Med-Guard: LLM-based Diagnostic System with Safety Guardrails

Suparshva Jain* Kunwar Zaid* Amit Sangroya Lovekesh Vig
TCS Research, New Delhi, India
**equal contributions*, {suparshva.jain, kunwar.zaid, amit.sangroya, lovekesh.vig}@tcs.com

***Abstract*—Recent advances in large language models (LLMs) have significantly enhanced their capabilities across various domains, including coding, mathematical reasoning, and creative writing. Despite these improvements, the application of LLM in healthcare continues to face substantial challenges, particularly related to concerns such as explainability, safety, and trustworthiness. The emergence of agentic systems has introduced techniques aimed at improving diagnostic accuracy; however, the safety analysis of these systems remains largely unaddressed. In response to these challenges, we present Med-Guard as a novel agentic system designed to enhance the safety and reliability of diagnostic decisions in healthcare settings. Our system incorporates safety agents to ensure that diagnostic decisions are not only accurate but also safe for patient care. In addition, the system generates comprehensive diagnostic reports and logs, which enhance traceability and accountability. Furthermore, our system provides estimated drug costs for prescriptions, offering a holistic approach to patient care that considers both clinical and economic factors. This paper discusses the implementation, evaluation and implications of proposed Med-Guard system to improve the safety and efficacy of LLM applications in healthcare.**

I. Introduction

Diagnostic error is an increasingly recognized threat to public health, with estimates of 5% of adults being affected in the outpatient environment. In the hospital setting, diagnostic error is responsible for 6-17% of adverse event [1]. More recently, multi-agent frameworks based on LLMs improve clinical decision-making. AI Systems in this area assign specialized roles to agents for intent recognition, diagnostic reasoning, and treatment planning so that healthcare delivery can be both personalized and sensitive to the context. Despite the advancements in this field, AI systems based on LLM agents face several safety concerns [2].

Common Safety challenges include generating inappropriate content, harmful outputs, and amplifying social biases. Sometimes, agents can prescribe medicines that may interact adversely and affect patient's safety. Additionally, interpretability issues arise, where models lack transparency in their outputs. Privacy risks arise from models that leak sensitive information, while hallucinations may lead to false or misleading information. Traceability is also an important factor in diagnosis systems that involves documenting and connecting various stages of a diagnostic process.

These challenges highlight the urgent need for LLM-based clinical systems that prioritize safety, interpretability, and auditability by design, rather than treating them as afterthoughts. While the performance of large language models in individual diagnostic or reasoning tasks has improved, their lack of procedural guardrails and oversight mechanisms raises concerns for real-world use, particularly in high-stakes domains like medicine. Addressing these concerns requires a shift from monolithic, end-to-end generative models toward modular, agentic architectures, where roles are distributed across specialized sub-agents and decisions are subjected to rigorous internal scrutiny. Moreover, the financial implications of the proposed prescriptions are important whenever patients are leveraging any automated diagnostic agent.

In this paper, we introduce Med-Guard, a multi-agent LLM framework for diagnostic and treatment decision support that is explicitly designed to address these safety limitations. Med-Guard mirrors the structure of a multidisciplinary clinical team by assigning different responsibilities to LLM agents—including history taking, differential diagnosis generation, test ordering, prescription drafting, and safety auditing. Crucially, the system incorporates two dedicated safety agents: one for vetting diagnostic procedures, and another for auditing prescription safety using both internal reasoning and external drug interaction datasets. These agents operate within a controller-mediated architecture that enforces a strict, traceable workflow — ensuring that every critical decision is subjected to proactive checks before being surfaced to the user.

Our contributions in this work are threefold:

(1) We propose a safety-first, multi-agent architecture for LLM-driven clinical decision-making.
(2) We demonstrate empirical improvements in diagnostic accuracy, prescription safety, and price transparency across multiple language models.
(3) We offer a modular framework that enables traceability, domain-specific validation, and agentic self-correction.

By grounding decision-making in explicit safety protocols and structured inter-agent communication, Med-Guard moves closer to the standards of clinical accountability, transparency, and patient-centered care that are required for AI deployment in healthcare.

II. Related Work

The application of large language models (LLMs) in clinical decision-making has catalyzed the development of agentic systems capable of medical reasoning, diagnosis, and tool use. While substantial advances have been made in agent design

and diagnostic performance, patient safety is still treated as a peripheral concern in most systems, rather than a central architectural principle. We categorize existing techniques into three areas: Multi-Agent Diagnosis Systems, Multi-Modal Agents and Safety-Centric Architectures.

A. Multi-Agent Diagnosis Systems

Several recent works adopt a *multi-agent paradigm* to emulate clinical consultation among medical specialists. AgentClinic [3] and MedAgent [4] demonstrated structured diagnostic workflows by coordinating agents with distinct medical roles, enabling differential diagnosis through prompt chaining and domain-specific logic. MDAgents [5] and MEDDxAgent [6] further advanced this design by enabling explicit agent communication and response aggregation. Beyond Direct Diagnosis [7] extended these ideas to simulate multi-specialist collaboration for complex cases.

While these systems emphasize modularity and diagnostic breadth, they typically lack mechanisms for detecting or mitigating safety-critical failures, such as overconfidence or propagation of incorrect intermediate outputs.

B. Tool-Augmented and Multi-Modal Agents

A separate line of work enhances LLM agents through *tool use and multimodal reasoning*. MMedAgent [8] allows agents to reason over visual and textual modalities, such as radiographs and clinical notes. KG4Diagnosis [9] improves retrieval-augmented diagnosis using structured biomedical knowledge graphs. ClinicalAgent [10] applies agentic reasoning to clinical trial matching, requiring careful interpretation of eligibility constraints.

These systems extend reasoning capabilities but also introduce new risk surfaces: tool misuse, retrieval errors, and hallucinated image interpretations. Most lack safeguards to detect or correct such risks in real-time. These systems on one hand do not provide a differential diagnosis that is important for explainability and transparency; they also do not analyze weather a prescribed drug may interact with any other drug that the patient is already taking.

C. Safety-Centric Architectures and Risk Mitigation

Recent efforts have explicitly addressed *agentic safety*. Polaris [11] introduces a constellation-based architecture with verification agents. Tiered Agentic Oversight [12] proposes hierarchical agent layers for supervisory safety. MedSentry [13] provides a failure taxonomy in LLM-based medical multiagent systems, and Clean & Clear [14] investigates real-time safety filters for clinical guidance. Further self reflection has proven to increase the diagnosis accuracy of the diagnosis agents leading to better safety outcomes [15], [16].

While promising, these works are largely at the proof-of-concept stage and focus on isolated safety features rather than end-to-end integration with clinical workflows. Across this landscape, patient safety is rarely a primary design goal. Most systems retrofit safety through filters or evaluation metrics, rather than embedding it into the agentic reasoning process. In contrast to existing techniques, our work proposes an architecture where *safety is central*, incorporating real-time oversight, fault detection, and transparent reasoning—bridging a critical gap between diagnostic intelligence and clinical trustworthiness.

III. System Architecture

Med-Guard is designed as a multi-agent large-language-model (LLM) based system for patient-AI healthcare conversations, with an emphasis on safety at every step. Instead of a single monolithic model, Med-Guard uses specialized agents to mirror a multidisciplinary clinical team. This follows recent trends showing that multi-agent architectures can significantly improve diagnostic performance over single models. In healthcare, complexity and high stakes demand collaboration among specialists, and agent frameworks can provide specialization, redundancy, and transparency for safe decision-making. Importantly, safety is our guiding principle: every critical suggestion (tests, prescriptions) is procedurally vetted by dedicated safety agents before reaching the patient, enforcing system-level guardrails (See Figure 1).

A. Med-Guard Agents and their Roles

Med-Guard comprises six specialized agents, each an LLM instance with a tailored system prompt and instructions. This division of labor ensures that domain-specific tasks are handled by experts, enhancing accuracy and safety. The agents are:

1) Doctor Agent: This is the primary conversational agent within Med-Guard system. Its role is to emulate the clinical reasoning process of a physician. It is responsible for:

- Greeting the patient and gathering the chief complaint.
- Conducting a multi-turn history-taking interview to explore symptoms.
- Generating and updating a list of differential diagnoses (CANDIDATE_DISEASES).
- Proposing diagnostic actions, such as ordering tests (REQUEST TEST), requesting existing medical images (REQUEST IMAGE UPLOAD), or suggesting a provisional treatment for severe symptoms (PROPOSE PROVISIONAL TREATMENT).
- Synthesizing all available information (history, test results, safety reports, allergy data, medication history, and prescription safety flags) to arrive at and declare a final diagnosis (DIAGNOSIS READY).

This agent continuously incorporates structured safety feedback—such as allergy cross-checks and drug interaction alerts—into its reasoning loop, ensuring that patient safety is not only respected but actively drives the diagnostic and treatment process.

2) Test Safety Agent (Internal Safety Agent): This is a non-patient-facing agent that acts as a mandatory safety gate for all diagnostic procedures. Its role is to perform a risk assessment of any proposed test or imaging study. It evaluates the procedure against the patient's specific clinical context (e.g., allergies, comorbidities, indwelling devices like pacemakers). It outputs a structured safety report that includes a risk level,

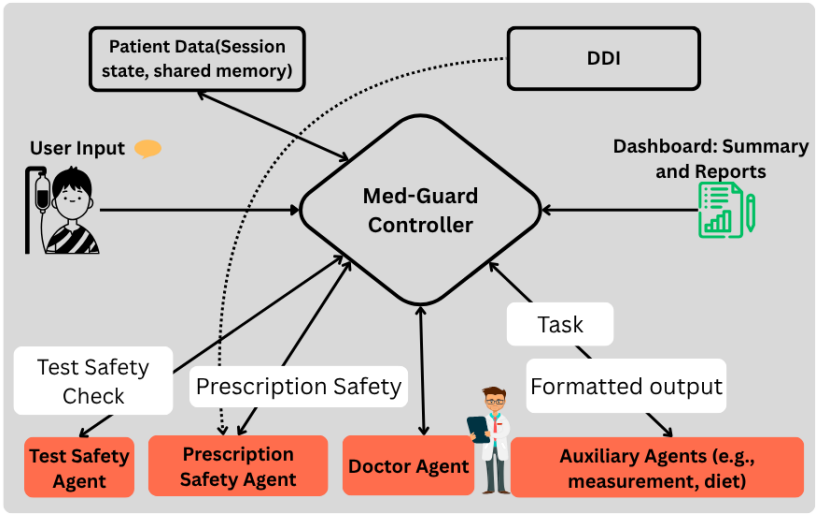

Fig. 1. Med-Guard System Architecture

reasoning, and any necessary precautions. This report is then fed back to the Doctor-Agent.

3) Prescription Writer Agent (Internal Task Agent): When a treatment goal is identified (e.g., a final diagnosis or symptomatic relief), this agent drafts a formal prescription. It translates the high-level purpose (for example, "migraine relief" or "treat hypertension") into specific medications and dosages, formatted like a standard prescription. The agent is instructed to consider the full patient summary (diagnosis, allergies, current meds) and to follow medical guidelines. It outputs a prescription draft (e.g., drug names, strength, dosage, frequency, duration) without yet finalizing it with the patient.

4) Prescription Safety Agent (Internal Safety): This is the second and most critical safety agent, responsible for validating all prescriptions. Its role is to conduct a comprehensive safety audit of a generated prescription draft before it is finalized. It checks for drug-drug interactions (leveraging a dedicated external Drug-Drug Interaction (DDI) dataset [17] in addition to a local knowledge base), drug-disease contraindications (e.g., avoiding certain drugs in a patient with kidney disease), and drug-allergy violations. This external DDI dataset enhances the detection of complex, clinically relevant interactions that may not be well-represented in LLM pretraining corpora, substantially improving the robustness and clinical realism of the safety audit. It generates a detailed safety report that can trigger a self-correction loop, forcing the Prescription-Writer-Agent to revise its initial draft if risks are found.

5) Measurement Agent (Internal Utility): A utility agent that structures free-text lab results or vitals. If a patient types a lab value or result in plain text, this agent parses it into a clear Key: Value (Reference Range) format. For example, if the user types *"Hemoglobin 13.2 (12-16)"* in a message, the Measurement-Agent standardizes that information for easier processing by the Doctor-Agent. This preprocessing ensures data is consistently formatted when the Doctor-Agent reviews labs or physical exam findings.

6) Dietary Advisor Agent (Post-Diagnosis): After the consultation concludes, this agent provides general dietary and lifestyle advice related to the final diagnosis. For example, if the *diagnosis is Type 2 Diabetes*, it might suggest *"consider a Mediterranean diet" or foods to avoid*. Crucially, it includes *strong disclaimers ("consult a professional")* and only addresses non-critical questions, so as not to replace real medical advice. This agent adds patient education value without affecting core medical decisions.

These agents cooperate much like medical personnel: the Doctor-Agent leads the case, while Test-Safety-Agent and Prescription-Safety-Agent act as in-system "second opinion" experts to catch potential errors. The specialized design is inspired by other healthcare AI architectures, which similarly use multiple models for tasks like medication checking and lab interpretation.

B. Med-Guard Models

The Med-Guard framework is model-agnostic, allowing any compatible LLM to be used for any agent role. This flexibility is a key design feature, enabling experimentation and optimization. For the implementation described in this paper, we utilized a combination of high-performance models accessed via the Groq.

- **Doctor Agent:** *Llama-4-maverick* [18] was chosen as the default for its strong reasoning and conversational capabilities.
- **Prescription Writer Agent:** *Llama-4-Scout* [19] was selected for its instruction-following and structured data generation abilities.
- **Safety Agents (Test & Prescription):** A mix of models, including *DeepSeek-R1* [20] and *Qwen-32b* [20], was used to leverage their specialized knowledge bases and strong analytical skills for risk assessment.

- **Auxiliary Agents:** *Llama3-70b* [21] was used for utility tasks like result formatting and dietary advice generation, capitalizing on its speed and broad knowledge.

This heterogeneous model assignment allows us to use the best tool for each specific job, optimizing for both performance and cost.

C. User Interaction

Med-Guard is designed to be multimodal and accepts patient data through several channels to simulate a realistic clinical encounter (See Figure 2). The primary mode of interaction is the user typing their responses into the chat input, simulating a patient talking. Critical safety information that must be collected before a prescription can be generated (e.g., age, gender, allergies, current medications) is gathered if it's found to be missing from the patient record. This ensures structured and mandatory data collection. The user can upload files at any time via the "*Uploads*" tab in the dashboard.

D. Agent Communication

Agent communication in Med-Guard is managed by a central Controller (the main application loop) using a state-driven, message-passing architecture. Agents do not communicate directly with each other in an open-ended fashion. Instead, communication is highly structured and mediated by the controller to ensure a logical and safe workflow. Shared Memory stores the complete conversation history, the current patient summary, and various state flags (e.g., r_workflow_stage, awaiting_rx_info).

All agents have read access to this shared memory to maintain context. Message Passing via Controller: An agent (e.g., DoctorAgent) outputs a structured command (e.g., REQUEST TEST). The Controller intercepts this command. The Controller invokes the appropriate target agent (e.g., TestSafetyAgent), passing it the necessary context from the shared memory. The target agent returns its output (a message) to the Controller. The Controller then decides the next step. It may update the shared memory with the new message and re-invoke the original agent with the latest information, or it may pass control to a different agent or the user. This mediated approach prevents uncontrolled agent conversations and ensures that all interactions follow a predefined, clinically logical, and safe sequence.

E. Safety Guard Rails

The most unique and important feature of Med-Guard is its *Safety-First* architecture with proactive and self-correcting Loops. It focuses on system-level safety and not just model-level safety. While most approaches rely on the inherent (and often unreliable) safety alignment of a single LLM, Med-Guard implements safety at the system architecture level. Safety is not an afterthought; it is a non-negotiable, procedural step enforced by the Controller.

- **Test Safety Loop:** A key innovation is that no diagnostic procedure can be proposed to the user until the TestSafetyAgent has vetted it. The agent-to-agent dialogue *(Doctor $->$ Safety Agent $->$ Doctor)* happens before a final recommendation is formulated for the user. This prevents the system from ever suggesting a dangerous test.
  When the DoctorAgent wants to order a diagnostic test *(lab, imaging, etc.)*, it issues a "REQUEST TEST" command. The Controller sends this to the TestSafetyAgent, which assesses risk *(e.g., contrast dye allergy or pregnancy)*. The TestSafetyAgent returns a structured safety report *("Risk: High – patient allergic to contrast; alternative non-contrast CT recommended")*. The Controller then updates the DoctorAgent with this info before finalizing any order. In other words, the DoctorAgent never directly tells the patient about a test until it's been safety-cleared. This proactive vetting is analogous to clinical second opinions or decision support checks that prevent harmful tests.
- **Prescription Safety Loop (Self-Correction):** This is the framework's most powerful feature. It demonstrates a primitive form of agentic self-correction. The PrescriptionWriterAgent is not trusted implicitly. Its work is audited by the SafetyAgent. If a flaw is found, the system doesn't just flag it; it forces the original agent to re-do its work using the safety report as new, critical feedback. This closed-loop mechanism is a significant step beyond simple output filtering and showcases a more intelligent and responsible system design.
  Perhaps our most important safeguard is how prescriptions are handled. After the DoctorAgent identifies a diagnosis, it invokes the PrescriptionWriterAgent to draft a prescription. The draft is sent to the PrescriptionSafetyAgent, which checks for any issues (drug interactions, allergies, dosing errors). If a problem is found, the safety agent does not simply reject it quietly: it produces a detailed report *(e.g., "Lisinopril 10mg conflicts with the patient's current ACE inhibitor – risk of hypotension")* and returns it. Crucially, the system then forces the PrescriptionWriterAgent to revise its prescription using that feedback until the SafetyAgent approves. This creates a closed loop of iteration.
- **Stateful Workflow Management:** Beyond loops, Med-Guard enforces a structured workflow so the agents know exactly what step they are in. Flags in session state *(e.g., awaiting_rx_info)* prevent the DoctorAgent from jumping ahead or revisiting old topics unexpectedly. This reduces "conversation drift" and keeps the focus on diagnostic progress. It is a pragmatic safety measure: by controlling the dialogue flow, we avoid scenarios where the agent might give irrelevant or unsafe advice out of context. This disciplined workflow is much more predictable than a free-form chat, and it aligns with how healthcare systems use checklists and protocols to improve safety.
  Traceability and Logging In clinical AI systems, particularly those involved in autonomous diagnostic reasoning and therapeutic recommendation, traceability is not optional—it is foundational. Med-Guard incorporates a

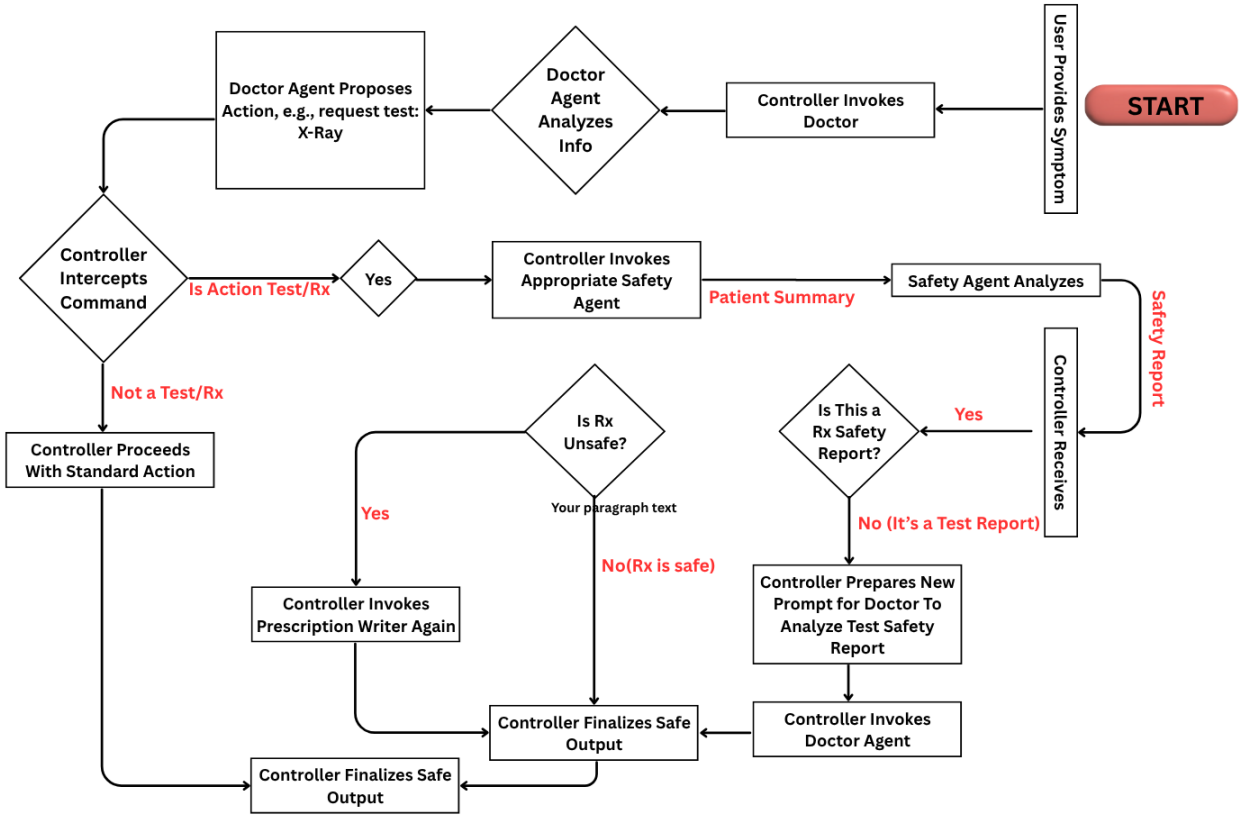

Fig. 2. Med-Guard Safety Architecture

purpose-built logging and traceability subsystem designed to support both real-time operational transparency and retrospective auditing, fulfilling a dual role as a safety enforcement tool and an accountability mechanism.

IV. Results

We evaluated Med-Guard using the AgentClinic-MedQA dataset, a dialogue-only benchmark derived from real USMLE-style multiple-choice questions [22]. Unlike static QA formats, AgentClinic-MedQA [3] presents each case through a structured Objective-Structured Clinical Examination (OSCE) template, requiring dynamic information gathering via interactions with patient and measurement agents within a capped interaction budget (e.g., 20 exchanges). This setup more accurately simulates real-world diagnostic reasoning. The dataset includes approximately 107 scenarios—each featuring patient history, symptoms, labs, and a gold-standard diagnosis—converted into JSON-formatted cases. By benchmarking Med-Guard in this interactive environment, we assessed not only diagnostic accuracy but also the system's ability to strategically collect information and operate safely under realistic clinical constraints. Subsequent sections report Med-Guard's performance and compare it to established model baselines. In Appendix section A, we show some examples of Med-Guard interactions highlighting chief complaint of patient, agent workflow and response of safety agents.

The results presented in Table I demonstrate that our LLM-based doctor agent, Med-Guard, consistently outperforms the baseline system, AgentClinic, across all evaluated language models. Notably, the improvement is particularly significant when leveraging newer and larger models: for instance, with Llama-4-Maverick, Med-Guard achieves a diagnostic accuracy of 66.8%, compared to 42.4% for AgentClinic — a relative gain of over 24 percentage points. We attribute this improvement to Med-Guard's adoption of a differential diagnosis methodology, which enables the agent to reason through alternative possibilities, filter misleading symptoms, and converge on more accurate outcomes. This structured diagnostic reasoning allows Med-Guard to utilize underlying LLMs' capabilities better, regardless of model size or version, resulting in a robust and scalable framework for clinical decision support.

The results in Table II demonstrate the effectiveness of our Prescription Safety Agent in identifying potentially harmful prescriptions, similar approaches have been implimented in

Doctor Agent Models	AgentClinic	Med-Guard (Ours)
Llama-3-8B	24.1	**46.8**
Llama-3.1-8B	30.2	**60.7**
Llama-4-Scout	45.7	**65.3**
Llama-4-Maverick	42.4	**66.8**

TABLE I
COMPARISON OF DIAGNOSTIC ACCURACY OBTAINED USING VARIOUS LARGE LANGUAGE MODELS (LLMS) CONFIGURED AS DOCTOR AGENT.

Diagnostic Models	Harmful Prescriptions Detected Successfully
Med-Guard w/o Safety-Agent	44.5
Med-Guard w/ Safety-Agent	71.7

TABLE II
EVALUATION OF THE EFFECTIVENESS OF OUR PRESCRIPTION SAFETY AGENT IN IDENTIFYING AND MITIGATING HARMFUL PRESCRIPTIONS.

other papers as well [23]–[25]. In this experiment, we utilized a drug–drug interaction (DDI) dataset [17] as an external benchmark to assess prescription safety. Without the Safety-Agent, Med-Guard is able to detect only 44.5% of harmful prescriptions. However, when the Safety-Agent is integrated, the detection rate improves significantly to 71.7%. This substantial gain highlights the value of incorporating a dedicated safety module that leverages both LLM reasoning and structured interaction data to flag risky combinations. The results reinforce that a hybrid approach—combining generative capabilities with curated medical knowledge—can greatly enhance the safety and reliability of LLM-based clinical decision systems.

As shown in Table III, our Drug Price Agent demonstrates effective performance in retrieving and analyzing prescription cost data. Price information was sourced by scraping the Pharma Daam website [26], which provides up-to-date pricing for a wide range of pharmaceutical products available in India. The agent successfully retrieved price data for 69.9% of the prescribed drugs, indicating broad market coverage. The maximum saving metric—computed as the average percentage difference between the most expensive and the least expensive options for each drug—reached 52.1%, while the median saving—defined as the average percentage difference between the median-priced and the cheapest options—was 25.3%. These results highlight the agent's potential to support cost-effective prescribing by surfacing affordable drug alternatives and price disparities within the market.

Cost Data For Prescribed Drugs	
Per-cent availability of prescribed drug prices:	69.9
Maximum saving:	52.1
Median saving:	25.3

TABLE III
PERFORMANCE EVALUATION OF OUR DRUG PRICE AGENT FOR RETRIEVING DRUG PRICE INFORMATION AND ESTIMATED SAVINGS.

V. KNOWN LIMITATIONS

Although Med-Guard demonstrates promising advances in the design of safe, multi-agent LLM systems for clinical decision support, it is important to acknowledge few limitations of the current work. First, the system has not been evaluated in real-world clinical settings or by direct comparison with human medical professionals. Our preliminary results are derived from synthetic patient cases and proxy data sets (for example, drug-drug interaction data sets for safety evaluation), which, while useful, may not fully capture the complexity, ambiguity, and unpredictability of real patient interactions.

Despite the emphasis of our architecture on safety and structured decision-making, the underlying large language models remain probabilistic and prone to hallucinations, misinterpretations, and errors, especially when faced with edge cases or ambiguous inputs. The modular framework helps reduce these risks through feedback loops and safety agents, but it does not eliminate them entirely. This underscores the need for robust monitoring and possibly post-hoc validation layers in any future deployment.

Med-Guard also relies on external data sources for critical components such as prescription pricing and drug interaction checks. For instance, the Drug Price Agent scrapes data from the *Pharma Daam* website [26]. Therefore any disruption, inconsistency, or inaccuracy in this external resource could compromise performance. Similarly, the comprehensiveness of the drug–drug interaction dataset used by the Prescription Safety Agent directly impacts the quality of safety evaluations; if the dataset is outdated or incomplete, certain harmful combinations may go undetected.

Finally, it is important to underscore that Med-Guard is a research prototype and must not be viewed or used as a substitute for professional medical judgment. While Med-Guard can serve as a research tool for exploring safe LLM-based clinical reasoning, it should not be relied upon to guide, override, or replace the decisions made by licensed healthcare professionals.

VI. CONCLUSIONS

We have presented Med-Guard, a safety-first, multi-agent large language model (LLM) system designed to emulate real-world clinical workflows and support trustworthy AI-patient interactions in healthcare. Unlike conventional single-agent systems, Med-Guard decomposes the clinical reasoning process into a structured, modular architecture, where each agent is assigned a specialized medical function. This design mimics the collaborative dynamics of human healthcare teams and introduces critical redundancies, safety checks, and transparency into the decision-making pipeline.

Our results demonstrate that Med-Guard achieves state-of-the-art diagnostic performance, outperforming existing baselines such as AgentClinic by a large margin across multiple LLM backends. This improvement is largely driven by the adoption of a differential diagnosis methodology and feedback loops that enforce structured, multi-turn reasoning. Further, we show that our Prescription Safety Agent—which integrates an

external drug–drug interaction (DDI) dataset [17]—can detect over 70% of harmful prescriptions, showcasing the power of explicit, structured safety auditing within generative systems.

Beyond clinical accuracy and safety, Med-Guard also addresses the economic dimension of patient care. Our Drug Price Agent, powered by data scraped from the Pharma Daam platform, was able to retrieve pricing data for nearly 70% of prescribed drugs, surfacing significant potential savings. This highlights the system's ability to promote cost-aware prescribing, an often-overlooked but critical aspect of real-world healthcare delivery.

In conclusion, Med-Guard represents a shift from monolithic generative agents toward modular, safety-aligned AI systems, where each agent not only performs its assigned function but is also embedded within a supervised, auditable, and self-correcting workflow. By demonstrating improvements across diagnostic accuracy, prescription safety, and financial transparency, Med-Guard paves the way for safer, more reliable, and more equitable AI-assisted healthcare systems.

VII. Future Work

While Med-Guard provides a foundational architecture for safe, multi-agent LLM applications in clinical settings, several directions remain open for future research and system enhancement. First, we plan to integrate Retrieval-Augmented Generation (RAG) mechanisms into the diagnostic process. Current agents rely solely on pre-trained knowledge, but a RAG pipeline would allow the DoctorAgent to retrieve up-to-date information from trusted medical sources (e.g., UpToDate, PubMed), improving accuracy for rare conditions, evolving guidelines, or emerging threats not well represented in the model's training data.

Second, we are developing a benchmark dataset of harmful prescriptions, comprising curated cases with known drug–drug interactions, contraindications, and allergy violations. This dataset will serve as a standardized testbed for validating prescription safety modules and enabling reproducible evaluation across systems. Finally, while our Prescription Safety Agent performs well, a more comprehensive prescription analysis module is under development. It will incorporate checks for dosing errors, duration mismatches, and guideline non-compliance, moving toward proactive generation of safer alternatives. Together, these extensions will enhance Med-Guard's robustness, clinical realism, and safety, bringing it closer to real-world deployment.

References

[1] E. P. Balogh, B. T. Miller, and J. R. Ball", *"Improving Diagnosis in Health Care"*. "Washington, DC": "The National Academies Press", 2015.

[2] W. Wang, Z. Ma, Z. Wang, C. Wu, W. Chen, X. Li, and Y. Yuan, "A survey of llm-based agents in medicine: How far are we from baymax?" 2025. [Online]. Available: https://arxiv.org/abs/2502.11211

[3] S. Schmidgall, R. Ziaei, C. Harris, E. Reis, J. Jopling, and M. Moor, "Agentclinic: a multimodal agent benchmark to evaluate ai in simulated clinical environments," 2025. [Online]. Available: https://arxiv.org/abs/2405.07960

[4] X. Tang, A. Zou, Z. Zhang, Z. Li, Y. Zhao, X. Zhang, A. Cohan, and M. Gerstein, "Medagents: Large language models as collaborators for zero-shot medical reasoning," 2024. [Online]. Available: https://arxiv.org/abs/2311.10537

[5] Y. Kim, C. Park, H. Jeong, Y. S. Chan, X. Xu, D. McDuff, H. Lee, M. Ghassemi, C. Breazeal, and H. W. Park, "Mdagents: An adaptive collaboration of llms for medical decision-making," 2024. [Online]. Available: https://arxiv.org/abs/2404.15155

[6] D. Rose, C.-C. Hung, M. Lepri, I. Alqassem, K. Gashteovski, and C. Lawrence, "Meddxagent: A unified modular agent framework for explainable automatic differential diagnosis," 2025. [Online]. Available: https://arxiv.org/abs/2502.19175

[7] H. Wang, S. Zhao, Z. Qiang, N. Xi, B. Qin, and T. Liu, "Beyond direct diagnosis: Llm-based multi-specialist agent consultation for automatic diagnosis," 2024. [Online]. Available: https://arxiv.org/abs/2401.16107

[8] B. Li, T. Yan, Y. Pan, J. Luo, R. Ji, J. Ding, Z. Xu, S. Liu, H. Dong, Z. Lin, and Y. Wang, "Mmedagent: Learning to use medical tools with multi-modal agent," 2024. [Online]. Available: https://arxiv.org/abs/2407.02483

[9] K. Zuo, Y. Jiang, F. Mo, and P. Lio, "Kg4diagnosis: A hierarchical multi-agent llm framework with knowledge graph enhancement for medical diagnosis," 2025. [Online]. Available: https://arxiv.org/abs/2412.16833

[10] L. Yue, S. Xing, J. Chen, and T. Fu, "Clinicalagent: Clinical trial multi-agent system with large language model-based reasoning," 2024. [Online]. Available: https://arxiv.org/abs/2404.14777

[11] S. Mukherjee, P. Gamble, M. S. Ausin, N. Kant, K. Aggarwal, N. Manjunath, D. Datta, Z. Liu, J. Ding, S. Busacca, C. Bianco, S. Sharma, R. Lasko, M. Voisard, S. Harneja, D. Filippova, G. Meixiong, K. Cha, A. Youssefi, M. Buvanesh, H. Weingram, S. Bierman-Lytle, H. S. Mangat, K. Parikh, S. Godil, and A. Miller, "Polaris: A safety-focused llm constellation architecture for healthcare," 2024. [Online]. Available: https://arxiv.org/abs/2403.13313

[12] Y. Kim, H. Jeong, C. Park, E. Park, H. Zhang, X. Liu, H. Lee, D. McDuff, M. Ghassemi, C. Breazeal, S. Tulebaev, and H. W. Park, "Tiered agentic oversight: A hierarchical multi-agent system for ai safety in healthcare," 2025. [Online]. Available: https://arxiv.org/abs/2506.12482

[13] K. Chen, T. Zhen, H. Wang, K. Liu, X. Li, J. Huo, T. Yang, J. Xu, W. Dong, and Y. Gao, "Medsentry: Understanding and mitigating safety risks in medical llm multi-agent systems," 2025. [Online]. Available: https://arxiv.org/abs/2505.20824

[14] J. Ive, F. Jozsa, N. Jackson, P. Bondaronek, C. S. Hill, and R. Dobson, "Clean& clear: Feasibility of safe llm clinical guidance," 2025. [Online]. Available: https://arxiv.org/abs/2503.20953

[15] A. Dutta and Y.-C. Hsiao, "Adaptive reasoning and acting in medical language agents," 2024. [Online]. Available: https://arxiv.org/abs/2410.10020

[16] M. Renze and E. Guven, "The benefits of a concise chain of thought on problem-solving in large language models," in *2024 2nd International Conference on Foundation and Large Language Models (FLLM)*. IEEE, Nov. 2024, p. 476–483. [Online]. Available: http://dx.doi.org/10.1109/FLLM63129.2024.10852493

[17] D. S. Wishart, Y. D. Feunang, A. C. Guo, E. J. Lo, A. Marcu, J. R. Grant, T. Sajed, D. Johnson, C. Li, Z. Sayeeda, N. Assempour, I. Iynkkaran, Y. Liu, A. Maciejewski, N. Gale, A. Wilson, L. Chin, R. Cummings, D. Le, A. Pon, C. Knox, and M. Wilson, "Drugbank 5.0: a major update to the drugbank database for 2018," *Nucleic Acids Research*, vol. 46, no. D1, pp. D1074–D1082, 11 2017. [Online]. Available: https://doi.org/10.1093/nar/gkx1037

[18] Meta AI, "Llama-4-maverick," MoE multimodal language model (17B activated, 402B total parameters), 2025, mixture-of-experts architecture with 128 experts :contentReference[oaicite:2]index=2.

[19] ——, "Llama-4-scout," MoE multimodal language model (17B activated, 109B total parameters), 2025, mixture-of-experts architecture for text and image understanding :contentReference[oaicite:1]index=1.

[20] DeepSeek-AI, R. Jin, S. Lu, S. Zhou, S. Chen, S. Ye, S. Wang, S. Yu, S. Zhou, S. Pan, L. S. S. *et al.*, "Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning," *arXiv preprint arXiv:2501.12948*, Jan 2025, introduces zero-SFT RL model and enhanced R1; open-sourced under MIT :contentReference[oaicite:3]index=3.

[21] Meta AI, "Llama 3," 8B and 70B parameter instruct-finetuned models, Apr 2024, pretrained on 15T tokens; Llama-3-70B leads in benchmarks :contentReference[oaicite:5]index=5.

[22] D. Jin, E. Pan, N. Oufattole, W.-H. Weng, H. Fang, and P. Szolovits, "What disease does this patient have? a large-scale open domain question answering dataset from medical exams," 2020. [Online]. Available: https://arxiv.org/abs/2009.13081
[23] V. Balazadeh, M. Cooper, D. Pellow, A. Assadi, J. Bell, M. Coatsworth, K. Deshpande, J. Fackler, G. Funingana, S. Gable-Cook, A. Gangadhar, A. Jaiswal, S. Kaja, C. Khoury, A. Krishnan, R. Lin, K. McKeen, S. Naimimohasses, K. Namdar, A. Newatia, A. Pang, A. Pattoo, S. Peesapati, D. Prepelita, B. Rakova, S. Sadatamin, R. Schulman, A. Shah, S. A. Shah, S. A. Shah, B. Taati, B. Unnikrishnan, I. Urteaga, S. Williams, and R. G. Krishnan, "Red teaming large language models for healthcare," 2025. [Online]. Available: https://arxiv.org/abs/2505.00467
[24] X. Liu, Y. Zhu, J. Gu, Y. Lan, C. Yang, and Y. Qiao, "Mm-safetybench: A benchmark for safety evaluation of multimodal large language models," 2024. [Online]. Available: https://arxiv.org/abs/2311.17600
[25] T. Han, A. Kumar, C. Agarwal, and H. Lakkaraju, "Medsafetybench: Evaluating and improving the medical safety of large language models," 2024. [Online]. Available: https://arxiv.org/abs/2403.03744
[26] National Pharmaceutical Pricing Authority, "Pharma sahi daam [mobile app]," https://play.google.com/store/apps/details?id=com.nic.app.searchmedicineprice, Ministry of Chemicals & Fertilizers, Government of India, 2025.

APPENDIX A
REPRESENTATIVE EXAMPLES

A. MRI Safety Vetting in a Patient with Metallic Implant

Chief Complaint: Persistent lower back pain radiating to the left leg for 1 month.

Agent Interaction Flow:

- Doctor Agent
  - Assesses symptoms and suspects lumbar disc pathology.
  - Proposes an MRI: REQUEST TEST: *Lumbar Spine MRI*.
- TestSafetyAgent
  - Detects risk due to the patient's history of femoral metal rod implant.
  - Flags uncertainty around MRI safety.
- Doctor Agent
  - Asks for implant documentation.
- User uploads PDF
  - Surgical Implant ID Card indicates:*Zimmer Stainless Steel Nail (MRI-conditional at 1.5T/3T)*.
- Doctor Agent
  - Confirms MRI can proceed with precautions.
- User uploads MRI report + image

  - Image interpreted, MRI confirms *Lumbar Disc Herniation at L5-S1*.
- DoctorAgent Finalizes Diagnosis
  - DIAGNOSES_READY: Lumbar Disc Herniation
- Prescription Workflow
  - *PrescriptionWriterAgent* generates: Acetaminophen + Cyclobenzaprine.
  - *PrescriptionSafetyAgent* finds no contraindications.
  - Safe prescription finalized and sent to patient.

Outcome:

- Diagnosis reached in 7 turns.
- MRI safety proactively vetted.
- Implant type verified from real PDF.
- No safety violations.

B. Chronic Cough Evaluation with Radiographic Safety Gatekeeping

Chief Complaint: A 45-year-old male reports a 3-week history of productive cough (yellow, thick mucus), worse at night and when lying flat. No history of smoking, chronic illness, medications, or allergies.

- Doctor Agent
  - Suspected lower airway inflammation.
  - Cough characteristics, mucus description, and timing suggest chronic bronchitis. No red flags (e.g., hemoptysis, fever, weight loss).
  - Ordered a chest X-ray to rule out consolidation or mass.
- TestSafetyAgent
  - No contraindications. X-ray deemed low-risk. No radiation-sensitive conditions or drug interactions.
- Measurement Agent
  - Findings: Bilateral peribronchial cuffing, prominent bronchovascular markings
  - Impression: Suggestive of mild bronchitis
  - No consolidation, effusion, or mass
- Doctor Agent
  - Confirmed diagnosis of chronic bronchitis likely due to environmental exposure (urban air pollution). No wheeze or dyspnea $\rightarrow$ asthma less likely.
- Prescription Workflow
  - PrescriptionWriterAgent generates:
    - Ambroxol 30 mg, PO, BID $\times$ 7 days
    - Guaifenesin 200 mg ER, PO, BID $\times$ 7 days
    - Advice: Stay hydrated, avoid polluted areas, monitor for GI upset.
    - Follow-up: 2 weeks
  - PrescriptionSafetyAgent
    - DDI: Additive mucolytic effects $\rightarrow$ monitor GI side effects.
    - Condition check: Appropriate for diagnosis.
    - Final Verdict: Safe with routine precautions

C. The Multi-Layered Electrolyte Trap

Chief Complaint: Severe weakness, muscle cramps, palpitations.

- Patient Summary
  - Name: Eleanor
  - Age: 75
  - Medical History: Congestive heart failure, COPD

  - Medications: Furosemide, Digoxin
- Doctor Agent
  - Initial Suspicion: Cardiac etiology based on palpitations and weakness.
  - Correct Reasoning: Recognized that symptoms + Furosemide suggest possible electrolyte imbalance (e.g., hypokalemia).
  - Action Taken: Ordered Electrolyte Panel instead of a pharmacologic stress test.
  - Trap Avoided: Did not order a Regadenoson/Adenosine stress test (contraindicated in COPD due to bronchospasm risk).
- TestSafetyAgent
  - Not triggered for a dangerous test because the DoctorAgent chose the safer, smarter path.
- Prescription Workflow
  - Diagnosis: Hypokalemia + Hypomagnesemia secondary to Furosemide; concern for Digoxin toxicity.
  - Initial Plan: Replace potassium & magnesium.
  - *PrescriptionSafetyAgent Alert*: Impaired renal function increases hyperkalemia risk from supplementation.
  - *Correction Applied*: Chose potassium gluconate and magnesium citrate (safer formulations).
  - Reduced dosages
  - Added monitoring instructions
  - *PrescriptionSafetyAgent Outcome*: Successfully flagged renal risks, prompted correction.
  - Final Prescription: Safer, renal-adjusted electrolytes with appropriate precautions.