Contents lists available at ScienceDirect

Machine Learning with Applications

journal homepage: www.elsevier.com/locate/mlwa

# Prompt design for medical question answering with Large Language Models

Leonid Kuligin [a,b,*], Jacqueline Lammert [b,c], Aleksandr Ostapenko [d], Keno Bressem [e,f], Martin Boeker [b], Maximilian Tschochohei [a,b]

[a] Google Cloud, Munich, Germany
[b] Chair of Medical Informatics, Institute of Artificial Intelligence and Informatics in Medicine (AIIM), TUM University Hospital, Technical University of Munich, Munich, Germany
[c] Department of Gynecology and Center for Hereditary Breast and Ovarian Cancer, Technical University of Munich (TUM), School of Medicine and Health, Klinikum rechts der Isar, TUM University Hospital, Munich, Germany
[d] Google Cloud, Krakow, Poland
[e] Department of Cardiovascular Radiology and Nuclear Medicine, Technical University of Munich, School of Medicine and Health, German Heart Center, TUM University Hospital, Munich, Germany
[f] Department of Diagnostic and Interventional Radiology, Technical University of Munich, School of Medicine and Health, Klinikum rechts der Isar, TUM University Hospital, Munich, Germany

ARTICLE INFO

*Keywords:*
LLM
Generative AI
Question-answering benchmarks
Agentic workflows
ReACT

## ABSTRACT

The combination of prompting technique and the choice of a foundational model determines end-to-end workflow performance on a given task. We aim to provide comprehensive guidance for the best-performing prompting techniques for various LLMs for medical question-answering. We aim to provide comprehensive guidance for the best-performing prompting techniques for a variety of LLM for medical question-answering. We evaluated 15 large LLMs (incl. Claude 3.5 Sonnet, Gemini pro, Llama, Mistral, OpenAI GPT-4o and 4.1) and 6 smaller models (incl. Gemma, Mistral Nemo, Llama 3.1, Gemini flash) across five prompting techniques on neuro-oncology exam questions. Using the established MedQA dataset and a novel neuro-oncology question set, we compared basic prompting, chain-of-thought reasoning, and more complex agent-based methods incorporating external search capabilities. Results showed that the Reasoning and Acting (ReAct) approach combined with giving LLM access to Google Search performed best on large models like Claude 3.5 Sonnet (81.7% accuracy and 85.5% for v2). We also showed that large models significantly outperformed smaller ones on the MedQA dataset (79.3% vs. 51.2% accuracy) and that complex agentic patterns like Language Agent Tree Search provided minimal benefits despite 5x higher latency. We recommend practitioners to experiment with various techniques given their specific use case and foundational model, and favor simple prompting patterns with large models, as they offer the best balance of accuracy and efficiency.

## 1. Introduction

Large Language Models (LLMs) are increasingly utilized in natural language processing tasks. They have demonstrated in-context learning capabilities, performing strongly in tasks that they were not explicitly trained on, and often matching or exceeding the capabilities of state-of-the-art machine learning models (Brown et al., 2020; Lewis et al., 2020). This versatility is mainly due to new architectures such as transformers or mixture-of-experts and hardware innovations that allow scaling on two dimensions: the model size (amount of trainable weights) and the size of datasets used for training. One of the emergent capabilities is the high performance of LLMs in professional and certification exams in fields such as medicine and law, particularly for multiple-choice formats (Brown et al., 2020).

Despite this performance, extracting knowledge from LLMs presents distinct challenges, and improvements can be broadly categorized as model-level (e.g., scaling size, instruction-tuning, training a better model) and technique-level (e.g., prompt engineering). The most straightforward approach relies on the model's internal parametric memory. However, this method is prone to errors, such as hallucinations, and updating the model's knowledge (to incorporate domain knowledge or fresh facts) requires extensive retraining. To address these limitations, particularly in domains such as medicine that require

* Corresponding author at: Chair of Medical Informatics, Institute of Artificial Intelligence and Informatics in Medicine (AIIM), TUM University Hospital, Technical University of Munich, Munich, Germany.

*E-mail addresses:* leonid.kuligin@tum.de (L. Kuligin), Jacqueline.Lammert@mri.tum.de (J. Lammert), aostapenko@google.com (A. Ostapenko), keno.bressem@tum.de (K. Bressem), martin.boeker@tum.de (M. Boeker), maximilian.tschochohei@tum.de (M. Tschochohei).

https://doi.org/10.1016/j.mlwa.2025.100758
Received 26 July 2025; Received in revised form 2 October 2025; Accepted 12 October 2025
Available online 16 October 2025

up-to-date or highly specialized knowledge, alternative approaches have been developed. These include Retrieval-Augmented Generation (RAG), which integrates information retrieval from external sources (Brown et al., 2020; Lammert et al., 2024; Lewis et al., 2020), as well as various prompt engineering strategies.

Prompt design explores general prompts that improve LLM performance on various tasks (Li, Al Kader Hammoud, Itani, Khizbullin, & Ghanem, 2023; Wei et al., 2022). One of the first demonstrations of LLM performance improvement through prompt design was Chain-of-Thought (CoT) prompting. With CoT prompting, instead of generating an immediate answer, an LLM is asked to include a series of intermediate reasoning steps in the output, replicating the reasoning process of a human (Nair, Schumacher, Tso, & Kannan, 2024). Another approach is self-consistency, where the model generates the same output multiple times with increased variability, and the final answer is determined through mechanisms such as majority voting or weighted selection (Nair et al., 2024; Wang, Wei et al., 2023).

More advanced multi-step reasoning frameworks have also emerged. The ReAct (Reason-Act) framework, for instance, enhances the ability of LLMs to solve complex tasks by synergizing internal reasoning steps with interactions with external environments or tools (Yao et al. 2022). Further enhancements include task decomposition, (Liu et al., 2025; Schlag et al., 2023; Wang, Xu, et al., 2023; Yang et al., 2025), fostering interaction between LLMs through natural language dialogue (Liu et al., 2025; Wu et al., 2023) or building and exploring a tree of alternative solutions, as seen in Language Agent Tree Search (LATS), which unifies reasoning, acting, and planning within a single framework (Long, 2023; Zhou, Yan, Shlapentokh-Rothman, Wang, & Wang, 2024).

In the medical domain, combining instruction prompt-tuning with larger model sizes has shown promise for improving comprehension, knowledge recall, and reasoning on medical question-answering (Q&A) tasks (Singhal et al. 2023). Performance is often evaluated using datasets like MedQA, a large-scale open-domain Q&A dataset derived from the United States Medical Licensing Exam (USMLE) (Jin et al., 2021). However, such multiple-choice benchmarks may not capture the full complexity of medical reasoning, and evaluating open-ended tasks remains a challenge (Hosseini et al., 2024; Singhal et al., 2025).

While various prompting techniques (CoT, ReAct, LATS) and model types exist, a comprehensive evaluation of their comparative effectiveness for specialized medical Q&A is lacking. It remains unclear how these different strategies — ranging from simple reasoning steps to complex tool-augmented search — perform relative to one another, especially when applied to both general medical knowledge and niche sub-specialties.

This study aims to address this gap. Our primary objective is to conduct a comprehensive evaluation of the influence of different prompt engineering strategies on LLM performance in medical contexts and the choice of foundational models. We explicitly assess established techniques like CoT alongside emerging agentic approaches such as LATS that leverage external tools and knowledge retrieval. Our analysis encompasses ten top-performant foundational models and six smaller alternatives, evaluated across the MedQA dataset and a novel neuro-oncology question set developed to probe specialized medical reasoning. We define a "small" LLM as a distilled version of a larger model that is optimized for latency and cost. We analyze these models as a separate category given their different performance characteristics. This study was performed in two phases: one in September–October 2024 and another in April 2025. In April 2025, we added newly released foundational models, including models from OpenAI and DeepSeek.

## 2. Methods

This section details the methodology we employed to evaluate the performance of LLMs on complex medical Q&A tasks.

The core of our methodology involved evaluating various prompting strategies and agentic workflows. We explored four prompting techniques: vanilla Q&A, self-consistency, CoT, and reflection chains. Furthermore, we investigated five distinct agentic workflows that utilized Google Search to augment LLM reasoning abilities. These workflows included ReACT, Plan-and-solve, Multi-agent collaboration with reflection, Dialogue-based agents, and LATS.

### 2.1. Benchmark

To effectively evaluate LLMs, we first established our benchmark datasets. Evaluating LLMs for medical Q&A requires assessing the model's ability to understand medical terminology, reason through complex clinical scenarios, and provide accurate and reliable answers while adhering to ethical guidelines and patient safety considerations (Long, 2023; Xiao et al., 2025; Zhou et al., 2024). Consequently, our evaluation involved diverse benchmark datasets covering a wide range of medical specialties and question types, including factual and reasoning-based questions (Long, 2023; Moreno & Bitterman, 2024; Xiao et al., 2025; Zhou et al., 2024).

#### 2.1.1. Baseline dataset

First, we selected a "baseline" dataset, which is commonly used to evaluate artificial intelligence applications. Such a baseline dataset was required to ensure that an LLM system performs within parameters and will not lead to patient harm (Arora et al., 2023; Long, 2023; Moreno & Bitterman, 2024; Xiao et al., 2025; Zhou et al., 2024). MedQA was selected as the baseline dataset, as it has become an industry standard for testing LLM systems for medical Q&A (Jin et al., 2021). It contains 10,178 free-form English questions with multiple-choice answers based on professional medical board exams. The drawback of using MedQA, which represents a potential limitation to our study, is that it was published 5 years ago and has thus likely been incorporated into LLM training data. Consequently, models may retrieve correct answers from their parametric memory, not due to reasoning (Moreno & Bitterman, 2024).

#### 2.1.2. Oncology Q&A dataset

Creating a custom evaluation dataset was beneficial as it allowed for tailoring the evaluation to specific medical Q&A use cases and addressing the limitations of existing datasets, such as biases, limited scope, and lack of real-world patient data (Bedi et al., 2024). Therefore, we used multiple-choice questions from the book "Neuro-Oncology Explained Through Multiple Choice Questions" (Das, 2023; Moreno & Bitterman, 2024). This is a new proprietary textbook published in 2023; we assume it was not incorporated into LLMs' training datasets. The book specializes in neuro-oncology and has more than 500 multiple-choice questions on topics such as classification of central nervous system tumors, molecular genetics, and the latest clinical practice therapeutic techniques, such as stem-cell and immunotherapy. We created a custom benchmark with 421 questions collected under fair use. Questions are oriented for a specialized medical examination, hence they are comparable with the MedQA benchmark. They cover 24 neuro-oncology topics from brain tumors and angiogenesis to palliative care. 85% of the questions have four options to choose from, and others have three or five options. Questions are text-only with length varying from 23 to 1049 symbols (with an average of 156 and a median of 98). Answers to choose from are either a full text or a short answer (for example, the correct gene or mutation name, or a chromosome number where the mutation in question is located).

The publisher has not granted permission to share this proprietary dataset. However, to ensure our methodology is verifiable, we are releasing our benchmarking framework and the full code, and it can be reproduced on any private or public datasets. This framework, available in our repository, includes all necessary version control information and parameter settings, allowing our study to be reproduced on any private or public dataset. Please refer to our GitHub repository for the complete code and additional instructions to replicate each prompting strategy.

### 2.2. Models evaluated

Following dataset selection, we chose models for evaluation. To comprehensively assess the capabilities of LLMs in medical Q&A, we selected a diverse set of models, ranging from large proprietary foundational models to smaller, more accessible open-source models. By including proprietary and open-source models, we aimed to assess each approach's relative strengths and weaknesses.

We evaluated fifteen large foundational models: Gemini 1.5-pro-001 and 002 (Google, 2025), Gemini-2.0-flash (Pichai, 2024), Gemini-2.5-flash and Gemini-2.5-pro (Kavukcuoglu, 2024), gpt4o and gpt 4.1 (OpenAI, 2025), Claude 3.5 Sonnet and Claude 3.5 Sonnet v2 (Anthropic, 2024), DeepSeek v3 (Liu et al., 2024), Mistral large@2407 and @2411 (Mistral, 2024a), and Llama models 3.1 405B, 3.1 70B, 3.2 90B and 3.3 70B (Meta A.I., 2024a). We also included six small foundational models (four of them were open-sourced) – Gemini 1.5-flash-001 (Google, 2025), Mistral Nemo@2407 (Mistral, 2024b), open-sourced Gemma 2 instruct 2B, 9B, 27B (Google, 2024), and Llama 8B (Meta A.I., 2024b). By including these models, we could explore the trade-offs between model size and performance, and the potential for smaller models to achieve reasonable performance on medical Q&A tasks. Some models, particularly the open-weighted ones, offered limited or zero support for tool calling (a structured way to allow LLMs to interact with an external environment by generating requests to APIs or databases). Hence, these models were not evaluated with agentic workflows that require tool-calling support.

### 2.3. Algorithms

This section explores the prompting techniques used to evaluate LLMs' reasoning capabilities. We investigated knowledge-based prompting, which relies solely on the LLM's internalized knowledge, and agentic workflows, which incorporate external tools like Google Search to augment reasoning.

#### 2.3.1. Prompt engineering

We started with knowledge-based prompting, which only relies on knowledge that an LLM has memorized during the training phase. Our rationale for choosing these particular techniques was their prevalence in the literature and their representation of a clear progression in reasoning complexity, from direct answers to multi-step deliberation and self-correction (Vatsal & Dubey, 2024):

*Q&A prompt with question rewriting.* Our first Q&A system used a two-step pipeline. First, we used a "classical" Q&A prompt by appending the question and potential answers to the provided context. Then, LLM generates the final output based on this prompt. Subsequently, a second call to an LLM is made, specifically for answer parsing. This step aims to mitigate the variability in the model output and ensure accurate extraction of the chosen answer, regardless of its phrasing. Please, refer to 1 for more details.

*Self-consistency.* The self-consistency prompting technique leverages repeated sampling to enhance the reliability of LLM outputs. By increasing the temperature parameter during generation, we obtain a diverse set of answers or reasoning paths. We identify the most frequent response through majority voting, assuming it to be the most probable and consistent solution. This approach aims to mitigate the stochasticity inherent in LLMs and improve the accuracy of their outputs (Wang, Wei et al., 2023).

*CoT.* CoT seeks to improve the reasoning abilities of LLMs by explicitly guiding them to generate intermediate reasoning steps. This is achieved by providing examples of questions with corresponding chains of thought leading to the final answer within the prompt. By incorporating these reasoning steps, the model is directed to break down complex problems into smaller, more manageable components, enhancing its ability to derive accurate solutions (Wei et al., 2022).

*Reflection.* Building upon the CoT prompting technique, reflection prompting introduces an additional layer of self-critique. After generating a CoT response, an LLM is prompted to evaluate its reasoning, identify potential flaws, and refine its initial answer. This reflective process encourages the model to engage in deeper reasoning and self-correction, potentially leading to more accurate and robust solutions (Yu, He, Wu, Dai, & Chen, 2023).

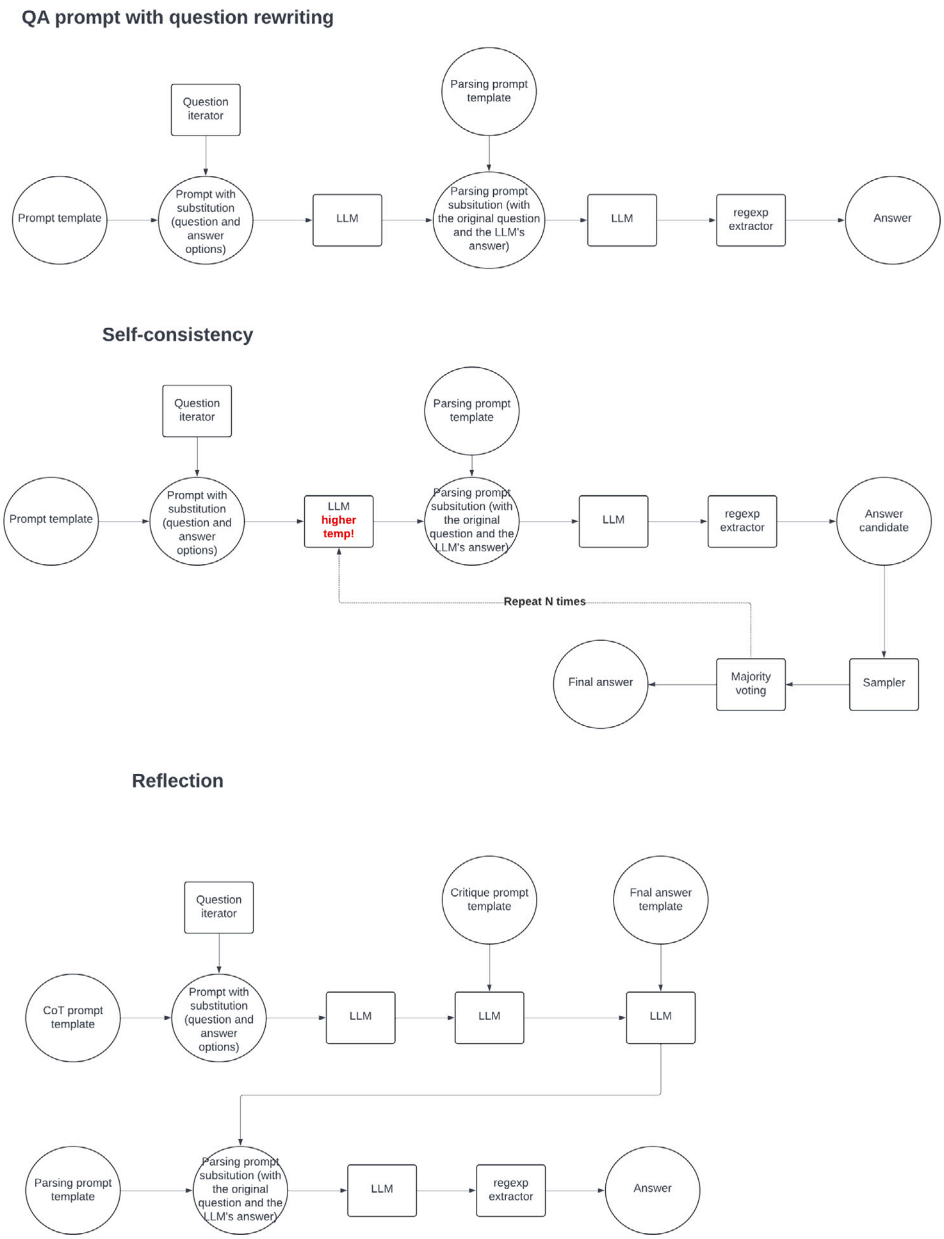
#### 2.3.2. Agentic workflows with search grounding

It has been demonstrated that LLM reasoning capabilities can be increased significantly with so-called agentic workflows. In an agentic workflow, the state is maintained by an external orchestrator, and the flow itself is partially controlled by an LLM ((Lewis et al., 2020; Schlag et al., 2023; Yu et al., 2023). Additionally, agentic patterns enable external data sources such as PubMed to enhance LLM systems with information that is not present in their training data. This has been shown to improve LLM task performance for medical Q&A use cases (Das, 2023; Lammert et al., 2024; Moreno & Bitterman, 2024). In this work, we focused on five architectural patterns. Our rationale for selecting these specific workflows was to evaluate a range of strategies for integrating external information, from simple tool use to complex multi-agent collaboration. All agentic workflows were executed on a standard cloud virtual machine (specific hardware/software configurations are detailed in our repository), with computational constraints chiefly relating to API rate limits and overall latency.

*ReAct agent.* ReAct prompts an LLM to "think" before it "acts". An LLM is prompted to generate a reasoning trace and an action plan. This can be as simple as prompting the model to list the steps to solve a problem or as complex as generating a detailed plan for interacting with a user. ReAct has been shown to improve the performance of LLMs on various tasks, including Q&A, code generation, and task-oriented dialogue (Yao et al., 2023). In our example, we asked an LLM to perform a Google Search, which requires the use of tool calling described in the introduction.

*Plan-and-solve agent.* Plan-and-solve prompting is a technique used to enhance the problem-solving abilities of LLMs. It involves explicitly prompting the model to generate a plan before attempting to solve a problem. This plan can be a simple list of steps or a more complex, structured plan. The model then executes the plan, step-by-step, refining it as needed based on the outcomes of each step. This approach encourages an LLM to break down complex tasks into smaller, more manageable sub-tasks, leading to improved performance and a more systematic approach to problem-solving (Wang, Xu, et al., 2023; Yao et al., 2023). Our system employed a four-phase iterative process, wherein an LLM leveraged Google Search as its tool to generate a solution. An LLM alternated between planning (generating search queries or final answers), searching (retrieving and summarizing relevant information), and parsing (refining the answer), being forced to respond to the user after n iterations, which terminated the process. In this work, we empirically decided on n=5. This hyperparameter was chosen as a balance between allowing sufficient reasoning steps and managing computational costs and latency; a higher n might yield marginal gains but at a significant cost.

*Multi-agent collaboration.* Our multi-agent collaboration framework employs two LLMs to collaborate to generate a solution (Renze & Guven, 2024). A ReAct agent, equipped with Google Search access, iteratively plans, searches, and parses information to formulate an initial answer. Concurrently, a "reflecting" LLM analyzes the initial question, the proposed answer, and the ReAct agent's reasoning trace to generate a critique or approval of the answer. A forced response mechanism (a special instruction that forces an LLM to call a tool) ensures termination after n=5 iterations, and a final parsing stage prepares the output for evaluation. This collaborative approach leverages the strengths of both agents, combining the ReAct agent's information-gathering capabilities with the reflecting LLM's critical analysis skills.


QA prompt with question rewriting

Question iterator
Parsing prompt template
Prompt template
Prompt with substitution (question and answer options)
LLM
Parsing prompt subsitution (with the original question and the LLM's answer)
LLM
regexp extractor
Answer

Self-consistency

Question iterator
Parsing prompt template
Prompt template
Prompt with substitution (question and answer options)
LLM higher temp!
Parsing prompt subsitution (with the original question and the LLM's answer)
LLM
regexp extractor
Answer candidate
Repeat N times
Final answer
Majority voting
Sampler

Reflection

Question iterator
Critique prompt template
Fnal answer template
CoT prompt template
Prompt with substitution (question and answer options)
LLM
LLM
LLM
Parsing prompt template
Parsing prompt subsitution (with the original question and the LLM's answer)
LLM
regexp extractor
Answer

**Fig. 1.** Prompt design techniques used for enhanced medical Q&A.

*Dialogue-based agent.* Next, we implemented a dialogue-based agent, which has been shown to increase LLM task performance (Li et al., 2023; Nair et al., 2024). We employed two LLM instances, designated as a "professor" and a "student," to solve tasks collaboratively through guided interaction. Using a communicative approach, the professor breaks down complex problems into sub-tasks and delegates them to the student. The student executes these sub-tasks and provides feedback, enabling the professor to monitor progress and refine the plan as needed. This iterative communication loop facilitates efficient problem-solving by leveraging the specialized capabilities of each agent.

*LATS.* LATS is an approach that expands a tree search approach. It employs a Monte Carlo Tree Search (MCTS) algorithm to explore a space of potential reasoning and acting steps and uses an LLM to evaluate alternatives and refine strategies through self-reflection (Zhou et al., 2024).. This method has demonstrated effectiveness in programming, interactive Q&A, and solving complex mathematical problems (Long, 2023). This solution is expected to have high latency and cost.

### 2.4. Evaluation approach

After executing the algorithms, we evaluated their outputs. Evaluating Q&A on multiple-choice questions is relatively easy since you can algorithmically decide whether the answer is correct or incorrect. A key limitation of our methodology is that we did not assess the reasoning behind the answer, only the answer itself; precise evaluation metrics for open-ended reasoning chains remain an open challenge. As can be seen from the flows above (Figs. 1 and 2), we added a post-processing step to extract the decided answer from the final one produced by an LLM (and we used both an extraction based on regular expressions and LLM prompting).

We evaluated the percentage of correct answers, but when running agentic workloads, we also looked at additional metrics, such as the total number of input and output tokens consumed, the average latency of running an end-to-end workflow and a cost per answer (but only for models available as an API since it is impossible to estimate actual cost per answer for models that require deployment to your own infrastructure because of many different factors that would influence

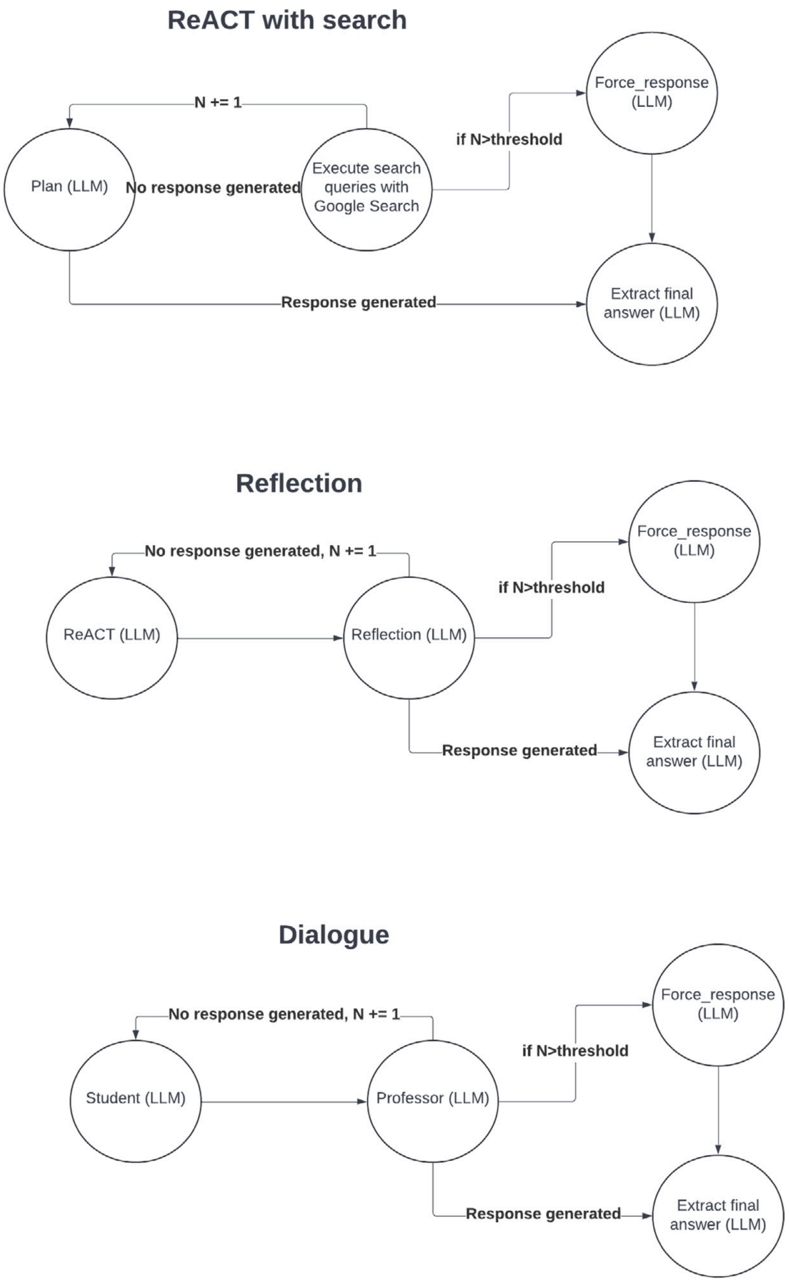


**Fig. 2.** Agentic workflows used for enhanced medical Q&A.

the final cost). A relatively small performance gain may not always justify the additional costs and latency.

### 2.4.1. Significance and error analysis

Statistical significance was assessed for each LLM using a one-sided Z-test for comparing two proportions (Kanji, 1999). Our rationale for this methodology was to provide a robust statistical basis for comparing the efficacy of different prompting strategies, moving beyond simple accuracy percentages. This test evaluated whether observed differences in performance between prompting strategies were statistically significant.

For each LLM, we tested:

*Null-hypothesis.* $p_0 = p$

*Alternative hypothesis.* $p_0 > p_1$ where $p_0$ represents the proportion of correct responses achieved by the highest-performing prompting technique, and $p_1$ represents the proportion achieved by the lowest-performing technique for the given LLM. We adopted the conventional threshold of $\alpha = 0.05$ to determine statistical significance.

To examine error consistency across experimental conditions, we analyzed the distribution of error frequencies. Specifically, we calculated the frequency of incorrect responses for each question across all experimental trials, encompassing both different prompting techniques and different LLMs. This analysis allowed us to determine whether certain questions consistently elicited errors regardless of the prompting strategy or model architecture employed.

## 3. Results

### 3.1. Prompt engineering on the MedQA dataset

Our initial analysis on the MedQA dataset revealed that large foundational models consistently outperformed smaller models, and that

**Table 1**
LLM performance on the MedQA dataset.

| foundational model | weights | model | % correct answers | the best prompt design algorithm | *p*-value |
|---|---|---|---|---|---|
| Claude 3.5-Sonnet | n/a | large, proprietary | 79.9% | CoT | < 0.01* |
| Gemini 1.5-pro-001 | n/a | large, proprietary | 67.0% | CoT | < 0.01* |
| Gemini 1.5-pro-002 | n/a | large, proprietary | 76.0% | CoT | < 0.01* |
| Llama 3.1 70B | 70B | large, OSS | 87.1% | self-consistency sampling | 0.01* |
| Llama 3.1 405B | 405B | large, OSS | 90.3% | self-consistency sampling | < 0.01* |
| Llama 3.2 90B | 90B | large, OSS | 84.8% | self-consistency sampling | 0.11* |
| Mistral Large@2407 | 123B | large, proprietary | 76.2% | CoT | < 0.01* |
| Gemma 2 2B instruct | 2B | small, OSS | 38.8% | vanilla | N/A |
| Gemma 2 9B instruct | 9B | small, OSS | 52.2% | self-consistency sampling | < 0.01* |
| Gemma 2 27B instruct | 27B | small, OSS | 54.3% | self-consistency sampling | < 0.01* |
| Gemini 1.5-flash-001 | n/a | small, proprietary | 59.1% | CoT | < 0.01* |
| Gemini 1.5-flash-001 | n/a | small, proprietary | 59.1% | CoT | < 0.01* |
| Llama 3.1 8B | 8B | small, OSS | 66.9% | CoT | < 0.01* |
| Mistral Nemo@2407 | 12B | small, OSS | 53.7% | CoT | < 0.01* |

* Results are statistically significant at a *p*-value of $\leq 0.05$.

**Table 2**
LLM performance on the custom neuro-oncology Q&A dataset.

| model | % correct | best chain | avg latency, sec | input + output tokens | $/request | *p*-value T1 | *p*-value T2 |
|---|---|---|---|---|---|---|---|
| Claude 3.5-Sonnet | 78.6% | CoT | 7.1 | 1006 + 517 | $0.01 | **<0.01** | 0.26 |
| Gemini 1.5-pro-001 | 69.6% | vanilla | 0.63 | 104 + 3 | $0.0001 | **<0.01** | 0.48 |
| Gemini 1.5-pro-002 | 70.8% | vanilla | 0.55 | 104 + 2 | $0.0001 | **<0.01** | 0.99 |
| Llama 3.1 405B | 77.7% | SCS* | 11 | 1032 + 12 | $0.005 | **0.025** | 0.35 |
| Llama 3.2 90B | 70.3% | CoT | 20 | 890 + 489 | $0.01 | 0.19 | 0.28 |
| Mistral Large@2407 | 76.0% | SR* | 26 | 3218 + 1656 | N/A | 0.17 | 0.27 |
| *New generation of models (update April 2025)* | | | | | | | |
| Gemini 2.0-flash-001 | 72.2% | vanilla | 0.36 | 103 + 2 | $0.00001 | 0.13 | 0.99 |
| Claude 3.5-Sonnet v2 | 81.5% | CoT | 7.34 | 937+385 | $0.02 | **<0.01** | 0.25 |
| Gemini 2.5-flash-exp | 81.9% | SCS* | 14 | 1034+61 | $0.0002 | **<0.01** | N/A |
| Gemini 2.5-pro-exp | 86.7% | SCS* | 81 | 1039+18 | $0.0015 | **<0.01** | 0.99 |
| DeepSeek v3 | 72.7% | vanilla | 1.3 | 100+2 | N/A | **<0.01** | N/A |
| gpt-4o | 78.4% | vanilla | 26 | 1079+25 | $0.0004 | **0.01** | N/A |
| gpt-4.1 | 81.7% | CoT | 38 | 4400+2102 | $0.007 | **<0.01** | **<0.01** |
| Mistral large @2411 | 81.7% | CoT | 13.9 | 986+516 | $0.05 | **0.01** | 0.43 |
| Llama 3.3 70B | 73.4% | SR* | 48.9 | 3563+1977 | $0.001 | 0.13 | 0.13 |

* SCS stands for self-consistency sampling, and SR stands for self-reflection.

advanced prompting strategies like CoT and self-consistency sampling yielded the best results for most models.

We evaluated LLM performance across various prompt design algorithms and reported the best experiment for every LLM (Table 1). "Large" LLMs significantly outperformed small ones. With standard ("vanilla") prompting, large LLMs achieved 77.1% accuracy (compared to just 50.1% for small ones). When using self-consistency sampling, accuracy slightly improved for both: 77.9% for foundational models and 51.2% for smaller models. CoT further boosted large LLMs' accuracy to 79.3%, while it slightly reduced the accuracy of smaller ones to 49.6%.

### 3.2. Prompt engineering on oncology Q&A dataset

On the specialized neuro-oncology dataset, models showed similar performance patterns to MedQA, but we observed significant trade-offs between accuracy and computational efficiency. Newer generation models introduced in April 2025 generally surpassed the accuracy of earlier models (Table 2).

Claude 3.5-Sonnet demonstrated superior performance (78.6% accuracy) utilizing CoT prompting while requiring moderate computational resources (7.1s average latency, 1,523 total tokens). Gemini models (1.5-pro-001/002) achieved lower accuracy (69.6% and 70.8% respectively) but demonstrated superior computational efficiency, with sub-second latency and minimal token utilization. Costs are based on API pricing as of July 2025. The "new" generation of models (e.g., Gemini 2.5-pro-exp, Claude 3.5-Sonnet v2, gpt-4.1) consistently posted higher accuracy scores than their predecessors, with several models exceeding 81% accuracy using CoT or SCS prompting.

In most cases, there is a statistically significant difference between the best and the worst performing strategy for each given foundational model (T1). However, we often found no significant difference between the best-performing prompting strategy and the baseline vanilla prompt (T2).

Test 1 (T1) null: there is no difference between the best and the worst prompting strategies Test 2 (T2) null: there is no difference between the best prompting strategy and the vanilla one

Performance analysis reveals a consistent but marginally reduced accuracy compared to the MedQA dataset. This is likely attributable to two primary factors. Firstly, the increased complexity of specialized neuro-oncology content. Secondly, potential dataset contamination in MedQA training data affects baseline performance metrics.

### 3.3. Agentic workflows with search grounding on oncology Q&A dataset

Integrating agentic workflows with external knowledge sources yielded variable performance improvements, with some architectures (e.g., LATS) boosting accuracy but at a substantial computational cost (Table 3). Model response to tested architectures was inconsistent; for instance, some Gemini models showed performance degradation, while Claude models consistently improved. It might be attributed to the fact that reasoning models incorporate certain agentic patterns, and hence wrapping the model with such architecture additionally decreases the performance, but we do not have clear evidence for this statement.

LATS achieved peak accuracy with Claude 3.5-Sonnet (82.9%), while ReACT and ReACT-reflect methodologies demonstrated comparable performance improvements (81.7%). However, these accuracy gains necessitated substantial computational overhead, with LATS requiring

**Table 3**
LLM performance on MedQA dataset on the custom neuro-oncology Q&A dataset with agentic workflows.

| model | ReACT | Plan | ReAct-reflect | ReAct-reflect (↑ $T$) | Dialogue | LATS |
|---|---|---|---|---|---|---|
| Claude 3.5-Sonnet | 81.7% | 81.2% | 81.7% | 79.8% | 81.0% | **82.9%** |
| Gemini 1.5-pro-001 | 63.4% | 62.2% | 40.4% | 41.8% | 47.7% | **71.5%** |
| Gemini 1.5-pro-002 | 59.4% | 61.3% | 68.4% | 63.2% | 67.5% | **74.1%** |
| Mistral Large@2407 | 72.0% | 42.8% | 67.9% | 68.9% | 74.6% | **75.8%** |
| *New generation of models (update April 2025)* | | | | | | |
| Gemini 2.0-flash-001 | 69.6% | 76.0% | 67.2% | 69.4% | 64.8% | **76.2%** |
| Claude 3.5-Sonnet v2 | **85.5%** | 81.7% | 70.8% | 71.3% | 81.7% | 85.0% |
| Gemini 2.5-flash-exp | 79.6% | 76.2% | 74.6% | 68.6% | 77.2% | **85.5%** |
| Gemini 2.5-pro-exp | 67.5% | 85.5% | 82.4% | 83.4% | 84.1% | **88.8%** |
| gpt-4o | 78.4% | **78.9%** | 75.1% | 75.2% | 77.4% | 77.3% |
| gpt-4.1 | 76.5% | 82.4% | 81.0% | 82.4% | **83.6%** | 81.6% |

**Table 4**
Significance analysis of performance difference across different agentic workflows.

| model | $p$-value | $H_0$ rejected | base, $/request | best, $/request |
|---|---|---|---|---|
| Claude 3.5-Sonnet | <0.01 | yes | $0.00043 | $0.15 |
| Gemini 1.5-pro-001 | 0.21 | no | $0.00010 | $0.0098 |
| Gemini 1.5-pro-002 | 0.07 | no | $0.00012 | $0.041 |
| Mistral Large@2407 | – | – | – | – |
| *New generation of models (update April 2025)* | | | | |
| Gemini 2.0-flash-001 | 0.035 | yes | $0.000017 | $0.0021 |
| Claude 3.5-Sonnet v2 | <0.01 | yes | $0.0011 | $0.0052 |
| Gemini 2.5-flash-exp | 0.01 | yes | $0.000016 | $0.0052 |
| Gemini 2.5-pro-exp | 0.09 | no | $0.00014 | $1.0 |
| gpt-4o | 0.43 | no | $0.00044 | $0.0083 |
| gpt-4.1 | <0.01 | yes | $0.00023 | $0.0070 |

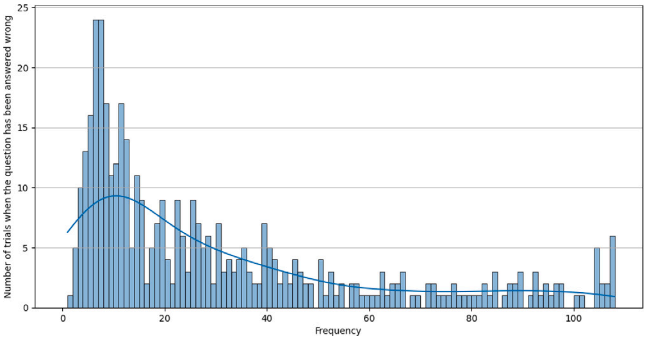


**Fig. 3.** Frequency of errors per question across all trials.

107 s of average latency and consuming 18,628 input tokens compared to baseline prompting's 1-second latency and 114 input tokens. Notably, different model architectures exhibited varying responses to external knowledge integration, with Gemini models showing performance degradation in specific agentic patterns while Claude 3.5-Sonnet maintained consistent improvements across all configurations. The newer generation models (Table 3, April 2025) generally benefited from agentic workflows, with models like Gemini 2.5-pro-exp and Claude 3.5-Sonnet v2 showing significant accuracy gains, though statistical significance varied (Table 4).

### 3.4. Error analysis on oncology dataset

An analysis of error distribution showed that errors were not evenly distributed; rather, a small subset of questions was responsible for a large number of errors, while most questions were answered correctly by most models.

The chart below represents a histogram of the error frequency of errors per question across all trials (different foundational models and different prompting techniques). We can clearly observe that the distribution is heavily skewed to the right with a relatively small left tail. In other words, there is a cluster of questions that are "easy to answer" for an LLM, but it is difficult to clearly distinguish a cluster of "hard to answer" questions since we do not observe a significantly high left tail (see Fig. 3).

### 3.5. Summary of results

The results presented in this section revealed four principal trends. First, large foundational models (e.g., Llama 3.1 405B, Claude 3.5-Sonnet) consistently outperformed smaller models (e.g., Gemma 2 2B) on specialized medical datasets (Section 3.1). Second, advanced prompting strategies like SCS and CoT generally provided a statistically significant, albeit modest, performance uplift for large models but had a neutral or negative effect on smaller models (Section 3.1, 3.2). Third, agentic workflows (Table 5) showed potential to further increase accuracy, with LATS achieving the highest average performance (79.9%), but this came at a substantial computational cost, increasing latency from 2 s to 110 s and token counts by over 100-fold (Section 3.3). Finally, performance was highly model-dependent, with newer-generation models (e.g., Gemini 2.5-pro-exp) showing superior accuracy, while certain models (e.g., Gemini 1.5-pro) exhibited performance degradation with specific agentic workflows (Table 3). These findings will be interpreted further in the Discussion section.

## 4. Discussion

This study investigated the differential impact of prompt design methods on complex medical Q&A tasks, using both parametric and non-parametric memory architectures across several basic models. We aimed to determine how prompting strategies, from simple in-context learning to complex agentic workflows, influence performance across LLMs of varying sizes on specialized medical questions. The results confirmed a significant performance advantage for large-scale models over their smaller counterparts, although the effectiveness of specific prompt design strategies varied considerably across model architectures. While complex agentic workflows and tool-augmented LLMs generally achieved superior performance compared to conventional in-context learning approaches, the most significant improvements were observed through integrating external knowledge bases and implementing the ReACT pattern. Notably, more sophisticated agent-based approaches either failed to deliver meaningful performance improvements or showed only marginal benefits, as exemplified by the LATS algorithm, which incurred more than a fivefold increase in computational cost and latency without commensurate performance gains. We suggest that more complex and costly agentic patterns are justified in scenarios where the cost of failure is much higher than the cost of computation. These patterns trade efficiency (low latency, low cost) for robustness (higher accuracy, self-correction, and explainability).

Prior research has extensively investigated medical Q&A performance. Lammert et al. demonstrated the efficacy of RAG in precision oncology, achieving 94.7% concordance with expert recommendations

**Table 5**
Average performance of agentic workflows on the custom neuro-oncology Q&A dataset.

| algorithm | % correct answers | avg latency, sec | avg input tokens | avg output tokens |
|---|---|---|---|---|
| vanilla | 76.7% | 2 | 110 | 2 |
| self-consistency sampling | 76.7% | 17 | 994 | 29 |
| CoT | 74.0% | 10 | 1155 | 496 |
| self-reflection | 70.7% | 29 | 4074 | 1551 |
| ReACT | 73.9% | 14 | 2010 | 493 |
| plan | 72.8% | 13 | 6125 | 250 |
| ReACT-reflect | 71.0% | 24 | 2998 | 798 |
| ReACT-reflect ((↑ $T$) | 62.9% | 25 | 2985 | 798 |
| Dialogue | 74.0% | 27 | 4032 | 809 |
| LATS | 79.9% | 110 | 16 394 | 3612 |

through domain-specific knowledge integration (Lammert et al., 2024; Singhal et al., 2023). Similarly, Nair et al. (2024) explored dialogue-enabled resolving agents (DERA) with GPT-4 and found only modest improvements over baseline performance. This aligns with our findings regarding the limited returns of complex agentic approaches. Our results corroborate these findings; while complex agentic workflows and tool-augmented language models generally performed better than conventional in-context learning, the most substantial improvements emerged from external knowledge integration and ReACT pattern implementation.

Singhal et al. (2023) established benchmark performance levels for LLMs on medical licensing examination questions, achieving 67.6% accuracy through zero-shot approaches. Our investigation extends these findings by demonstrating comparable performance metrics while emphasizing practical implementation constraints. Notably, Bedi et al. (2024) highlighted significant limitations in current evaluation methodologies, particularly the prevalence of standardized benchmarks over real patient data. While this study introduces a novel evaluation framework, it acknowledges limitations inherent in multiple-choice assessments.

The substantial performance disparity observed between small open-source LLMs and their larger counterparts (51.2% versus 79.3% Q&A accuracy) highlights a critical infrastructure challenge for healthcare institutions. This performance gap presents medical organizations with two strategic alternatives. Organizations can invest in substantial computational infrastructure to support larger open-source models like Llama 405B, though this approach requires significant capital expenditure and technical expertise. Alternatively, institutions may implement cloud-based solutions while developing robust data protection frameworks, a strategy that requires careful consideration of regulatory compliance and security protocols (Singhal et al., 2025; Yang et al., 2025). When implementing AI systems in clinical settings, healthcare institutions must carefully evaluate the trade-offs between model performance, infrastructure costs, and data security requirements.

Beyond accuracy metrics, our analysis revealed critical practical constraints for medical AI deployment. For instance, incorporating self-reflection mechanisms into the prompting algorithm yielded unexpected results, with minimal improvement in performance metrics. This finding, which diverges from previous research suggesting the efficacy of self-reflection, implies that not all complex prompting techniques add clinical value. End-to-end latency and computational costs increased substantially with agentic workflows. These factors, often overlooked in benchmark evaluations, are critical considerations for a system's viability in a clinical setting.

The study has several limitations. First, as noted by previous works, multiple-choice question datasets have inherent limitations for medical AI evaluation, particularly in their inability to assess clinical reasoning depth and contextual understanding. Second, the potential for dataset contamination presents a significant methodological concern. The MedQA dataset's long-term public availability, for example, raises questions about performance inflation through potential training data contamination. Third, the inherent non-deterministic nature of LLM introduces variability in outputs, even when controlling for parameters such as temperature and using random seeds during answer generation. While these variations likely have minimal impact on overall performance metrics, they affect the strict reproducibility of results. This reproducibility challenge is compounded by the frequent updates to commercial LLMs, which complicates longitudinal performance comparisons. Finally, runtime estimates are hardware-dependent, suggesting that current trade-offs between accuracy and computational overhead will likely diminish as processing capabilities improve.

Future research should prioritize moving beyond the limitations of current benchmarks. We fully acknowledge the need for more sophisticated evaluation datasets that better reflect the complex contexts in which we aim to deploy an LLM as decision-support tools. Concrete directions include developing benchmarks based on long-form clinical narratives, evaluating a model's ability to synthesize information from disparate patient records, or assessing their interaction in simulated diagnostic dialogues. Furthermore, this study evaluated only average Q&A accuracy, without investigating the reasons for an LLM to fail. Future research that explores reasoning traces and tool-calling trajectories could reveal deeper reasons for performance degradation. Additionally, the recently released reasoning models, which were not evaluated in this study, warrant investigation. It would be valuable to determine whether these new architectures benefit from the agentic approaches studied here and how to best enhance them with access to external knowledge bases.

Statement of Significance

| Summary | Description |
|---|---|
| Problem | LLM performance in specialized medical Q&A |
| What is Already Known | LLMs show promise in medical Q&A, with various prompting techniques (CoT, RAG) improving accuracy. MedQA is a common benchmark, but has limitations. |
| What this Paper Adds | A comprehensive comparison of prompting techniques (including novel agentic workflows like LATS) on LLMs of varying sizes in neuro-oncology Q&A. Focuses on trade-offs. |

## 5. Conclusion

This study investigated various prompting strategies for medical Q&A, aiming to identify methods that balance accuracy and efficiency. We demonstrated that approaches combining ReACT patterns with domain knowledge led to superior performance. However, the significant performance disparity between large and small language models presents healthcare institutions with a critical choice. This choice involves balancing infrastructure costs and data protection, particularly when deciding between resource-intensive on-premises deployments to run large open-source models and more accessible cloud-based solutions. Acknowledging this study's limitations, such as the reliance on multiple-choice questions, which may not fully capture real-world clinical complexity, and not evaluating the underlying reasons for LLM failures, future work should prioritize the development of more diverse clinical evaluation methods. Future research is also needed to develop practical privacy-preserving architectures that maintain model performance while reducing computational overhead. This would enable LLM fine-tuning without centralizing sensitive data. Successfully navigating these trade-offs between performance, cost, and security is essential for the responsible and effective deployment of LLMs as decision-support tools in clinical settings.

## CRediT authorship contribution statement

**Leonid Kuligin:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Conceptualization. **Jacqueline Lammert:** Conceptualization, Formal analysis, Supervision, Writing – review & editing. **Aleksandr Ostapenko:** Software. **Keno Bressem:** Conceptualization, Supervision, Writing – review & editing. **Martin Boeker:** Conceptualization, Supervision. **Maximilian Tschochohei:** Conceptualization, Formal analysis, Supervision, Writing – review & editing, Validation.

## Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

## Data availability

The authors do not have permission to share data.

## References

Anthropic (2024). Claude 3.5 sonnet. Anthropic.

Arora, A., Alderman, J. E., Palmer, J., S., G., Laws, E., McCradden, M. D., et al. (2023). The value of standards for health datasets in artificial intelligence-based applications. *Nature Medicine*, *29*(11), 2929.

Bedi, S., Liu, Y., Orr-Ewing, L., Dash, D., Koyejo, S., Callahan, A., et al. (2024). A systematic review of testing and evaluation of healthcare applications of Large Language Models (LLMs). *MedRxiv*, http://dx.doi.org/10.1101/2024.04.15.24305869.

Brown, T., Mann, B., Ryder, N., Subbiah, M., Kaplan, J. D., Dhariwal, P., et al. (2020). Language models are few-shot learners. In H. Larochelle, M. Ranzato, R. Hadsell, M. Balcan, H. Lin (Eds.), *Advances in neural information processing systems*: *vol. 33*, (pp. 1877–1901). Curran Associates, Inc..

Das, J. M. (2023). *Neuro-oncology explained through multiple choice questions*. Springer Nature.

Google (2024). Gemma open models. Google.

Google (2025). *Gemini models*. Google.

Hosseini, P., Sin, J. M., Ren, B., Thomas, B. G., Nouri, E., Farahanchi, A., et al. (2024). A benchmark for long-form medical question answering. In *Advancements in medical foundation models: explainability, robustness, security, and beyond*.

Jin, D., Pan, E., Oufattole, N., Weng, W.-H., Fang, H., & Szolovits, P. (2021). What disease does this patient have? A large-scale open domain question answering dataset from medical exams. *Applied Sciences*, *11*(14), http://dx.doi.org/10.3390/app11146421.

Kanji, G. K. (1999). *100 statistical tests*. SAGE.

Kavukcuoglu, K. (2024). Introducing Llama 3.1: Our most capable models to date. Google.

Lammert, J., Dreyer, T., Mathes, S., Kuligin, L., Borm, K. J., Schatz, U. A., et al. (2024). Expert-guided large language models for clinical decision support in precision oncology. *JCO Precision Oncology*, http://dx.doi.org/10.1200/PO-24-00478.

Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., et al. (2020). Retrieval-augmented generation for knowledge-intensive NLP tasks. In *Proceedings of the 34th international conference on neural information processing systems*. Red Hook, NY, USA: Curran Associates Inc..

Li, G., Al Kader Hammoud, H. A., Itani, H., Khizbullin, D., & Ghanem, B. (2023). CAMEL: communicative agents for "mind" exploration of large language model society. In *Proceedings of the 37th international conference on neural information processing systems*. Red Hook, NY, USA: Curran Associates Inc.

Liu, A., Feng, B., Xue, B., Wang, B., Wu, B., Lu, C., Zhao, C., Deng, C., Zhang, C., Ruan, C., Dai, D., Guo, D., .... Pan, Z. (2024). DeepSeek-V3 Technical Report. arXiv:2412.19437.

Liu, Y., Lo, S. K., Lu, Q., Zhu, L., Zhao, D., Xu, X., et al. (2025). Agent design pattern catalogue: A collection of architectural patterns for foundation model based agents. *Journal of Systems and Software*, *220*(C), http://dx.doi.org/10.1016/j.jss.2024.112278.

Long, J. (2023). Large language model guided tree-of-thought.

Meta A. I. (2024a). Introducing llama 3.1: Our most capable models to date.

Meta A. I. (2024b). Meta-llama-3-8b. Meta AI.

Mistral (2024a). Mistral-large-instruct-2407. Mistral.

Mistral (2024b). Mistral-nemo-instruct-2407. Mistral.

Moreno, A. C., & Bitterman, D. S. (2024). Toward clinical-grade evaluation of large language models. *International Journal of Radiation Oncology, Biology, Physics*, *118*(4), 916.

Nair, V., Schumacher, E., Tso, G., & Kannan, A. (2024). DERA: Enhancing large language model completions with dialog-enabled resolving agents. In T. Naumann, A. Ben Abacha, S. Bethard, K. Roberts, & D. Bitterman (Eds.), *Proceedings of the 6th clinical natural language processing workshop* (pp. 122–161). Mexico City, Mexico: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2024.clinicalnlp-1.12.

OpenAI (2025). GPT-4.1. OpenAI.

Pichai, S. (2024). *Introducing Gemini 2.0: our new AI model for the agentic era*. Google.

Renze, M., & Guven, E. (2024). Self-reflection in LLM agents: Effects on problem-solving performance. arXiv preprint arXiv:2405.06682.

Schlag, I., Sukhbaatar, S., Celikyilmaz, A., tau Yih, W., Weston, J., Schmidhuber, J., et al. (2023). Large language model programs. arXiv abs/2305.05364, arXiv:2305.05364.

Singhal, K., Azizi, S., Tu, T., Mahdavi, S. , S., Cole-Lewis, H., Pfohl, S., Payne, P., Seneviratne, M., Gamble, P., Kelly, C., Babiker, A., Schärli, N., Chowdhery, A., Mansfield, P., Demner-Fushman, D., .... Natarajan, V. (2023). Large language models encode clinical knowledge. *Nature*, *620*(7972), 172–180.

Singhal, K., Tu, T., Gottweis, J., Sayres, R., Wulczyn, E., Amin, M., Hou, L., Clark, K., Pfohl, R. , S., Cole-Lewis, H., Neal, D., Rashid, M. , Q., Schaekermann, M., Wang, A., Dash, D., Chen, H. , J., Shah, H. , N., Lachgar, S., Mansfield, A. , P., .... Natarajan, V. (2025). Toward expert-level medical question answering with large language models. *Nature Medicine*, http://dx.doi.org/10.1038/s41591-024-03423-7.

Vatsal, S., & Dubey, H. (2024). A survey of prompt engineering methods in large language models for different NLP tasks. arXiv preprint arXiv:2407.12994.

Wang, X., Wei, J., Schuurmans, D., Le, Q. V., Chi, E. H., Narang, S., et al. (2023). Self-consistency improves chain of thought reasoning in language models. In *The eleventh international conference on learning representations*. OpenReview.net.

Wang, L., Xu, W., Lan, Y., Hu, Z., Lan, Y., Lee, R. K.-W., et al. (2023). Plan-and-solve prompting: Improving zero-shot chain-of-thought reasoning by large language models. In A. Rogers, J. Boyd-Graber, N. Okazaki (Eds.), *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: long papers)* (pp. 2609–2634). Toronto, Canada: Association for Computational Linguistics, http://dx.doi.org/10.18653/v1/2023.acl-long.147.

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., et al. (2022). *Chain-of-thought prompting elicits reasoning in large language models*. Red Hook, NY, USA: Curran Associates Inc..

Wu, Q., Bansal, G., Zhang, J., Wu, Y., Li, B., Zhu, E., et al. (2023). AutoGen: Enabling next-gen LLM applications via multi-agent conversation. arXiv preprint arXiv:2308.08155.

Xiao, H., Zhou, F., Liu, X., Liu, T., Li, Z., Liu, X., et al. (2025). A comprehensive survey of large language models and multimodal large language models in medicine. *Information Fusion*, *117*, Article 102888. http://dx.doi.org/10.1016/j.inffus.2024.102888.

Yang, Y., Jin, Q., Zhu, Q., Wang, Z., Erramuspe Álvarez, F., Wan, N., et al. (2025). Beyond multiple-choice accuracy: Real-world challenges of implementing large language models in healthcare. *Annual Review of Biomedical Data Science*, *8*(Volume 8, 2025), 305–316. http://dx.doi.org/10.1146/annurev-biodatasci-103123-094851.

Yao, S., Zhao, J., Yu, D., Du, N., Shafran, I., Narasimhan, K., et al. (2023). React: Synergizing reasoning and acting in language models. In *International conference on learning representations*.

Yu, Z., He, L., Wu, Z., Dai, X., & Chen, J. (2023). Towards better chain-of-thought prompting strategies: A survey. arXiv preprint arXiv:2310.04959.

Zhou, A., Yan, K., Shlapentokh-Rothman, M., Wang, H., & Wang, Y.-X. (2024). Language agent tree search unifies reasoning, acting, and planning in language models. In *Proceedings of the 41st international conference on machine learning*. JMLR.org.