, Zhaozhuo Xu, and Chaoyang He. 2024b. Torchopera: A compound ai system for llm safety. *ArXiv*, abs/2406.10847.

Zixu Hao, Huiqiang Jiang, Shiqi Jiang, Ju Ren, and Ting Cao. 2024. Hybrid slm and llm for edge-cloud collaborative inference. *Proceedings of the Workshop on Edge and Mobile Foundation Models*.

Shabnam Hassani. 2024. Enhancing legal compliance and regulation analysis with large language models. *2024 IEEE 32nd International Requirements Engineering Conference (RE)*.

Shabnam Hassani, Mehrdad Sabetzadeh, Daniel Amyot, and Jain Liao. 2024. Rethinking legal compliance automation: Opportunities with large language models. *2024 IEEE 32nd International Requirements Engineering Conference (RE)*.

Harry Hill, Cristina Roadevin, Stephen Duffy, Olena Mandrik, and Adam Brentnall. 2024. Cost-effectiveness of ai for risk-stratified breast cancer screening. *JAMA Network Open*, 7(9):e2431715–e2431715.

Sihao Hu, Tiansheng Huang, Ka-Ho Chow, Wenqi Wei, Yanzhao Wu, and Ling Liu. 2024a. Zipzap: Efficient training of language models for large-scale fraud detection on blockchain. In *Proceedings of the ACM Web Conference 2024*, WWW '24, page 2807–2816, New York, NY, USA. Association for Computing Machinery.

Zijian Hu, Jipeng Zhang, Rui Pan, Zhaozhuo Xu, Shanshan Han, Han Jin, Alay Dilipbhai Shah, Dimitris Stripelis, Yuhang Yao, Salman Avestimehr, Chaoyang He, and Tong Zhang. 2024b. Fox-1 technical report. *arXiv preprint arXiv:2411.05281*.

Swayambhoo Jain, Ravi Raju, Bo Li, Zoltan Csaki, Jonathan Li, Kaizhao Liang, Guoyao Feng, Urmish Thakkar, Anand Sampat, Raghu Prabhakar, and Sumati Jairath. 2024. Composition of experts: A modular compound ai system leveraging large language models. *ArXiv*, abs/2412.01868.

Rajendra P. Joshi and Neeraj Kumar. 2021. Artificial intelligence for autonomous molecular design: A perspective. *Molecules*, 26.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. 2020. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*.

Omar Khattab, Arnav Singhvi, Paridhi Maheshwari, Zhiyuan Zhang, Keshav Santhanam, Sri Vardhamanan, Saiful Haq, Ashutosh Sharma, Thomas T Joshi, Hanna Moazam, et al. 2023. Dspy: Compiling declarative language model calls into self-improving pipelines. *arXiv preprint arXiv:2310.03714*.

Dongyoung Kim, Kimin Lee, Jinwoo Shin, and Jaehyung Kim. 2024a. Spread preference annotation: Direct preference judgment for efficient llm alignment.

Ji Yeon Kim, Seong Hyeon Jo, Ha Sang-hyun, Ki Hwan Kim, Young Jin Kang, and Seok Chan Jeong. 2024b. Comparison of ai model serving efficiency: Response time and memory usage analysis. *AHFE International*.

Sandeep Konakanchi. 2024. Next-generation low-latency architectures for real-time ai-driven cloud services. *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*.

Vikram Kumaran, Jonathan Rowe, and James Lester. 2024. Narrativegenie: generating narrative beats and dynamic storytelling with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence and Interactive Digital Entertainment*, volume 20, pages 76–86.

Jay Lee and Hanqi Su. 2023. A unified industrial large knowledge model framework in industry 4.0 and smart manufacturing. *arXiv preprint arXiv:2312.14428*.

Patrick Lewis, Ethan Perez, Aleksandara Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Kuttler, Mike Lewis, Wen tau Yih, Tim Rocktäschel, Sebastian Riedel, and Douwe Kiela. 2020a. Retrieval-augmented generation for knowledge-intensive nlp tasks. *ArXiv*, abs/2005.11401.

Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020b. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems*, 33:9459–9474.

Dawei Li, Bohan Jiang, Liangjie Huang, Alimohammad Beigi, Chengshuai Zhao, Zhen Tan, Amrita Bhattacharjee, Yuxuan Jiang, Canyu Chen, Tianhao Wu, Kai Shu, Lu Cheng, and Huan Liu. 2024a. From generation to judgment: Opportunities and challenges of llm-as-a-judge. *ArXiv*, abs/2411.16594.

Dongfang Li, Xinshuo Hu, Zetian Sun, Baotian Hu, Shaolin Ye, Zifei Shan, Qian Chen, and Min Zhang. 2024b. Truthreader: Towards trustworthy document assistant chatbot with reliable attribution. *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*.

Muyang Li, Tianle Cai, Jiaxin Cao, Qinsheng Zhang, Han Cai, Junjie Bai, Yangqing Jia, Ming-Yu Liu, Kai Li, and Song Han. 2024c. Distrifusion: Distributed parallel inference for high-resolution diffusion models. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 7183–7193.

Xingxuan Li, Ruochen Zhao, Yew Ken Chia, Bosheng Ding, Shafiq Joty, Soujanya Poria, and Lidong Bing. 2024d. Chain-of-knowledge: Grounding large language models via dynamic knowledge adapting over heterogeneous sources. In *The Twelfth International Conference on Learning Representations*.

Jiaxin Liu, Wenhui Zhou, Hong Wang, Zhong Cao, Wen-Hui Yu, Cheng-Yu Zhao, Ding Zhao, Diange Yang, and Jun Li. 2022. Road traffic law adaptive decision-making for self-driving vehicles. *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, pages 2034–2041.

Brady D. Lund and Wang Ting. 2023. Chatting about chatgpt: How may ai and gpt impact academia and libraries? *SSRN Electronic Journal*.

Luca Marzari, Ameya Pore, Diego Dall'Alba, Gerardo Aragon-Camarasa, Alessandro Farinelli, and Paolo Fiorini. 2021. Towards hierarchical task decomposition using deep reinforcement learning for pick and place subtasks. *2021 20th International Conference on Advanced Robotics (ICAR)*, pages 640–645.

Xianghui Meng. 2024. Optimization of algorithmic efficiency in ai: Addressing computational complexity and scalability challenges. *Applied and Computational Engineering*.

Washim Uddin Mondal, Mridul Agarwal, Vaneet Aggarwal, and Satish V. Ukkusuri. 2021. On the approximation of cooperative heterogeneous multi-agent reinforcement learning (marl) using mean field control (mfc). *ArXiv*, abs/2109.04024.

Guangtao Nie, Rong Zhi, Xiaofan Yan, Yufan Du, Xiangyang Zhang, Jianwei Chen, Mi Zhou, Hongshen Chen, Tianhao Li, Ziguang Cheng, Sulong Xu, and Jinghe Hu. 2024. A hybrid multi-agent conversational recommender system with llm and search engine in e-commerce. In *ACM Conference on Recommender Systems*.

Harsha Nori, Yin Tat Lee, Sheng Zhang, Dean Carignan, Richard Edgar, Nicolo Fusi, Nicholas King, Jonathan Larson, Yuanzhi Li, Weishung Liu, et al. 2023. Can generalist foundation models outcompete special-purpose tuning? case study in medicine. *arXiv preprint arXiv:2311.16452*.

Olanrewaju M. Okuyelu and Ojima Ojodumi Adaji. 2024. Ai-driven real-time quality monitoring and process optimization for enhanced manufacturing performance. *Journal of Advances in Mathematics and Computer Science*.

Omobolaji Olufunmilayo Olateju, Samuel Ufom Okon, Oluwaseun Oladeji Olaniyi, Amaka Debie Samuel-Okon, and Christopher Uzoma Asonze. 2024. Exploring the concept of explainable ai and developing information governance standards for enhancing trust and transparency in handling customer data. *Journal of Engineering Research and Reports*.

Reza Olfati-Saber, J. Alex Fax, and Richard M. Murray. 2007. Consensus and cooperation in networked multi-agent systems. *Proceedings of the IEEE*, 95:215–233.

Yuan Peng, Franziska Bathelt, Richard Gebler, Robert Gött, Andreas Heidenreich, Elisa Henke, Dennis Kadioglu, Stephan Lorenz, Abishaa Vengadeswaran, and Martin Sedlmayr. 2023. Use of metadata-driven approaches for data harmonization in the medical domain: Scoping review. *JMIR Medical Informatics*, 12.

Thomas Pouplin, Hao Sun, Samuel Holt, and Mihaela van der Schaar. 2024. Retrieval-augmented thought process as sequential decision making. *ArXiv*, abs/2402.07812.

Yujia Qin, Shi Liang, Yining Ye, Kunlun Zhu, Lan Yan, Ya-Ting Lu, Yankai Lin, Xin Cong, Xiangru Tang, Bill Qian, Sihan Zhao, Runchu Tian, Ruobing Xie, Jie Zhou, Marc H. Gerstein, Dahai Li, Zhiyuan Liu, and Maosong Sun. 2023. Toolllm: Facilitating large language models to master 16000+ real-world apis. *ArXiv*, abs/2307.16789.

María Estrella Vallecillo Rodríguez, Arturo Montejo Ráez, and María Teresa Martín-Valdivia. 2024. Sinai at pan 2024 textdetox: Application of chain of thought with self-consistency strategy in large language models for multilingual text detoxification. In *Conference and Labs of the Evaluation Forum*.

Daniel Russo, Stefano Menini, Jacopo Staiano, and Marco Guerini. 2024. Face the facts! evaluating rag-based fact-checking pipelines in realistic settings. *ArXiv*, abs/2412.15189.

Keshav Santhanam, Deepti Raghavan, Muhammad Shahir Rahman, Thejas Venkatesh, Neha Kunjal, Pratiksha Thaker, Philip Levis, and Matei Zaharia. 2024. Alto: An efficient network orchestrator for compound ai systems. *Proceedings of the 4th Workshop on Machine Learning and Systems*.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, Maria Lomeli, Eric Hambro, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2024. Toolformer: Language models can teach themselves to use tools. *Advances in Neural Information Processing Systems*, 36.

Weizhou Shen, Chenliang Li, Hongzhan Chen, Ming Yan, Xiaojun Quan, Hehong Chen, Ji Zhang, and Fei Huang. 2024. Small llms are weak tool learners: A multi-llm agent. *ArXiv*, abs/2401.07324.

Md Kamrul Siam, Huanying Gu, and Jerry Q Cheng. 2024. Programming with ai: Evaluating chatgpt, gemini, alphacode, and github copilot for programmers. *arXiv preprint arXiv:2411.09224*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *ArXiv*, abs/2408.03314.

Zayne Sprague, Fangcong Yin, Juan Diego Rodriguez, Dongwei Jiang, Manya Wadhwa, Prasann Singhal, Xinyu Zhao, Xi Ye, Kyle Mahowald, and Greg Durrett. 2024. To cot or not to cot? chain-of-thought helps mainly on math and symbolic reasoning. *ArXiv*, abs/2409.12183.

Dimitris Stripelis, Zijian Hu, Jipeng Zhang, Zhaozhuo Xu, Alay Dilipbhai Shah, Han Jin, Yuhang Yao, Salman Avestimehr, and Chaoyang He. 2024. Tensoropera router: A multi-model router for efficient llm inference. *arXiv preprint arXiv:2408.12320*.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M Dai, Anja Hauth, Katie Millican, et al. 2023. Gemini: a family of highly capable multimodal models. *arXiv preprint arXiv:2312.11805*.

Trieu Trinh, Yuhuai Wu, Quoc Le, He He, and Thang Luong. 2024. Solving olympiad geometry without human demonstrations. *Nature*.

Nazmi Ekin Vural and Sefer Kalaman. 2024. Using artificial intelligence systems in news verification: An application on x. *İletişim Kuram ve Araştırma Dergisi*.

Jianrui Wang, Yitian Hong, Jiali Wang, Jiapeng Xu, Yang Tang, Qing-Long Han, and Jürgen Kurths. 2022. Cooperative and competitive multi-agent systems: From optimization to games. *IEEE/CAA Journal of Automatica Sinica*, 9:763–783.

Mengxin Wang, Dennis J. Zhang, and Heng Zhang. 2024. Large language models for market research: A data-augmentation approach. *arXiv preprint arXiv:2412.19363*.

K. Lalith Williams, Y. Durga Prasanth, and M. Jeyaselvi. 2024. Hybrid ai architecture using edge-cloud computing for secure v2x communication. *2024 9th International Conference on Communication and Electronics Systems (ICCES)*, pages 913–920.

Elizabeth Nathania Witanto, Yustus Eko Oktian, and Sang-Gon Lee. 2022. Toward data integrity architecture for cloud-based ai systems. *Symmetry*, 14:273.

Peng Xiao and Liang Chen. 2024. Athena: Retrieval-augmented legal judgment prediction with large language models. *arXiv:2410.11195*.

Yuhang Yao, Han Jin, Alay Dilipbhai Shah, Shanshan Han, Zijian Hu, Yide Ran, Dimitris Stripelis, Zhaozhuo Xu, Salman Avestimehr, and Chaoyang He. 2024a. Scalellm: A resource-frugal llm serving framework by optimizing end-to-end efficiency. *arXiv preprint arXiv:2408.00008*.

Yuhang Yao, Jianyi Zhang, Junda Wu, Chengkai Huang, Yu Xia, Tong Yu, Ruiyi Zhang, Sungchul Kim, Ryan Rossi, Ang Li, et al. 2024b. Federated large language models: Current progress and future directions. *arXiv preprint arXiv:2409.15723*.

Matei Zaharia, Omar Khattab, Lingjiao Chen, Jared Quincy Davis, Heather Miller, Chris Potts, James Zou, Michael Carbin, Jonathan Frankle, Naveen Rao, and Ali Ghodsi. 2024a. The shift from models to compound ai systems. https://bair.berkeley.edu/blog/2024/02/18/compound-ai-systems/.

Matei Zaharia et al. 2024b. The shift from models to compound ai systems. *Berkeley Artificial Intelligence Research, February*, 18.

Esteban Zavaleta-Monestel, Ricardo Quesada-Villaseñor, Sebastián Arguedas-Chacón, Jonathan García-Montero, Monserrat Barrantes-López, Juliana Salas-Segura, Adriana Anchía-Alfaro, Daniel Nieto-Bernal, and Daniel E Diaz-Juan. 2024. Revolutionizing healthcare: Qure.ai's innovations in medical diagnosis and treatment. *Cureus*, 16.

Bowen Zhang and Harold Soh. 2024. Extract, define, canonicalize: An llm-based framework for knowledge graph construction. *arXiv preprint arXiv:2404.03868*.

Boyu Zhang, Hongyang Yang, Tianyu Zhou, Muhammad Ali Babar, and Xiao-Yang Liu. 2023a. Enhancing financial sentiment analysis via retrieval augmented large language models. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, ICAIF '23, page 349–356, New York, NY, USA. Association for Computing Machinery.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B. Tenenbaum, and Chuang Gan. 2023b. Planning with large language models for code generation. *ArXiv*, abs/2303.05510.