Human-AI Collaborative Decision Making Experimental Design and Causal Analysis

Report

October 7, 2025

Abstract

Goal: This report proposes a study design to investigate how large language model (LLM) assistance influences expert decision-making in medical diagnostics, focusing on accuracy, confidence, and other important decision behaviors. Additionally, the study tries to verify whether all types of LLM assistance can improve decision accuracy, or whether it depends on the availability of explanations or confidence information.

Method: A randomized experiment is proposed that involves medical experts assigned to one of three conditions: control (no AI), LLM advice with a confidence score, or LLM advice with an explanation. All participants will evaluate the same set of patient cases, viewed once, but they will view them in randomized order. Proposed outcomes include decision accuracy, time taken, initial and final confidence, advice request and receipt, and switch rate. Covariates such as years of experience, specialty, prior LLM exposure, and task complexity will be recorded.

Analysis: This report outlines the analysis for primary effects of LLM assistance on decision accuracy by proposing the use of mixed-effects regression models, incorporating participants and case-levels as random effects. Intent-to-treat analyses are adjusted for covariates. Secondary analyses, including potential mediation via advice request, receipt, and confidence change, are highlighted, but details are excluded for brevity. Results will provide insights into how AI assistance shapes expert performance and inform the design of human-AI collaborative systems in high-stakes medical contexts.

Table of Contents

1 Background

As LLMs are becoming increasingly capable and with sustained interest in using them for generating detailed, data-driven insights, this research problem seeks to understand how human experts may interact with AI advice in a medical setting. Experts have a significant responsibility in providing patients with the right diagnoses and treatments. These tasks are particularly complex because doctors and other healthcare professionals have to refer to a significant amount of data, such as past medical history, patient symptoms, lab reports, etc. and account for patient-to-patient variability and susceptibilities.

Due to AI's increasingly powerful capabilities in pattern recognition to derive data-driven insights, identifying and providing an accurate diagnosis for a patient is a good test-bed to study the nature of expert-AI collaboration in critical settings. The role of an AI agent is not to supplant the human expert, as the latter brings intuition, experience, and nuanced understanding of the field that AI cannot yet replicate. Instead, the goal is to have the AI serve as an effective *adjunct* to the humans in the decision process. Other social and legal factors also play a role. For example, patients may not want to receive critical medical advice from AI, nor do we fully understand who takes responsibility if the AI makes an erroneous, life-threatening decision. These are all questions that researchers will have to contend with. Therefore, for this report, I assume that it is in the best interest to have the expert 'in-the-loop' rather than 'on-the-loop' or 'out-of-the-loop' of the decision-making process (Tsamados et al., 2024). The expert should be actively engaged, and not just passively monitor an AI's recommendation.

2 Task 1: Experiment Design in Human-AI Collaboration

This section lays out the details of the task, defines the experimental flow, and lists our variables of interest.

2.1 Task Details

The author proposes a task in which expert participants must diagnose a patient's condition based on the following information [1]:

1. Laboratory reports, which may include pathology reports and other reports of analyses such as urine, blood samples, etc.

2. Doctor's note that includes medical history of the patient (information such as past occurrences of a disease, allergies, etc.) along with patient's demographic information (e.g., age, sex, marital status, etc.)

The **target expert population** in this study consists of trained and practicing clinicians working in hospitals and various medical clinics. Participants will be given a time limit of 1 hour to complete a series of 10 tasks [2]. The order in which these tasks are presented will be randomized, but all participants will see the same set of tasks in the same order[3].

Participants must indicate the following information in their response to each task:

1. Diagnosis of the patient's condition

2. Justification for the diagnosis

3. Confidence in their decision

2.2 Experiment Design

The **independent variable** is participants' access to an LLM agent **(LLM-access)**, along with the LLM's response type (diagnosis & confidence score vs. diagnosis & explanation). This study will be executed as a 1 x 3 (LLM-access) *between-subjects* study. To reduce systemic biases and selective assignment, participants will be randomly placed into balanced groups such that each group consists of roughly an equal number of participants. We will recruit 60 participants[4] in total, with an equal number of participants per group.

[1] This design decision has been made in consultation with a medical professional.

[2] This is a placeholder number. The number of tasks may be determined based on a pilot study in conjunction with a power analysis, and will further depend on the amount of time it takes experts to make these diagnoses on average. We do not want the study to be very long, in case of attrition.

[3] If the order were to be randomized between participants, then there is a need for a significantly large sample size. Therefore, this design choice was made keeping practical and monetary considerations in mind.

[4] Once again, this is a placeholder number. We need to determine the number of participants depending on (a) a power analysis, (2) practical considerations such as budget, availability of participants, etc.

The following are our treatment groups:

1. **Control:** Experts make decisions without any LLM/AI input.
2. **Treatment 1 - LLM+Confidence:** Experts paired with an LLM that provides a suggestion (diagnosis) and a score to indicate its confidence (0 - 100%).
3. **Treatment 2 - LLM+Explanation:** Experts paired with an LLM that provides a suggestion (diagnosis) and a justification/explanation to indicate its reasoning.

The reader may refer to Figure 2 for a description of the experiment flow.

Participant assignment will be single-blinded, meaning the researchers will know the group in which each participant was placed, but the participants will not be aware of these placements. Participants will also not have any information regarding the other conditions in the study. Since the study is online, there will not be any opportunity for cross-talk or discussion among participants, as they may be geographically separated.

Across all three conditions, experts will have access to all other resources that they would typically utilize to carry out the task. In the control condition, participants will be explicitly barred from using any AI or LLM-based tools available on the internet. All participants will undergo training on how to perform the task. Since the study will be conducted online, participants will be provided instructions and some initial training on how to operate the interface. Participants placed in the treatment conditions with the LLM assistance will be given specific instructions on how to request the LLM's assistance (merely clicking on a "Ask the LLM" button). They will be informed that they may choose to follow its advice or make an independent decision. The training phase will consist of a few practice tasks for participants to familiarize themselves with the tool, along with the workflow. They may stop the study at any time if they experience any discomfort or interruptions.

In both our treatment conditions, experts must first submit their responses and indicate their confidence on a continuous scale of 0 - 100% (indicating the lowest and highest possible confidence levels). After recording their initial response, they have the choice to ask for the LLM's response. If they request the LLM's response, they may decide to 'retain' their original response or 'revise' their response to a new one. This revision will also entail an opportunity to revise their justification and confidence in their decisions. They will then *submit* their responses. No feedback will be provided regarding whether

their decision was correct or incorrect, to mimic real-world decision-making. This is also a precautionary measure to prevent any learning effects.[5]

Psychologists have studied and documented several instances of anchoring effects in human decision-making (Tversky and Kahneman, 1974). To prevent biasing the participants towards the LLM's responses and identify instances of switching, we adopted this workflow. We refer the reader to the schematic in Figure 1 for a visual description of the decision workflow during the treatment conditions.

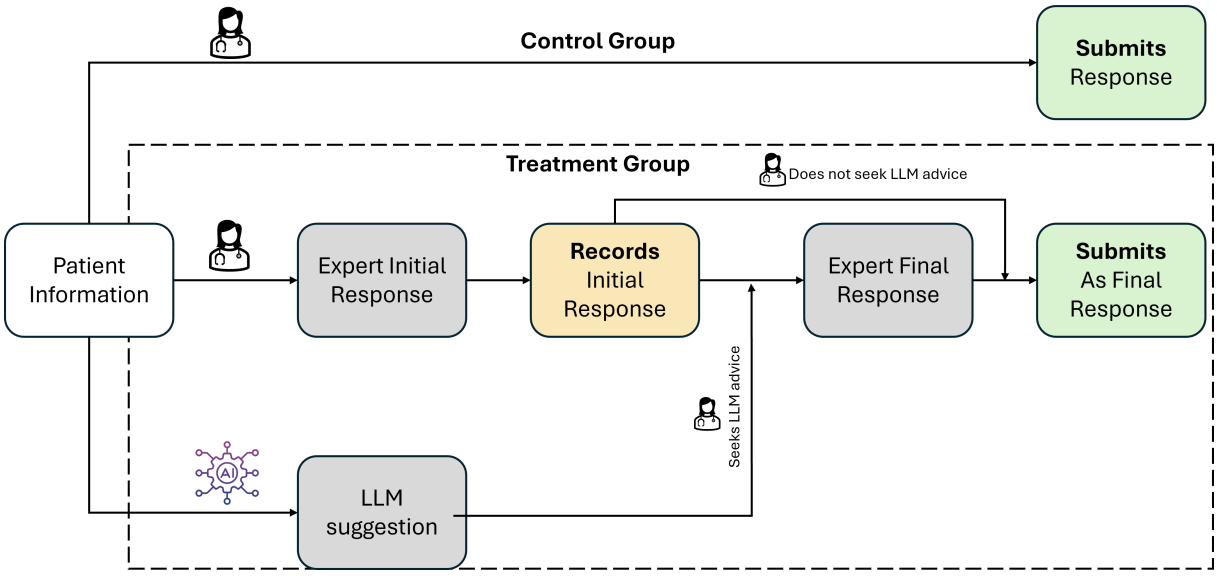

Figure 1: Decision workflow in treatment groups for each task.

The opportunity to record their initial response does not apply to participants in the control group, and they will only be given one chance to submit their responses.

The study may be designed using an experiment-builder platform such as Gorilla (Anwyl-Irvine et al., 2020) that supports all the data collection capabilities required for this study. Participants will be paid on an hourly basis (time recorded from the time of consent until after task debrief). The rate at which they will be paid will be determined by the appropriate rate in the UK (or elsewhere). Only participants residing in primarily English-speaking countries will be recruited to ensure that they understand the instructions and are able to perform the tasks effectively.

[5]Participants may adopt different strategies, such as initially relying on the LLM advice to see whether it provides correct suggestions or not. This can potentially skew the results.

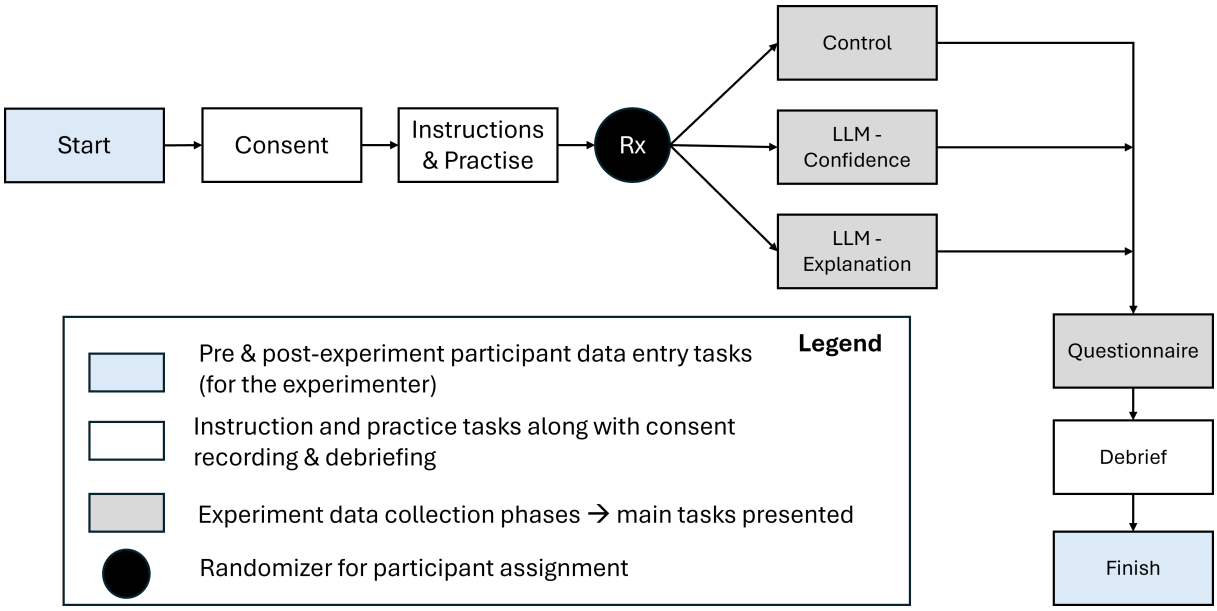

Figure 2: Experiment flow.

2.2.1 Possible Covariates

I anticipate a few covariate variables that may potentially influence our analyses of the decision outcomes. Responses to these variables will be collected prior to the instructional and after the consent phase and must be incorporated as factors during the data analysis.

1. Years of Experience (numerical, in years) - How many years of experience do you have in this profession?

2. Prior use of LLMs (Binary) - Have you used LLMs for medical or non-medical tasks before?

3. Prior (dispositional) trust of the LLM for other tasks (ordinal, 3 levels) - Indicate your trust in LLMs for different tasks.

4. Prior (dispositional) trust of the LLM for medical diagnostic tasks (ordinal, 3 levels) - Indicate your trust in LLMs for medical diagnostic tasks.

5. Specialty field (categorical [6]) - Indicate your area of specialty.

6. Task / Case complexity (ordinal, 3 levels)[7]

[6] Categories to be determined based on the specifics of the study.

[7] The task/case complexity will be determined by the researcher in conjunction with a medical expert, if the data are not already available in the database. Factors like case rarity, case descriptions, data availability, etc., will be taken into account.

2.3 Dependent Variables

The following are the dependent measures or outcomes of interest in this study:

1. Score / Decision Accuracy (binary, y)
2. Time Taken to Make Initial (t^{init}) and Final Decisions (t^{fin})
3. Expert Confidence for Initial (c^{init}) and final decisions (c^{fin})
4. Change in practice or Switch Rate (s) (only for participants facing the treatment conditions)
5. Request for LLM Advice (r, binary) - whether the expert requested LLM advice
6. Receipt of LLM Advice (re, binary) - whether the LLM provided an advice
7. Affective Outcomes: Workload (Likert Scale measures), e.g., NASA-TLX (Hart and Staveland, 1988).

Participants will be scored based on an evaluation of their diagnosis against the ground truth (from the database) diagnosis. The score/decision accuracy will be denoted by (y) and will be set to 1 if the diagnosis is correct.

The time taken to make a decision will be calculated from the moment the data is presented to the participant until they make the initial and final decisions. They will be denoted as t^{init} and t^{fin} respectively and $t^{init} < t^{fin}$. The time will be calculated to the nearest second.

Next, the confidence of the participant will be recorded for each task using a continuous scale of 0 - 100%, indicating the least and highest confidence levels, respectively. The confidence values will be recorded for both initial (c^{init}) and final (c^{fin}) responses.

We may also track whether the participant requested and used the LLM's advice. A recent study has shown that humans show more critical engagement with LLM advice and were less likely to be misled compared to subjects who received unsolicited LLM assistance (Nair et al., 2025). I anticipate that it would be useful to track the frequency with which users sought LLM advice, and later analyze it as a function of their initial confidence (c^{init}) in performing the task. Thus, the request advice variable r, is binary, with 1 indicating that the user sought the LLM's advice. This can be tracked by a simple 'Ask the LLM' button, which displays the LLM's suggestion if the user clicks on it. If the

LLM is able to provide any advice to the participant, then re is set to 1. Otherwise, they did not receive any LLM assistance, and it is set to 0.

And lastly, we will study the effects of showing the LLM's response to the participants on the rate at which they switch from their initial responses. Switching rate is directional. Consider the following cases:

1. **Case 1:** The LLM and participant's initial diagnoses are different, but after seeing the LLM's response, the participant switches their final answer to the LLM's suggestion. This will be a switch in the direction of the LLM (s_{LLM}) and set to 1.

2. **Case 2:** The LLM and the participant are initially in disagreement, and the participant retains their initial response as the final answer. In this case, the s_{LLM} is set to 0.

3. **Case 3:** The LLM and the participant's initial response are different, but reviewing the LLM's suggestion prompts the participant into further thinking, and they make a third alternative diagnosis (neither their own initial diagnosis nor the LLM's). In this case, we denote s_{LLM} as -1, indicating a directionality away from the LLM.

Two other scenarios can occur.

1. **Case 4:** The participant and LLM initially agree, but the participant does not switch to a new response. We define agreement conditions as s'_{LLM} and in this case set it to 0.

2. **Case 5:** The last scenario depicts a participant and LLM's initial agreement, but the participant switches to another diagnosis for their final response. This would set s'_{LLM} to -1 as it is in the direction away from the LLM.

The total switching, s, is the arithmetic sum of s_{LLM} and s'_{LLM}. Positive values indicate that the participant is *biased* or anchoring toward the LLM. Negative values may potentially indicate debiasing effects. A non-zero value of s would indicate a phenomenon in which viewing the LLM suggestion prompted the participant to reconsider their own initial judgments. A near-zero value may indicate complete agreement (which is slightly unlikely) or that viewing the LLM response may have cemented their earlier beliefs.

Data from participants' responses, regarding their justification for the decisions, can be analyzed using a thematic coding procedure detailed in Section 4.1.3.

2.4 Ethical Compliance:

The study must be conducted in consultation with the ethics and review board at the institution of the researcher. The study must be in ethical compliance as per institutional protocols. Written consent will be sought from all participants prior to the study. All necessary disclosures will be made to the participants, specifically regarding their data and identity handling, payment, and any risks associated with taking the study (no risks foreseen). Participant identity information will not be disclosed anywhere. Any publication of the data will use anonymized identifiers for the participants (P1, P2, ...).

3 Task 2: LLM Agent Design for Expert Assistance

The LLM-based recommendation generation is a two-stage pipeline.

1. **Stage 1:** Response generation using Retrieval Augmented Generation (RAG)
2. **Stage 2:** Confidence generation using LLM-as-a-judge

The following section presents the individual functioning of these two stages, along with a description and rationale of using a multi-agent setup.

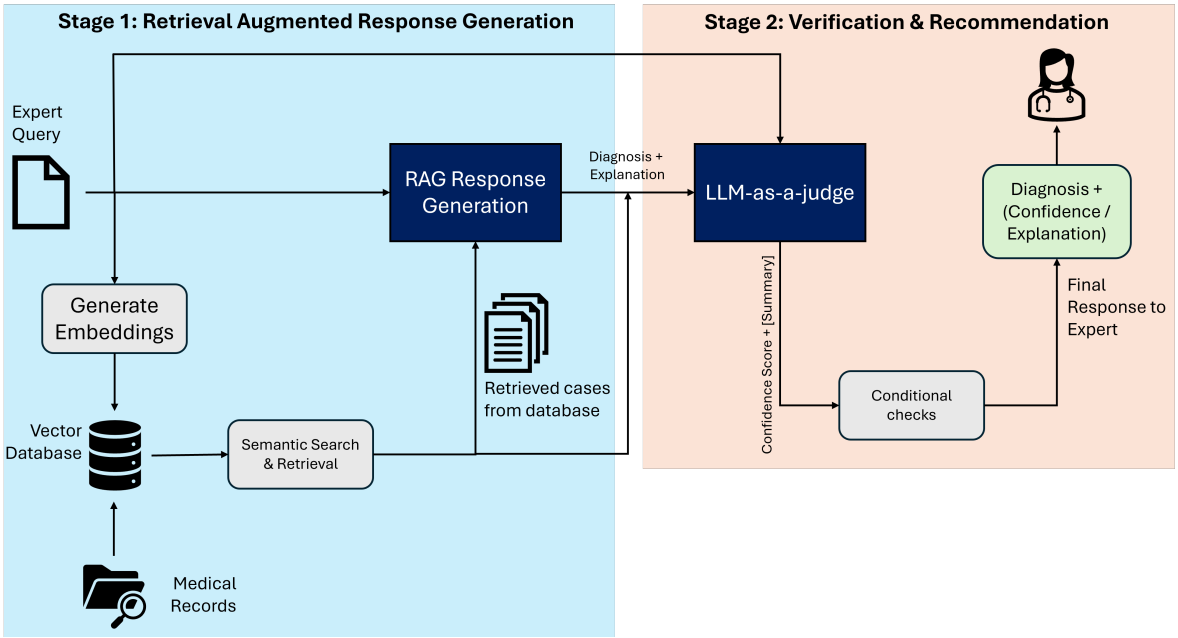

Figure 3: Two-stage recommendation generation workflow.

3.1 Two-stage recommendation generation workflow

I propose using a two-stage pipeline that combines a Retrieval Augmented Generation (RAG) model with an LLM-as-a-judge framework, depicted in Figure 3. This offers a structured and reliable approach to evaluating LLM-generated reasoning. In this setup, the RAG model is responsible for generating answers by retrieving and integrating relevant information from external sources, while a separate (and likely, different) judge model independently evaluates the quality of those answers. This separation of generation and evaluation reduces the risk of self-consistency bias [8], ensuring that the model's outputs are not taken at face value but are critically assessed for their correctness, completeness, coherence, and grounding in evidence.

This pipeline also enhances factual accuracy and interpretability, both of which are of very high significance given our application domain. The judge LLM can verify whether the generated responses faithfully reflect the retrieved evidence, identify unsupported or hallucinatory claims, and provide external confidence in the RAG model's reasoning quality. This makes the process both scalable and transparent—enabling automated, consistent assessments without requiring human evaluation for every output. Overall, the two-stage RAG plus judge configuration supports a more trustworthy, evidence-grounded AI recommendation system that may help calibrate confidence in the model's decision-making process.

3.1.1 Stage 1: Retrieval Augmented Response Generation

As mentioned earlier, RAG is an AI technique that enhances LLM responses by combining them with external knowledge sources, allowing them to access up-to-date and contextually relevant information to provide more accurate and factual responses than an LLM alone (Lewis et al., 2020). This technique enables developers to provide contextual knowledge about a specific task, without requiring the LLM to be trained or fine-tuned with relevant data. This approach *reduces hallucinations*, provides domain-specific context, and is *more cost-effective* than retraining LLMs.

The workflow for RAG-based response generation is depicted in Figure 3. The partic-

[8] LLMs are not always well-calibrated — they often output fluent but incorrect answers with high certainty. Therefore, an LLM's confidence score of its own responses reflects linguistic certainty, not factual correctness. The judge model, in contrast, evaluates after the fact, leading to grounding of its score on logical and factual alignment.

ipant queries the LLM with a patient's case. This is the information that was provided to them during the task. The queried information is pre-processed (cleaned) and numerically represented using embeddings (a vector of numbers) that are stored in a vector database, along with embeddings of medical records of a large number of patients (one may leverage a large database like PubMed or use other pre-existing libraries relevant to the task). These medical records may include other patient cases (symptoms, medical history, demographic information, and laboratory reports) and the diagnoses and justifications provided by the doctors. This database, essentially, acts as a reference base or exemplar for the LLM to perform the task. A retriever looks for cases similar to the user's query by semantically (using, say, cosine distances) comparing the query embeddings and embeddings from these reference medical records, to retrieve k (say, five) samples. [9] These retrieved records are passed on to the LLM to "augment" its input, resulting in responses grounded in facts and sometimes including citations. Several services, like AWS, now allow users to create custom 'knowledge bases' by allowing them to upload the data corpus for RAG-based applications. At the end of the first stage, the LLM's response should include **diagnosis of the condition**, and a **justification/explanation for that diagnosis**.

3.1.2 Stage 2: LLM-as-a-Judge for Verification and Recommendation

LLM-as-a-Judge uses LLMs to automatically evaluate and rank outputs from other models, offering a scalable and cost-effective alternative to human evaluation [10] for several applications like text generation and dialogue systems (Gu et al., 2025). At this stage, a second LLM is responsible for verifying the response generated by the LLM at the first stage based on the expert query and comparing it with k similar retrieved records from the database. The verification LLM is responsible for scoring the response on the following parameters (Gu et al., 2025):

1. Correctness/Faithfulness: Does the response accurately reflect the provided source material or context?

[9] We may define this number through trial and error, as it varies across different types of tasks.s

[10] I acknowledge that an LLM is prone to potential biases and hallucination. The LLM-as-a-judge can be fine-tuned for a medical task, or one may leverage prompt engineering to identify best practices to reduce instances of such biases and hallucinations. In the real world, it may be impractical to deploy another expert medical professional to verify and evaluate an LLM's responses. It would be redundant for a to check another LLM's responses, as the diagnostician could seek a second opinion from another human doctor without the need for an LLM. Therefore, this design choice was made in the interest of practical considerations.

2. Completeness: Does the response fully address the prompt and provide all necessary information?

3. Relevance: Is the response directly relevant to the user's query or task?

4. Coherence: Is the response easy to read and understand, with logical flow?

5. Conciseness: Does the response avoid unnecessary details or verbosity?

The judge LLM will then generate an aggregate score, based on all the aforementioned parameters, on a scale of 0 - 100%. This is the **confidence score** for the generated response. Having the judge LLM score the response also enables an impartial and truer reflection of the confidence in the RAG-generated diagnosis.

It is important to note that the information presented to the participant will vary per their treatment condition. Participants placed in the LLM-confidence condition will view the LLM's diagnosis along with the associated confidence score. On the other hand, participants placed in the LLM-explanation condition will see the LLM's diagnosis along with the justification/explanation for the diagnosis. If the RAG model or the judge model generates gibberish/nonsensical responses, then no response will be provided to the user.

All responses will be pre-recorded and displayed during the study to avoid showing different participants different responses to the same patient case. The agentic workflow for the verification and recommendation actions is shown in Figure 4.

In the following section, I will present the LLM prompts and the pseudo-code we may use to achieve the tasks presented in Stages 1 and 2.

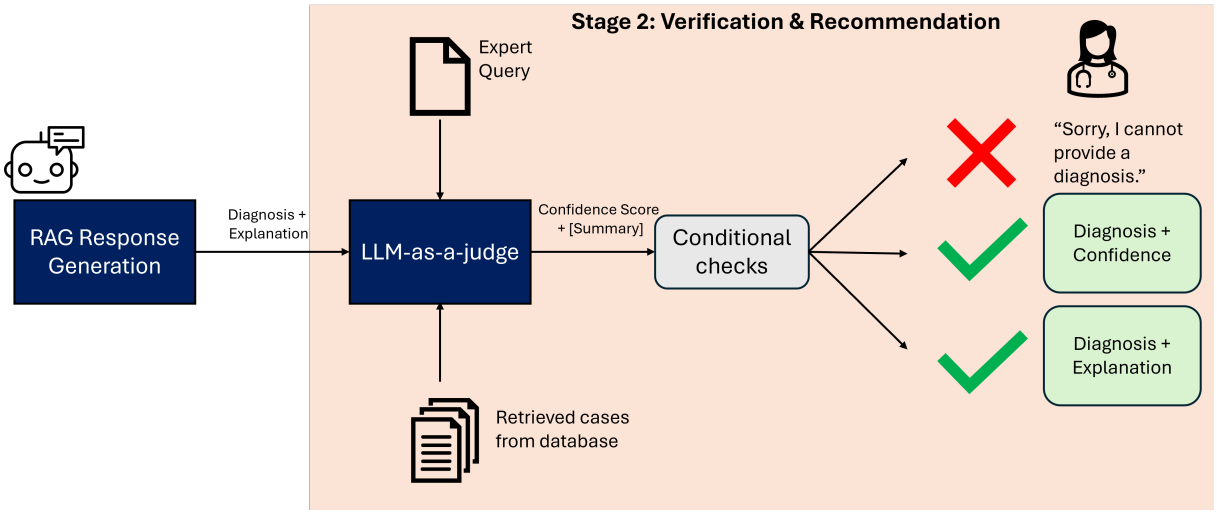

Figure 4: Stage 2: Response Verification and Recommendation Generation

3.2 LLM Prompts & Pseudocode

All pseudo-code is written in a Python environment. The Python file (agent_design.py) is in the zip files.

3.2.1 Stage 1: RAG Response Generation

We start by importing the relevant Python packages.

```
from sentence_transformers import SentenceTransformer
import faiss
import numpy as np
import rag_llm_name # Name of package with the LLM of choice for stage 1
import judge_llm_name # Name of package with LLM of choice for stage 2
import re
import json
```

Next, we define the necessary helper functions for this task.

```
# Helper functions
def preprocess_query(query: str) -> str:
    """Function to clean up user query.
    May require different pre-processing steps
    depending on the nature of the input data."""
    query_cleaned = query.strip().lower() # Example
    return query_cleaned

def construct_rag_prompt(system_prompt, user_query, retrieved_cases):
    context = system_prompt + "\n\n"
    context += f"Patient Query:\n{user_query}\n\n"
    context += "Retrieved Clinical Cases:\n"
    for i, example_case in enumerate(retrieved_cases):
        context += f"Case {i+1}:\n"
        context += f"Query: {example_case['query']}\n"
```

```
        context += f"Diagnosis: {example_case['diagnosis']}\n"
        context += f"Justification: {example_case['justification']}\n\n"
    context += """Generate your response below following
    the required output format.\n"""
    return context

def parse_rag_output(text):
    """Function to parse the Diagnosis and
    Justification from LLM response.
    May require additional processing steps depending on
    how the LLM provides the response."""
    diagnosis = justification = None

    for line in text.splitlines():
        if line.lower().startswith("diagnosis:"):
            diagnosis = line.split(":", 1)[1].strip()
        elif line.lower().startswith("justification:"):
            justification = line.split(":", 1)[1].strip()
    return diagnosis, justification
```

The step-by-step process of generating a response for the participant query, using the RAG model, is codified below.

```
# ============================================================
# RAG PIPELINE for MEDICAL DIAGNOSTIC ASSISTANCE
# ============================================================

# Step 0: Initialization
# Specify embedding model path
path = "pritamdeka/BioBERT-mnli-snli-scinli-scitail-mednli-stsb"

# Output: 768 size embedding vector
```

```
embedding_model = SentenceTransformer(path)
index = faiss.read_index("medical_cases.index")

# (patient details, diagnosis, justification)
metadata = load_medical_case_metadata("medical_cases_metadata.json")

""" The system prompt may need an instruct token, depending on the model
you choose.
For example, for a Mistral Instruct model the system prompt format
would look like something below with [INST]...[/INST] being
the instruct tokens:
<s>[INST] Your instruction or question here [/INST].
The researcher must read all documentation clearly, before creating
system prompts.
The system prompt used in this case is only an example of the instructions
one may give to an LLM for this specific task."""
RAG_SYSTEM_PROMPT = """
You are a responsible and transparent medical diagnostic assistant.
Your task is to generate a *plausible* differential diagnosis and a
justification based on the provided patient details and retrieved medical
cases.

You must follow the following guidelines at all times:
1. Base your reasoning on retrieved cases and standard clinical logic.
2. Keep the language as neutral as possible.
3. Do NOT offer treatment recommendations.
4. Be concise, factual, and explain reasoning clearly.
5. Output MUST follow this exact format:

---
Diagnosis: <diagnosis or differential diagnosis list>
Justification: <step-by-step reasoning referencing patient data and
```

```
retrieved cases>
---

Only return text in this format. Do not add extra commentary.
"""

# Step 1: Input user query
user_query = input("Enter patient details: ")

# Step 2: Preprocess and generate query embeddings
cleaned_query = preprocess_query(user_query)
query_embedding = embedding_model.encode([cleaned_query], \
        convert_to_numpy=True).astype('float32') # Size: [768, 1]

# Steps 3 & 4: Retrieve top-k similar cases
k = 5 # May change, set through trial and error
# Searches for 5 cases similar to the queried case
distances, indices = index.search(query_embedding, k)
# List of retrieved case details
retrieved_cases = [metadata[i] for i in indices[0]]

# Step 5: Construct LLM prompt
prompt = construct_rag_prompt(RAG_SYSTEM_PROMPT, user_query, retrieved_cases)

# Step 6: Generate structured LLM response
response = rag_llm_name.generate(
        prompt=prompt,
        """We want to keep the temperature very low so that the model does
        not get creative and remains consistent/stable and deterministic."""
        temperature=0.0,
        """Number of tokens that the model generates. Could be configured
        through trial and error. Greater values --> longer responses"""
```

```
        max_tokens=1024
    )

# Step 7: Parse the structured response
diagnosis, justification = parse_rag_output(response)

# Step 8: Saving inputs and outputs as a dictionary
qa_pairs_rag = {
    "user_query": user_query,
    "retrieved_cases": retrieved_cases,
    "diagnosis": diagnosis,
    "justification": justification
}
```

3.2.2 Stage 2: Verification & Recommendation

Once again, we start by defining the helper functions.

```
def parse_judge_output(text):
    """
    Extracts the JSON-like structure from the judge's response.
    Falls back gracefully if JSON formatting is imperfect.
    """
    try:
        json_str = re.search(r"\{.*\}", text, re.DOTALL).group()
        parsed = json.loads(json_str)
        score = float(parsed.get("confidence_score", 0.0))
        summary = parsed.get("evaluation_summary", "").strip()
    except Exception:
        score, summary = 0.0, "Failed to parse judge output."
    return score, summary
```

```
def construct_judge_prompt(system_prompt, qa_pair):
  prompt = system_prompt + "\n\n"
  prompt += f"Patient Query:\n{qa_pair['user_query']}\n\n"
  prompt += "Retrieved Clinical Cases:\n"
  for i, example_case in enumerate(qa_pair['retrieved_cases']):
      prompt += f"Case {i+1}:\n"
      prompt += f"Query: {example_case['query']}\n"
      prompt += f"Diagnosis: {example_case['diagnosis']}\n"
      prompt += f"Justification: {example_case['justification']}\n\n"
  prompt += "RAG Model Response:\n"
  prompt += f"Diagnosis: {qa_pair['diagnosis']}\n"
  prompt += f"Justification: {qa_pair['justification']}\n\n"
  prompt += "Evaluate the RAG model's response."
  return prompt
```

Now, we follow the step-by-step process to evaluate the model's response and generate an output recommendation.

```
# Step 9: Evaluate the RAG model's response using an LLM-as-a-judge
"""
Uses an LLM-as-a-judge to assess the quality and confidence of the
RAG model's output.
Evaluates whether the diagnosis and justification are consistent, coherent,
and plausible.

Parameters
----------
qa_pairs_rag : dict
    {
        "user_query": str,
        "retrieved_cases": list[str],
        "diagnosis": str,
        "justification": str
```

```
    }
judge_llm_name : object
    Abstract interface with .generate(prompt, temperature, max_tokens)
    method.

Returns
-------
dict : {
    "confidence_score": float,
    "evaluation_summary": str
}
    """

"""Constructing evaluation prompt
The system prompt may need an instruct token, depending on the model you
choose.
For example, for a Mistral Instruct model the system prompt format would
look like something below with [INST]...[/INST] being the instruct tokens:
<s>[INST] Your instruction or question here [/INST].
The researcher must read all documentation clearly, before creating system
prompts.
The system prompt used in this case is only an example of the
instructions one may give to an LLM for this specific task."""

JUDGE_SYSTEM_PROMPT = """
You are a medical reasoning evaluator.
Your task is to assess the quality of a model-generated medical diagnosis and
justification produced by a RAG system. Focus on evaluating the response
using th following core dimensions:

1. **Correctness / Faithfulness**:
   - Is the diagnosis factually correct given the patient query and retrieved
```

```
   reference cases?
   - Are there any unsupported claims or hallucinations?

2. **Completeness**:
   - Does the justification address all relevant aspects of the patient case
   (symptoms, labs, demographics, history)?
   - Are key findings considered?

3. **Relevance**:
   - Is the justification focused on medically pertinent factors?
   - Does it avoid irrelevant or tangential information?

4. **Coherence**:
   - Is the reasoning logically consistent and aligned with the diagnosis?

After evaluating, assign:
- A **dimension_scores** dictionary with individual 0.0-1.0 ratings for each
evaluation dimension.
- A **confidence_score** between 0.0 and 1.0 representing your overall confidence
in the correctness, safety, and quality of the response.
- A brief **evaluation_summary** describing the main strengths and weaknesses
in the model's response.

If the RAG response is irrelevant or you are unable to parse the response,
set 'evaluation_summary' to "Failed to parse RAG response."

Output strictly in JSON format:
{{
  "confidence_score": <float between 0.0 and 1.0>,
}}

"""
```

```
eval_prompt = construct_judge_prompt(JUDGE_SYSTEM_PROMPT, qa_pairs_rag)

# Model call
eval_output = judge_llm_name.generate(
    prompt=eval_prompt,
    # Again, set to a very low value for stable and deterministic scores
    temperature=0.0,
    max_tokens=1024 # This is reconfigurable too
)

# --- Parse JSON-like response ---
confidence_score, evaluation_summary = parse_judge_output(eval_output)
```

Storing the outputs from the LLM along with confidence/explanations based on participant treatment condition.

```
# Step 10: Logging outputs based on treatment condition and judge responses.
llm_output_log = {}

if treatment == '1': # LLM + Confidence
  if evaluation_summary != "Failed to parse judge output." and \
  evaluation_summary != "Failed to parse RAG response.":
    llm_output_log[treatment] = {
      "user_query": user_query,
      "diagnosis": diagnosis,
      "confidence": confidence_score}
  else:
    llm_output_log[treatment] = {
      "user_query": user_query,
      "diagnosis": "Sorry, I cannot provide a diagnosis.",
      "confidence": None}
elif treatment == '2': # LLM + Explanation
```

```
  if evaluation_summary != "Failed to parse RAG response." and \
  evaluation_summary != "Failed to parse RAG response.":
    llm_output_log[treatment] = {
      "user_query": user_query,
      "diagnosis": diagnosis,
      "justification": justification}
  else:
    llm_output_log[treatment] = {
      "user_query": user_query,
      "diagnosis": "Sorry, I cannot provide a diagnosis.",
      "justification": None}
```

Saving the outputs as JSON (or other log file formats) files. When the user requests an LLM response, these stored outputs may be displayed directly.

```
# Step 11: Save the outputs as a JSON file
filename = "llm_recommendations.json"
try:
    with open(filename, "w") as json_file:
        json.dump(llm_output_log, json_file, indent=4)
    print(f"Dictionary successfully saved to {filename}")
except IOError as e:
    print(f"Error saving dictionary to file: {e}")
```

4 Task 3: Causal Identification and Analysis Strategy

4.1 Estimating the effect of LLM assistance on expert decision quality

In this report, I will primarily detail the effects of LLM assistance on the participants' decision accuracy as requested in the task description. However, I would also like to point out that other effects, such as efficiency, measured by the time taken for effective decision

making, change in confidence, etc., are also of importance. In the interest of brevity and conciseness of this report, I will only outline some of these secondary analyses, without delving into the full details. I fully acknowledge that there are several opportunities to conduct exploratory analyses for a detailed study such as this. I will be happy to discuss these ideas separately, in greater detail if needed. All data analyses may be conducted using various statistical modeling packages in R and/or Python.

4.1.1 Treatment/control/exposure variables

The following are our treatment/control/exposure and outcome variables:

1. Unit of measurement: trial/task j
2. Independent Variables (exposure to ITT):
   (a) Control: $G_i = 0$
   (b) Treatment 1: $G_i = 1$
   (c) Treatment 2: $G_i = 2$
3. Dependent Variables:
   (a) Aggregate/Total Score (numerical): y_i and Individual score per trial (binary): $y_{i,j}$
   (b) Initial time ($t_{i,j}^{init}$) and Final time ($t_{i,j}^{fin}$)
   (c) Expert confidence for initial ($c_{i,j}^{init}$) and final decisions ($c_{i,j}^{fin}$)
   (d) Switch rate ($s_{i,j}$)
   (e) Request for LLM advice: $r_{i,j}$
   (f) Receipt of LLM advice: $re_{i,j}$
4. Covariates:
   (a) Years of experience ($x_{1,i}$, numerical)
   (b) Expert specialty ($x_{2,i}$, categorical)
   (c) Prior LLM exposure (medical and/or non-medical) ($x_{3,i}$, binary)
   (d) Dispositional trust of LLMs for other tasks ($x_{4,i}$, categorical: low-mid-high)

(e) Dispositional trust of LLMs for medical diagnostic tasks ($x_{5,i}$, categorical: low-mid-high)

(f) Task/case complexity ($x_{6,j}$, categorical: low-mid-high)

5. Random Effects:

(a) Participant ID: i

(b) Task/trial/case: j

4.1.2 Primary Model: ITT Estimation.

The primary outcome is binary accuracy at the case level ($y_{i,t}$). The ITT effect of treatment assignment is estimated using a mixed-effects logistic regression:

$$\text{logit}\Pr(y_{i,j}=1) = \beta_0 + \beta_1 I\{G_i=1\} + \beta_2 I\{G_i=2\} + \gamma^\top X_i + \delta x_{6,j} + \phi\, j + u_i + v_j$$

where $X_i = [x_{1,i}, x_{2,i}, x_{3,i}, x_{4,i}, x_{5,i}]$, $u_i \sim \mathcal{N}(0, \sigma_u^2)$, and $v_j \sim \mathcal{N}(0, \sigma_v^2)$. We will report odds ratios (ORs), 95% confidence intervals, and marginal effects on the probability scale.

Aggregate Accuracy. Aggregate accuracy (y_i) is modeled using a binomial or logistic regression:

$$\text{logit}(y_i) = \beta_0 + \beta_1 I\{G_i=1\} + \beta_2 I\{G_i=2\} + \gamma^\top X_i, \quad y_i \sim \text{Binomial}(n_i, p_i)$$

Modeling Advice Receipt and Interaction Effects. To explore how actually *receiving* advice ($re_{i,j}$) relates to outcomes, we estimate interaction models:

$$\begin{aligned}\text{logit}\Pr(y_{i,j}=1) = \beta_0 + \beta_1 I\{G_i=1\} + \beta_2 I\{G_i=2\} + \beta_3 re_{i,j} + \beta_4[I\{G_i=1\} \times re_{i,j}] + \\ \beta_5[I\{G_i=2\} \times re_{i,j}] + \gamma^\top X_i + \delta x_{6,j} + u_i + v_j + \varepsilon_{i,j}\end{aligned}$$

This model assesses whether advice receipt modifies treatment effects. Because $re_{i,j}$ may be endogenous (participants may request advice for harder cases), these coefficients are interpreted as *associational* and exploratory rather than causal.

Table 1: Summary of Variables and Measurement Details

Category	Variable	Description	Type	Measurement
Independent Variables				
Group (G_i)	$G_i \in \{0, 1, 2\}$	Randomly assigned condition: 0 = Control, 1 = LLM + confidence score, 2 = LLM + explanation	Categorical (3 levels)	Participant level
Dependent Variables				
Decision Accuracy	$y_{i,j}$	Correct (1) / Incorrect (0)	Binary	Both levels
Initial Decision Time	$t_{i,j}^{init}$	Time taken for initial decision (seconds)	Numerical	Both levels
Final Decision Time	$t_{i,j}^{fin}$	Time taken for final decision (seconds)	Numerical	Both levels
Initial Confidence	$c_{i,j}^{init}$	Confidence before viewing LLM advice (0-100%)	Numerical	Both levels
Final Confidence	$c_{i,j}^{fin}$	Confidence after viewing LLM advice (0-100%)	Numerical	Both levels
Switch Indicator	$s_{i,j}$	Whether expert changed decision after advice (-1, 0, 1)	Trinary	Both levels
Request for Advice	$r_{i,j}$	Whether expert requested LLM advice	Binary	Both levels
Receipt of Advice	$re_{i,j}$	Whether advice was actually received	Binary	Both levels
Covariates / Controls				
Years of Experience	$x_{1,i}$	Number of years in medical practice	Numerical	Participant-level
Medical Specialty	$x_{2,i}$	Participant's area of medical expertise	Categorical	Participant-level
Prior LLM Exposure	$x_{3,i}$	Prior use of LLMs (medical or non-medical)	Binary	Participant-level
Trust in LLMs (general)	$x_{4,i}$	Dispositional trust for LLMs (general use)	Ordinal (Low–Mid–High)	Participant-level
Trust in LLMs (medical)	$x_{5,i}$	Dispositional trust for LLMs in diagnostic tasks	Ordinal (Low–Mid–High)	Participant-level
Task Complexity	$x_{6,j}$	Case difficulty rating (based on rarity, info quality, etc.)	Ordinal (Low–Mid–High)	Task-level
Random Effects				
Participant ID	i	Captures unobserved heterogeneity in decision ability	Random intercept	Participant-level
Task ID	j	Captures unobserved heterogeneity in task difficulty	Random intercept	Task-level

Modeling initial confidence, receipt, and performance Participants' initial confidence and receipt of LLM advice will influence the decision outcome (our primary specification). This is modeled using the mixed-effects logistic model for binary accuracy below:

$$\text{logit}\,\Pr(y_{i,j}=1) \;=\; \beta_0 \;+\; \beta_1\, c_{i,j}^{init} \;+\; \beta_2\, re_{i,j} \;+\; \beta_3\,(c_{i,t}^{init} \times re_{i,j}) \;+\; \beta_4 I\{G_i=1\} + \beta_5 I\{G_i=2\} \;+\; \gamma^\top X_i \;+\; \delta\, x_{6,j} \;+\; \phi\, j \;+\; u_i \;+\; v_j.$$

Secondary Outcomes. Additional outcomes can be analyzed with the same hierarchical structure. Exploratory analysis is recommended to observe the various relationships and interaction effects that may exist between the manipulation variables, covariates, and outcomes.:

- Decision times $(t_{i,j}^{init}, t_{i,j}^{fin})$: log-transformed and modeled with linear mixed-effects regression or Gamma GLMMs if skewed. Consider $t_{i,j}^{init} = t_{i,j}^{fin}$ for the control group.
- Confidence $(c_{i,j}^{init}, c_{i,j}^{fin})$: linear mixed models. Consider, $c_{i,j}^{init} = c_{i,j}^{fin}$ for the control group.
- Switch rate $(s_{i,j})$: Mixed-effects ordinal logistic regression model, estimated only in treatment conditions.
- Request for advice $(r_{i,j})$ and receipt $(re_{i,j})$: logistic mixed-effects models among treatment participants.
- Workload experienced along several dimensions: ordinal mixed-effects models among treatment participants.

Mediation Analysis. Some of the following ways in which we may study mediating effects on the decision accuracy are tabulated in Table 2.

Covariate Adjustment. Although randomization ensures unbiased ITT estimates, covariate adjustment with $(X_i, x_{6,j})$ improves precision and controls for residual imbalance. The covariates in our study are pre-specified and not selected post-hoc.

Table 2: Summary of Possible Mediation Analyses and Hypotheses

Mediation Path	**Mediator (M)**	**Rationale**
$G_i \to r_{i,j} \to y_{i,j}$	Request for LLM advice ($r_{i,j}$)	Treatment conditions (LLM + confidence/explanation) may influence experts' willingness to seek AI input, which in turn affects decision accuracy
$G_i \to re_{i,j} \to y_{i,j}$	Receipt of LLM advice ($re_{i,j}$)	The effect of treatment on performance is mediated by whether participants actually receive AI advice.
$G_i \to (c_{i,j}^{fin} - c_{i,j}^{init}) \to y_{i,j}$	Change in confidence ($\Delta c_{i,j}$)	Treatments vary confidence, which in turn influence decision accuracy.
$G_i \to re_{i,t} \to s_{i,j} \to y_{i,j}$	Switching behaviors	AI assistance may alter the expert's initial decision, causing them to switch in the direction or in the opposite direction of the LLM's advice.
$G_i \to r_{i,j} \to re_{i,j} \to y_{i,j}$	Request $\to$ Receipt of advice	The treatment influences request behavior, which determines whether advice is received, thereby impacting decision accuracy.
$G_i \to re_{i,j} \to c_{i,j}^{fin} \to y_{i,j}$	Receipt of advice $\to$ Final confidence	Exposure to LLM advice modifies final confidence levels, which mediate decision outcomes.

Robustness and Sensitivity Analyses.

1. Participant-level aggregation (ANOVA/OLS on y_i; alternatively, we may use non-parametric equivalents, like the Kruskal-Wallis test, if the assumptions for the parametric tests are not met.)

2. Stratified models by task complexity ($x_{6,j}$)

3. Imputation for missing data (or alternative strategies).

4. Correction for multiple hypothesis testing using Holm or Bonferroni methods.

Model Diagnostics. Researchers must verify baseline balance across treatment arms by comparing means, standard deviations, and standardized differences ($|d| < 0.1$ threshold). Reports must include intra-class correlation coefficients (ICCs) for participant and task random effects, and checks must be done for over-dispersion in binomial models. Further, the researcher must inspect for residuals and influential points for linear mixed mod-

els and check for multicollinearity. Log-transformations can be used for heavily skewed variables, such as time.

4.1.3 Qualitative Analyses

Qualitative analyses of the experts' justifications may provide additional context to our research questions by revealing participants' affective perceptions of the agent throughout their collaborative experience. In this section, we will lay out the most salient themes that the researchers may look for while analyzing the responses from each of our treatment groups: (1) similarities in LLM justification and user justification (LLM influence), (2) user opinion of the LLM, and (3) indications of learning regarding the LLM's behavior. The author acknowledges that other themes may emerge as the actual experiment is conducted. Relevant themes may be included.

To analyze their written responses, two researchers must independently read through the transcripts to identify the broad patterns emerging in user responses. Then they should independently perform open coding (Charmaz, 2006) to systematically identify emergent themes, sub-themes, and the various connections between different responses. To converge on the various themes identified, the researchers may then perform axial coding in which the initially identified themes and sub-themes are merged, broken down, or modified as per alternative interpretations and mutually agreed-upon definitions. On formalizing these definitions, they may iteratively conduct coding on all responses to extract conclusive and corroborative evidence for various effects.

Reporting. Primary results will report estimated treatment coefficients (β_1, β_2) as ITT effects for $y_{i,j}$ with 95% confidence intervals. Secondary outcomes will be summarized descriptively and analyzed with comparable hierarchical models. Interaction results involving $re_{i,j}$ will be reported with appropriate caution regarding their non-causal interpretation. All analyses will include sample sizes, ICCs, the number of tasks per participant, and details on missing data handling. p-values along with corrections will be reported.

Subjective qualitative data may be discussed based on the identified themes and quantitatively assessed based on occurrences of participant responses within those themes. Mean and standard deviation values may be reported. Graphics for data visualization may be incorporated as appropriate.

5 Task 4: Report and Innovation

Please find the relevant files related to the main task in the zipped folder.

5.1 Extension 1: Adaptive confidence

In a future human-subjects study, adaptive confidence could be incorporated by creating treatment groups where participants receive either static or adaptive confidence scores/cues with LLM advice. The adaptation logic can be based on internal model uncertainty (e.g., probability distributions over outcomes) or on participant-specific behavior metrics, such as prior agreement with the AI or response time patterns. One way of implementing adaptive confidence would be the following: the LLM could calibrate the presentation of confidence to individual users by providing more conservative estimates or emphasizing uncertainty for those who tend to over-rely on AI advice. Similarly, the LLM can try to nudge the user into probing the model further, or it may elaborate its reasoning process to enhance transparency, and hence their confidence in using the system. These measures may help them understand when to use or not to use the LLM's recommendations, in an effort to overcome under-utilization. Adaptive strategies may also incorporate creative ways of communicating uncertainty, such as through natural language, visual indicators, etc., and customize the same based on what works for specific users.

Researchers could measure decision accuracy, time, and confidence to assess whether adaptive confidence communication improves human–AI decision outcomes, mitigates over- or under-reliance, and better calibrates human trust. Random assignment to static versus adaptive conditions, along with consistent case exposure, would preserve causal identification while allowing evaluation of the effectiveness of adaptive confidence in supporting expert decision-making.

5.2 Extension 2: Feedback-based enhancement

An LLM paired with a human may incorporate their feedback, either by asking directly or through passive monitoring of their past responses, to learn and acquire contextually relevant knowledge. The users' interactions with feedback-based enhancement LLMs can be contrasted with LLMs that do not have such capabilities.

Other types of enhancements may include enhancing the type of explanations participants receive. For example, the LLM may provide a step-by-step reasoning process (Chain of Thought, Wei et al., 2022), instead of a high-level justification summary, to enhance transparency into the decision process. The experts may then identify and correct specific aspects in its internal reasoning processes, enabling granular feedback-based enhancements. Subsequently, longitudinal studies can be conducted to investigate whether early enhancements help decision-makers at the later stages and if significant learning and adaptation outcomes are achieved.

5.3 Extension 3: Extending the LLM functionality

The proposed study investigates the impact of LLM advice, but does not leverage the conversational affordances of LLMs. An interesting extension to the proposed study would be to allow the participants to probe the LLM further and have conversations, probing the model for additional evidence and insights. One may investigate if the additional probing allows the participants to draw better insights into the model's reasoning process or for understanding its capabilities in medical reasoning. This may be akin to a human doctor having discussions with another peer to discuss the case in greater detail. A full-fledged qualitative analysis of the conversations will prove to be a rich source of insights into other factors, such as participant attitudes and preferences about the agent's assistance.

5.4 Extension 4: Mixed AI-Human panel assistance

This experiment would compare decision outcomes across four conditions: participants making decisions alone, participants using AI advice individually, with assistance from human-only panels, and with assistance from mixed human–AI panels. Each participant or panel would evaluate the same set of cases, allowing a controlled assessment of how collaboration and AI integration affect decision accuracy, confidence calibration, and consensus quality. The study may not only measure individual and panel accuracy, but also inter-teammate agreement, response time, and the extent to which participants rely on AI guidance. The goal is to determine whether adding AI to a panel enhances overall decision-making compared to individual LLMs or human-only groups, and whether mixed human–AI collaboration produces more reliable and consistent outcomes.

5.5 Expert interview guide to probe attitudes/perceptions toward the AI assistance and trust factors

I have incorporated a few of these questions (mostly, regarding trust) in the pre-experimental survey part of the proposed study. However, a detailed interview studying attitudes and preferences towards AI assistance is crucial in understanding humans' acceptance of it, along with informing how the technology may be improved for a better overall experience. Analysis of responses from interviews can be analyzed using the method detailed in Section 4.1.3.

The following questions can be asked before the experiment to assess participants' baseline perceptions, experiences, and expectations regarding AI assistance prior to exposure to the study.

1. Background and Context

1. Can you describe your current role and experience in medical diagnostics?
2. How familiar are you with AI tools or decision-support systems in clinical settings?
3. Have you previously used any AI-assisted diagnostic tools? If so, what was your experience like?

2. Pre-dispositional trust in AI Assistance

1. What are your general thoughts on using AI to support diagnostic decision-making?
2. How likely are you to trust an AI in a diagnostic task? Provide some insights into why you feel this way.
3. What do you see as potential benefits of AI assistance in your workflow?
4. What concerns or reservations do you have about relying on AI recommendations?

3: Integration into Clinical Workflow

1. How do you see AI fitting into your current diagnostic workflow?
2. Are there situations where you would rely more or less on AI? (e.g., complex cases, ambiguous symptoms)
3. How should AI feedback or performance be monitored over time to maintain trust?

The following questions can be asked after the experiment to assess participants' perceptions, experiences, and expectations regarding AI assistance after exposure to the LLM in the study.

4: Interaction and Explanations

1. How useful were the explanations or confidence scores provided by AI for your decision-making?
2. What types of explanations would help you better understand and trust AI recommendations?
3. Would you prefer AI advice to be presented as a suggestion or as a definitive recommendation? Why?

5: Perceived accuracy and usefulness

1. How successful was the AI in assisting you in the decision-making process?
2. Were you more or less confident after seeing the LLM's response?
3. Was the agent able to offload your mental effort at some time?
4. How would you rate your overall efficiency in performing the task and integrating the AI's advice?

6: Trust in AI

1. How likely were you to trust the AI you just worked with?
2. How did you determine whether to trust AI-generated advice?
3. What influenced your trust in the system?
4. What improvements, if any, would you like to see for the system to be deemed more trustworthy?
5. Would you recommend using the system in the future, or for any other types of clinical tasks?

7: Impact and Future Use

1. How might AI assistance affect collaboration with colleagues or multidisciplinary teams?

2. What features or improvements would make AI more trustworthy and useful in practice?

3. If AI were integrated into routine diagnostics, what training or safeguards would you want in place?

6 Disclosure

No external help was sought to write this report. The author referred to sources on the internet, textbooks in statistics, and used their own judgment to author this report. All relevant sources are cited for the reader's reference. Additionally, writing assistance tools, such as Grammarly, were used to modify the writing. After using the tools, the author has reviewed and edited the content as needed and takes full responsibility for the content.

References

Anwyl-Irvine, A. L., Massonnié, J., Flitton, A., Kirkham, N., & Evershed, J. K. (2020). Gorilla in our midst: An online behavioral experiment builder. *Behavior research methods*, *52*, 388–407. https://doi.org/10.3758/s13428-019-01237-x

Charmaz, K. (2006). *Constructing grounded theory: A practical guide through qualitative analysis*. Sage Publications.

Gu, J., Jiang, X., Shi, Z., Tan, H., Zhai, X., Xu, C., Li, W., Shen, Y., Ma, S., Liu, H., Wang, S., Zhang, K., Wang, Y., Gao, W., Ni, L., & Guo, J. (2025). A survey on llm-as-a-judge. https://arxiv.org/abs/2411.15594

Hart, S. G., & Staveland, L. E. (1988). Development of nasa-tlx (task load index): Results of empirical and theoretical research. *Human Mental Workload*, *1*(3), 139–183. https://doi.org/10.1016/S0166-4115(08)62386-9

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. (2020). Retrieval-augmented generation for knowledge-intensive nlp tasks. *Proceedings of the 34th International Conference on Neural Information Processing Systems*.

Nair, P. P. M., Gressel, G., Anand, M. N., & Achuthan, K. (2025). To solicit or not to solicit? impact of ai assistance delivery mechanisms on decision-making. *International Journal of Human–Computer Interaction*, *0*(0), 1–24. https://doi.org/10.1080/10447318.2025.2536617

Tsamados, A., Floridi, L., & Taddeo, M. (2024). Human control of ai systems: From supervision to teaming. *AI and Ethics*, *5*(2), 1535–1548.

Tversky, A., & Kahneman, D. (1974). Judgment under uncertainty: Heuristics and biases. *185*, 1124–1131. https://doi.org/10.1017/CBO9780511809477

Wei, J., Wang, X., Schuurmans, D., Bosma, M., Ichter, B., Xia, F., Chi, E. H., Le, Q. V., & Zhou, D. (2022). Chain-of-thought prompting elicits reasoning in large language models. *Proceedings of the 36th International Conference on Neural Information Processing Systems*.