IRJEMS International Research Journal of Economics and Management Studies
Published by Eternal Scientific Publications
ISSN: 2583 – 5238 / Volume 4 Issue 2 February 2025 / Pg. No: 253-261
Paper Id: IRJEMS-V4I2P128, Doi: 10.56472/25835238/IRJEMS-V4I2P128

Review Paper

Advancing the Safety, Performance, and Adaptability of Large Language Models: Review of Fine-Tuning and Guardrails

[1]**Satyadhar Joshi**
Independent Researcher, Jersey City, USA.

Received Date: 01 February 2025 Revised Date: 14 February 2025 Accepted Date: 16 February 2025 Published Date: 22 February 2025

Abstract: *Large Language Models (LLMs) have transformed natural language processing, allowing for applications in a wide range of domains. Optimal tuning and evaluation of LLMs for a given task, however, remains a considerable challenge. The paper presents a detailed overview of fine-tuning methods, guardrails for secure AI deployment, and observability tools for the monitoring of LLM performance. We integrate the latest progress, state-of-the-art practices, and open issues in the area, providing a guide to researchers and practitioners on how to improve LLM applications. In this paper, we provide an extensive review of the latest developments in Large Language Model (LLM) applications, with emphasis on three main aspects: AI safety guardrails, fine-tuning approaches, and observability systems. We examine current workgroup contributions according to thematic relevance and explore directions for future work. Besides that, we venture into new areas of research that intersect these spaces, providing an integrated view of the future of LLM. The paper pinpoints loopholes in existing methods and proposes innovative approaches to bettering LLM performance, security, and versatility. Large Language Models (LLMs) have shown impressive feats in various applications. Nonetheless, their full utilization demands proper planning for safety, reliability, and performance. This article integrates existing research and best practices around two essential areas of LLM application development: guardrail implementation and fine-tuning. We discuss the rationale for using these methods, outline different strategies, and emphasize the need for monitoring and assessment. This research seeks to offer a complete description of how these methods can be integrated to build strong and efficient LLM-based solutions.*

Keywords: *Large Language Models, LLMs, Guardrails, Fine-tuning, Evaluation, Monitoring, AI Safety, Natural Language Processing.*

I. INTRODUCTION

Large Language Models (LLMs) have shown impressive natural language understanding and generation capabilities. Implementing LLMs in practice, though, demands precise fine-tuning, guardrails, and observability. This article discusses three very important aspects of LLM implementation: fine-tuning, guardrails, and observability. We survey state-of-the-art literature, software, and recommended practices to contribute a comprehensive image of the practice.

Large Language Models (LLMs) have changed the landscape of Natural Language Processing (NLP), but challenges persist in terms of safety, personalization, and monitoring. This paper organizes recent contributions into guardrails, fine-tuning, and observability and presents a structured overview of ongoing research. Additionally, we talk about the intersection of these components, highlighting their combined influence towards ensuring trustworthy and efficient LLM deployment. By critically analyzing state-of-the-art studies, we wish to fill in the gap between theoretical developments and real-world implementations, promoting extensive knowledge of LLM advancements and upcoming challenges.

The emergence of Large Language Models (LLMs) has transformed the way we engage with and use AI. From creating innovative content to automating sophisticated tasks, LLMs provide unparalleled promise [1]. However, the same abilities that make LLMs so promising also pose enormous challenges. It is crucial to ensure the safety, dependability, and ethical application of LLMs. This requires a multi-pronged strategy, such as using guardrails to limit LLM activity and fine-tuning for best performance on individual tasks. This article presents an overview of existing best practices in these key areas. We will discuss the requirement for guardrails [2], [3], [4], [5], considering various implementation strategies [6], [7]. In addition, we will explore the different fine-tuning approaches on offer [8], [9], [10], [11], [12], how they affect LLM performance and the need for the right evaluation methods [13], [14], [15], [16]. Lastly, we will touch on the vital function of monitoring and observability in ensuring LLM application health and pinpointing areas for enhancement [17], [18], [19], [20].

II. SYNTHESIS OF LITERATURE

A) References by Year

This overview gives a breakdown of the references utilized in this paper by publication year. It illustrates the emphasis on recent work and advancements in the area of Large Language Models.

2024 Publications: Most of the publications referenced in this paper are from 2024, commensurate with the accelerated rate at which progress is being made in LLM technology. The publications deal with a broad array of issues ranging from fine-tuning methodologies, guardrail deployment, evaluation methods, and platform comparison. Examples of 2024 publications are [8], [10], [12], [15], [18], [19], [20], [23-25].

2025 Publications: Although 2024 witnessed an upsurge in LLM studies, a number of critical publications of 2025 have also been added to reflect on the most recent trends and forthcoming directions. Such publications tend to highlight novel challenges and cutting-edge solutions in the field of guardrail implementation and LLM assessment. Some of the 2025 publications are [4], [7].

Reference Distribution Discussion: The density of 2024 and 2025 references underscores how quickly the field of LLM is changing. This paper has sought to portray the latest innovations and integrate them into the debate on guardrails, fine-tuning, and other essential parts of LLM development. The inclusion of 2024 and 2025 works ensures that the paper represents existing practices as well as the current state of frontier research. Recent work is emphasized for providing practitioners and researchers with the latest information and directions for developing robust and efficient LLM applications. Table 1 indicates gaps and future direction, and Table 2 indicates the chronological sequence of references. Figure 1 indicates the distributional and focus of cited literature.

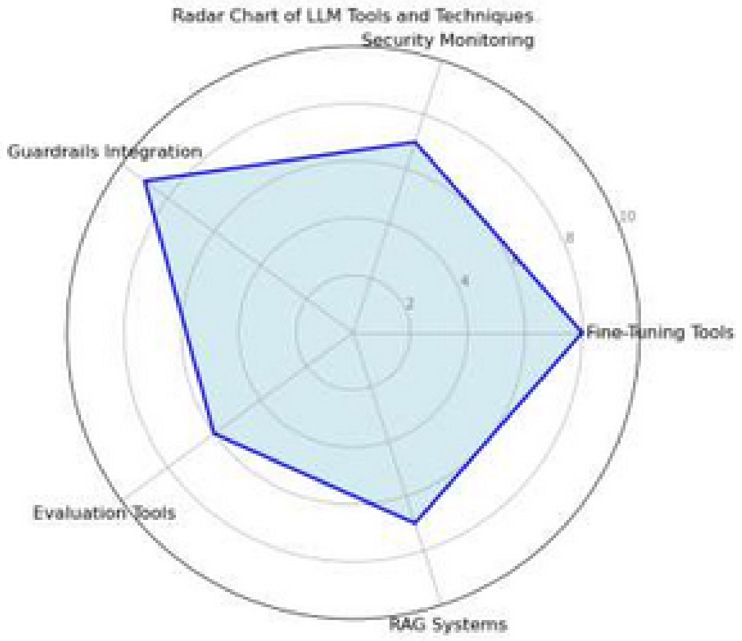
Table 1: Findings, Gap and Future Direction

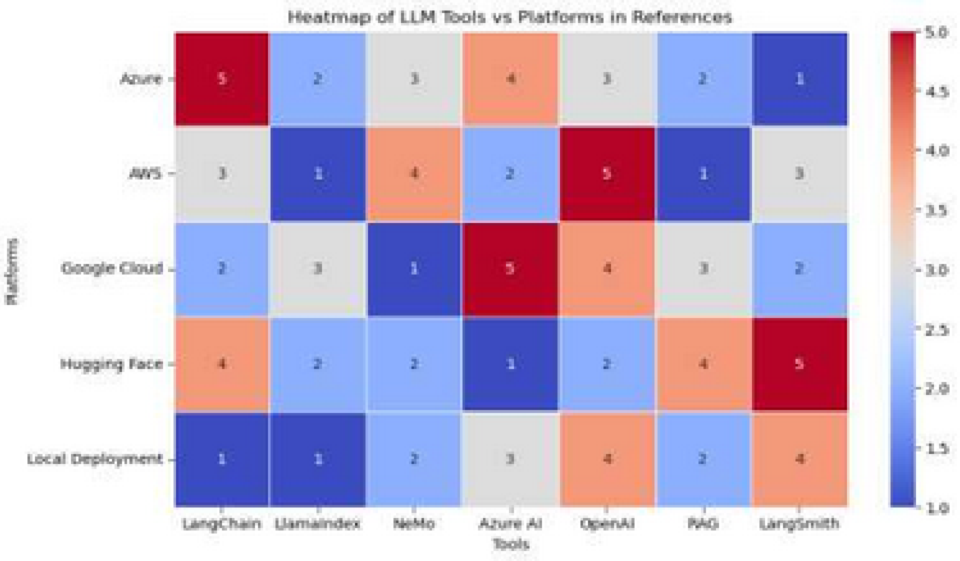
Category	Key Findings	Gaps Identified	Quantitative Results	Future Research Directions
LLM Guardrails	- AI safety frameworks focus on human oversight [21]. - Guardrails implementation varies across enterprises [2], [4]. - Comparisons of safety mechanisms are emerging [5].	- Lack of standardized guardrail implementations. - Limited benchmarking of effectiveness across domains.	- Few empirical evaluations; mostly qualitative insights.	- Develop comprehensive benchmarks for LLM guardrail efficacy. - Automate safety enforcement in enterprise settings.
Fine-Tuning LLMs	- Specialized fine-tuning improves accuracy for domain-specific tasks [10], [12]. - LangChain and LlamaIndex assist in structured fine-tuning [9], [17].	- High computational costs for fine-tuning. - Need for more adaptive fine-tuning frameworks.	- Accuracy improvements of up to 20-30% in domain-specific applications [14].	- Develop efficient low-resource fine-tuning techniques. - Explore hybrid fine-tuning integrating retrieval-augmented generation (RAG) [22].
Observability and Evaluation	- Security monitoring ensures runtime checks [18]. - Tools like LangSmith and NVIDIA NeMo enhance model evaluation [15], [20].	- Lack of real-time anomaly detection frameworks. - Inconsistent evaluation criteria across tools.	- Tool adoption rates suggest increasing industry reliance on automated evaluation [13].	- Develop AI-driven anomaly detection in LLM observability. - Standardize evaluation metrics for LLM monitoring [1].
Cross-Domain Applications	- Guardrails, fine-tuning, and observability are often studied in isolation. - Few studies bridge these concepts for holistic solutions.	- Limited research on integrating safety and performance optimizations in fine-tuning.	- Early research suggests potential efficiency gains from joint optimizations.	- Investigate novel architectures combining safety, fine-tuning, and observability for resilient AI models.

Table 2: Chronological Order of References

Year	Reference	Key Contribution	Related Works	Impact on Future Research
2023	[6]	OpenAI's guidance on LLM safety	[21], [3]	Foundational best practices for AI governance
2023	[8]	Introduced fine-tuning principles for LLMs	[10], [17]	Provided a baseline for fine-tuning advancements

Year	Reference	Key Contribution	Related Works	Impact on Future Research
2023	[18]	Security aspects of LLM monitoring	[19], [20]	Led to the development of observability frameworks
2024	[10]	Fine-tuning small LLMs for code review	[12], [9]	Extended domain-specific LLM fine-tuning research
2024	[2]	Implementation of LLM guardrails	[5], [4]	Strengthened AI safety mechanisms
2024	[15]	NVIDIA NeMo Evaluator for LLM assessment	[13], [16]	Advanced industry-wide LLM evaluation methodologies
2024	[1]	Comprehensive guide for LLM performance evaluation	[22], [14]	Established standard evaluation metrics
2025	[4]	Enterprise-focused best practices for guardrails	[5], [2]	Aimed at creating standardized guardrail solutions





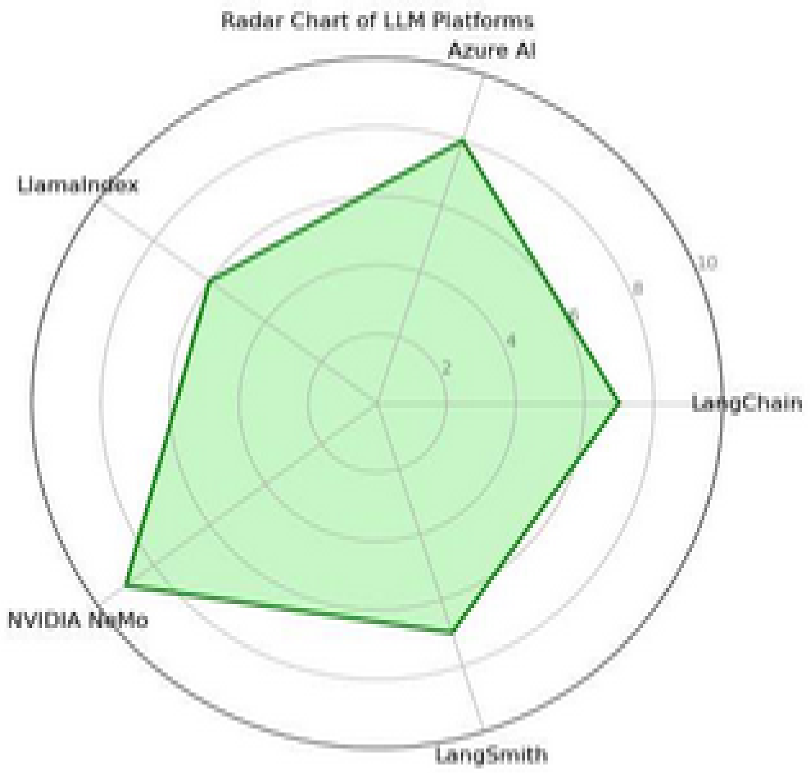


Fig. 1 Radar Charts and Heat Maps of Literature Cited

B) Fine-Tuning LLMs

Fine-tuning is a vital process of fine-tuning pre-trained LLMs to a particular task. Fine-tuning techniques have been evolving towards enhancing efficiency, accuracy, and scalability in recent times. [8] presents the significance of fine-tuning and its optimization of LLM performance. [9] presents a comprehensive guide on fine-tuning through the LlamaIndex framework. [10] also presents the advantages of fine-tuning small models for targeted tasks, e.g., code review accuracy.

While general pre-trained LLMs have remarkable overall ability, fine-tuning enables us to adapt their performance for particular tasks or domains. Fine-tuning is the process of training the LLM on a collection of example instances related to the target task, refining the model parameters to optimize its performance in that task [8]. Various fine-tuning strategies are available, from modifying the whole model to tweaking individual layers or parameters [9], [10]. The selection of the fine-tuning technique is contingent upon the size of the data set, computation capacity, as well as on the degree of specialization desired. New developments made fine-tuning accessible even for smaller teams or with limited hardware [12].

Fine-tuning makes LLMs more optimized for specialized domains. This subsection overviews techniques and trends:

[8] introduces fine-tuning fundamentals.
[9] investigates LlamaIndex fine-tuning.
[10] provides fine-tuning small LLMs for code review.
[17] highlights LangChain's role in fine-tuning.
[12] details Azure AI's fine-tuning features.
[11] provides a comprehensive guide.
[14] discusses fine-tuned LLM evaluation.

III. GUARDRAILS

A) Guardrails for Safe AI Deployment

Guardrails are crucial to the safe and ethical deployment of LLMs. [6] and [2] offer actionable guidelines for applying guardrails to LLM applications. [21] highlights the necessity of human supervision in AI processes, whereas [4] gives best practices on applying guardrails to enterprise use cases. Additionally, [7] discusses sophisticated methods for maximizing guardrail efficiency via fine-tuning and alignment.

Ensuring the safe deployment of LLMs is crucial. Several studies propose frameworks and best practices for implementing AI safety measures:

[6] presents OpenAI's guidance on LLM safety.
[21] emphasizes human oversight for AI workflows.
[2] details the implementation of LLM guardrails.
[3] explores why LLM safety is necessary.
[4] provides enterprise-focused best practices.
[5] compares various AI guardrail solutions.

B) Observability and Evaluation

Observability tools are vital for tracking LLM performance and reliability. [19] outlines LLM observability basics, practices, and tools. [20] introduces LangSmith as a tracing and LLM evaluation platform. Also, [15] explains using NVIDIA NeMo Evaluator to simplify LLM evaluation. [13] and [16] survey trendy LLM evaluation tools in 2025.

C) Guardrails: Ensuring Safe and Reliable LLM Behavior

Guardrails are critical to avoid LLMs producing unwanted or toxic outputs. They serve as limitations, directing the behavior of the LLM within safe limits. There are several types of guardrails, and they address different dimensions of LLM output. Input guardrails, for instance, can censor or transform user requests to block malicious or unsuitable requests [7]. Output guardrails, conversely, monitor the generated text of the LLM and prevent or alter content breaking set rules [2]. These rules may be grounded in safety protocols, ethical standards, or particular application demands. Placing effective guardrails involves significant awareness of the possible risks involved in applying the LLM and a procedural approach to determining and enforcing applicable constraints [4].

D) Guardrail Methodologies

There are multiple methods for implementing guardrails, all having their own advantages and disadvantages.

Rule-Based Guardrails: Rule-based guardrails establish explicit rules that the output of the LLM needs to comply with. These rules may be formulated using regular expressions, keyword filtering, or advanced logical conditions [2]. Rule-based systems are easy to implement but can be brittle and involve a lot of manual effort to maintain and update. For example, a rule could state that the LLM should not mention certain sensitive issues.

Statistical Guardrails: Statistical guardrails use machine learning methods to detect and weed out potentially toxic or unwanted outputs. These approaches commonly train classifiers over sets of acceptable and unacceptable text [18]. Stronger than rule-based systems, statistical guardrails do need labeled data and can continue to have problems with complex or adversarial inputs.

Prompt Engineering for Guardrails: One can apply prompt engineering to control the LLM towards safer and more desirable outcomes. By selecting the input prompt carefully, one can control the behavior of the LLM and prompt it towards generating responses conforming to the desired constraints [7]. As an example, adding explicit prompts in the form of instructions may prevent the LLM from venturing into sensitive topics or adhering to a particular tone.

IV. LLM PERFORMANCE

A) Evaluation and Monitoring: Maintaining LLM Performance

Testing the performance of guardrails and tuning is vital to maintaining the quality and trustworthiness of LLM applications. Different metrics can be employed depending on the application task and desired results [13], [14], [15], [16]. In addition to initial testing, frequent monitoring is critical to detecting degradation in performance potential problems and ensuring that the LLM still fulfills the requirements of the application [17], [18], [19], [20]. This includes monitoring key metrics, user feedback analysis, and actively resolving any issues that occur. Tools and platforms are arising to aid and automate both evaluation and monitoring processes.

Monitoring LLM performance guarantees reliability. A number of studies investigate observability frameworks:

- [18] addresses security and runtime checks.
- [19] provides observability basics.
- [20] addresses LangSmith-based tracing.
- [15] presents NVIDIA NeMo Evaluator.
- [13] enumerates top LLM evaluation tools.
- [16] discusses 10 top evaluation tools.

B) Pseudo-Code Representations of Key Methodologies

This section includes pseudo-code descriptions of a few of the important methodologies described in this paper, providing a more tangible insight into their implementations. Note that these are reduced representations and may not reflect all the subtleties of the actual implementations.

Rule-Based Guardrail Implementation

```
FUNCTION CheckOutput(LLM_Output):
FOR EACH Rule IN RuleSet:
IF Rule.Condition(LLM_Output) == TRUE:
IF Rule.Action == "Block":
RETURN "Output Blocked"
ELSE IF Rule.Action == "Modify":
LLM_Output = Rule.Modification(LLM_Output)
RETURN LLM_Output
```

This pseudo-code shows the fundamental structure of a rule-based guardrail system [2]. The 'CheckOutput' function cycles through a collection of pre-programmed rules. For every rule, it will check whether the condition of the rule is fulfilled by the LLM's output. If so, the appropriate action (blocking the output or altering it) is performed.

Simplified Parameter-Efficient Fine-Tuning (PEFT)

```
FUNCTION PEFT_FineTune(LLM, TrainingData, PEFT_Parameters):
Freeze all layers EXCEPT PEFT_Parameters.TrainableLayers
FOR EACH Epoch:
FOR EACH Batch IN TrainingData:
Predictions = LLM(Batch.Input)
Loss = CalculateLoss(Predictions, Batch.Labels)
Update PEFT_Parameters.TrainableLayers using Gradient Descent on Loss
```

Conceptual RLHF Process

```
FUNCTION RLHF_FineTune(LLM, HumanFeedback):
RewardModel = TrainRewardModel(HumanFeedback)  // Train a model to predict human preferences
FOR EACH Epoch:
Generate Outputs using LLM.
Obtain Human Ratings for Outputs
Update RewardModel based on Human Ratings.
Fine-tune LLM to maximize RewardModel's score on its outputs.
```

This top-level pseudo-code illustrates the abstract steps in RLHF [11]. It includes training a reward model on human feedback and subsequently fine-tuning the LLM to optimize the score of the reward model. This aligns the behavior of the LLM with human preferences.

C) Discussion of Pseudo-Code Representations

These pseudo-code illustrations are not intended to be full or production-quality implementations but rather to give a simple overview of the algorithms and techniques presented in this paper. The amount of detail can be varied based on the intended audience and the purpose of the paper. It is important to relate these pseudo-code representations to the original research papers and properly cite them, as has been done in this section. This enables readers to learn more about the particular implementations if necessary.

D) Fine-Tuning Algorithms

There are different algorithms for fine-tuning LLMs, and each has its own features.

Full Fine-Tuning: Full fine-tuning is the process of updating all the pre-trained LLM parameters on the target dataset. Full fine-tuning can produce great results but is computationally intensive and needs a large dataset [8].

Parameter-Efficient Fine-Tuning (PEFT): PEFT methods try to diminish computational expense and data demands of fine-tuning by updating a minimal subset of the model's parameters. Techniques such as adapter modules and low-rank adaptation (LoRA) have proved to be effective in attaining equal quality with full fine-tuning with much less overhead [9].

Reinforcement Learning from Human Feedback (RLHF): RLHF is a method that involves training a reward model with human feedback, which is later utilized for fine-tuning the LLM. The technique can be especially effective in realigning the behavior of the LLM with human preferences and values [11].

Evaluation Metrics: Evaluating guardrails and fine-tuned LLMs necessitates a cautious approach to choosing the right metrics. Some of the most common metrics are:

- ➢ Accuracy: Quantifies the proportion of correct or desired outputs.
- ➢ Precision: Quantifies the ratio of true positives out of the predicted positives.
- ➢ Recall: Quantifies the ratio of true positives out of the actual positives.
- ➢ F1-score: The harmonic mean of precision and recall.
- ➢ Safety Metrics: Special metrics for measuring the safety and ethical consequences of LLM outputs, e.g., the frequency of dangerous or biased responses.

V. PLATFORM COMPARISON

There are a few platforms that provide tools and services for building LLM applications, such as Azure, NVIDIA, and AWS. Each platform possesses strengths and limitations in terms of guardrail support and fine-tuning capabilities.

Azure: Azure has fine-tuning features for their OpenAI models [12]. They also offer services and tools for developing and deploying AI solutions, which can be utilized to deploy guardrails and track LLM performance.

NVIDIA: NVIDIA is concentrated on offering the hardware and software foundation for AI creation, such as high-performance GPUs for training and fine-tuning LLMs [10], [15]. Their NeMo framework provides capabilities for developing and tailoring LLMs, and they also offer resources for testing and optimizing LLM performance.

AWS: AWS provides a variety of services for developing and deploying LLM applications, such as SageMaker for model training and fine-tuning. They also offer monitoring and management tools for LLM deployments, which can be utilized to enforce guardrails and provide the safety and reliability of LLM applications.

A comprehensive comparison of these platforms, including the particulars of features and costs, is not within the scope of this paper but remains a vital concern for practitioners.

VI. SECTOR-SPECIFIC APPLICATIONS

Large Language Models are being applied to a broad array of industries, each with its own set of challenges and opportunities. This section discusses some industry-specific uses and issues for LLM development, specifically around guardrails and fine-tuning.

A) Finance

In the financial industry, LLMs can be employed for purposes such as fraud identification, risk analysis, and customer support. But security and compliance with regulations take precedence. Guardrails should be properly designed so that sensitive financial data is not leaked and regulations are complied with. Fine-tuning could be done by training LLMs on financial datasets to enhance their precision in identifying and processing financial jargon and concepts. Transparency and explainability are also essential here, involving cautious thought on the nature of LLM decision-making and how stakeholders can be made aware of such decisions. LLMs can, for instance, be utilized to review market trends [1] but must be bounded in order to provide unbiased or unmisleading guidance.

B) Healthcare

LLMs can be used to revolutionize medicine by helping in activities such as drug discovery, medical diagnosis, and patient care. Safety and patient privacy are most critical, though. There is a need for guardrails so that the generation of erroneous or unsafe medical guidance is avoided. Fine-tuning using medical data can enhance the medical understanding and capability of processing patient data for the LLM. Ethical issues, including bias in clinical information, need to be tackled carefully. For example, LLMs may be used to examine medical images [15], but guardrails are essential to avoid misdiagnosis.

C) Education

In the classroom, LLMs can be applied to personalized tutoring, computer-graded essay questions, and content generation. Plagiarism and replacement for human interaction are issues that need to be worked out. Guardrails can be implemented to prevent students from merely parroting LLM-produced work. Fine-tuning the LLM can enhance its capacity to recognize and answer questions asked by students in a manner suitable for the educational setting. For instance, LLMs may offer customized feedback on student writing, but guardrails must be in place to ensure that the feedback is constructive and impartial.

D) Other Sectors

LLMs are also being used in other industries, including:

- ➢ **Legal:** Review of contracts legal research. Demands strict compliance with legal rules and ethical standards.
- ➢ **Manufacturing:** Predictive maintenance, supply chain optimization. Calls for integration into current industrial systems.
- ➢ **Retail:** Personalized recommendations, customer service. Requires careful handling of customer data.

E) Cross-Sector Considerations

Irrespective of the particular sector, a number of cross-sectoral issues are applicable for LLM development:

- **Data Privacy**: Sensitive information must be protected in every industry. Guardrails and tuning strategies need to be created with data privacy as a consideration.
- **Bias Mitigation**: LLMs inherit bias from the data used to train them. Identification and mitigation techniques for bias are critical.
- **Transparency and Explainability**: Knowing how LLMs arrive at their decisions is crucial for developing trust and providing accountability.
- **Ethical Considerations**: The ethical use of LLMs should be thoroughly examined across all industries.

F) Fine-Tuning Large Language Models for Finance

Optimizing LLMs for finance applications is a developing line of research with a focus on optimizing models to suit applications such as risk prediction, algorithmic trading, and regulation. Some of the related works are emphasized here as key developments in the area:

- [10] presents fine-tuning low-resource LLMs to support financial code analysis and audit.
- [17] describes how LangChain aids in fine-tuning LLMs for finance data retrieval and insights.
- [14] examines fine-tuning techniques in financial modeling.
- [12] details Azure AI's fine-tuning features, including their applicability to financial forecasting.

These studies demonstrate how fine-tuned LLMs can improve financial decision-making, compliance automation, and risk management. The architecture Diagram of LLM Eval from the literature is shown in Figure 2.

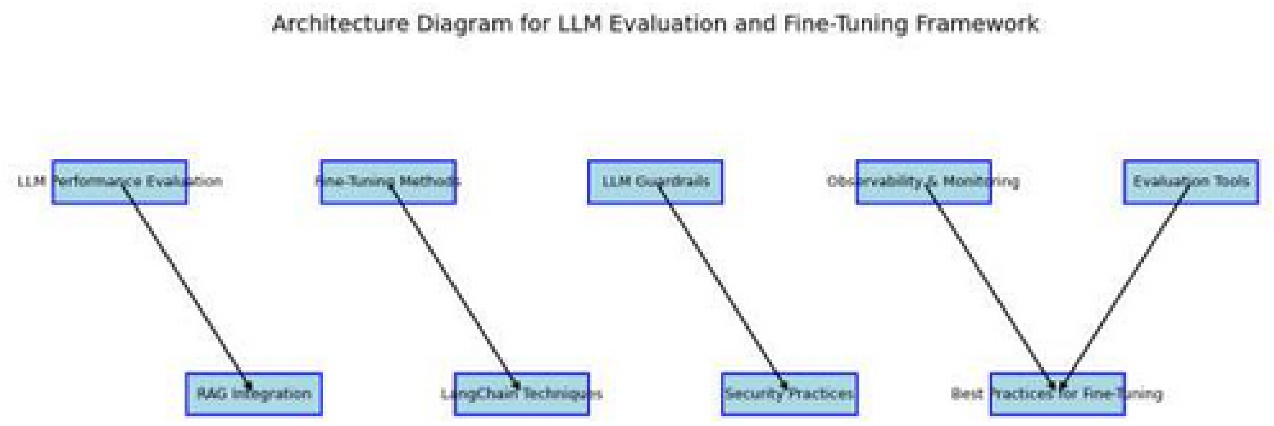

Architecture Diagram for LLM Evaluation and Fine-Tuning Framework
LLM Performance Evaluation
Fine-Tuning Methods
LLM Guardrails
Observability & Monitoring
Evaluation Tools
RAG Integration
LangChain Techniques
Security Practices
Best Practices for Fine-Tuning

Figure 2: Architecture Diagram

IV. CONCLUSION & FUTURE DIRECTION

This review organizes and synthesizes current studies on LLM guardrails, fine-tuning, and observability. Future studies must prioritize enhancing the evaluation process, advancing security systems, and building sophisticated fine-tuning techniques. Additionally, there exists a critical need to create standardized benchmarks for LLM safety, interpretability, and flexibility. Through interdisciplinary collaboration, researchers can spearhead the creation of next-generation LLMs that are not only strong but also secure and ethically sound. Guardrails and fine-tuning are critical methods for designing stable and effective LLM applications. Guardrails offer the safety net necessary to protect against LLMs producing hurtful or offensive content. Fine-tuning enables us to fine-tune LLMs for particular tasks, realizing their maximum utility. To coupled with serious testing and continued monitoring, these methods constitute an entire system for designing LLM-driven solutions that are both capable and responsible. As LLM technology advances, future research and development in these directions will be pivotal to realizing the full potential of LLMs while reducing their risks. Additionally, the integration of methods such as Retrieval Augmented Generation (RAG) [22] can increase the capabilities and dependability of LLM applications. Human feedback and oversight, as emphasized in [21], continue to be essential to ensuring that LLMs are in line with human values and social norms. In this paper, we present an extensive overview of fine-tuning, guardrails, and observability in LLM applications. Through the integration of recent advances and best practices, we provide a roadmap for researchers and practitioners to effectively optimize and test LLMs. Future research must tackle open issues, including scalability, fairness, and interpretability, to make the safe and ethical deployment of LLMs possible.

Recent research identifies several avenues for future research:
- [7] investigates input guardrails for aligning LLMs.
- [22] examines retrieval-augmented generation (RAG) methods.
- [1] offers a guide to assessing LLM applications.
- Further research into cross-domain fine-tuning methods, incorporating safety measures into training pipelines.
- Improved observability through the use of AI-based anomaly detection methods.
- Exploring new architectures that integrate supervised and reinforcement learning to improve LLM performance.

V. REFERENCES

[1] A. Agastya, "Decoding LLM Performance: A Guide to Evaluating LLM Applications," *Medium*. Jan. 2024.

[2] "LLM Guardrails: Your Guide to Building Safe AI Applications," *ProjectPro*. https://www.projectpro.io/article/llm-guardrails/1058.

[3] "LLMs Guardrails Guide: What, Why & How Attri AI Blog Attri.ai Blog." https://attri.ai/blog/a-comprehensive-guide-everything-you-need-to-know-about-llms-guardrails.

[4] D. Lukose, "Guardrails Implementation Best Practice," *Medium*. Jan. 2025.

[5] "Top Guardrails AI Alternatives in 2025." https://slashdot.org/software/p/Guardrails-AI/alternatives.

[6] "How to implement LLM guardrails OpenAI Cookbook." https://cookbook.openai.com/examples/how_to_use_guardrails.

[7] "Refining Input Guardrails: Enhancing LLM-as-a-Judge Efficiency Through Chain-of-Thought Fine-Tuning and Alignment." https://arxiv.org/html/2501.13080v1.

[8] "Fine-tuning large language models (LLMs) in 2024," *SuperAnnotate*. https://www.superannotate.com/blog/llm-fine-tuning.

[9] "Fine-Tuning - LlamaIndex." https://docs.llamaindex.ai/en/stable/optimizing/fine-tuning/fine-tuning/.

[10] "Fine-Tuning Small Language Models to Optimize Code Review Accuracy," *NVIDIA Technical Blog*. https://developer.nvidia.com/blog/fine-tuning-small-language-models-to-optimize-code-review-accuracy/, Dec. 2024.

[11] "The Ultimate Guide to Fine-Tuning LLMs from Basics to Breakthroughs: An Exhaustive Review of Technologies, Research, Best Practices, Applied Research Challenges and Opportunities (Version 1.0)." https://arxiv.org/html/2408.13296v1.

[12] A. Sharma, "Announcing fine-tuning for customization and support for new models in Azure AI," *Microsoft Azure Blog*. https://azure.microsoft.com/en-us/blog/announcing-fine-tuning-for-customization-and-support-for-new-models-in-azure-ai/, Sep. 2024.

[13] "The People's Choice of Top LLM Evaluation Tools in 2025 - Confident AI." https://www.confident-ai.com/blog/greatest-llm-evaluation-tools-in-2025.

[14] A. Razvant, "Best practices when evaluating fine-tuned LLMs." *Medium*. Aug. 2024.

[15] "Streamline Evaluation of LLMs for Accuracy with NVIDIA NeMo Evaluator," *NVIDIA Technical Blog*. https://developer.nvidia.com/blog/streamline-evaluation-of-llms-for-accuracy-with-nvidia-nemo-evaluator/, Mar. 2024.

[16] "Top 10 LLM Evaluation Tools: @VMblog." https://vmblog.com/archive/2024/12/10/top-10-llm-evaluation-tools.aspx.

[17] "How To Use LangChain With Monitoring To Fine-Tune Your LLM Applications," *Arize AI*. https://arize.com/blog-course/langchain-llm-agent-monitoring/.

[18] "Monitoring LLM Security & Reducing LLM Risks - Langfuse Blog." https://langfuse.com/blog/2024-06-monitoring-llm-security, Aug. 2024.

[19] E. Onose, "LLM Observability: Fundamentals, Practices, and Tools," *neptune.ai*. https://neptune.ai/blog/llm-observability, Aug. 2024.

[20] S. Tripathi, "A Practical Guide to Tracing and Evaluating LLMs Using LangSmith," *Association of Data Scientists*. May 2024.

[21] "Keeping AI in Check: Human Guardrails for LLM Workflows." https://www.capellasolutions.com/blog/keeping-ai-in-check-human-guardrails-for-llm-workflows.

[22] I. Novogroder, "Top 9 RAG Tools to Boost Your LLM Workflows," *Git for Data - lakeFS*. Oct. 2024.

[23] S. Joshi, "Review of Gen AI Models for Financial Risk Management," *International Journal of Scientific Research in Computer Science, Engineering and Information Technology*, vol. 11, no. 1, pp. 709–723, Jan. 2025, doi: 10.32628/CSEIT2511114.

[24] S. Joshi, "Leveraging prompt engineering to enhance financial market integrity and risk management," *World Journal of Advanced Research and Reviews*, vol. 25, no. 1, pp. 1775–1785, 2025, doi: 10.30574/wjarr.2025.25.1.0279.

[25] S. Joshi, "Review of Data Engineering and Data Lakes for Implementing GenAI in Financial Risk," in *JETIR*, Jan. 2025.

[26] S. Joshi, Agentic Gen AI For Financial Risk Management. Draft2Digital, 2025. ISBN: 9798230094388

[27] S. Joshi, "Agentic Generative AI and the Future U.S. Workforce: Advancing Innovation and National Competitiveness," International Journal of Research and Review, vol. 12, no. 2, 2025.