CoVaPh: A Vision-Language Multi-Agent Dialogue System for Tool-Augmented Pharmacogenetic Reasoning and Personalized Guidance

Shang-Chun Luke Lu[1†*] **Hsin Yang**[2*] **Hui-Hsin Xue**[1] **Ping Lin Tsai**[3] **Yu Jing Weng**[3]
Shiou-Chi Li[4] **Jen-Wei Huang**[2†] **Hui Hua Chang**[3,5,6†]

[1]Miin Wu School of Computing, [2]Dept. of Electrical Engineering,
[3]Inst. of Clinical Pharmacy and Pharmaceutical Sciences,
[4]Inst. of Computer and Communication Engineering,
[5]School of Pharmacy, National Cheng Kung University, Tainan, Taiwan
[6]Dept. of Pharmacy, National Cheng Kung University Hospital, Tainan, Taiwan
{slu18@illinois.edu, jwhuang@mail.ncku.edu.tw, huihua@ncku.edu.tw}

Abstract

The post-pandemic healthcare labor crisis has intensified the demand for accessible, high-precision pharmaceutical care. To meet this challenge, we introduce CoVaPh, a multi-agent pharmacogenetic framework that integrates information retrieval with Large Language Model (LLM) and Vision-Language Model (VLM) technologies. At its core, a fine-tuned query rewriting module transforms clinical inquiries into structured search indices, ensuring precise multimodal retrieval from CPIC and PharmGKB while mitigating hallucination risks. By synthesizing structured API data with unstructured evidence from guidelines, our framework delivers highly reliable, context-aware responses, surpassing benchmarks by 10% on expert-curated datasets. This approach provides a scalable solution to alleviate clinical workloads and democratize access to specialized medical knowledge.

1 Introduction

The success of personalized medicine depends on our ability to accurately interpret complex pharmacogenetic (PGx) guidelines. For clinicians, translating a patient's genetic data into precise dosing recommendations is a critical but time-consuming task. This process requires searching through scattered information found in dense medical papers and regulatory documents. These sources contain diverse types of data, ranging from plain text and complex figures to tables and web APIs. Doing this search and summary by hand is not only slow but also prone to human error, where even a single mistake can cause serious health issues for patients. Adding to this challenge is the global shortage of healthcare workers (Mercer, 2025), which highlights the urgent need for intelligent dialogue systems that can support clinicians and improve access to high-precision care.

While recent advancements in agentic AI offer a path forward, existing systems fall short. Prior work has laid foundational stones but reveals key gaps. Tool-augmented LLMs like Toolformer (Schick et al., 2023) and ToolLLM (Qin et al., 2023) demonstrate effective API integration for general tasks, yet they rarely address multimodal domains like pharmacogenomics. Retrieval-augmented generation (RAG) techniques enhance knowledge grounding, but standard text-based retrievers falter on visual elements in medical guidelines. Domain-specific efforts, such as the PGQA benchmark (Gehrmann et al., 2024) for pharmacogenomic question answering, highlight the need for precision in this field, where any errors can be devastating. VLMs like Qwen-VL (Qwen Team, 2024) excel at image-text understanding, but their application to agentic reasoning in pharma is underexplored. Moreover, most advanced systems rely on proprietary giants (e.g., GPT-4) or ultra-large open models (70B+ parameters), limiting accessibility for resource-constrained settings.

To bridge this gap, we present CoVaPh, a vision-language multi-agent framework for pharmacogenetic reasoning. CoVaPh orchestrates a team of specialized AI agents built upon an accessible 32B parameter open-source reasoning model. At its core, a fine-tuned "Experienced Query Rewriter" transforms ambiguous clinical questions into precise, structured queries optimized for visual data retrieval in the pharmacogenomics domain. These queries then trigger a hybrid retrieval pipeline that fuses two critical information sources: (1) a multi-

[*] These authors contributed equally to this work.
[†] Corresponding authors.

Proceedings of the 16th International Workshop on Spoken Dialogue Systems Technology, pages 441–451
February 26–March 1, 2026. ©2026 Association for Computational Linguistics

modal RAG system that extracts information from both the text and visual elements of CPIC guideline documents, and (2) direct API calls to the CPIC database for real-time, patient-specific dosing recommendations and population-level allele frequencies.

Our work makes the following contributions:

1. A Novel Multi-Agent Architecture: We design and implement a collaborative system where a 32B LLM works with specialized agents for query rewriting, multimodal retrieval, and real-time API interaction, addressing the gap in developing versatile capabilities on more accessible models.

2. Hybrid Multimodal & API-Driven RAG: We introduce a retrieval mechanism that uniquely combines a multimodal vector database for interpreting tables and figures in PDFs with live API calls for the most current, structured pharmacogenomic data, a limitation in most standard RAG systems.

3. Domain-Specific Fine-Tuning for Safety: We fine-tune our query-rewriting agent on a curated dataset of valid and invalid queries, teaching it to explicitly deny retrieval when no official guideline exists—a critical safety feature to prevent model hallucination.

4. Evaluation Findings: On our four-metric benchmark, CoVaPh attains competitive performance relative to Gemini 2.5 Pro and Grok-4: +2–3% in overall score and +6–9% in accuracy, with no material differences in completeness, clarity, or relevance.

By automating and error-proofing the complex task of guideline interpretation, CoVaPh presents a tangible pathway to alleviating the burden on healthcare professionals, enhancing patient safety, and promoting equitable access to personalized medicine.

2 Related Work

Our research integrates advancements across several key AI domains to empower a mid-sized (32B) open-source language model (LM) with capabilities previously limited to massive proprietary systems. By augmenting the model with vision modules, search APIs, and retrieval mechanisms, we address a gap in the literature. This review covers five areas: tool-augmented LMs, VLMs, RAG, query rewriting, and domain-specific retrieval for pharmacogenomics.

2.1 Tool-Augmented Language Models

Enhancing Language Models (LMs) with external tools overcomes intrinsic limitations like knowledge cutoffs. **Toolformer** (Schick et al., 2023) pioneered a self-supervised method for LMs to learn API calls, improving zero-shot performance on knowledge-intensive tasks. The **ReAct** framework (Yao et al., 2022) synergized reasoning and acting, creating more robust execution traces that reduce hallucinations. This concept evolved into agentic frameworks like **ToolLLaMA** (Qin et al., 2023) and multi-agent systems such as **AutoGen** (Wu et al., 2023), where specialized agents collaborate to solve problems.

However, high-performing tool-augmented models are typically proprietary or very large (70B+). Our work addresses this gap by developing versatile, multi-tool capabilities on an accessible 32B parameter model, the OpenReasoning-Nemotron 32B.

2.2 Multimodal Vision–Language Models

Integrating vision has transformed LMs into powerful multimodal systems. A key breakthrough was visual instruction tuning, pioneered by **LLaVA** (Liu et al., 2023), which projected visual features into an LLM's embedding space to create powerful, open-source multimodal chat models (Zhang et al., 2023; Yin et al., 2023).

We use the text-based **OpenReasoning-Nemotron-32B** (Qwen Team, 2024) as our core reasoner and orchestrate it with a separate, lightweight vision module, the **Nemotron Nano VLM** (NVIDIA, 2024). This approach fills a research gap, as most work uses monolithic VLMs rather than integrating a powerful text reasoner with a distinct vision tool.

2.3 Retrieval-Augmented Generation (RAG)

Retrieval-Augmented Generation (RAG) mitigates hallucinations by grounding LMs in external data (Lewis et al., 2020). The field has evolved to advanced techniques that improve retrieval relevance and adaptivity. These include generating hypothetical documents (**HyDE** (Gao et al., 2022)) and enabling models to self-critique and decide when to retrieve (**Self-RAG** (Asai et al., 2023), **Corrective RAG (CRAG)** (Shi et al., 2024)). In

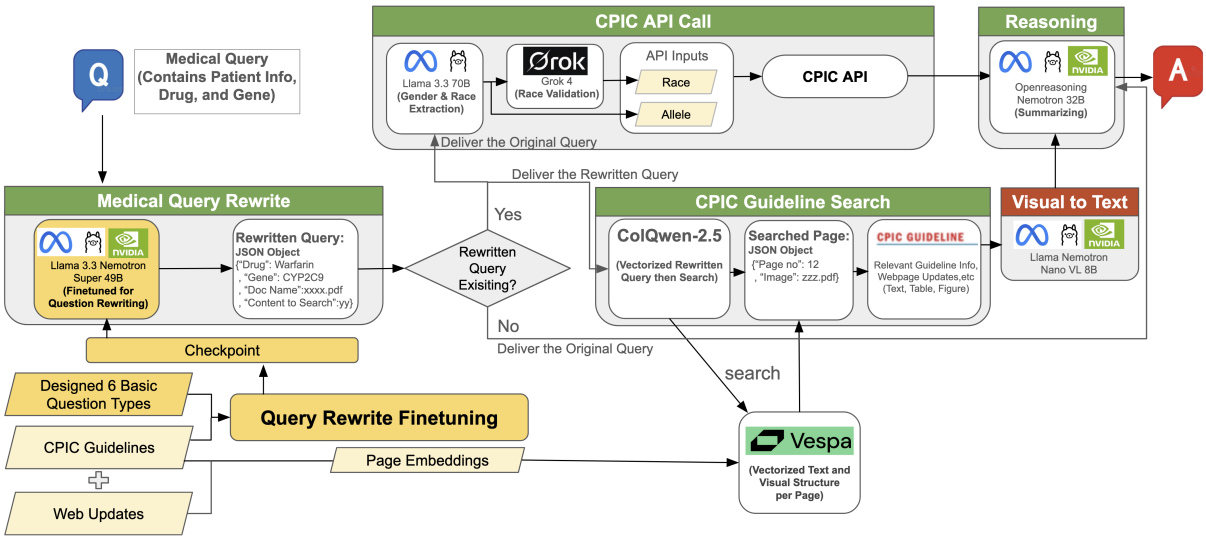

Figure 1: Overview of the CoVaPh pipeline, from patient questions to final organized answers.

medicine, iterative retrieval has also proven effective (**i-MedRAG** (Xiong et al., 2024)).

A key limitation of existing RAG systems is their reliance on static, unstructured text. Our work overcomes this by creating a hybrid RAG system that retrieves from both a vector database of static clinical guidelines and live, structured data via the CPIC API.

2.4 Query Rewriting for Dense Retrieval

Effective RAG depends on high-quality queries, as raw user inputs are often ambiguous. Modern approaches use LLMs to reformulate initial queries into more effective formats for retrieval. Techniques include generating multiple query variations (Ren et al., 2024; Li et al., 2024) or explicitly rewriting the query before retrieval, as in the **Rewrite-Retrieve-Read** framework (Mao et al., 2023).

We adopt this paradigm by fine-tuning the OpenReasoning-Nemotron-32B as a specialized "Experienced Query Rewriter." It transforms ambiguous clinical questions into precise queries optimized for our pharmacogenomic vector database, bridging the gap between user intent and our structured knowledge base.

2.5 Domain-Specific Retrieval: Pharmacogenomics

Applying LMs to specialized domains like pharmacogenomics (PGx) requires high precision and access to structured knowledge. The Clinical Pharmacogenetics Implementation Consortium (**CPIC®**) provides this via peer-reviewed guidelines and a structured RESTful API. Recent work includes benchmarks like **PGxQA** (Gehrmann et al., 2024) and RAG systems for clinical guidelines like **Quicker** (Sharma et al., 2024).

However, these systems typically reason over static text corpora. The critical research gap is the lack of systems performing real-time, API-driven reasoning on structured PGx data. To our knowledge, our work is the first to combine a 32B model with a fine-tuned rewriter and a hybrid RAG system that queries both a structured guideline vector database and the live APIs, enabling nuanced question answering in this high-stakes domain.

3 Methodology

3.1 Specialized Query Rewriting via Fine-tuning

Automating the retrieval of CPIC guidelines in response to patient-specific questions poses two key challenges. First, drug and gene names are often long and syntactically complex, causing a plain LLM to mis-align a drug–gene pair with its correct guideline. Second, some drug–gene combinations have no corresponding guideline, yet a naive model may hallucinate a recommendation. To overcome these issues, we fine-tune our LLM on a synthetic corpus that explicitly teaches (1) how to map valid drug–gene queries to canonical CPIC guideline titles and (2) how to deny retrieval when no guideline

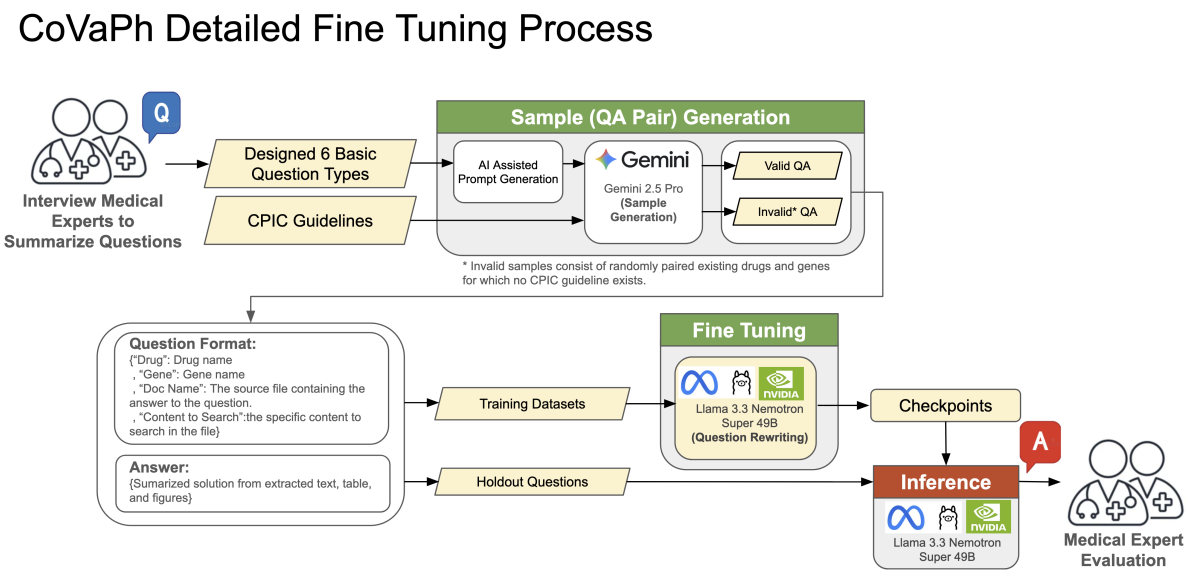

Figure 2: Detailed finetuning process of the query re-writer based on Nemotron Super 49B model.

exists.

Model and Fine-tuning Setup. We choose the Llama Nemotron 3.3 Super 49B as our rewriting model, leveraging PEFT's LoRA adapters for parameter-efficient fine-tuning. We further augment the training with a dataset generated by frontier closed model (Google Gemini 2.5 Pro) to enrich query diversity, and we craft a specialized system prompt during instruction fine-tuning to guide the model's behavior toward precise mapping and explicit denial.

Template Generation. We define six question types to cover the basic spectrum of CPIC use cases, with bolded font highlighting the gene or drug that applies to the given problem:

1. *Dose Adjustment Guidance*: "What is the recommended **efavirenz** dose for a **CYP2B6** poor metabolizer?"
2. *Clinical Risk Stratification*: "What is the increased risk of CNS adverse effects for a **CYP2B6** poor metabolizer?"
3. *Pharmacokinetic/Pharmacodynamic Data*: "What is the suggested therapeutic range for plasma **efavirenz** concentrations?"
4. *Genotype–Phenotype Mapping*: "Which diplotypes define a **CYP2B6** intermediate metabolizer?"
5. *Basic Gene/Drug Information*: "What is the function of the **CYP2B6*6** allele?"
6. *Special Populations*: "What is the dosing recommendation for **efavirenz** in children $<$40 kg with a **CYP2B6** poor metabolizer phenotype?"

Valid vs. Invalid Examples. For each template, we generate two kinds of examples. The valid examples pair an existing guideline title with a normalized drug–gene pair, ensuring that the model learns from real CPIC recommendations. In contrast, the invalid examples are constructed where no CPIC guideline exists, which teaches the model to respond with an explicit denial rather than fabricating an answer.

Examples

Valid: Input: "Dose adjustment for clopidogrel in CYP2C19 poor metabolizers?"

Output:

```
{
    "Drug Name":"clopidogrel",
    "Gene Name":"CYP2C19",
    "CPIC Guideline":
        "Clinical Pharmacogenetics
        Implementation Consortium
        Guideline for CYP2C19
        and Clopidogrel Dosing",
    "Content to Search":
      "dose adjustment recommendation
       for CYP2C19 poor metabolizers"
}
```

Invalid: Input: "What guideline covers fluoxetine and CYP2D6*37?"

Output:

```
"No CPIC guideline information
    available."
```

By fine-tuning on this mixed dataset of template-driven valid and invalid queries, the model learns to (1) produce exact guideline titles for supported combinations and (2) refuse retrieval for unsupported ones.

$$Query_{Rewritten} = \mathrm{LLM}_{\mathrm{Rewrite}}(Q_{patient})$$

3.2 Hybrid Multimodal Search Database

Traditional retrieval of PDF documents relies on text-only indexing (e.g., BM25 over OCR-extracted text), which cannot fully exploit visual structure such as tables, figures, and pathway diagrams. We apply a hybrid multimodal search database in Vespa Cloud that fuses both text and image embeddings to build richer representations and support a scalable two-phase retrieval process.

Two-Phase Retrieval At query time, the patient question Q_{patient} is first rewritten by our fine-tuned LLM:

$$Query_{Rewritten} = \mathrm{LLM}_{\mathrm{Rewrite}}(Q_{patient})$$

This rewritten query—normalized to match CPIC drug–gene terminology—serves as input to Vespa's retrieval pipeline:

1. **Phase 1 (BM25 Recall):** Match `Rewritten Query` against the `text` field to retrieve top candidates.
2. **Phase 2 (Late Interaction Reranking):** Embed each token of `Rewritten Query` with ColQwen-2.5, compute dot-products against document patch embeddings stored in the `embedding` tensor, take the maximum per token, and sum these maxima to produce final scores.

VLM Extraction and Normalization We begin by calling NVIDIA's NIM service with `llama-3.1-nemotron-nano-vl-8b` to process each PNG image. The VLM first extracts all table contents, identifying rows, columns, and any associated footnotes (such as a, b, or c), and converts them into structured table objects. After completing table extraction, the model then processes the remaining non-tabular body text, including narrative updates and free-form guideline commentary, and appends these segments after the table data. Finally, the response generated by the VLM is stored and inserted into the prompt of the reasoning model, serving as input for the final answer generation.

Downstream Integration The resulting outputs containing text and table structure from CPIC guideline will be fed into our OpenReasoning engine, enabling final decision made.

3.3 Patient Information Extraction and CPIC API Call

Although our VLM-based retrieval pipeline covers most guideline content, certain patient-specific details—such as personalized dosing recommendations and phenotype frequencies—require direct access to CPIC's backend APIs. We integrate two CPIC endpoints to enrich model outputs:

1. **Recommendation Lookup:**
   Use the patient's diplotype to fetch dosing guidance:

   ```
   GET /rpc/recommendation_lookup?
       diplotypelookup={
           "GENE":
               {"ALLELE":count}
       }
   ```

   Here, the JSON parameter reflects the extracted star-allele counts.

2. **Population Frequency:**
   Obtain phenotype probabilities for a specific ancestry:

   ```
   GET /rpc/population_frequency?
     frequencylookup=
       {
           "GENE":GENE,
       }
   ```

Patient questions Q_{patient} are first processed by large language models to extract, here we use the Llama 3.3 70B to extract both of these information, and extracted information will be checked by Grok 4 for validity:

- *Allele Number*: CPIC encodes diplotypes as a JSON map from the gene symbol to a dictionary of star-allele copy counts (integers, typically 0–2). Each inner key is a star allele and its value is the observed count. For example, a CYP2C9 *1/*3 diplotype is `{"CYP2C9":{"*1":1,"*3":1}}`, while a homozygous CYP2D6 *4/*4 is `{"CYP2D6":{"*4":2}}`. Multiple genes can be included by adding additional top-level keys; alleles not present may be omitted or set to 0. We pass this structure verbatim to the CPIC API and persist it with retrieved evidence for reproducibility.

- *Race*: For CPIC population-frequency lookups, the database stores ancestry under nine canonical groups, and the `race` field must be a single string chosen from the following set: *Latino, American, European, Oceanian, East Asian, Near Eastern, Central/South Asian, Sub-Saharan African, African American/Afro-Caribbean*.

These fields are serialized and URL-encoded automatically when sending the above GET requests. The Recommendation Lookup response is then filtered by the drug term produced by the rewritten query, ensuring only relevant dosing advice is retained. Population Frequency results will also be filtered using the race information extracted by Llama 3.3 70B, and it estimates the likelihood of each phenotype within the specified demographic.

The workflow demonstrates how combining query rewriting, structured extraction, and direct CPIC API calls can streamline clinical decision support and reduce manual lookup time.

3.4 Prompt-Guided Answer Synthesis

In the final stage, all parsed contexts are aggregated into a single evidence bundle $INFO_{\text{Retrieved}}$. This bundle comprises:

3.5 CPIC Guideline Extracted Information

Information derived from CPIC guidelines appears in two complementary forms. The first is structured tabular content—such as dosage adjustment matrices, phenotype–genotype associations, and mappings from diplotypes to predicted clinical function—which provides concise, standardized references that enable direct comparison across genes and drugs. The second is narrative text that explains clinical context, exceptions, limitations, and footnotes; these passages clarify nuances that cannot be fully captured in tables and are essential for interpreting edge cases, comedications, or population-specific considerations.

3.6 CPIC API Retrieved Information

The CPIC API supplies patient-specific, computable data that augment the guideline text. It exposes population frequency estimates that describe the expected prevalence of phenotypes across ancestral groups, helping clinicians anticipate variability among patients. It also returns gene–drug dosing recommendations and risk guidance conditioned on a patient's diplotype and the prescribed medication, thereby operationalizing the link between genotype information and actionable prescribing decisions.

We then employ a domain-expert prompt template that guides the reasoning model to produce a transparent, step-by-step answer grounded solely in $INFO_{\text{Retrieved}}$. Formally, the synthesis is performed as:

Response = OpenReasoning$(\text{Prompt}(Q_{\text{user}}, INFO_{\text{Retrieved}}))$

Q_{user} is the original patient question; $\text{Prompt}(\cdot)$ formats Q_{user} and $INFO_{\text{Retrieved}}$ into a structured instruction $\text{OpenReasoning}(\cdot)$ is the reasoning model that consumes the prompt and returns the final recommendation with a concise, model-generated justification.

This separation—fine-tuned rewriting for query normalization and a bespoke reasoning model for answer synthesis—ensures that our system delivers accurate, transparent, and source-faithful recommendations.

4 Experiments and Results

4.1 Dataset

We constructed two datasets for our experiments. The first is the fine-tuning dataset, where we fine-tune the Llama-3.3 Super 49B (Nemotron) model on 4,391 guideline QA pairs automatically generated with Gemini 2.5 Pro. Among these, 1,633 are *valid* samples with coverage in existing CPIC guidelines, while 2,758 are *invalid* samples without a corresponding guideline entry. This dataset is partitioned into training, validation, and test sets with a 70/15/15 split.

The second is the evaluation dataset, designed to assess QA capability on pharmacogenetics. It consists of 12 expert-curated multi-aspect ques-

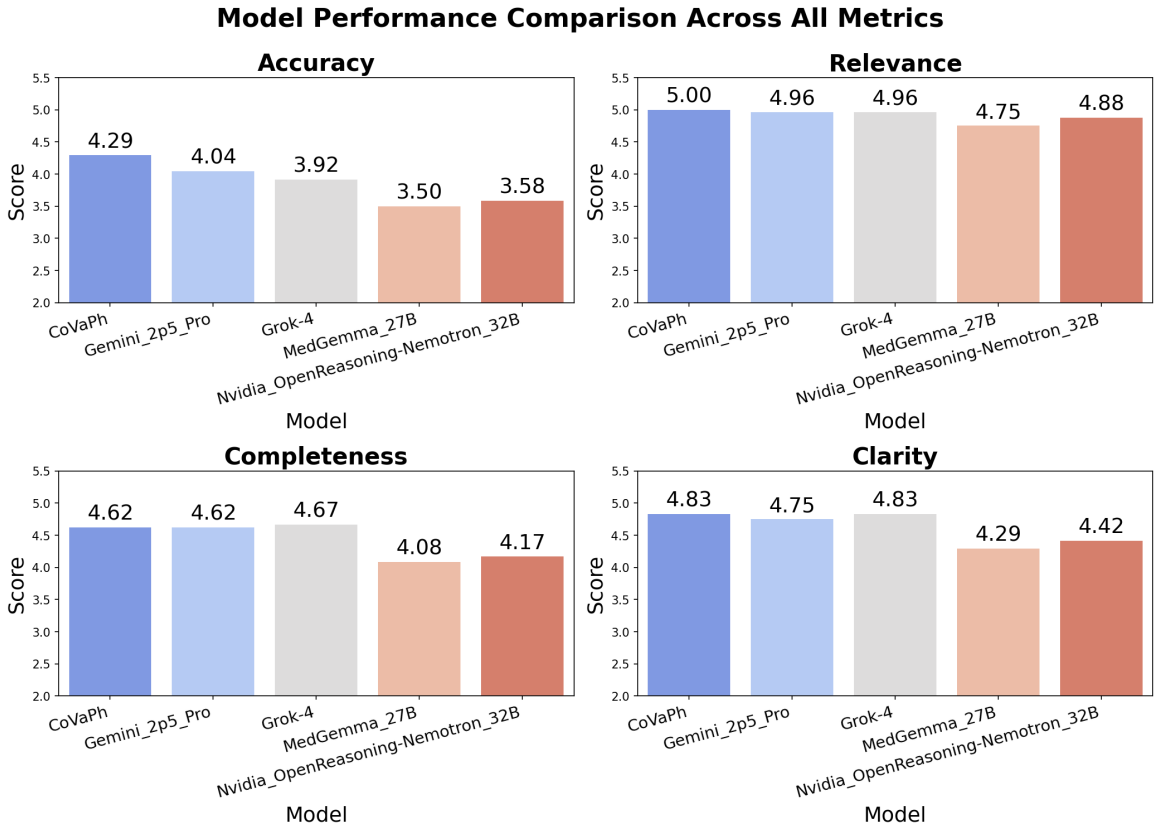

Figure 3: Comparisons of 4 metrics for all models.

tions, which require deep searches, and other 800-1000 more common clinical questions. For example, one case describes a 62-year-old African American female patient with atrial fibrillation and a history of ischemic stroke, who is initiated on warfarin but genotyping reveals she is a CYP2C9 poor metabolizer (*3/*3 genotype) and carries the VKORC1 –1639G>A variant (homozygous A/A). The task requires analyzing the heightened risk of bleeding, recommending dose initiation and adjustment strategies per CPIC guidelines, and outlining a monitoring plan including INR targets and the potential switch to direct oral anticoagulants.

Our fine-tune dataset is drawn from 28 CPIC guideline PDFs on CPIC official site: https://cpicpgx.org/guidelines/

4.2 Experimental Setup

All fine-tuning experiments were conducted on two NVIDIA A100 80 GB GPUs over approximately 12 hours, corresponding to three epochs, using the NeMo-run recipe `llama33_nemotron_super_49b_finetune_recipe` with PEFT LoRA. The optimization strategy employed the AdamW optimizer with a learning rate of 5×10^{-5} and a weight decay of 0.01. Learning rate scheduling followed a CosineAnnealing schedule with 100 warmup steps. For batching, each GPU processed a micro-batch size of 1 with gradient accumulation set to 1. The LoRA configuration used a rank of $r = 8$, scaling factor $\alpha = 16$, and dropout probability of 0.05. Training was carried out with BF16 mixed precision enabled to balance computational efficiency and numerical stability.

4.3 Evaluation Metrics

We evaluated all models on 12 held-out hard questions plus 819 simulated clinical questions based on recommendations from the CPIC guidelines, generated by frontier models, then selected by pharmaceutical professionals. Answers were carefully curated by pharmacists, then scored by an LLM-as-a-Judge on four dimensions (1–5 scale):

1. **Accuracy:** Factual correctness against CPIC sources.

2. **Completeness:** Coverage of relevant guideline points.

3. **Clarity:** Readability and structure of the response.

4. **Relevance:** Alignment with the question.

4.4 Baselines

We compare six configurations in our study. **Grok-4** and **Gemini 2.5 Pro** serve as strong proprietary LLM baselines. **MedGemma 27B Multimodal** is included as an off-the-shelf vision–language model. **OpenReasoning Nemotron-32B** is our base reasoning model without any CPIC context. Building on this base, CoVaPh augments Nemotron-32B with retrieval of guideline text and figures, and CoVaPh+CPIC API further injects structured signals from the CPIC API, specifically the *Recommendation* fields and population/phenotype frequency statistics.

4.5 Results and Ablation Study

We evaluate CoVaPh on four human-judged criteria—**Accuracy**, **Completeness**, **Relevance**, and **Clarity** and all results are shown in Figure 3. Across these metrics, CoVaPh performs comparably to, and on several metrics exceeds, strong proprietary baselines such as Grok-4 and Gemini 2.5 Pro. The consistently high **Relevance** scores indicate that mid- to large-scale models remain on topic and avoid off-prompt responses.

Our answer generator is **OpenReasoning Nemotron-32B**. Starting from a no-retrieval baseline, adding **Question-to-Query rewriting** and **ColQwen** embeddings to retrieve CPIC-guideline evidence (i.e., the full CoVaPh RAG pipeline) yields immediate gains: **Accuracy** $3.58 \rightarrow 4.13$, **Completeness** $4.17 \rightarrow 4.42$, and **Clarity** $4.42 \rightarrow 4.67$. Incorporating structured signals via the **CPIC API Search**—specifically *Phenotype Frequency* and *Recommendation*—further improves performance: **Accuracy** $4.13 \rightarrow 4.29$, **Completeness** $4.42 \rightarrow 4.63$, and **Clarity** $4.67 \rightarrow 4.83$. These ablations indicate that precise query reformulation, dense retrieval over guidelines, and structured pharmacogenomic knowledge jointly and materially improve final answer quality.

In our ablation studies, we examine 6 settings of CoVaPh - (a) bare Nemotron super 49B without API Search, CPIC Guidelines, and Query Re-writer; (b) without CPIC Guideline and Query Re-writer; (c) without API Search and Query Re-writer; (d) without Query Re-writer; (e) without API Search; (f) using all tools (full pipeline), with two llm judge models, GPT-5 medium and Grok-4-fast with tools. Both LLMs show almost identical increasing trends when adding more data and tools to the pipeline, which indicates the evident benefits of augmenting our models with proper tools.

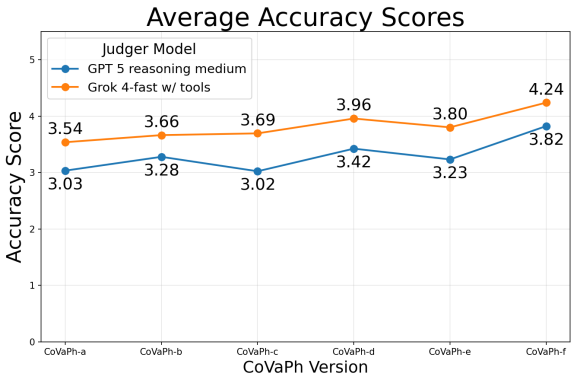

Figure 4: Ablation Study for Accuracy. 6 flavors - (a) no API Search, CPIC Guidelines, and Query Re-writer; (b) no CPIC Guideline and Query Re-writer; (c) no API Search and Query Re-writer; (d) no Query Re-writer; (e) no API Search; (f) all (full pipeline).

5 Conclusion

This paper introduces CoVaPh, a multi-agent, tool-augmented framework designed to address the complex and error-prone challenge of pharmacogenetic reasoning. We have demonstrated that by teaming up several specialized agents with an accessible 32B open-source model, it is possible to achieve state-of-the-art performance in a high-stakes medical domain. Our approach integrates a fine-tuned "Experienced Query Rewriter" that translates ambiguous clinical questions into precise, structured queries, which then drive a hybrid retrieval pipeline. This pipeline uniquely fuses a multimodal RAG system capable of interpreting visual data from PDF guidelines with live API calls to the CPIC database for real-time, patient-specific information.

Our key contributions include the design of this novel multi-agent architecture on a mid-sized model, the development of a hybrid retrieval system that overcomes the limitations of standard RAG, and a safety-oriented fine-tuning process that teaches the model to deny unsupported queries, thereby mitigating hallucinations. The empirical results are clear: CoVaPh surpasses the accuracy of leading proprietary models like Gemini 2.5 Pro and Grok-4 by a significant margin. By automating this critical task, CoVaPh presents a tangible pathway to alleviating the burden on healthcare profession-

als, enhancing patient safety, and democratizing access to personalized medicine through scalable, open, and high-fidelity AI systems.

Acknowledgments

We thank the generous supports of funding and computes from the Miin Wu School of Computing at NCKU.

References

A. Asai, Z. Wu, Y. Wang, and et al. 2023. SELF-RAG: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511*.

L. Gao, X. Ma, J. Lin, and J. Callan. 2022. Precise zero-shot dense retrieval without relevance labels. *arXiv preprint arXiv:2212.10496*.

S. Gehrmann, S. Douglas, S. Hyland, and et al. 2024. PGxQA: A benchmark for question answering on pharmacogenomic guidelines. *arXiv preprint arXiv:2404.14498*.

P. Lewis, E. Perez, A. Piktus, and et al. 2020. Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Advances in Neural Information Processing Systems (NeurIPS)*.

Z. Li, J. Wang, Z. Jiang, and et al. 2024. DMQR-RAG: Diverse multi-query rewriting for retrieval-augmented generation. *arXiv preprint arXiv:2411.13154*.

H. Liu, C. Li, Q. Wu, and Y. J. Lee. 2023. Visual instruction tuning. *arXiv preprint arXiv:2304.08485*.

J. Mao, Y. Liu, J. Ma, and et al. 2023. Query rewriting for retrieval-augmented large language models. *arXiv preprint arXiv:2305.14283*.

Mercer. 2025. 2025 global talent trends study. Technical report, Mercer LLC.

NVIDIA. 2024. NVIDIA nemotron nano VL. `https://huggingface.co/nvidia/Llama-Nemotron-Nano-VL-8B`.

Y. Qin, S. Liang, Y. Wen, and et al. 2023. ToolLLM: Facilitating large language models to master 16000+ real-world APIs. *arXiv preprint arXiv:2307.16789*.

Qwen Team. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2411.16223*.

X. Ren, L. Wang, and Y. Yang. 2024. A surprisingly simple yet effective multi-query rewriting method for dense retrieval. *arXiv preprint arXiv:2406.18960*.

T. Schick, J. Dwivedi-Yu, R. Dessì, and et al. 2023. Toolformer: Language models can teach themselves to use tools. *arXiv preprint arXiv:2302.04761*.

M. Sharma, S. Bhalla, K. Dalal, and et al. 2024. Quicker: A question answering system for restrictive clinical practice guidelines. *arXiv preprint arXiv:2405.10174*.

W. Shi, X. Chen, H. Wang, and et al. 2024. Corrective retrieval augmented generation. In *The Twelfth International Conference on Learning Representations (ICLR)*.

Q. Wu, G. Bansal, J. Zhang, and et al. 2023. AutoGen: Enabling next-gen LLM applications via multi-agent conversation. *arXiv preprint arXiv:2308.08155*.

G. Xiong, Q. Jin, X. Wang, and et al. 2024. Improving retrieval-augmented generation in medicine with iterative follow-up questions. *arXiv preprint arXiv:2408.00727*.

S. Yao, J. Zhao, D. Yu, and et al. 2022. ReAct: Synergizing reasoning and acting in language models. *arXiv preprint arXiv:2210.03629*.

S. Yin, C. Fu, S. Zhao, and et al. 2023. A survey on vision-language-action models. *arXiv preprint arXiv:2311.17143*.

L. Zhang, B. Li, S. Li, and et al. 2023. A survey on vision-language models: Towards comprehensive representation. *arXiv preprint arXiv:2306.09243*.

A Appendix

A.1 Additional Ablation

We include additional ablation plots for clarity, relevance, and completeness in Figure 5 in this appendix.

A.2 Prompt Templates

We list all prompts verbatim. Placeholders are in braces (e.g., `{context}`).

Final-Answer-Prompt

```
As the world's smartest and most knowledgeable healthcare
      professional/pharmacist,
you apply the most rigorous reasoning capabilities to any
      pharmacogenetics-related
queries/questions presented to you to understand them using
      first principles
thinking to the core of every question meticulously, and give
      your best answers.
Always perform to your highest standard with no reservation.

Please read the following pharmacogenetic question with the
      provided contexts carefully
and provide your step-by-step reasoning process for each
      question to reach your final answers.
------------------------------
{context}
------------------------------
Related information from CPIC will also be given:
1. Phenotype frequency
2. Recommendation
------------------------------
{CPIC}
------------------------------
Pharmacogenetic Question:
------------------------------
{user}
------------------------------
```

Figure 5: Ablation of 3 metrics except accuracy for all models.

CPIC-Guideline-Extraction-Prompt

```
Please do the following 1. first then 2.:

1) Identify table contents if any (actual tables with rows/
      columns, not just titles)
   in the image and extract them with associated footnotes (a,
      b, c, d, etc.).
2) Extract the body of text (non-tabular) such as narrative
      contents and updates,
   and paste them after the extracted tabular contents.
```

Race-Extraction-Prompt

```
You are an expert pharmacogeneticist.
Your task is to identify the patient's race from the clinical
      text.

Your answer must be exactly one of:
{{race_list}}

If race is not mentioned or ambiguous, respond "Unknown".

Input:
-------------------------------
{{question}}
-------------------------------
Answer:
```

Allele-Number-Extraction-Prompt

```
You are given a question that mentions gene symbols. Your task
      is to:

1. Identify the gene symbols in the question.
2. Count how many times each allele appears for that gene.
3. Return the result as a JSON object where:
   - Each top-level key is the gene symbol.
   - Each value maps allele names (strings) to integer counts.

Example output:
{"VKORC1": {"A": 2}}

Respond with the JSON object only. No extra text.

Question:
"""{{question}}"""

Answer:
```

Query-Rewrite-Prompt

```
You are a well-informed pharmacogenomics query parsing
  assistant.
You'll follow a sequence of rules with input given, then
  generate the output as stated below.

Input
- User question

Output
- If a valid guideline for a pair of drug and gene
  mentioned is found -> JSON object with four keys
- If no guideline meets the strict criteria -> the plain
  string "No CPIC guideline information available."

Keys for the JSON object
- "Drug Name"
- "Gene Name"
- "CPIC Guideline Name"
- "Content to Search"

Rules:
1. Analyze the User's Question: Deconstruct the query to
  understand its components.
2. Entity Extraction:
   - Drug and Gene Names must come from the question; if
  both are absent, return "No CPIC guideline information
  available."
   - Multiple drugs and/or genes may be present.
3. CPIC Guideline Matching:
   - Consider all extracted drugs and genes.
   - Match all pairs that have an existing CPIC Guideline
  (title explicitly mentions both or clearly encompasses
  the pair).
4. Content to Search:
   - Must mention both drug and gene, <= 150 words.
   - Justify relevance and specify the exact information
  to retrieve.
5. Format the Output:
   - Use: "Drug Name: [...], Gene Name: [...], CPIC
  Guideline Name: [...], Content to Search: [...]"
   - Place all such dictionaries into a single Python list.


Example:
Input: "What is the relationship between ivacaftor and
  CFTR?"
Output: [{
  'Drug Name': 'ivacaftor',
  'Gene Name': 'CFTR',
  'CPIC Guideline Name': 'Clinical Pharmacogenetics
  Implementation Consortium (CPIC) Guidelines for
  Ivacaftor Therapy in the Context of CFTR Genotype (March
   2014).pdf',
  'Content to Search': 'Recommended ivacaftor dosage for
  patients with the CFTR G551D genotype.'
}]

If there is no match:
Output: "No CPIC guideline information available."
```

Evaluation-Prompt

```
As the world's smartest and most knowledgeable healthcare
      professional/pharmacist,
you apply the most rigorous reasoning capabilities to any
      pharmacogenetics-related queries/questions.
Please read the attached model responses by 8 different models
       to 1 clinical PGx question and evaluate them.

Scoring (1-5, increments of 0.5):
- Accuracy: factual correctness and up-to-date info.
- Relevance: alignment with the question asked.
- Completeness: coverage of necessary details.
- Clarity: organization and readability.

Return JSON:
{
  "Current Question Number": {{Question Number}},
  "Scores": {
    "Model 1": {"Accuracy": X, "Relevance": Y, "Completeness":
       Z, "Clarity": W},
    ...
    "Model 8": {"Accuracy": X, "Relevance": Y, "Completeness":
       Z, "Clarity": W}
  }
}

Include:
Question: {{Question}}
Reference Answers: (as provided)
Model 1's Response: {{Model 1 Response}}
...
Model 8's Response: {{Model 8 Response}}
Ground Truth: {{Ground Truth}}
```

Response-Generation-Prompt

```
As the world's smartest and most knowledgeable healthcare
      professional/pharmacist,
you apply the most rigorous reasoning capabilities to any
      pharmacogenetics-related
queries/questions presented to you to understand them using
      first principles thinking,
and give your best answers. Always perform to your highest
      standard with no reservation.

Please read the following pharmacogenetic question with the
      provided contexts carefully
and provide your step-by-step reasoning process to reach your
      final answers.
-------------------------------
{context}
-------------------------------
Related information from CPIC:
1. Phenotype frequency
2. Recommendation
-------------------------------
{CPIC}
-------------------------------
Pharmacogenetic Question:
-------------------------------
{user}
-------------------------------
```