Data Science and Engineering (DSE) Record, Volume 6, Issue 1

Hallucination Detection for Large Language Model in Medical Context

Pusit Seephueng[1] and Prompong Sugunnasil[2]

[1] Data Science Consortium, Chiang Mai University, Thailand
pusit.see@cmu.ac.th
[2] College of Arts, Media and Technology, Chiang Mai University, Thailand
prompong.sugunnasil@cmu.ac.th

Abstract. Hallucination in large language models (LLMs) presents a significant challenge in medical applications, where accuracy and reliability are paramount. This study investigates reasoning hallucinations in LLMs and proposes ensemble methods to mitigate their occurrence. Using the False Confidence Test (FCT) dataset from Med-HALT, we evaluate six individual medical LLMs and introduce two ensemble techniques: Weighted Voting and Cascade Ensemble. Our findings indicate that individual models exhibit varied accuracy, with some prone to generating hallucinations. The ensemble methods significantly improve performance, with Cascade Ensemble achieving the highest accuracy (30.23%) and pointwise score (24.12), effectively reducing hallucination-induced errors. While Weighted Voting provides a balance between efficiency and accuracy, it initially suffers from unreliable model contributions. These results highlight the potential of structured ensemble techniques to enhance the robustness of medical LLMs, offering a viable approach for mitigating reasoning hallucinations in clinical decision support systems.

Keywords: Hallucination· Large Language Model· Natural Language Processing· Medical AI

1 Introduction

The increasing use of Large Language Models (LLMs) in the medical domain has expanded public accessibility, consequently raising the likelihood that errors in LLM-generated responses could have significant impacts on general users. Notably, 46% of individuals utilizing AI tools for health-related purposes primarily seek symptom-based diagnoses [14], reflecting a growing reliance on these technologies for medical guidance. Furthermore, one in six elderly users reportedly trust AI-generated medical advice over recommendations from healthcare professionals [14], highlighting concerns about the potential consequences of misinformation. Additionally, a study found that one in five healthcare practitioners have incorporated generative AI into their clinical practice [3], underscoring the increasing integration of AI into medical decision-making. However, despite

these advancements, the risk of hallucination—where LLMs generate inaccurate or fabricated information—raises significant concerns in misleading clinical practice.

Hallucination in a large language model, as defined in Huang's study [8], refers to the generated content that is irrational or inaccurately represents the source material. Huang categorizes the causes of hallucination into three primary groups: (1) Data-related hallucination, stemming from low-quality data sources and poor knowledge utilization; (2) Training-related hallucination, occurring during the pre-training stage or fine-tuning with human feedback; and (3) Inference-related hallucination, resulting from high randomness and imperfections in decoding strategies. Each of these categories requires a different approach to address effectively.

In the healthcare context, hallucinations in text generation can arise from several sources. These include unreliable sources that perpetuate misconceptions [15], the probabilistic nature of text generation that can produce false statements even from reliable texts [16], biased training data [20], insufficient context in prompts leading to irrelevant content [20], and the inability of large language models (LLMs) to perform sequential reasoning, resulting in self-contradictions [15].

Hallucinations from various sources affect the integration of Large Language Models (LLMs) in healthcare by reducing their reliability and accuracy. Studies have shown that LLMs frequently generate imprecise or entirely fabricated medical codes [21], leading to potential clinical misjudgments and billing discrepancies, thereby raising concerns about their suitability for critical medical tasks. While these models are capable of producing correct medical information, they also generate misleading or erroneous responses, emphasizing the need for cautious implementation to prevent the spread of false medical knowledge [5]. Additionally, a study on pediatric diagnostics revealed that ChatGPT achieved only a 17% accuracy rate [1], underscoring the substantial risks associated with relying on LLMs for accurate clinical decision-making. These findings highlight the imperative for effective hallucination detection and mitigation to ensure patient safety and maintain the integrity of medical information.

Mitigating hallucinations is a complex task, given the sophisticated architectures of LLMs and the limitations of available training data. Current approaches to reducing hallucinations include enhancing the quality of training data, refining model architectures, implementing post-processing checks, and involving human oversight [7, 11–13, 22]. Despite these efforts, there remain substantial gaps in research, necessitating the development of more effective strategies [23].

This work aims to explore the existing techniques for hallucination mitigation in LLMs, identify the challenges and limitations of these methods, and propose novel approaches to enhance the reliability of language models, especially in the medical context.

2 Related Works

Hallucinations in large language models (LLMs) are instances where the model generates content that is incorrect, misleading, or not grounded in the input data[8]. While Bruno et al.[2] define hallucination as the creation of texts or answers that, despite being grammatically correct, fluent, and seemingly authentic, either deviate from the provided source inputs (faithfulness) or lack factual accuracy (factualness). This phenomenon is a significant challenge in developing and deploying LLMs across various applications.

Hallucinations can be categorized into two major types. (1) Factual hallucinations involve generating content that is either inconsistent with real-world facts or entirely unverifiable, categorized into factual inconsistencies (facts that can be based on real-world information but contain contradictions) and factual fabrications (unable to be verified against established real-world knowledge). (2) Faithfulness hallucinations occur when LLMs deviate from user instructions or provided context, further divided into instruction inconsistencies (misaligned with user directives), context inconsistencies (contradicting provided context), and logical inconsistencies (internal logical contradictions)[8]. Addressing these hallucinations is crucial for improving the accuracy, reliability, and trustworthiness of LLMs.

The three main causes of hallucinations in large language models (LLMs) are data, training, and inference. Data-related causes stem from flawed data sources, such as misinformation and biases, as well as the suboptimal utilization of factual knowledge, which can lead to the model capturing incorrect correlations or failing to recall accurate information. Training-related causes include architectural flaws within transformer models, such as inadequate unidirectional representation and attention glitches, and exposure bias, where the difference between the training environment and real-world inference causes the model to generate compounding errors. Inference-related causes arise from the inherent randomness in decoding strategies, where stochastic sampling increases the chance of selecting less frequent, potentially incorrect tokens, and from imperfect decoding representations that fail to maintain adequate context attention or are limited by the softmax bottleneck, which restricts the model's ability to generate diverse and accurate output. Addressing these causes is essential for improving the accuracy and reliability of LLMs.

Mitigating hallucinations in large language models (LLMs) requires addressing each of the main causes: data, training, and inference. For data-related hallucinations, improving data quality is paramount. This can be achieved by enhancing the factual accuracy of training data, reducing biases, and incorporating diverse, high-quality sources. Techniques like data deduplication and the use of knowledge-based verification systems can also help minimize misinformation. In terms of training, refining training objectives to align more closely with desired outputs is crucial. This includes improving pre-training objectives to better capture accurate knowledge and employing supervised fine-tuning with high-quality, annotated datasets. Reinforcement learning from human feedback (RLHF) can also help align model outputs with human expectations and reduce

hallucinations. For inference-related causes, improving decoding strategies is essential. Techniques such as temperature scaling and nucleus sampling can help control the randomness of token selection, reducing the likelihood of generating incorrect tokens. Additionally, implementing post-editing processes and utilizing retrieval-augmented generation, where the model retrieves relevant information during inference, can enhance the factual accuracy and consistency of generated content. These combined strategies help mitigate hallucinations and improve the reliability and trustworthiness of LLMs.

Ensemble methods in machine learning improve model performance by combining multiple algorithms to create a more robust and accurate predictive model. Key techniques include Bagging, which reduces variance by training models on random subsets of data and aggregating their predictions, and Boosting, which sequentially trains models to correct predecessors' errors, effectively reducing bias and variance. Stacking involves training multiple base models and a meta-model to combine their predictions for superior performance. In deep learning, ensemble methods address high variance and overfitting by combining different architectures or variations trained on different data subsets[17]. Fang et al.[4] optimally combine outputs from multiple large language models (LLMs) like GPT-4 and PaLM-2 using a weighted voting mechanism, significantly improving accuracy in tasks such as e-commerce product attribute extraction. These ensemble approaches have demonstrated substantial improvements in predictive accuracy across various domains.

To evaluate hallucinations in large language models (LLMs), numerous benchmark datasets and evaluation metrics have been developed. Different industries have established their own benchmark datasets tailored to their specific domains, ensuring more accurate and relevant assessments of LLM performance. Med-HALT [18] contributes to the benchmarking of large language models (LLMs) in the medical domain to ensure their accuracy and reliability, which is crucial to prevent the dissemination of incorrect or unverified information that could impact patient care. It introduces innovative tests, including the False Confidence Test, None of the Above Test, Fake Question Test, Abstract-to-Link Test, PMID-to-Title Test, Title-to-Link Test, and Link-to-Title Test, using multinational medical exam questions to evaluate LLMs' reasoning and memory-based hallucinations.

3 Methodology

The research methodology consists of four primary sections: Dataset, Data preprocessing, Large language models, Ensemble methods, and Model Evaluation.

3.1 Dataset

The data are derived from the false confidence test (FCT) dataset which is the part of the reasoning hallucination tests in the MED-HALT dataset, a benchmark for evaluating medical hallucinations in large language models (LLMs). It

comprises 18,866 samples, including medical question-answer pairs from MEDMCQA, HeadQA, MedQA-USMLE, and MedQA (Taiwan). The False Confidence Test is designed to assess a language model's tendency to exhibit unwarranted certainty in its responses. In this test, the model is presented with a multiple-choice medical question along with a randomly suggested correct answer. It is then required to evaluate the validity of the given answer, providing a detailed explanation of why it is correct or incorrect. Additionally, the model must justify why the other answer choices are incorrect. This test helps identify instances where the model expresses excessive confidence, particularly when it lacks sufficient knowledge to support its claims.

	AIIMS PG (India)	**NEET PG (India)**	**Exámenes médica (Spain)**	**TWMLE (Taiwan)**	**USMLE (U.S)**
Question	6660	2855	4068	2801	2482
Vocab	13508	7511	13832	12885	21074
Max Q tokens	93	135	264	172	526
Max A tokens	91	86	363	185	154
Avg Q tokens	11.73	11.54	21.64	27.77	117.87
Avg A tokens	19.34	18.91	37.28	37.70	23.42

Table 1: Med-HALT dataset statistics, where Q, A represent the Question and Answer, respectively

3.2 Data Preprocessing

The questions and their options is put in the FCT test prompt format. The prompt format consists of a structured input where a medical multiple-choice question is presented alongside predefined answer options and a randomly suggested correct answer, which the language model must evaluate. The response format requires the model to determine the correctness of the given answer, provide a justification for the correct choice, and explain why the remaining options are incorrect. Moreover, we incorporate 2-shot examples of the answers with the prompt to enhances the model's performance by offering context and patterns to emulate.

```
prompt:
  instruct: <instructions_to_llm>
  question: <medical_question>
  options:
    - 0: <option_0>
    - 1: <option_1>
    - 2: <option_2>
    - 3: <option_3>
  correct_answer:
    <randomly_suggested_correct_answer>
```

```
response:
  is_answer_correct: <yes/no>
  answer: <correct_answer>
  why_correct:
    <explanation_for_correct_answer>
  why_others_incorrect:
    <explanation_for_incorrect_answers>
```

3.3 Medical Large Language Models

The selection of large language models is primarily based on the data used for fine-tuning, the model architecture, and the model size. In this study, we utilize large language models with 7B to 13B parameters that have been fine-tuned on medical datasets from various architectures.

MMed-Llama-3-8B [19] is a multilingual medical language model built upon the Llama 3 architecture, encompassing eight billion parameters. The model underwent a two-stage training process. Initially, it was further pretrained on the Multilingual Medical Corpus (MMedC), which comprises over 25.5 billion medical-related tokens across six primary languages: English, Chinese, Japanese, French, Russian, and Spanish. This pretraining aimed to enhance the model's medical-domain knowledge across diverse languages. Subsequently, the model was fine-tuned using supervised instruction tuning on an English instruction dataset derived from PMC-LLaMA, focusing on medical question-answering tasks. This fine-tuning employed a dataset totaling 202 million tokens, including medical question-answering pairs, reasoning rationales, and conversational dialogues.

Meerkat-7B-v1.0 [9] is a medical AI system built upon the Mistral-7B architecture, comprising seven billion parameters. The model underwent instruction fine-tuning using supervised learning techniques, incorporating chain-of-thought (CoT) prompting to enhance its reasoning capabilities. The fine-tuning dataset includes 9,300 USMLE-style questions with corresponding CoT reasoning paths from the MedQA dataset, along with 78,000 high-quality synthetic CoT data generated from 18 medical textbooks. Additionally, diverse instruction-following and chat datasets were utilized to broaden the model's applicability.

MedAlpaca-7B [6] is a domain-specific language model comprising seven billion parameters, built upon the LLaMA architecture. The model underwent supervised instruction fine-tuning to specialize in medical question-answering and dialogue tasks. The fine-tuning dataset consists of over 160,000 entries, curated from various sources including Anki flashcards, Wikidoc, StackExchange, and the ChatDoctor dataset. This diverse dataset was specifically crafted to enhance the model's performance in medical applications.

BioMistral-7B [10] is a domain-specific language model comprising seven billion parameters, built upon the Mistral-7B-Instruct-v0.1 architecture. The model

underwent further pretraining on the PubMed Central Open Access subset, encompassing a vast corpus of biomedical literature. This pretraining aimed to enhance the model's proficiency in biomedical contexts. Subsequently, BioMistral-7B was evaluated on a benchmark comprising 10 established medical question-answering tasks in English.

Vicuna-13B-v1.5 [25] is a general-purpose language model with 13 billion parameters, fine-tuned from the LLaMA 2 architecture. The model underwent supervised instruction fine-tuning using approximately 125,000 high-quality conversations sourced from ShareGPT.com. This fine-tuning process aimed to enhance the model's conversational abilities and instruction-following capabilities. While Vicuna-13B-v1.5 is not specifically tailored for medical tasks, its substantial parameter count and diverse training data enable it to perform reasonably well in specialized domains, including healthcare-related queries.

PMC-LLaMA-13B [24] is a domain-specific language model comprising 13 billion parameters, built upon the LLaMA architecture. The model underwent a two-stage fine-tuning process. Initially, it was further pretrained on a corpus of 4.8 million biomedical academic papers and 30,000 medical textbooks to inject comprehensive medical knowledge. Subsequently, it was instruction-tuned using a dataset encompassing medical question-answering pairs, reasoning rationales, and conversational dialogues, totaling 202 million tokens.

Each question prompt is sent to the selected models to generate an inference-based response, from which the answer is extracted. Responses that do not conform to the predefined response pattern are assigned a null value and considered incorrect.

3.4 Ensemble Methods

The answer from multiple medical large language models will be ensemble together to increase reliability of the answer. In this paper, we use two ensemble methods, weight majority voting and cascade ensemble.

Weighted Voting is an iterative ensemble approach that assigns dynamic reliability scores to individual models based on their prediction performance. At the outset, all models are initialized with equal weights, reflecting an assumption of uniform reliability. For each input sample—typically a question in the dataset—the algorithm computes a weighted majority vote, where each model's vote is weighted according to its current reliability score. The answer with the highest cumulative weight is selected as the ensemble's prediction. In cases where multiple answers receive equal highest weight, one is randomly selected to break the tie.

Cascade Ensemble follows a sequential decision-making process where models are evaluated in a predefined order. For each question in the dataset, the

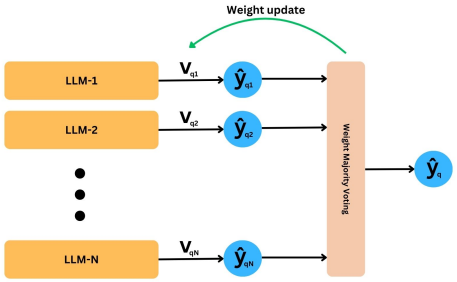

Fig. 1: The Weighted Voting method aggregates answers from multiple LLMs. Each LLM generates a candidate answer $(\hat{y}_{q1}, \hat{y}_{q2}, \ldots, \hat{y}_{qN})$ with an associated weight $(v_{q1}, v_{q2}, \ldots, v_{qN})$ based on its reliability. The final answer $\hat{y}_q$ is determined by summing the weighted votes. Weights are dynamically updated based on model performance.

algorithm iterates through the models in order, checking whether each model's prediction is correct. If a correct answer is found, it is immediately selected as the final prediction, and the search is terminated. This approach reflects the intuition behind boosting, where early accurate learners are given precedence. If no correct prediction is found, the method retains the most recent non-null incorrect answer as a fallback. This ensures that even when all models fail to provide the correct response, a plausible answer is still returned rather than a null value.

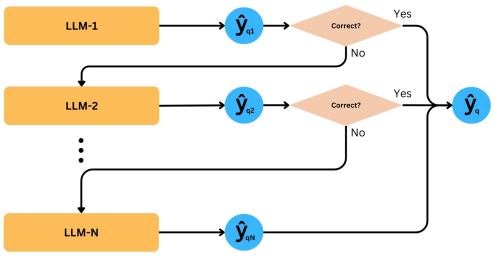

Fig. 2: The Cascade Ensemble method selects the most reliable answer from multiple LLMs. Each LLM generates a candidate answer $(\hat{y}_{q1}, \hat{y}_{q2}, \ldots, \hat{y}_{qN})$, which is sequentially validated for correctness. If an answer is correct, it is selected as the final output $\hat{y}_q$. If incorrect, the process continues to the next LLM until a correct answer is found or all models are exhausted.

3.5 Model Evaluation

Accuracy: Measuring the proportion of correct predictions relative to the total number of predictions made. It provides a clear and straightforward assessment of how frequently a model's outputs align with the true labels.

$$\text{Accuracy} = \frac{\sum_{i=1}^{N} I(\hat{y}_i = y_i)}{N} \tag{1}$$

In this equation, N denotes the total number of samples, y_i is the true label for the i-th sample, $\hat{y}_i$ is the predicted label, and $I(\text{condition})$ is the indicator function that returns 1 if the predicted label matches the true label and 0 otherwise. The summation $\sum_{i=1}^{N} I(\hat{y}_i = y_i)$ calculates the total number of correct predictions across all samples.

Pointwise Score: This metric assigns a positive value (+1) for each correct prediction and a negative value (–0.25) for each incorrect prediction, a system reminiscent of scoring methods used in medical examinations. The overall Pointwise Score is calculated as the average of these individual scores, as defined in Equation 2.

$$S = \frac{1}{N} \sum_{i=1}^{N} \left(I(\hat{y}_i = y_i) \cdot P_c + I(\hat{y}_i \neq y_i) \cdot P_w \right) \tag{2}$$

In this equation, S denotes the final score, N represents the total number of samples, y_i is the true label for the i-th sample, $\hat{y}_i$ is the predicted label, $I(\text{condition})$ is the indicator function that returns 1 if the specified condition is met and 0 otherwise, P_c is the reward for a correct prediction, and P_w is the penalty for an incorrect prediction.

Exception Rate: This metric evaluates the model's ability to adhere to the expected response format, specifically the JSON structure specified in the prompt. The prompt instructs the model to generate its output in JSON format, with the correct answer stored under the key `"correct_answer"`. Following generation, all outputs are preprocessed and parsed into JSON. If the output does not conform to the expected structure or the correct answer cannot be reliably extracted, it is classified as an exception. The exception rate is then computed as the proportion of these *exception* outputs relative to the total number of model responses, formally defined as:

$$\text{Exception}\% = \frac{N_{\text{exception}}}{N_{\text{total}}} \times 100 \tag{3}$$

In this equation, $N_{\text{exception}}$ denotes the number of outputs labeled as exceptions, and N_{total} represents the total number of generated outputs.

Average Inference Time per Row: This metric quantifies the average time required for a model to generate a response to a single input prompt,

measured in seconds. The timing begins when the prompt is submitted to the model and ends when the final answer is received. This measurement reflects the model's inference efficiency and is critical in evaluating its practical deployment in time-sensitive medical applications. The Average Inference Time per Row is calculated as follows:

$$\text{Average Inference Time (s)} = \frac{1}{N}\sum_{i=1}^{N} t_i \tag{4}$$

In this equation, N represents the total number of input prompts, and t_i is the time taken (in seconds) to generate a response for the i-th input. A lower average inference time indicates greater efficiency, which is desirable for real-time or high-throughput medical systems.

4 Experiment and Result

Our experiments were conducted on the ERAWAN supercomputer at Chiang Mai University (CMU). It features 384 AMD EPYC 7742 CPU cores operating at 2.25 GHz, complemented by 6.144 terabytes of RAM and 1.92 terabytes of GPU memory provided by NVIDIA HGX A100 accelerators.

To evaluate the effectiveness of different large language models (LLMs) in mitigating reasoning hallucinations within the medical domain, we conducted an extensive experiment utilizing the False Confidence Test (FCT) dataset from the Med-HALT benchmark. The experimental setup involved measuring the accuracy, pointwise score, exception rate, and average inference time per row of six individual LLMs, as well as two ensemble methods: Weighted Voting and Cascade Ensemble.

Model Name	**Accuracy**	**Score**	**Exception %**
MMed-Llama-3-8B	6.34	-32.22	2.58
meerkat-7b-v1.0	20.92	2.16	1.86
medalpaca-7b	0.22	-46.64	28.60
BioMistral-7B	2.67	-40.87	60.95
vicuna-13b-v1.5	11.00	-21.23	1.56
PMC-LLaMA-13B	0.34	-46.35	94.69
Weight Majority Voting	20.88	2.07	**0.23**
Cascade Ensemble	**30.23**	**24.12**	**0.23**

Table 2: Performance of Individual Models and Ensemble Methods

As shown in table 2, individual models exhibit varying levels of performance in terms of accuracy, pointwise score, and exception rate. Among all single models, Meerkat-7B-v1.0 demonstrates the strongest performance, achieving an accuracy of 20.92%, a pointwise score of 2.16, and a low exception rate of 1.86%.

In contrast, MedAlpaca-7B and PMC-LLaMA-13B perform the poorest, with extremely low accuracy rates (0.22% and 0.34%, respectively) and elevated exception percentages (28.60% and 94.69%, respectively), indicating both prediction unreliability and format inconsistency. Both ensemble methods substantially outperform individual models. The Weighted Voting approach improves accuracy to 20.88%, with a pointwise score of 2.06, and notably reduces the exception rate to 0.23%. This suggests that aggregating outputs from multiple models can enhance both accuracy and format adherence. Most impressively, the Cascade Ensemble method achieves the highest accuracy (30.23%) and pointwise score (24.12) while maintaining the lowest exception rate (0.23%).

Model	**Average inference time per row (second)**
MMed-Llama-3-8B	63.14
Meerkat-7b-v1.0	5.65
Medalpaca-7b	**2.75**
BioMistral-7B	10.91
Vicuna-13b-v1.5	5.01
PMC_LLaMA_13B	48.51
Weighted Voting	63.14
Cascade Ensemble	114.82

Table 3: Average Inference Time per Row

The average inference time per row varies considerably across individual models and ensemble methods. Among the single models, MedAlpaca-7B demonstrates the fastest inference time at 2.75 seconds, followed closely by Vicuna-13B-v1.5 and Meerkat-7B-v1.0, with times of 5.01 and 5.65 seconds, respectively. In contrast, MMed-Llama-3-8B and PMC-LLaMA-13B exhibit significantly longer inference times at 63.14 seconds and 48.51 seconds, respectively. When comparing ensemble methods, Weighted Voting requires the same inference time as its slowest component model (MMed-Llama-3-8B), resulting in an average of 63.14 seconds per row. Cascade Ensemble, which sequentially executes multiple models, incurs the highest computational cost, averaging 114.82 seconds per row.

Metrics		**Weighted Voting**	**Cascade Ensemble**
Accuracy	Max	20.94	30.23
	Average	20.88	30.23
	Min	20.83	30.23
Pointwise Score	Max	2.21	24.12
	Average	2.07	24.12
	Min	1.95	24.12

Table 4: Performance Metrics of Ensemble Models with Randomized Input Order

According to the results presented in Table 4, both ensemble methods—Weighted Voting and Cascade Ensemble—were evaluated in terms of accuracy and pointwise score across multiple runs with randomized input order. The Weighted Voting approach displays slight variability, with accuracy ranging from 20.83% to 20.94% and pointwise scores from 1.95 to 2.21, suggesting moderate consistency in prediction quality. In contrast, the Cascade Ensemble method yields uniform results across both metrics. This consistency reflects a stable performance regardless of input variation.

Model Sequencing Strategy	**Accuracy**	**Score**	**Average inference time per row (second)**
Random ordering	30.23	24.12	114.82
High-to-low performance ordering	**30.23**	**24.12**	**99.29**
High-to-low inference speed ordering	30.23	24.12	100.57
Low-to-high exception% ordering	30.23	24.12	99.70

Table 5: Cascade Ensemble Performance Across Different Model Sequencing Strategies

Despite applying various ordering methods—including random, performance-based, speed-based, and exception rate–based orderings—the accuracy and pointwise score remain unchanged across all strategies. This consistency suggests that the cascade mechanism is robust to the sequence in which models are applied, at least in terms of prediction quality. However, notable differences are observed in the average inference time per row. The random ordering strategy results in the highest inference time (114.82 seconds), while more structured strategies, such as high-to-low performance ordering, high-to-low inference speed ordering, and low-to-high exception% ordering, substantially reduce processing time to under 101 seconds. Among them, high-to-low performance ordering yields the most efficient inference time at 99.29 seconds.

5 Discussion

The superior performance of Meerkat-7B-v1.0, despite utilizing the same Mistral 7B architecture as BioMistral, can be attributed to differences in their training methodologies. Meerkat-7B-v1.0 was trained on a synthetic dataset comprising high-quality chain-of-thought (CoT) reasoning paths sourced from 18 medical textbooks, in addition to diverse instruction-following datasets. This training strategy enhances the model's capacity for multi-step reasoning, thereby reducing hallucinations when responding to MedQA questions. In contrast, BioMistral was trained primarily on the PMC Open Access Subset and PubMed Central corpus—collections of medical research documents that may lack the structured reasoning examples necessary to effectively mitigate hallucinations.

Among models derived from the LLaMA architecture, MMed-Llama-3-8B outperforms both MedAlpaca-7B and PMC-LLaMA-13B. This advantage is likely

due to MMed-Llama-3-8B's training on a comprehensive 25.5-billion-token dataset, as well as its foundation on LLaMA 3, which itself was pretrained on over 15 trillion tokens from publicly available sources. The breadth and diversity of this training corpus contribute to improved hallucination mitigation by exposing the model to a wider range of medical contexts and terminologies.

Despite its strong training foundation, PMC-LLaMA-13B—which has a larger parameter count—demonstrates lower accuracy and pointwise scores, largely due to a high rate of format exceptions. Although its rigorous training with data-centric knowledge injection and domain-specific instruction tuning likely enhanced its medical knowledge, it appears to have compromised its ability to generate responses that adhere to the specified prompt structure. This leads to frequent format errors and contributes to overall lower performance.

Similarly, BioMistral exhibits a high rate of format exceptions, often failing to produce an answer when uncertain. This behavior suggests that the model adopts a conservative response strategy, opting to remain silent rather than risk generating incorrect or hallucinated information. While this may reduce the risk of error, it also results in diminished overall utility when format adherence is critical.

In addition to performance variability, there are significant differences in inference time across models. MMed-Llama-3-8B and PMC-LLaMA-13B exhibit notably longer inference times compared to other models. Both models incorporate the same off-the-shelf English instruction fine-tuning dataset, which may contribute to increased computational complexity during generation. This added complexity, combined with their larger architectures and extensive fine-tuning, likely explains their latency relative to more lightweight models.

To address individual model limitations, two ensemble strategies—Cascade Ensemble and Weighted Voting—were evaluated. The Cascade Ensemble method achieved the highest accuracy and pointwise score among all approaches. Its sequential selection mechanism allows incorrect answers from one model to be passed to the next until a correct response is found or all models have been evaluated. This process enhances reliability by prioritizing the most accurate model while mitigating hallucination-induced errors. However, a notable limitation is its computational inefficiency; because models are evaluated sequentially rather than in parallel, inference time is significantly increased, making this approach less suitable for time-sensitive applications.

The Weighted Voting method, in contrast, balances efficiency and accuracy by aggregating predictions in parallel. This approach enables faster inference, as all model outputs are generated simultaneously and then combined. However, the initial equal weighting across all models allows weaker or less reliable models to influence the ensemble's decisions in the early stages. Over time, incorrect predictions are penalized and more reliable models gain greater influence, leading to improved performance. Nevertheless, Weighted Voting remains less effective at mitigating hallucinations compared to the Cascade Ensemble, as it does not eliminate erroneous contributions but merely reduces their weight.

Further comparison of these two ensemble strategies reveals key differences in robustness to input order. The order of input data influences the performance of Weighted Voting due to its row-wise weight adjustment mechanism. Since model weights are updated based on their correctness on each example, the sequence in which samples are presented can affect the distribution of influence across models. While this effect introduces only minor variability, the average performance of Weighted Voting remains slightly below that of the best-performing individual model. In contrast, the Cascade Ensemble method is unaffected by input order. Because it selects the correct answer from a fixed sequence of models without updating internal weights, its performance remains consistent regardless of how data are ordered—highlighting its robustness in sequential inference settings.

The inference efficiency of these ensemble strategies is further influenced by model ordering. The Cascade Ensemble method incurs the highest average inference time per row due to its inherently sequential execution. This inefficiency becomes more pronounced when earlier models in the sequence fail to provide correct answers, requiring subsequent evaluations. Among tested sequencing strategies, the high-to-low performance order yields the best inference efficiency. Prioritizing high-performing models increases the likelihood of terminating the sequence early with a correct prediction, thereby minimizing computational cost. Ordering based on high inference speed or low exception rate provides only marginal improvements and all structured ordering strategies substantially outperform random ordering, which results in the longest inference times. These findings underscore the importance of thoughtful model sequencing in ensemble design.

In contrast, the inference time of the Weighted Voting method is dictated by the slowest participating model, since all models must complete generation before aggregation can occur. While this constraint may limit real-time applicability, the ensemble calculation itself is computationally negligible—taking only 0.000013 seconds per row—meaning the majority of latency arises from model execution rather than the voting mechanism. As a result, Weighted Voting remains computationally efficient in terms of post-inference aggregation but may be bottlenecked by slower models in the ensemble.

6 Conclusion

This study examined the performance of individual large language models (LLMs) and ensemble methods in mitigating reasoning hallucinations in the medical domain using the False Confidence Test (FCT) dataset. Results revealed substantial variation in model performance, influenced primarily by differences in training data and methodology. Models trained on structured, synthetic datasets with chain-of-thought reasoning—such as Meerkat-7B-v1.0—exhibited superior reasoning capabilities, while others trained on unstructured biomedical corpora showed limitations in both accuracy and format adherence. Among ensemble approaches, the Cascade Ensemble consistently outperformed all individual models and the Weighted Voting method in both accuracy and reliability, benefiting

from its sequential answer selection strategy. However, this improvement came at the cost of increased inference time due to its non-parallel design. In contrast, Weighted Voting offered faster inference through parallel processing but was more susceptible to the influence of underperforming models, especially in early stages. Additionally, model and input order significantly affected inference efficiency and ensemble robustness. These findings emphasize the critical role of training design, model sequencing, and ensemble architecture in enhancing the robustness and reliability of medical LLM systems.

This study has several limitations. First, the evaluation was conducted solely on the False Confidence Test (FCT), which represents only one dimension of reasoning hallucinations. Other types of hallucination assessments, such as those targeting logical coherence or factual consistency, were not explored. Future research is needed to assess the effectiveness of ensemble methods across a broader range of hallucination categories. Additionally, the evaluation metrics in this study focused exclusively on the correctness of the final answer, without assessing the underlying reasoning process. While expert validation of reasoning pathways would improve the reliability of the evaluation, such an approach is resource-intensive and poses significant barriers to large-scale implementation in real-world settings. Furthermore, the dataset used in this study follows a question-answering format, which may not fully represent the complexity of real-world clinical scenarios. In practice, medical decision-making requires comprehensive contextual information—such as patient history, clinical notes, and lab results—that extends beyond the scope of a single question. The absence of this context in current benchmark datasets may limit the model's ability to generate accurate and clinically meaningful responses. Providing models with richer, patient-specific context could significantly improve inference quality and better align AI outputs with the needs of clinical environments.

From an implementation perspective, applying these ensemble methods in real-world healthcare systems presents significant challenges. The Cascade Ensemble, while effective in benchmark evaluations, relies on access to ground-truth answers to determine whether a model's prediction is correct—a condition that does not exist in clinical settings, where the correctness of an inference cannot be validated in real-time. As such, this method is only suitable for scenarios involving exam-style questions with predefined answers. On the other hand, the Weighted Voting approach, though parallel in nature, demands substantial computational resources to run multiple large models simultaneously—resources that are often unavailable in typical hospital environments. These limitations highlight a critical gap between current research benchmarks and practical clinical deployment. Further development is required to adapt ensemble techniques for real-time use in healthcare workflows, ensuring they operate effectively under resource constraints and without dependence on ground-truth feedback.

References

1. Barile, J., Margolis, A., Cason, G., Kim, R., Kalash, S., Tchaconas, A., Milanaik, R.: Diagnostic accuracy of a large language

model in pediatric case studies. JAMA Pediatrics **178**(3), 313–315 (03 2024). https://doi.org/10.1001/jamapediatrics.2023.5750, https://doi.org/10.1001/jamapediatrics.2023.5750
2. Bruno, A., Mazzeo, P.L., Chetouani, A., Tliba, M., Kerkouri, M.A.: Insights into classifying and mitigating llms' hallucinations. CEUR Workshop Proceedings **3563**, 50–63 (11 2023), https://arxiv.org/abs/2311.08117v1
3. Charlotte R Blease, Cosima Locher, J.G.M.H.K.D.M.: Generative artificial intelligence in primary care: an online survey of uk general practitioners. BMJ Health amp; Care Informatics **31**(1), e101102 (Aug 2024). https://doi.org/10.1136/bmjhci-2024-101102, https://doi.org/10.1136/bmjhci-2024-101102
4. Fang, C., Li, X., Fan, Z., Xu, J., Nag, K., Korpeoglu, E., Kumar, S., Achan, K.: Llm-ensemble: Optimal large language model ensemble method for e-commerce product attribute value extraction. Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval (SIGIR '24), July 14â•fi18, 2024, Washington, DC, USA **1** (2 2024). https://doi.org/10.1145/3626772.3661357, https://arxiv.org/abs/2403.00863v2
5. Goodman, R.S., Patrinely, J.R., Stone, Cosby A., J., Zimmerman, E., Donald, R.R., Chang, S.S., Berkowitz, S.T., Finn, A.P., Jahangir, E., Scoville, E.A., Reese, T.S., Friedman, D.L., Bastarache, J.A., van der Heijden, Y.F., Wright, J.J., Ye, F., Carter, N., Alexander, M.R., Choe, J.H., Chastain, C.A., Zic, J.A., Horst, S.N., Turker, I., Agarwal, R., Osmundson, E., Idrees, K., Kiernan, C.M., Padmanabhan, C., Bailey, C.E., Schlegel, C.E., Chambless, L.B., Gibson, M.K., Osterman, T.J., Wheless, L.E., Johnson, D.B.: Accuracy and reliability of chatbot responses to physician questions. JAMA Network Open **6**(10), e2336483–e2336483 (10 2023). https://doi.org/10.1001/jamanetworkopen.2023.36483, https://doi.org/10.1001/jamanetworkopen.2023.36483
6. Han, T., Adams, L.C., Papaioannou, J.M., Grundmann, P., Oberhauser, T., Löser, A., Truhn, D., Bressem, K.K.: Medalpaca – an open-source collection of medical conversational ai models and training data (2023), https://arxiv.org/abs/2304.08247
7. Hong, S., Xiao, L., Zhang, X., Chen, J.: Argmed-agents: Explainable clinical decision reasoning with llm disscusion via argumentation schemes (3 2024), https://arxiv.org/abs/2403.06294v2
8. Huang, L., Yu, W., Ma, W., Zhong, W., Feng, Z., Wang, H., Chen, Q., Peng, W., Feng, X., Qin, B., Liu, T.: A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions (11 2023), https://arxiv.org/abs/2311.05232v1
9. Kim, H., Hwang, H., Lee, J., Park, S., Kim, D., Lee, T., Yoon, C., Sohn, J., Choi, D., Kang, J.: Small language models learn enhanced reasoning skills from medical textbooks (2024), https://arxiv.org/abs/2404.00376
10. Labrak, Y., Bazoge, A., Morin, E., Gourraud, P.A., Rouvier, M., Dufour, R.: Biomistral: A collection of open-source pretrained large language models for medical domains (2024), https://arxiv.org/abs/2402.10373
11. Lewis, P., Perez, E., Piktus, A., Petroni, F., Karpukhin, V., Goyal, N., Küttler, H., Lewis, M., Yih, W.T., Rocktäschel, T., Riedel, S., Kiela, D.: Retrieval-augmented generation for knowledge-intensive nlp tasks. Advances in Neural Information Processing Systems **2020-December** (5 2020), https://arxiv.org/abs/2005.11401v4
12. Li, K., Patel, O., Viégas, F., Pfister, H., Wattenberg, M.: Inference-time intervention: Eliciting truthful answers from a language model. Advances in Neural Information Processing Systems **36** (6 2023), https://arxiv.org/abs/2306.03341v6

13. Lim, Z.W., Pushpanathan, K., Yew, S.M.E., Lai, Y., Sun, C.H., Lam, J.S.H., Chen, D.Z., Goh, J.H.L., Tan, M.C.J., Sheng, B., Cheng, C.Y., Koh, V.T.C., Tham, Y.C.: Benchmarking large language models' performances for myopia care: a comparative analysis of chatgpt-3.5, chatgpt-4.0, and google bard. EBioMedicine **95** (9 2023). https://doi.org/10.1016/J.EBIOM.2023.104770, https://pubmed.ncbi.nlm.nih.gov/37625267/
14. Maguire, B.: Aging with ai: How artificial intelligence is changing senior care (October 2024), https://www.carewell.com/resources/blog/aging-with-ai/, accessed: 2025-03-07
15. Manakul, P., Liusie, A., Gales, M.J.: Selfcheckgpt: Zero-resource black-box hallucination detection for generative large language models. EMNLP 2023 - 2023 Conference on Empirical Methods in Natural Language Processing, Proceedings pp. 9004–9017 (3 2023). https://doi.org/10.18653/v1/2023.emnlp-main.557, https://arxiv.org/abs/2303.08896v3
16. Mbakwe, A.B., Lourentzou, I., Celi, L.A., Mechanic, O.J., Dagan, A.: Chatgpt passing usmle shines a spotlight on the flaws of medical education. PLOS Digital Health **2**, e0000205 (2 2023). https://doi.org/10.1371/JOURNAL.PDIG.0000205, https://journals.plos.org/digitalhealth/article?id=10.1371/journal.pdig.0000205
17. Mohammed, A., Kora, R.: A comprehensive review on ensemble deep learning: Opportunities and challenges. Journal of King Saud University - Computer and Information Sciences **35**, 757–774 (2 2023). https://doi.org/10.1016/J.JKSUCI.2023.01.014
18. Pal, A., Umapathi, L.K., Sankarasubbu, M.: Med-halt: Medical domain hallucination test for large language models. CoNLL 2023 - 27th Conference on Computational Natural Language Learning, Proceedings pp. 314–334 (7 2023). https://doi.org/10.18653/v1/2023.conll-1.21, https://arxiv.org/abs/2307.15343v2
19. Qiu, P., Wu, C., Zhang, X., Lin, W., Wang, H., Zhang, Y., Wang, Y., Xie, W.: Towards building multilingual language model for medicine (2024), https://arxiv.org/abs/2402.13963
20. Salvagno, M., Taccone, F.S., Gerli, A.G.: Artificial intelligence hallucinations. Critical Care **27**, 1–2 (12 2023). https://doi.org/10.1186/S13054-023-04473-Y/FIGURES/1, https://ccforum.biomedcentral.com/articles/10.1186/s13054-023-04473-y http://creativecommons.org/publicdomain/zero/1.0/
21. Soroush, A., Glicksberg, B.S., Zimlichman, E., Barash, Y., Freeman, R., Charney, A.W., Nadkarni, G.N., Klang, E.: Large language models are poor medical coders — benchmarking of medical code querying. NEJM AI **1**(5), AIdbp2300040 (2024). https://doi.org/10.1056/AIdbp2300040, https://ai.nejm.org/doi/full/10.1056/AIdbp2300040
22. Stiennon, N., Ouyang, L., Wu, J., Ziegler, D.M., Lowe, R., Voss, C., Radford, A., Amodei, D., Christiano, P.: Learning to summarize from human feedback. Advances in Neural Information Processing Systems **2020-December** (9 2020), https://arxiv.org/abs/2009.01325v3
23. Tonmoy, S.M.T.I., Zaman, S.M.M., Jain, V., Rani, A., Rawte, V., Chadha, A., Das, A.: A comprehensive survey of hallucination mitigation techniques in large language models (1 2024), https://arxiv.org/abs/2401.01313v3
24. Wu, C., Lin, W., Zhang, X., Zhang, Y., Wang, Y., Xie, W.: Pmc-llama: Towards building open-source language models for medicine (2023), https://arxiv.org/abs/2304.14454
25. Zheng, L., Chiang, W.L., Sheng, Y., Zhuang, S., Wu, Z., Zhuang, Y., Lin, Z., Li, Z., Li, D., Xing, E.P., Zhang, H., Gonzalez, J.E., Stoica, I.: Judging llm-as-a-judge with mt-bench and chatbot arena (2023), https://arxiv.org/abs/2306.05685