Joint Contrastive-Generative Architecture for Domain-Specific Multi-Agent Orchestration in Clinical Decision Assistance

Mrishika Nair, Georgios Ioannides*, Jeremy Roghair*, Rohit Thekkanal, Huan Song, Hannah Marlowe, Sharlina Keshava

ABSTRACT

Multi-agent clinical decision support systems—i.e., systems where specialized modules collaborate on tasks such as patient history retrieval, literature search, and diagnostic inference—typically require separate embedding and generation models, creating computational and operational overhead[10] that limits deployment in resource-constrained healthcare settings. This work presents Joint-CGA, a unified embedding-generation architecture fine-tuned from GritLM and distribution-aware contrastive loss optimization for clinical reasoning tasks. Empirical evaluations on the MedReason dataset show a **7%** increase in retrieval accuracy and **18%** improvement in generation quality over Llama-3.1-8B, with evidence of cross-task learning between embedding and generative objectives. Deployed within a multi-agent clinical decision workflow using the Amazon Strands SDK, the unified architecture achieves over **60%** speed improvement, consistent with gains reported for the base GritLM framework [**?**], while eliminating inter-model handoff latency and reducing deployment complexity to a single endpoint. These results position unified embedding-generation architectures as a practical foundation for scalable, accessible multi-agent AI systems in healthcare.

KEYWORDS

Joint Contrastive-Generative Architecture (Joint-CGA), Multi-agent orchestration, Low Rank Adaptation (LoRA), Semantic representation learning

1 INTRODUCTION

The World Health Organization estimates a global shortage of 15 million healthcare workers, with 95% of this shortfall concentrated in low- and middle-income countries—a gap projected to remain at 10 million by 2030 [4]. This persistent deficit has driven growing interest in AI-based clinical decision support as a means to extend the reach of existing providers and improve access to timely, evidence-based care.

The development of Large Language Model (LLM) based multi-agent systems—where multiple LLM instances are assigned distinct roles and coordinated to complete a shared task—mark an important step toward collaborative problem-solving that exceeds the capacity of any single model. By distributing roles such as retrieval, generation, and planning, these systems offer a flexible framework for knowledge access, adaptive decision-making, and coordination between different tasks. Their adoption is growing in domains where challenges require heterogeneous capabilities and scalable orchestration, particularly in healthcare where clinical decision support can significantly impact patient outcomes and care quality [19][21][22]. Yet current Artificial Intelligence (AI) architectures present a critical paradox: while designed to address healthcare access gaps, their computational complexity and infrastructure requirements create deployment barriers that exclude the resource-constrained facilities serving the populations most in need.

Retrieval-Augmented Generation (RAG) has been a dominant architecture for providing LLMs in external knowledge. A RAG pipeline relies on two separate components—a retriever and a generator. The retriever encodes the user's query into an embedding space (i.e. vector space) and performs semantic search over a vector database of pre-chunked document embeddings via similarity metrics (e.g. cosine similarity) that aim to capture semantic meaning; the generator then integrates these retrieved passages with the original prompt to produce a final response. This modularity improves retrieval accuracy, but introduces inefficiencies at the seams between components: each handoff adds latency, computational overhead, and a risk of information loss [15]. As multi-agent systems scale to broader domains, these inefficiencies compound, making orchestration increasingly complex and reducing downstream inference reliability.

Throughout this work, *reasoning* refers to structured, multi-step text generation: at each step, the language model head produces a distribution over the vocabulary and selects the next token based on patterns learned during training. The resulting outputs may exhibit logical coherence, but this coherence emerges from token-level pattern matching. In retrieval contexts, it denotes multi-hop semantic matching beyond surface similarity; in generative contexts, it denotes producing structured clinical text by sequentially selecting tokens conditioned on the input and all previously generated tokens. This distinction underscores that all model outputs require clinician verification.

Recent work addresses this by exploring unified architectures that combine retrieval and generation within a single model. The GRIT (Generative Representational Instruction Tuning) framework [13] unifies two previously separate training paradigms—(1) **generative instruction tuning**, where the model learns to generate responses using causal attention, and (2) **representational instruction tuning**, where the model learns to produce meaningful embeddings using bidirectional attention and mean pooling over final hidden states. GRIT applies distinct instruction-specific loss functions to separate these behaviors but keeps the underlying model weights shared. This allows a single model to generate embeddings and produce text outputs without architectural switching, reducing the operational complexity of multi-model RAG pipelines and supporting flexible multi-turn interaction.

Building on this foundation, this work introduces Joint-CGA, a unified embedding-generation model fine-tuned from GritLM to support domain-specific retrieval and generation in clinical contexts. The approach uses low-rank adaptation (LoRA)[6] fine-tuning and contrastive loss functions — including triplet loss, InfoNCE, and

*Equal contribution.

distributive loss with careful hyperparameter tuning to improve retrieval precision while maintaining strong generative performance for clinical decision support. This consolidation yields four concrete advantages for multi-agent systems: (1) *efficiency*, by reducing cumulative latency across retrieval and generation stages; (2) *coherence*, by maintaining contextual representations between retrieved knowledge and generative outputs; (3) *operational simplicity*, by removing redundant handoffs and eliminating coordinated multi-model deployment; and (4) *adaptability*, by supporting modular multi-turn workflows without separate retrievers and generators. Beyond these performance gains, Joint-CGA provides an efficient foundation for multi-agent orchestration, enabling context-aware retrieval, generation, and decision-making while minimizing model switching and resource overhead. Reduced latency and computational footprint were achieved, making AI-powered clinical triage accessible to resource-constrained settings, helping reduce diagnostic delays which can improve care quality. To assess its effectiveness for agentic AI frameworks, the fine-tuned model is integrated with Amazon Strands agents and benchmarked on a clinical decision assistance workflow involving patient history retrieval, literature search, clinical inference, and evidence-based recommendations.

This work contributes to (i) the design of a unified embedding-generation architecture, (ii) methods for enhancing domain-specific text generation and retrieval precision through language modeling and contrastive loss optimization, (iii) an empirical demonstration of the framework's impact on multi-agent orchestration efficiency and scalability, and (iv) validation of how architectural efficiency enables more efficient access to AI-powered healthcare.

2 RELATED WORK

Dense retrieval - Modern retrieval-augmented systems rely heavily on dense retrieval, in which the queries and documents are embedded into a shared latent space for semantic matching. Early dense retrievers, particularly bi-encoder models based on pretrained language models (PLMs), outperformed classical sparse methods such as BM25 by capturing deeper semantic relationships[26]. Unsupervised pre-training techniques, such as the Inverse Cloze Task (ICT), improve zero-shot and open-domain retrieval by training models to predict document context from randomly sampled query spans, this enables strong performance without task-specific labels.

Contrastive learning is now a key component in training robust dense retrievers [7]. By bringing semantically relevant query–document pairs closer in the latent space while pushing unrelated or negative pairs apart. Contrastive objectives can improve embedding quality and model discrimination. Positive pairs are often constructed from different spans of the same document, while negative examples are sampled from within or across batches, encouraging fine-grained semantic differentiation. Techniques like independent cropping and MoCo (Momentum Contrast) stabilize learning across large negative pools and batches, as implemented in Contriever.

These unsupervised pretraining and contrastive learning techniques produce high-quality, generalizable embeddings that form the foundation for retrieval models.

Reasoning-Oriented Retriever Models - While dense retrieval works well for semantic similarity, it struggles with reasoning-heavy tasks. Reasoning-oriented retrievers are designed to handle queries that require multi-hop matching, relational evaluation, or compositional retrieval across multiple concepts. ReasonIR-8B [20] addresses this by generating synthetic, reasoning-intensive queries along with hard negatives. This approach achieves new state-of-the-art results on the reasoning-intensive BRIGHT dataset and improves performance on benchmarks such as MMLU and GPQA, particularly when applied in retrieval-augmented generation scenarios. Methods like structured query expansion and Chain-of-Thought (CoT) reasoning further improve query quality. This allows the retriever to provide relevant evidence that LLMs can use to generate accurate and coherent answers.

LongRAG [8] and Search-o1 [11] build on this by grouping long documents and refining evidence during retrieval. LongRAG groups related documents into 4K-token units to preserve semantic continuity, while Search-o1 uses an agentic "Reason-in-Documents" module to refine retrieved evidence, reducing noise and enhancing logical flow and confidence in complex multi-hop retrieval tasks.

Unified Architectures - Embedding and generation models evolved along separate tracks. Embedding models first focused on word representations, later extending to sentence-level encoders like InferSent and SBERT [16], but still required different models for symmetric and asymmetric tasks to achieve strong performance. Generative models, on the other hand, were initially tailored to single tasks such as translation or question answering, before large-scale pretraining and instruction tuning enabled broad generalization across a diverse set of tasks.

Unified architectures aim to bridge these two tracks and combine embedding and generative reasoning in a unified model, reducing the latency and integration overhead of modular pipelines. ULLME [12] introduces Generation-augmented Representation Learning (GRL) to align embeddings with generative relevance, while GRITLM [13] integrates embedding and generative instruction tuning, handling both symmetric and asymmetric tasks within one framework. GRITLM also accelerates RAG by over 60% while maintaining strong performance across embedding and generative benchmarks. These models demonstrate the potential of combining high-quality representation and generation within a single, efficient framework.

Multi-Agent Pipelines - Beyond single retriever–generator pipelines, multi-agent orchestrated systems extend RAG by decomposing tasks into specialized roles such as query disambiguation, information extraction, and synthesis.

MA-RAG [14] coordinates multiple agents — Planner, Step Definer, Extractor, and QA — to resolve ambiguities and multi-hop reasoning challenges. It operates without fine-tuning, using on-demand agent invocation and chain-of-thought prompting to progressively refine retrieval and synthesis. RAG-KG-IL [24] integrates retrieval with knowledge graphs and incremental learning, supporting continuous knowledge updates, structured reasoning, and lower computational overhead. It significantly reduced hallucination rates and improved completeness and accuracy over GPT-4o and a RAG-only baseline, showing that modular multi-agent design with specialized agents and structured knowledge integration provides complementary strengths.

Despite these benefits, coordinating multiple agents introduces challenges. Modular pipelines can suffer from error propagation and latency, as handoffs between agents may amplify noise or inconsistencies. For instance prior works report a 28.6% accuracy

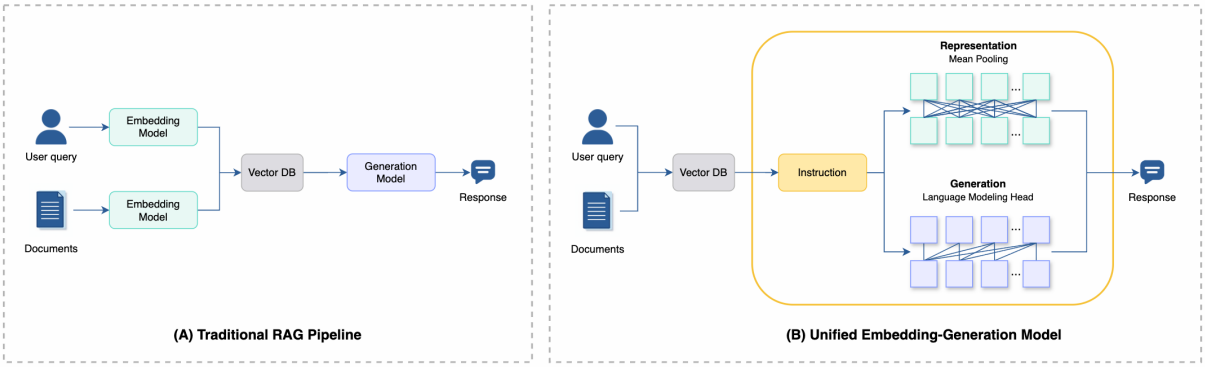

Figure 1: Comparison of retrieval–generation paradigms. (A) RAG pipeline with separate embedding and generation models—user queries and documents are embedded, stored in a vector database, and then passed to a separate LLM for response generation. (B) Unified embedding-generation model, where a single architecture performs both representation and generation. Instructions guide the model to switch between representation learning (bidirectional attention over the input with mean pooling over the final hidden states) and generation (causal attention with a language modeling head to predict the next tokens).

drop for Flan-T5 when paired with Contriever [2]. Hallucinations also remain a persistent problem, appearing as input-conflicting, context-conflicting, or fact-conflicting outputs [25]. These limitations motivate unified orchestration frameworks that can preserve interpretability while reducing overhead.

3 METHODOLOGY

3.1 Dataset

This work uses the UCSC-VLAA MedReason[23], a large-scale medical dataset designed for explainable problem-solving, where clinical question-answer pairs are encoded as structured chains of inference ("thinking paths") derived from a medical knowledge graph. It contains 32,682 samples of medical reasoning tasks, with each sample comprising a question, an answer, a reasoning explanation, and four answer options. For fine-tuning experiments, a constrained split of 1,000 samples is used for training, 200 for validation, and 100 for evaluation.

Component	*Mean ± Std*	*Median*	*Range*
Question	50.3 ± 55.7	21	1-1987
Answer	46.7 ± 55.5	34	1-1263
Reasoning	370.9 ± 69.1	372	148-695

Table 1: Overview of the MedReason dataset and annotation schema

3.1.1 Preprocessing. Two dataset formats are prepared to support the generative and embedding objectives.

- Generative Data format: Each sample is stored as a single text field containing the medical query concatenated with its evidence-grounded answer.
- Embedding Data format: Each sample includes a query, the positive answer with the explanation, and three negative options. This format supports contrastive loss fine-tuning, where the model promotes the query alignment with the positive instance and maximizes separation from the negative instances.

3.1.2 Synthetic Query Generation. To enhance training diversity and simulate diagnostic variability, situational queries are generated by transforming simple, fact-based medical questions into context-rich clinical scenarios or patient vignette with symptoms, history, etc (Fig. 2).

These queries are produced using a reasoning-intensive document-to-query procedure inspired by ReasonIR (Shah et al., 2024). The doc-to-query pipeline samples prompts, queries Meta-Llama-3.1-70B-Instruct, and generates reasoning-focused queries paired with positive and hard-negative examples retrieved via BM25 ranking function.

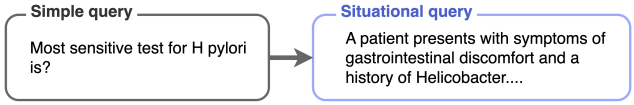

Figure 2: Transformation of simple medical queries into contextually-rich situational queries with patient symptoms, history, and real-world decision-making context.

3.2 Fine-tuning

The model is fine-tuned for a medical-domain task using a supervised fine-tuning (SFT) paradigm on a labeled dataset. This approach allows the model to learn domain-specific reasoning patterns while maintaining factual alignment with clinical terminology[9]. The fine-tuning pipeline follows two common steps for both objectives: hyperparameter optimization to ensure generalizable and stable

adaptation on sensitive clinical text, and LoRA-based parameter-efficient tuning to preserve pretrained capabilities while introducing targeted domain specialization.

Hyperparameters	
Learning Rate	2e-5
Temperature	0.02
Warmup Ratio	0.06
Weight Decay	0.01
Precision	BF16

LoRA Settings	
Rank	32
Scaling Factor	128
Dropout	0.1
Layers	Attn & FFN

Table 2: Fine-tuning hyperparameters and LoRA settings.

3.2.1 Generative Instruction Fine-Tuning. For domain-specific medical inference, the model undergoes instruction fine-tuning on GritLM. The model is trained to generate responses conditioned on instruction prompts, leveraging causal attention for next-token prediction[17]. LoRA adapters enabled efficient adaptation of the generative head while preserving the base model's performance. With this configuration, the model generates clinical responses by selecting tokens sequentially from the language model head, conditioned on learned medical associations from the training data.

3.2.2 Embedding Task Fine-Tuning. To improve retrieval performance we experimented with three contrastive loss functions. In each case, an *anchor* is the encoded query q_i, a *positive* is the embedding of its correct answer p_{y_i}, and *negatives* are embeddings of the remaining incorrect answer options $\{p_j\}_{j \neq y_i}$ within the batch.

Triplet Loss. It minimises the distance between an anchor and its positive while ensuring that every negative lies at least a margin $m > 0$ further:

$$\mathcal{L}_{\text{triplet}} = \max\big(d(q_i, p_{y_i}) - d(q_i, p_n) + m,\ 0\big),$$

where d denotes Euclidean distance in the normalised embedding space and m is the margin. This loss enforces fine-grained local separation, which is valuable for medical datasets where semantically similar terms must be distinguished. But it is sensitive to the triplet selection strategy, and requires exactly one positive per anchor which conflicts with the dataset structure of one positive paired with three negatives.

InfoNCE Loss. It improves robustness by contrasting each positive pair against all other samples in the batch [**?**]. Given embeddings z_i for query q_i and z_j for its positive p_{y_i}, the loss for a batch of size B is:

$$\mathcal{L}_{\text{InfoNCE}} = -\frac{1}{B}\sum_{i=1}^{B} \log \frac{\exp\big(\text{sim}(z_i, z_j)/\tau\big)}{\sum_{k=1}^{B} \exp\big(\text{sim}(z_i, z_k)/\tau\big)},$$

where the denominator sums over all B keys in the batch. InfoNCE benefits from multiple in-batch negatives, but its pointwise pairing limits its ability to shape the global batch-level similarity distribution, showing weaker separation between fine-grained medical concepts.

Distributive Loss. Pushes the model to maximize the similarity between each query q_i and its corresponding positive passage p_{y_i}, while minimizing similarity with all other passages in the batch. By considering the entire batch as negatives, it captures global similarity structure rather than just individual pairs[3]. It improves semantic clustering across a large corpora and provides richer training signals for generalization.

$$L_{\text{Distributive}} = \frac{1}{B}\sum_{i=1}^{B} -\log \frac{\exp\left(q_i^\top p_{y_i}/\tau\right)}{\sum_{j=1}^{N\times W} \exp\left(q_i^\top p_j/\tau\right)}$$

Embedding fine-tuning uses bidirectional attention and mean pooling over the final hidden states to generate robust vector representations. LoRA adapters are used to enable parameter-efficient adaptation by introducing low-rank perturbations to the attention and projection weights, this avoids the need for full model retraining.

3.3 Evaluation

Separate evaluation metrics are used for the generative and embedding tasks to assess each fine-tuning objective. The goal of the experiments is to quantify the performance gains achieved by the unified Joint-CGA architecture over the baseline models in both information retrieval and conditional text generation.

3.3.1 Generative Evaluation. The generative component is evaluated using a Large Language Model-as-a-Judge (LLMaaJ) framework, in which a frontier model—Claude 3.5 Sonnet—provides structured comparative judgments over system outputs. Responses generated by each model are assessed along four predefined axes:

- precision in the use of medical terminology
- factual accuracy relative to established clinical knowledge
- completeness of the generated response
- linguistic clarity

Each model generates responses constrained to 500 tokens to ensure consistent evaluation length across models. Evaluations are conducted for GritLM, Llama-3.1-8B-Instruct, Joint-CGA (Generative Mode), and Joint-CGA (Unified Mode).

The final scores quantify the degree of factual grounding, coherence of generated text, and adherence to domain-specific medical language exhibited by each model.

Discussion on LLM-as-a-judge

The LLM-as-a-Judge (LLMaaJ) paradigm uses a high-capacity language model as an automated evaluator to score model outputs based on a standardized, rubric-guided criteria[5]. This approach has gained traction because its consistent, scales more efficiently than manual annotation, and approximates expert judgment for complex tasks such as medical question answering.

But this method does have some limitations: (i) Results may not be exactly reproducible, as frontier models evolve over time and updates can alter scoring behavior. (ii) Judge models exhibit inherent biases, including stylistic and reasoning biases, preferences for certain answer patterns, which may favour outputs that resemble the judge's training distribution. (iii) The judge LLM lacks access to ground-truth facts and evaluates outputs primarily for textual plausibility.

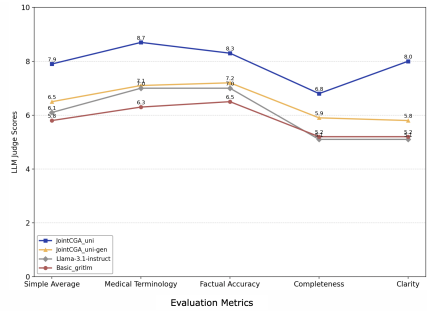

Figure 3: Comparative LLM-as-a-Judge evaluation across models. Joint-CGA (Unified) achieves the highest mean score across all evaluation axes.

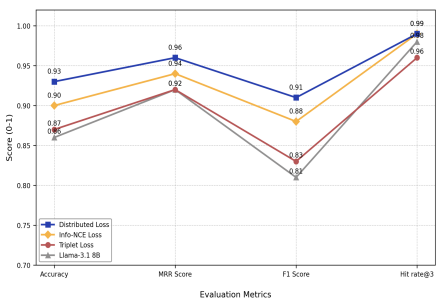

Figure 4: Embedding fine-tuning results under different contrastive loss functions.

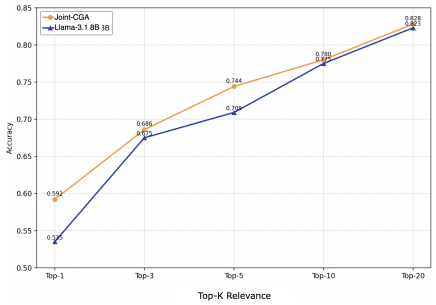

Figure 5: Retrieval accuracy comparison across multiple Top-K thresholds. Joint-CGA demonstrates significant Top-1 and Top-3 gains over base models.

To mitigate these limitations, the exact judge model version and scoring rubric is specified and mean scores across multiple evaluation runs are reported. These measures improve the reliability and generalisability of the results, although full reproducibility remains constrained by the evolving nature of the judge model.

3.3.2 Embedding Evaluation. To assess the impact of embedding fine-tuning, the models were evaluates across four retrieval-oriented metrics: Accuracy, Mean Reciprocal Rank (MRR), F1 Score, and Hit Rate@3. Each metric captures a distinct aspect of retrieval performance -

- Accuracy measures the proportion of queries for which the single most relevant document is retrieved
- MRR evaluates ranking quality by averaging the reciprocal ranks of the first relevant result across a set of queries, emphasizing early retrieval of correct items
- F1 balances precision and recall to evaluate overall retrieval effectiveness
- Hit Rate@3 quantifies the proportion of relevant documents retrieved within the top three candidates.

Additionally, **Top-K relevance analysis** ($K \in \{1, 3, 5, 10, 20\}$) is conducted to evaluate each model's ability to retrieve the most relevant documents at varying retrieval depths. Evaluating multiple K values helps measure how each model's relevance scales as more candidates are considered.

- $K = 1$ tests the model's ability to identify the single most relevant item.
- $K = 3$ and $K = 5$ capture short-range ranking accuracy typical in RAG contexts.
- $K = 10$ and $K = 20$ evaluate recall over broader retrieval windows.

These specific K values balance granularity and interpretability in retrieval benchmarking, allowing comparison with prior RAG and Information Retrieval literature.

3.4 Results

This section presents the experimental results across generative and embedding tasks, followed by the performance gains in the unified model.

3.4.1 Generative Fine-Tuning Performance. Figure 3 summarizes the comparative performance of the models under the LLM-as-a-Judge evaluation. The **Joint-CGA Unified Mode** achieved the highest overall scores, outperforming the other models across all four evaluation axes.

Metric-wise Trends:

- Medical Terminology: Joint-CGA-unified achieves the highest score (8.7) and the other variants show modest improvement over the base GritLM model. This suggests that the fine-tuned model has a strong specialization in domain-specific understanding.
- Factual Accuracy: Joint-CGA-unified (8.3) and Joint-CGA-generative (7.2) maintain strong performance, indicating reliable precision in generating medically accurate responses.
- Completeness and Clarity: There is a dip in completeness, since the answer was restricted to 500 tokens. But Joint-CGA-unified still outperforms others, shows that the fine-tuned model was able to answer the question well within 500 tokens while the other struggle with comprehensive and clearly when contained.

The overall better performance in the unified model shows the evidence of **cross-task learning** when fine-tuned for both the objectives together.

3.4.2 Embedding Optimization and Loss Function Analysis. Figure 4 shows the comparative performance of the four embedding objectives across Accuracy (HitRate@1), MRR, F1, and HitRate@3.

The Distributed Loss objective achieves the highest scores in all metrics: **0.935 Accuracy**, **0.961 MRR**, **0.914 F1**, and a near-perfect **0.99 Hit Rate@3**. These results suggest that explicitly modeling the full batch-level similarity distribution promotes more globally coherent semantic clustering within the embedding space, improving both top-rank precision and retrieval robustness at larger k.

InfoNCE loss objective benefits from multiple in-batch negatives, but its optimization is limited by pointwise pairing and lacks the ability to shape the batch-level distribution. This leads to weaker separation between fine-grained medical concepts. Triplet Loss yielded stable but lower performance due to sampling sensitivity, and reliance on informative anchor–positive–negative triplets,

which goes against the dataset structure - one positive paired with three negatives per query.

But all the contrastive loss objectives substantially outperform the baseline Llama model. In particular, the distribution-aware contrastive objective produces the strongest gains, achieving a 7% absolute improvement in accuracy (HitRate@1) over the baseline.

3.4.3 Retrieval and Ranking Performance. Figure 5, reports retrieval accuracy across Top-K thresholds ($K \in \{1, 3, 5, 10, 20\}$) from 1,000 documents corpus. The unified Joint-CGA model outperforms the base Llama-3.1-8B model at all K values, with the largest improvement at **Top-1 (+5.9%)** and a smaller gain at **Top-3 (+1.1%)**. The performance gap decreases as K increases, and both models tend to converge at higher K (Top-10, Top-20). This indicates that unified training majorly enhances the highest-ranked retrieval precision, which is an essential property for inference-driven and context-sensitive retrieval tasks.

3.4.4 Unified Performance Summary. Overall, the *Joint-CGA Unified Fine-Tuning Framework*—which integrates both generative and embedding optimization objectives—produced consistent improvements across all evaluated dimensions. These results suggest that aligning shared semantic representations between the embedding and generative heads enhances both retrieval precision and the factual quality of generated clinical text. The synergy between these components enables more robust retrieval-augmented generation and domain-specific precision in downstream medical applications.

3.5 Agent Integration

This section describes the architectural design and deployment rationale for integrating Joint-CGA into a multi-agent clinical decision support pipeline. The workflow presented here constitutes a system design contribution: it demonstrates how a unified embedding-generation model can serve as the core information retrieval and generation backbone for a coordinated multi-agent system, and characterizes the qualitative properties-interpretability, modularity, and operational efficiency, that this architecture enables. Quantitative end-to-end evaluation of the full agentic pipeline is planned as future work (Section 6).

3.5.1 Design Rationale: Modularity over Monolithism. Monolithic systems that handle retrieval and generation in a single context present two problems.

- Growing context length degrades reliability as patient history, retrieved literature, and generated inference traces accumulate in a single context window, attention quality degrades and critical information can be lost.
- Accountability is obscured when a single agent both retrieves evidence and makes diagnostic suggestions, it becomes difficult to audit which information drove which decision.

Multi-agent decomposition addresses both problems. Joint-CGA is the underlying model in both the agents enabling aligned semantic representations without maintaining separate embedding models. The multi-agent workflow has a deliberate division between specialist and generalist agents. Generalist agents (Claude 3.5 Sonnet) handle tasks requiring broad contextual understanding - summarization and query classification, but specialist agents (Joint-CGA) handle domain-specific tasks - literature retrieval, multi-step clinical inference on bounded inputs to produce a differential diagnosis with transparent inference traces.

This separation makes the system computationally efficient, auditable, and trustworthy in ways a single opaque model call cannot achieve.

3.5.2 Human-Agent Collaboration in Clinical Settings. The modularity enables three distinct properties that support responsible deployment.

- **Interpretability for Clinical Teams:** Bounded context windows per agent prevent information overload in complex cases. By constraining each generation step to specific inputs (patient history, top-3 documents, query intent), the system produces inference traces that clinicians can inspect and validate, enabling meaningful critique and verification at the point of care.
- **Institutional Governance:** The separation of specialized AI agents aligns with how regulators evaluate clinical AI systems. Regulatory frameworks distinguish between evidence quality and diagnostic inference quality. Modular systems would enable institutions to assess each component against its corresponding regulatory standard, creating clear accountability layers.
- **Operational Sustainability & Adaptability:** As clinical guidelines change, individual agents can be added, updated or retrained in isolation. This creates a flexible maintenance surface rather than an all-or-nothing choice, where institutions get locked into outdated versions or forced to conduct expensive full retraining.

3.5.3 Multi-Agent Framework: Clinical Decision Assistant. The framework comprises specialized agents that collaborate to process user inputs, retrieve supporting materials, and produce medically coherent diagnostic responses. Figure 1 provides an overview of the interaction flow. When the user submits a query, the system initiates a multi-stage clinical decision pipeline managed by an agent orchestrator:

(1) **Initial Assessment Agent (Claude 3.5 Sonnet):** Classifies the incoming prompt and determines whether it constitutes a medical query. Non-medical inputs are filtered out to ensure controlled workflow execution.
(2) **Patient History Agent (Claude 3.5 Sonnet):** For confirmed medical queries, the system requests the patient's name. The patient history agent retrieves relevant entries from a structured clinical records dataset and generates a summary of prior visits, diagnoses, medications, and other context.
(3) **Literature Retrieval Agent (Joint-CGA model):** This agent searches across a curated corpus of 1,000 medical literature documents to surface the top-3 sources that are most relevant to the case. It relies on semantic similarity and evidence-based retrieval.
(4) **Inference Agent (Joint-CGA model):** This agent generates multi-step clinical assessments by conditioning token selection on patient history, retrieved evidence, and query intent. It produces a differential diagnosis by sequentially generating tokens from the model head, and outputs inference

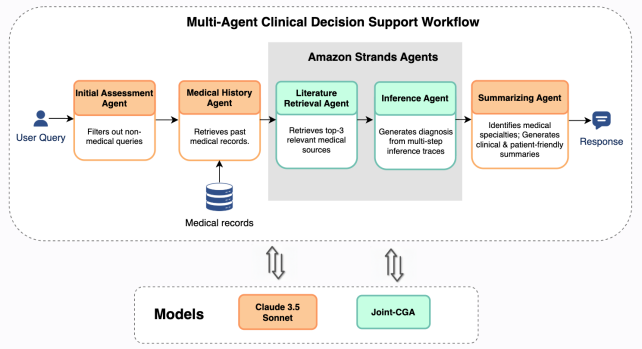

Figure 6: Agent workflow

traces showing which evidence informed each diagnostic step.

(5) **Summary and Recommendation Agent (Claude 3.5 Sonnet):** Uses the generated diagnosis to identify relevant medical specialties to consult and generates two outputs: a technical summary for clinicians and a patient-friendly explanation for lay users.

3.5.4 Deployment Pipeline.

- A HuggingFace container with a compatible deep learning image URI is used for execution within the AWS environment.
- The model is deployed as a SageMaker Endpoint to support low-latency inference.
- The endpoint is integrated into the **Amazon Strands**[1] framework using its custom model provider, enabling seamless invocation by multiple agents.

3.5.5 Integration Insights. This agentic design demonstrates that unifying embedding-based retrieval and generation enables structured, evidence-grounded clinical support. The modular agent setup promotes explainability and robustness by allowing domain-specific components (retrieval, generation, summarization) to operate concurrently while maintaining semantic alignment through a shared model representation. Having a unified architecture in the agentic pipeline achieve:

- Consistent information exchange across agents, reducing contradictions that typically arise in modular pipelines.
- Improved evidence grounding, as both retrieval and inference operate on aligned embeddings.
- Robustness and modularity, enabling individual agents to evolve independently while maintaining coherence through a common model backbone.
- Efficiency in low-resource settings, as the unified architecture reduces computational overhead by eliminating the need to maintain separate models. This shared representation improves data efficiency, minimizes cross-model drift, and enables reliable deployment even in environments with limited hardware or constrained clinical datasets.

The unified architecture significantly enhances multi-agent coordination, reduces integration complexity, and enables more reliable, clinically grounded decision support, valuable for agentic systems operating in both high-stakes and low-resource clinical environments.

3.6 Social Impact: Healthcare Accessibility through architectural efficiency

Clinical AI adoption faces barriers beyond model performance. Organizations struggle with the operational complexity of deploying, maintaining, and coordinating multiple specialized models each requiring distinct infrastructure, optimization pipelines, and integration work[18]. Documented surveys find that 34% of organizations cite complexity and integration as major obstacles to AI adoption. The challenge is most acute in smaller institutions; 39% of organizations with revenue below $250M report lack of qualified technical staff as a significant constraint.

3.6.1 Where Unified Architecture Reduces Deployment Barriers. By requiring only a single model instead of coordinating separate retriever and generator components, the Joint-CGA architecture simplifies three concrete aspects of deployment:

- **Initial Setup:** A single API endpoint replaces the need to coordinate and integrate separate services, reducing deployment complexity and timeline.
- **Maintenance:** Updating the system requires fine-tuning a single model, eliminating cross-component compatibility testing and version coordination.
- **Monitoring:** Single endpoint metrics and rollback procedures reduce the surface area for integration failures.

These are operational improvements grounded in the practical challenge of managing multiple models; they do not depend on claims beyond what the architecture directly provides.

3.6.2 Potential for Broader Access. Making AI-powered clinical triage more accessible to resource-constrained settings can help reduce diagnostic delays and improve care quality across diverse communities. The system's ability to retrieve relevant medical literature and generate evidence-based recommendations provides clinicians in resource-limited settings with decision support tools previously available only to well-resourced institutions. By reducing computational overhead and operational complexity, the architecture enables deployment in contexts where separate retriever-generator systems would be impractical:

- Community health centers with limited IT infrastructure
- Healthcare systems in regions where computational resources are expensive
- Smaller institutions with limited technical staff for model coordination

Deployment in these contexts is viable with this architecture but requires additional work beyond what the architecture provides: clinical validation in local populations, integration with existing EHR systems, clinician training, and ongoing performance monitoring.

3.6.3 Realistic Scope of Impact. The unified architecture contributes to broader healthcare AI deployment by reducing one specific barrier: computational and operational complexity. This is meaningful

for resource-constrained settings but is not sufficient for healthcare access without addressing remaining barriers:

- **Regulatory validation:** Clinical validation and approval in target populations and settings
- **System integration:** Connecting to existing clinical workflows and EHR infrastructure
- **Operational governance:** Clinician training, institutional policies for system use and oversight
- **Performance assurance:** Ongoing monitoring and failure analysis to ensure reliability in deployment

Reducing deployment complexity is a necessary step toward broader adoption, not a sufficient one. The contribution of Joint-CGA's unified architecture removes one category of technical barriers in a system where many barriers remain.

4 LIMITATIONS

Evaluation Methodology Constraints. Retrieval accuracy is measured using standard information retrieval metrics, while generation quality is assessed using Claude 3.5 Sonnet as an LLM-as-a-Judge. This evaluation approach presents reproducibility challenges, as frontier models evolve over time and updates alter scoring behavior. Judge models exhibit stylistic and structural biases, favoring certain answer patterns that resemble their training distribution. The judge LLM evaluates outputs for textual plausibility without access to factual ground truth, constraining clinical accuracy assessment.

Dataset and Training Constraints. The training split of 1,000 samples limits exposure to diverse medical scenarios, potentially affecting generalization across clinical contexts and specialties. The 500-token response constraint creates completeness trade-offs, constraining comprehensive clinical text generation for complex scenarios. The size of the evaluation set restricts statistical power for drawing broader conclusions, as the results present point estimates without confidence intervals.

5 DISCUSSION

Baseline Selection Rationale. The choice of Llama-3.1-8B as the embedding baseline (Figures 4 and 5) reflects the core architectural philosophy of unified model management. While Llama-3.1 is not specifically trained for retrieval tasks, specialized retrieval models lack generative capabilities and cannot serve as meaningful baselines for evaluating a unified system. By comparing against Llama-3.1-8B, we establish a fair baseline that shares the same generative foundation as Joint-CGA. This comparison directly demonstrates the value added by our contrastive fine-tuning approach: the 7% improvement in retrieval accuracy and 5.9% gain in Top-1 retrieval represent the enhancement achieved through distribution-aware contrastive objectives while preserving full generative capabilities. A comparison against retrieval-only models would conflate architectural differences with training methodology, obscuring the specific contribution of our unified approach.

Hybrid Training Methodology. Joint-CGA employs a hybrid training approach that combines two complementary learning paradigms. Embedding fine-tuning utilizes contrastive-based methods that learn representations by distinguishing between positive and negative examples in the embedding space. Generative fine-tuning employs reconstruction-based approaches through supervised fine-tuning, where the model learns to reconstruct target sequences given input contexts. Alternative paradigms exist —purely contrastive methods could be applied to both tasks, or reconstruction-based techniques could be adapted for embedding learning. Each paradigm presents distinct trade-offs: contrastive approaches excel at learning discriminative representations with limited labeled data but require careful negative sampling strategies, while reconstruction-based methods leverage abundant unlabeled text but may not optimize directly for retrieval precision. The hybrid approach balances these considerations, using contrastive learning where discrimination matters most (embeddings) and reconstruction where sequence modeling is paramount (generation). But systematic exploration of alternative combinations would be valuable future work. Domain-specific characteristics influence the relative effectiveness of different contrastive objectives and optimal loss selection varies across domains based on factors such as query complexity, document heterogeneity, and the semantic structure of the knowledge base, this suggests that domain-adaptive loss selection strategies could further enhance the framework's generalizability.

6 FUTURE WORK

Training Methodology. Exploring alternative training paradigms—purely contrastive, reconstruction-based, or hybrid—across diverse domains would help establish guidelines for selecting objectives based on task characteristics and data availability.

Clinical Validation and Explainability. Human-in-the-loop validation with clinicians would enable iterative refinement and accuracy verification. Interpretability enhancements such as attention visualization and counterfactual explanations can improve clinical trust and support regulatory compliance.

Multimodal Integration. Extending the architecture to support vision-language tasks (e.g., radiograph or CT interpretation) can enable diagnostic generation that combines textual patient data with medical imaging within a single framework.

7 CONCLUSION

This work presents Joint-CGA, a unified embedding-generation architecture that consolidates retrieval and generation within a single model, delivering measurable gains in both retrieval accuracy and domain-specific text generation quality. By eliminating the need for separate retriever–generator pipelines, the architecture reduces deployment complexity, streamlines maintenance, and lowers infrastructure overhead. The multi-agent integration further demonstrates that unified models can preserve interpretability through clear separation of agent responsibilities. The design enables resource efficiency, auditability, and modular updates as clinical requirements evolve—addressing key operational and governance barriers that healthcare organizations frequently cite in AI adoption. Future research should prioritize prospective clinical validation, seamless integration with existing clinical workflows, and real-time human-in-the-loop feedback mechanisms to rigorously evaluate real-world adoption, and fairness across diverse patient populations. Overall, this work lays the foundation for scalable, interpretable, and accessible multi-agent AI systems in healthcare.

REFERENCES

[1] 2025. Strands Agents SDK: A technical deep dive into agent architectures and observability | Artificial Intelligence. https://aws.amazon.com/blogs/machine-learning/strands-agents-sdk-a-technical-deep-dive-into-agent-architectures-and-observability/ Section: Amazon Bedrock.

[2] Parishad BehnamGhader, Santiago Miret, and Siva Reddy. 2023. Can Retriever-Augmented Language Models Reason? The Blame Game Between the Retriever and the Language Model. https://doi.org/10.48550/arXiv.2212.09146 arXiv:2212.09146 [cs].

[3] Yihao Chen, Xianbiao Qi, Jianan Wang, and Lei Zhang. 2023. DisCo-CLIP: A Distributed Contrastive Loss for Memory Efficient CLIP Training. https://doi.org/10.48550/arXiv.2304.08480 arXiv:2304.08480 [cs].

[4] World Economic Forum. [n.d.]. WEF Patient First Health with Generative AI.

[5] Jiawei Gu, Xuhui Jiang, Zhichao Shi, Hexiang Tan, Xuehao Zhai, Chengjin Xu, Wei Li, Yinghan Shen, Shengjie Ma, Honghao Liu, Saizhuo Wang, Kun Zhang, Yuanzhuo Wang, Wen Gao, Lionel Ni, and Jian Guo. 2024. A Survey on LLM-as-a-Judge. https://doi.org/10.48550/ARXIV.2411.15594 Version Number: 6.

[6] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. LoRA: Low-Rank Adaptation of Large Language Models. https://doi.org/10.48550/arXiv.2106.09685 arXiv:2106.09685 [cs].

[7] Gautier Izacard, Mathilde Caron, Lucas Hosseini, Sebastian Riedel, Piotr Bojanowski, Armand Joulin, and Edouard Grave. 2022. Unsupervised Dense Information Retrieval with Contrastive Learning. https://doi.org/10.48550/arXiv.2112.09118 arXiv:2112.09118 [cs].

[8] Ziyan Jiang, Xueguang Ma, and Wenhu Chen. 2024. LongRAG: Enhancing Retrieval-Augmented Generation with Long-context LLMs. https://doi.org/10.48550/arXiv.2406.15319 arXiv:2406.15319 [cs].

[9] Prannay Khosla, Piotr Teterwak, Chen Wang, Aaron Sarna, Yonglong Tian, Phillip Isola, Aaron Maschinot, Ce Liu, and Dilip Krishnan. 2021. Supervised Contrastive Learning. https://doi.org/10.48550/arXiv.2004.11362 arXiv:2004.11362 [cs].

[10] Naveen Krishnan. 2025. Advancing Multi-Agent Systems Through Model Context Protocol: Architecture, Implementation, and Applications. https://doi.org/10.48550/arXiv.2504.21030 arXiv:2504.21030 [cs].

[11] Xiaoxi Li, Guanting Dong, Jiajie Jin, Yuyao Zhang, Yujia Zhou, Yutao Zhu, Peitian Zhang, and Zhicheng Dou. 2025. Search-o1: Agentic Search-Enhanced Large Reasoning Models. https://doi.org/10.48550/arXiv.2501.05366 arXiv:2501.05366 [cs].

[12] Hieu Man, Nghia Trung Ngo, Franck Dernoncourt, and Thien Huu Nguyen. 2024. ULLME: A Unified Framework for Large Language Model Embeddings with Generation-Augmented Learning. https://doi.org/10.48550/arXiv.2408.03402 arXiv:2408.03402 [cs].

[13] Niklas Muennighoff, Hongjin Su, Liang Wang, Nan Yang, Furu Wei, Tao Yu, Amanpreet Singh, and Douwe Kiela. 2025. Generative Representational Instruction Tuning. https://doi.org/10.48550/arXiv.2402.09906 arXiv:2402.09906 [cs].

[14] Thang Nguyen, Peter Chin, and Yu-Wing Tai. 2025. MA-RAG: Multi-Agent Retrieval-Augmented Generation via Collaborative Chain-of-Thought Reasoning. https://doi.org/10.48550/arXiv.2505.20096 arXiv:2505.20096 [cs].

[15] Agada Joseph Oche, Ademola Glory Folashade, Tirthankar Ghosal, and Arpan Biswas. 2025. A Systematic Review of Key Retrieval-Augmented Generation (RAG) Systems: Progress, Gaps, and Future Directions. https://doi.org/10.48550/arXiv.2507.18910 arXiv:2507.18910 [cs].

[16] Nils Reimers and Iryna Gurevych. 2019. Sentence-BERT: Sentence Embeddings using Siamese BERT-Networks. https://doi.org/10.48550/arXiv.1908.10084 arXiv:1908.10084 [cs].

[17] Raanan Y. Rohekar, Yaniv Gurwicz, Sungduk Yu, Estelle Aflalo, and Vasudev Lal. 2025. A Causal World Model Underlying Next Token Prediction: Exploring GPT in a Controlled Environment. https://doi.org/10.48550/arXiv.2412.07446 arXiv:2412.07446 [cs] version: 4.

[18] Jacob T. Rosenthal, Ashley Beecy, and Mert R. Sabuncu. 2025. Rethinking clinical trials for medical AI with dynamic deployments of adaptive systems. *npj Digital Medicine* 8, 1 (May 2025), 252. https://doi.org/10.1038/s41746-025-01674-3

[19] Samuel Schmidgall, Carl Harris, Ime Essien, Daniel Olshvang, Tawsifur Rahman, Ji Woong Kim, Rojin Ziaei, Jason Eshraghian, Peter Abadir, and Rama Chellappa. 2024. Addressing cognitive bias in medical language models. https://doi.org/10.48550/arXiv.2402.08113 arXiv:2402.08113 [cs].

[20] Rulin Shao, Rui Qiao, Varsha Kishore, Niklas Muennighoff, Xi Victoria Lin, Daniela Rus, Bryan Kian Hsiang Low, Sewon Min, Wen-tau Yih, Pang Wei Koh, and Luke Zettlemoyer. 2025. ReasonIR: Training Retrievers for Reasoning Tasks. https://doi.org/10.48550/arXiv.2504.20595 arXiv:2504.20595 [cs].

[21] Duojin Wang, Jiawan Liu, Qinglian Lin, and Hongliu Yu. 2024. A decision-making system based on case-based reasoning for predicting stroke rehabilitation demands in heterogeneous information environment. *Applied Soft Computing* 154 (March 2024), 111358. https://doi.org/10.1016/j.asoc.2024.111358

[22] Wenxuan Wang, Zizhan Ma, Zheng Wang, Chenghan Wu, Jiaming Ji, Wenting Chen, Xiang Li, and Yixuan Yuan. 2025. A Survey of LLM-based Agents in Medicine: How far are we from Baymax? https://doi.org/10.48550/arXiv.2502.11211 arXiv:2502.11211 [cs].

[23] Juncheng Wu, Wenlong Deng, Xingxuan Li, Sheng Liu, Taomian Mi, Yifan Peng, Ziyang Xu, Yi Liu, Hyunjin Cho, Chang-In Choi, Yihan Cao, Hui Ren, Xiang Li, Xiaoxiao Li, and Yuyin Zhou. 2025. MedReason: Eliciting Factual Medical Reasoning Steps in LLMs via Knowledge Graphs. https://doi.org/10.48550/arXiv.2504.00993 arXiv:2504.00993 [cs].

[24] Hong Qing Yu and Frank McQuade. 2025. RAG-KG-IL: A Multi-Agent Hybrid Framework for Reducing Hallucinations and Enhancing LLM Reasoning through RAG and Incremental Knowledge Graph Learning Integration. https://doi.org/10.48550/arXiv.2503.13514 arXiv:2503.13514 [cs].

[25] Yue Zhang, Yafu Li, Leyang Cui, Deng Cai, Lemao Liu, Tingchen Fu, Xinting Huang, Enbo Zhao, Yu Zhang, Chen Xu, Yulong Chen, Longyue Wang, Anh Tuan Luu, Wei Bi, Freda Shi, and Shuming Shi. 2025. Siren's Song in the AI Ocean: A Survey on Hallucination in Large Language Models. https://doi.org/10.48550/arXiv.2309.01219 arXiv:2309.01219 [cs].

[26] Wayne Xin Zhao, Jing Liu, Ruiyang Ren, and Ji-Rong Wen. 2022. Dense Text Retrieval based on Pretrained Language Models: A Survey. https://doi.org/10.48550/arXiv.2211.14876 arXiv:2211.14876 [cs].