Special Review | Artificial Intelligence

eISSN 2005-8330
https://doi.org/10.3348/kjr.2025.1522
Korean J Radiol 2025;26(12):1123-1132

Korean Journal of Radiology
KJR

Minimum Reporting Items for Clear Evaluation of Accuracy Reports of Large Language Models in Healthcare (MI-CLEAR-LLM): 2025 Updates

Seong Ho Park[1], Chong Hyun Suh[1], Jeong Hyun Lee[2], Ali S. Tejani[3], Seng Chan You[4], Charles E. Kahn, Jr[5], Linda Moy[6]

[1]Department of Radiology and Research Institute of Radiology, University of Ulsan College of Medicine, Asan Medical Center, Seoul, Republic of Korea
[2]Department of Radiology and Center for Imaging Science, Samsung Medical Center, Sungkyunkwan University School of Medicine, Seoul, Republic of Korea
[3]Department of Radiology and Biomedical Imaging, University of California, San Francisco, CA, USA
[4]Department of Biomedical Systems Informatics, Yonsei University College of Medicine, Seoul, Republic of Korea
[5]Department of Radiology and Institute for Biomedical Informatics, University of Pennsylvania, Philadelphia, PA, USA
[6]Department of Radiology, New York University Grossman School of Medicine, New York, NY, USA

Recent systematic reviews have raised concerns about the quality of reporting in studies evaluating the accuracy of large language models (LLMs) in medical applications. Incomplete and inconsistent reporting hampers the ability of reviewers and readers to assess study methodology, interpret results, and evaluate reproducibility. To address this issue, the MInimum reporting items for CLear Evaluation of Accuracy Reports of Large Language Models in healthcare (MI-CLEAR-LLM) checklist was developed. This article presents an extensively updated version. While the original version focused on proprietary LLMs accessed via web-based chatbot interfaces, the updated checklist incorporates considerations relevant to application programming interfaces and self-managed models, typically based on open-source LLMs. As before, the revised MI-CLEAR-LLM focuses on reporting practices specific to LLM accuracy evaluations: specifically, the reporting of how LLMs are specified, accessed, adapted, and applied in testing, with special attention to methodological factors that influence outputs. The checklist includes essential items across categories such as model identification, access mode, input data type, adaptation strategy, prompt optimization, prompt execution, stochasticity management, and test data independence. This article also presents reporting examples from the literature. Adoption of the updated MI-CLEAR-LLM can help ensure transparency in reporting and enable more accurate and meaningful evaluation of studies.

Keywords: Large language model; Large multimodal model; Generative; Artificial intelligence; Chatbot; Application programming interface; Local deployment; Reporting; Guideline; Checklist; Healthcare; Medicine; Radiology

INTRODUCTION

Since the launch of ChatGPT, large language models (LLMs) have generated widespread interest for their potential use across diverse healthcare tasks [1,2]. Naturally, a growing number of studies have evaluated the accuracy of LLMs in medical applications. However, recent systematic reviews have raised concerns about the quality of reporting in these studies, including those published in top-tier journals [3-5]. Inconsistent and incomplete reporting hampers the ability of the reviewers and readers to evaluate the methodology and results of the studies, as well as to assess the reproducibility of the findings.

To address this issue, the MInimum reporting items for CLear Evaluation of Accuracy Reports of Large Language Models in healthcare (MI-CLEAR-LLM) checklist was developed to provide a minimal set of essential items for

Received: October 13, 2025 **Accepted:** October 16, 2025
Corresponding author: Seong Ho Park, MD, PhD, Department of Radiology and Research Institute of Radiology, University of Ulsan College of Medicine, Asan Medical Center, 88 Olympic-ro 43-gil, Songpa-gu, Seoul 05505, Republic of Korea
• E-mail: parksh.radiology@gmail.com

transparent reporting of clinical studies evaluating LLM accuracy in healthcare applications [6]. The original version of MI-CLEAR-LLM primarily targeted studies using proprietary LLMs accessed via web-based chatbot interfaces (e.g., ChatGPT). Since its publication, however, an increasing number of studies have adopted application programming interfaces (APIs) [7] and self-managed LLMs (typically based on open-source models such as LLaMA or DeepSeek), prompting updates to reflect these developments. As with the original version, the updated MI-CLEAR-LLM focuses on key reporting considerations specific to LLM accuracy studies: specifically, the reporting of how LLMs are specified, accessed, adapted, and applied in testing, with particular attention to methodological elements that influence model outputs. To assist researchers, this article also provides examples from the published literature illustrating how these items have been reported.

A few broader reporting guidelines have recently emerged to support studies involving LLMs in healthcare [8-10]. These frameworks cover the full structure of research manuscripts, from title to conclusions, and also tend to include general items applicable to all types of clinical artificial intelligence research. In contrast, as noted above, MI-CLEAR-LLM complements such frameworks by more narrowly and practically addressing critical elements related to the selection and use of LLMs in research studies, which are areas under-addressed in wider-scope guidelines.

MINIMUM REPORTING ITEMS

The minimum items for transparent reporting are outlined in Table 1, which comprises eight item categories, each containing multiple specific elements. While prompt optimization is technically a subcategory of adaptation strategies, it is presented here as a separate category because it currently represents the most frequently used approach and warrants particular attention. Researchers are encouraged to clearly report as many relevant items as possible. Some items may not be applicable depending on the study context and design. Detailed explanations of each item are provided in the subsequent ELABORATION section.

ELABORATION

Model Identification

LLM performance may vary across model variations and versions; therefore, it is essential to clearly identify the specific model under investigation. At a minimum, studies should report the model's name, version (ideally, including minor version details), developer, and whether it utilizes proprietary or open-source models.

Proprietary LLMs—particularly those accessed via web-based chatbot interfaces—are typically updated on an ongoing basis, so users may not have full visibility into recent updates [11,12]. For instance, an interface advertising "gpt-4o" may actually be running any of the available minor versions (referred to as snapshots), such as gpt-4o-2024-05-13, gpt-4o-2024-08-06, or gpt-4o-2024-11-20, without notice [13]. In such cases, documenting the exact date of model access and query execution enhances transparency.

For open-source models self-managed through local deployments, additional identification details become particularly important. Beyond the base model name and version, authors should specify any modifications made to the model architecture or configuration files, the source of the model weights (e.g., official repository, third-party implementation), and the specific commit hash or release tag when available. This level of detail is essential for reproducibility, as even minor differences in implementation or weight initialization can affect model behavior.

If feasible, authors are encouraged to share the specific model used in a study in an executable form by uploading it to a public online repository and providing a URL link. This enhances transparency and reproducibility and may enable others to replicate or build upon the study using the same model configuration. Additionally, when known, the cutoff date of the model's training data should be reported, as it shows the scope and currency of the model's knowledge base.

Model Access Mode

LLMs can be accessed through various modes, including web-based chatbot interfaces (e.g., ChatGPT), APIs (e.g., OpenAI API for GPT models) [7], and self-managed local deployment. Each access mode has distinct characteristics that may influence model performance (Table 2). Therefore, the specific access mode used in a study should be clearly stated.

Early studies evaluating the accuracy of LLMs in healthcare often used proprietary web-based chatbot interfaces due to their ease of use [14-16]. However, this access mode has notable limitations (Table 2). For instance, proprietary chatbot interfaces typically offer limited customization—such as minimal or no control

https://doi.org/10.3348/kjr.2025.1522

Korean Journal of Radiology
KJR

Table 1. Minimum items for the transparent reporting of clinical studies that present the performance of LLMs

Checklist item category	Details	Yes/No/NA
Model identification	• Model name, version (ideally, including minor version details), and developer • Whether the model is proprietary or open-source • For proprietary LLMs: date of access and query execution • For self-managed open-source models: modifications to the base model, the source of model weights, and the commit hash or release tag when available • If feasible, sharing of the specific model used, along with a public repository URL • When known, cutoff date of training data	
Access mode	• Access method: web-based chatbot, API, or self-managed local deployment • Rationale for chosen access mode • Disclosure of any system-level features beyond the LLM itself (e.g., system prompts, intersession memory) when known • For self-managed local deployment, key computational environment details such as (GPU type and memory)	
Input data type	• Sufficient details on the type and format of data used with, or as part of, input prompts for LLM evaluation, to enable replication	
Adaptation strategy used	• Specification of the adaptation method(s) used, including a clear statement on whether model weights were altered (e.g., fine-tuning) or not (e.g., prompt optimization, RAG) • Use of precise terminology for non-parametric adaptation, e.g., "adaptation data" or "prompt development data" rather than ambiguous terms like "fine-tuning" or "training" data • Provision of a detailed methodological description of the adaptation process (extended details can be included in supplementary materials, if space is limited)	
Prompt optimization procedures	• Steps taken to create the prompts • Rationale behind selecting specific wording over alternatives (e.g., standard terminology, guideline alignment) • Specification of any deliberate prompting strategies used (e.g., chain-of-thought, reflection, instruction, few-shot) • Full, directly executable (i.e., copy-paste ready) text of representative prompts and, if feasible, a complete record of the prompts used as supplementary materials • Summary of the prompt optimization process, such as the number of iterations and interim versions of prompts (as supplementary materials, if space is limited)	
Prompt execution setup	• Specific query submission configuration, such as: - For chatbot interface use: whether all questions were entered simultaneously or submitted sequentially over the course of a dialogue - For API use: whether queries were submitted as independent calls or as part of a constructed dialogue (e.g., including prior exchanges in the input) • If feasible, the entire experiment script used for prompt execution as supplementary materials	
Stochasticity management	• Settings of technical parameters, such as the temperature, that modify the level of randomness • Number of querying attempts made • Method for synthesizing multiple responses (e.g., majority vote, average score, at least one correct answer across attempts), and the rationale behind it • Analysis of the reliability of the LLM outputs across multiple attempts	
Independence of test data	• Clear disclosure of any overlap between test data and either training or adaptation data • Specification of the nature and source of data used for model adaptation and test • For test data sourced online: exact URLs, accessibility, and potential prior exposure within the model's training corpus	

LLM = large language model, NA = not applicable, URL = uniform resource locator, API = application programming interface, GPU = graphics processing unit, RAG = retrieval-augmented generation

 https://doi.org/10.3348/kjr.2025.1522

Table 2. Comparison of typical access modes for LLMs

Characteristic	Web-based chatbot interface (e.g., ChatGPT)	API access to proprietary LLMs (e.g., OpenAI API for GPT models)	Local deployment (e.g., open-source models such as LLaMA)
Ease of use	Very easy; no programming skills needed	Requires basic programming or scripting	Requires advanced technical skills for setup and use
Customization & control	Minimal; predefined settings	High, including control over hyperparameters (e.g., temperature), output format (e.g., JSON), and integration with external data sources (e.g., RAG)	Very high: full control, also including fine-tuning through additional training on domain-specific data
Behavior transparency	May include opaque features (e.g., intersession memory, system-level prompts)	Transparent and controllable behavior	Fully transparent; all components user-controlled
Batch processing	Not supported or limited	Supported; suitable for automation	Fully supported; customizable workflows
Data security	Data sent to external servers	Data sent to external servers	Data remains local; highest level of security
Cost	Often free or subscription-based	Token-based; can become costly at scale	Minimal usage cost after setup; requires hardware resources

LLM = large language model, GPT = generative pretrained transformer, API = application programming interface, JSON = JavaScript object notation, RAG = retrieval-augmented generation

over hyperparameters like temperature (which influences response randomness, or stochasticity)—and usually lack the ability to integrate with external data sources (e.g., through retrieval-augmented generation [RAG] [17,18]). Proprietary chatbot interfaces may involve opaque system-level features beyond the LLM itself, such as intersession memory or hidden system prompts, which often include elements designed to personalize responses for individual users. These system-level features can be updated dynamically and without notice. They can introduce user-specific variability and undermine the reproducibility of results. Therefore, the system-level features should be reported when known.

A key characteristic of chatbot interfaces is the use of contextual memory within a chat session. Multiple user queries and the model's responses in a chat session are linked together, and the model's answer to a given query is influenced by the preceding exchanges within the session. In contrast, API access treats each query as an independent call, meaning the response to one question is unaffected by previous interactions, unless the user deliberately constructs a chat session by including prior exchanges in the input [7]. Thus, unless the task evaluated in a study inherently involves sequential dialogue (e.g., evaluation of an LLM simulating a patient interview), web-based chatbot interfaces are generally not ideal for evaluating LLM accuracy. For example, when assessing an LLM's performance in answering standalone medical questions—such as case-based quiz items from medical journals [14,19] or generating differential diagnoses based on structured clinical vignettes (e.g., a set of history, physical examination, and laboratory results) [20]—API access, with each question submitted as a separate API call, is more appropriate. This approach ensures that each question is handled independently and minimizes sources of variability.

Self-managed local deployment involves researchers deploying and controlling the model infrastructure themselves, typically using open-source models. This mode offers maximal transparency, flexibility, and data privacy protection by processing data locally without transmission to external servers—particularly important when handling sensitive patient information. However, self-managed local deployment requires significant computational resources and technical expertise. For self-managed local deployment, we recommend researchers report key computational environment details such as hardware specifications (e.g., graphics processing unit type and memory), processing time when feasible, and other relevant infrastructure requirements. This documentation promotes reproducibility and provides valuable evidence for assessing practical feasibility in clinical settings.

Input Data Type

Authors should clearly specify the type of data used with, or as part of, the input prompts to evaluate the LLM. Common examples include structured or unstructured

https://doi.org/10.3348/kjr.2025.1522

Korean Journal of Radiology
KJR

electronic health record data as text (such as radiology reports, clinical notes, or lab results) as well as medical images. Sufficient detail on the data type and format should be provided to enable replication by readers.

Model Adaptation

Studies often employ various model adaptation strategies to improve LLM performance for specific tasks or domains under investigation. These strategies generally fall into two fundamentally different categories [21,22]:

- Non-parametric approaches, which do not alter the model's internal parameters (i.e., weights), such as prompt optimization or integration with external knowledge via RAG or web search tools.
- Parametric approaches, which do modify the model's parameters—most commonly through fine-tuning through additional model training using domain-specific datasets.

It is important to clearly distinguish between these two types of adaptation. Parametric adaptation leads to permanent changes in the model itself, whereas non-parametric methods affect performance only within the specific study setup. Because the latter does not alter the model's weights, their effects are not inherently reproducible unless the same adaptation procedures are applied prior to model use.

Despite their fundamental differences, these approaches are sometimes described in the literature without clear distinction, using terms such as "fine-tuning" or "training" in a broader, less precise sense. As technical jargon, these terms specifically refer to the process of modifying a model's parameters using additional training data. Using these terms more generally to refer to any procedure intended to refine model performance can create confusion. Precise terminology is essential for clear communication. For instance, the term "training data" is sometimes used by authors to refer to data used in non-parametric adaptation. However, this can be misleading, as the small amount of data used for prompt development or retrieval setup is fundamentally different from the training data used in traditional machine learning pipeline. To promote clarity, it is preferable to use more specific terms such as "adaptation data" or "prompt development data."

Study reports should describe the adaptation strategy in specific terms and provide sufficient methodological detail. If space is limited, such information can be included in supplementary materials.

Prompt Optimization

Prompt optimization, including various forms of prompt engineering, currently appears to be the most frequently used adaptation strategy to improve model performance in clinical studies evaluating LLMs and warrants careful attention.

Thorough documentation of both the methods used and the rationale behind prompt design is essential. Even small changes in prompt wording—such as replacing a single word—can result in substantial variation in the model's outputs, a phenomenon known as prompt brittleness [12,23]. For example, in a radiology study, the difference between phrasing a task as "Calculate the LI-RADS category" versus "Determine the LI-RADS category," though subtle, resulted in substantially different model outputs [24]. When applicable, authors should explain the rationale for specific word choices, such as the use of standardized terminology or alignment with terms from clinical guidelines.

If more deliberate prompting strategies beyond basic prompt phrasing were used, these should be explicitly described. Common examples include chain-of-thought prompting, which guides the model to reason step by step; reflection prompting, which encourages the model to critique or revise its own responses; instruction prompting, which provides clear task directives; and few-shot prompting, which demonstrates task structure by including a few examples in the input prompt for in-context learning [21].

Given the sensitivity of LLM outputs to prompt formulation, complete transparency is essential. Authors should provide the full, exact text of representative prompts in a form that is directly executable (i.e., copy-paste ready) by readers. This also includes any custom instructions, if applicable (e.g., "You are an experienced physician..."). This level of detail is critical to ensure both reproducibility and accurate interpretation of study findings.

If feasible, it is even better if authors provide a complete record of the prompts used, such as in the form of the entire experiment script, as supplementary materials, to further support reproducibility. Additionally, it is encouraged to provide a summary of the prompt optimization process, such as the number of iterations or testing rounds involved and interim versions of prompts as supplementary materials. Reporting unsuccessful prompt variations or optimization attempts can be valuable. When certain prompt formulations or strategies were tested but did not yield satisfactory results, documenting these negative results—including the rationale for abandonment—can prevent others

 https://doi.org/10.3348/kjr.2025.1522

from repeating ineffective approaches and contribute to collective learning in the field.

Prompt Execution

A clear description of how queries (prompts) were executed is essential, as this directly affects the reproducibility of LLM responses. If a chatbot interface was used, further clarification is needed on whether all questions were entered simultaneously or submitted sequentially over the course of a dialogue. For API-based use, an explicit statement on whether queries were submitted as independent calls or as part of a constructed dialogue enhances transparency. When an API or local deployment was used, providing the entire experiment script as supplementary materials is encouraged, since it transparently conveys not only the prompt text but also the execution-specific settings including hyperparameters as well as exact model name and version.

Stochasticity Management

Unlike traditional AI models that produce consistent outputs for given inputs through deterministic operations, LLMs can generate different responses even when prompted repeatedly with the exact same input. This phenomenon, known as stochasticity, arises from inherent random elements in the way LLMs generate outputs [12,25]. For example, when an LLM generates a response to the prompt, "The most likely diagnosis is...," in the context of a patient presenting with fever, cough, and shortness of breath, it predicts the next word based on learned probabilities. It might assign different probabilities to words like "pneumonia," "COVID-19," or "pneumothorax." Rather than always selecting the most probable word, the model introduces a degree of randomness. As a result, while probability remains the dominant factor, a less likely word may occasionally be chosen, and the output can vary from one attempt to another.

The level of randomness in an LLM's behavior can be adjusted. A key parameter is temperature, which controls how closely the model follows the highest-probability output. Lower temperature values (approaching zero) make the model more deterministic, producing more consistent responses across attempts, whereas higher values increase variability [25].

Given this inherent variability, researchers should clearly report how stochasticity was managed in their study. This includes describing relevant technical settings [26]—particularly the temperature value used—and specifying whether a single query or multiple querying attempts were made for each input. If repeated querying was employed, the number of attempts should be reported, along with an explanation of how multiple responses were synthesized for analysis—such as accepting any correct answer across attempts, using the response from the first attempt, calculating an average score, or applying a majority vote. The rationale for these choices should be provided. In addition, where applicable, authors should assess the consistency of responses across attempts, as this informs the reliability of the model's performance under repeated conditions [27].

Test Data Independence

Clarification on the independence of test data from both the foundational model's training data and any data used for model adaptation is essential. Even in non-parametric adaptation—whether through prompt design, example selection, or retrieval strategy—researchers often use a small dataset to iteratively refine and optimize the adaptation setup. If any data were used during this process, it is essential to clearly describe the nature of those data, separately from the description of the dataset used for model testing, as would be expected in any well-documented AI study involving data use [28].

Any overlap between datasets used for model adaptation and testing or between the foundational model's training data and test data can result in data leakage, which may lead to an overestimation of the LLM's performance. Moreover, the issue of data independence extends beyond direct data duplication. If individuals involved in model adaptation were not blinded to the test data, researchers familiar with the test set may inadvertently craft prompts or select examples that favor performance on that test. This can result in indirect leakage, even when the same data are not reused. Clarifying whether such blinding was maintained is therefore recommended.

Additionally, because LLMs are typically trained on massive datasets collected through extensive scraping of online sources, including publicly available internet content, there is a risk that test data obtained from such sources—for example, online question banks or items from a medical journal—may have been included in the model's original training set, introducing a risk of unintentional data leakage [12,29]. If test data were sourced from the internet, the exact origin (including URLs), accessibility

https://doi.org/10.3348/kjr.2025.1522

status, and whether copies may exist elsewhere online should be clearly reported.

REPORTING EXAMPLES FROM THE LITERATURE

The following examples illustrate how the reporting elements have been addressed in recent studies. Quotations italicized are taken directly from the original sources, with ellipses (...) used to indicate omitted text for brevity.

LLM Identification: Minor Versions and Knowledge Cutoff Dates

"For pilot testing, we selected several established open-weight models from the LMSYS Chatbot Arena LLM Leaderboard:
- *Microsoft: Phi-3-mini, Phi-3-medium (both with October 2023 knowledge cut-offs)*
- *Mistral AI: Mistral-7B-v0.3 (undisclosed cut-off)*
- *Meta: Llama-3-8b-instruct (March 2023 cut-off), Llama-3-70b-instruct (December 2023 cut-off)*
- *Google: Gemma-2-9b-it, Gemma-2-27b-it (undisclosed cut-offs)."* [30]

LLM Identification: Access Dates

"The artificial intelligence models used in this study were LLMs with vision capabilities: GPT-4V, GPT-4o, Gemini, and Claude. The four LLMs were accessed between April 29 and May 15, 2024." [31]

Knowledge Cutoff Dates and Access Modes

"Responses for each case were collected using the chat web interfaces of OpenAI's OpenAI o1 (knowledge cutoff: October 2023), GPT-4o (knowledge cutoff: October 2023), and GPT-4 (knowledge cutoff: December 2023)... Responses were recorded using the application programming interfaces for Google's Gemini 1.5 Pro and Gemini 1.5 Flash (knowledge cutoff: August 2024), and Meta's Llama 3.2-90B-Vision and Llama 3.2-11B-Vision (knowledge cutoff: December 2023)." [32]

Access via API and Query Independence From Prior Interactions

"Since the software uses the OpenAI API, the experiments for this study were also conducted using the API. In addition, using the API eliminated the bias that could result from ChatGPT's ability to reference previous requests." [33]

Self-Managed Model Deployment and Computational Environment

"Running the Llama 3.2-11B-Vision model requires a high-end graphics processing unit (GPU) with at least 22 GB of GPU memory for efficient inference, whereas the Llama 3.2-90B-Vision model requires at least 180 GB of GPU memory to accommodate its full parameter set. For this study, a single 80-GB GPU Nvidia A100 was used for the 11B model, and three 80-GB Nvidia A100 GPUs were used for the 90B model through distributed inference using the HuggingFace application programming interface." [32]

Input Data Types

Text—*"All cases in this study were based on actual patients and included information available on initial diagnostic evaluation, such as history, physical examination, and laboratory test results... A representative example is included in eTable 1 in Supplement 2."* [20]

Image and text—*"The case vignettes were captured as screenshots with a size of 1285 x 768 pixels, whereas the corresponding questions were documented separately in text files."* [32]

Image and image capture of text—*"Patient history, original images, and figure legends (without imaging findings) were extracted from PDF files of published cases and used as input images... There were two sets of input images. The first image set was composed of extracted original images acquired with various imaging modalities, including radiography, US, CT, MRI, fluoroscopy, digital subtraction angiography, bone scintigraphy, and PET/CT. The second image set was composed of captured images of text from the Diagnosis Please cases, which were the patient history and figure legends."* [34]

Prompt Optimization as an Adaptation Strategy

"There are 3 ways to prompt engineering: Zero-shot, One-shot, and Few-shot. In the Zero-shot prompting method, the model is given natural language instructions without examples or demonstrations. In the One-shot prompting method, the model is provided instructions using a single example... The model was provided with 2 examples in the Few-shot prompting method. Details of the instructions and examples used in this study are provided in Supplementary Table 1." [33]

RAG as an Adaptation Strategy

"Additional details regarding the prompt format, reference

https://doi.org/10.3348/kjr.2025.1522

standard process, and LLM settings are provided in Appendix S1… Appendix S1: The above process was then performed with RAG integration added, using the embedding models RadSearch and GTE-large. In this evaluation, RadSearch was given the report finding description as a search query and retrieved the most similar full report (n=1). This report was then added to the LLM input as context to assist the LLM in providing the correct diagnosis for the report finding description." [35]

Fine-Tuning as an Adaptation Strategy

"The data set for fine-tuning was obtained from the following three sources: medical instruction sets (from medical books, guidelines, case reports, and knowledge graphs), radiology reports, and innocuous public instruction sets. A total of 800 radiology reports were sampled for fine-tuning, which were balanced based on radiologic modality and anatomic site. The data for fine-tuning were pairs of instructions and corresponding outputs. Instructions are inputs that prompt the model to produce specific outputs, usually describing specific tasks in natural language. GPT-4 (OpenAI) was used to automatically extract instructions and outputs from medical text (Appendix S1) and for radiologists to manually create instructions and outputs for radiology reports. The instructions and outputs were extended by the Self-Instruct and Evol-Instruct methods… To fine-tune the model using instruction learning, the instructions was preprocessed (Appendix S1). First, duplicates were removed based on their similarity with a deduplication threshold of 0.95. Second, the instruction-following difficulty was calculated to select data samples with the potential to enhance LLM instruction tuning. Pretraining and fine-tuning were run on a Linux platform (Ubuntu 20.04; Canonical) with eight graphics processing units (GPUs) (A800; NVIDIA)." [36]

Stochasticity Management via Deterministic Settings

"The temperature hyperparameter controls this randomness, with a high temperature adding more randomness. For this analysis, Vicuna outputs were obtained using a temperature setting of 0, thus removing the randomness." [37]

"Model inference was performed using the Transformers library (v4.43) and Python (v3.10.14). Model responses were constrained to JSON format to facilitate evaluation. Greedy search decoding was applied to ensure deterministic output. Due to VRAM constraints, quantization was applied using the bitsandbytes library: 4-bit for Llama-3-70b-instruct and 8-bit for Gemma-2-27b-it." [30]

Stochasticity Management via Repeated Queries

"Considering the inherent stochasticity in responses… each test question was presented to ChatGPT three times in three distinct sessions. The results from the initial session of ChatGPT analysis for each academic year were used for the main analysis. The consistency of ChatGPT's responses across three separate sessions was analyzed using the Fleiss' kappa." [38]

"The majority vote of the three runs at the default temperature setting of 0.7 was determined and compared with the output of Vicuna with a temperature setting of 0." [37]

"We computed the average risk score from the five iterations for each subject and then calculated the AUROC for this average risk score… We determined the coefficient of variation (CV) for the iterations per subject and calculated the average CV across all subjects to quantify the variability of the GPT-based risk score." [39]

"The LLMs were tasked with providing three differential diagnoses, repeated five times at temperatures 0, 0.5, and 1… The result correct if the generated diagnoses included the final diagnosis after five repetitions." [34]

Test Data Independence

"Three radiologists generated 160 fictitious free-text liver MRI reports… Of these, 20 were used for prompt engineering, and 140 formed the internal test cohort. Seventy-two genuine reports, authored by 17 radiologists were collected and de-identified for the external test cohort." [40]

"A radiologist with experience in prompt engineering performed manual refinement of the prompts… Only the prompt development set was used for this process, ensuring that the internal validation and test sets remained unseen to prevent data leakage." [30]

"Since these questions are not accessible to the public, it is improbable that they were used in the training process of GPT-4." [38]

CONCLUSION

Careful attention to the considerations outlined in this article can help ensure transparency in reporting and enable more accurate and meaningful evaluation of studies assessing LLM performance in healthcare applications. As MI-CLEAR-LLM specifically addresses the reporting of how LLMs are specified, accessed, adapted, and applied in testing, we also encourage researchers to consult recently

https://doi.org/10.3348/kjr.2025.1522

published, more comprehensive reporting guidelines for LLM-related studies [8-10].

Conflicts of Interest

Seong Ho Park: Editor-in-Chief of the *Korean Journal of Radiology* without involvement in the editorial evaluation or decision to publish this article; honoraria from Bayer and Korean Society of Radiology; support for travel from Korean Society of Radiology; consulting fees from Ministry of Food and Drug Safety of the Republic of Korea; consulting fees from National Evidence-based healthcare Collaborating Agency of the Republic of Korea.
Chong Hyun Suh: Assistant to the Editor of the *Korean Journal of Radiology*, was not involved in the editorial evaluation or decision to publish this article.
Jeong Hyun Lee: Assistant to the Editor of the *Korean Journal of Radiology*, was not involved in the editorial evaluation or decision to publish this article.
Ali S. Tejani: Nothing to disclose.
Seng Chan You: Grants from Daiichi Sankyo; compensation from the *Journal of the American College of Cardiology* (*JACC*) as Associate Editor; Chief executive officer of PHI Digital Healthcare.
Charles E. Kahn, Jr: Salary support from the Radiological Society of North America (RSNA) paid to employer.
Linda Moy: Grants from Siemens; consulting fees from Bracco, Guerbet, and Medscape; support for travel from European Society of Radiology and Korean Society of Radiology; participation on board of ACR DSMB; Society of Breast Imaging board.

Author Contributions

Writing—original draft: Seong Ho Park. Writing—review & editing: Chong Hyun Suh, Jeong Hyun Lee, Ali S. Tejani, Seng Chan You, Charles E. Kahn, Jr, Linda Moy.

ORCID IDs

Seong Ho Park
https://orcid.org/0000-0002-1257-8315
Chong Hyun Suh
https://orcid.org/0000-0002-4737-0530
Jeong Hyun Lee
https://orcid.org/0000-0002-7125-8899
Ali S. Tejani
https://orcid.org/0000-0002-6862-5299
Seng Chan You
https://orcid.org/0000-0002-5052-6399
Charles E. Kahn, Jr
https://orcid.org/0000-0002-6654-7434
Linda Moy
https://orcid.org/0000-0001-9564-9360

Funding Statement

None

REFERENCES

1. Kim S, Lee CK, Kim SS. Large language models: a guide for radiologists. *Korean J Radiol* 2024;25:126-133
2. Kim K, Cho K, Jang R, Kyung S, Lee S, Ham S, et al. Updated primer on generative artificial intelligence and large language models in medical imaging for medical professionals. *Korean J Radiol* 2024;25:224-242
3. Huo B, Boyle A, Marfo N, Tangamornsuksan W, Steen JP, McKechnie T, et al. Large language models for chatbot health advice studies: a systematic review. *JAMA Netw Open* 2025;8:e2457879
4. Ko JS, Heo H, Suh CH, Yi J, Shim WH. Adherence of studies on large language models for medical applications published in leading medical journals according to the MI-CLEAR-LLM checklist. *Korean J Radiol* 2025;26:304-312
5. Suh CH, Yi J, Shim WH, Heo H. Insufficient transparency in stochasticity reporting in large language model studies for medical applications in leading medical journals. *Korean J Radiol* 2024;25:1029-1031
6. Park SH, Suh CH, Lee JH, Kahn CE, Moy L. Minimum reporting items for clear evaluation of accuracy reports of large language models in healthcare (MI-CLEAR-LLM). *Korean J Radiol* 2024;25:865-868
7. Park CR, Heo H, Suh CH, Shim WH. Uncover this tech term: application programming interface for large language models. *Korean J Radiol* 2025;26:793-796
8. Gallifant J, Afshar M, Ameen S, Aphinyanaphongs Y, Chen S, Cacciamani G, et al. The TRIPOD-LLM reporting guideline for studies using large language models. *Nat Med* 2025;31:60-69
9. Tripathi S, Alkhulaifat D, Doo FX, Rajpurkar P, McBeth R, Daye D, et al. Development, evaluation, and assessment of large language models (DEAL) checklist: a technical report. *NEJM AI* 2025;2:AIp2401106
10. CHART Collaborative. Reporting guideline for chatbot health advice studies: the CHART statement. *JAMA Netw Open* 2025;8:e2530220
11. Thirunavukarasu AJ, Ting DSJ, Elangovan K, Gutierrez L, Tan TF, Ting DSW. Large language models in medicine. *Nat Med* 2023;29:1930-1940
12. Kaddour J, Harris J, Mozes M, Bradley H, Raileanu R, McHardy R. Challenges and applications of large language models. arXiv [Preprint]. 2023 [accessed on October 12, 2025]. Available at: https://doi.org/10.48550/arXiv.2307.10169
13. OpenAI Platform. Models: GPT-4o [accessed on October 12,

2025]. Available at: https://platform.openai.com/docs/models/gpt-4o
14. Kanjee Z, Crowe B, Rodman A. Accuracy of a generative artificial intelligence model in a complex diagnostic challenge. *JAMA* 2023;330:78-80
15. Rahsepar AA, Tavakoli N, Kim GHJ, Hassani C, Abtin F, Bedayat A. How AI responds to common lung cancer questions: ChatGPT vs Google Bard. *Radiology* 2023;307:e230922
16. Bhayana R, Krishna S, Bleakney RR. Performance of ChatGPT on a radiology board-style examination: insights into current strengths and limitations. *Radiology* 2023;307:e230582
17. Ng KKY, Matsuba I, Zhang PC. RAG in health care: a novel framework for improving communication and decision-making by addressing LLM limitations. *NEJM AI* 2025;2:AIra2400380
18. Fink A, Rau A, Reisert M, Bamberg F, Russe MF. Retrieval-augmented generation with large language models in radiology: from theory to practice. *Radiol Artif Intell* 2025;7:e240790
19. Han T, Adams LC, Bressem KK, Busch F, Nebelung S, Truhn D. Comparative analysis of multimodal large language model performance on clinical vignette questions. *JAMA* 2024;331:1320-1321
20. Goh E, Gallo R, Hom J, Strong E, Weng Y, Kerman H, et al. Large language model influence on diagnostic reasoning: a randomized clinical trial. *JAMA Netw Open* 2024;7:e2440969
21. Kim TT, Makutonin M, Sirous R, Javan R. Optimizing large language models in radiology and mitigating pitfalls: prompt engineering and fine-tuning. *Radiographics* 2025;45:e240073
22. Bluethgen C, Van Veen D, Zakka C, Link KE, Fanous AH, Daneshjou R, et al. Best practices for large language models in radiology. *Radiology* 2025;315:e240528
23. Kim W. Seeing the unseen: advancing generative AI research in radiology. *Radiology* 2024;311:e240935
24. Lee JH, Shin J. How to optimize prompting for large language models in clinical research. *Korean J Radiol* 2024;25:869-873
25. Bhayana R. Chatbots and large language models in radiology: a practical primer for clinical and research applications. *Radiology* 2024;310:e232756
26. Wolf T, Debut L, Sanh V, Chaumond J, Delangue C, Moi A, et al. Transformers: state-of-the-art natural language processing [accessed on October 12, 2025]. Available at: http://doi.org/10.18653/v1/2020.emnlp-demos.6
27. Park SH, Kim N. Challenges and proposed additional considerations for medical device approval of large language models beyond conventional AI. *Radiology* 2024;312:e241703
28. Park SH, Suh CH. Reporting guidelines for artificial intelligence studies in healthcare (for both conventional and large language models): what's new in 2024. *Korean J Radiol* 2024;25:687-690
29. Sahoo SS, Plasek JM, Xu H, Uzuner Ö, Cohen T, Yetisgen M, et al. Large language models for biomedicine: foundations, opportunities, challenges, and best practices. *J Am Med Inform Assoc* 2024;31:2114-2124
30. Lee JH, Min JH, Gu K, Han S, Hwang JA, Choi SY, et al. Automated resectability classification of pancreatic cancer CT reports with privacy-preserving open-weight large language models: a multicenter study. *J Med Syst* 2025;49:118
31. Suh PS, Shim WH, Suh CH, Heo H, Park KJ, Kim PH, et al. Comparing large language model and human reader accuracy with New England Journal of Medicine image challenge case image inputs. *Radiology* 2024;313:e241668
32. Hou B, Mukherjee P, Batheja V, Wang KC, Summers RM, Lu Z. One year on: assessing progress of multimodal large language model performance on RSNA 2024 case of the day questions. *Radiology* 2025;316:e250617
33. Kim H, Jin HM, Jung YB, You SC. Patient-friendly discharge summaries in Korea based on ChatGPT: software development and validation. *J Korean Med Sci* 2024;39:e148
34. Suh PS, Shim WH, Suh CH, Heo H, Park CR, Eom HJ, et al. Comparing diagnostic accuracy of radiologists versus GPT-4V and Gemini Pro Vision using image inputs from diagnosis please cases. *Radiology* 2024;312:e240273
35. Savage CH, Chaudhari G, Smith AD, Sohn JH. RadSearch, a semantic search model for accurate radiology report retrieval with large language model integration. *Radiology* 2025;315:e240686
36. Zhang L, Liu M, Wang L, Zhang Y, Xu X, Pan Z, et al. Constructing a large language model to generate impressions from findings in radiology reports. *Radiology* 2024;312:e240885
37. Mukherjee P, Hou B, Lanfredi RB, Summers RM. Feasibility of using the privacy-preserving large language model vicuna for labeling radiology reports. *Radiology* 2023;309:e231147
38. Kim H, Kim P, Joo I, Kim JH, Park CM, Yoon SH. ChatGPT vision for radiological interpretation: an investigation using medical school radiology examinations. *Korean J Radiol* 2024;25:403-406
39. Han C, Kim DW, Kim S, You SC, Park JY, Bae S, et al. Evaluation of GPT-4 for 10-year cardiovascular risk prediction: insights from the UK Biobank and KoGES data. *iScience* 2024;27:109022
40. Gu K, Lee JH, Shin J, Hwang JA, Min JH, Jeong WK, et al. Using GPT-4 for LI-RADS feature extraction and categorization with multilingual free-text reports. *Liver Int* 2024;44:1578-1587

https://doi.org/10.3348/kjr.2025.1522