SIIM CAIMI26 Conference on Artificial Intelligence in Medical Imaging
OCT 26-27 | University of Pennsylvania, Philadelphia, PA

Multi-Agent Reasoning Enhances Local Small-Model Performance in Medicine

Saisha Pradeep Shetty, MS, University of California, Davis; Austin Lin; Abhimaneu Pandey; Satvik Tripathi; Ebubechukwu D. Enwerem; Jacinta Arnod; Tessa Cook, MD PhD, CIIP, FSIIM

Introduction/Background
Clinical large language model pipelines still largely rely on single prompting strategies that combine information extraction, reasoning, and answer formatting within one prompt, limiting interpretability and adaptability. We developed MARC, a configurable, locally deployable multi-agent framework that integrates automated task decomposition with structured orchestration. In this study, we evaluate whether MARC generalizes across diverse medical knowledge domains and outperforms single-agent prompting.

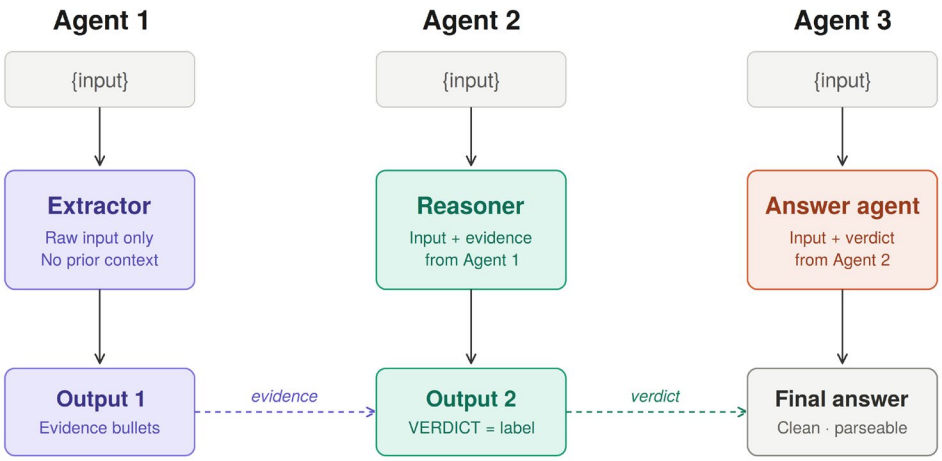
Methods/Intervention
MARC (Multi-Agent Reasoning and Coordination) is a model-agnostic Python pipeline that executes role-specialized agents for extraction, reasoning, and answer generation in a deterministic sequence. It uses explicit context passing and logs intermediate outputs, enabling stage-wise failure attribution. An integrated Decomposer converts a single plain-language task description into three subtasks and automatically generates role-specific agent prompts, replacing manual prompt engineering. We compared single-agent prompting against MARC on 1,089 questions from six MMLU medical subsets: medical genetics, professional medicine, clinical knowledge, anatomy, college medicine, and college biology. Both configurations used identical weights, quantization, prompts, decoding settings (zero-shot, temperature=0), and identical post-hoc answer-extraction logic, isolating orchestration as the only variable. All MARC agents used MedGemma 1.5 4B; the Decomposer used Gemma 4 E4B. Both models were served locally via Ollama at Q4_K_M quantization on CPU, without external API dependencies.

Results/Outcome
MARC outperformed single-agent prompting across all 1,089 questions (69.7% [759/1,089] vs. 58.8% [640/1,089]; +10.9 percentage points; macro-average 69.0% vs. 59.0%). Accuracy improved on five of six subsets: professional medicine (54.4%→75.0%; +20.6), college biology (56.3%→72.9%; +16.7), medical genetics (57.0%→70.0%; +13.0), clinical knowledge (63.4%→70.9%; +7.5), and college medicine (53.8%→60.7%; +6.9). Anatomy was the sole decrease (68.9%→64.4%; −4.4).

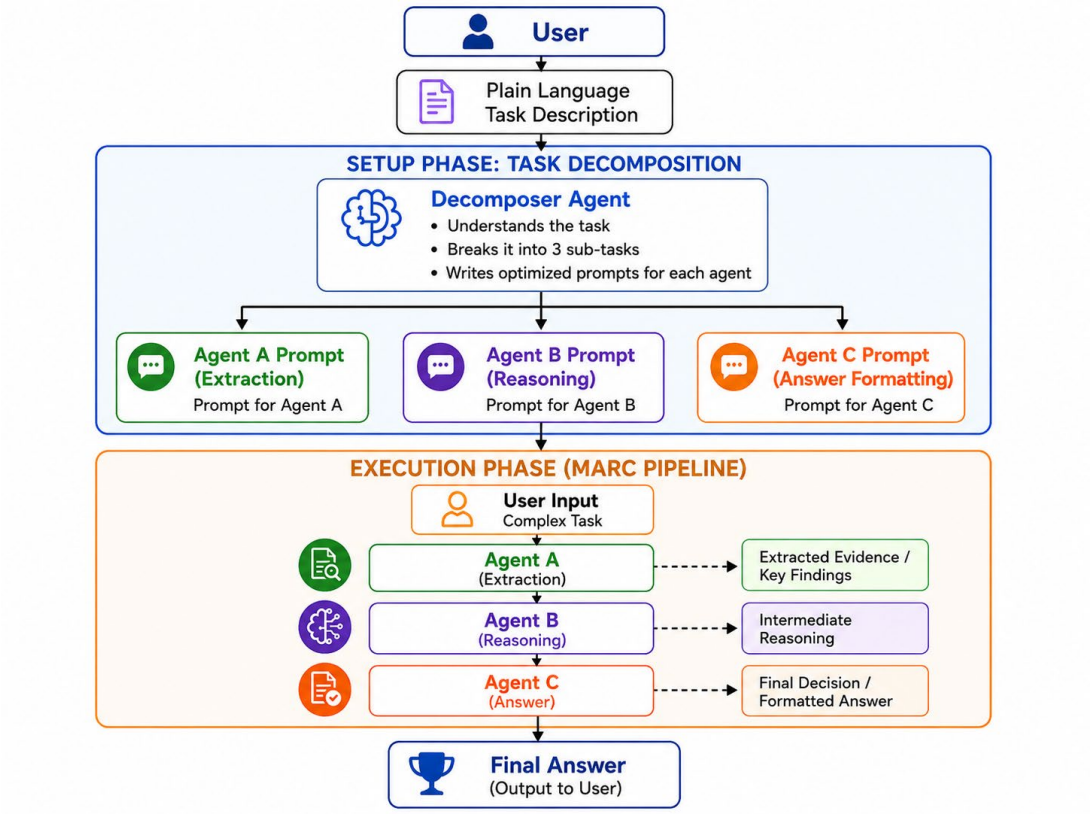
Conclusion
In locally deployed, 4-bit quantized small-model settings, structured multi-agent orchestration substantially improves accuracy over single-agent prompting, indicating that separating extraction, reasoning, and answer generation into distinct roles recovers capability that monolithic prompting leaves unused at small model scale. MARC runs on CPU without external APIs, critical where clinical data cannot leave the institution. The integrated Decomposer generates task-specific agent prompts from a plain-language description, removing manual prompt engineering.

Statement of Impact

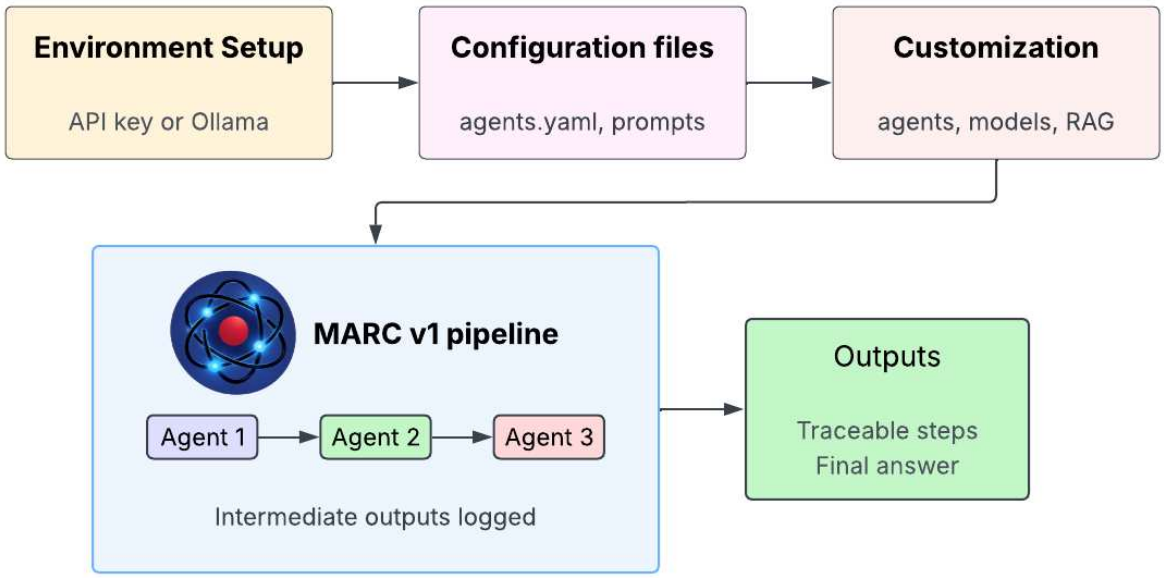
Multi-agent orchestration lets a small, open model run on local CPU hardware at accuracy well above single-agent use, enabling transparent, auditable clinical AI without sending patient data to external APIs.



MARC agent collaboration and context passing. The extractor receives only the raw input and produces evidence bullets; the reasoner receives the input plus that evidence and emits a standardized verdict; the answer agent returns the verdict verbatim as a clean, parseable prediction. Intermediate outputs are logged at each stage, enabling attribution of an error to a specific agent.



Automated task decomposition. In the setup phase, a single plain-language task description is decomposed into three subtasks with automatically generated role-specific agent prompts; in the execution phase, the pipeline runs with those prompts, passing outputs sequentially from extraction through reasoning to final answer formatting.



Local deployment and configuration workflow. Agent roles, model assignments, prompts, and optional context files are defined declaratively in configuration files, allowing pipelines to be modified without code changes and executed on local CPU-only infrastructure.

Keywords

Multi-agent systems**;** Large language models; Medical question answering; Clinical decision support; Local deployment