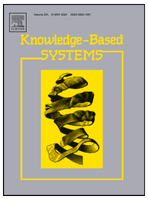
Contents lists available at ScienceDirect

Knowledge-Based Systems

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



User-defined trade-offs in LLM benchmarking: balancing accuracy, scale, and sustainability

Ana Gjorgjevikj [a,*], Ana Nikolikj [a,b], Barbara Koroušić Seljak [a,b], Tome Eftimov [a,b]

[a] Computer Systems Department, Jožef Stefan Institute, Ljubljana, Slovenia
[b] Jožef Stefan International Postgraduate School, Ljubljana, Slovenia

ARTICLE INFO

Keywords:
Machine learning
Natural language processing
Large language model
Benchmarking
Multi-criteria decision-making
Promethee method

ABSTRACT

This paper presents xLLMBench, a transparent, decision-centric benchmarking framework that empowers decision-makers to rank large language models (LLMs) based on their preferences across diverse, potentially conflicting performance and non-performance criteria, e.g., domain accuracy, model size, energy consumption, CO_2 emissions. Existing LLM benchmarking methods often rely on individual performance criteria (metrics) or human feedback, so methods systematically combining multiple criteria into a single interpretable ranking lack. Methods considering human preferences typically rely on direct human feedback to determine rankings, which can be resource-intensive and not fully aligned with application-specific requirements. Motivated by current limitations of LLM benchmarking, xLLMBench leverages multi-criteria decision-making methods to provide decision-makers with the flexibility to tailor benchmarking processes to their requirements. It focuses on the final step of the benchmarking process (robust analysis of benchmarking results) which in LLMs' case often involves their ranking. The framework assumes that the selection of datasets, metrics, and LLMs involved in the experiment is conducted following established best practices. We demonstrate xLLMBench's usefulness in two scenarios: combining LLM results for one metric across different datasets and combining results for multiple metrics within one dataset. Our results show that while some LLMs maintain stable rankings, others exhibit significant changes when correlated datasets are removed, when the focus shifts to contamination-free datasets or fairness metrics. This highlights that LLMs have distinct strengths/weaknesses, going beyond overall performance. Our sensitivity analysis reveals robust rankings, while the diverse visualizations enhance transparency. xLLMBench can be used with existing platforms to support transparent, reproducible, and contextually-meaningful LLM benchmarking.

1. Introduction

Large language models (LLMs) are gradually becoming an integral part of modern living. They are revolutionizing both academia and industry by improving research capabilities, automating complex tasks, and enabling more efficient knowledge dissemination [1]. Their usage spans from answering simple everyday questions to helping with complex professional tasks in knowledge-intensive domains. In academia, LLMs assist in generating literature reviews, facilitating data analysis, and offering new insights through advanced natural language processing. In industry, they streamline customer services with sophisticated chatbots, optimize content creation, and improve decision-making through predictive analysis. LLMs are driving innovation and efficiency by bridging the gaps between large datasets and actionable insights. The number of available LLMs is constantly increasing, but they differ in the degree to which they exhibit different abilities in different domains and machine learning (ML) tasks (e.g., text generation, text summarization, open/closed question answering). Furthermore, LLMs are highly complex models with poor interpretability of their decision-making process [2], prone to challenges that differ from those common in task-specific language models. One such challenge is hallucination, which refers to the generation of content that may seem credible at first sight, but which is non-factual [3]. Therefore, well-established evaluation protocols may be insufficient for a thorough evaluation of LLM capabilities, so reliable benchmarking methods that provide interpretable results are urgently needed [2].

Benchmarking best practices in general suggest five key steps, i.e., (B1) selecting diverse and unbiased problem portfolio (benchmark datasets), (B2) choosing complementary algorithm portfolio (LLMs in this case), (B3) ensuring a fair experimental design with consistent train-test splits, (B4) selecting suitable performance metrics, and (B5) conducting robust analyses. The formation of the problem portfolio (B1)

[*] Corresponding author.
E-mail addresses: ana.gjorgjevikj@ijs.si (A. Gjorgjevikj), ana.nikolikj@ijs.si (A. Nikolikj), barbara.korousic@ijs.si (B. Koroušić Seljak), tome.eftimov@ijs.si (T. Eftimov).

https://doi.org/10.1016/j.knosys.2025.114405
Received 27 May 2025; Received in revised form 19 August 2025; Accepted 1 September 2025
Available online 10 September 2025

usually involves landscape analysis, clustering, and graph algorithms to ensure that the benchmarks cover the entire feature space [4–9]. Studies in various fields recommend combining benchmark datasets for representative data. The algorithm portfolio selection (B2), which significantly affects the results, should result in a portfolio of models with complementary strengths. The guidelines suggest using high-level algorithm properties or performance-based comparisons [10]. Fair experiment design (B3) and choice of performance metrics (B4) are also critical [11], as correlated metrics can skew results. Metrics can include accuracy, fairness [12], bias, human feedback [13], robustness, to name a few. Finally, the analyses (B5) should give robust and transparent results. In the case of LLMs, B5 often involves LLM ranking to facilitate the selection of an LLM for a specific use case based on multiple (often conflicting) criteria, since LLMs are rather non-transparent models, the capabilities of which require thorough evaluation on diverse tasks and datasets.

In recent years, numerous datasets have been proposed to evaluate various LLM capabilities, with a recent survey identifying 112 such datasets [14]. However, these datasets span different domains, address different ML tasks (e.g., text summarization, question answering), and employ diverse performance metrics. This diversity makes the selection of appropriate datasets for application-specific LLM evaluation increasingly challenging [14]. Even when such datasets can be identified, defining clear performance thresholds for an LLM to qualify as a suitable candidate remains difficult. Tasks and datasets do not carry equal importance for every application, making it challenging to determine which (and to what extent) should be given higher importance than others. Furthermore, real-world ML model selection involves non-trivial trade-offs and constraints, such as tight product release schedules, limited hosting infrastructure, low subscription costs, and minimal environmental impact, all of which can restrict the set of viable models (for an overview of such requirements in ML projects, see [15]). Selecting an LLM based on these multiple, often conflicting, criteria is therefore a demanding task for decision-makers (e.g., researchers conducting studies or practitioners building a product), where suboptimal choices can lead to reduced performance on the target use case, additional time spent re-evaluating alternatives, and increased costs. To address this complexity, most LLM benchmarking platforms attempt to calculate an aggregate metric across tasks, datasets, and individual performance metrics to produce a single LLM ranking (e.g., see [16]). However, these platforms also acknowledge that LLM capabilities in different datasets are often uncorrelated, highlighting the need for decision-makers to select LLMs based on the capabilities relevant to their application [16]. The recent discontinuation of Hugging Face's widely used Open LLM Leaderboard[1] further demonstrates the non-trivial challenges involved in benchmarking and ranking LLMs.

From the above, it can be concluded that existing LLM benchmarking approaches predominantly rely on individual metrics or direct human feedback to determine rankings. However, real-world LLM selection is rarely driven by a single performance metric, but instead requires balancing multiple, often conflicting, application-specific requirements. Although existing methods provide valuable insights, they often fail to systematically integrate multiple performance and non-performance criteria relevant to the decision-makers into a single, interpretable ranking. Additionally, relying solely on human feedback to determine rankings can be resource-intensive and may not fully reflect the practical, application-specific requirements.

Our contribution: To address these limitations, this paper presents xLLMBench, a transparent, decision-centric benchmarking framework that enables decision-makers to rank LLMs based on their own preferences across diverse, potentially conflicting, performance and non-performance criteria. It presents a set of benchmarking scenarios, together with guidelines on how decision-maker preferences can be incorporated within the ranking process and how outcomes can be interpreted to reveal each LLM's strengths and weaknesses beyond its aggregate score. The selection of criteria is entirely determined by the decision-maker, allowing the approach to remain highly flexible and adaptable across diverse contexts. We present the framework using (non-)performance criteria obtained from publicly available sources [17,18], so we do not introduce new metrics, nor provide guidelines on what should be measured during LLM benchmarking. This study specifically targets the last benchmarking step (B5 - robust analysis of benchmarking results), with the assumption that the selection of criteria (metrics), datasets, and LLMs (previous steps B1-B4) has been conducted following established benchmarking best practices. By leveraging a multi-criteria decision-making (MCDM) method, Preference Ranking Organization Method for Enrichment Evaluations (PROMETHEE) II, xLLMBench offers decision-makers the flexibility to tailor a benchmarking pipeline specific to their own application-specific requirements and priorities. We justify our choice of the PROMETHEE II algorithm, analyze correlations among performance and non-performance metrics, and validate the framework in two experimental benchmarking contexts: (1) combining LLM results for one performance metric across different datasets, and (2) combining LLM results for multiple evaluation metrics on a single dataset. Extensive sensitivity analyses of different method parameters and preference functions were performed, and the method was compared with other established MCDM methods. Diverse visualizations were provided to enhance the transparency of the method. In both scenarios, the results show that changing preferences across criteria can significantly reshape the rankings for some LLMs, while keeping them consistent for others, showing LLMs' unique characteristics. The sensitivity analyses reveal the robustness of the selected method. xLLMBench can be seamlessly used with any existing LLM benchmarking platform, helping provide contextually relevant, reproducible, and actionable LLM rankings. For reproducibility, the source code is available on GitHub [19]. Further details on experiment reproducibility and extensibility of the framework are available in Appendix A.

Outline: The paper is organized as follows. Section II reviews current LLM benchmarking datasets and platforms, as well as MCDM methods. Section III details the method used to run the benchmarking scenarios. Section IV describes the experimental setup. Section V presents the results, while Section VI discusses them. Section VII concludes the paper with a summary of the key findings.

2. Related work

This section begins with a brief overview of benchmark datasets commonly used in LLM evaluation. Then it describes existing benchmarking toolkits that integrate multiple datasets that are used to assess a larger set of LLM capabilities. Finally, it provides an overview of commonly used MCDM methods and their applications in LLM benchmarking.

2.1. LLM benchmark datasets

In general, the benchmark datasets used to evaluate LLMs can be categorized into static and dynamic. Static benchmark datasets generally consist of question-answering (QA) tasks with predefined answers and test cases, covering domains such as mathematics, coding, reasoning, and general knowledge. However, common challenges related to static datasets include dataset contamination, model overfitting, and misalignment with human perspectives. Several examples include the *Massive Multitask Language Understanding (MMLU) dataset* [20] that evaluates knowledge in 57 subjects at different difficulty levels, *GSM-8K* for grade-school math reasoning [21], *E2E* for end-to-end natural language generation [22], and *HumanEval* for code generation [23].

Recent extensions of static benchmarks aim to address the aforementioned limitations. *MMLU-Pro* [24], based on the MMLU dataset,

[1] https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard/discussions/1135

increases problem difficulty by using 10-option multiple-choice questions instead of four, requiring robust chain-of-thought reasoning to achieve high accuracy. *AGIEval* [25] evaluates general cognitive capabilities using approximately 8,000 questions drawn from human exams such as SAT, LSAT, and the Chinese college entrance exam (Gaokao), testing both knowledge and advanced reasoning. *Graduate-Level Google-Proof QA Benchmark (GPQA)* [26] targets PhD-level questions in physics, chemistry, and biology, written by domain experts and designed to be "Google-proof" (difficult for highly skilled non-experts with unrestricted access to the web), while *ProcBench* [27] evaluates multi-step procedural reasoning by testing whether models can follow explicit step-by-step instructions, analyzing both intermediate (step-wise accuracy) and final outputs. In mathematics, while LLMs perform well on datasets like GSM-8K, the *MATH* [20] benchmark remains a significant challenge, including competition-level problems that require multi-step algebraic derivations and formal reasoning decomposition.

Beyond QA with predefined answers, some benchmarks include open-ended questions assessed by human evaluation, such as *MT-bench* [28], *AlphaEval*, and *Chatbot Arena* [29], which rely on crowdsourcing to assess helpfulness, correctness, and user preference alignment, addressing limitations of purely automated metrics.

Dynamic benchmark datasets introduce continually refreshed content to reduce overfitting and memorization risks. *DynaBench* [30] and *LiveBench* [31] release contamination-free instances regularly to maintain the validity of the evaluation. In coding, *DyCodeEval* [32] uses metamorphic testing to dynamically generate paraphrased problem variations, revealing generalization gaps where models often fail on minor rephrasings. *Prism* [33] goes further by using Monte Carlo Tree Search to generate maximally challenging code problems adaptively, co-evolving with model capabilities to identify brittleness and performance ceilings. Such dynamic approaches address static dataset limitations, ensuring that evaluations remain meaningful as LLM performance saturates.

2.2. LLM benchmark toolkits

Various toolkits support LLM benchmarking by integrating benchmark datasets, evaluation metrics, and analysis/ranking methods. The *Beyond the Imitation Game benchmark (BIG-bench)* [34] consists of over 200 diverse tasks spanning linguistics, biology, math, physics, commonsense reasoning, social bias, and software development, intentionally including problems beyond current LLM capabilities to drive progress. *BIG-bench Hard (BBH)* [35] targets complex multi-hop reasoning tasks such as logical puzzles, causal inference, and creative problem solving. It includes 23 challenging tasks from BIG-Bench, in which the average human rater is not yet outperformed by LLMs. *LiveBench* [31] is another such toolkit featuring frequently updated questions from recent sources to ensure contamination-free and current LLM evaluation. Questions come from newly-released math competitions, arXiv papers, and similar sources, as well as tasks from well-known benchmarks in the field, ensuring they are contamination-free.

The *Holistic Evaluation of Language Models (HELM)* [17] benchmarks LLMs across seven metrics, including accuracy, calibration, robustness, fairness, bias, toxicity, and efficiency. The metrics vary by learning scenarios, such as QA, information retrieval, summarization, sentiment analysis, toxicity detection, and miscellaneous text classification, with results presented separately for each scenario and metric, along with correlation analysis between metric pairs. *Hugging Face* offers benchmarking toolkits with two well-known leaderboards in the field. Version 1 ranks LLMs by average accuracy across benchmarks, while Version 2 [18] normalizes scores relative to a baseline and a maximum for fairer ranking. Version 2 integrates benchmarks such as MMLU-Pro [24], GPQA [26], Multistep Soft Reasoning (MuSR) [36], MATH [20], Instruction-Following Eval (IFEval) [37], and BBH [35] to address limitations of static evaluations.

CEBench [38] introduces a benchmarking toolkit that balances effectiveness with cost considerations, employing multi-objective optimization strategies to enhance performance while minimizing inference costs. *Chatbot Arena* [29] provides an open evaluation platform where human users compare responses of two anonymized models on the same prompt. Results are aggregated via the Bradley-Terry model [39] to derive rankings. Prompt diversity is ensured by clustering prompts using BERTopic [40] to sample varied topics. *AutoArena* [41] follows a similar approach but replaces human votes with multi-agent LLM debates and internal voting, yielding rankings highly correlated with human judgments while reducing annotation costs. *Benchmarking Agreement Testing (BAT)* [10] analyzed over 40 benchmark datasets with multiple LLMs, exploring performance correlations and revealing that benchmark rankings can depend heavily on the chosen model portfolio or evaluation metric, potentially introducing bias.

2.3. MCDM methods

MCDM proposes systematic methods that allow decision-makers to weigh different alternatives using a range of (often conflicting) criteria of different type (e.g., numerical computations, qualitative judgments), aiming to give decision-makers a range of possible solutions to the problem instead of one best solution [42]. MCDM commonly involves several general steps, (M1) defining the decision problem, objectives, criteria, and constraints, (M2) selecting a method to weight criteria based on relative importance, (M3) identifying available alternatives, (M4) evaluating the alternatives against the criteria with a performance score, (M5) selecting a method to aggregate criteria scores into a single one, (M6) sensitivity analysis of the criteria weight changes, and (M7) selecting one or multiple highest scoring alternatives [42]. Some of those steps overlap with the benchmarking steps B1-B4, for which this paper assumes are already robustly done in benchmarking platforms.

On the other hand, there is no single best MCDM method that fits all use cases, but different methods may yield different rankings of alternatives, which is even true for different parameter configurations of the same method [43]. For that reason, the application of MCDM methods to LLM ranking is not a straightforward task but a non-trivial problem that has to be explored through an extensive set of experiments. The research literature suggests that MCDM methods can be categorized across several different criteria. The PROMETHEE methods [44] belong to the category of outranking methods, with PROMETHEE I allowing partial ranking of a finite set of alternatives and PROMETHEE II allowing their complete ranking. Several other alternatives have been proposed in the research literature [45]. PROMETHEE appears as a particularly convenient method for LLM ranking as it allows specifying decision-makers' preferences for each different criterion at two levels, through the alternative preference function and criteria weights. Additionally, it enables specification of the decision-maker's indifference or preference towards certain thresholds of alternatives pairwise differences. This makes the method highly configurable to different use cases when the decision-makers clearly know their preferences, but at the same time, offering automated algorithms for selecting certain parameters when it is not the case. However, different parameter configurations can lead to different rankings of alternatives [43], so sensitivity analyses are essential [42]. Furthermore, MCDM methods can be difficult for non-expert decision-makers to understand, particularly when the number of alternatives and/or criteria is large, so approaches that enhance the interpretability of the decision process are crucial.

Our literature review showed that MCDM methods are not commonly used in solving the LLM selection problem. At the time of writing, only one paper [46] proposed a fuzzy analytical hierarchy process to calculate weights for a set of expert-specified criteria and associated metrics relevant to selecting an LLM for the healthcare domain. The paper identified and weighted nine quantitative criteria, i.e., robustness, reliability, bias/fairness, performance, availability, resilience, usability, predictability, and cost, together with 12 sub-criteria.

The main difference between our paper and those referenced in Sections 2.1 and 2.2 is that it does not provide a new benchmark dataset with new prompt instances, nor does it focus on providing dynamically evolving instances, since it does not target the first step of the benchmarking process (B1) that is concerned with the quality of the data. Instead, it goes beyond those studies by fusing different performance and non-performance metrics into a single robust and interpretable ranking, rather than providing simple descriptive statistics for each metric separately. Namely, it targets the last step of the benchmarking process (B5 - robust analysis of the benchmarking results). Furthermore, the method proposed in this paper can be integrated with all existing benchmarks, providing decision-makers with the flexibility to tailor an LLM benchmarking pipeline to their objectives. To the extent of our knowledge, this is the first systematic study of the applicability of MCDM methods to the B5 step of the LLM benchmarking process, utilizing data from well-known LLM benchmarking platforms, thoroughly analyzing the results from a large set of benchmarking scenarios, and performing sensitivity analysis across different MCDM methods and method parameters.

3. Method

This section presents two common LLM benchmarking contexts in which decision-makers must select LLMs suitable for their use case. The first involves LLM performance across multiple benchmark datasets as criteria, while the second considers LLM performance calculated with multiple different performance metrics on the same benchmark dataset. Although we analyze two example contexts and several scenarios in each, many others tailored to specific decision-maker requirements can be easily configured and analyzed using the same approach and published source code on GitHub (for additional details see Appendix A). The section also describes our use of PROMETHEE II for this purpose. We note here that PROMETHEE II was selected as the MCDM method because it provides greater flexibility in modeling preferences, allowing them to be expressed through (1) preference functions, (2) weights, or (3) both (i.e., preference functions and weights), for each separate criterion. However, we also compare the results with two other widely used MCDM methods, Technique for Order Preference by Similarity to an Ideal Solution (TOPSIS) [47] and VlseKriterijumska Optimizacija I Kompromisno Resenje (VIKOR) [48]. The results of the comparison, presented in Appendix C, demonstrate that the resulting rankings are stable and robust.

The general flowchart of the proposed framework is presented in Fig. 1. The process starts by obtaining tabular LLM benchmarking data as input, which consists of a selected portfolio of LLMs (each in a separate row) and (non-)performance criteria measurements (each in a separate column). A complete dataset without missing values is required. The next step involves configuring the benchmarking context by specifying, among the rest, the preference functions and weights for each criterion in a predefined format (refer to the GitHub repository for the exact format). The framework then analyzes the data and outputs a ranked list of LLMs, together with additional data and visualizations to help decision-makers better interpret the resulting rankings.

3.1. LLM benchmarking contexts

Multiple benchmark datasets. Typically, LLM benchmarking is performed on a set of datasets using a single performance metric, often reporting the average performance of LLMs across all datasets. However, decision-makers may instead want to choose to weight datasets' importance differently, in a way that closely matches their application-specific requirements. In this benchmarking context, performance calculated using a single performance metric is aggregated across datasets with an MCDM method, placing greater emphasis on datasets relevant to the use case.

Multiple performance metrics. Trustworthy Artificial Intelligence (AI) principles require evaluating many different aspects of ethics and trustworthiness in AI models. For example, the Ethics Guidelines for Trustworthy AI of the High-Level Expert Group on AI set up by the European Commission [49] define seven key requirements from a “trustworthy” AI systems, i.e., (1) human agency and oversight, (2) technical robustness and safety, (3) privacy and data governance, (4) transparency, (5) diversity, non-discrimination and fairness, (6) environmental and societal well-being, and (7) accountability. Such requirements impose the use of a range of different metrics instead of a single one, to evaluate a wide range of capabilities beyond technical robustness (overview of quality attributes relevant to ML systems, metrics, and attribute trade-offs is available in [15]). For those reasons, the second LLM benchmarking context enables LLM comparison across different metrics such as accuracy, effectiveness, fairness, bias, resource utilization and etc., calculated on a single benchmark dataset, while allowing decision-makers to specify their application-specific preferences on each metric before aggregating them into a single ranking.

3.2. PROMETHEE II application to LLM ranking

This section illustrates the application of PROMETHEE II in the second benchmarking context, which involves multiple performance metrics within a single dataset. In the first benchmarking context (using

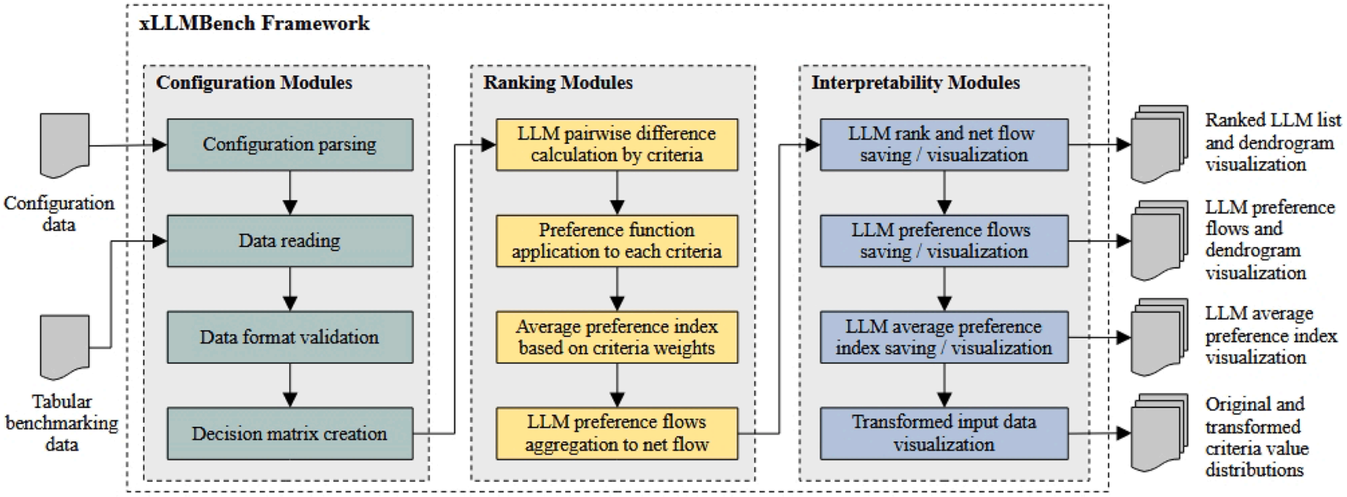

xLLMBench Framework
Configuration Modules
Configuration parsing
Data reading
Data format validation
Decision matrix creation
Ranking Modules
LLM pairwise difference calculation by criteria
Preference function application to each criteria
Average preference index based on criteria weights
LLM preference flows aggregation to net flow
Interpretability Modules
LLM rank and net flow saving / visualization
LLM preference flows saving / visualization
LLM average preference index saving / visualization
Transformed input data visualization
Configuration data
Tabular benchmarking data
Ranked LLM list and dendrogram visualization
LLM preference flows and dendrogram visualization
LLM average preference index visualization
Original and transformed criteria value distributions

Fig. 1. xLLMBench flowchart consisting of modules conceptually organized in three groups, (i) configuration modules reading the configuration data (i.e., criteria preference functions and weights) and LLM benchmarking data provided by the decision-maker as input, (ii) ranking modules implementing LLM ranking specifics, and (iii) interpretability modules outputting a ranked LLM list, accompanied with data and visualizations for improved result interpretability.

Table 1
Alternative/criteria matrix.

A/P	p_1	p_2	…	p_n
LLM_1	$p_1(LLM_1)$	$p_2(LLM_1)$	…	$p_n(LLM_1)$
LLM_2	$p_1(LLM_2)$	$p_2(LLM_2)$	…	$p_n(LLM_2)$
$\vdots$	$\vdots$	$\vdots$	$\vdots$	
LLM_m	$p_1(LLM_m)$	$p_2(LLM_m)$	…	$p_n(LLM_m)$

Table 2
A brief description of the LLM benchmark datasets used in this paper.

Dataset	Description
MMLU-Pro	Dataset extending the MMLU benchmark [20] with more challenging, reasoning-focused questions, expanding the number of choices from 4 to 10, and eliminating the trivial/noisy questions. [24]
GPQA	Highly difficult multiple-choice question dataset created by experts in biology, physics, and chemistry (even experts having/pursuing a PhD in the fields reach an accuracy of 65%). [26]
MuSR	Algorithmically created dataset of multi-step reasoning tasks written in a natural language, which require reasoning over a long context. [36]
MATH	High-school level math competition problems [20]. Only level 5 math questions are included in the Open LLM Leaderboard.
IFEval	Standardized benchmark for evaluation of LLM natural language instruction following capabilities. [37]
BBH	A subset of 23 most challenging tasks from Big-Bench [34] on which language models had not yet surpassed average human-rater performance at the time of its creation. [35]
NarrativeQA	QA benchmark for reading comprehension over long stories (e.g., book or movie script). [51]
NaturalQuestions	Question answering benchmark of naturally-occurring queries. In the open-book version, the input includes the Wikipedia page with the answer, while in the closed-book it does not. [17,52]

a single performance metric across multiple datasets), the same logic applies as only the meaning of the criteria changes, i.e., performance metrics from the second context are replaced by datasets in the first.

Let $A = \{LLM_1, LLM_2, \dots, LLM_m\}$ represent a portfolio of m selected LLMs (alternatives in MCDM terminology). Our objective is to compare them using a set of performance metrics (criteria in MCDM) $P = \{p_1, p_2, \dots, p_n\}$ chosen by the decision-maker (we use the term for consistency with the MCDM terminology), where n denotes the number of such metrics. Let us define an $m \times n$ matrix (see Table 1) that contains the values of performance metrics obtained for each LLM individually. Each row corresponds to an LLM, while each column corresponds to a different performance metric calculated within a single benchmark dataset.

To compare the LLMs on the selected set of performance metrics while allowing decision-makers to specify their preferences, the PROMETHEE II method is utilized. PROMETHEE methods aid decision-making by evaluating alternatives based on often conflicting criteria. In our case, the ensemble heuristic leverages pairwise comparisons of LLMs for each performance metric. A key advantage is that decision-makers can set preferences for each metric by defining a preference function, indicating the degree of preference of one LLM (LLM_1) over another (LLM_2). A preference function for the j performance metric, P_j, is defined as given in Eq. (1), where $d_j(LLM_1, LLM_2) = p_j(LLM_1) - p_j(LLM_2)$ is the difference in values for the j-th performance metric and $q_j(\cdot)$ is a user-assigned preference function for the metric. The user can choose from six generalized preference functions [50], i.e., usual criterion, quasi criterion, linear preference, level preference, linear preference with indifference area, and Gaussian preference. Additionally, the method allows decision-makers to define their own generalized preference function.

$$P_j(LLM_1, LLM_2) = \begin{cases} q_j(d_j(LLM_1, LLM_2)), \text{subj. to max. of} p_j \\ q_j(-d_j(LLM_1, LLM_2)), \text{subj. to min. of} p_j \end{cases} \quad (1)$$

Using the preference function p_j, decision-makers can assess the comparison between two LLMs based on a single criterion. To compare the LLMs across all criteria, the average preference index should then be calculated as given in Eq. (2). Here, w_j denotes the weight of the $j - th$ criterion, with higher weights indicating greater importance in ranking. Weights can be set manually (providing another chance to incorporate decision-makers' preferences over the criteria) or determined empirically using predefined weighting functions. Note that the average preference index is asymmetric, i.e., $\pi(LLM_1, LLM_2) \neq \pi(LLM_2, LLM_1)$.

$$\pi(LLM_1, LLM_2) = \frac{1}{n} \sum_{j=1}^{n} w_j p_j(LLM_1, LLM_2). \quad (2)$$

The average preference index compares two LLMs across all performance metrics. To compare all LLMs, we need to calculate the net flow, the difference between the positive ($\phi(LLM_i^+)$) and negative ($\phi(LLM_i^-)$) preference flows. The positive flow shows how an LLM outperforms the other LLMs from the portfolio, while the negative indicates how it is outperformed by the other LLMs. Their definition is given with Eq. (3). The net flow of an LLM is its difference, given with Eq. (4). PROMETHEE II ranks the LLMs by ordering them based on decreasing net flow values. Instead of using multiple performance metrics, the same analysis can be done with a single metric by applying it to various benchmark datasets, allowing preferences to be assigned at the dataset level.

$$\phi(LLM_i^+) = \frac{1}{(m-1)} \sum_{x \in A} \pi(LLM_i, x),$$
$$\phi(LLM_i^-) = \frac{1}{(m-1)} \sum_{x \in A} \pi(x, LLM_i). \quad (3)$$

$$\phi(LLM_i) = \phi(LLM_i^+) - \phi(LLM_i^-). \quad (4)$$

4. Experimental design

A brief description of the datasets used in our two experimental contexts defined in Section 3.1, is given in Table 2. To demonstrate the advantages of incorporating practices from MCDM in LLM ranking and their scalability, in each context, we define multiple scenarios aligned with different decision-maker requirements, as described below. We also illustrate how the output of those scenarios can be interpreted. We select performance criteria that are available from publicly available sources [17,18]. All performance metrics used are for the inference phase and not for the LLM fine-tuning phase. All used performance metrics have been reported based on cross-validation settings consistent across all LLMs, ensuring that the metrics are derived from the same train and test splits and aggregated in the same way. We would like to point out that evaluating LLMs on different datasets is beyond the scope of this paper, since the selected publicly available benchmarking data are sufficient to show how the proposed framework works.

4.1. Multiple benchmark datasets

In the first experimental context, we use public data from Hugging Face Open LLM Leaderboard (version 2)[2], which includes at the time of writing 4,576 open LLMs evaluated on six benchmarks (i.e., datasets). However, at the time of writing, the benchmarking platform has already been archived, the reason for which has been announced in its authors' post[3]. Each benchmark is associated with a single performance metric, accuracy or exact match[4], subject to maximization. We use the 1,000 LLMs ranked highest based on the average performance across

[2] https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard

[3] https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard/discussions/1135

[4] https://huggingface.co/docs/leaderboards/open_llm_leaderboard/about

the six benchmarks, as calculated in the platform. The missing data handling strategy was not required, since the selected subset did not contain any missing data. We define several scenarios (S) in which the decision-makers' preferences prioritize certain benchmarks based on their domain or characteristics.

In the first set of scenarios, we rank LLMs using different preference functions applied to the six benchmarks to account for the differences in the LLMs' performance. **S1** applies the usual preference function to each benchmark dataset and assigns equal weights to all datasets, i.e., treats them as equally important. **S2** applies the linear preference function, while **S3** the Gaussian in the same way as S1. To emphasize LLMs' performance on certain benchmarks more than on others, we create a new controlled scenario by selecting different preference functions for different benchmarks. Therefore, **S4** aims to prioritize datasets with minimal contamination, i.e., gives greater importance to GPQA, MuSR, and MMLU-Pro datasets [16] through the Gaussian preference function. The linear preference function is applied to the remaining three datasets. Equal weights are assigned to all datasets so that the importance is emphasized only through the preference function. Additionally, we demonstrate that preferences can be modeled by assigning different weights (importance) to different benchmarks. For that purpose, **S5** repeats S1, S2, and S3, but assigns different weights to the GPQA, MuSR, and MMLU-Pro datasets. The same preference function is applied to all datasets, since now the preference is emphasized through the assigned weights. This results in scenarios S5.1, S5.2, and S5.3, respectively. We then describe a scenario that lets users balance LLM parameter size with accuracy. In scenario **S6**, we add the number of parameters as a seventh criterion, applying a Gaussian preference function to it, while applying a linear preference function to each of the other six benchmark accuracy metrics. All seven criteria receive equal weight; LLM parameter size is set as a minimization objective (favoring smaller models), and the six accuracy metrics as maximization objectives (favoring higher scores). This configuration promotes compact models that still perform well across every benchmark. Finally, we present a scenario where decision-makers can set preferences on the trade-off between the CO_2 cost (CO_2 emissions during model evaluation in kg)[5] and its accuracy across different benchmarks. **S7** incorporates the CO_2 cost as an additional criterion, prioritizing it with the Gaussian preference function. Again, the linear preference function is applied to the remaining six benchmark datasets, and equal weight is assigned to all seven criteria. The CO_2 cost is subject to minimization, while the other six metrics are subject to maximization. This scenario should favor models with lower CO_2 emissions that perform well across all benchmarks.

In summary, S1 aggregates the wins and losses of the LLM pairs in all benchmarks. S2 linearly emphasizes the accuracy differences between LLM pairs. S3 prioritizes larger pairwise differences as more significant wins. S4 emphasizes larger pairwise differences in contamination-free datasets using a Gaussian preference function, while applying a linear preference function to the remaining datasets. S5.1, S5.2, and S5.3 are modified versions of S1, S2, and S3 by assigning higher weights to contamination-free benchmarks. S6 favors LLMs with a smaller parameter size and strong performance in all six benchmarks, while S7 favors LLMs with lower CO_2 emissions and good performance. In all scenarios 1 to 7, we compare the resulting rankings with those based on the average score reported on the Open LLM Leaderboard website. This "average-based" ranking uses the mean of the normalized scores from Open LLM Leaderboard[6], which may be skewed if a model performs inconsistently across the benchmark datasets.

4.2. Multiple performance metrics

In the second set of experiments, we use the Question Answering results from HELM Classic Leaderboard v0.4.0 (2023-11-17)[7], which was the latest release available at the time of our analysis. Three of the nine datasets were selected, i.e., NaturalQuestions (open-book), NaturalQuestions (closed-book), and NarrativeQA, excluding those with missing performance values. We compare LLMs using seven metrics. Those are the F1 scores from the accuracy, fairness, and robustness categories (subject to maximization), where the second refers to the worst case over fairness-related word perturbations and the third to the worst case over robustness-related word perturbations. We also use the toxic fraction (fraction of toxic outputs) from the toxicity category (subject to minimization). Finally, we use metrics related to bias, specifically, stereotypical associations related to gender groups with target professions (based on co-occurrence statistics of gender terms and professions) and potentially uneven representation of racial and gender groups based on the frequency of racially-associated names and gender-related terms (subject to minimization). Please refer to [17] for additional details on the metrics. Since the selected data subset still contained a small amount of missing values, we selected the list-wise deletion strategy, in which the alternatives that contain at least one missing value across datasets and metrics were deleted from the dataset. This resulted in a subset of 49 LLMs used in the subsequent analysis, out of 67 available in the leaderboard, i.e., 73%. In the scenarios, the decision-maker's preferences are assigned to different performance metrics calculated on the same benchmark dataset.

We evaluate several scenarios (S) on each of the three datasets separately. In the first set of scenarios, the LLMs are ranked over the selected metrics using the usual, linear, and Gaussian preference functions. **S1** uses the usual preference function for each of the seven metrics, treating all metrics as equally important by assigning them equal weights. **S2** follows the same approach but uses the linear preference function. **S3** uses the Gaussian. Then we analyze the impact of the use of correlated metrics on the ranking. Specifically, in **S4** we conduct a correlation analysis of the seven performance metrics and select one metric from each cluster of correlated metrics. The Pearson correlation coefficient is used to calculate the correlation between the performance values of all criteria pairs, followed by clustering to identify correlated criteria. This approach helps to avoid biased results in favor of certain LLMs, which can happen when highly correlated metrics are included. Then, we repeated scenarios S1, S2, and S3 on the selected set of uncorrelated metrics, resulting in S4.1 (usual), S4.2 (linear), and S4.3 (Gaussian), respectively. Additionally, we repeated the S4.1-3 scenarios k times, each repetition incorporating different metrics from the set of correlated metrics, in order to examine their impact on the final rankings. In **S5**, using the set of uncorrelated metrics, we repeat S4 by applying the Gaussian preference function to the bias-related metrics and the linear preference function to all other, i.e., giving higher preference to bias-related metrics to favor unbiased LLMs. We would like to emphasize that this study neither proposes new bias/fairness metrics nor provides guidelines for their measurement, since that is part of the benchmarking step B4. It only utilizes metrics from publicly available benchmarking platforms and assigns them a higher priority during the benchmarking step B5. All experiments are conducted on each benchmark dataset separately.

4.3. Generalized preference functions

We evaluate three generalized preference functions mentioned above when describing the experiments, i.e., the usual, the linear, and the Gaussian, the calculation of which is given in Eq. 5–7, respectively. The linear preference function is parameterized by the indifference a and preference b thresholds, below/above which the difference is

[5] https://huggingface.co/docs/leaderboards/open_llm_leaderboard/emissions

[6] https://huggingface.co/docs/leaderboards/open_llm_leaderboard/normalization

[7] https://crfm.stanford.edu/helm/classic/latest/#/leaderboard

considered negligible/significant and projected to 0/1, respectively. All pairwise differences between a and b are linearly projected between 0 and 1. These thresholds are set to the minimal and maximal value in the metrics pairwise difference matrices in our experiments, i.e., we have no indifference/preference towards specific pairwise differences. The Gaussian preference function is parameterized by the standard deviation σ, which is set to the standard deviation of the pairwise distance matrices. We also perform a sensitivity analysis on the parameters of the linear (a, b) and Gaussian (σ) preference functions. These parameters are increased and decreased by 10% and 20%, where applicable, and all experiments involving these preference functions are recalculated with the updated parameters. The details of the sensitivity analysis and its results are available in Appendix B. The resulting rankings remain robust and stable across all experiments.

$$Usual(x) = \begin{cases} 0, & x \leq 0 \\ 1, & x > 0 \end{cases} \tag{5}$$

$$Linear(x) = \begin{cases} 0, & x \leq a \\ \frac{x-a}{b-a}, & a < x \leq b \\ 1, & x > b \end{cases} \tag{6}$$

$$Gaussian(x) = \begin{cases} 0, & x \leq 0 \\ 1 - \exp\left(-\frac{x^2}{2\sigma^2}\right), & x > 0 \end{cases} \tag{7}$$

4.4. Weighting method

According to [43], the three most popular methods for objective criteria weighting include, (1) assigning equal weights to the criteria, (2) the entropy method taking into consideration the uncertainty of the information, and (3) the standard deviation method which assigns small weights to the criteria with low standard deviation between their values. In this paper, we allow manual specification of criteria weights, and most of the scenarios use the equal weighting method where the n criteria weights are calculated as given in Eq. (8).

$$w_j = 1/n \tag{8}$$

In other scenarios, a larger amount of the total weight is equally distributed across several (more relevant criteria), while a smaller amount of the total weight is again equally distributed among the remaining criteria.

In addition, we conduct an experiment in which the weights are calculated using the Analytic Hierarchy Process (AHP) [53], a well-known MCDM method. These weights are then integrated into the PROMETHEE II method. This approach is adopted because setting weights manually can be a challenging task for non-experts, while the AHP method offers a more user-friendly way to specify them. The description and results of this experiment are presented in Appendix D. The rankings obtained using AHP weights are consistent with those obtained with manually specified weights.

5. Results

The LLM's ranking results are shown separately for the two experimental contexts, the first involving multiple benchmark datasets and the second multiple performance metrics. The results can be reproduced using the source code from our GitHub repository [19], after obtaining the previously described input data. For more details, see Appendix A.

5.1. Multiple benchmark datasets

Fig. 2 presents LLM rankings across different scenarios (described in Section 4.1), where each row represents an LLM and each column shows its ranking in different scenarios. The first column gives the LLM ranking based on Hugging Face's average score. In the figure, the LLMs are uniquely described through their name, number of parameters, and Hugging Face's average score. The portfolio shown in the figure includes the union of the 10 LLMs with the highest ranking from each scenario (resulting in 23 LLMs). The figure also includes a dendrogram of the selected LLMs, clustered using Euclidean distance and average linkage based on their rankings across different scenarios. This figure indicates that several LLMs maintain stable rankings with only minor deviations across different scenarios. The top four LLMs (based on the average column) exemplify this stability. In contrast, other LLM clusters show varying rankings depending on the scenario, i.e., decision-makers' preferences influence their ranking. The second cluster of LLMs presented on the bottom of the heatmap shows LLMs that perform poorly in scenarios S6-S7, compared to their average scores (e.g., 5th, 6th, 7th, and 8th ranked LLMs), emphasizing that the model size and the CO_2 emissions are large (we preferred small models with smaller CO_2 emissions in those scenarios). For most LLMs, there is no significant deviation between the rankings in scenarios S1-S5.3. However, for some, those rankings deviate from their average-based ranking (e.g., 10th, 24th, 28th), while for others remain comparable (e.g., 8th, 14th, 15th, 21st).

We can conclude that the top three LLMs (based on the "average" ranking) are the best overall choices across all scenarios. Their stable performance ensures that they remain largely unaffected by the decision-maker's preferences in the benchmarking process. The 12th, 15th, 24th, 28th, 50th, and 57th-ranked LLMs improve their ranking under S3 (Gaussian preference function), indicating they achieve significantly larger wins across benchmarks. Conversely, LLMs ranked 5th through 11th, except for the 8th, worsen their ranking, suggesting that other LLMs have more substantial wins against them. Next, the 8th, 12th, 15th, 28th, and 57th-ranked LLMs based on the average score, maintain a good ranking under S4, indicating strong performance on contamination-free datasets where larger wins are preferred (through the Gaussian preference function). The 57th-ranked LLM performs much worse across the other scenarios S1-S3 and S5.1-S5.3. S4 particularly suggests that prioritizing contamination-free benchmarks does not work in favor of this model. When preferring smaller model sizes and lower CO_2 emissions, the two LLMs ranked 12th by the average score are as preferable as the 1st-4th LLM, since they rank 1st and 2nd in S6-S7 scenarios. They are also not vulnerable to contamination, as they perform better when contamination-free benchmarks are prioritized (S4), making them more competitive against other LLMs. A solid choice would be the LLMs ranked as 1st-4th, and 12th based on the "average" score, as their performance remains stable even when prioritizing the contamination-free benchmarks. To confirm these findings, Fig. 3 shows the positive and negative preference flows of all 23 LLMs shown in Fig. 2.

To improve the interpretability of the MCDM process and show how the positive and negative flows are calculated, Fig. 4 illustrates the distribution of LLM performance scores by the criteria (accuracy or exact match, subject to maximization) available on the Open LLM Leaderboard. The first 1000 LLMs by average-based ranking are included in the visualization. It is evident that certain benchmark datasets are more difficult for most of the LLMs, such as GPQA and MUSR. A lower standard deviation σ corresponds to smaller pairwise differences between LLM performance values, as illustrated in Fig. 5. It illustrates the LLM pairwise difference distribution by criteria before applying a preference function (Fig. 5 (a)) and after applying the three different ones. It can be seen that the usual preference function simply maps the differences to either 0.0 or 1.0 (Fig. 5 (b)), and the linear maps them linearly to the interval between 0.0 and 1.0 (Fig. 5 (c)). The Gaussian maps the negative values to 0.0 and for the positive uses a Gaussian distribution centered at 0.0 with a standard deviation σ equal to that of the values in the pairwise distance matrix (predefined parameter described in Section 4.3), as illustrated in Fig. 5 (d). To further clarify the differences between the preference functions, Fig. 6 illustrates the mapping between the original pairwise distances and the target values outputted by each preference function. It is clearly visible that the Gaussian preference function

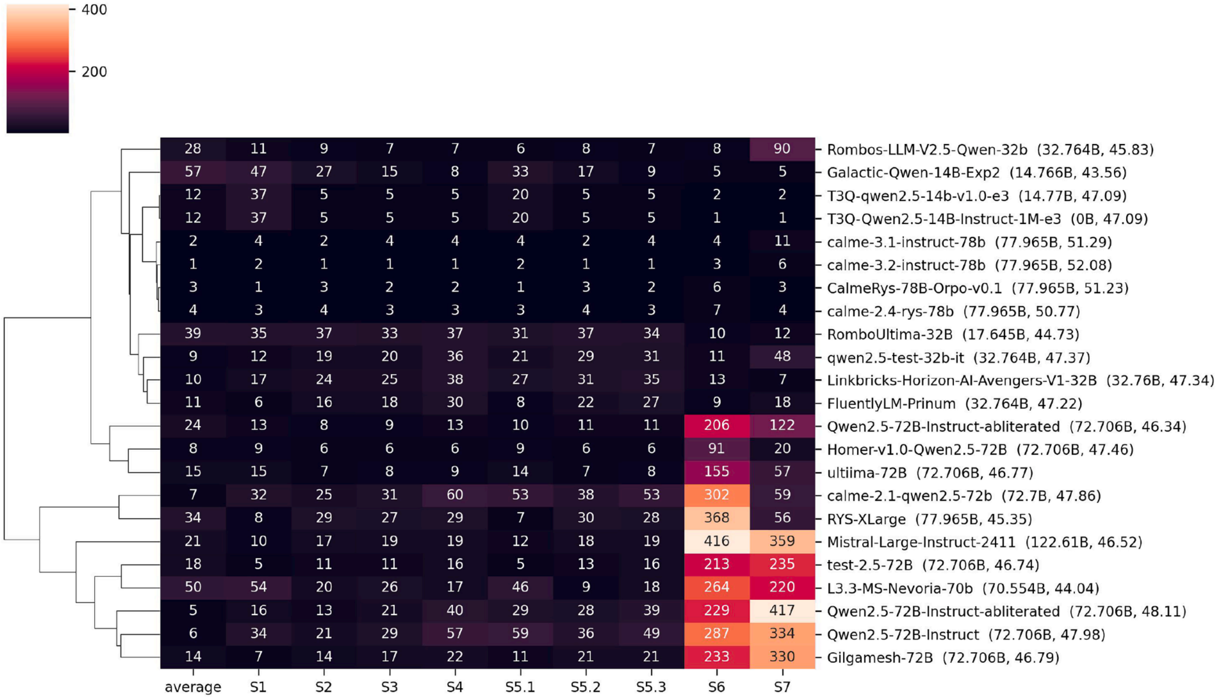


Fig. 2. Comparison of LLM rankings (rows) across scenarios (columns) on Open LLM Leaderboard dataset. The portfolio includes the union of the 10 LLMs with the highest ranking from each scenario. The first column shows the rankings based on the average Hugging Face score.

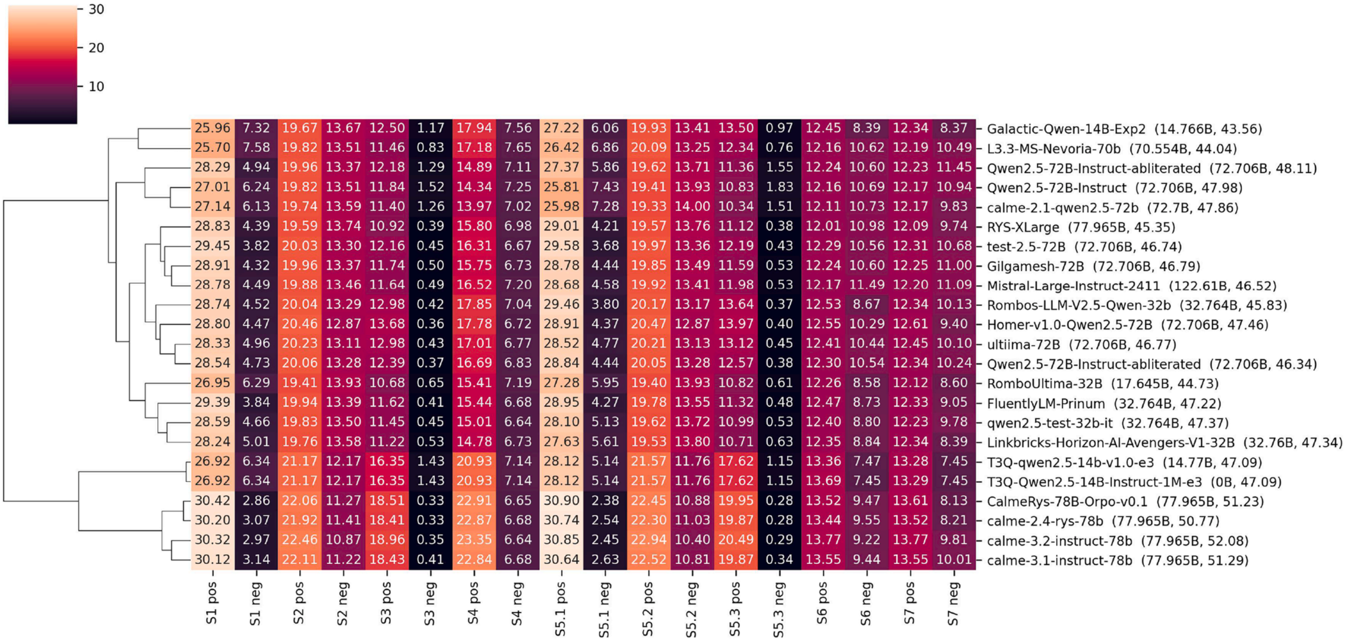


Fig. 3. Comparison of LLM positive and negative preference flows (rows) across scenarios (columns) on Open LLM Leaderboard dataset. The portfolio includes the union of the 10 LLMs with the highest ranking from each scenario (same LLMs as in Fig. 2).

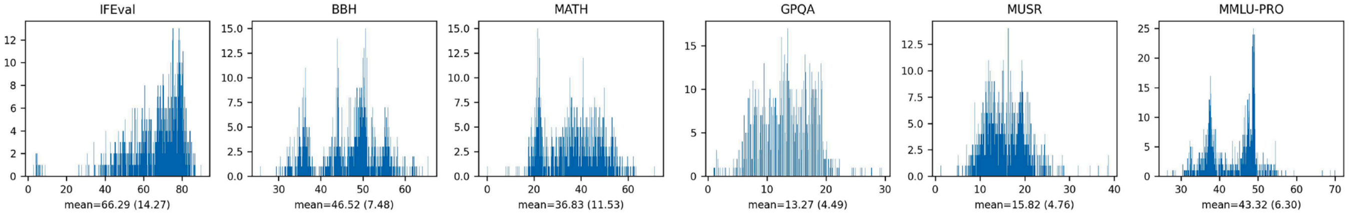
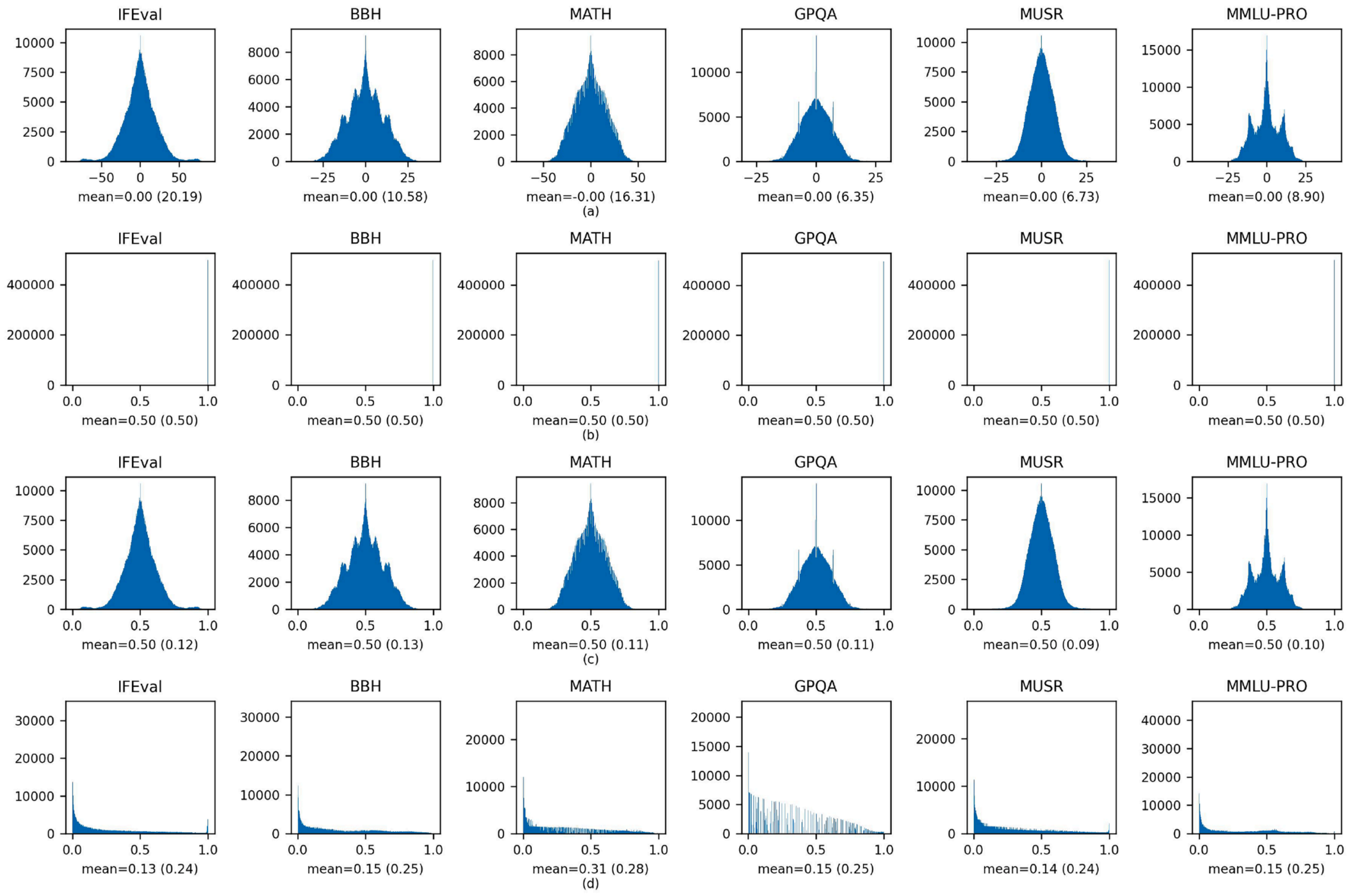


Fig. 4. LLM performance score distribution by Open LLM Leaderboard dataset.



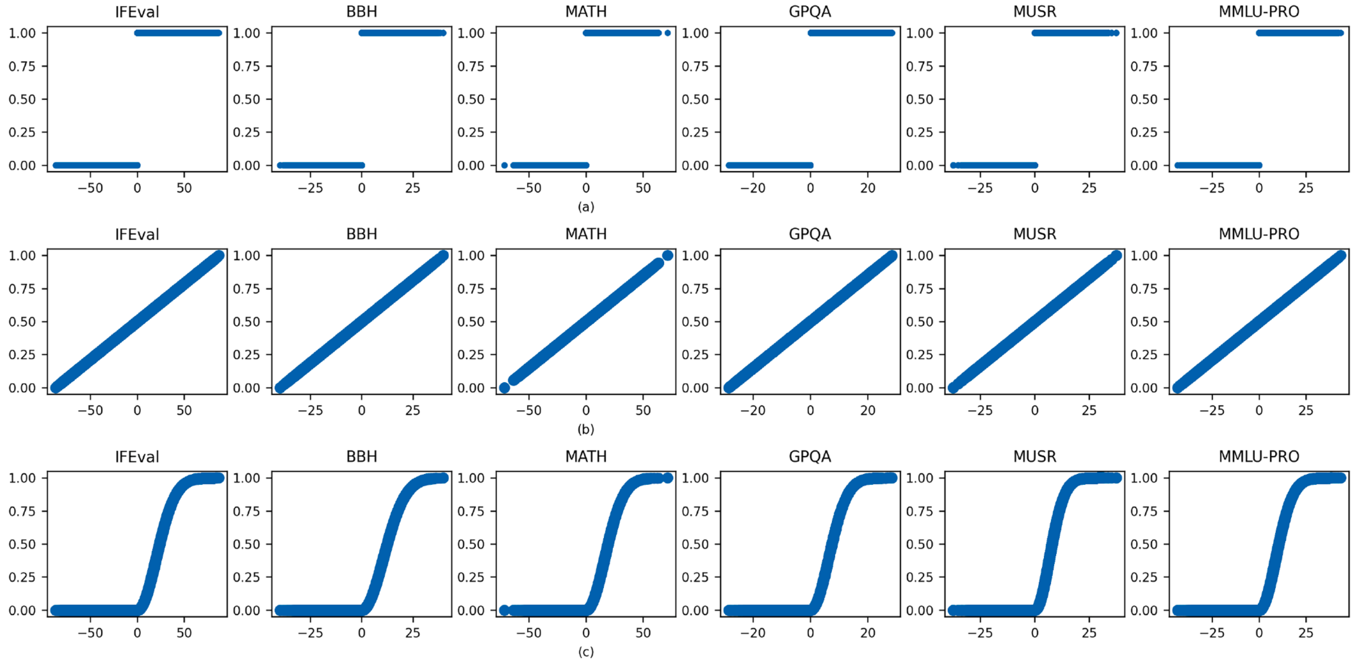
(a)

(b)

(c)

(d)

Fig. 5. LLM pairwise distance distribution by Open LLM Leaderboard dataset. (a) Before applying a preference function. (b) After applying the usual preference function. (c) After applying the linear preference function. (d) After applying the Gaussian preference function. For clarity of the visualization, the negative pairwise distances that are mapped to 0.0 with the Gaussian preference function are not shown.



(a)

(b)

(c)

Fig. 6. Mapping of the LLM pairwise distances by Open LLM Leaderboard dataset (x-axis) to target values between 0.0 and 1.0 (y-axis) with different preference functions. (a) Usual preference function. (b) Linear preference function. (c) Gaussian preference function.

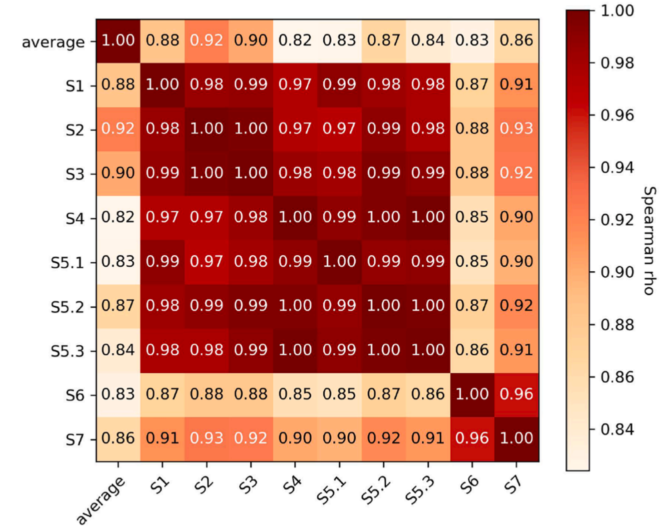


Fig. 7. Spearman rank correlation between the Open LLM Leaderboard average-based ranking and the rankings outputted in different scenarios.

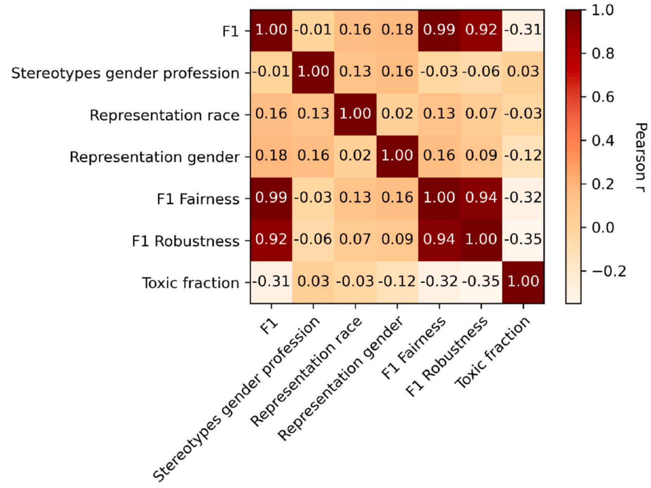


Fig. 8. Pearson correlation between LLM scores with different performance metrics from HELM, calculated separately on each of the three datasets and then averaged across the datasets.

emphasizes the extreme differences (small or large) more than the linear one. The actual performance values in Open LLM Leaderboard for a selected set of LLMs are illustrated in Appendix E and discussed in relation to the results presented in this section.

Finally, Fig. 7 presents the Spearman rank correlation between the average-based ranking in Open LLM Leaderboard and the final rankings outputted in all other scenarios. The results indicate that the rankings produced by scenarios S1-S5.3 are rather robust, as the correlation is very high. The rankings outputted in scenarios S6-S7 have higher correlation between themselves, but lower correlation with the other scenarios, which is expected due to the introduction of new criteria - parameter size and CO_2 emissions. On the other hand, the higher correlation between S6 and S7 is also expected, as LLMs with a larger number of parameters are usually associated with a higher CO_2 emissions. All scenarios have lower correlation with the average-based ranking. Scenarios S2 and S5.2 (which use the linear preference function) show a slightly increased correlation with the average-based ranking compared to the other scenarios.

Overall, we note that the rankings in S1, S2, and S3 and in S5.1, S5.2, and S5.3, respectively, are similar because the same preference function is applied to each benchmark dataset in each group, with only the weighting method differing. In S5.1, S5.2, and S5.3, it only linearly scales the preference function value accordingly. In these scenarios, the weight of the preferred criteria is 1.5 times larger than the rest (i.e., $w_{IFEval} = 0.13$, $w_{BBH} = 0.13$, $w_{MATH} = 0.13$, $w_{GPQA} = 0.20$, $w_{MUSR} = 0.20$, $w_{MMLU-PRO} = 0.20$, all weights summing to 1.0), but to emphasize the difference even more, larger weight variations would be needed (planned for future work). Additional sensitivity analysis of the weights are reported in Appendix D.

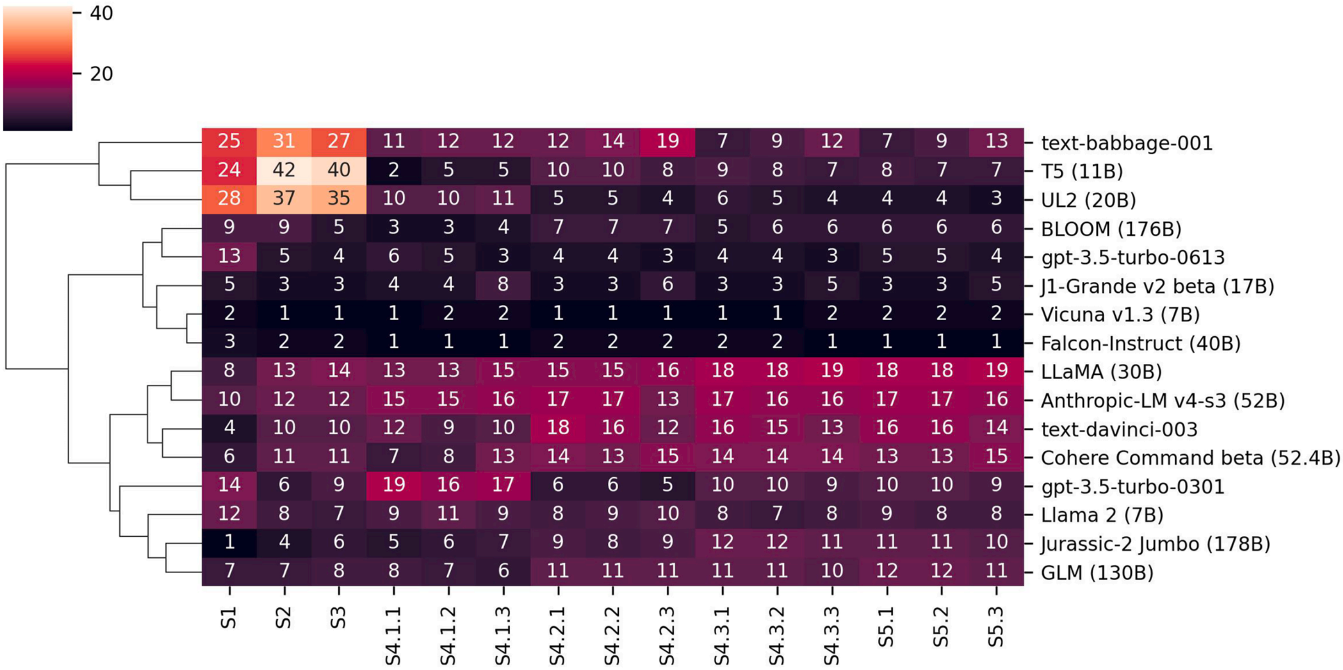
5.2. Multiple performance metrics

In the second experimental context, the rankings are based on multiple performance metrics calculated on a single dataset. We apply this approach to three benchmark datasets. In the beginning, we analyze the correlations among the metrics used.

Fig. 8 illustrates the Pearson correlation between various performance metrics, calculated separately for each dataset and then averaged across all three datasets. The results reveal that the only cluster of correlated metrics includes the F1 scores related to accuracy, fairness, and robustness. Therefore, in S4.1 (usual preference function), S4.2 (linear), and S4.3 (Gaussian), we repeat the experiments three times, each time selecting one metric from the F1 scores along with all other remaining metrics to assess their influence. These variations are denoted as S4.1.1 (F1 retained), S4.1.2 (F1 fairness retained), and S4.1.3 (F1 robustness retained). It is also important to note that we did not include the “mean win rate” metric from HELM, as it is calculated by metric group (i.e., across all benchmark datasets), whereas our analyses are conducted by dataset (i.e., across different metrics).

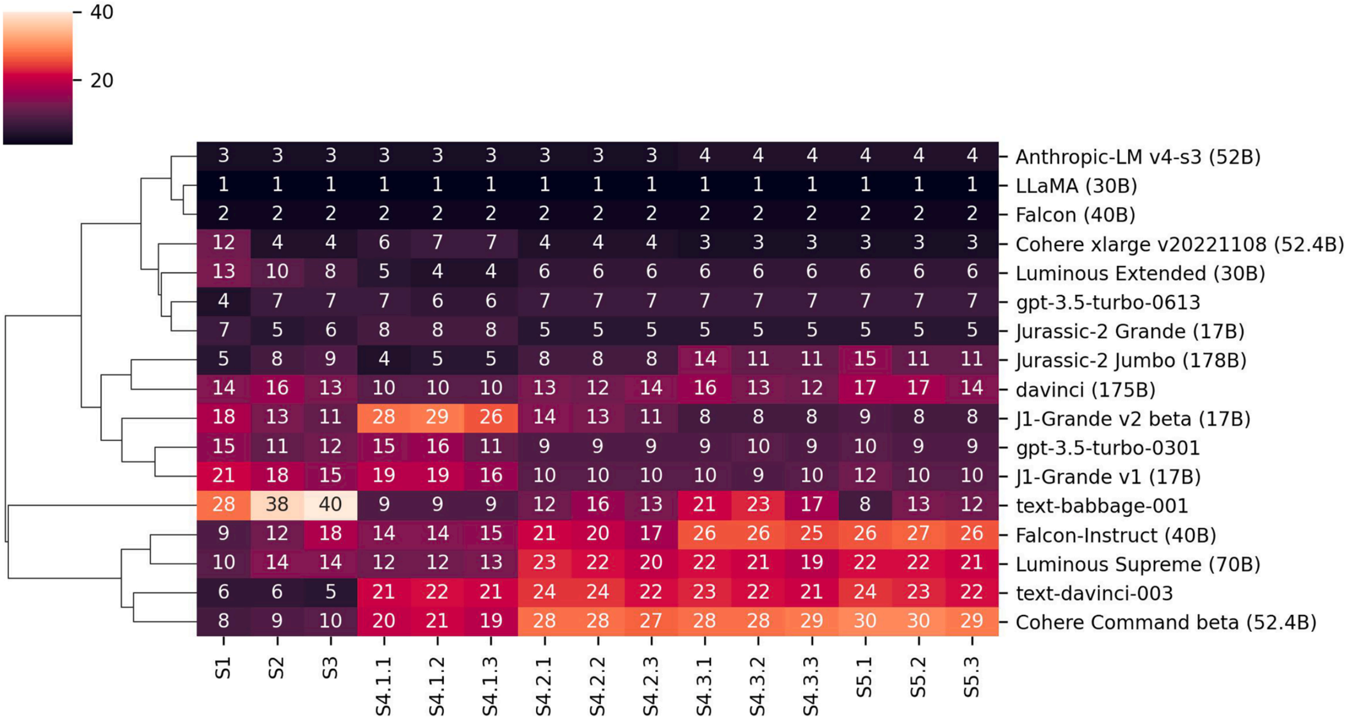
Figs. 9–11 compare LLM rankings across different scenarios for each dataset separately, i.e., NarrativeQA (Fig. 9), NaturalQuestions (open-book) (Fig. 10), and NaturalQuestions (closed-book) (Fig. 11). They display the union of the 10 LLMs with the highest ranking from each scenario. Each row represents an LLM, while each column corresponds to one scenario. Overall, the rankings across the three datasets reveal that some LLMs consistently perform well, with their rankings remaining stable regardless of the correlated metrics (no significant deviations between rankings in S1, S2, and S3 compared to S4.1, S4.2, and S4.3, respectively). Examples include Vicuna v1.3 (7B) and Falcon Instruct (40B) for NarrativeQA; LLaMa (30B), Falcon (40B), and Anthropic-LM v4-s3 (52B) for NaturalQuestions (open-book); and Falcon (40B), Mistral v0.1 (7B), and GPT-3.5 Turbo-0301 for NaturalQuestions (closed-book). However, for each dataset, some LLMs exhibit changes in their rankings - either increases or decreases - indicating that the inclusion of correlated metrics can influence their rankings. Examples of models that show increased (worse) rankings when correlated metrics are excluded across NarrativeQA and NaturalQuestions (open-book) datasets include text-davinci-003 and Cohere Command Beta (52.4B). Such examples in the NaturalQuestions (closed-book) involve Falcon-instruct (40B) and LLaMA (30B). Conversely, the text-babbage-001 model demonstrates improved (better) rankings when correlated metrics are removed, particularly for NarrativeQA and NaturalQuestions (open-book) datasets. However, it does not appear in the top 10 ranked LLMs in any evaluation scenario on NaturalQuestions (closed-book) datasets. Other examples with improved ranking when omitting correlated metrics are T5 (11B) and UL2 (20B) on the NarrativeQA dataset, and Cohere large 20220720 (13.1B), Alpaca (7B), Falcon (7B) (slight improvement), and GLM (130B) on the NaturalQuestions (closed-book) dataset.

By examining the S4.1, S4.2, and S4.3 scenarios and their three repetitions with a single metric from the correlated cluster, we find that the rankings remain robust and reliable. The maximum deviation is only 3 within each scenario separately, indicating that the results remain consistent even when different correlated metrics are included. In S5, where the focus is on bias metrics related to race, gender, and gender-profession, for each dataset individually, we can identify LLMs that are unbiased but still have good overall performance



	S1	S2	S3	S4.1.1	S4.1.2	S4.1.3	S4.2.1	S4.2.2	S4.2.3	S4.3.1	S4.3.2	S4.3.3	S5.1	S5.2	S5.3
text-babbage-001	25	31	27	11	12	12	12	14	19	7	9	12	7	9	13
T5 (11B)	24	42	40	2	5	5	10	10	8	9	8	7	8	7	7
UL2 (20B)	28	37	35	10	10	11	5	5	4	6	5	4	4	4	3
BLOOM (176B)	9	9	5	3	3	4	7	7	7	5	6	6	6	6	6
gpt-3.5-turbo-0613	13	5	4	6	5	3	4	4	3	4	4	3	5	5	4
J1-Grande v2 beta (17B)	5	3	3	4	4	8	3	3	6	3	3	5	3	3	5
Vicuna v1.3 (7B)	2	1	1	1	2	2	1	1	1	1	1	2	2	2	2
Falcon-Instruct (40B)	3	2	2	1	1	1	2	2	2	2	2	1	1	1	1
LLaMA (30B)	8	13	14	13	13	15	15	15	16	18	18	19	18	18	19
Anthropic-LM v4-s3 (52B)	10	12	12	15	15	16	17	17	13	17	16	16	17	17	16
text-davinci-003	4	10	10	12	9	10	18	16	12	16	15	13	16	16	14
Cohere Command beta (52.4B	6	11	11	7	8	13	14	13	15	14	14	14	13	13	15
gpt-3.5-turbo-0301	14	6	9	19	16	17	6	6	5	10	10	9	10	10	9
Llama 2 (7B)	12	8	7	9	11	9	8	9	10	8	7	8	9	8	8
Jurassic-2 Jumbo (178B)	1	4	6	5	6	7	9	8	9	12	12	11	11	11	10
GLM (130B)	7	7	8	8	7	6	11	11	11	11	11	10	12	12	11

Fig. 9. Comparison of LLM rankings (rows) across scenarios (columns) on the NarrativeQA dataset. The portfolio includes the union of the 10 LLMs with the highest ranking from each scenario.

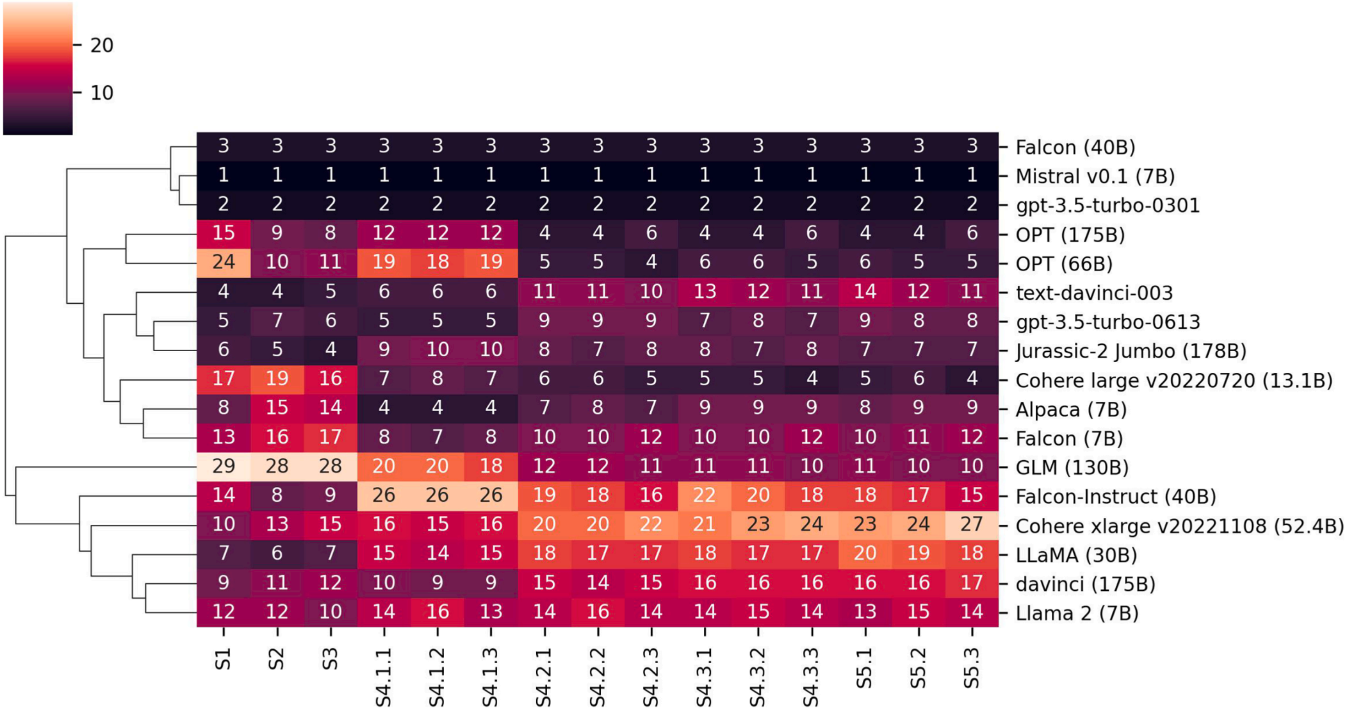


	S1	S2	S3	S4.1.1	S4.1.2	S4.1.3	S4.2.1	S4.2.2	S4.2.3	S4.3.1	S4.3.2	S4.3.3	S5.1	S5.2	S5.3
Anthropic-LM v4-s3 (52B)	3	3	3	3	3	3	3	3	3	4	4	4	4	4	4
LLaMA (30B)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
Falcon (40B)	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
Cohere xlarge v20221108 (52.4B)	12	4	4	6	7	7	4	4	4	3	3	3	3	3	3
Luminous Extended (30B)	13	10	8	5	4	4	6	6	6	6	6	6	6	6	6
gpt-3.5-turbo-0613	4	7	7	7	6	6	7	7	7	7	7	7	7	7	7
Jurassic-2 Grande (17B)	7	5	6	8	8	8	5	5	5	5	5	5	5	5	5
Jurassic-2 Jumbo (178B)	5	8	9	4	5	5	8	8	8	14	11	11	15	11	11
davinci (175B)	14	16	13	10	10	10	13	12	14	16	13	12	17	17	14
J1-Grande v2 beta (17B)	18	13	11	28	29	26	14	13	11	8	8	8	9	8	8
gpt-3.5-turbo-0301	15	11	12	15	16	11	9	9	9	9	10	9	10	9	9
J1-Grande v1 (17B)	21	18	15	19	19	16	10	10	10	10	9	10	12	10	10
text-babbage-001	28	38	40	9	9	9	12	16	13	21	23	17	8	13	12
Falcon-Instruct (40B)	9	12	18	14	14	15	21	20	17	26	26	25	26	27	26
Luminous Supreme (70B)	10	14	14	12	12	13	23	22	20	22	21	19	22	22	21
text-davinci-003	6	6	5	21	22	21	24	24	22	23	22	21	24	23	22
Cohere Command beta (52.4B)	8	9	10	20	21	19	28	28	27	28	28	29	30	30	29

Fig. 10. Comparison of LLM rankings (rows) across scenarios (columns) on the NaturalQuestions (open-book) dataset. The portfolio includes the union of the 10 LLMs with the highest ranking from each scenario.

on the other metrics. For the NarrativeQA dataset, the Falcon-Instruct (40B) and Vicuna v1.3 (7B) are identified as top unbiased LLMs. On the NaturalQuestions (open-book) dataset, those are LLaMA (30B) and Falcon (40B). When looking at the NaturalQuestions (closed-book) dataset, those are Mistral v0.1 (7B), gpt-3.5-turbo-0301, and Falcon (40B).

LLM performance varies by dataset; for example, the LLaMA (30B) model performs well and remains unbiased in the NaturalQuestions (open-book) dataset but shows bias in other datasets. Once users identify an optimal evaluation scenario for their needs, they can use it to rank LLMs across datasets. These rankings can then be statistically tested (e.g., with the Friedman test) for significant differences [54]. Due to the limited number of suitable publicly available datasets for our study (only three), we did not conduct that analysis, as ten or more datasets are generally required.



	S1	S2	S3	S4.1.1	S4.1.2	S4.1.3	S4.2.1	S4.2.2	S4.2.3	S4.3.1	S4.3.2	S4.3.3	S5.1	S5.2	S5.3
Falcon (40B)	3	3	3	3	3	3	3	3	3	3	3	3	3	3	3
Mistral v0.1 (7B)	1	1	1	1	1	1	1	1	1	1	1	1	1	1	1
gpt-3.5-turbo-0301	2	2	2	2	2	2	2	2	2	2	2	2	2	2	2
OPT (175B)	15	9	8	12	12	12	4	4	6	4	4	6	4	4	6
OPT (66B)	24	10	11	19	18	19	5	5	4	6	6	5	6	5	5
text-davinci-003	4	4	5	6	6	6	11	11	10	13	12	11	14	12	11
gpt-3.5-turbo-0613	5	7	6	5	5	5	9	9	9	7	8	7	9	8	8
Jurassic-2 Jumbo (178B)	6	5	4	9	10	10	8	7	8	8	7	8	7	7	7
Cohere large v20220720 (13.1B)	17	19	16	7	8	7	6	6	5	5	5	4	5	6	4
Alpaca (7B)	8	15	14	4	4	4	7	8	7	9	9	9	8	9	9
Falcon (7B)	13	16	17	8	7	8	10	10	12	10	10	12	10	11	12
GLM (130B)	29	28	28	20	20	18	12	12	11	11	11	10	11	10	10
Falcon-Instruct (40B)	14	8	9	26	26	26	19	18	16	22	20	18	18	17	15
Cohere xlarge v20221108 (52.4B)	10	13	15	16	15	16	20	20	22	21	23	24	23	24	27
LLaMA (30B)	7	6	7	15	14	15	18	17	17	18	17	17	20	19	18
davinci (175B)	9	11	12	10	9	9	15	14	15	16	16	16	16	16	17
Llama 2 (7B)	12	12	10	14	16	13	14	16	14	14	15	14	13	15	14

Fig. 11. Comparison of LLM rankings (rows) across scenarios (columns) on the NaturalQuestions (closed-book) dataset. The portfolio includes the union of the 10 LLMs with the highest ranking from each scenario.

To clarify how these rankings were derived, Figs. 12–14 each illustrate, for one of the three datasets, the positive and negative preference flows. In this context, the positive flow measures how much a given LLM outperforms other LLMs across all performance metrics, while the negative flow measures how much it is outperformed by them. Although we do not elaborate further here, the goal is to maximize positive flow and minimize negative flow, as their difference determines the final ranking. The dendrograms on each heatmap (one per benchmark dataset) show clusters of LLMs based on these flows. Additionally, the positive and negative flows depend on the chosen preference function, which takes as input the difference between the performance values of two LLMs.

Fig. 15–17 illustrate the distribution of LLM performance scores by the criteria (performance metric) involved in our experiments on NarrativeQA, NaturalQuestions (open-book), and NaturalQuestions (closed-book) datasets, respectively. As already described in the previous section, the lower standard deviation σ translates into smaller pairwise differences between LLM performance scores, but those illustrations are not included here due to their large number and the already provided explanation of the mapping process in the first experimental context (Section 5.1). The actual performance scores in all three datasets for a selected set of LLMs are illustrated in Appendix E and discussed in relation to the results presented in this section.

Finally, in Figs. 18–20, we present the Spearman rank correlation between the final rankings outputted in all scenarios by dataset, to test their robustness across scenarios. The results indicate that for all three datasets, the rankings are rather robust to changes of the preference function, as the correlation is high (applies to S1-3, S4.x.1-3 ($x \in \{1, 2, 3\}$), S5.1-3).

6. Discussion

This paper presents a transparent, decision-centric benchmarking framework, xLLMBench, which enables decision-makers to rank LLMs based on their preferences across various (potentially conflicting) performance and non-performance criteria. It introduces a set of benchmarking scenarios motivated by real-world application requirements, with guidelines on how such requirements (preferences) can be integrated into the ranking process. The scenarios belong to two experimental contexts: (1) combining results for a single performance metric across different datasets, and (2) combining multiple performance metrics calculated on a single benchmark dataset. The inclusion or exclusion of specific (non-)performance criteria is entirely up to the decision-makers. The paper focuses on the fifth step of the benchmarking process (robust analysis of benchmarking results), assuming that previous steps - selection of performance criteria (metrics), datasets, and LLMs have been conducted following established benchmarking best practices. We select LLM performance criteria from widely-recognized, publicly available sources [17,18]. This decision to reuse reliable publicly available benchmarking data aligns with calls for more effective LLM benchmarking, pointing to high computational costs while evaluating large LLM portfolios on a wide range of datasets [55,56]. We again emphasize that this paper does not introduce new performance metrics, nor does it impose what should be measured during LLM benchmarking, extensive research fields by themselves, outside the scope of this paper. However, our framework allows the inclusion of different types of metrics - whether technical, environmental, or socially grounded based on sentiment, toxicity, or user reactions [57] - directly into the ranking process. Additionally, we clarify that the included measurement of LLM CO_2 emissions refers to the inference phase, as provided by Hugging Face's Open LLM Leaderboard platform[8]. However, future work can include metrics such as carbon footprint or energy consumption per training iteration, particularly if the research focus shifts toward ranking LLMs for fine-tuning purposes and if comparable measurements across different LLMs are publicly available in their model cards.

We show how different decision-makers' preferences can be integrated into the method (1) by specifying preferences over LLMs' pairwise differences through preference functions and (2) by specifying preferences over criteria through weights used to calculate the average

[8] https://huggingface.co/docs/leaderboards/open_llm_leaderboard/emissions

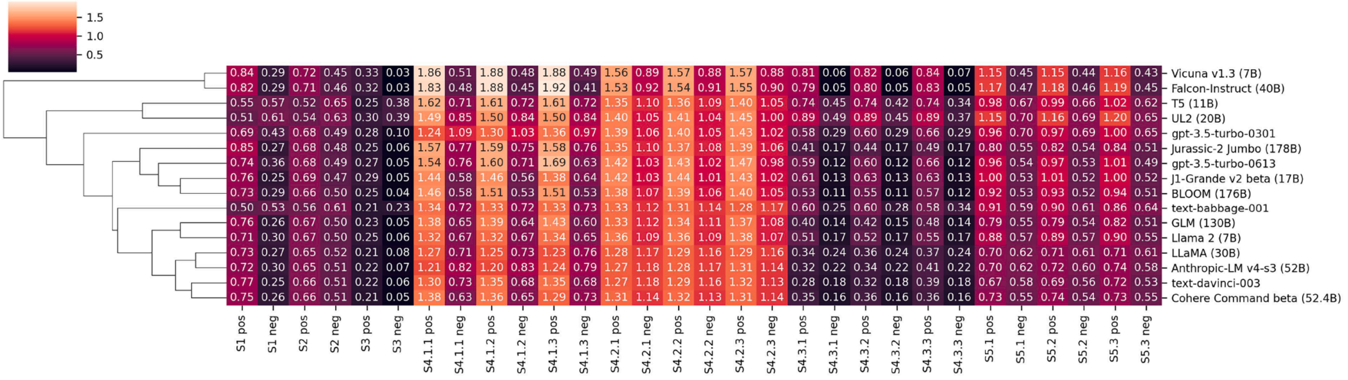


Fig. 12. Comparison of LLM positive and negative preference flows (rows) across scenarios (columns) on the NarrativeQA dataset. The portfolio includes the union of the 10 LLMs with the highest ranking from each scenario.

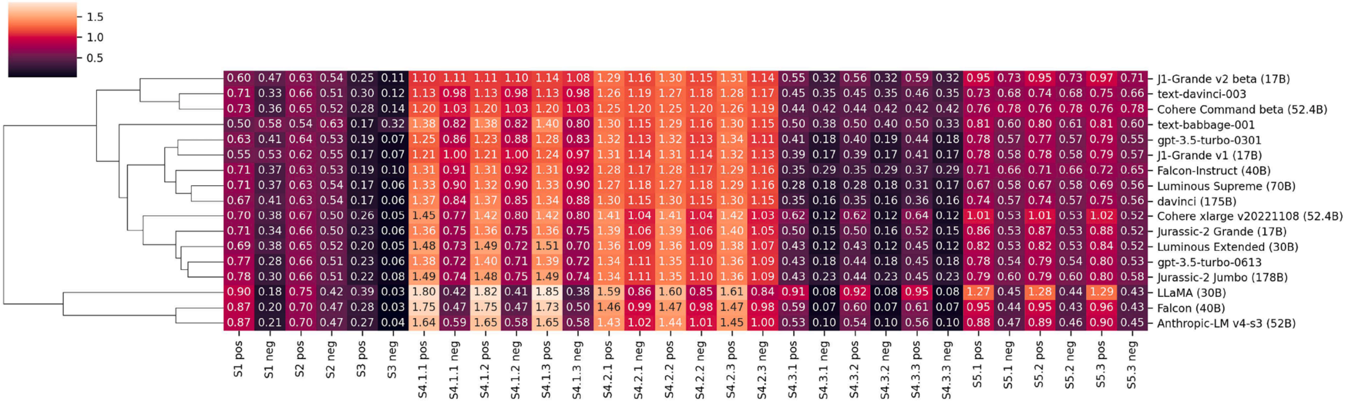


Fig. 13. Comparison of LLM positive and negative preference flows (rows) across scenarios (columns) on the NaturalQuestions (open-book) dataset. The portfolio includes the union of the 10 LLMs with the highest ranking from each scenario.

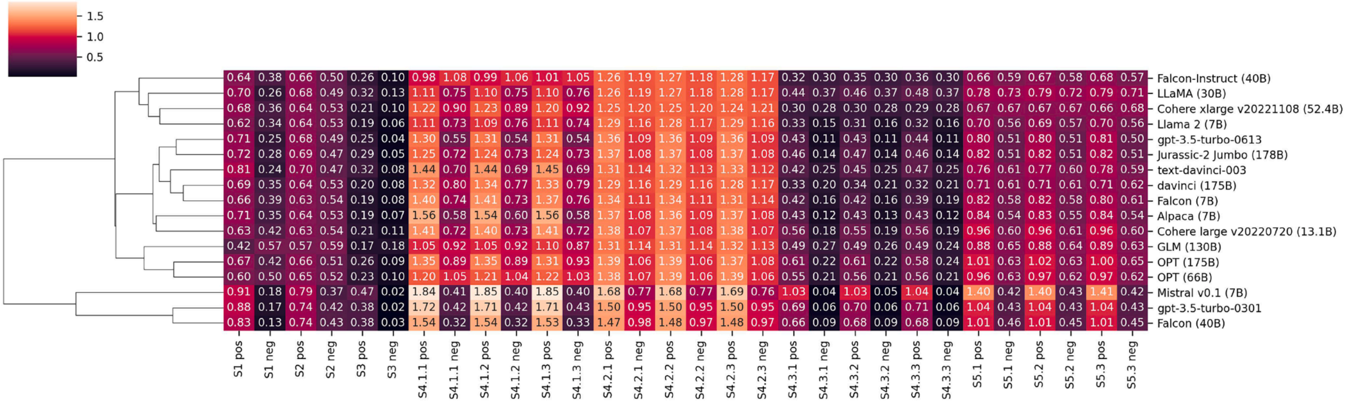


Fig. 14. Comparison of LLM positive and negative preference flows (rows) across scenarios (columns) on the NaturalQuestions (closed-book) dataset. The portfolio includes the union of the 10 LLMs with the highest ranking from each scenario.

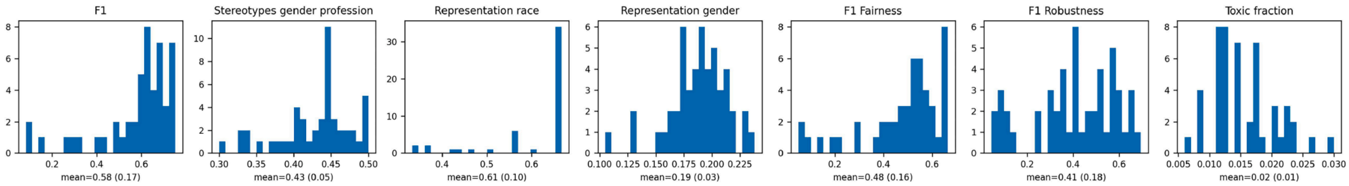


Fig. 15. LLM score distribution by performance metric for the NarrativeQA dataset.

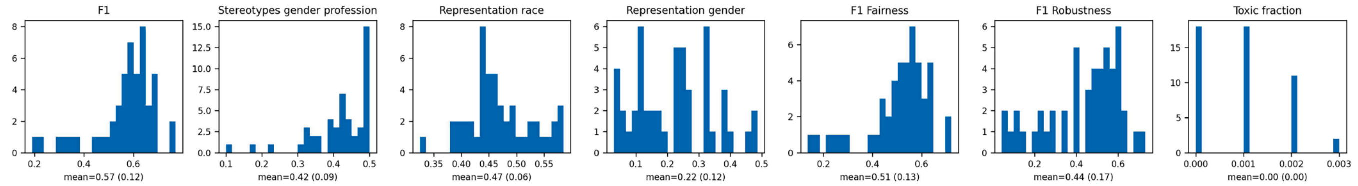


Fig. 16. LLM score distribution by performance metric for the NaturalQuestions (open-book) dataset.

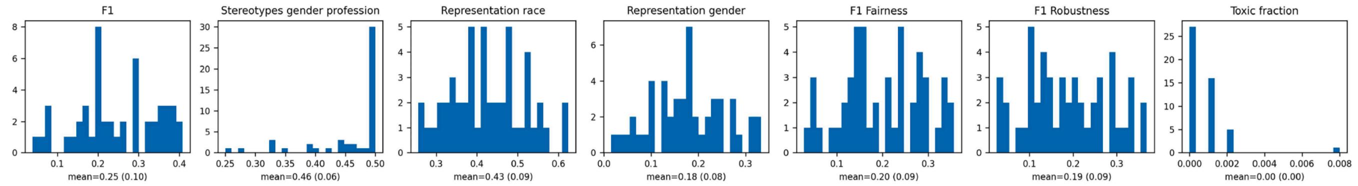
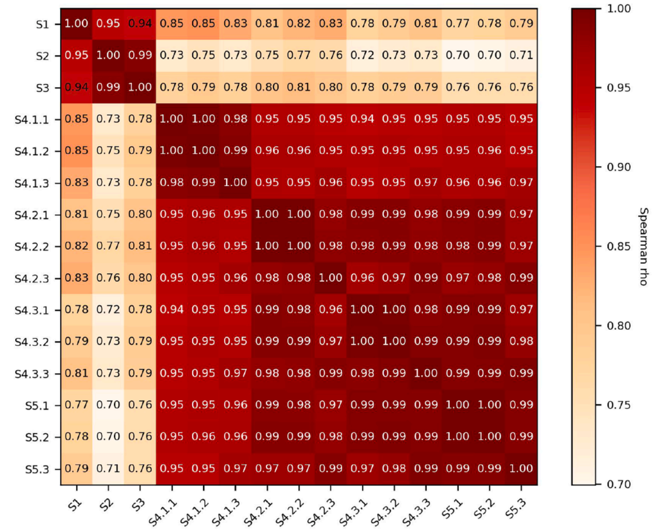
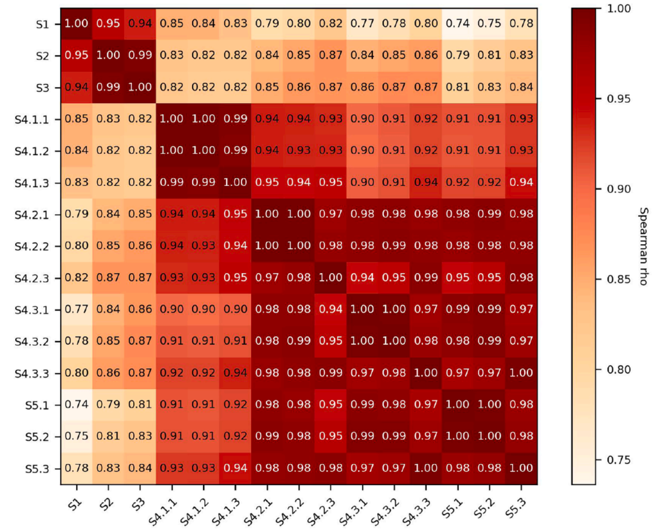


Fig. 17. LLM score distribution by performance metric for the NaturalQuestions (closed-book) dataset.



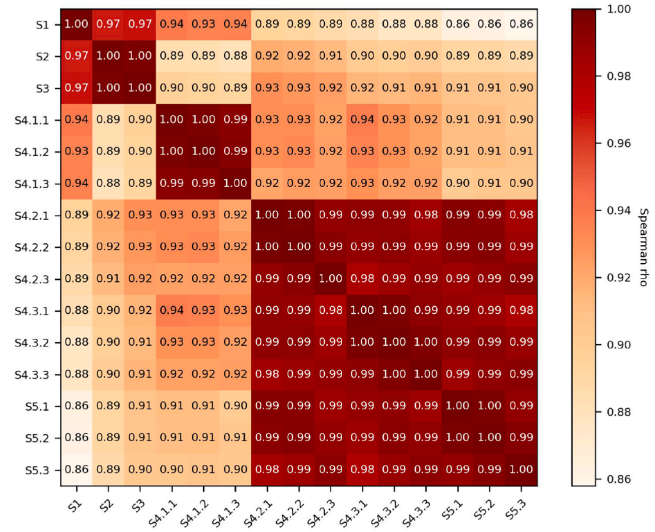
	S1	S2	S3	S4.1.1	S4.1.2	S4.1.3	S4.2.1	S4.2.2	S4.2.3	S4.3.1	S4.3.2	S4.3.3	S5.1	S5.2	S5.3
S1	1.00	0.95	0.94	0.85	0.85	0.83	0.81	0.82	0.83	0.78	0.79	0.81	0.77	0.78	0.79
S2	0.95	1.00	0.99	0.73	0.75	0.73	0.75	0.77	0.76	0.72	0.73	0.73	0.70	0.70	0.71
S3	0.94	0.99	1.00	0.78	0.79	0.78	0.80	0.81	0.80	0.78	0.79	0.79	0.76	0.76	0.76
S4.1.1	0.85	0.73	0.78	1.00	1.00	0.98	0.95	0.95	0.95	0.94	0.95	0.95	0.95	0.95	0.95
S4.1.2	0.85	0.75	0.79	1.00	1.00	0.99	0.96	0.96	0.95	0.95	0.95	0.95	0.95	0.96	0.95
S4.1.3	0.83	0.73	0.78	0.98	0.99	1.00	0.95	0.95	0.96	0.95	0.95	0.97	0.96	0.96	0.97
S4.2.1	0.81	0.75	0.80	0.95	0.96	0.95	1.00	1.00	0.98	0.99	0.99	0.98	0.99	0.99	0.97
S4.2.2	0.82	0.77	0.81	0.95	0.96	0.95	1.00	1.00	0.98	0.98	0.99	0.98	0.98	0.99	0.97
S4.2.3	0.83	0.76	0.80	0.95	0.95	0.96	0.98	0.98	1.00	0.96	0.97	0.99	0.97	0.98	0.99
S4.3.1	0.78	0.72	0.78	0.94	0.95	0.95	0.99	0.98	0.96	1.00	1.00	0.98	0.99	0.99	0.97
S4.3.2	0.79	0.73	0.79	0.95	0.95	0.95	0.99	0.99	0.97	1.00	1.00	0.99	0.99	0.99	0.98
S4.3.3	0.81	0.73	0.79	0.95	0.95	0.97	0.98	0.98	0.99	0.98	0.99	1.00	0.99	0.99	0.99
S5.1	0.77	0.70	0.76	0.95	0.95	0.96	0.99	0.98	0.97	0.99	0.99	0.99	1.00	1.00	0.99
S5.2	0.78	0.70	0.76	0.95	0.96	0.96	0.99	0.99	0.98	0.99	0.99	0.99	1.00	1.00	0.99
S5.3	0.79	0.71	0.76	0.95	0.95	0.97	0.97	0.97	0.99	0.97	0.98	0.99	0.99	0.99	1.00

Fig. 18. Spearman rank correlation between the rankings outputted in the different scenarios on the NarrativeQA dataset.



	S1	S2	S3	S4.1.1	S4.1.2	S4.1.3	S4.2.1	S4.2.2	S4.2.3	S4.3.1	S4.3.2	S4.3.3	S5.1	S5.2	S5.3
S1	1.00	0.95	0.94	0.85	0.84	0.83	0.79	0.80	0.82	0.77	0.78	0.80	0.74	0.75	0.78
S2	0.95	1.00	0.99	0.83	0.82	0.82	0.84	0.85	0.87	0.84	0.85	0.86	0.79	0.81	0.83
S3	0.94	0.99	1.00	0.82	0.82	0.82	0.85	0.86	0.87	0.86	0.87	0.87	0.81	0.83	0.84
S4.1.1	0.85	0.83	0.82	1.00	1.00	0.99	0.94	0.94	0.93	0.90	0.91	0.92	0.91	0.91	0.93
S4.1.2	0.84	0.82	0.82	1.00	1.00	0.99	0.94	0.93	0.93	0.90	0.91	0.92	0.91	0.91	0.93
S4.1.3	0.83	0.82	0.82	0.99	0.99	1.00	0.95	0.94	0.95	0.90	0.91	0.94	0.92	0.92	0.94
S4.2.1	0.79	0.84	0.85	0.94	0.94	0.95	1.00	1.00	0.97	0.98	0.98	0.98	0.98	0.99	0.98
S4.2.2	0.80	0.85	0.86	0.94	0.93	0.94	1.00	1.00	0.98	0.98	0.99	0.98	0.98	0.98	0.98
S4.2.3	0.82	0.87	0.87	0.93	0.93	0.95	0.97	0.98	1.00	0.94	0.95	0.99	0.95	0.95	0.98
S4.3.1	0.77	0.84	0.86	0.90	0.90	0.90	0.98	0.98	0.94	1.00	1.00	0.97	0.99	0.99	0.97
S4.3.2	0.78	0.85	0.87	0.91	0.91	0.91	0.98	0.99	0.95	1.00	1.00	0.98	0.98	0.99	0.97
S4.3.3	0.80	0.86	0.87	0.92	0.92	0.94	0.98	0.98	0.99	0.97	0.98	1.00	0.97	0.97	1.00
S5.1	0.74	0.79	0.81	0.91	0.91	0.92	0.98	0.98	0.95	0.99	0.98	0.97	1.00	1.00	0.98
S5.2	0.75	0.81	0.83	0.91	0.91	0.92	0.99	0.98	0.95	0.99	0.99	0.97	1.00	1.00	0.98
S5.3	0.78	0.83	0.84	0.93	0.93	0.94	0.98	0.98	0.98	0.97	0.97	1.00	0.98	0.98	1.00

Fig. 19. Spearman rank correlation between the rankings outputted in the different scenarios on the NaturalQuestions (open-book) dataset.



	S1	S2	S3	S4.1.1	S4.1.2	S4.1.3	S4.2.1	S4.2.2	S4.2.3	S4.3.1	S4.3.2	S4.3.3	S5.1	S5.2	S5.3
S1	1.00	0.97	0.97	0.94	0.93	0.94	0.89	0.89	0.89	0.88	0.88	0.88	0.86	0.86	0.86
S2	0.97	1.00	1.00	0.89	0.89	0.88	0.92	0.92	0.91	0.90	0.90	0.90	0.89	0.89	0.89
S3	0.97	1.00	1.00	0.90	0.90	0.89	0.93	0.93	0.92	0.92	0.91	0.91	0.91	0.91	0.90
S4.1.1	0.94	0.89	0.90	1.00	1.00	0.99	0.93	0.93	0.92	0.94	0.93	0.92	0.91	0.91	0.90
S4.1.2	0.93	0.89	0.90	1.00	1.00	0.99	0.93	0.93	0.92	0.93	0.93	0.92	0.91	0.91	0.91
S4.1.3	0.94	0.88	0.89	0.99	0.99	1.00	0.92	0.92	0.92	0.93	0.92	0.92	0.90	0.91	0.90
S4.2.1	0.89	0.92	0.93	0.93	0.93	0.92	1.00	1.00	0.99	0.99	0.99	0.98	0.99	0.99	0.98
S4.2.2	0.89	0.92	0.93	0.93	0.93	0.92	1.00	1.00	0.99	0.99	0.99	0.99	0.99	0.99	0.99
S4.2.3	0.89	0.91	0.92	0.92	0.92	0.92	0.99	0.99	1.00	0.98	0.99	0.99	0.99	0.99	0.99
S4.3.1	0.88	0.90	0.92	0.94	0.93	0.93	0.99	0.99	0.98	1.00	1.00	0.99	0.99	0.99	0.98
S4.3.2	0.88	0.90	0.91	0.93	0.93	0.92	0.99	0.99	0.99	1.00	1.00	1.00	0.99	0.99	0.99
S4.3.3	0.88	0.90	0.91	0.92	0.92	0.92	0.98	0.99	0.99	0.99	1.00	1.00	0.99	0.99	0.99
S5.1	0.86	0.89	0.91	0.91	0.91	0.90	0.99	0.99	0.99	0.99	0.99	0.99	1.00	1.00	0.99
S5.2	0.86	0.89	0.91	0.91	0.91	0.91	0.99	0.99	0.99	0.99	0.99	0.99	1.00	1.00	0.99
S5.3	0.86	0.89	0.90	0.90	0.91	0.90	0.98	0.99	0.99	0.98	0.99	0.99	0.99	0.99	1.00

Fig. 20. Spearman rank correlation between the rankings outputted in the different scenarios on the NaturalQuestions (closed-book) dataset.

preference index. This paper analyzed three well-known PROMETHEE II preference functions, but as priorities for deploying LLMs differ across organizations, the framework allows defining customized preference functions aligned with application-specific requirements (see Appendix A). However, since that task requires appropriate skills, it should be addressed by dedicated data scientists in the organizations. For illustrative purposes, in this paper, we set the preference functions' parameters in a data-driven manner, but also do a sensitivity analysis to assess the effects of their change in Appendix B. It is important to note that those parameters can be set manually by decision-makers, as they are best positioned to determine what is the criteria's practical significance to their use case or when a certain pairwise difference in the performance of two LLMs is not meaningful from a practical perspective. The AHP method also offers a user-friendly way of specifying pairwise criteria importance, and its use with PROMETHEE II is demonstrated in Appendix D. Through visualization of each step of the decision-making process, we attempt to make the method as transparent to the decision-makers as possible and facilitate such a customized configuration.

The results show that although most LLMs maintain consistent rankings across scenarios, some drop in performance on contamination-free datasets, and some stand out as unbiased when bias-related metrics are prioritized. Overall, this approach gives better insights in each LLM's unique strengths and weaknesses than any single aggregate performance metric. To this end, we show that PROMETHEE II rankings are traceable via the average preference indices, the positive and negative preference flows. To present them in a user-friendly manner, we use established visualization methods such as dendrograms and heatmaps, which clearly separate good from bad alternatives in separate criteria, as well as cumulatively. They also visually cluster similar alternatives for improved interpretability. We visualize and explain the mapping of criteria original values to the target values outputted by three different preference functions, as well as visualize the criteria and LLM ranking correlations. We believe that in such a manner, we make the decision-making process more interpretable and traceable even to non-expert decision-makers. We hope to motivate future research on making MCDM methods even more interpretable for this purpose.

A few practical use cases where the xLLMBench can be useful are discussed below. The use cases are selected for illustrative purposes only, so the list is not an exhaustive one. Future work will analyze each use case in detail, using domain-specific LLM benchmark data and scenarios defined with the help of domain experts. The xLLMBench ranking approach is also applicable to the selection of other types of ML models in various domains (e.g., health [58] or even cybersecurity [59]), when benchmarking data for different relevant (non-)performance metrics is available.

- In the financial sector, LLMs are increasingly adopted to support market sentiment analysis, forecasting market trends, and financial document information extraction / summarization [60]. However, this domain is characterized by unique vocabulary (different meanings of certain words from their general meaning), so domain-specific models are often required [61]. A recent study has shown that different LLMs differ in their capabilities to perform financial tasks, and while most excel at information extraction and text analysis, at the same time, they show lower performance in text generation and forecasting [62]. Furthermore, selecting an appropriate LLM often requires balancing conflicting objectives such as domain-specific performance vs. inference cost, evaluation on multiple domain-specific tasks, and use of domain-specific evaluation metrics instead of general ML metrics [63]. Compliance with ethical standards, legal, privacy, and security regulations is key requirement as well [60]. Since comprehensive financial LLM benchmarks are becoming available [62,64,65], xLLMBench can support decision-making by enabling financial organizations' AI teams to systematically embed multiple (conflicting) (non-)performance organizational priorities into the LLM ranking process, e.g., prioritizing accuracy (performance metric) over operational cost (non-performance metric). Furthermore, xLLMBench provides transparency and reproducibility, key requirements in financial decision-making.
- In the heavily-regulated healthcare domain, LLMs are also attracting attention in research, educational, and clinical contexts [66]. LLMs are used in medical question answering, text analysis, information retrieval/extraction, X-ray analysis, to name a few use cases [67]. Responsible LLM deployment in healthcare requires a selection process accounting not only for performance but also for data privacy, fairness, safety, robustness to known vulnerabilities, compliance with strict regulatory requirements, to name a few. Furthermore, recency and expert validation of the data used to train LLMs are relevant requirements in the LLM selection process, to avoid inaccuracies [68]. Therefore, integration of LLMs in healthcare requires meticulous strategies for change management and risk mitigation [69]. As medical LLM benchmarks become available [70], xLLMBench is particularly suitable for balancing performance with the large set of non-performance criteria, resulting in LLM selection with transparent justifications which can be validated by human experts, a crucial requirement in healthcare.
- AI companies providing Software as a Service (SaaS) solutions based on LLMs, face trade-offs between performance, operational cost, and environmental sustainability. Due to their increased usage, concerns related to LLMs energy consumption, carbon emissions, and water use are raised [71], not just in their training but also in their inference (operational) phase [72,73]. Benchmarking large portfolios of LLMs across a broad range of capabilities can be computationally expensive as well, so calls for more effective benchmarking arise [55,56]. Customers increasingly demand high-performing yet environment-friendly AI solutions, which creates complex decision landscapes for companies, related to backend LLM training, fine-tuning, or selection. By integrating xLLMBench in their internal benchmarking pipelines, these providers can rank candidate LLMs based on customizable (in this case, conflicting) priorities that reflect both market demands on performance and societal demands on sustainability, improving LLM choices, increasing client trust, and positioning the company as a socially responsible one.

While this paper focuses on the use of MCDM methods for LLM ranking, we acknowledge that the problem can also be addressed using multi-objective optimization algorithms [74,75]. Our decision to use MCDM methods aligns with the call to action by the optimization community [76], which highlights the proliferation of algorithms offering no clear benefits, often introducing existing methods under different names without meaningful contributions, and reporting biased benchmarking results. For this reason, we believe that applying multi-objective optimization methods to this problem - an open direction for future research - should be undertaken by research groups with a primary focus on multi-objective optimization.

Finally, a limitation of this study is the use of publicly available benchmark data, since the dataset and LLM selection steps of the benchmarking process (B1 and B2) are outside the scope of this study. Future work will address this by applying meta-learning to analyze dataset complementarity and ensure dataset diversity. Revision of the LLM selection will also be made, ensuring the diversity of the LLM portfolio. Finally, the development of an automated algorithm selection process is planned, enabling the selection of the most appropriate LLM for new benchmark datasets.

7. Conclusion

The widespread adoption of large language models (LLMs) requires meticulous evaluation of their capabilities across multiple benchmark datasets, using diverse performance metrics. Selecting an LLM for a real-world application often involves balancing conflicting performance and non-performance criteria such as high domain accuracy vs. low energy consumption, fair outcomes, or low CO_2 emissions. However, traditional LLM benchmarking approaches predominantly rely on individual performance metrics or direct human feedback when ranking LLMs, which can be resource-intensive and fail to address the complex, real-world application-specific requirements. Additionally, they rarely provide systematic methods for combining multiple criteria into a unified and interpretable ranking.

To address these limitations, this paper presents xLLMBench, a transparent, decision-centric benchmarking framework designed to empower decision-makers to rank LLMs based on their specific preferences across diverse and often conflicting criteria. By framing LLM ranking as a multi-criteria decision-making problem and adapting the well-established PROMETHEE II method to incorporate decision-makers' preferences in the process, it outputs highly customized and practically meaningful LLM rankings. xLLMBench targets the fifth step of the benchmarking process (robust analysis of benchmarking results), assuming prior selection of datasets, metrics, and LLMs following established best practices. The benefits of its use are demonstrated through a set of scenarios organized under two experimental paradigms: (1) combining comparable metrics calculated on different benchmark datasets and (2) combining multiple metrics calculated on a single dataset. Additionally, we analyze how variations in criteria weights and method parameters influence the ranking outcomes, as well as explore the effects of correlated criteria on the outcomes.

Our findings reveal that while certain LLMs retain stable rankings across most scenarios, others vary when contamination-free datasets or bias-related metrics are prioritized. Such findings provide a deeper understanding of LLM-specific strengths and weaknesses beyond simple aggregate scores. To enhance transparency and interpretability, xLLMBench integrates visualizations and detailed explanations at each stage of the ranking process. By facilitating the integration of application-specific preferences into customized decision-making workflows, xLLMBench introduces a novel and impactful approach to LLM benchmarking, further enhancing the utility and relevance of existing benchmarking platforms.

Declaration of generative AI and AI-assisted technologies in the writing process

During the preparation of this work, the authors used ChatGPT to check and correct grammar. After using this tool/service, the authors reviewed and edited the content as needed and take full responsibility for the content of the published article.

CRediT authorship contribution statement

Ana Gjorgjevikj: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Resources, Methodology, Investigation, Funding acquisition, Formal analysis, Data curation, Conceptualization; **Ana Nikolikj:** Writing – review & editing, Validation, Resources, Investigation; **Barbara Koroušić Seljak:** Writing – review & editing, Validation, Supervision, Project administration, Investigation, Funding acquisition; **Tome Eftimov:** Writing – review & editing, Writing – original draft, Validation, Supervision, Project administration, Methodology, Investigation, Funding acquisition, Formal analysis, Conceptualization.

Data availability

The link to the code is available in the text.

Declaration of competing interest

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

Acknowledgment

This work is funded by the Slovenian Research and Innovation Agency under program grant P2-0098, project grant No. GC-0001, and young researcher grant No. PR-12897 to AN; and by the European Union under Grant Agreement 101211695 (HE MSCA-PF AutoLLMSelect) and Grant Agreement 101187010 (HE ERA Chair AutoLearn-SI).

Appendix A. Reproducibility and extensibility

The GitHub repository for xLLMBench [19] contains the code and all the configurations of the experimental scenarios described in this paper that are needed to reproduce the experiments. After obtaining the necessary input data (described in Section 4), the users should make sure that it contains no missing values (listwise deletion is used in this paper). The users can then run the processing pipeline with the selected configuration, following the examples provided in the repository. Users can also modify the configuration according to their application-specific requirements to run it on their application-specific data. We emphasize that custom configurations should be defined by skilled data scientists who can translate application-specific requirements to appropriate framework configurations, and estimate the effects of such configurations on the results both theoretically and empirically. While running, the pipeline outputs visualizations at each step of the process, i.e., distribution of the pairwise distances by criteria before and after application of preference functions, the average preference index matrix, positive and negative preference flows, and the final ranking, accompanied by the net flow values. Since running the visualizations can be resource-intensive for larger input datasets, users have the opportunity to specify if visualizations should be skipped.

The framework allows for the addition of other preference functions, their registration at appropriate places in the code, and use in the experiment configuration, following the implementation of the three currently available preference functions. However, we again emphasize that such steps should be done only by skilled data scientists, be based on solid theoretical grounding, and have their effects on the results evaluated through extensive experiments.

Appendix B. Sensitivity analysis of preference functions parameters

Sensitivity analysis of the parameters of the linear and Gaussian preference function is performed to study the changes in the rankings outputted by the different configurations. The indifference a and preference b parameters of the linear function are initially set to the minimum and maximum value of the appropriate criteria, i.e., there is no indifference and preference for LLM pairwise differences under any of the criteria. During the sensitivity analysis, the indifference parameter a is increased by 10% and 20% of its initial value, while at the same time the preference parameter b is decreased by 10% and 20%, accordingly. Therefore, in the first case, there is an indifference for 10% of the lowest pairwise differences and a preference for 10% of the highest pairwise differences for all criteria to which the linear preference function is applied (depending on the configuration of the scenario). In scenarios in which the linear function is applied to some criteria, while other preference functions are applied to other criteria, the parameters of the other preference functions are fixed to their default values. For the Open LLM Leaderboard dataset, those scenarios are S2, S4, S5.2, S6, and S7, while for HELM datasets those are S2, S4.2.1, S4.2.2, S4.2.3, S5.1, S5.2, and S5.3. The Spearman correlation coefficient between the LLM rankings outputted by the different configurations of each different scenario for the Open LLM Leaderboard dataset is given in Fig. B.1, for HELM NaturalQuestions (closed-book) dataset in Fig. B.4, for HELM NaturalQuestions (open-book) dataset in Fig. B.3, and for HELM NarrativeQA in Fig. B.2. The results show that in all cases the correlation is above 0.98, indicating stable and robust PROMETHEE II rankings in all scenarios included in the sensitivity analysis.

In the case of the Gaussian preference function, the standard deviation parameter σ is set by default to the standard deviation of the values under each criterion (σ_{def}). During the sensitivity analysis, this parameter of the Gaussian preference function is changed to values $\pm 10/20\%$, i.e, $\sigma_{def} + n * \sigma_{def}, n \in \{-0.2, -0.1, 0.1, 0.2\}$. All experiments that use the Gaussian preference function for at least one criterion are included in the sensitivity analysis while keeping the default parameter values for all other preference functions applied to other criteria. For Open LLM Leaderboard those sub-scenarios are S3, S4, S5.3, S6, and S7, while for HELM those are S3, S4.3.1, S4.3.2, S4.3.3, S5.1, S5.2, and S5.3. The Spearman correlation coefficient between the LLM rankings outputted by the different configurations of each different experiment for Open LLM Leaderboard dataset is given in Fig. B.5, for HELM NaturalQuestions (closed-book) dataset in Fig. B.8, for HELM NaturalQuestions (open-book) dataset in Fig. B.7, and for HELM NarrativeQA in Fig. B.6. For all datasets and experiments, the correlation is above 0.95, again indicating stable and robust PROMETHEE II rankings in all experiments included in the sensitivity analysis.

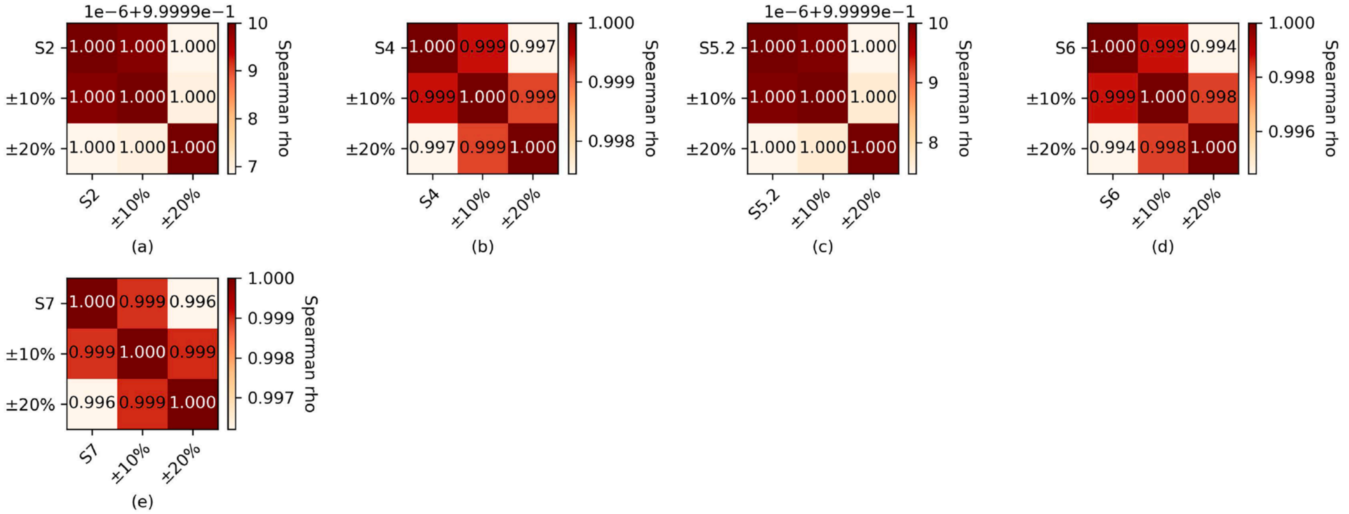


Fig. B.1. Sensitivity analysis of the parameters of the linear preference function in different scenarios on the Open LLM Leaderboard dataset. Scenario (a) S2, (b) S4, (c) S5.2, (d) S6, and (e) S7.

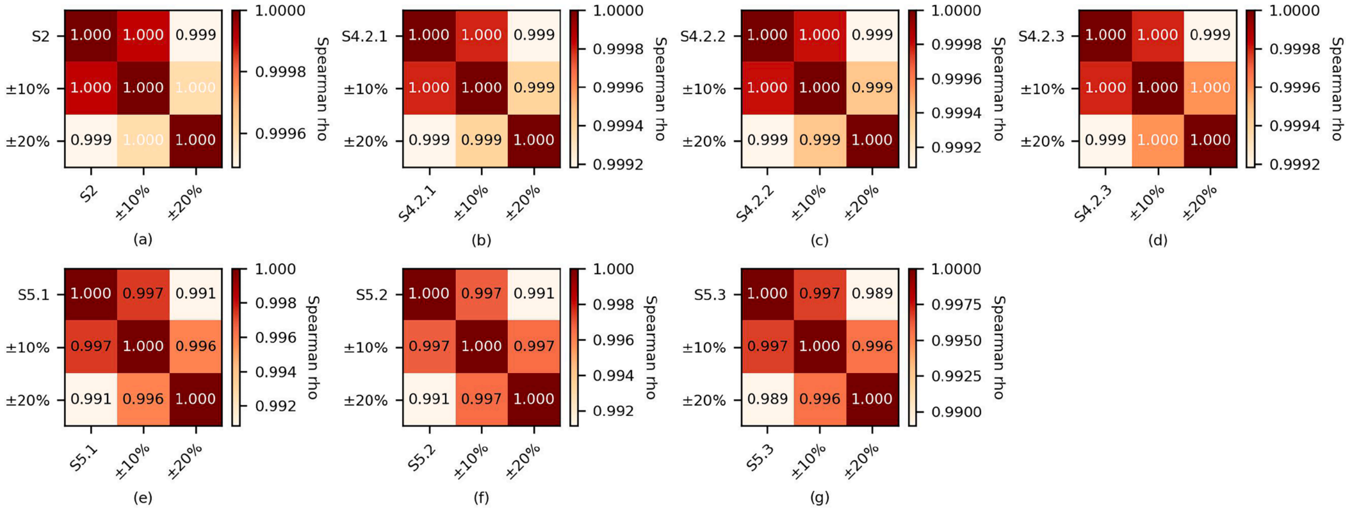


Fig. B.2. Sensitivity analysis of the parameters of the linear preference function in different scenarios on HELM NarrativeQA dataset. Scenario (a) S2, (b) S4.2.1, (c) S4.2.2, (d) S4.2.3, (e) S5.1, (f) S5.2, and (g) S5.3.

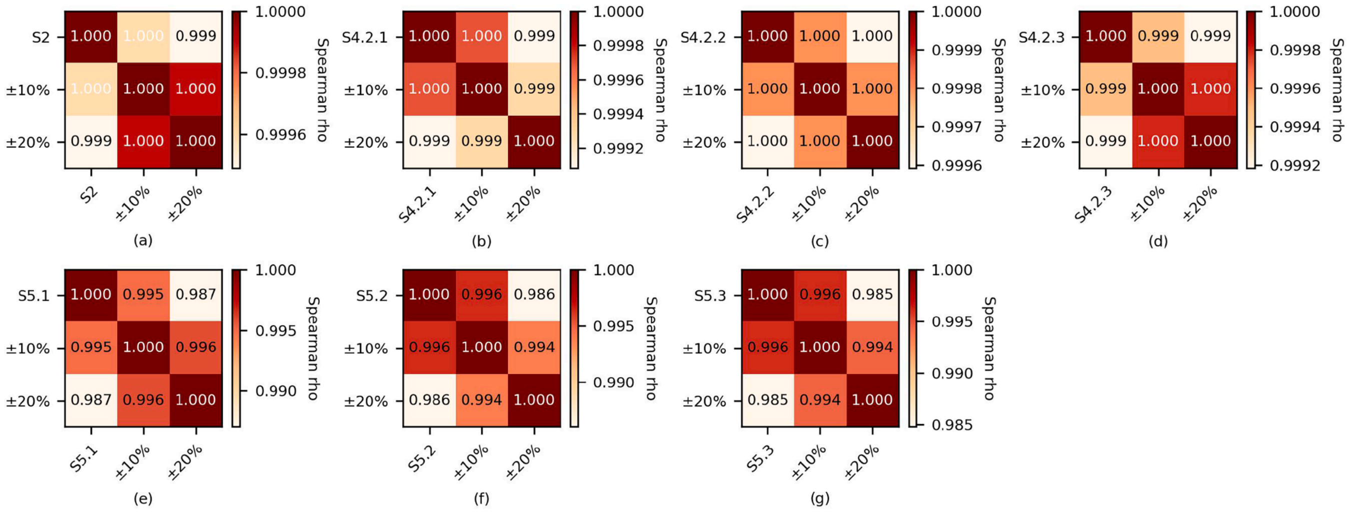


Fig. B.3. Sensitivity analysis of the parameters of the linear preference function in different scenarios on HELM NaturalQuestions (open-book) dataset. Scenario (a) S2, (b) S4.2.1, (c) S4.2.2, (d) S4.2.3, (e) S5.1, (f) S5.2, and (g) S5.3.

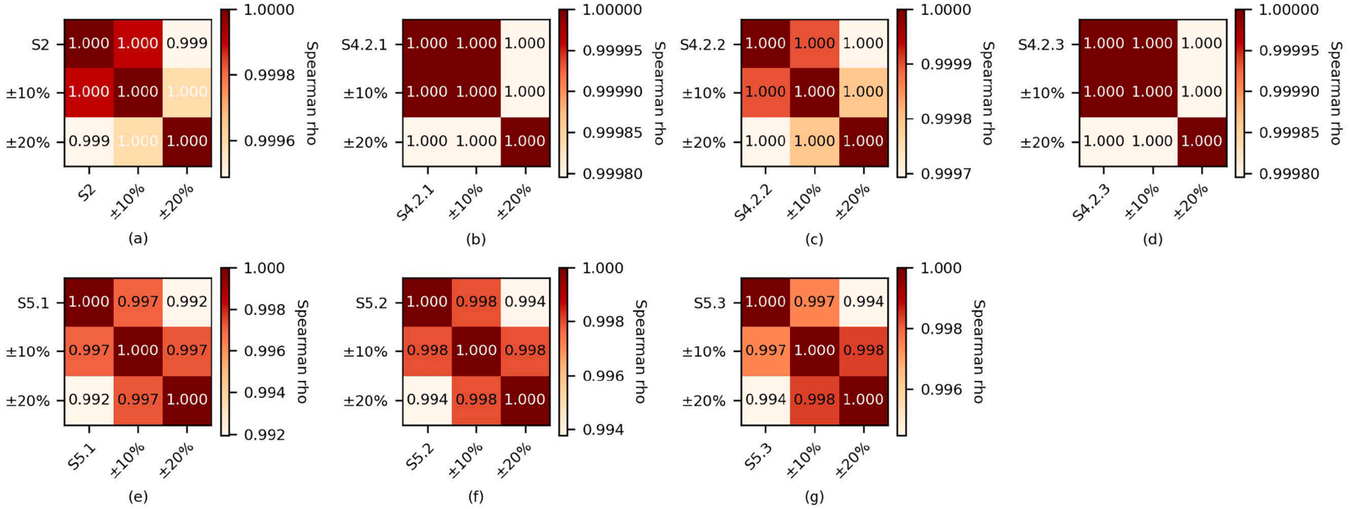
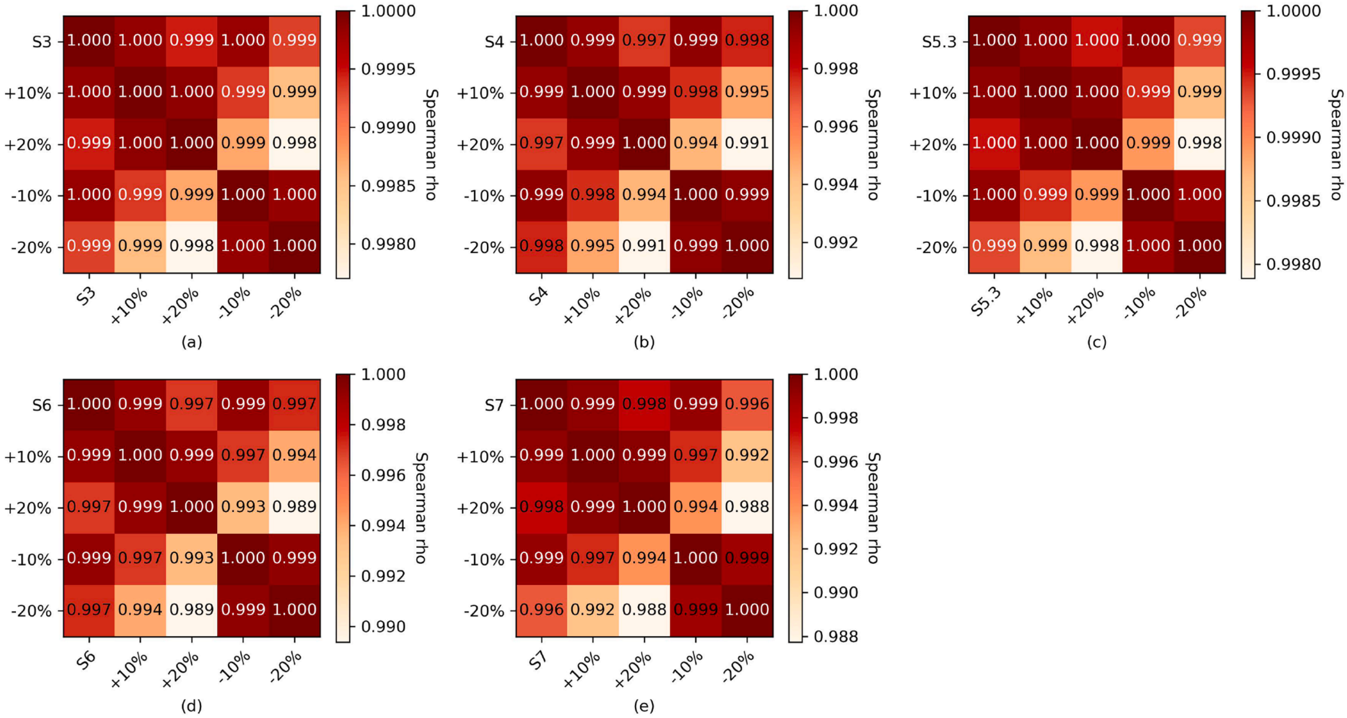


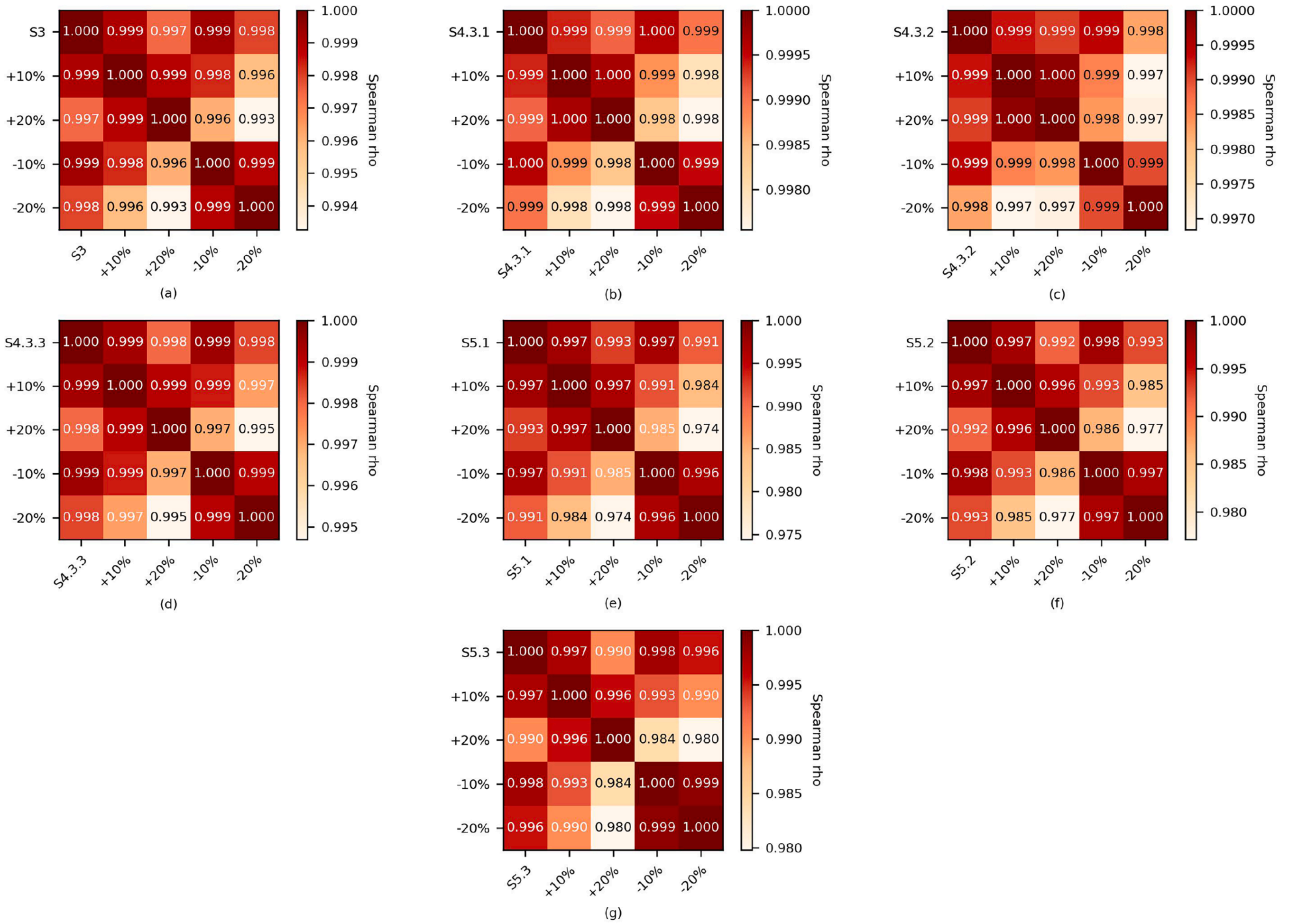
Fig. B.4. Sensitivity analysis of the parameters of the linear preference function in different scenarios on HELM NaturalQuestions (closed-book) dataset. Scenario (a) S2, (b) S4.2.1, (c) S4.2.2, (d) S4.2.3, (e) S5.1, (f) S5.2, and (g) S5.3.



(a) (b) (c)

(d) (e)

Fig. B.5. Sensitivity analysis of the standard deviation parameter of the Gaussian preference function in different scenarios on the Open LLM Leaderboard dataset. Scenario (a) S3, (b) S4, (c) S5.3, (d) S6, and (e) S7.



(a)

(b)

(c)

(d)

(e)

(f)

(g)

Fig. B.6. Sensitivity analysis of the standard deviation parameter of the Gaussian preference function in different scenarios on the HELM NarrativeQA dataset. Scenario (a) S3, (b) S4.3.1, (c) S4.3.2, (d) S4.3.3, (e) S5.1, (f) S5.2, and (g) S5.3.

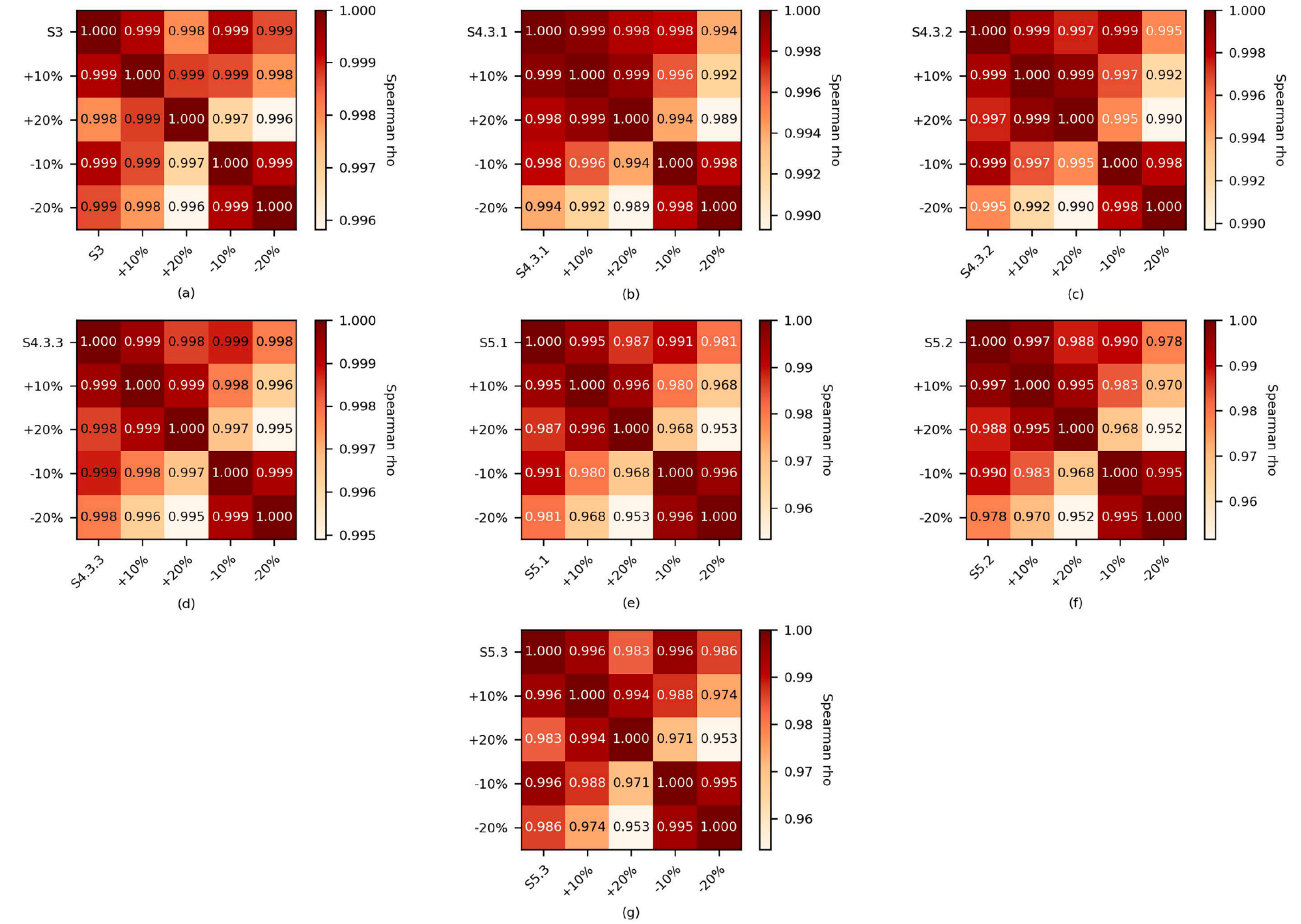


Fig. B.7. Sensitivity analysis of the standard deviation parameter of the Gaussian preference function in different scenarios on the HELM NaturalQuestions (open-book) dataset. Scenario (a) S3, (b) S4.3.1, (c) S4.3.2, (d) S4.3.3, (e) S5.1, (f) S5.2, and (g) S5.3.

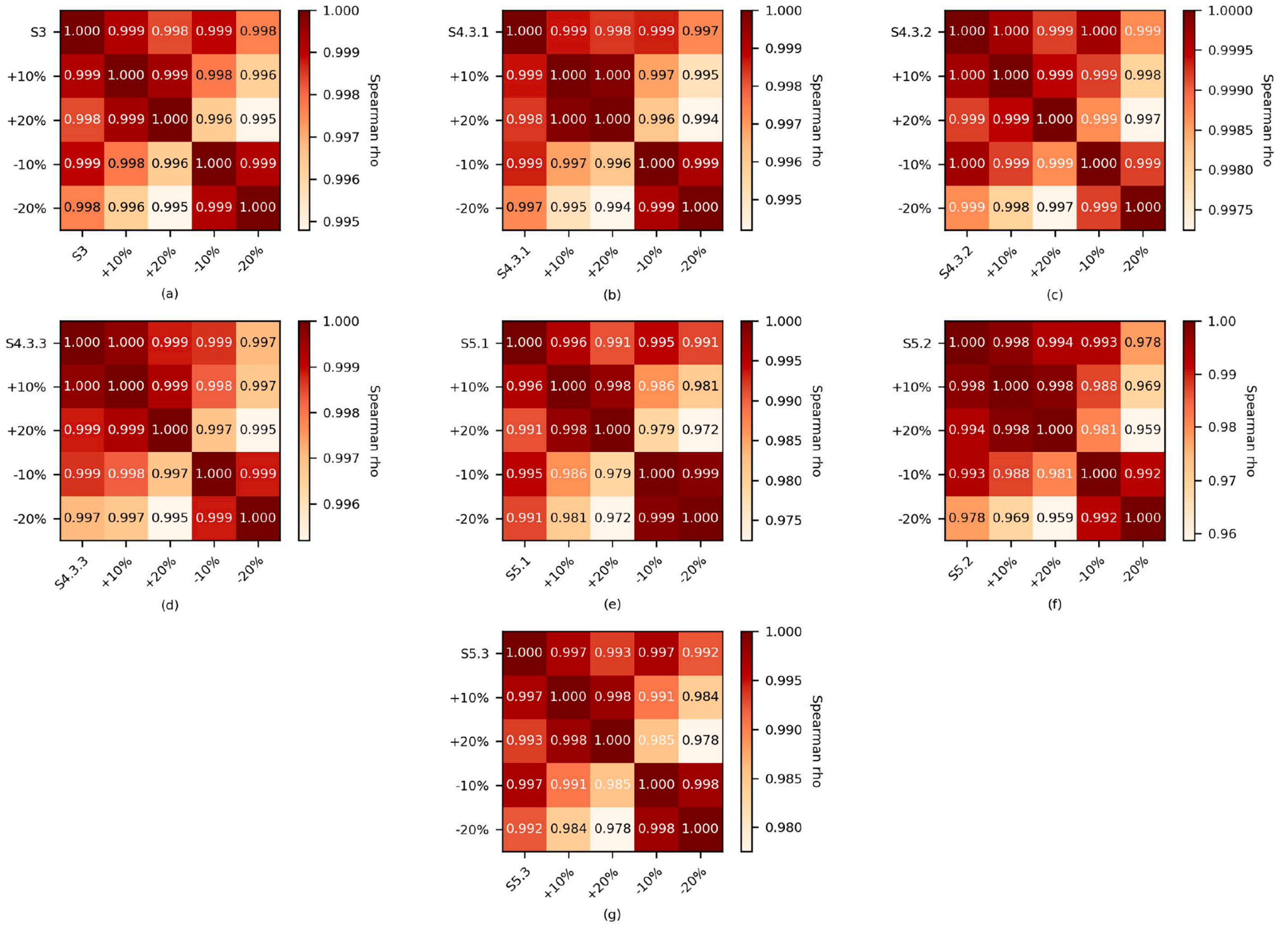


Fig. B.8. Sensitivity analysis of the standard deviation parameter of the Gaussian preference function in different scenarios on the HELM NaturalQuestions (closed-book) dataset. Scenario (a) S3, (b) S4.3.1, (c) S4.3.2, (d) S4.3.3, (e) S5.1, (f) S5.2, and (g) S5.3.

Appendix C. Comparison with other MCDM methods

The large number of MCDM methods proposed in the research literature to date may output different rankings for the same set of alternatives (particularly when alternatives are similar), and clear guidelines on selecting the most appropriate MCDM method are not available [77]. Doing a comparative analysis of different methods is often suggested to see their similarities and differences in the use case of interest. We follow that approach to compare PROMETHEE II rankings outputted in different scenarios with the rankings of two widely used MCDM methods according to research literature [78]. Those were TOPSIS [47] and VIKOR [48]. The two methods differ from PROMETHEE II (an outranking method) since they rank alternatives in terms of an “ideal” solution, although in different ways. TOPSIS prefers a solution with a small vector-based distance to a positive-ideal solution and a large distance to a negative-ideal solution, while VIKOR prefers a compromise solution maximizing the so-called “group utility” and minimizing the “individual regret” [79].

Since TOPSIS and VIKOR only allow specifying preference toward criteria through weights, certain scenarios are redundant in this comparison. For example, in the case of the Open LLM Leaderboard dataset, scenarios S1, S2, S3, changing the PROMETHEE II preference function have the same configuration when running TOPSIS and VIKOR, so they are run only once. The same applies to S5.1, S5.2, and S5.3. S4, which involves different preference functions and the same criteria weights, is inapplicable for TOPSIS and VIKOR. In the case of HELM datasets, S1, S2, and S3 have the same configuration for TOPSIS and VIKOR, so they are run only once. The same applies to scenarios S4.x.1, S4.x.2, S4.x.3, $x \in \{1, 2, 3\}$, which are run only once accordingly. Scenarios S5.1, S5.2, and S5.3 are inapplicable. With TOPSIS, we use the min-max normalization of the decision matrix. With VIKOR, we do not use normalization, and the value of the parameter v is set to 0.5. Their implementation in the pymcdm software library is used [80]. To compare the rankings, the Spearman correlation coefficient is calculated.

Fig. C.1 illustrates the Spearman correlation by scenario run on the Open LLM Leaderboard dataset and includes the Hugging Face average-base ranking as well. The rankings outputted by PROMETHEE II, TOPSIS, and VIKOR are highly correlated, with a correlation above 0.9 in almost all cases. TOPSIS has a slightly higher correlation with PROMETHEE II compared to VIKOR, but in all cases, the correlation is the lowest with the average-based rankings. Similar conclusions can be drawn from Fig. C.2 HELM NarrativeQA dataset, Fig. C.3 for the HELM NaturalQuestions (open-book) dataset, and Fig. C.4 for the HELM NaturalQuestions (closed-book) dataset. Again, the correlation is high, particularly between PROMETHEE II and TOPSIS, and slightly lower between VIKOR and the other two methods. VIKOR’s difference to the other two methods is slightly more pronounced for the NaturalQuestions (open-book) dataset. The observations are in line with those of other studies in the field [43].

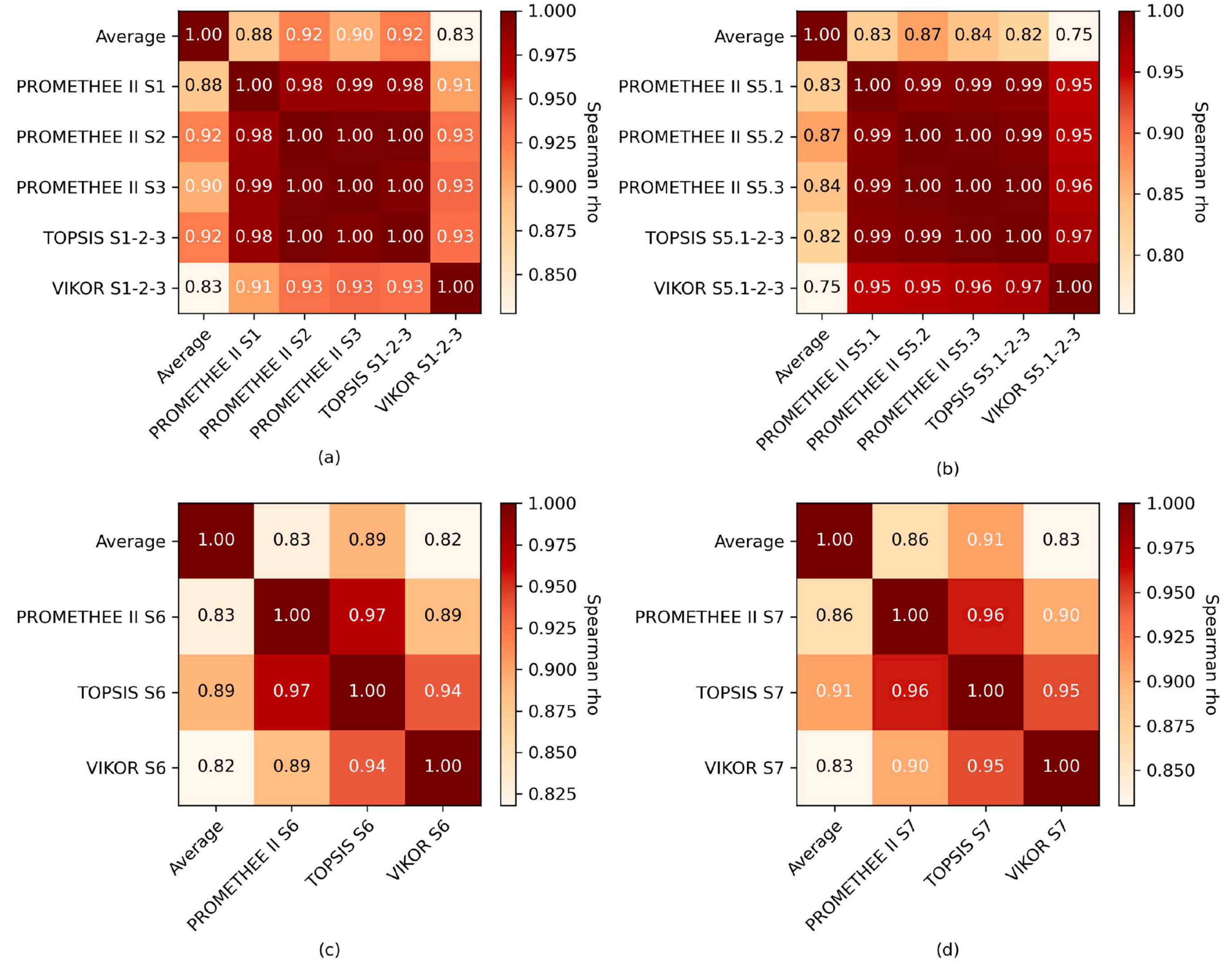


Fig. C.1. Comparison of PROMETHEE II rankings with those of other MCDM methods across scenarios on the Open LLM Leaderboard dataset. Scenarios (a) S1, S2, and S3; (b) S5.1, S5.2, and S5.3; (c) S6; and (d) S7.

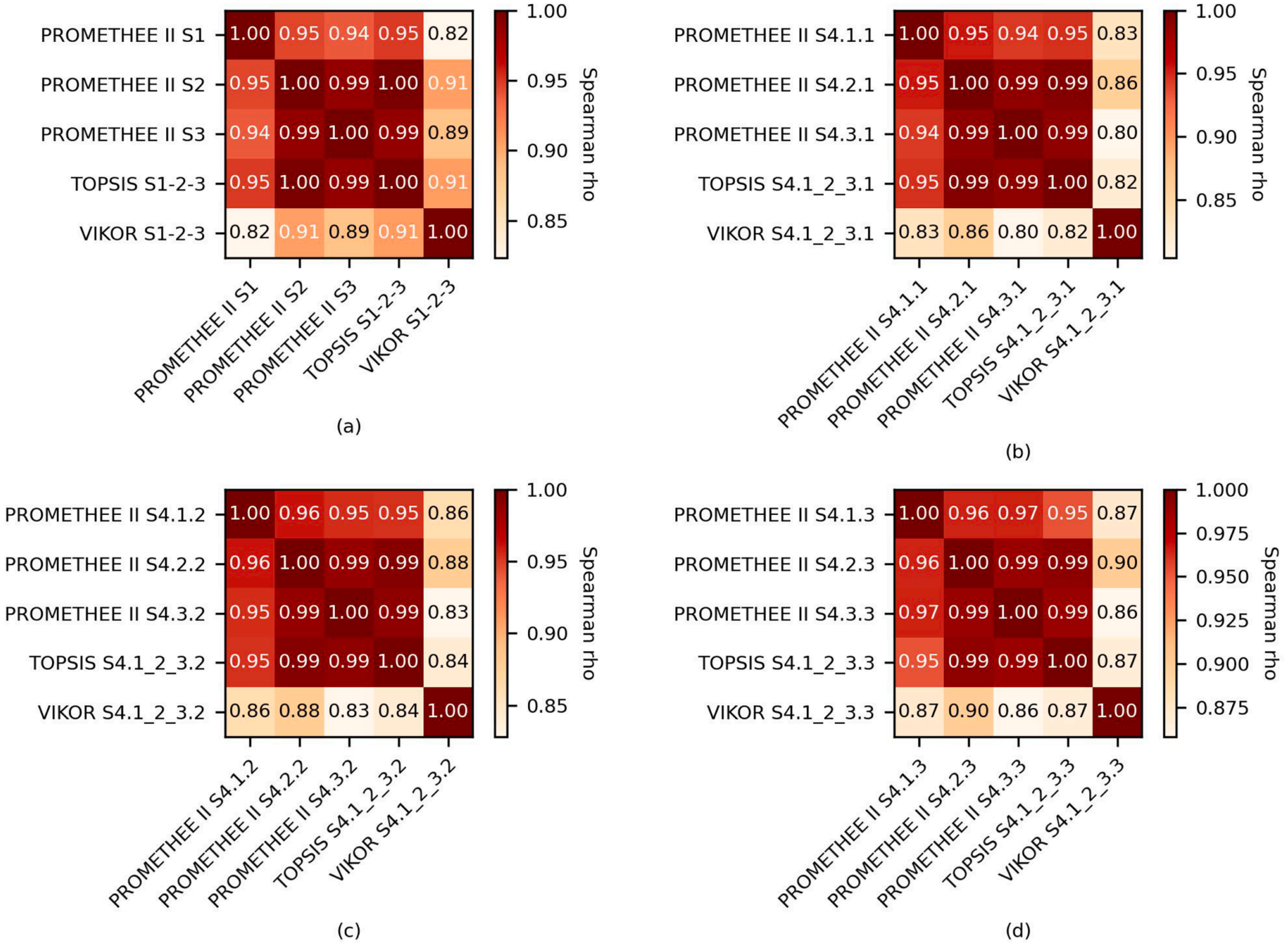


Fig. C.2. Comparison of PROMETHEE II rankings with those of other MCDM methods across scenarios on the HELM NarrativeQA dataset. Scenarios (a) S1, S2, and S3; (b) S4.1.1, S4.2.1, and S4.3.1; (c) S4.1.2, S4.2.2, and S4.3.2; and (d) S4.1.3, S4.2.3, and S4.3.3.

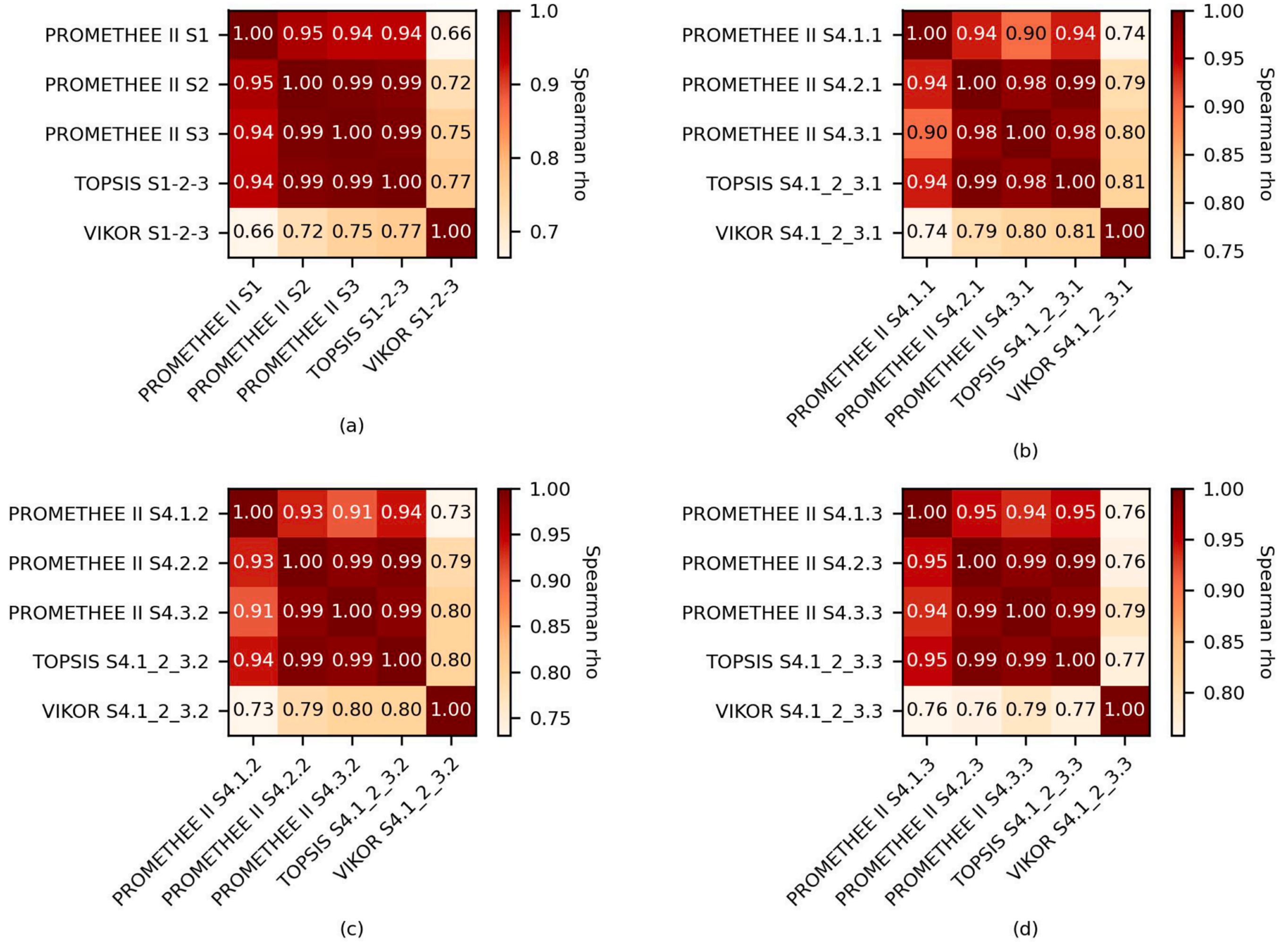


Fig. C.3. Comparison of PROMETHEE II rankings with those of other MCDM methods across scenarios on the HELM NaturalQuestions (open-book) dataset. Scenarios (a) S1, S2, and S3; (b) S4.1.1, S4.2.1, and S4.3.1; (c) S4.1.2, S4.2.2, and S4.3.2; and (d) S4.1.3, S4.2.3, and S4.3.3.

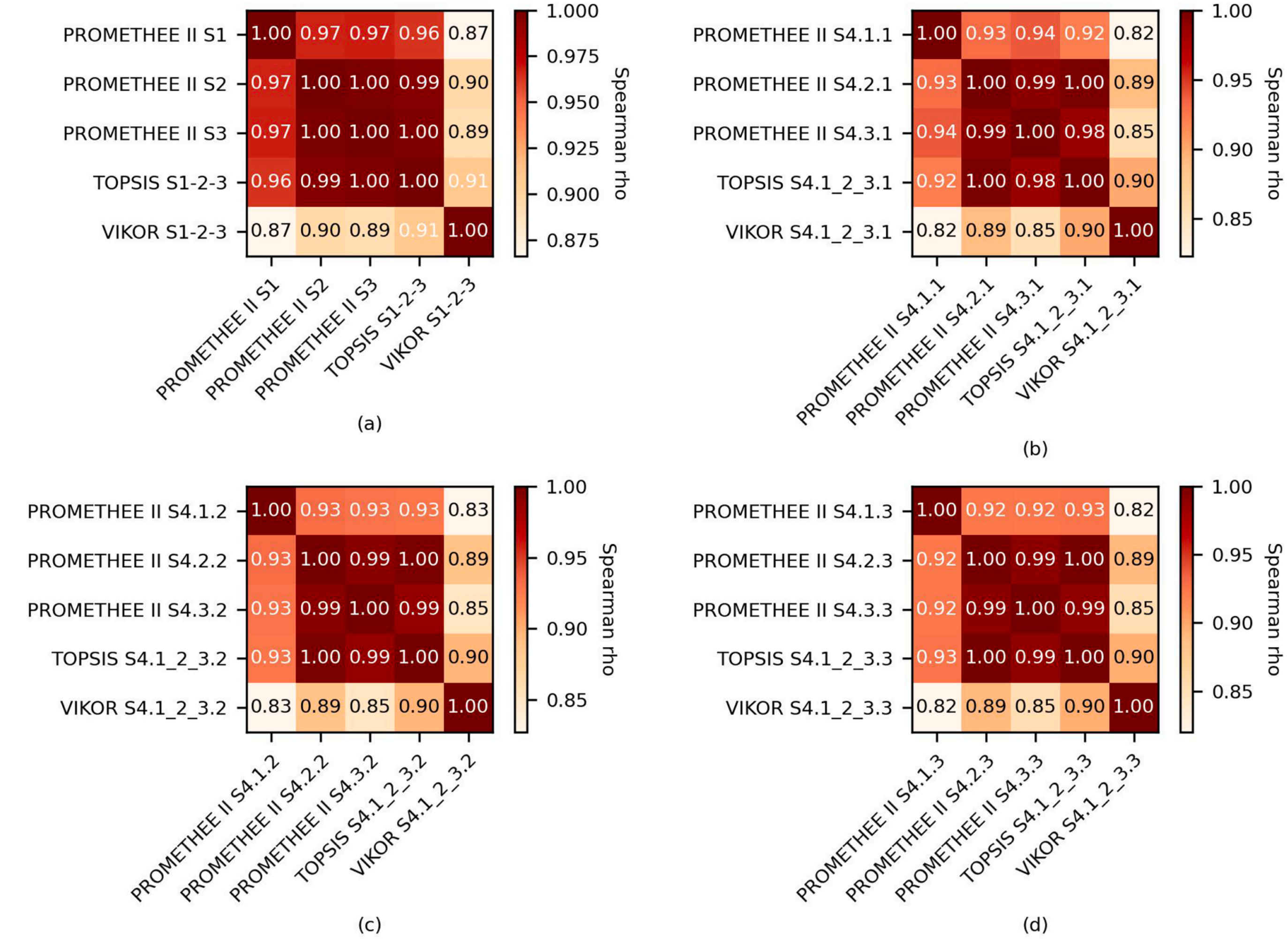


Fig. C.4. Comparison of PROMETHEE II rankings with those of other MCDM methods across scenarios on the HELM NaturalQuestions (closed-book) dataset. Scenarios (a) S1, S2, and S3; (b) S4.1.1, S4.2.1, and S4.3.1; (c) S4.1.2, S4.2.2, and S4.3.2; and (d) S4.1.3, S4.2.3, and S4.3.3.

Appendix D. Analytic Hierarchy Process as weighting method

AHP [53] is one of the most widely used MCDM methods [78,81]. AHP allows hierarchical organization of the criteria and their subcriteria, as well as user-defined pairwise judgments on the importance of one criterion over another. The importance is defined on a scale from 1 to 9, where 1 represents equal importance of the two criteria, 3 - moderate importance of one criterion over the other, 5 - essential or strong importance, 7 - very strong importance, and 9 - extreme importance [53]. Based on the matrix of pairwise importance judgments, the criterion weights are calculated. AHP can be used with other MCDM methods, including PROMETHEE II, to calculate criteria weights [82,83].

Several scenarios presented in this paper include different user-defined weights for the criteria in combination with the same preference function. Those are scenarios S5.1, S5.2, and S5.3 involving the Open LLM Leaderboard dataset, in which bias-related metrics have higher user-defined weights than the rest. Instead of explicitly specifying weights for each criterion, we explore whether AHP allows a more user-friendly way to specify preferences for criterion pairs. Using the AHP importance scale, for each criterion pair we define an importance p if the first criterion is more important than the second, or $1/p$ if the opposite. We form an $m \times m$ matrix, where m is the number of criteria, and use it to calculate the AHP weights. We then rerun the three scenarios with those specific weights and obtain PROMETHEE II rankings. We repeat the whole procedure for four values of p, $p \in \{3, 5, 7, 9\}$. We then calculate the Spearman correlation coefficient between all pairs of rankings by scenario. The criteria weights explicitly defined from our side in comparison with the AHP calculated weights are illustrated in Fig. D.1, which shows a slightly different weight distribution of the first compared to the rest. The ranking correlation by scenario is given in Fig. D.2. It can be concluded that the calculated AHP-based weights result in highly correlated rankings, the correlation of which slightly decreases as the difference of the pairwise importance value increases. The correlation is only slightly lower with the explicitly defined weights. As expected, it is the highest when the pairwise importance of one criterion over another is moderate (value of 3), which is the closest to our manually defined weight distribution. In addition to the stability of the rankings, this analysis shows how combining PROMETHEE II with AHP allows decision-makers a user-friendly way to specify preferences over criteria pairs and get a weight distribution calculated by a well-established MCDM method.

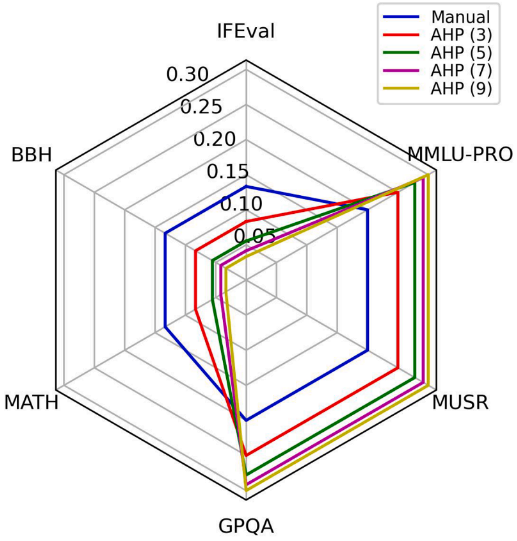
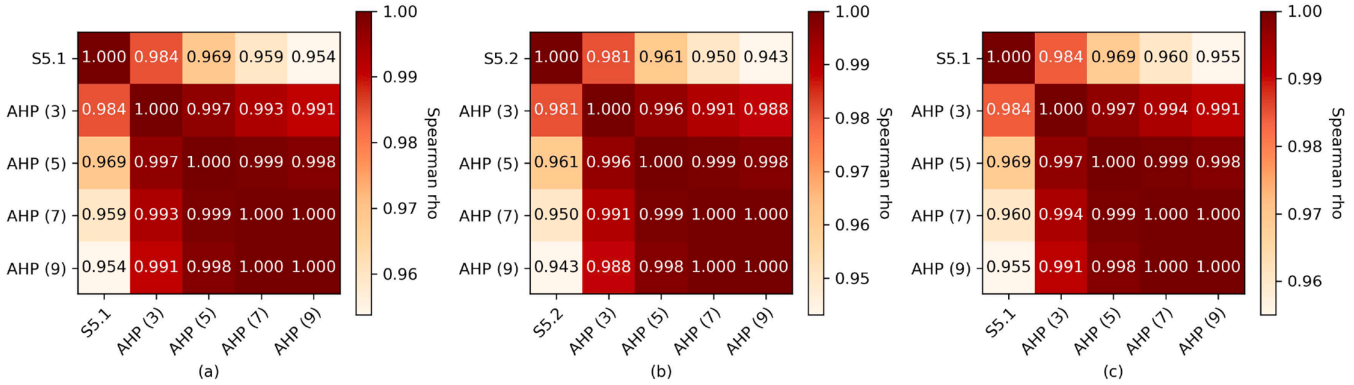


Fig. D.1. Manual criteria weights and criteria weights computed using the AHP method in scenarios S5.1, S5.2, and S5.3 on the Open LLM Leaderboard dataset. The pairwise criteria importance parameter p used in the AHP method is given in brackets.



(a) (b) (c)

Fig. D.2. Spearman correlation of the rankings outputted by PROMETHEE II using manual criteria weights and AHP-derived weights across scenarios on the Open LLM Leaderboard dataset. Scenario (a) S5.1; (b) S5.2; (c) S5.3.

Appendix E. Original LLM performance scores across benchmark datasets

To further enhance the interpretability of the results presented in Section 5, for each dataset we select three LLMs that have constantly the highest ranking in all scenarios presented in Section 5, and illustrate their actual scores for each criterion in Fig. E.1. The figure shows that the selected LLMs do not differ significantly in their scores calculated with most of the metrics. However, in the case of the Open LLM Leaderboard dataset (Fig. E.1 (a)), the LLMs differ in the CO_2 cost, where CalmeRys-78B-Orpo-v0.1 has a lower (better) score than the other two LLMs. This lower score then results in a better ranking of CalmeRys-78B-Orpo-v0.1 compared to the other two LLMs in scenario S7, as illustrated in Fig. 2. Another such difference is visible for the NaturalQuestions (open-book) dataset (Fig. E.1 (c)), where LLaMA (30B) has a lower (better) score on the metric referring to stereotypical associations related to gender groups with the target profession. This lower score results in a better ranking of LLaMA (30B) than the other two LLMs in scenarios S5.1, S5.2, and S5.3, as given in Fig. 10, and higher positive preference flows, as given in Fig. 13. Although other such approximate conclusions can be drawn by this analogy, we emphasize that the results presented in Section 5 are calculated on the full portfolio of LLMs, so for their precise interpretation, looking at the full portfolio of LLMs and their scores calculated with all metrics is required.

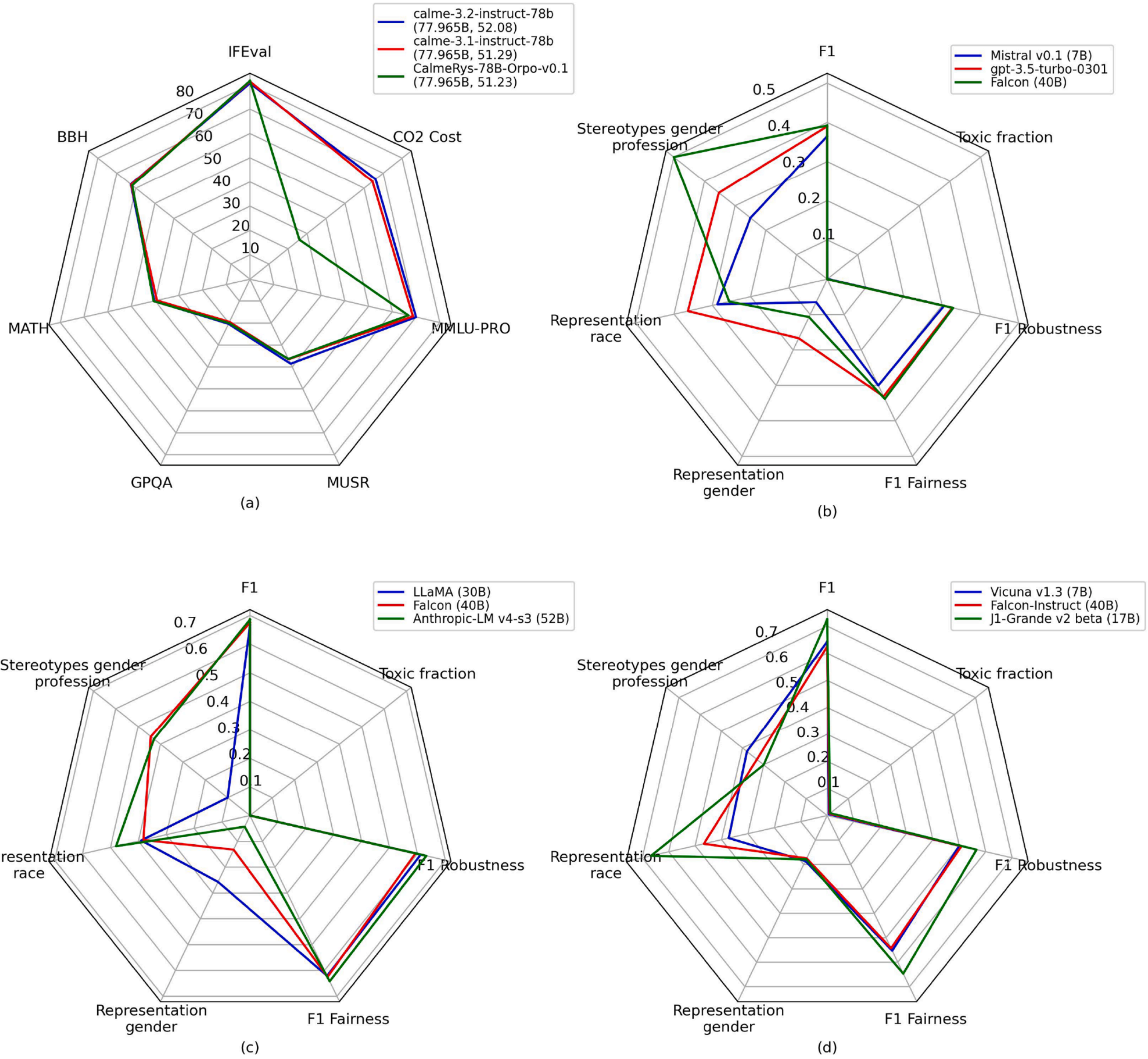


Fig. E.1. Original scores of the three LLMs with the highest stable rankings achieved on the different criteria, shown by dataset: (a) Open LLM Leaderboard; (b) HELM NaturalQuestions (closed-book); (c) HELM NaturalQuestions (open-book); (d) HELM NarrativeQA.

References

[1] W.X. Zhao, K. Zhou, J. Li, T. Tang, X. Wang, Y. Hou, Y. Min, B. Zhang, J. Zhang, Z. Dong, et al., A survey of large language models, (2023) arXiv:2303.18223

[2] Y. Chang, X. Wang, J. Wang, Y. Wu, L. Yang, K. Zhu, H. Chen, X. Yi, C. Wang, Y. Wang, et al., A survey on evaluation of large language models, ACM Trans. Intell. Syst. Technol. 15 (3) (2024) 1–45.

[3] L. Huang, W. Yu, W. Ma, W. Zhong, Z. Feng, H. Wang, Q. Chen, W. Peng, X. Feng, B. Qin, et al., A survey on hallucination in large language models: Principles, taxonomy, challenges, and open questions, ACM Trans. Inf. Syst. 43 (2) (2025) 1–55.

[4] T. Eftimov, G. Petelin, G. Cenikj, A. Kostovska, G. Ispirova, P. Korošec, J. Bogatinovski, Less is more: Selecting the right benchmarking set of data for time series classification, Expert Syst. Appl. 198 (2022) 116871.

[5] G. Cenikj, R.D. Lang, A.P. Engelbrecht, C. Doerr, P. Korošec, T. Eftimov, Selector: selecting a representative benchmark suite for reproducible statistical comparison, in: Proceedings of The Genetic and Evolutionary Computation Conference, 2022, pp. 620–629.

[6] U. Škvorc, T. Eftimov, P. Korošec, Understanding the problem space in single-objective numerical optimization using exploratory landscape analysis, Applied Soft Comput. 90 (2020) 106138.

[7] R. Kohli, M. Feurer, K. Eggensperger, B. Bischl, F. Hutter, Towards quantifying the effect of datasets for benchmarking: A look at tabular machine learning, in: ICLR Workshop, 2, 2024, p. 6.

[8] G. Ispirova, T. Eftimov, S. Džeroski, B. Koroušić Seljak, MsGEN: Measuring generalization of nutrient value prediction across different recipe datasets, Expert Syst. Appl. 237 (2024) 121507.

[9] C. Benjamins, G. Cenikj, A. Nikolikj, A. Mohan, T. Eftimov, M. Lindauer, Instance selection for dynamic algorithm configuration with reinforcement learning: improving generalization, (2024) arXiv:2407.13513

[10] Y. Perlitz, A. Gera, O. Arviv, A. Yehudai, E. Bandel, E. Shnarch, M. Shmueli-Scheuer, L. Choshen, Benchmark agreement testing done right: a guide for LLM benchmark evaluation, (2024) arXiv:2407.13696

[11] T. Hu, X.-H. Zhou, Unveiling LLM evaluation focused on metrics: challenges and solutions, (2024) arXiv:2404.09135

[12] Z. Chu, Z. Wang, W. Zhang, Fairness in large language models: A taxonomic survey, ACM SIGKDD Explorations Newsletter 26 (1) (2024) 34–48.

[13] L. Lin, L. Wang, J. Guo, K.-F. Wong, Investigating Bias in LLM-based bias detection: disparities between LLMs and human perception, (2024) arXiv:2403.14896

[14] Y. Liu, J. Cao, C. Liu, K. Ding, L. Jin, Datasets for large language models: a comprehensive survey, (2024) arXiv:2402.18041

[15] A. Gjorgjevikj, K. Mishev, L. Antovski, D. Trajanov, Requirements engineering in machine learning projects, IEEE Access 11 (2023) 72186–72208.

[16] C. Fourrier, N. Habib, A. Lozovskaya, K. Szafer, T. Wolf, Performances are plateauing, let's make the leaderboard steep again, 2024, ([Online]. Available: https://huggingface.co/spaces/open-llm-leaderboard/blog).

[17] P. Liang, R. Bommasani, T. Lee, D. Tsipras, D. Soylu, M. Yasunaga, Y. Zhang, D. Narayanan, Y. Wu, A. Kumar, et al., Holistic evaluation of language models, (2022) arXiv:2211.09110

[18] C. Fourrier, N. Habib, A. Lozovskaya, K. Szafer, T. Wolf, Open LLM Leaderboard v2, 2024, ([Online]. Available: https://huggingface.co/spaces/open-llm-leaderboard/open_llm_leaderboard).

[19] A. Gjorgjevikj, A. Nikolikj, B. Koroušić Seljak, T. Eftimov, xLLMBench, 2025, ([Online]. Available: https://github.com/gjorgjevik/xLLMBench).

[20] D. Hendrycks, C. Burns, S. Kadavath, A. Arora, S. Basart, E. Tang, D. Song, J. Steinhardt, Measuring mathematical problem solving with the math dataset, (2021) arXiv:2103.03874

[21] Q. Zhong, K. Wang, Z. Xu, J. Liu, L. Ding, B. Du, D. Tao, Achieving> 97
% on GSM8K: deeply understanding the problems makes LLMs perfect reasoners, (2024) arXiv:2404.14963

[22] J. Novikova, O. Dušek, V. Rieser, The E2E dataset: new challenges for end-to-end generation, (2017) arXiv:1706.09254

[23] D. Li, L. Murr, HumanEval on latest GPT models–2024, (2024) arXiv:2402.14852

[24] Y. Wang, X. Ma, G. Zhang, Y. Ni, A. Chandra, S. Guo, W. Ren, A. Arulraj, X. He, Z. Jiang, et al., MMLU-pro: a more robust and challenging multi-task language understanding benchmark, (2024) arXiv:2406.01574

[25] W. Zhong, R. Cui, Y. Guo, Y. Liang, S. Lu, Y. Wang, A. Saied, W. Chen, N. Duan, Agieval: A human-centric benchmark for evaluating foundation models, (2023) arXiv:2304.06364

[26] D. Rein, B.L. Hou, A.C. Stickland, J. Petty, R.Y. Pang, J. Dirani, J. Michael, S.R. Bowman, GPQA: a graduate-level google-proof q&a benchmark, (2023) arXiv:2311.12022

[27] I. Fujisawa, S. Nobe, H. Seto, R. Onda, Y. Uchida, H. Ikoma, P.-C. Chien, R. Kanai, ProcBench: benchmark for multi-step reasoning and following procedure, (2024) arXiv:2410.03117

[28] G. Bai, J. Liu, X. Bu, Y. He, J. Liu, Z. Zhou, Z. Lin, W. Su, T. Ge, B. Zheng, et al., Mt-bench-101: a fine-grained benchmark for evaluating large language models in multi-turn dialogues, (2024) arXiv:2402.14762

[29] W.-L. Chiang, L. Zheng, Y. Sheng, A.N. Angelopoulos, T. Li, D. Li, H. Zhang, B. Zhu, M. Jordan, J.E. Gonzalez, et al., Chatbot arena: an open platform for evaluating llms by human preference, (2024) arXiv:2403.04132

[30] D. Kiela, M. Bartolo, Y. Nie, D. Kaushik, A. Geiger, Z. Wu, B. Vidgen, G. Prasad, A. Singh, P. Ringshia, et al., Dynabench: rethinking benchmarking in NLP, (2021) arXiv:2104.14337

[31] C. White, S. Dooley, M. Roberts, A. Pal, B. Feuer, S. Jain, R. Shwartz-Ziv, N. Jain, K. Saifullah, S. Naidu, et al., LiveBench: a challenging, contamination-Free LLM Benchmark, (2024) arXiv:2406.19314

[32] S. Chen, P. Pusarla, B. Ray, DyCodeEval: Dynamic Benchmarking of Reasoning Capabilities in Code Large Language Models Under Data Contamination, in: Forty-second International Conference on Machine Learning, 2025. https://openreview.net/forum?id=3BZyQqbytZ.

[33] V. Majdinasab, A. Nikanjam, F. Khomh, Prism: dynamic and flexible benchmarking of LLMs code generation with Monte Carlo tree search, (2025) arXiv:2504.05500

[34] A. Srivastava, A. Rastogi, A. Rao, A.A.M. Shoeb, A. Abid, A. Fisch, A.R. Brown, A. Santoro, A. Gupta, A. Garriga-Alonso, et al., Beyond the imitation game: Quantifying and extrapolating the capabilities of language models, (2022) arXiv:2206.04615

[35] M. Suzgun, N. Scales, N. Schärli, S. Gehrmann, Y. Tay, H.W. Chung, A. Chowdhery, Q.V. Le, E.H. Chi, D. Zhou, et al., Challenging big-bench tasks and whether chain-of-thought can solve them, (2022) arXiv:2210.09261

[36] Z. Sprague, X. Ye, K. Bostrom, S. Chaudhuri, G. Durrett, Musr: Testing the limits of chain-of-thought with multistep soft reasoning, arXiv:2310.16049 (2023).

[37] J. Zhou, T. Lu, S. Mishra, S. Brahma, S. Basu, Y. Luan, D. Zhou, L. Hou, Instruction-following evaluation for large language models, (2023) arXiv:2311.07911

[38] W. Sun, J. Wang, Q. Guo, Z. Li, W. Wang, R. Hai, CEBench: A Benchmarking Toolkit for the Cost-Effectiveness of LLM Pipelines, (2024) arXiv:2407.12797

[39] D.R. Hunter, MM algorithms for generalized Bradley-Terry models, The annals of statistics 32 (1) (2004) 384–406.

[40] M. Grootendorst, BERTopic: neural topic modeling with a class-based TF-IDF procedure, (2022) arXiv:2203.05794

[41] R. Zhao, W. Zhang, Y.K. Chia, D. Zhao, L. Bing, Auto Arena of LLMs: automating LLM evaluations with agent peer-battles and committee discussions, (2024) arXiv:2405.20267

[42] G. Demir, P. Chatterjee, D. Pamucar, Sensitivity analysis in multi-criteria decision making: A state-of-the-art research perspective using bibliometric analysis, Expert Syst. Appl. 237 (2024) 121660.

[43] W. Sałabun, J. Wątróbski, A. Shekhovtsov, Are mcda methods benchmarkable? a comparative study of topsis, vikor, copras, and promethee ii methods, Symmetry 12 (9) (2020) 1549.

[44] J.-P. Brans, R. Nadeau, M. Landry, L'ingénierie de la décision, Elaboration d'instruments d'aide à la décision. La méthode PROMETHEE. In l'Aide à la Décision: Nature, Instruments et Perspectives d'Avenir (1982) 183–213.

[45] M. Behzadian, R.B. Kazemzadeh, A. Albadvi, M. Aghdasi, PROMETHEE: A comprehensive literature review on methodologies and applications, Eur. J. Oper. Res. 200 (1) (2010) 198–215.

[46] H.M. Alabool, Large Language Model Evaluation Criteria Framework in Healthcare: Fuzzy MCDM Approach, SN Comput. Sci. 6 (1) (2025) 1–28.

[47] S.-J. Chen, C.-L. Hwang, Fuzzy multiple attribute decision making methods, in: Fuzzy multiple attribute decision making: Methods and applications, Springer, 1992, pp. 289–486.

[48] S. Opricovic, Multicriteria optimization of civil engineering systems, Faculty civil Eng. Belgrade 2 (1) (1998) 5–21.

[49] High-Level Expert Group on AI, European Commission, Ethics guidelines for trustworthy AI, 2019.

[50] M. Gul, E. Celik, A.T. Gumus, A.F. Guneri, A fuzzy logic based PROMETHEE method for material selection problems, Beni-Suef University J. Basic Appl. Sci. 7 (1) (2018) 68–79.

[51] T. Kočiskỳ, J. Schwarz, P. Blunsom, C. Dyer, K.M. Hermann, G. Melis, E. Grefenstette, The narrativeqa reading comprehension challenge, Trans. Assoc. Comput. Linguistics 6 (2018) 317–328.

[52] T. Kwiatkowski, J. Palomaki, O. Redfield, M. Collins, A. Parikh, C. Alberti, D. Epstein, I. Polosukhin, J. Devlin, K. Lee, et al., Natural questions: a benchmark for question answering research, Trans. Assoc. Comput. Linguistics 7 (2019) 453–466.

[53] R.W. Saaty, The analytic hierarchy process–what it is and how it is used, Math. Model. 9 (3-5) (1987) 161–176.

[54] J. Demšar, Statistical comparisons of classifiers over multiple data sets, J. Mach. Learn. Res. 7 (2006) 1–30.

[55] Y. Perlitz, E. Bandel, A. Gera, O. Arviv, L. Ein-Dor, E. Shnarch, N. Slonim, M. Shmueli-Scheuer, L. Choshen, Efficient benchmarking of language models, (2023) arXiv:2308.11696

[56] F.M. Polo, L. Weber, L. Choshen, Y. Sun, G. Xu, M. Yurochkin, tinyBenchmarks: evaluating LLMs with fewer examples, (2024) arXiv:2402.14992

[57] V. Su, N. Thakur, COVID-19 on YouTube: a data-driven analysis of sentiment, toxicity, and content recommendations, in: 2025 IEEE 15th Annual Computing and Communication Workshop and Conference (CCWC), IEEE, 2025, pp. 00715–00723.

[58] A.K. Mishra, R. Sharma, J. Singh, P. Singh, M. Diwakar, M. Tiwari, Real-time health monitoring of patients in emergencies through machine learning and iot integrated with edge computing, in: 2024 14th International Conference on Cloud Computing, Data Science & Engineering (Confluence), IEEE, 2024, pp. 285–290.

[59] Z. Wang, Y. Zhou, H. Liu, J. Qiu, B. Fang, Z. Tian, ThreatInsight: Innovating Early Threat Detection through Threat-Intelligence-Driven Analysis and Attribution, IEEE Trans. Knowl. Data Eng. 36 (12) (2024) 9388–9402.

[60] Y. Nie, Y. Kong, X. Dong, J.M. Mulvey, H.V. Poor, Q. Wen, S. Zohren, A survey of large language models for financial applications: Progress, prospects and challenges, (2024) arXiv:2406.11903

[61] K. Mishev, A. Gjorgjevikj, I. Vodenska, L.T. Chitkushev, D. Trajanov, Evaluation of sentiment analysis in finance: from lexicons to transformers, IEEE Access 8 (2020) 131662–131682.

[62] Q. Xie, W. Han, Z. Chen, R. Xiang, X. Zhang, Y. He, M. Xiao, D. Li, Y. Dai, D. Feng, et al., Finben: a holistic financial benchmark for large language models, Adv. Neural Inf. Process. Syst. 37 (2024) 95716–95743.

[63] J. Lee, N. Stevens, S.C. Han, Large language models in finance (finllms), Neural Comput. Appl. (2025) 1–15.

[64] P. Islam, A. Kannappan, D. Kiela, R. Qian, N. Scherrer, B. Vidgen, Financebench: a new benchmark for financial question answering, (2023) arXiv:2311.11944

[65] Z. Chen, W. Chen, C. Smiley, S. Shah, I. Borova, D. Langdon, R. Moussa, M. Beane, T.-H. Huang, B. Routledge, et al., Finqa: a dataset of numerical reasoning over financial data, (2021) arXiv:2109.00122

[66] J. Haltaufderheide, R. Ranisch, The ethics of ChatGPT in medicine and healthcare: a systematic review on Large Language Models (LLMs), NPJ Digital Medicine 7 (1) (2024) 183.

[67] M. Raza, Z. Jahangir, M.B. Riaz, M.J. Saeed, M.A. Sattar, Industrial applications of large language models, Sci. Rep. 15 (1) (2025) 13755.

[68] A.J. Thirunavukarasu, D.S.J. Ting, K. Elangovan, L. Gutierrez, T.F. Tan, D.S.W. Ting, Large language models in medicine, Nature Medicine 29 (8) (2023) 1930–1940.

[69] S. Reddy, Generative AI in healthcare: an implementation science informed translational path on application, integration and governance, Implement. Sci. 19 (1) (2024) 27.

[70] S. Bedi, H. Cui, M. Fuentes, A. Unell, M. Wornow, J.M. Banda, N. Kotecha, T. Keyes, Y. Mai, M. Oez, et al., MedHELM: holistic evaluation of large language models for medical tasks, (2025) arXiv:2505.23802

[71] S. Ren, B. Tomlinson, R.W. Black, A.W. Torrance, Reconciling the contrasting narratives on the environmental impact of large language models, Sci. Rep. 14 (1) (2024) 26310.

[72] A.A. Chien, L. Lin, H. Nguyen, V. Rao, T. Sharma, R. Wijayawardana, Reducing the Carbon Impact of Generative AI Inference (today and in 2035), in: Proceedings of the 2nd workshop on sustainable computer systems, 2023, pp. 1–7.

[73] S. Nguyen, B. Zhou, Y. Ding, S. Liu, Towards sustainable large language model serving, ACM SIGENERGY Energy Inf. Rev. 4 (5) (2024) 134–140.

[74] D. Adalja, K. Kalita, L. Čepová, P. Patel, N. Mashru, P. Jangir, et al., Advancing Truss Structure Optimization–A Multi-Objective Weighted Average Algorithm with Enhanced Convergence and Diversity, Results Eng. (2025) 104241.

[75] P. Jangir, Arpita, S.P. Agrawal, S.B. Pandya, A. Parmar, S. Kumar, G.G. Tejani, L. Abualigah, A cooperative strategy-based differential evolution algorithm for robust PEM fuel cell parameter estimation, Ionics 31 (1) (2025) 703–741.

[76] C. Aranha, C.L. Camacho Villalón, F. Campelo, M. Dorigo, R. Ruiz, M. Sevaux, K. Sörensen, T. Stützle, Metaphor-based metaheuristics, a call for action: the elephant in the room, Swarm Intell. 16 (1) (2022) 1–6.

[77] B. Ceballos, M.T. Lamata, D.A. Pelta, A comparative analysis of multi-criteria decision-making methods, Progress Artif. Intell. 5 (2016) 315–322.

[78] H. Taherdoost, M. Madanchian, Multi-criteria decision making (MCDM) methods and concepts, Encyclopedia 3 (1) (2023) 77–87.

[79] S. Opricovic, G.-H. Tzeng, Compromise solution by MCDM methods: a comparative analysis of VIKOR and TOPSIS, Eur. Journal Oper. Res. 156 (2) (2004) 445–455.

[80] B. Kizielewicz, A. Shekhovtsov, W. Sałabun, pymcdm–The universal library for solving multi-criteria decision-making problems, SoftwareX 22 (2023) 101368.

[81] O.S. Vaidya, S. Kumar, Analytic hierarchy process: An overview of applications, Eur. J. Oper. Res. 169 (1) (2006) 1–29.

[82] Z. Babic, N. Plazibat, Ranking of enterprises based on multicriterial analysis, International journal of production economics 56 (1998) 29–35.

[83] C. Macharis, J. Springael, K. De Brucker, A. Verbeke, PROMETHEE and AHP: The design of operational synergies in multicriteria analysis.: Strengthening PROMETHEE with ideas of AHP, Eur. J. Oper. Res. 153 (2) (2004) 307–317.