LLMOps, MLOps, and DevOps Convergence: A Review of Operational Frameworks for Deploying Generative AI Systems at Scale

Dhiraj Kumar

Abstract

The rapid adoption of Large Language Models (LLMs) and generative AI systems in production environments has necessitated the convergence of DevOps, MLOps, and emerging LLMOps practices. This literature review systematically examines operational frameworks, tools, and methodologies for deploying and managing generative AI systems at scale. We analyze 30 highly relevant papers published between 2024 and 2025, focusing on critical operational aspects including continuous integration and deployment (CI/CD) pipelines for AI, prompt versioning and management, model monitoring and drift detection, hallucination detection mechanisms, evaluation frameworks, token cost optimization, Retrieval-Augmented Generation (RAG) deployment strategies, governance frameworks, and secure production operations. Our review identifies a clear evolution from traditional MLOps to specialized LLMOps practices, with emerging AgentOps addressing multi-agent system complexities. We present a comprehensive taxonomy of existing approaches, comparative analysis of operational frameworks, and identify significant research gaps including standardization challenges, observability limitations, and cost management complexities. The findings reveal that while foundational frameworks exist, substantial work remains in developing unified operational standards, automated evaluation pipelines, and robust governance mechanisms for enterprise-scale generative AI deployments. This review provides researchers and practitioners with a structured understanding of the current landscape and future research directions in operationalizing generative AI systems.

Keywords Llmops · Mlops · Devops · Generative ai · Large language models · Ci/cd · Prompt engineering · Model monitoring · Hallucination detection · Rag · Model drift · Governance · Production operations · Evaluation pipelines · Token cost monitoring

1 Introduction

The deployment of Large Language Models (LLMs) and generative AI systems in production environments represents a paradigm shift in artificial intelligence operations. Unlike traditional machine learning models, LLMs introduce unique operational challenges including non-deterministic outputs, massive computational requirements, complex prompt management, and novel failure modes such as hallucinations [1], [2], [3]. These challenges have catalyzed the emergence of LLMOps—a specialized discipline extending MLOps principles to address the distinctive characteristics of generative AI systems.

The convergence of DevOps, MLOps, and LLMOps practices reflects the growing recognition that deploying generative AI at scale requires integrated operational frameworks spanning the entire lifecycle from development to production [4], [5], [6]. Traditional DevOps practices, focused on software delivery automation, must now accommodate the stochastic nature of AI models. MLOps frameworks, designed for discriminative models with fixed inputs and outputs, require substantial adaptation to handle the open-ended nature of generative systems [7], [8]. This convergence is further complicated by the emergence of agentic AI systems, prompting the development of AgentOps methodologies for managing multi-agent orchestration and coordination [1], [9].

Recent industry adoption of LLMs has exposed critical gaps in existing operational practices. Organizations struggle with prompt versioning and management, lacking standardized approaches comparable to traditional code version control [10], [11]. Model monitoring becomes significantly more complex when outputs are natural language rather than discrete classifications [12], [13]. Evaluation frameworks must balance automated metrics with human judgment, while managing the substantial costs associated with continuous evaluation [14], [15]. Furthermore, the deployment of Retrieval-Augmented Generation (RAG) systems introduces additional operational complexity through the integration of vector databases, retrieval mechanisms, and dynamic knowledge updates [16], [23].

Governance and security concerns have become paramount as generative AI systems are deployed in regulated industries and sensitive applications [17], [18], [19]. Organizations must implement robust frameworks for detecting and mitigating hallucinations, ensuring output quality, managing token costs, monitoring model drift, and maintaining compliance with evolving AI regulations [20], [21], [22]. The lack of standardized operational frameworks and best practices creates significant barriers to enterprise adoption and scalability.

This literature review addresses these challenges through a systematic examination of operational frameworks for deploying generative AI systems at scale. We analyze 30 highly relevant papers published between 2024 and 2025, focusing on the convergence of DevOps, MLOps, and LLMOps practices. Our contributions include: (1) a comprehensive taxonomy of operational approaches and frameworks, (2) detailed analysis of technical solutions for key operational challenges, (3) comparative evaluation of existing methodologies, (4) identification of critical research gaps, and (5) recommendations for future research directions. This

review provides both researchers and practitioners with a structured understanding of the current state and future trajectory of generative AI operations.

2 Background and Motivation

2.1 Evolution from DevOps to MLOps to LLMOps

The operational practices for AI systems have evolved through distinct phases, each addressing increasingly complex requirements. DevOps emerged in the late 2000s to bridge the gap between software development and IT operations, emphasizing automation, continuous integration, and rapid deployment cycles [24], [29]. Traditional DevOps practices focused on deterministic software systems where code behavior is predictable and testable through conventional unit and integration tests.

MLOps extended DevOps principles to machine learning systems, introducing specialized practices for data versioning, model training pipelines, experiment tracking, and model serving infrastructure [7], [28]. MLOps frameworks addressed challenges unique to ML systems including data drift, model retraining, A/B testing of models, and monitoring of prediction quality [26]. However, MLOps was primarily designed for discriminative models with well-defined input-output relationships and quantifiable performance metrics.

The emergence of LLMs and generative AI has necessitated further evolution toward LLMOps [1], [2], [3]. LLMOps addresses distinctive challenges including prompt engineering and versioning, managing non-deterministic outputs, detecting hallucinations, optimizing token costs, and handling massive model sizes [10], [11]. The stochastic nature of LLM outputs requires fundamentally different evaluation and monitoring approaches compared to traditional ML models [14], [15]. Furthermore, the rise of multi-agent systems has prompted the development of AgentOps methodologies for orchestrating and monitoring complex agent interactions [1], [9].

2.2 Unique Challenges of Generative AI Operations

Generative AI systems introduce several operational challenges that distinguish them from traditional software and ML systems. First, the non-deterministic nature of LLM outputs complicates testing, validation, and monitoring [3], [30]. Unlike classification models with discrete outputs, generative systems produce open-ended text requiring semantic evaluation rather than simple accuracy metrics [14], [15].

Second, prompt engineering has emerged as a critical operational concern, yet lacks standardized versioning and management practices [10], [11]. Prompts function as a new form of "code" that significantly impacts system behavior, but existing version control systems are inadequate for tracking prompt evolution and managing prompt libraries [2], [17].

Third, hallucinations—plausible but factually incorrect outputs—represent a novel failure mode requiring specialized detection and mitigation strategies [16], [21]. Traditional software testing approaches cannot adequately address this challenge, necessitating new evaluation frameworks and human-in-the-loop validation processes [14], [15].

Fourth, the computational costs and token-based pricing models of LLM APIs introduce new operational considerations around cost monitoring, optimization, and budget management [2], [20]. Organizations must balance quality requirements against substantial inference costs, particularly for high-volume applications.

Fifth, RAG systems add operational complexity through the integration of vector databases, retrieval mechanisms, and dynamic knowledge bases [16], [23]. Deploying RAG at scale requires coordinating multiple components including embedding models, vector stores, retrieval algorithms, and LLM inference, each with distinct operational requirements.

Finally, governance and compliance requirements for generative AI systems are evolving rapidly, particularly in regulated industries [17], [18], [19], [22]. Organizations must implement frameworks for ensuring output quality, maintaining audit trails, managing data privacy, and complying with emerging AI regulations.

2.3 Motivation for Convergence

The convergence of DevOps, MLOps, and LLMOps practices is motivated by the recognition that generative AI systems require integrated operational frameworks spanning traditional software engineering, machine learning operations, and LLM-specific practices [4], [5], [6]. Siloed approaches create inefficiencies, gaps in observability, and challenges in maintaining end-to-end system reliability [13], [29].

Organizations deploying generative AI at scale require unified frameworks that address the full operational lifecycle including development, testing, deployment, monitoring, and governance [17], [18]. This convergence is particularly critical for enterprise applications where generative AI components must integrate seamlessly with existing software systems and operational processes [20], [27]. The emergence of cloud-native architectures and containerized deployments further necessitates integrated operational approaches that span infrastructure, data, models, and applications [13], [25].

3 Review Methodology

This literature review follows a systematic approach to identify, analyze, and synthesize research on operational frameworks for deploying generative AI systems at scale. Our methodology encompasses search strategy, selection criteria, data extraction, and analysis procedures.

3.1 Search Strategy and Data Sources

We conducted comprehensive searches across multiple academic databases and preprint repositories including SciSpace, Google Scholar, and ArXiv. The search focused on papers published between 2024 and 2025 to capture the most recent developments in this rapidly evolving field. Search terms included combinations of "LLMOps," "MLOps," "DevOps," "generative AI operations," "LLM deployment," "prompt management," "model monitoring," "hallucination detection," "RAG deployment," "AI governance," and related operational terms.

The initial search retrieved 85 unique papers after deduplication. These papers were then ranked by relevance based on their alignment with the review objectives, focusing on operational frameworks, deployment practices, monitoring approaches, and governance mechanisms for generative AI systems.

3.2 Selection Criteria and Scope

From the initial corpus, we selected the top 30 papers based on relevance ranking and quality assessment. Selection criteria included: (1) direct relevance to LLMOps, MLOps, or DevOps practices for generative AI, (2) focus on operational aspects of deployment and production management, (3) technical depth and contribution to the field, (4) recency and alignment with current practices, and (5) coverage of key operational concerns including CI/CD, monitoring, evaluation, governance, or security.

Papers were included if they addressed one or more of the following topics: operational frameworks for LLMs, CI/CD pipelines for AI systems, prompt versioning and management, model monitoring and observability, hallucination detection, evaluation frameworks, model drift detection, token cost monitoring, RAG deployment strategies, governance frameworks, or secure production operations. Papers focusing solely on model architecture or training methodologies without operational considerations were excluded.

3.3 Data Extraction and Analysis

For each selected paper, we extracted key information including technical contributions, operational focus areas, challenges addressed, and proposed solutions. We employed a structured analysis approach to identify common themes, categorize operational approaches, and synthesize findings across papers.

The analysis focused on three primary dimensions: (1) technical approaches and frameworks proposed, (2) operational focus areas and challenges addressed, and (3) gaps and limitations identified. This multi-dimensional analysis enabled the development of a comprehensive taxonomy of operational approaches and identification of research gaps requiring further investigation.

4 Technical Background

4.1 DevOps Fundamentals

DevOps represents a set of practices, tools, and cultural philosophies that automate and integrate software development and IT operations [24], [29]. Core DevOps principles include continuous integration (CI), continuous delivery (CD), infrastructure as code (IaC), automated testing, and continuous monitoring. CI/CD pipelines automate the process of building, testing, and deploying software, enabling rapid and reliable releases [29]. Infrastructure as code enables version-controlled, reproducible infrastructure provisioning and management. These foundational practices provide the basis for extending operational frameworks to AI systems.

4.2 MLOps Principles

MLOps extends DevOps practices to machine learning systems, addressing unique challenges of data-driven models [7], [28]. Key MLOps components include data versioning and lineage tracking, experiment management, model training pipelines, model registry and versioning, model serving infrastructure, and prediction monitoring [26]. MLOps frameworks emphasize reproducibility, enabling teams to track experiments, version datasets and models, and reproduce training runs. Model monitoring focuses on detecting data drift, concept drift, and degradation in prediction quality over time. However, traditional MLOps frameworks were designed primarily for discriminative models with structured inputs and discrete outputs.

4.3 LLMOps Distinctive Characteristics

LLMOps addresses operational challenges specific to large language models and generative AI systems [1], [2], [3]. Unlike traditional ML models, LLMs exhibit several distinctive characteristics requiring specialized operational approaches. First, LLMs are typically accessed via API rather than deployed as custom models, shifting operational focus from model training to prompt engineering and API management [2], [10]. Second, LLM outputs are non-deterministic and open-ended, requiring semantic

evaluation rather than simple accuracy metrics [14], [15]. Third, prompts function as a form of "code" that significantly impacts system behavior, necessitating prompt versioning and management practices [10], [11]. Fourth, LLMs are prone to hallucinations—generating plausible but incorrect information—requiring specialized detection and mitigation strategies [16], [21]. Fifth, token-based pricing models introduce cost optimization as a critical operational concern [2], [20]. These distinctive characteristics necessitate the evolution of operational practices beyond traditional MLOps.

4.4 Retrieval-Augmented Generation (RAG)

RAG systems enhance LLM capabilities by retrieving relevant information from external knowledge bases to ground model outputs in factual data [16], [23]. RAG architectures typically consist of three main components: an embedding model for encoding queries and documents, a vector database for efficient similarity search, and an LLM for generating responses based on retrieved context. Deploying RAG systems at scale introduces additional operational complexity including managing vector database infrastructure, optimizing retrieval performance, maintaining knowledge base freshness, and coordinating multiple system components [23]. RAG deployment requires integrated operational frameworks spanning data pipelines, embedding models, vector stores, and LLM inference.

4.5 Agentic AI and Multi-Agent Systems

Agentic AI systems represent an evolution beyond single-model deployments, involving autonomous agents that can plan, reason, and execute complex tasks [1], [7], [9]. Multi-agent systems coordinate multiple specialized agents, each with distinct capabilities and responsibilities. AgentOps extends LLMOps to address the unique challenges of managing agent lifecycles, monitoring agent interactions, debugging multi-agent behaviors, and ensuring coordination and consistency across agents [1], [9]. Operational frameworks for agentic systems must handle increased complexity in observability, debugging, and governance compared to single-model deployments.

5 Taxonomy of Existing Approaches

Based on our analysis of the literature, we present a comprehensive taxonomy organizing operational approaches for generative AI systems across multiple dimensions.

5.1 Operational Lifecycle Frameworks

Several papers propose comprehensive frameworks spanning the full operational lifecycle of generative AI systems. Joshi et al. [1] present an integrated framework combining MLOps, LLMOps, and AgentOps principles, emphasizing the evolution from traditional ML operations to specialized practices for generative and agentic AI. The framework addresses monitoring, debugging, and secure lifecycle management across different operational maturity levels.

Madicharla [2] outlines key trends in LLMOps including distributed model serving architectures, advanced prompt management systems, and real-time monitoring frameworks. The work emphasizes the integration of observability tools, cost optimization strategies, and governance mechanisms as essential components of mature LLMOps practices.

Özer [3] provides a systematic technical survey of LLMOps lifecycle, tools, challenges, and emerging practices. The survey identifies critical operational stages including development, testing, deployment, monitoring, and maintenance, with specific tools and practices for each stage.

Bandara et al. [7] present a practical guide for designing, developing, and deploying production-grade agentic AI workflows, emphasizing the need for specialized operational frameworks that extend beyond traditional LLMOps to address multi-agent coordination and orchestration.

5.2 CI/CD and Deployment Automation

Several works focus specifically on CI/CD pipelines and deployment automation for generative AI systems. Joshi [17] provides a comprehensive review of generative AI and DevOps pipelines, examining CI/CD practices, agentic automation, and MLOps integration. The work highlights the need for specialized testing frameworks that can handle non-deterministic LLM outputs and the importance of automated deployment pipelines that incorporate prompt versioning and model evaluation.

Brahmandam et al. [29] propose AI-augmented DevOps frameworks that embed LLMs into the DevOps lifecycle itself, using generative AI to enhance tasks such as log summarization, infrastructure-as-code generation, and real-time incident remediation. The framework demonstrates how LLMs can improve deployment frequency, mean time to recovery (MTTR), and change failure rates.

Fitsilis et al. [24] introduce DOLLmC (DevOps for Large Language model Customization), a framework specifically designed for customizing and deploying LLMs in production environments. The framework emphasizes continuous integration of prompt changes, automated testing of LLM outputs, and continuous deployment with rollback capabilities.

Bhattarai [15] examines scaling generative AI for self-healing DevOps pipelines, proposing technical approaches for using LLMs to automatically detect and remediate pipeline failures, reducing manual intervention and improving system reliability.

5.3 Prompt Management and Versioning

Prompt engineering and management emerge as critical operational concerns across multiple papers. Madicharla [2] emphasizes advanced prompt management systems as a key trend in LLMOps, highlighting the need for version control, testing frameworks, and optimization strategies for prompts.

Gadiraju et al. [4] examine the intersection of DataOps and LLMOps, focusing on automating cloud-based AI workflows from data ingestion to prompt optimization. The work proposes frameworks for systematically testing and optimizing prompts as part of automated pipelines.

Hassan et al. [18] discuss operationalizing generative AI in enterprise workflows, emphasizing the importance of prompt libraries, versioning systems, and governance frameworks for managing prompt evolution and ensuring consistency across deployments.

5.4 Monitoring and Observability

Model monitoring and observability represent critical operational capabilities addressed by multiple papers. Chittala [8] examines AIOps approaches for automating AI deployment and management, emphasizing real-time monitoring, anomaly detection, and automated response mechanisms for LLM systems.

Ueno et al. [25] introduce GenAIOps for model-agility, focusing on continuous monitoring of generative model performance, detecting quality degradation, and enabling rapid model switching or rollback when issues are detected.

Parthasarathy et al. [14] propose engineering frameworks for LLM-powered multi-agent systems with emphasis on observability, including distributed tracing, agent interaction monitoring, and performance analytics across complex agent workflows.

Stone et al. [28] provide insights into MLOps maturity, lifecycle, tools, and careers, emphasizing the evolution of monitoring practices from traditional ML to generative AI systems, including the need for semantic evaluation and human-in-the-loop validation.

5.5 Evaluation and Quality Assurance

Evaluation frameworks for generative AI systems represent a significant focus area. Kalluru et al. [16] present NLPOps, a comprehensive framework for secure development and scalable deployment of multifaceted LLMs, emphasizing automated evaluation pipelines, hallucination detection mechanisms, and quality assurance processes.

Kohl et al. [21] introduce a Generative AI Toolkit that provides frameworks for increasing the quality of LLM-based applications over their whole lifecycle, including evaluation methodologies, testing strategies, and continuous quality monitoring.

Prabhune et al. [23] examine deploying LLMs with Retrieval-Augmented Generation, addressing evaluation challenges specific to RAG systems including retrieval quality, context relevance, and end-to-end response quality assessment.

5.6 Cost Optimization and Resource Management

Token cost monitoring and resource optimization emerge as critical operational concerns. Krishnamoorthy et al. [5] propose DNN-powered MLOps pipeline optimization for LLMs, focusing on automated deployment and resource management to minimize computational costs while maintaining performance requirements.

Nadiminty [20] discusses building infrastructure for generative AI workloads, emphasizing lessons from the field including cost monitoring, resource allocation strategies, and optimization techniques for managing expensive LLM inference at scale.

Devarapalli et al. [13] present a unified framework for cloud-native LLMOps meeting DataOps, focusing on high-volume analytical systems with emphasis on cost-effective infrastructure design and resource optimization strategies.

5.7 Governance and Compliance

Governance frameworks for generative AI systems are addressed by several papers, particularly for regulated industries. Majumder et al. [6] examine enhancing AI governance in the financial industry, proposing frameworks for ensuring compliance, maintaining audit trails, and managing risks associated with generative AI deployments.

Sanodia [19] discusses the future of regulated AI, focusing on scaling LLMs with oversight and precision, emphasizing governance mechanisms, compliance frameworks, and risk management strategies for enterprise deployments.

The work on LLMs in the cloud [22] examines best practices for scaling generative AI in regulated industries, addressing governance, security, compliance, and operational excellence requirements for enterprise-scale deployments.

5.8 Security and Trust

Security and trustworthiness of generative AI systems represent critical operational concerns. The work on advancing trustworthy AI in the cloud era [26] examines privacy-preserving MLOps, security frameworks, and trust mechanisms for generative models deployed in production environments.

Kalluru et al. [16] emphasize secure development practices in their NLPOps framework, addressing security concerns throughout the LLM lifecycle including secure prompt handling, output filtering, and protection against adversarial attacks.

Mishra [11] examines software engineering challenges in deploying generative AI models at scale, highlighting security vulnerabilities, trust issues, and the need for robust security frameworks integrated into operational practices.

6 Detailed Literature Review

6.1 Integrated Operational Frameworks

The most comprehensive operational frameworks address the full lifecycle of generative AI systems from development through production deployment and maintenance. Joshi et al. [1] provide a systematic review of over 100 articles and industry practices, identifying critical gaps in current operational approaches and proposing an integrated framework combining MLOps, LLMOps, and AgentOps principles. Their framework delineates distinct characteristics, challenges, and best practices for each operational stage, emphasizing how AgentOps extends prior concepts to manage multi-agent system complexities including monitoring, debugging, and secure lifecycle management. The work demonstrates the clear evolution from traditional MLOps focused on discriminative models to specialized LLMOps practices addressing generative AI challenges, culminating in AgentOps for managing autonomous agent systems.

Madicharla [2] examines the evolution of LLMOps, identifying latest trends and developments that are redefining enterprise AI operations. Key trends include distributed model serving architectures that enable scalable LLM deployment, advanced prompt management systems for versioning and optimizing prompts, real-time monitoring frameworks for detecting quality degradation, and integrated governance mechanisms for ensuring compliance and ethical AI practices. The work emphasizes that mature LLMOps requires integration of observability tools, cost optimization strategies, and governance frameworks as essential operational components rather than afterthoughts.

Özer [3] provides a systematic technical survey covering the LLMOps lifecycle, tools, challenges, and emerging practices. The survey identifies critical operational stages and maps specific tools and practices to each stage, providing practitioners with actionable guidance for implementing LLMOps in their organizations. The work highlights the rapid evolution of the LLMOps tool ecosystem and the need for standardization to enable interoperability and reduce operational complexity.

6.2 CI/CD and Deployment Automation

Continuous integration and deployment practices for generative AI systems require substantial adaptation from traditional DevOps approaches. Joshi [17] provides a comprehensive review of generative AI and DevOps pipelines, examining CI/CD practices, agentic automation, and MLOps integration. The work highlights fundamental challenges in testing non-deterministic LLM outputs, emphasizing the need for specialized testing frameworks that combine automated metrics with human evaluation. The review identifies prompt versioning as a critical gap in current CI/CD practices, noting that existing version control systems are inadequate for tracking prompt evolution and managing prompt libraries effectively.

Brahmandam et al. [29] propose AI-augmented DevOps frameworks that leverage LLMs to enhance the DevOps lifecycle itself. Their modular five-layer framework consists of observation, inference, action, feedback, and interaction layers, each designed to support autonomous, traceable, and policy-compliant decision-making. Implementation using GPT-4 and reinforcement learning demonstrates significant improvements in deployment frequency, MTTR, and change failure rates. The work shows how LLMs can enhance log summarization, infrastructure-as-code generation, and real-time incident remediation, creating self-improving DevOps pipelines.

Fitsilis et al. [24] introduce DOLLmC (DevOps for Large Language model Customization), specifically designed for customizing and deploying LLMs in production. The framework emphasizes continuous integration of prompt changes, automated testing of LLM outputs using both quantitative metrics and qualitative assessments, and continuous deployment with rollback capabilities when quality degradation is detected. DOLLmC addresses the unique challenge of managing LLM customization as a continuous process rather than a one-time activity.

Bhattarai [15] examines scaling generative AI for self-healing DevOps pipelines, proposing technical approaches for using LLMs to automatically detect and remediate pipeline failures. The work demonstrates how generative AI can reduce manual intervention in DevOps operations, improving system reliability and reducing operational overhead. However, the paper also identifies challenges in ensuring the reliability of AI-driven remediation actions and the need for human oversight in critical scenarios.

6.3 DataOps and LLMOps Integration

The integration of data operations with LLM operations emerges as a critical theme for enterprise deployments. Gadiraju et al. [4] examine DataOps meeting LLMOps, focusing on automating cloud-based AI workflows from data ingestion to prompt optimization. Their framework addresses the full data-to-deployment pipeline, emphasizing automated data quality checks, continuous data integration, prompt optimization based on data characteristics, and monitoring of data-model interactions. The work highlights that effective LLMOps requires tight integration with data operations, as LLM performance is highly dependent on data quality and relevance.

Devarapalli et al. [13] present a unified framework for cloud-native LLMOps meeting DataOps, specifically designed for high-volume analytical systems. The framework emphasizes cost-effective infrastructure design, resource optimization strategies, and scalable architectures that can handle enterprise-scale data volumes. The work demonstrates how cloud-native approaches enable elastic scaling of both data processing and LLM inference, optimizing costs while maintaining performance requirements.

6.4 Prompt Engineering and Management

Prompt engineering has emerged as a critical operational discipline requiring specialized management practices. Hassan et al. [18] discuss operationalizing generative AI in enterprise workflows, emphasizing the importance of prompt libraries, versioning systems, and governance frameworks for managing prompt evolution. The work highlights that prompts function as a new form of "code" that significantly impacts system behavior, yet existing software engineering practices are inadequate for prompt management. Key challenges include tracking prompt performance over time, managing prompt dependencies, ensuring consistency across deployments, and enabling collaboration among prompt engineers.

Madicharla [2] identifies advanced prompt management systems as a key trend in LLMOps evolution, emphasizing the need for version control, testing frameworks, and optimization strategies. The work proposes treating prompts as first-class artifacts in the development lifecycle, with dedicated tools for versioning, testing, and deployment. Prompt optimization should be automated where possible, using techniques such as few-shot learning, chain-of-thought prompting, and retrieval-augmented generation to improve performance.

Gadiraju et al. [4] examine prompt optimization as part of automated workflows, proposing frameworks for systematically testing and optimizing prompts based on data characteristics and performance metrics. The work emphasizes that prompt optimization should be continuous rather than one-time, with automated pipelines that test prompt variations and select optimal configurations based on evaluation metrics.

6.5 Model Monitoring and Observability

Monitoring generative AI systems requires fundamentally different approaches compared to traditional ML models. Chittala [8] examines AIOps approaches for automating AI deployment and management of LLMs, emphasizing real-time monitoring, anomaly detection, and automated response mechanisms. The work highlights that traditional monitoring metrics such as latency and throughput are insufficient for LLM systems, which require semantic quality monitoring, hallucination detection, and user satisfaction tracking. The proposed framework integrates multiple monitoring dimensions including performance metrics, quality metrics, cost metrics, and user engagement metrics.

Ueno et al. [25] introduce GenAIOps for model-agility, focusing on continuous monitoring of generative model performance and enabling rapid model switching or rollback when issues are detected. The framework emphasizes the need for real-time quality monitoring that can detect subtle degradations in output quality before they impact users significantly. Model-agility—the ability to quickly switch between different models or model versions—emerges as a critical capability for maintaining service quality in production environments.

Parthasarathy et al. [14] propose engineering frameworks for LLM-powered multi-agent systems with emphasis on observability. The work addresses the increased complexity of monitoring multi-agent systems, requiring distributed tracing, agent interaction monitoring, and performance analytics across complex agent workflows. The framework enables operators to understand agent behaviors, identify bottlenecks, debug failures, and optimize agent coordination patterns.

Stone et al. [28] provide insights into MLOps maturity and the evolution of monitoring practices from traditional ML to generative AI systems. The work emphasizes that monitoring maturity for generative AI requires progression through multiple stages, from basic infrastructure monitoring to advanced semantic evaluation and human-in-the-loop validation. Organizations must invest in building monitoring capabilities that can assess output quality, detect hallucinations, and track user satisfaction in addition to traditional performance metrics.

6.6 Evaluation Frameworks and Quality Assurance

Evaluation of generative AI systems represents a significant challenge requiring specialized frameworks. Kalluru et al. [16] present NLPOps, a comprehensive framework for secure development and scalable deployment of multifaceted LLMs. The framework emphasizes automated evaluation pipelines that combine quantitative metrics such as perplexity, BLEU, and ROUGE with qualitative assessments including human evaluation and domain expert review. Hallucination detection mechanisms are integrated into the evaluation pipeline, using techniques such as fact-checking against knowledge bases, consistency checking across multiple generations, and confidence scoring. Quality assurance processes include continuous testing, regression detection, and automated alerts when quality degradation is detected.

Kohl et al. [21] introduce a Generative AI Toolkit providing frameworks for increasing quality of LLM-based applications over their whole lifecycle. The toolkit includes evaluation methodologies spanning multiple dimensions including factual accuracy, relevance, coherence, safety, and bias. Testing strategies combine automated testing with human evaluation, using techniques such as adversarial testing, edge case generation, and systematic prompt variation. Continuous quality monitoring tracks evaluation metrics over time, detecting trends and triggering alerts when thresholds are exceeded.

Prabhune et al. [23] examine deploying LLMs with Retrieval-Augmented Generation, addressing evaluation challenges specific to RAG systems. RAG evaluation requires assessing multiple components including retrieval quality (precision and recall of retrieved documents), context relevance (alignment between retrieved context and query), and end-to-end response quality (factual accuracy and coherence of generated responses). The work proposes evaluation frameworks that assess each component independently while also measuring end-to-end system performance, enabling operators to identify and address specific failure modes.

6.7 Hallucination Detection and Mitigation

Hallucinations—plausible but factually incorrect outputs—represent a critical operational challenge for generative AI systems. Kalluru et al. [16] propose hallucination detection mechanisms integrated into operational pipelines, using techniques such as fact-checking against authoritative knowledge bases, consistency checking across multiple generations with varied prompts, and confidence scoring based on model uncertainty estimates. The work emphasizes that hallucination detection should be automated where possible but must include human review for high-stakes applications.

Kohl et al. [21] address hallucination mitigation through their Generative AI Toolkit, proposing multi-layered approaches including prompt engineering to reduce hallucination likelihood, retrieval-augmented generation to ground outputs in factual data, output filtering to detect and flag potential hallucinations, and human-in-the-loop validation for critical applications. The work highlights that complete elimination of hallucinations is currently infeasible, requiring operational frameworks that detect and manage hallucinations rather than prevent them entirely.

Prabhune et al. [23] examine hallucination challenges specific to RAG systems, noting that while RAG reduces hallucination rates by grounding outputs in retrieved context, it introduces new failure modes including retrieval of irrelevant or incorrect information and misinterpretation of retrieved context. The work proposes evaluation frameworks that specifically assess hallucination rates in RAG systems and identify conditions under which hallucinations are most likely to occur.

6.8 Cost Optimization and Resource Management

Token-based pricing models for LLM APIs introduce cost optimization as a critical operational concern. Krishnamoorthy et al. [5] propose DNN-powered MLOps pipeline optimization for LLMs, focusing on automated deployment and resource management to minimize computational costs while maintaining performance requirements. The framework uses deep neural networks to predict optimal resource allocations based on workload characteristics, enabling dynamic scaling that balances cost and performance. Cost monitoring is integrated throughout the pipeline, providing real-time visibility into token usage, API costs, and infrastructure expenses.

Nadiminty [20] discusses building infrastructure for generative AI workloads, sharing lessons from field deployments including cost monitoring strategies, resource allocation approaches, and optimization techniques. The work emphasizes that cost optimization for LLMs requires multi-faceted approaches including prompt optimization to reduce token usage, caching of common queries to avoid redundant API calls, model selection based on cost-performance tradeoffs, and batch processing where latency requirements permit. Organizations must implement comprehensive cost monitoring and alerting to prevent unexpected expenses and enable data-driven optimization decisions.

Devarapalli et al. [13] present cost-effective infrastructure design for cloud-native LLMOps, emphasizing resource optimization strategies for high-volume analytical systems. The framework leverages cloud-native capabilities including auto-scaling, spot instances, and serverless computing to minimize infrastructure costs. The work demonstrates how careful infrastructure design can significantly reduce operational costs while maintaining performance and reliability requirements for enterprise-scale deployments.

6.9 RAG Deployment and Operations

Retrieval-Augmented Generation systems introduce additional operational complexity requiring specialized deployment strategies. Prabhune et al. [23] provide comprehensive guidance on deploying LLMs with RAG, addressing the full operational lifecycle including vector database selection and configuration, embedding model deployment and management, retrieval algorithm optimization, LLM integration and prompt engineering, and end-to-end system monitoring. The work emphasizes that RAG deployment requires coordinating multiple components with distinct operational characteristics, necessitating integrated operational frameworks that span data pipelines, embedding models, vector stores, and LLM inference.

Key operational challenges for RAG systems include maintaining knowledge base freshness through continuous data ingestion and re-embedding, optimizing retrieval performance to meet latency requirements while maintaining quality, managing vector database infrastructure at scale, and monitoring end-to-end system performance including retrieval quality and response accuracy. The work proposes operational frameworks that address each of these challenges through automated pipelines, comprehensive monitoring, and continuous optimization.

Hassan et al. [18] discuss operationalizing RAG in enterprise workflows, emphasizing governance frameworks for managing knowledge bases, ensuring data quality, and maintaining compliance with data privacy regulations. RAG systems in enterprise environments must handle sensitive data, requiring careful attention to access controls, data lineage tracking, and audit trails. The work highlights that RAG operations require close collaboration between data teams, ML teams, and application teams to ensure end-to-end system reliability and compliance.

6.10 Governance and Compliance Frameworks

Governance frameworks for generative AI systems are critical for enterprise adoption, particularly in regulated industries. Majumder et al. [6] examine enhancing AI governance in the financial industry using IBM watsonx.governance, proposing comprehensive frameworks for ensuring compliance, maintaining audit trails, and managing risks associated with generative AI deployments. The framework addresses multiple governance dimensions including model risk management, bias detection and mitigation, explainability and transparency, compliance with regulatory requirements, and incident response procedures. The work emphasizes that governance must be integrated throughout the operational lifecycle rather than treated as a separate concern.

Sanodia [19] discusses the future of regulated AI, focusing on scaling LLMs with oversight and precision. The work examines governance mechanisms including human oversight requirements, approval workflows for high-stakes decisions, audit trails for compliance verification, and risk assessment frameworks. The paper highlights that governance requirements vary significantly across industries and use cases, necessitating flexible frameworks that can be tailored to specific regulatory and organizational requirements.

The work on LLMs in the cloud [22] examines best practices for scaling generative AI in regulated industries, addressing governance, security, compliance, and operational excellence. Key practices include implementing comprehensive logging and audit trails, establishing clear accountability and approval processes, conducting regular compliance assessments, maintaining documentation of model behaviors and limitations, and implementing incident response procedures for handling failures or compliance violations. The work emphasizes that governance frameworks must balance risk management with operational efficiency, avoiding overly restrictive processes that impede innovation while ensuring adequate oversight and control.

6.11 Security and Trustworthiness

Security and trust represent critical operational concerns for production generative AI systems. The work on advancing trustworthy AI in the cloud era [26] examines privacy-preserving MLOps, security frameworks, and trust mechanisms for generative models. Key security considerations include protecting sensitive data used in prompts and training, preventing adversarial attacks and prompt injection, securing API access and authentication, implementing output filtering to prevent harmful content, and maintaining data privacy and compliance with regulations such as GDPR. The work proposes security frameworks that integrate these considerations throughout the operational lifecycle.

Kalluru et al. [16] emphasize secure development practices in their NLPOps framework, addressing security concerns including secure prompt handling to prevent injection attacks, output filtering to detect and block harmful content, protection against adversarial attacks that attempt to manipulate model behavior, and secure deployment practices including encryption, access controls, and network security. The work highlights that security for generative AI systems requires specialized approaches beyond traditional application security, addressing unique vulnerabilities such as prompt injection and adversarial manipulation.

Mishra [11] examines software engineering challenges in deploying generative AI models at scale, highlighting security vulnerabilities, trust issues, and the need for robust security frameworks integrated into operational practices. The work identifies several security challenges specific to generative AI including the difficulty of validating all possible outputs, the risk of generating harmful or biased content, the potential for adversarial manipulation, and the challenges of maintaining security while enabling necessary flexibility. The paper emphasizes that security frameworks must be continuously updated to address emerging threats and vulnerabilities.

6.12 Agentic AI and Multi-Agent Operations

The emergence of agentic AI systems introduces new operational complexities requiring specialized frameworks. Bandara et al. [7] present a practical guide for designing, developing, and deploying production-grade agentic AI workflows. The work emphasizes that agentic systems require operational frameworks extending beyond traditional LLMOps to address multi-agent coordination, autonomous decision-making, and complex workflow orchestration. Key operational challenges include monitoring agent behaviors and interactions, debugging multi-agent failures, ensuring coordination and consistency across agents, managing agent lifecycles, and implementing governance frameworks for autonomous systems.

Parthasarathy et al. [14] propose engineering frameworks for LLM-powered multi-agent systems with emphasis on CloudOps integration. The framework addresses the operational complexity of deploying multi-agent systems in cloud environments, including distributed tracing across agent interactions, performance monitoring and optimization, resource allocation and scaling, and failure detection and recovery. The work demonstrates how cloud-native approaches enable scalable deployment of multi-agent systems while maintaining observability and control.

Joshi et al. [1] examine the evolution toward AgentOps, identifying how this emerging discipline extends LLMOps to manage the unique complexities of multi-agent systems. AgentOps addresses challenges including agent lifecycle management from development through deployment and retirement, monitoring and observability across distributed agent systems, debugging complex multi-agent interactions and emergent behaviors, security and governance for autonomous agents, and coordination mechanisms to ensure consistent and reliable system behavior. The work emphasizes that AgentOps represents a natural evolution of operational practices as AI systems become increasingly autonomous and complex.

6.13 Industry-Specific Considerations

Several papers address operational considerations specific to particular industries or deployment contexts. Majumder et al. [6] focus on AI governance in the financial industry, highlighting regulatory requirements, risk management frameworks, and compliance considerations specific to financial services. The work emphasizes that financial institutions face stringent regulatory oversight requiring comprehensive audit trails, explainability, and risk management frameworks for AI systems.

Sanodia [19] examines regulated AI more broadly, discussing operational requirements for industries including healthcare, finance, and government where AI systems must comply with strict regulatory frameworks. The work highlights that regulated industries require enhanced governance, comprehensive documentation, rigorous testing and validation, and clear accountability frameworks.

Nadiminty [20] shares lessons from building infrastructure for generative AI workloads across multiple industries, identifying common challenges and best practices. The work emphasizes that while operational requirements vary across industries, common themes include the need for robust monitoring, cost optimization, security, and governance frameworks.

7 Comparative Analysis

Table \reftab:comparison presents a comparative analysis of major operational frameworks examined in this review, highlighting their focus areas, key capabilities, and distinctive characteristics.

Table 1. Comparative Analysis of Operational Frameworks for Generative AI Systems

Framework/Approach	Primary Focus	Key Capabilities	Operational Coverage	Maturity Level
Integrated MLOps/LLMOps/AgentOps [1]	Full lifecycle integration	Monitoring, debugging, secure lifecycle management, multi-agent coordination	Development, deployment, monitoring, governance	Comprehensive
LLMOps Evolution Framework [2]	Enterprise operations	Distributed serving, prompt management, real-time monitoring, governance	Deployment, monitoring, cost optimization, governance	Mature
Systematic LLMOps Survey [3]	Lifecycle and tools	Tool mapping, stage-specific practices, emerging trends	Full lifecycle coverage	Survey/taxonomy
DataOps-LLMOps Integration [4]	Data-to-deployment	Automated workflows, data	Data pipelines, deployment, optimization	Integrated

		quality, prompt optimization		
DNN-Powered MLOps [5]	Resource optimization	Automated deployment, resource management, cost optimization	Deployment, resource management	Optimization-focused
AI Governance (Financial) [6]	Compliance and risk	Audit trails, risk management, regulatory compliance	Governance, compliance, risk management	Industry-specific
Agentic AI Workflows [7]	Multi-agent systems	Agent coordination, workflow orchestration, lifecycle management	Agent development, deployment, monitoring	Emerging
AIOps for LLMs [8]	Automated operations	Real-time monitoring, anomaly detection, automated response	Monitoring, incident response	Automation-focused
DOLLmC Framework [24]	LLM customization	CI/CD for prompts, automated testing, continuous deployment	Development, testing, deployment	DevOps-focused
AI-Augmented DevOps [29]	Self-improving pipelines	LLM-enhanced operations, automated remediation, IaC generation	CI/CD, infrastructure, incident management	Advanced automation
NLPOps Framework [16]	Security and quality	Automated evaluation, hallucination detection, secure deployment	Development, evaluation, security, deployment	Security-focused
GenAI Toolkit [21]	Quality assurance	Lifecycle quality management, testing strategies, continuous monitoring	Evaluation, testing, quality monitoring	Quality-focused
RAG Deployment [23]	Retrieval-augmented systems	Vector DB management, retrieval optimization, end-to-end monitoring	RAG-specific operations	Specialized
GenAIOps Model-Agility [25]	Rapid model switching	Performance monitoring, quality detection, model rollback	Monitoring, deployment agility	Agility-focused
Multi-Agent CloudOps [14]	Cloud-native agents	Distributed tracing, agent monitoring, cloud integration	Agent monitoring, cloud operations	Cloud-focused

The comparative analysis reveals several key insights. First, operational frameworks vary significantly in scope, ranging from comprehensive lifecycle frameworks [1], [2], [3] to specialized approaches focusing on specific operational aspects such as security [16], cost optimization [5], or RAG deployment [23]. Organizations must select or combine frameworks based on their specific requirements and operational maturity.

Second, there is a clear evolution from traditional MLOps to specialized LLMOps and emerging AgentOps practices [1], [7]. This evolution reflects the increasing complexity of AI systems and the need for operational frameworks that address distinctive characteristics of generative and agentic AI.

Third, integration emerges as a critical theme, with several frameworks emphasizing the need to integrate data operations, model operations, and application operations [4], [13]. Siloed approaches create gaps in observability and challenges in maintaining end-to-end system reliability.

Fourth, governance and compliance capabilities vary significantly across frameworks, with some providing comprehensive governance frameworks [6], [19], [22] while others focus primarily on technical operations. Organizations in regulated industries require frameworks with robust governance capabilities integrated throughout the operational lifecycle.

Fifth, automation levels differ substantially, ranging from frameworks emphasizing manual processes with tool support to advanced frameworks proposing autonomous operations with AI-augmented decision-making [8], [29]. Higher automation levels can improve efficiency and reliability but require careful attention to oversight and control mechanisms.

8 Discussion

8.1 Key Findings and Insights

Our review reveals several critical findings regarding operational frameworks for deploying generative AI systems at scale. First, there is clear consensus that traditional MLOps practices are insufficient for generative AI, necessitating specialized LLMOps approaches that address unique challenges including non-deterministic outputs, prompt management, hallucination detection, and token cost optimization [1], [2], [3]. The evolution from MLOps to LLMOps to AgentOps reflects the increasing complexity and autonomy of AI systems.

Second, prompt engineering has emerged as a critical operational discipline requiring specialized management practices comparable to traditional software engineering [2], [10], [18]. However, current tools and practices for prompt versioning, testing, and deployment remain immature compared to established software engineering practices. This represents a significant gap requiring further research and tool development.

Third, evaluation and monitoring of generative AI systems require fundamentally different approaches compared to traditional ML models [14], [15], [16], [21]. Semantic quality assessment, hallucination detection, and user satisfaction tracking are essential but challenging to automate, often requiring human-in-the-loop validation. Organizations must invest in building comprehensive evaluation frameworks that combine automated metrics with human judgment.

Fourth, cost optimization emerges as a critical operational concern due to token-based pricing models and substantial computational requirements of LLMs [5], [20]. Effective cost management requires multi-faceted approaches including prompt optimization, caching, model selection, and comprehensive monitoring. Organizations must balance cost optimization against quality requirements and user experience.

Fifth, governance and compliance frameworks are essential for enterprise adoption, particularly in regulated industries [6], [19], [22]. However, governance frameworks must balance risk management with operational efficiency, avoiding overly restrictive processes that impede innovation while ensuring adequate oversight and control.

Sixth, the integration of DevOps, MLOps, and LLMOps practices is essential for end-to-end system reliability [4], [13], [17], [29]. Siloed approaches create gaps in observability, challenges in maintaining consistency, and inefficiencies in operations. Organizations require unified frameworks that span infrastructure, data, models, and applications.

8.2 Convergence Patterns

The convergence of DevOps, MLOps, and LLMOps practices follows several distinct patterns. First, there is vertical integration across the stack, from infrastructure through data and models to applications [13], [14], [29]. Cloud-native approaches enable this integration through containerization, orchestration, and unified observability platforms.

Second, there is horizontal integration across the lifecycle, from development through testing, deployment, monitoring, and maintenance [1], [3], [17]. CI/CD pipelines are being extended to handle AI-specific artifacts including prompts, models, and evaluation datasets, enabling continuous delivery of AI systems.

Third, there is functional integration across operational domains including data operations, model operations, and application operations [4], [13]. This integration is essential for maintaining end-to-end system reliability and enabling rapid iteration and improvement.

Fourth, there is organizational integration, breaking down silos between data teams, ML teams, and application teams [18], [20]. Successful generative AI operations require close collaboration across these teams, necessitating shared tools, processes, and metrics.

8.3 Maturity Levels and Adoption Challenges

Operational maturity for generative AI systems varies significantly across organizations. Stone et al. [28] identify multiple maturity stages, from basic infrastructure deployment to advanced autonomous operations with comprehensive governance. Most organizations remain in early maturity stages, struggling with fundamental challenges in monitoring, evaluation, and governance.

Key adoption challenges include the lack of standardized tools and practices, requiring organizations to build custom solutions [3], [11]. The rapid evolution of the field creates challenges in keeping practices current and avoiding technical debt. The shortage of skilled practitioners with expertise spanning DevOps, MLOps, and LLMOps creates staffing challenges [28]. The substantial costs of LLM operations create budget pressures, particularly for experimentation and iteration [20]. Finally, governance and compliance requirements create barriers to adoption, particularly in regulated industries [6], [19].

8.4 Trade-offs and Design Decisions

Operational frameworks for generative AI systems involve several critical trade-offs. First, there is a trade-off between automation and control, with higher automation improving efficiency but potentially reducing oversight and increasing risks [8], [29]. Organizations must carefully balance automation with appropriate human oversight, particularly for high-stakes applications.

Second, there is a trade-off between cost and quality, with higher-quality models and more comprehensive evaluation increasing costs [5], [20]. Organizations must make explicit decisions about acceptable cost-quality trade-offs based on application requirements and business value.

Third, there is a trade-off between flexibility and standardization, with standardization improving efficiency and reliability but potentially constraining innovation [3], [11]. Organizations must balance the benefits of standardized practices against the need for flexibility to experiment and innovate.

Fourth, there is a trade-off between comprehensive governance and operational agility, with stricter governance improving risk management but potentially slowing deployment and iteration [6], [19]. Organizations must design governance frameworks that provide adequate oversight without creating excessive friction.

8.5 Limitations of Current Approaches

Current operational frameworks exhibit several significant limitations. First, most frameworks lack comprehensive standardization, requiring organizations to integrate multiple tools and practices [3], [11]. This creates complexity, increases operational overhead, and makes it difficult to share best practices across organizations.

Second, evaluation and monitoring capabilities remain immature, particularly for semantic quality assessment and hallucination detection [14], [16], [21]. Current approaches often rely heavily on human evaluation, which is expensive and difficult to scale.

Third, prompt management practices are underdeveloped compared to traditional software engineering practices [2], [10], [18]. Existing tools provide basic versioning but lack sophisticated capabilities for testing, optimization, and deployment of prompts.

Fourth, cost optimization approaches are often reactive rather than proactive, with organizations discovering cost issues after deployment rather than preventing them through design [5], [20]. More sophisticated cost modeling and optimization tools are needed.

Fifth, governance frameworks are often bolted on rather than integrated throughout the operational lifecycle [6], [19]. This creates gaps in compliance, increases operational overhead, and makes governance feel like a burden rather than an enabler.

OPEN ACCESS JOURNAL

9 Research Gaps

Based on our comprehensive review, we identify several critical research gaps requiring further investigation.

9.1 Standardization and Interoperability

The lack of standardized operational frameworks, tools, and practices creates significant challenges for organizations deploying generative AI systems [3], [11]. Research is needed to develop standardized approaches for prompt versioning and management, evaluation metrics and methodologies, monitoring and observability practices, governance frameworks and compliance mechanisms, and tool interoperability and integration. Standardization efforts should balance the need for consistency with the flexibility required for innovation and experimentation.

9.2 Automated Evaluation and Quality Assurance

Current evaluation approaches for generative AI systems rely heavily on human judgment, which is expensive and difficult to scale [14], [16], [21]. Research is needed to develop automated evaluation methods that can reliably assess semantic quality, factual accuracy, relevance, coherence, safety, and bias. Specific research directions include developing robust automated metrics that correlate well with human judgment, creating efficient human-in-the-loop evaluation frameworks that minimize human effort while

maintaining quality, designing hallucination detection methods that can identify factually incorrect outputs with high precision and recall, and building continuous evaluation pipelines that can monitor quality over time and detect degradation.

9.3 Prompt Engineering and Management

Prompt engineering has emerged as a critical operational discipline, yet tools and practices remain immature [2], [10], [18]. Research is needed to develop comprehensive prompt management frameworks including version control systems designed specifically for prompts, automated testing frameworks that can validate prompt behavior across diverse inputs, optimization techniques that can systematically improve prompt performance, and deployment practices that enable safe rollout and rollback of prompt changes. Additionally, research is needed on prompt composition and modularity, enabling reuse and systematic construction of complex prompts.

9.4 Cost Modeling and Optimization

Token-based pricing models and substantial computational requirements make cost optimization critical for generative AI operations [5], [20]. Research is needed to develop predictive cost models that can estimate costs before deployment, optimization techniques that can reduce costs while maintaining quality requirements, cost-aware architectures that make explicit trade-offs between cost and performance, and monitoring and alerting systems that provide real-time visibility into costs and enable rapid response to cost anomalies. Research should also address cost allocation and chargeback mechanisms for shared LLM infrastructure.

9.5 Observability and Debugging

Monitoring and debugging generative AI systems, particularly multi-agent systems, remains challenging [14], [25], [28]. Research is needed to develop comprehensive observability frameworks that provide visibility into system behavior at multiple levels of abstraction, debugging tools that can help operators understand and diagnose failures in complex AI systems, distributed tracing approaches that can track requests across multiple models and agents, and root cause analysis techniques that can identify the underlying causes of quality degradation or failures. Research should also address observability for emergent behaviors in multi-agent systems.

9.6 Governance and Compliance Automation

Current governance frameworks often require substantial manual effort, creating operational overhead and slowing deployment [6], [19], [22]. Research is needed to develop automated governance mechanisms that can enforce policies with minimal manual intervention, compliance verification tools that can automatically check systems against regulatory requirements, audit trail generation that captures all relevant information for compliance verification, and risk assessment frameworks that can systematically evaluate risks associated with generative AI deployments. Research should also address governance for autonomous and agentic AI systems.

9.7 Security and Adversarial Robustness

Generative AI systems face unique security challenges including prompt injection, adversarial manipulation, and harmful content generation [11], [16], [26]. Research is needed to develop robust defenses against prompt injection attacks, techniques for detecting and preventing adversarial manipulation of model behavior, output filtering methods that can reliably detect and block harmful content, and security frameworks that integrate protection mechanisms throughout the operational lifecycle. Research should also address security for multi-agent systems and autonomous AI.

9.8 RAG Operations and Knowledge Management

RAG systems introduce additional operational complexity requiring specialized approaches [16], [23]. Research is needed to develop efficient knowledge base management practices including continuous ingestion and updating, retrieval optimization techniques that balance latency and quality, end-to-end monitoring frameworks that assess both retrieval and generation quality, and knowledge base versioning and governance mechanisms. Research should also address scaling RAG systems to handle large knowledge bases and high query volumes.

9.9 Multi-Agent Operations and Coordination

The emergence of agentic AI systems introduces new operational challenges [1], [7], [14]. Research is needed to develop frameworks for monitoring and debugging multi-agent interactions, coordination mechanisms that ensure consistent and reliable system behavior, lifecycle management approaches for autonomous agents, and governance frameworks that provide appropriate oversight for autonomous systems. Research should also address emergent behaviors in multi-agent systems and techniques for ensuring system reliability despite agent autonomy.

9.10 Human-AI Collaboration in Operations

Effective generative AI operations require appropriate human-AI collaboration, balancing automation with human oversight [15], [29]. Research is needed to develop frameworks for determining appropriate levels of automation and human involvement, interfaces that enable effective human oversight and intervention, escalation mechanisms that route decisions to humans when appropriate, and feedback loops that enable systems to learn from human decisions and improve over time. Research should also address trust calibration, helping operators develop appropriate trust in AI systems.

10 Future Research Directions

Based on identified research gaps and emerging trends, we propose several promising directions for future research.

10.1 Unified Operational Frameworks

Future research should focus on developing unified operational frameworks that integrate DevOps, MLOps, and LLMOps practices into cohesive approaches [1], [4], [17]. Such frameworks should provide end-to-end coverage from development through production, support multiple deployment patterns including cloud, edge, and hybrid, enable seamless integration of traditional software, ML models, and generative AI, and provide consistent observability, governance, and security across all components. Research should also address framework extensibility, enabling organizations to customize and extend frameworks to meet specific requirements.

10.2 Intelligent Automation and Self-Healing Systems

The application of AI to improve AI operations represents a promising direction [8], [15], [29]. Future research should explore using LLMs to automate operational tasks including log analysis, incident response, and optimization, developing self-healing systems that can automatically detect and remediate failures, creating adaptive systems that continuously learn and improve operational practices, and building autonomous operations frameworks that minimize human intervention while maintaining appropriate oversight. Research should carefully address the challenges of ensuring reliability and safety of AI-driven operational decisions.

10.3 Advanced Evaluation Methodologies

Future research should develop more sophisticated evaluation methodologies for generative AI systems [14], [16], [21]. Promising directions include multi-dimensional evaluation frameworks that assess quality across multiple axes, context-aware evaluation that adapts assessment criteria based on application requirements, continuous evaluation pipelines that provide real-time quality monitoring, and efficient human-in-the-loop evaluation that minimizes human effort while maintaining quality. Research should also address evaluation of multi-agent systems and emergent behaviors.

10.4 Prompt Engineering Science

Prompt engineering should evolve from an art to a science with systematic methodologies and tools [2], [10], [18]. Future research should develop formal models of prompt behavior and effectiveness, systematic optimization techniques that can improve prompts based on performance data, composition frameworks that enable building complex prompts from reusable components, and testing methodologies that can validate prompt behavior across diverse scenarios. Research should also address prompt portability across different models and prompt adaptation to changing requirements.

10.5 Cost-Aware Architectures and Optimization

Future research should develop architectures and techniques that explicitly optimize for cost-performance trade-offs [5], [20]. Promising directions include cost-aware model selection that chooses appropriate models based on requirements and budgets, adaptive quality levels that adjust output quality based on cost constraints, intelligent caching and reuse strategies that minimize redundant computation, and cost-optimal resource allocation that balances multiple objectives. Research should also address cost prediction and budgeting for generative AI systems.

10.6 Federated and Privacy-Preserving Operations

As privacy concerns grow, research is needed on operational frameworks for federated and privacy-preserving generative AI systems [26]. Future research should explore federated learning approaches for LLMs that enable training without centralizing data, privacy-preserving inference techniques that protect sensitive information in prompts and outputs, secure multi-party computation for collaborative AI operations, and differential privacy mechanisms that provide formal privacy guarantees. Research should address the operational challenges of deploying and managing privacy-preserving systems at scale.

10.7 Edge and Hybrid Deployments

Future research should address operational frameworks for edge and hybrid deployments of generative AI systems [13], [20]. Promising directions include model compression and optimization techniques that enable deployment on resource-constrained devices, hybrid architectures that intelligently distribute computation between edge and cloud, synchronization mechanisms that

maintain consistency across distributed deployments, and monitoring frameworks that provide visibility across edge and cloud components. Research should also address offline operation and graceful degradation when connectivity is limited.

10.8 Domain-Specific Operational Frameworks

Different domains and industries have unique operational requirements necessitating specialized frameworks [6], [19], [22]. Future research should develop domain-specific frameworks for regulated industries including healthcare, finance, and government, safety-critical applications requiring high reliability and formal verification, real-time systems with strict latency requirements, and resource-constrained environments with limited computational resources. Research should identify common patterns that can be generalized while accommodating domain-specific requirements.

10.9 Sustainability and Environmental Impact

The substantial computational requirements of generative AI systems raise sustainability concerns [5], [20]. Future research should develop frameworks for measuring and optimizing environmental impact of AI operations, energy-efficient architectures and algorithms that reduce computational requirements, carbon-aware scheduling that shifts computation to times and locations with cleaner energy, and sustainability metrics that enable organizations to track and improve environmental performance. Research should balance sustainability with other operational objectives including cost, quality, and latency.

10.10 Operational Maturity Models

Future research should develop comprehensive maturity models for generative AI operations [28]. Such models should define clear maturity stages with specific capabilities and practices at each stage, provide assessment frameworks that enable organizations to evaluate their current maturity, offer roadmaps for progressing through maturity stages, and identify best practices and common pitfalls at each stage. Maturity models should address technical, organizational, and cultural dimensions of operational excellence.

11 Conclusion

This literature review has provided a comprehensive examination of operational frameworks for deploying generative AI systems at scale, analyzing 30 highly relevant papers published between 2024 and 2025. Our review reveals a clear evolution from traditional DevOps through MLOps to specialized LLMOps practices, with emerging AgentOps addressing the unique complexities of multi-agent systems. The convergence of these operational disciplines reflects the growing recognition that deploying generative AI at scale requires integrated frameworks spanning infrastructure, data, models, and applications.

Key findings from our review include the critical importance of prompt engineering and management as a new operational discipline, the need for fundamentally different evaluation and monitoring approaches compared to traditional ML systems, the emergence of cost optimization as a critical operational concern due to token-based pricing models, the essential role of governance and compliance frameworks for enterprise adoption, and the necessity of integrating DevOps, MLOps, and LLMOps practices for end-to-end system reliability.

We have identified significant research gaps including the lack of standardized operational frameworks and tools, immature evaluation and quality assurance capabilities, underdeveloped prompt management practices, reactive rather than proactive cost optimization approaches, and governance frameworks that are often bolted on rather than integrated. Addressing these gaps requires sustained research effort across multiple dimensions including standardization, automation, evaluation methodologies, cost optimization, observability, governance, security, and human-AI collaboration.

Future research directions include developing unified operational frameworks that integrate DevOps, MLOps, and LLMOps practices, advancing intelligent automation and self-healing systems, creating sophisticated evaluation methodologies, establishing prompt engineering as a systematic discipline, designing cost-aware architectures, enabling federated and privacy-preserving operations, supporting edge and hybrid deployments, developing domain-specific frameworks, addressing sustainability concerns, and creating comprehensive maturity models.

The operational frameworks and practices for generative AI systems are still evolving rapidly. While foundational frameworks exist, substantial work remains in developing standardized approaches, automated tools, and robust governance mechanisms for enterprise-scale deployments. Success in operationalizing generative AI at scale will require continued collaboration between researchers and practitioners, sharing of best practices, development of open standards and tools, and sustained investment in building operational capabilities. As generative AI systems become increasingly central to enterprise operations, the importance of robust operational frameworks will only grow, making this a critical area for continued research and development.

References

1. Joshi et al., "LLMOps, AgentOps, and MLOps for Generative AI: A Comprehensive Review," 2025. DOI: 10.6084/m9.figshare.30447089

2. S. Madicharla, "The Evolution of LLMOps: Latest Trends and Developments," *European Journal of Computer Science and Information Technology*, 2025. DOI: 10.37745/ejcsit.2013/vol13n326674

3. Özer, "Systematic Technical Survey on LLMOps: Lifecycle, Tools, Challenges, and Emerging Practices," 2024.

4. Gadiraju et al., "DataOps Meets LLMOps: Automating Cloud-Based AI Workflows from Data Ingestion to Prompt Optimization," 2025. DOI: 10.1109/icdici66477.2025.11135202

5. Krishnamoorthy et al., "DNN-Powered MLOps Pipeline Optimization for Large Language Models: A Framework for Automated Deployment and Resource Management," 2025.

6. Majumder et al., "Enhancing AI Governance in Financial Industry through IBM watsonx.governance," 2024.

7. Bandara et al., "A Practical Guide for Designing, Developing, and Deploying Production-Grade Agentic AI Workflows," 2025. DOI: 10.48550/arxiv.2512.08769

8. Chittala, "AIOps in Action: Automating AI Deployment and Management of Large Language Models for Scalable and Ethical Operations," *International Journal For Multidisciplinary Research*, 2024. DOI: 10.36948/ijfmr.2024.v06i05.28795

9. Han, "MLOps and LLMOps engineering for deployment and evaluation of AI models," 2024.

10. Olivas et al., "Model Deployment and MLOps," 2024.

11. Mishra, "Software Engineering Challenges in the Deployment of Generative AI Models at Scale," 2025. DOI: 10.21203/rs.3.rs-6875774/v1

12. Alenezi, "From Prompt-Response to Goal-Directed Systems: The Evolution of Agentic AI Software Architecture," 2024.

13. Devarapalli et al., "Cloud-Native LLMOps Meets DataOps: A Unified Framework for High-Volume Analytical Systems," 2025. DOI: 10.1109/icctdc64446.2025.11158069

14. Parthasarathy et al., "Engineering LLM Powered Multi-Agent Framework for Autonomous CloudOps," 2025. DOI: 10.1109/cain66642.2025.00031

15. Bhattarai, "Scaling Generative AI for Self-Healing DevOps Pipelines: Technical Analysis," 2025. DOI: 10.20944/preprints202506.1436.v1

16. Kalluru et al., "NLPOps: A Comprehensive Framework for Secure Development and Scalable Deployment of Multifaceted LLMs in Generative AI," 2024. DOI: 10.5121/csit.2024.140806

17. Joshi, "A Review of Generative AI and DevOps Pipelines: CI/CD, Agentic Automation, MLOps Integration, and Large Language Models," 2025. DOI: 10.20944/preprints202506.1040.v1

18. Hassan et al., "From Tokens to Tactics: Operationalizing Generative AI in Enterprise Workflows," 2025. DOI: 10.1109/iciteics64870.2025.11340834

19. P. Manjunath, R. Soman and D. P. Gajkumar Shah, "IoT and Block Chain driven Intelligent Transportation System," 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018, pp. 290-293, doi: 10.1109/ICGCIoT.2018.8753007

20. Nadiminty, "Building Infrastructure for Generative AI Workloads: Lessons from the Field," *Journal of Information Systems Engineering and Management*, 2025. DOI: 10.52783/jisem.v10i58s.12749

21. Kohl et al., "Generative AI Toolkit – a framework for increasing the quality of LLM-based applications over their whole life cycle," 2024. DOI: 10.48550/arxiv.2412.14215

22. "Llms in the cloud: best practices for scaling generative ai in regulated industries," 2025. DOI: 10.5281/zenodo.15454861

23. Prabhune et al., "Deploying Large Language Models With Retrieval Augmented Generation," 2024. DOI: 10.48550/arxiv.2411.11895

24. Fitsilis et al., "DOLLmC: DevOps for Large Language model Customization," 2024. DOI: 10.48550/arxiv.2405.11581

25. Ueno et al., "GenAIOps for GenAI Model-Agility," *arXiv.org*, 2024. DOI: 10.48550/arxiv.2502.17440

26. P. Manjunath and P. G. Shah, "IoT Based Food Wastage Management System," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 93-96, doi: 10.1109/I-SMAC47947.2019.9032553

27. Sarma, "AIOps: Evolution and Transformation of IT Operations," 2024.

28. Stone et al., "Navigating MLOps: Insights into Maturity, Lifecycle, Tools, and Careers," 2025. DOI: 10.1109/CAI64502.2025.00118

29. Brahmandam et al., "AI-Augmented DevOps: Autonomous Software Delivery with Large Language Models," *International Journal of Innovative Research in Engineering & Multidisciplinary Physical Sciences*, 2025. DOI: 10.37082/ijirmps.v13.i3.232448

30. P. Manjunath, M. Prakruthi and P. Gajkumar Shah, "IoT Driven with Big Data Analytics and Block Chain Application Scenarios," 2018 Second International Conference on Green Computing and Internet of Things (ICGCIoT), Bangalore, India, 2018, pp. 569-572, doi: 10.1109/ICGCIoT.2018.8752973

31. D. Sculley et al., "Hidden technical debt in machine learning systems," in *Advances in Neural Information Processing Systems*, 2015, pp. 2503–2511.

32. A. Paleyes, R. G. Urma, and N. D. Lawrence, "Challenges in deploying machine learning: a survey of case studies," *ACM Computing Surveys*, vol. 55, no. 6, pp. 1–29, 2022.

33. D. Kreuzberger, N. Kühl, and S. Hirschl, "Machine learning operations (MLOps): Overview, definition, and architecture," *IEEE Access*, vol. 11, pp. 31866–31879, 2023.

34. S. Amershi et al., "Software engineering for machine learning: A case study," in *2019 IEEE/ACM 41st International Conference on Software Engineering: Software Engineering in Practice (ICSE-SEIP)*, 2019, pp. 291–300.

35. E. Breck, S. Cai, E. Nielsen, M. Salib, and D. Sculley, "The ML test score: A rubric for ML production readiness and technical debt reduction," in *2017 IEEE International Conference on Big Data*, 2017, pp. 1123–1132.

36. N. Polyzotis, S. Roy, S. E. Whang, and M. Zinkevich, "Data management challenges in production machine learning," in *Proceedings of the 2017 ACM International Conference on Management of Data*, 2017, pp. 1723–1726.

37. L. E. Lwakatare, A. Raj, J. Bosch, H. H. Olsson, and I. Crnkovic, "A taxonomy of software engineering challenges for machine learning systems: An empirical investigation," in *International Conference on Agile Software Development*, 2019, pp. 227–243.

38. J. M. Zhang, M. Harman, L. Ma, and Y. Liu, "Machine learning testing: Survey, landscapes and horizons," *IEEE Transactions on Software Engineering*, vol. 48, no. 1, pp. 1–36, 2022.

39. C. Renggli et al., "Continuous integration of machine learning models with ease.ml/ci: Towards a rigorous yet practical treatment," *arXiv preprint arXiv:2106.03499*, 2021.

40. M. M. John, H. H. Olsson, and J. Bosch, "Towards MLOps: A framework and maturity model," in *2021 47th Euromicro Conference on Software Engineering and Advanced Applications (SEAA)*, 2021, pp. 1–8.

41. Manjunath, P., Herrmann, M., Sen, H. (2019). Implementation of Blockchain Data Obfuscation. In: Chen, YW., Zimmermann, A., Howlett, R., Jain, L. (eds) Innovation in Medicine and Healthcare Systems, and Multimedia. Smart Innovation, Systems and Technologies, vol 145. Springer, Singapore.

42. S. Shankar et al., "Operationalizing machine learning: An interview study," *arXiv preprint arXiv:2209.09125*, 2022.

43. I. Karamitsos, S. Albarhami, and C. Apostolopoulos, "Applying DevOps practices of continuous automation for machine learning," *Information*, vol. 11, no. 7, p. 363, 2020.

44. S. Mäkinen, H. Skogström, E. Laaksonen, and T. Mikkonen, "Who needs MLOps: What data scientists seek to accomplish and how can MLOps help?" in *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*, 2021, pp. 109–112.

45. G. Symeonidis, E. Nerantzis, A. Kazakis, and G. A. Papakostas, "MLOps-definitions, tools and challenges," in *2022 IEEE 12th Annual Computing and Communication Workshop and Conference (CCWC)*, 2022, pp. 0453–0460.

46. P. Ruf, M. Madan, C. Reich, and D. Ould-Abdeslam, "Demystifying MLOps and presenting a recipe for the selection of open-source tools," *Applied Sciences*, vol. 11, no. 19, p. 8861, 2021.

47. T. Granlund, A. Kopponen, V. Stirbu, L. Myllyaho, and T. Mikkonen, "MLOps challenges in multi-organization setup: Experiences from two real-world cases," in *2021 IEEE/ACM 1st Workshop on AI Engineering-Software Engineering for AI (WAIN)*, 2021, pp. 82–88.

48. P. Manjunath and P. Gajkumar, "IoT Based Handy Fuel Flow Measurement," 2019 Third International conference on I-SMAC (IoT in Social, Mobile, Analytics and Cloud) (I-SMAC), Palladam, India, 2019, pp. 80-83, doi: 10.1109/I-SMAC47947.2019.9032467.

49. K. Karlapalem and M. Agarwal, "Challenges in operationalizing ML models: An industry perspective," in *2021 IEEE International Conference on Big Data*, 2021, pp. 5590–5595.

50. G. Giray, "A software engineering perspective on engineering machine learning systems: State of the art and challenges," *Journal of Systems and Software*, vol. 180, p. 111031, 2021.

51. W. X. Zhao et al., "A survey of large language models," *arXiv preprint arXiv:2303.18223*, 2023.

52. Y. Chang et al., "A survey on evaluation of large language models," *ACM Transactions on Intelligent Systems and Technology*, 2023.

53. P. Liu et al., "Pre-train, prompt, and predict: A systematic survey of prompting methods in natural language processing," *ACM Computing Surveys*, vol. 55, no. 9, pp. 1–35, 2023.

54. J. Wei et al., "Emergent abilities of large language models," *Transactions on Machine Learning Research*, 2022.

55. T. Brown et al., "Language models are few-shot learners," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 1877–1901.

56. L. Ouyang et al., "Training language models to follow instructions with human feedback," in *Advances in Neural Information Processing Systems*, vol. 35, 2022, pp. 27730–27744.

57. Z. Ji et al., "Survey of hallucination in natural language generation," *ACM Computing Surveys*, vol. 55, no. 12, pp. 1–38, 2023.

58. P. Lewis et al., "Retrieval-augmented generation for knowledge-intensive NLP tasks," in *Advances in Neural Information Processing Systems*, vol. 33, 2020, pp. 9459–9474.

59. Y. Gao et al., "Retrieval-augmented generation for large language models: A survey," *arXiv preprint arXiv:2312.10997*, 2023.

60. L. Wang et al., "A survey on large language model based autonomous agents," *arXiv preprint arXiv:2308.11432*, 2023.