LLMs-as-Judges: A Comprehensive Survey on LLM-based Evaluation Methods

HAITAO LI, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China

QIAN DONG, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China

JUNJIE CHEN, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China

HUIXUE SU, Gaoling School of Artificial Intelligence, Renmin University of China, China

YUJIA ZHOU, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China

QINGYAO AI, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China

ZIYI YE, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China

YIQUN LIU, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, China

The rapid advancement of Large Language Models (LLMs) has driven their expanding application across various fields. One of the most promising applications is their role as evaluators based on natural language responses, referred to as "LLMs-as-judges". This framework has attracted growing attention from both academia and industry due to their excellent effectiveness, ability to generalize across tasks, and interpretability in the form of natural language. This paper presents a comprehensive survey of the LLMs-as-judges paradigm from five key perspectives: **Functionality**, **Methodology**, **Applications**, **Meta-evaluation**, and **Limitations**. We begin by providing a systematic definition of LLMs-as-Judges and introduce their functionality (Why use LLM judges?). Then we address methodology to construct an evaluation system with LLMs (How to use LLM judges?). Additionally, we investigate the potential domains for their application (Where to use LLM judges?) and discuss methods for evaluating them in various contexts (How to evaluate LLM judges?). Finally, we provide a detailed analysis of the limitations of LLM judges and discuss potential future directions.

Authors' addresses: Haitao Li, liht22@mails.tsinghua.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Qian Dong, dq22@mails.tsinghua.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Junjie Chen, chenjj826@gmail.com, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Huixue Su, suhuixue@ruc.edu.cn, Gaoling School of Artificial Intelligence, Renmin University of China, Beijing, China; Yujia Zhou, suhuixue@ruc.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Qingyao Ai, aiqy@tsinghua.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Ziyi Ye, yeziyi1998@gmail.com, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China; Yiqun Liu, yiqunliu@tsinghua.edu.cn, Department of Computer Science and Technology, Institute for Internet Judiciary, Tsinghua University, Beijing, China.

Through a structured and comprehensive analysis, we aim aims to provide insights on the development and application of LLMs-as-judges in both research and practice. We will continue to maintain the relevant resource list at https://github.com/CSHaitao/Awesome-LLMs-as-Judges.

1 INTRODUCTION

Studies on evaluation methods have long been a key force in guiding the development of modern Artificial Intelligence (AI) [23]. AI researchers have continuously sought to measure and validate the intelligence of AI models through various tasks [23, 75]. In the mid-20th century, AI evaluation primarily centered on assessing algorithm performance in specific tasks, such as logical reasoning and numerical computation [164]. Traditional machine learning tasks like classification and regression often use programmable and statistical metrics, including accuracy, precision, and recall. With the emergence of deep learning, the complexity of AI systems grew rapidly, prompting a shift in evaluation standards [119]. The evaluation of AI has expanded from pre-defined, programmable machine metrics to more flexible, robust evaluators for solving complex, realistic tasks. A typical example is the Turing Test [67, 222], which determines whether an AI model can exhibit human-like intelligent behavior through dialogue with humans. The Turing Test provides a fundamental guideline in the evaluation of AI models, especially on AI models' intelligence in flexible and realistic environments.

Recently, the emergence of Large Language Models (LLMs) and generative AI serves as a new milestone in the evolution of AI evaluation. LLMs exhibit remarkable generalization and adaptability, showcasing strong transfer capabilities across previously unseen tasks and diverse domains [1, 9]. However, their powerful capabilities also present new challenges for evaluation. Due to the highly generative and open-ended nature of their outputs, standardized metrics are often insufficient for a comprehensive evaluation. For example, in natural language generation (NLG) tasks, traditional metrics like BLEU [173] and ROUGE [139] often fail to capture key aspects such as text fluency, logical coherence, and creativity. Moreover, modern AI evaluation extends beyond task performance and must account for the ability to address complex, dynamic problems in real-world scenarios, including robustness, fairness, and interpretability. Human annotations, frequently regarded as the "ground truth," can offer comprehensive insights and valuable feedback. By gathering responses from experts or users, researchers can gain a deeper understanding of a model's performance, practicality, and potential risks. However, collecting them are typically time-consuming and resource-intensive, making it challenging to scale up for large-scale evaluation.

In this context, a new paradigm has emerged to replace humans and statistical metrics with LLMs in evaluation, referred to as LLMs-as-judges [5, 14, 14, 221]. Compared to traditional evaluation methods, LLMs-as-judges show significant strengths. First, LLM judges can adjust their evaluation criteria based on the specific task context, rather than relying on a fixed set of metrics, making the evaluation process more flexible and refined. Second, LLM judges can generate interpretive evaluations, offering more comprehensive feedback on model performance and enabling researchers to gain deeper insights into the evaluater's strengths and weaknesses. Finally, LLM judges offer a scalable and reproducible alternative to human evaluation, significantly reducing the costs and time associated with human involvement.

Despite its great potential and significant advantages, LLMs-as-judges also face several critical challenges. For example, the evaluation results of LLMs are often influenced by the prompt template, which can lead to biased or inconsistent assessments [260]. Considering that LLMs are trained on extensive text corpus, they may also inherit various implicit biases, impacting the fairness and reliability of their assessments [267]. Moreover, distinct tasks and domains require specific evaluation criteria, making it difficult for LLMs to adapt their standards dynamically to specific contexts.

Considering the vast potential of this field, this survey aims to systematically review and analyze the current state and key challenges of the LLMs-as-judges. As shown in Figures 1 and 2, we discuss existing research across five key perspectives: 1) **Functionality**: Why use LLM judges, 2) **Methodology**: How to use LLM judges, 3) **Application**: Where to use LLM judges, 4) **Meta-evaluation**: How to evaluate LLM judges and 5) **Limitation**: Existing problems of LLM judges. We explore the key challenges confronting LLMs-as-judges and hope to provide a clearer guideline for their future development.

In summary, the main contributions of this paper are as follows:

(1) **Comprehensive and Timely Survey**: We present the extensive survey on the emerging paradigm of LLMs-as-judges, systematically reviewing the current state of research and developments in this field. By examining LLMs as performance evaluators based on their generated natural language, we highlight the unique role of LLMs in shaping the future of AI evaluation.

(2) **Systematic Analysis Across Five Key Perspectives**: We organize our survey around five critical aspects: Functionality, Methodology, Application, Meta-evaluation, and Limitation. This structured approach allows for a nuanced understanding of how and why LLMs are utilized as evaluators, their practical implementations, and reliability concerns.

(3) **Current Challenges and Future Research Directions**: We discuss the existing challenges for adopting LLMs-as-judges, highlighting potential research opportunities and directions while offering a forward-looking perspective on the future development of this paradigm, encouraging researchers to delve deeper into this exciting area. We also provide an open-source repository at https://github.com/CSHaitao/Awesome-LLMs-as-Judges, with the goal of fostering a collaborative community and advancing best practices in this area.

The organization of this paper is as follows. In Section (§2), we provide the formal definition of LLMs-as-judges. Then, Section (§3) reviews existing work from the perspective of "Why use LLM judges". Following that, Section (§4) covers "How to use LLM judges", summarizing the current technical developments in LLMs-as-judges. Section (§5) discusses "Where to use LLM judges", focusing on their application domains. In Section (§6), we review the metrics and benchmarks used for evaluating LLMs-as-judges. Section (§7) discusses the limitations and challenges of LLM judges. We discuss major future work in Sections (§8) and (§9) to conclude the paper.

2 PRELIMINARIES

In this section, we will provide a formal definition of LLMs-as-judges, aiming to encompass all current evaluation paradigms and methods, thereby offering readers a clear and thorough understanding. Figure 3 presents an overview of the LLMs-as-judges system.

The LLMs-as-judges paradigm is a flexible and powerful evaluation framework where LLMs are employed as evaluative tools, responsible for assessing the quality, relevance, and effectiveness of generated outputs based on defined evaluation criteria. This framework leverages the extensive knowledge and deep contextual understanding of LLMs, enabling it to flexibly adapt to various
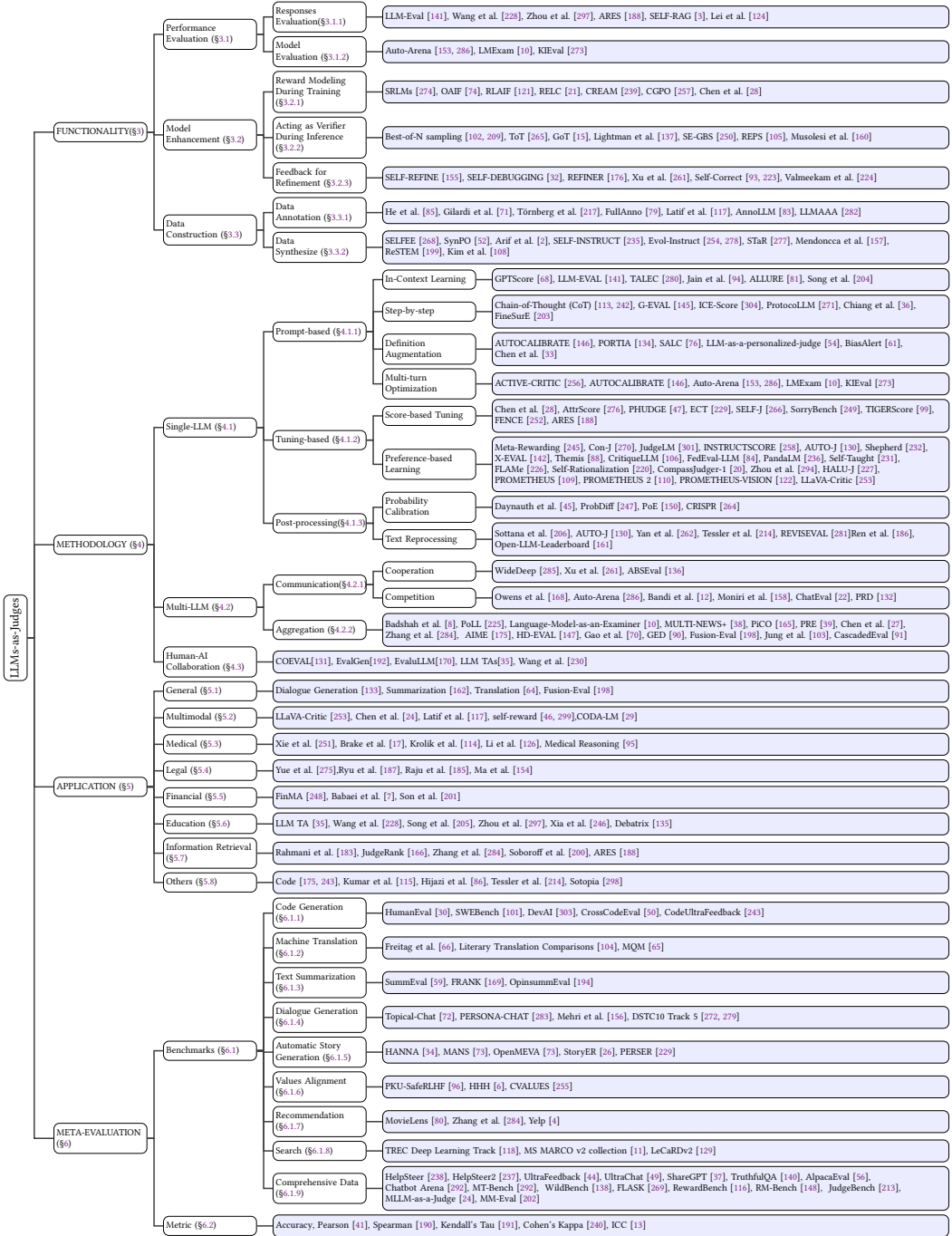
Fig. 1. Taxonomy of LLMs-as-judges in functionality, methodology, application, meta-evaluation.

tasks in NLP and machine learning. We formalize the input-output structure of the LLMs-as-Judges paradigm, unifying various evaluation scenarios into a unified perspective. Specifically, the
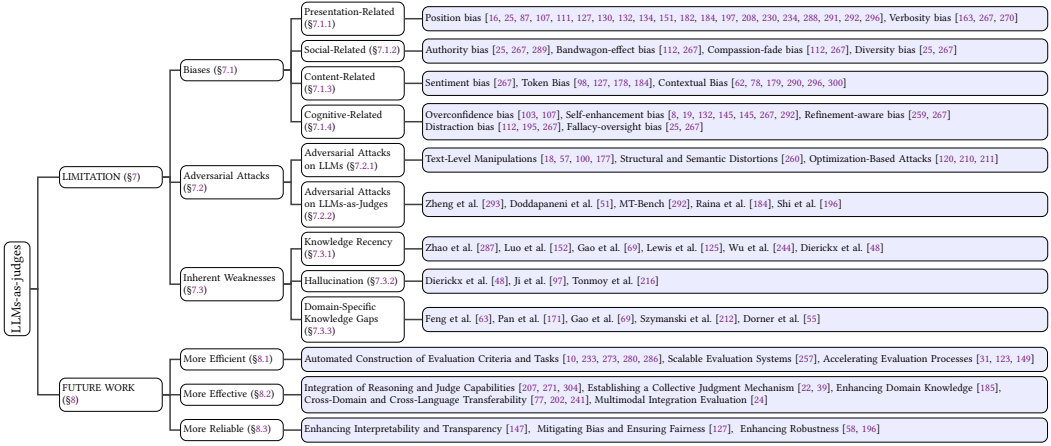
Fig. 2. Taxonomy of LLMs-as-judges in limitation and future work.

evaluation process can be defined as follows:

$$(\mathcal{Y}, \mathcal{E}, \mathcal{F}) = E(\mathcal{T}, \mathcal{C}, \mathcal{X}, \mathcal{R}) \tag{1}$$

where E is the evaluation function, taking the evaluation type $\mathcal{T}$, evaluation criteria $\mathcal{C}$, evaluation item $\mathcal{X}$ and optional references $\mathcal{R}$ as input. Based on these inputs, the LLM can produces three outputs: evaluation result $\mathcal{Y}$, explanation $\mathcal{E}$ and feedback $\mathcal{F}$. Different input-output configurations correspond to distinct methods and objectives. This unified formulation brings together diverse evaluation paradigms, offering a structured framework for categorizing and understanding various approaches within LLMs-as-judges.

2.1 Evaluation Function E

The evaluation function E in the context of LLMs-as-judges can be categorized into three primary configurations: Single-LLM systems, Multi-LLM systems, and Hybrid systems that combine LLMs with human evaluators. Each of these configurations serves distinct purposes, offers different advantages, and faces unique challenges.

- **Single-LLM Evaluation System [141, 145, 280]:** A single LLM evaluation system relies on a single model to perform the evaluation tasks. It is simple to deploy and scale, making it efficient for tasks that don't require specialized evaluation. However, its flexibility is limited, as it may struggle with tasks that demand specialized knowledge or reasoning over complex inputs. Additionally, if not properly trained, a single model may introduce biases, leading to inaccurate evaluations.
- **Multi-LLM Evaluation Systems [22, 39, 132]:** A Multi-LLM evaluation system combines multiple models that work together to perform evaluation tasks. These models may interact through various mechanisms, such as collaboration, or competition, to refine their outputs and achieve more accurate results. By leveraging the strengths of different models, a multi-model system can cover a broader range of evaluation criteria and provide a more comprehensive assessment. However, this comes at a higher computational cost and requires more resources, making deployment and maintenance more challenging, particularly for large-scale tasks. Moreover, while cooperation between models often enhances evaluation results, the methods through which these models achieve consensus or resolve differences remain key areas of ongoing exploration.
- **Human-AI Collaboration System [131, 192, 230]:** In this system, LLMs work alongside human evaluators, combining the efficiency of automated evaluation with the nuanced judgment

of human expertise. This configuration allows human evaluators to mitigate potential biases in the LLM's output and provide subjective insights into complex evaluation tasks. While this system offers greater reliability and depth, it comes with challenges in coordinating between the models and humans, ensuring consistent evaluation standards, and integrating feedback. Additionally, the inclusion of human evaluators increases both the cost and time required for the evaluation process, making it less scalable than purely model-based systems.

2.2 Evaluation Input

In the LLMs-as-judges paradigm, in addition to the evaluation item $\mathcal{X}$, LLM judges typically receive three other types of inputs: Evaluation Type $\mathcal{T}$, Evaluation Criteria $\mathcal{C}$, and Evaluation References $\mathcal{R}$. The following provides a detailed explanation:

2.2.1 Evaluation Type $\mathcal{T}$. The Evaluation Type $\mathcal{T}$ defines the specific evaluation mode, determining how the evaluation will be conducted. It typically includes three approaches: pointwise, pairwise, and listwise evaluation.

- **Pointwise Evaluation [109, 229, 266]:** This method evaluates each candidate item individually based on the specified criteria. For example, in a text summarization task, the LLM might evaluate each generated summary separately, assigning a score based on factors like informativeness, coherence, and conciseness. Although pointwise evaluation is simple and easy to apply, it may fail to capture the relative quality differences between candidates and can be influenced by biases arising from evaluating items in isolation.
- **Pairwise Evaluation [20, 84, 89, 188]:** This method involves directly comparing two candidate items to determine which one performs better according to the specified criteria. It is commonly used in preference-based tasks. For example, given two summaries of a news article, the LLM may be asked to decide which summary is more coherent or informative. Pairwise evaluation closely mirrors human decision-making processes by focusing on relative preferences rather than assigning absolute scores. This approach is especially effective when the differences between outputs are subtle and difficult to quantify.
- **Listwise Evaluation [87, 166, 263, 302]:** This method is designed to collectively evaluate the entire list of candidate items, evaluating and ranking them based on the specific criteria. It is often applied in ranking tasks, such as document retrieval in search engines, where the objective is to determine the relevance of the documents in relation to a user query. Listwise evaluation takes into account the interactions between multiple candidates, making it well-suited for applications that require holistic analysis.

In general, these three evaluation modes are not entirely independent. pointwise scores can be aggregated to create pairwise comparisons or used to construct a ranked list. Similarly, pairwise preferences can be organized into a complete ranking list for listwise analysis. However, these transformations are not always reliable within the LLMs-as-judges framework [149]. For example, in pointwise evaluation, output A may receive a score of 5, while output B receives a score of 4, yet, a direct pairwise comparison might not consistently yield $A > B$ due to potential bias. Additionally, LLM judges do not always satisfy transitivity in their judgments. For instance, given pairwise preferences where $z_i > z_j$ and $z_j > z_k$, the LLM may not necessarily yield $z_i > z_k$. These inconsistencies contribute to concerns about the reliability and trustworthiness of the LLM-as-Judge framework, which we will discuss in detail in Section (§7).

2.2.2 Evaluation Criteria $\mathcal{C}$. The evaluation criteria $\mathcal{C}$ define the specific standards that determine which aspects of the output should be assessed. These criteria are designed to cover a broad range
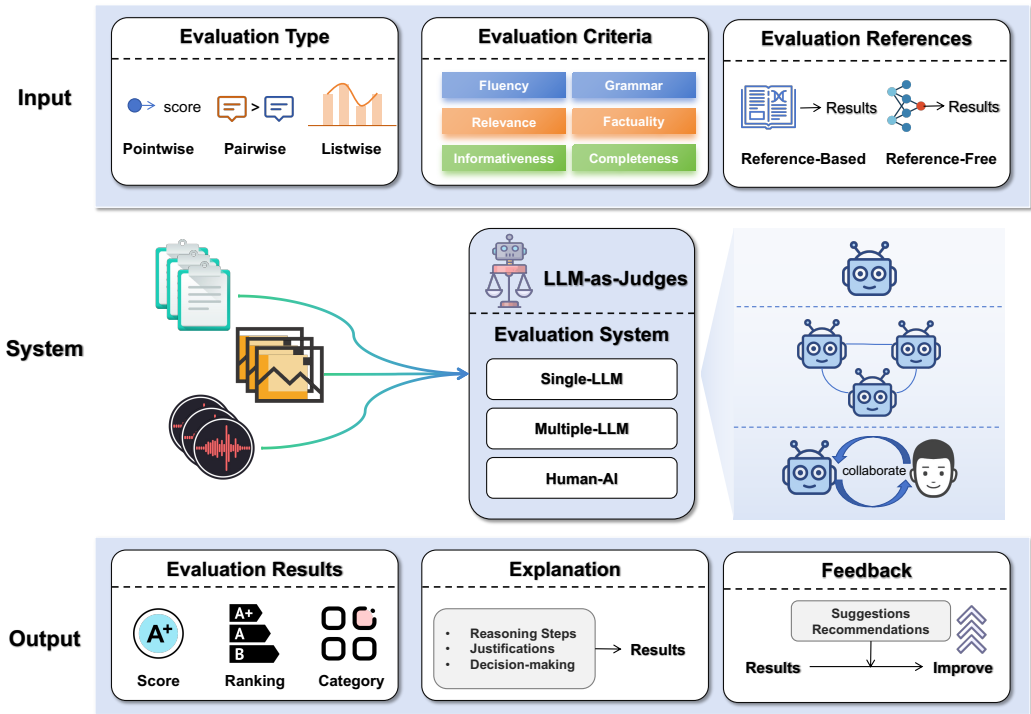
Fig. 3. Overview of the LLMs-as-judges system.

of quality attributes and can be tailored based on the nature of the task. Typically, the criteria encompass the following aspects:

- **Linguistic Quality [34, 59, 283]:** This category evaluates the language-related features of the output, such as fluency, grammatical accuracy, coherence, and Conciseness. Linguistic quality is crucial in tasks like text generation, machine translation, and summarization, where clarity and readability are essential.
- **Content Accuracy [30, 101, 213]:** This dimension focuses on the correctness and relevance of the content. It includes evaluating aspects such as factual accuracy, ensuring that the output does not contain misleading or incorrect information. Content accuracy is particularly crucial in tasks such as code generation and fact-checking.
- **Task-Specific Metrics [96, 138, 213]:** In addition to general quality metrics, many tasks require evaluation based on standards specific to their respective domains. These standards may include metrics such as informativeness (assessing whether the output provides comprehensive and valuable information) or completeness (ensuring all key aspects of the input are covered). Other criteria may include diversity, well-structured content, and logical clarity.

In addition to providing clear evaluation criteria, offering several examples can also be beneficial for the assessment. By incorporating well-structured examples, LLMs can better align its output with user expectations, especially when handling complex tasks or ambiguous queries.

2.2.3 Evaluation References $\mathcal{R}$. Evaluation References $\mathcal{R}$ are optional. Depending on the availability of evaluation reference, the evaluation process can be broadly divided into reference-based and reference-free scenarios.

- **Reference-Based Evaluation [66, 104]:** The reference-based evaluation leverages reference data to determine whether the performance meets the expected standards. It is commonly applied in tasks where the quality of the output can be objectively judged by its similarity to established reference. In Natural Language Generation (NLG) tasks, this method is widely used to evaluate the resemblance between generated content and reference content. For example, in machine translation or text summarization, an LLM can compare the generated translations or summaries against high-quality references. The key strength of this approach is its well-defined benchmarking process; however, its effectiveness may be constrained by the quality and variety of the reference data.
- **Reference-Free Evaluation [82, 194, 292]:** The reference-free evaluation does not rely on a specific reference $\mathcal{R}$, instead, it evaluates $\mathcal{X}$ based on intrinsic quality standards or its alignment with the source context. For example, when assessing language fluency or content coherence, an LLM can autonomously generate evaluation results using internal grammatical and semantic rules. This method is widely used in fields like sentiment analysis and dialogue generation. The main advantage of this approach is its independence from specific references, providing greater flexibility for open-ended tasks. However, its drawback lies in the difficulty of obtaining satisfactory evaluations in domains where the LLM lacks relevant knowledge.

2.3 Evaluation Output

In the LLMs-as-judges paradigm, the LLM typically generates three types of outputs: the evaluation result $\mathcal{Y}$, the explanation $\mathcal{E}$, and the feedback $\mathcal{F}$. Below are detailed descriptions.

- **Evaluation Result $\mathcal{Y}$ [188, 286]:** The evaluation result $\mathcal{Y}$ is the primary output, which can take the form of a numerical score, a ranking, a categorical label, or a qualitative assessment. It reflects the quality, relevance, or performance of the candidate items according to the specified evaluation criteria. For example, in a machine translation task, $\mathcal{Y}$ could be a score indicating translation quality, while in a dialogue generation task, it might be a rating of coherence and appropriateness on a scale from 1 to 5. The evaluation result $\mathcal{Y}$ provides a clear measure of performance, enabling researchers to effectively compare different models or outputs.
- **Explanation $\mathcal{E}$ [252, 270]:** The explanation $\mathcal{E}$ provides detailed reasoning and justifications for the evaluation result. It offers insights into why certain result received higher or lower scores, highlighting specific features of the candidate item that influenced the evaluation. For example, in a summarization task, the LLM judges might explain that the score was lowered due to missing critical information or the presence of redundant content. The explanation component enhances transparency, allowing users to understand the decision-making process of the LLM and gain deeper insights into the strengths and weaknesses of the evaluated content.
- **Feedback $\mathcal{F}$ [32, 155]:** The feedback $\mathcal{F}$ consists of actionable suggestions or recommendations aimed at improving the evaluated output. Unlike the evaluation result, which merely indicates performance, the feedback component is designed to guide the refinement of the content. For instance, in a creative writing task, feedback might include recommendations for enhancing the narrative flow or improving clarity. This component is especially valuable for the iterative development of the evaluated item, as it provides concrete pointers that help both the LLM and content creators enhance the quality of the generated outputs.

Depending on the intended purpose and specific requirements of the evaluation, the LLM judges can generate various combinations of the three outputs ($\mathcal{Y}$, $\mathcal{E}$, $\mathcal{F}$) for a given task. In most cases,
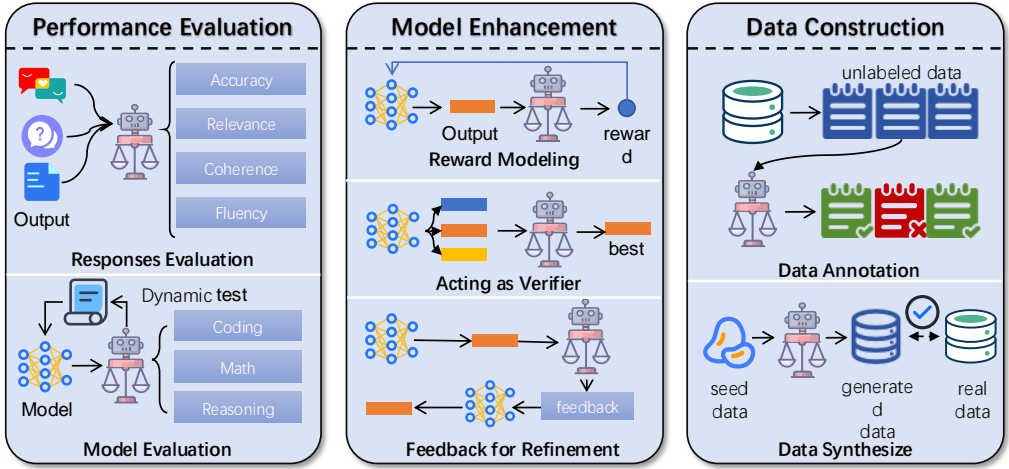
Fig. 4. Overview of the Functionality of LLMs-as-judges.

providing explanation $\mathcal{E}$ not only helps users better understand and trust the evaluation results but also leads to more human-aligned and accurate evaluation result $\mathcal{Y}$. Moreover, generating feedback $\mathcal{F}$ generally demands a higher level of model capability, as it requires not only assessing the quality of the input but also providing concrete, actionable recommendations for improvement.

3 FUNCTIONALITY

As an emerging evaluation paradigm, LLMs-as-judges play a significant role across various scenarios. Based on their functionality, we categorize the application of LLM evaluators into three main directions: **Performance Evaluation** (§3.1), **Model Enhancement** (§3.2), and **Data Construction** (§3.3). In this section, we will delve into these functionalities, explore their potential, and discuss specific implementation methods.

3.1 Performance Evaluation

Performance evaluation represents the most fundamental application objective of an LLM judges, serving as the cornerstone for understanding and optimizing their other function. It can be broadly divided into two components: **Responses Evaluation** (§3.1.1) and **Model Evaluation** (§3.1.2). Response Evaluation focuses on aspects such as the quality, relevance, and coherence, and fluency of the responses for a given task. In contrast, model evaluation takes a holistic approach, assessing the overall capabilities of LLMs. Although these two aspects are interconnected, they focus on different levels of analysis, providing multidimensional insights into performance.

3.1.1 Responses Evaluation. The purpose of evaluating responses is to identify better answers within the context of a specific question or task, which can enhance overall decision-making. These responses can originate from either AI models or humans. Evaluation criteria typically consider general attributes such as accuracy, relevance, coherence, and fluency. However, in practical applications, the evaluation of responses often requires customized metrics tailored to specific tasks. For instance, in the education domain, the focus may be more on the inspirational and educational value of the answers.

LLM judges have also been widely applied in the assessment of text response [141, 228, 297]. Lin et al. [141] propose LLM-Eval, a unified framework employing a single-prompt strategy to evaluate the performance of open-domain dialogue systems across multiple dimensions, including content, grammar, relevance, and appropriateness. Wang et al. [228] proposed an article scoring and feedback system tailored to different genres, such as essays, narratives, and question-answering articles. Using BERT and ChatGPT models, they enabled automated scoring and detailed feedback, showcasing the potential of LLMs in article evaluation. Moreover, Zhou et al. [297] conduct a detailed evaluation of whether LLMs can serve as reliable tools for automated paper review. Their findings indicate that current LLMs are still not sufficiently reliable for such tasks, particularly in scenarios requiring logical reasoning or a deep knowledge base.

Furthermore, the evaluation of a single response is not limited to assessing the quality of the final answer but can also extend to analyzing the response process [3, 124, 188]. For instance, this can include evaluating whether retrieval is necessary at a given step, the relevance of the retrieved documents, and the interpretability of the response. For example, ARES [188] uses LLM judges to evaluate RAG systems across three dimensions: Contextual Relevance, Answer Faithfulness, and Answer Relevance. Similarly, Asai et al. [3] proposed SELF-RAG, which employs reflective token to determine whether retrieval is required and to self-assess the quality of generated outputs. Lei et al. [124] introduced LLMs to evaluate the quality of generated explanations, demonstrating the effectiveness of LLMs in understanding and generating explanations for recommendation tasks.

3.1.2 Model Evaluation. Model evaluation typically begins with assessing individual responses and then extends to analyzing overall capabilities. This wider perspective aims to analyze the model's performance across various tasks or domains, such as coding ability, instruction-following proficiency, reasoning, and other specialized skills relevant to its intended applications.

A common and straightforward approach is to represent model performance using average performance on static benchmarks [138, 213, 292]. LLM judges assess the model's performance using a set of carefully designed metrics, which results in a performance ranking. This method is widely adopted due to its simplicity and comparability. For example, task sets can be designed to evaluate the model's knowledge coverage, reasoning depth, and language generation quality [116, 148, 202], or real-world scenarios can be simulated to assess the model's ability to handle complex situations [144, 219].

As the demand for evaluation increases, the evaluation process has gradually shifted from traditional static testing to more dynamic, interactive assessments [10, 273, 286]. LLMs-as-judges has pioneered this approach, similar to Chatbot Arena [292], a crowdsourced platform that collects anonymous votes on LLM performance and ranks them using Elo scores. Auto-Arena [153, 286] and LMExam [10] assess model capabilities by using LLMs as both question setters and evaluators. These frameworks innovatively combine diverse question generation, multi-turn question-answering evaluation, and a decentralized model-to-model evaluation mechanism, providing more detailed and granular performance assessments. Additionally, KIEval [273] introduces an LLM-driven "interactor" role, which evaluates the knowledge mastery and generation abilities of LLMs through dynamic multi-turn conversations. These dynamic evaluation methods effectively address data leakage and evaluation bias issues common in traditional benchmark tests.

3.2 Model Enhancement

In addition to Performance Evaluation, LLMs-as-judges is also widely used for Model Enhancement. From training to inference, LLMs-as-judges plays a key role in improving model performance. Its application in model enhancement offers a novel optimization pathway for artificial intelligence,

fostering the refinement and personalization of intelligent systems across a broader spectrum of real-world applications.

3.2.1 Reward Modeling During Training. A primary application of LLMs-as-judges is in reward modeling during training, particularly in reinforcement learning with feedback [21, 74, 239, 257, 274]. LLM judges assign scores to model outputs by evaluating them against human-defined criteria, guiding optimization toward desired behaviors. This ensures alignment with human values, improving the quality and relevance of the generated outputs and improving the effectiveness of LLMs in real-world tasks.

A series of works, such as SRLMs [274], OAIF [74], and RLAIF [121], have enabled LLMs to become their own reward models. This overcomes the traditional RLHF dependency on fixed reward models, allowing the model to iteratively reward and self-optimize, fostering self-evolution through continuous self-assessment. RELC [21] tackles the challenge of sparse rewards in traditional RL by introducing a Critic Language Model (Critic LM) to evaluate intermediate generation steps. This dense feedback at each step helps mitigate reward sparsity, offering more detailed guidance to the model during training.

However, using the same LLM for both policy generation and reward modeling can pose challenges in ensuring the accuracy of the rewards. This dual role setup may lead to accumulated biases and preference data noise, which can undermine the training effectiveness. To address this issue, CREAM [239] introduces cross-iteration consistency constraints to regulate the training process and prevent the model from learning unreliable preference data. This significantly enhances reward consistency and alignment performance. In addition, CGPO [257] groups tasks by category (such as dialogue, mathematical reasoning, safety, etc.) and uses "Mixed Judges" to assign a specific reward model to each task group. This ensures that the reward signals are closely aligned with the task objectives, thereby preventing conflicts between different goals.

3.2.2 Acting as Verifier During Inference. During inference, LLM judges serve as verifier, responsible for selecting the optimal response from multiple candidates [15, 15, 137, 160, 265]. By comparing the outputs based on various metrics, such as factual accuracy and reasoning consistency, they are able to identify the best fit for the given task or context, thereby optimizing the inference process or improving the quality of the generated results.

One of the simplest applications is Best-of-N sampling [102, 209], where the model is sampled N times, and the best result is selected to improve model performance. Similarly, Wang et al. [235] introduced a promising sampling method called self-consistency, where n samples are drawn from the judge model, and the average score is output. These sampling methods enhance inference stability by selecting the best result from multiple evaluations. Further optimization strategies include the Tree of Thoughts (ToT) [265] method, which models the problem-solving process as a tree structure. This allows the model to explore multiple solution paths and optimize path selection through self-assessment mechanisms. The Graph of Thoughts (GoT) [15] method extends this concept by introducing directed graphs, where the non-linear interactions between nodes improve the efficiency and precision of multi-step reasoning. In both methods, LLM judges play a crucial role in guiding the model to select the most promising paths, thereby enhancing the quality and accuracy of reasoning.

Similarly, Lightman et al. [137] discuss how step-by-step validation can enhance the performance of LLMs in multi-step reasoning tasks, particularly in the domain of mathematics. SE-GBS [250] integrates self-assessment into the multi-step reasoning decoding process, generating scores that reflect logical correctness and further ensuring the accuracy and consistency of the reasoning chain. The REPS [105] improves the accuracy and reliability of reasoning validation models by comparing reasoning paths pairwise, verifying their logical consistency and factual basis. Also, Musolesi et

al. [160] proposed Creative Beam Search, with the LLM acting as a judge to simulate the human creative selection process, thereby enhancing the diversity and creativity of the generated results.

3.2.3 Feedback for Refinement. After receiving the initial response, LLM judges provide actionable feedback to iteratively improve output quality. By analyzing the response based on specific task criteria, such as accuracy, coherence, or creativity, the LLM can identify weaknesses in the output and offer suggestions for improvement. This iterative refinement process plays a crucial role in applications that require adaptability [32, 93, 155, 176, 261].

SELF-REFINE [155] enables LLMs to iteratively improve output quality through feedback generated by the model itself, without requiring additional training or supervision data. On the other hand, SELF-DEBUGGING [32] demonstrates a practical application of self-correction in code generation by identifying and rectifying errors through self-explanation and feedback. This approach has significantly enhanced the performance of LLMs across various code generation tasks.

In addition to refining response quality, LLMs judges are also widely used to enhance reasoning abilities. For example, REFINER [176] optimizes the reasoning performance of LLMs through interactions between a generator model and a critic model. In this framework, the generator model is responsible for producing intermediate reasoning steps, while the critic model analyzes these steps and provides detailed feedback, such as identifying calculation errors or logical inconsistencies. Xu et al. [261] propose a multi-agent collaboration strategy to enhance the reasoning abilities of LLMs by simulating the academic peer review process. The framework is divided into three stages: generation, review, and revision. Agents provide feedback and attach confidence scores to refine the initial answers, with the final result determined through majority voting.

While the feedback and correction mechanisms of LLMs judges are continually evolving, the limitations of self-feedback in improving quality should not be overlooked. Research on Self-Correct [93, 223] shows that, the intrinsic self-correction capabilities of LLMs often fall short of effectively improving reasoning quality. Valmeekam et al. [224] also raise concerns about the effectiveness of LLMs as self-validation tools in the absence of reliable external validators. Future research can focus on improving the accuracy of feedback provided by these LLM judges and incorporating external validation mechanisms to optimize their performance in complex reasoning tasks.

3.3 Data Construction

Data collection is a crucial stage in the development of machine learning systems, especially those driven by the rapid advancements in deep learning. The quality of the data directly determines the performance of the trained models. The LLMs-as-judges has significantly transformed the landscape of data collection, substantially reducing reliance on human effort. In this section, we will explore the pivotal role of LLMs-as-judges in data collection from two key perspectives: **Data Annotation** (§3.3.1) and **Data Synthesize** (§3.3.2).

3.3.1 Data Annotation. Data Annotation involves leveraging LLM judges to label large, unlabeled datasets efficiently [71, 79, 85, 217]. By utilizing the advanced natural language understanding and reasoning capabilities of LLMs, the annotation process can be automated to a significant extent, enabling the generation of high-quality labels with reduced human intervention.

LLMs have demonstrated remarkable potential in text annotation tasks, consistently outperforming traditional methods and human annotators in various settings. He et al. [85] evaluated the performance of GPT-4 in crowdsourced data annotation workflows, particularly in text annotation tasks. Their comparative study revealed that, even with best practices, the highest accuracy achievable by MTurk workers was 81.5%, whereas GPT-4 achieved an accuracy of 83.6%. Similarly, Gilardi et al. [71] analyzed 6,183 tweets and news articles, demonstrating that ChatGPT outperformed

crowdsourced workers in tasks such as stance detection, topic detection, and framing. Törnberg et al. [217] further investigated the classification of Twitter users' political leanings based on their tweet content. Their findings revealed that ChatGPT-4 not only surpassed human classifiers in accuracy and reliability but also exhibited bias levels that were comparable to or lower than those of human classifiers.

As technology advances, more and more research is exploring their application in multimodal data annotation. For example, the FullAnno [79] uses the GPT-4V model to generate image annotations, significantly improving the quality of image descriptions through a multi-stage annotation process. Furthermore, Latif et al. [117] explored the application of LLMs in speech emotion annotation, demonstrating that, with data augmentation, LLM-annotated samples can significantly enhance the performance of speech emotion recognition models. By integrating text, audio features, and gender information, the effectiveness of LLM-based annotations was further improved, highlighting their potential in advancing multimodal annotation tasks.

As LLMs perform excellently in annotation tasks, researchers are actively exploring methods to further improve annotation quality and address potential challenges. For example, AnnoLLM [83] introducedthe "explain-then-annotate" method, which enhances both the accuracy and transparency of annotations by prompting the LLM to justify its label assignments. Additionally, the LLMAAA [282] framework incorporates an active learning strategy to efficiently select high-information samples for annotation, thereby mitigating the effects of noisy labels and reducing the reliance on costly human annotation. These approach not only enhance the performance of task-specific models but also offer new perspectives on the efficient application of LLMs in annotation workflows.

3.3.2 Data Synthesize. The goal of Data Synthesis is to create entirely new data, either from scratch or based on seed data, while ensuring it is similar in distribution to real data. Data Synthesis enables the generation of diverse data samples, enhancing a model's generalization ability to unseen examples while reducing reliance on sensitive real-world data [2, 52, 108, 157, 235, 268].

In recent years, advancements in LLMs have led to significant improvements in both the quality and efficiency of data synthesis methods. In this domain, methods like SELFEE [268] and SynPO [52] have effectively enhanced the alignment capabilities of LLMs by leveraging small amounts of labeled data and iteratively generating preference-aligned data. Arif et al. [2] also introduce a multi-agent workflow for generating optimized preference datasets.

SELF-INSTRUCT [235] and Evol-Instruct [254, 278] represent innovative approaches to improving model alignment and performance through self-generated instruction data. SELF-INSTRUCT [235] requires minimal human annotation, instead relying on self-generated instruction data to align pre-trained models. Evol-Instruct [254, 278] further enhances LLM performance by automatically generating instruction data, significantly boosting model capabilities.

STaR [277] and ReSTEM [199] are research efforts aimed at enhancing reasoning capabilities through synthetic data. STaR [277] employs a self-guided iterative process to improve model performance on complex reasoning tasks, offering an effective solution for tackling increasingly sophisticated reasoning challenges in the future. ReSTEM [199], on the other hand, utilizes a self-training approach based on the expectation-maximization framework to enhance the problem-solving capabilities of large language models, particularly in areas such as solving mathematical problems and generating code.

4 METHODOLOGY

The use of LLM judges requires careful methodological considerations to ensure the accuracy and consistency of judgments. Researchers have developed various approaches according to the
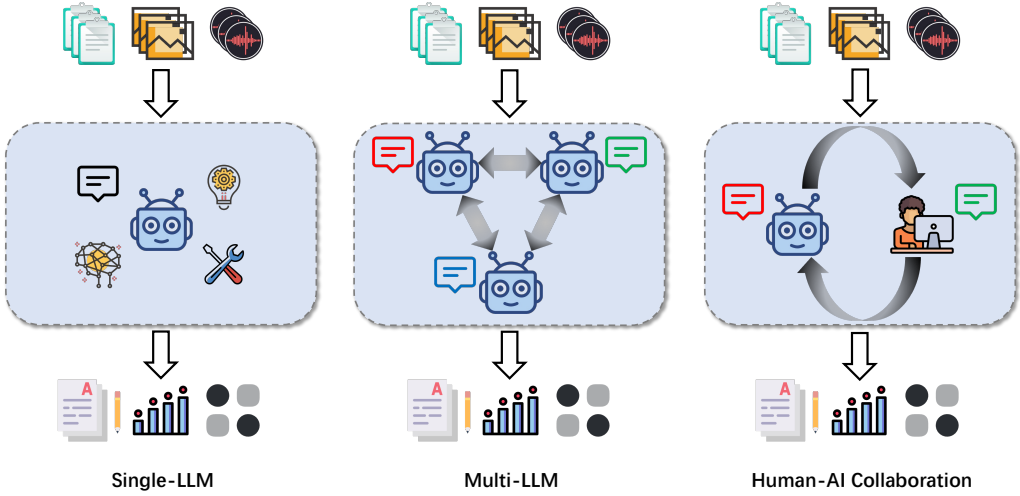
Fig. 5. Overview of the Methodology of LLMs-as-judges.

complexity and specific requirements of different judgment tasks, each offering unique advantages. In this section, we categorize these methodologies into three broad approaches: **Single-LLM System** (§4.1): evaluation by a single-LLM, **Multi-LLM System** (§4.2): evaluation by cooperation among multi-LLMs, and **Human-AI Collaboration** (§4.3): evaluation by cooperation of LLMs and Human. Figure 5 presents an overview of methodology.

4.1 Single-LLM System

Single-LLM System relies on a single model to perform judgment tasks, with its effectiveness largely determined by the LLM's capabilities and the strategies used to process input data. This approach can generally be divided into three fundamental components: **Prompt Engineering** (§4.1.1), **Tuning** (§4.1.2), and **Post-processing** (§4.1.3) of model outputs.

4.1.1 Prompt-based. Prompt engineering [189] involves crafting clear and structured input prompts tailored to elicit accurate and contextually appropriate responses from LLM judges. This approach is crucial for ensuring that LLMs grasp the complexities of specific tasks and provide relevant, consistent, and goal-aligned judgments. In many cases, well-designed prompts significantly reduce the need for extensive model training.

In-Context Learning. In-Context Learning (ICL) is a distinctive capability of LLMs that allows them to dynamically adapt to evaluation tasks using carefully curated examples or explanations within the prompt [53]. Several recent methods have demonstrated the power of ICL in LLM-as-judges, showcasing how it enhances the flexibility and performance of LLMs in diverse settings. For example, GPTScore [68] leverages the few-shot learning capability of generative pre-trained models to evaluate generated text. By using relevant examples to customize prompts, it provides a flexible, training-free approach to assess multiple aspects of text quality. Similarly, LLM-EVAL [141] incorporates carefully crafted examples into prompts, proposing a unified, multi-dimensional automatic evaluation method for open-domain dialogue. Another notable example is TALEC [280],

a model-based evaluation method that leverages in-context learning to enable users to set custom evaluation criteria for LLMs in specific domains. Through careful prompt engineering, users can iteratively adjust the examples to refine the evaluation process as needed. In addition, Jain et al. [94] proposed the In-Context Learning-based Evaluator (ICE) for multi-dimensional text evaluation. ICE leverages LLMs and a small number of in-context examples to evaluate generated text summaries, achieving competitive results.

While ICL can enable effective evaluation, it is not without challenges. One major issue is that the model's responses may be influenced by the selection of prompt examples, potentially leading to bias [62, 78, 290, 296]. To address this issue, Hasanbeig et al. proposed ALLURE [81], a comprehensive protocol designed to mitigate bias in ICL for LLMs during text evaluation. ALLURE [81] improves evaluator accuracy by iteratively incorporating discrepancies between its assessments and annotated data into the learning context. Moreover, after uncovering the existence of symbol bias within LLM evaluators when using ICL, Song et al. [204] proposed two effective mitigation strategy prompt templates, Many-Shot with Reference (MSwR) and Many-Shot without Reference (MSoR), to bolster the reliability and precision of LLM-based assessments.

Step-by-step. Step-by-step involves breaking down complex evaluation tasks into fine-grained components, leveraging the reasoning capabilities of LLMs to simplify the evaluation process. The most straightforward example of which is perhaps Chain-of-Thought (CoT) [113, 242]. Building on that, frameworks like G-EVAL [145] have been proposed to assess the quality of NLG outputs. G-EVAL [145] combines CoT with a form-filling paradigm, allowing the LLM to assess outputs in a structured manner. Similarly, ICE-Score [304] introduces a step-by-step framework for evaluating code, in which the LLM is instructed with task definitions, evaluation criteria, and detailed evaluation steps. By breaking the task down into clear steps, ICE-Score [304] improves the quality and consistency of code evaluation. Also, ProtocoLLM [271] employs a similar step-by-step approach to evaluate the specialized capabilities of LLMs in generating scientific protocols. Portia [134] achieves better evaluation results in a lightweight yet effective manner. It divides the answer into multiple parts, aligns similar content between candidate answers, and then merges them back into a single prompt for evaluation by the LLM.

Some studies break down evaluations into two steps: "explanation-rating." This approach suggests that providing an explanation enhances the reliability of the rating. Chiang et al. [36] offer empirical guidelines to improve the quality of LLM evaluations, demonstrating that combining rating with explanation (rate-explain) or explanation with rating (explain-rate) leads to higher correlations with human ratings. Another effective strategy is to decompose complex evaluation standards into specific, discrete criteria, allowing the LLM to assess each aspect independently. FineSurE [203] is an advanced example of this method, offering a framework for the fine-grained evaluation of text summarization quality. It breaks down the evaluation into multiple dimensions, such as faithfulness, completeness, and conciseness. Through detailed analysis, including fact-checking and key fact alignment, FineSurE [203] outperforms traditional methods in terms of evaluation accuracy.

Definition Augmentation. The Enhanced Definition approach involves refining prompts to inject improved evaluation criteria, establish assessment principles, or incorporate external knowledge into the LLM judge's decision-making process. Some studies focus on enriching and clarifying the prompts to ensure that the evaluation criteria are both comprehensive and well-defined.

For example, Liu et al. propose AUTOCALIBRATE [146], a multi-stage, gradient-free approach. This method involves the drafting, revision, and application of calibrated criteria, and it automatically calibrates and aligns an LLM-based evaluator to match human preferences for NLG quality assessment. Furthermore, SALC [76] enables LLMs to autonomously generate context-aware evaluation criteria for self-assessment, overcoming the limitations of static, human-defined metrics. On

the other hand, the LLM-as-a-Personalized-Judge approach [54] introduces a novel perspective by incorporating diverse evaluative roles and principles. This allows LLMs to adapt to complex, varied evaluation scenarios, resulting in more nuanced and context-sensitive assessments.

Another key aspect of Definition Augmentation is the retrieval of external knowledge, which helps reduce hallucinations and provides more factual support. For instance, BiasAlert [61], a tool designed to detect social bias in LLM-generated open-text outputs. It integrates external human knowledge with the LLM judge's inherent reasoning capabilities to reliably identify and mitigate bias, outperforming GPT4-as-A-Judge across various scenarios. Moreover, Chen et al. [33] found that within retrieval-augmented generation (RAG) frameworks, LLM judges do not exhibit a significant self-preference effect during evaluation.

Multi-turn Optimization. Multi-turn optimization involves iterative interactions between the evaluator and the evaluated entity, refining evaluation results through diverse forms of feedback, thus fostering deeper analysis and a progressive improvement in evaluation quality [295]. Unlike traditional methods that rely on predefined criteria, Xu et al. proposed ACTIVE-CRITIC [256], enabling LLMs to infer evaluation criteria from data and dynamically optimize prompts through multiple rounds of interaction. Moreover, Some studies [10, 153, 273, 286] leverage LLMs as question designers to engage in dynamic interactions with the evaluated entities, adjusting the questions and task design in real time. This allows for flexible modification of the evaluation content based on the performance of the evaluated entity, thereby enabling more comprehensive assessments.

4.1.2 Tuning-based. Tuning involves training a pre-existing LLM on a specialized dataset to adapt it to specific judgment tasks. It's especially useful when the judgment domain involves highly specialized knowledge or nuanced decision-making [92].

Score-based Tuning. Score-based tuning involves using data with scores to train models and enhance their ability to predict judgment scores based on specific evaluation criteria [28, 47, 229].

Many studies have explored the enhancement of LLM-as-judges by fine-tuning them on human-labeled datasets. For instance, PHUDGE [47], fine-tuned from the Phi-3 model, achieves state-of-the-art performance in terms of latency and throughput when automatically evaluating the quality of outputs from LLMs. This fine-tuning process equips the model with the necessary judgment skills, enabling it to assess various types of content in a structured and accurate manner. Additionally, ECT [229] introduces a novel method for transferring scoring capabilities from LLMs to lighter models. This allows the lighter models to function as effective reward models for sequence generation tasks, enhancing sequence generation models through reinforcement learning and reranking approaches. AttrScore [276] is another framework for evaluating attribution and identifying specific types of attribution errors, using a curated test set from a generative search engine and simulated examples from existing benchmarks. The above research highlights that LLMs can better align their decision-making process with humans through fine-tuning with human-constructed datasets.

In addition to human-labeled data, some studies have also attempted to fine-tune models using synthetic datasets like SorryBench [249] generated for evaluation tasks. These datasets are often created through rule-based methods or by generating artificial evaluation examples, which also give rise to some metrics like TIGERScore [99]. SELF-J [266] is a self-training framework for developing judge models to evaluate LLMs' adherence to human instructions without human-annotated quality scores. SELF-J [266] proposes selective instruction following, allowing systems to decline low-quality instructions. FENCE [252] is another factuality evaluator designed to provide claim-level feedback to language model generators. It details a data augmentation approach that enriches public datasets with textual critiques and diverse source documents from various tools, thereby enhancing factuality without introducing lesser-known facts. Utilizing synthetic training data to

fine-tune lightweight language model judges and employing prediction-powered inference (PPI) for statistical confidence to mitigate potential prediction errors, ARES [188] can automatically assess RAG systems.

Preference-based Learning. Preference-based learning focuses on training LLMs to make inferences and learn based on preferences, enabling the development of more adaptive and customizable evaluation capabilities.

Initially, researchers leverage these data in conjunction with advanced techniques like Direct Preference Optimization (DPO) [181] to train LLMs for more nuanced evaluative capabilities. In this method, the model is trained to predict which of two outputs is preferred according to human-like values, rather than learning a scalar reward signal. Such self-improving approach is well reflected in Meta-Rewarding [245]. Con-J [270] trains a generative judge by using the DPO loss on contrastive judgments and the SFT loss on positive judgments to align LLMs with human values. In terms of evaluating other LLMs effectively in open-ended scenarios, JudgeLM [301] addresses key biases in the fine-tuning process with a high-quality preference dataset. Another typical method is PandaLM [236], which is trained on a reliable human-annotated preference dataset, focusing extends beyond just the objective correctness of responses, and addresses vital subjective factors. Moreover, Self-Taught [231] is another approach to train LLMs as effective evaluators without relying on human-annotated preference judgments, using synthetic training data only. Through an iterative self-improvement scheme, LLM judges are able to produce reasoning traces and final judgments. Not quite the same, FedEval-LLM [84] fine-tunes many personalized LLMs without relying on labeled datasets to provide domain-specific evaluation, mitigating biases associated with single referees. It is designed to assess the performance of LLMs on downstream tasks, at the same time, ensuring privacy preservation.

As research has progressed, newer methods have emerged that combine both score-based and preference-based data to refine model evaluation capabilities, not to mention some novel metrics like INSTRUCTSCORE [258]. FLAMe [226] is an example of such an approach. It's a family of Foundational Large Autorater Models which significantly improves generalization to a wide variety of held-out tasks using both pointwise and pairwise methods during training. As generative judge model, AUTO-J [130] addresses challenges in generality, flexibility, and interpretability by training on a diverse dataset containing scoring and preference. To critique and refine the outputs of large language models, Shepherd [232] leverages a high-quality feedback dataset to identify errors and suggest improvements across various domains. In the domain of NLG, X-EVAL [142] consists of a vanilla instruction tuning stage and an enhanced instruction tuning stage that exploits connections between fine-grained evaluation aspects. Notably, Themis [88] also achieved outstanding results acting as a reference-free NLG evaluation language model designed for flexibility and interpretability. Similarly, CritiqueLLM [106] provides effective and explainable evaluations of LLM outputs, and uses a dialogue-based prompting method to generate high-quality referenced and reference-free evaluation data. Self-Rationalization [220] enhances LLM performance by iteratively fine-tuning the judge via DPO, which allows LLMs to learn from their own reasoning. Based on pointwise and pairwise dataset, CompassJudger-1 [20] acts as an open-source, versatile LLM for efficient and accurate evaluation of other LLMs. Likewise, Zhou et al. [294] introduces a systematic framework for bias reduction, employing calibration for closed-source models and contrastive training for open-source models. Apart from that, HALU-J [227] is designed to enhance hallucination detection in LLMs by selecting pertinent evidence and providing detailed critiques. PROMETHEUS [109] and PROMETHEUS 2 [110] are open-source LLMs specialized for fine-grained evaluation that can generalize to diverse, real-world scoring rubrics beyond a single-dimensional preference, supporting both direct assessment and pairwise ranking, and can evaluate based on custom criteria. What's more, the following PROMETHEUS-VISION [122] fills the gap in the visual field. As for

Table 1. An Overview of Fine-Tuning Methods in Single-LLM Evaluation (Sorted in ascending alphabetical order).

Method	Data Construction			Tuning Method		Base LLM
	Annotator	Domain	Scale	Evaluation Type	Technique	
ARES [188]	Human & LLM	RAG System	-	Pairwise	PPI	DeBERTa-v3-Large
AttrScore [276]	Human	Various	63.8K	Pointwise	SFT	Multiple LLMs
AUTO-J [130]	Human & GPT-4	Various	4396	Pointwise & Pairwise	SFT	Llama2-13B-Chat
CompassJudger-1 [20]	Human & LLM	Various	900K	Pointwise, Pairwise, & Generative	SFT	Qwen2.5 Series
Con-J [270]	Human & ChatGPT	Creation, Math, & Code	220K	Pairwise	SFT & DPO	Qwen2-7B-Instruct
CritiqueLLM [106]	Human & GPT-4	Various	7722	Pointwise & Pairwise	SFT	ChatGLM3-6B
ECT [229]	ChatGPT	Machine Translation, Text Style Transfer, & Summarization	-	Pointwise	SFT & RLHF	RoBERTa
FedEval-LLM [84]	Human	Instruct-tuning & Summary	5K, 10K, per client	Pairwise	LoRA	Llama-7B
FENCE [252]	Human & LLM	Summarization, QA, & Dialogue	-	Pointwise	SFT & DPO	Llama3-8B-Chat
FLAMe [226]	Human	Various	5.3M	Pointwise, Pairwise, Classification, & Open-ended generation	RLHF	PaLM-2-24B
HALU-J [227]	GPT-4-Turbo & GPT-3.5-Turbo	Multiple-Evidence Hallucination Detection	2663	Pointwise & Pairwise	SFT & DPO	Mistral-7B-Instruct
HelpSteer2 [237]	Human	Various	-	Pointwise & Pairwise	PPI & RLHF	Llama3.1-70B-Instruct
INSTRUCTSCORE [258]	GPT-4	Various	40K	Pointwise & Pairwise	SFT	Llama-7B
JudgeLM [301]	GPT-4	Open-ended Tasks	100K	Pairwise	SFT	Vicuna Series
LLaVA-Critic [253]	GPT-4o	Various	113K	Pointwise & Pairwise	DPO	LLaVA-OneVision(OV) 7B & 72B
Meta-Rewarding [245]	Llama3	Various	20K	Pairwise	DPO	Llama3-8B-Instruct
OffsetBias [174]	Human & LLM	Bias Detection	268K	Pairwise	RLHF	Llama3-8B-Instruct
PandaLM [236]	Human	Various	300K	Pairwise	SFT	Llama-7B
PHUDGE [47]	Human & GPT-4	NLG	-	Pointwise & Pairwise	LoRA	Phi-3
PROMETHEUS [109]	Human	Various	100K	Pointwise	SFT	Llama2-Chat-7B & 13B
PROMETHEUS2 [110]	Human	Various	300K	Pointwise & Paiwise	SFT	Mistral-7B & Mistral-8x7B
PROMETHEUS-VISION [122]	GPT-4V	Various	15K	Pointwise	SFT	Llava-1.5
SELF-J [266]	Human & GPT-4	Common, Coding, & Academic	5.7M	Pointwise	LoRA	Llama2-13B
Self-Rationalization [220]	LLM	Various	-	Pointwise & Pairwise	SFT & DPO	Llama3.1-8B-Instruct
Self-Taught [231]	LLM	Various	20K	Pairwise	-	Llama3-70B-Instruct
Shepherd [232]	Human	Various	-	Pointwise & Pairwise	SFT	Llama-7B
SorryBench [249]	Human & GPT-4	Unsafe Topics	2.7K	Pointwise	SFT	Multiple LLMs
Themis [88]	Human & GPT-4	NLG	67K	Pointwise & Pairwise	SFT & DPO	Llama3-8B
TIGERScore [99]	Human & GPT-4	Text Generation	42K	Pointwise	SFT	Llama2-7B & 13B
X-EVAL [142]	Human	NLG	55,602	Pointwise & Pairwise	SFT	Flan-T5

various multimodal tasks, LLaVA-Critic [253] demonstrates its effectiveness in providing reliable evaluation scores and generating reward signals for preference learning, highlighting the potential of open-source LMMs in self-critique and evaluation.

4.1.3 Post-processing. Post-processing involves further refining evaluation results to extract more precise and reliable outcomes. This step typically includes analyzing the initial outputs to identify patterns, inconsistencies, or areas requiring improvement, followed by targeted adjustments and in-depth analysis. By addressing these issues, post-processing ensures that the evaluation results are not only accurate but also aligned with the specific objectives and standards of the task.

Probability Calibration. During the post-hoc process of the model output, some studies use rigorous mathematical derivations to quantify the differences, thereby optimizing them. For instance, Daynauth et al. [45] investigates the discrepancy between human preferences and automated evaluations in language model assessments, particularly employs Bayesian statistics and a t-test to quantify bias towards higher token counts, and develops a recalibration procedure to adjust the GPTScorers. Apart from that, ProbDiff [247] is another novel self-evaluation method for LLMs that assesses model efficacy by computing the probability discrepancy between initial responses and their revised versions. Moreover, Liusie et al. [150] introduces a Product of Experts (PoE) framework for efficient comparative assessment using LLMs, which yield an expression that can be maximized with respect to the underlying set of candidates. This paper proposes two experts, a soft Bradley-Terry expert and a Gaussian expert that has closed-form solutions. Unlike from frameworks above, CRISPR [264] is a novel bias mitigation method for LLMs executing instruction-based tasks,

which identifies and prunes bias neurons with probability calibration, reducing bad performance without compromising pre-existing knowledge.

Text Reprocessing. In LLMs-as-judges, text reprocessing methods are essential for enhancing the accuracy and reliability of evaluation outcomes. Specifically, text processing can improve the evaluation process by integrating multiple evaluation results or outcomes from several rounds of assessment. For example, Sottana et al. [206] employs a multi-round evaluation process. Each round involves scoring model outputs based on specific criteria, with the human and GPT-4 evaluations ranking model performances from best to worst and averaging these rankings to mitigate subjectivity. For the single-response evaluation, AUTO-J [130] employs a "divide-and-conquer" strategy. Critiques that either adhere to or deviate from the scenario-specific criteria are consolidated to form a comprehensive evaluation judgment and then generate the final assessment. Consistent with former aforementioned studies, Yan et al. [262] introduces a post-processing method to consolidate the relevance labels generated by LLMs. It demonstrates that this approach effectively combines both the ranking and labeling abilities of LLMs through post-processing. Furthermore, REVISEVAL [281] is a novel evaluation paradigm that enhances the reliability of LLM Judges by generating response-adapted references through text revision capabilities of LLMs. Apart from that, Tessler et al. [214] explores the use of AI as a mediator in democratic deliberation, aiming to help diverse groups find common ground on complex social and political issues. With the goal of maximizing group approval, the researchers developed the "Habermas Machine", which iteratively generate group statements based on individual opinions.

Another category of text reprocessing methods involves task transformation, primarily focusing on the conversion between open-ended and multiple-choice question (MCQ) formats. Ren et al. [186] explores the use of self-evaluation to enhance the selective generation capabilities of LLMs. Specifically, the authors reformulate open-ended generation tasks into token-level prediction tasks, reduce sequence-level scores to token-level scores to improve quality calibration. Conversely, Myrzakhan et al. [161] introduces the Open-LLM-Leaderboard, a new benchmark for evaluating LLMs using open-style questions, which eliminates selection bias and random guessing issues associated with multiple-choice questions. It presents a method to identify suitable open-style questions and validate the correctness of LLM open-style responses against human-annotated ground-truths.

4.2 Multi-LLM System

Multi-LLM Evaluation harnesses the collective intelligence of multiple LLMs to bolster the robustness and reliability of evaluations. By either facilitating inter-model communication or independently aggregating their outputs, these systems can effectively mitigate biases, leverage complementary strengths across different models, refine decision-making precision, and foster a more nuanced understanding of complex judgments.

4.2.1 Communication. Communication means the dynamic flow of information between LLMs, which is pivotal for sparking insights and sharing rationales during the judgment process. Recent research has shown that communication among LLMs can enable emergent abilities through their interactions [261], leading to a cohesive decision-making process and better judgment performance. The Multi-LLM system can benefit from LLM interactions in two ways: cooperation and competition.

Cooperation. Multi-LLMs can work together to achieve a common goal with information and rationales sharing through interactions to enhance the overall evaluation process. For example, Zhang et al. [285] proposed an architecture named WideDeep to aggregate information at the LLM's neuro-level. In addition, Xu et al. [261] introduced a multi-agent collaboration strategy that mimics the academic peer review process to enhance complex reasoning in LLMs. The approach involves

agents creating solutions, reviewing each other's work, and revising their initial submissions based on feedback. Similarly, ABSEval [136] utilizes four agents for answer synthesize, critique, execution, and commonsense, to build the overall workflow. Although the cooperation can complement each other's strengths between LLMs to a certain degree, this method still includes the risk of groupthink, where similar models reinforce each other's biases rather than providing diverse insights.

Competition. Multi-LLMs systems can also benefit from competitive or adversarial communication, i.e., LLMs argue or debate to evaluate each other's outputs [22, 132, 158, 286]. Such multi-LLMs systems could be categorized into centralized and decentralized structures [168].

In the centralized structure, a single central LLM acts as the orchestrator of the conversation, highlighting the efficiency of a unified decision-making process. Auto-Arena [286] is such a novel framework that automates the evaluation of LLMs through agent peer battles and committee discussions, aiming to provide timely and reliable assessments. In detail, the framework conducts multi-round debates between LLM candidates, and uses an LLM judge committee to decide the winner. Inspired by courtroom dynamics, Bandi and Harrasse [12] propose two architectures, MORE and SAMRE, which utilize multiple advocates and iterative debates to dynamically assess LLM outputs.

In contrast, the decentralized structure emphasizes a collective intelligence where all models engage in direct communication, promoting a resilient and distributed decision-making structure. In the domain of LLM debates, Moniri et al. [158] introduced a unique automated benchmarking framework, employing another LLM as the judge to assess not only the models' domain knowledge but also their abilities in problem definition and inconsistency recognition. ChatEval [22] is another multi-agent debate framework that utilizes multiple LLMs with diverse role prompts and communication strategies on open-ended questions and traditional NLG tasks, significantly improves evaluation performance compared to single-agent methods. Moreover, PRD [132] applied peer rank and discussion to address issues like self-enhancement and positional bias in current LLM evaluation methods, leading to better alignment with human judgments and a path for fair model capability ranking.

4.2.2 Aggregation. Alternatively, in multi-LLM systems without communication, judgments are independently generated by multiple models, which are subsequently synthesized into a final decision through various aggregation strategies. Techniques such as majority vote, weighted averages, and prioritizing the highest confidence predictions, each play a crucial role. These methods allow each model to assess without interference, and eventually extract and combine the most effective elements from each model's response.

Simple voting methods, such as majority voting, by selecting the most frequent answers, offers a straightforward approach to synthesize evaluations. For example, Badshah et al. [8] introduced a reference-guided verdict method for evaluating free-form text using multiple LLMs as judges. Combining these LLMs through majority vote significantly improves the reliability and accuracy of evaluations, particularly for complex tasks. Furthermore, PoLL [225] demonstrates that using a diverse panel of smaller models as judges through max voting and average pooling is not only an effective method for evaluating LLM performance, but also reduces intra-model bias of a single large model. Language-Model-as-an-Examiner [10] is another benchmarking framework to evaluate the performance of foundation models on open-ended question answering through voting. In the peer-examination mechanism, the LM serves as a knowledgeable examiner that formulates questions and evaluates responses in a reference-free manner. What's more, multi-LLM evaluation could also be used in improving dataset quality. Choi et al. [38] provided an enhanced dataset, MULTI-NEWS+, which is the result of a cleansing strategy leveraging CoT and majority voting to identify and exclude irrelevant documents through LLM-based data annotation.

Weighted scoring aggregation involves assigning different importance to different model outputs, either by aggregating multiple overall scores for the same response or by combining assessments of different aspects of the response to form a comprehensive evaluation. On the one hand, through a peer-review mechanism, PiCO [165] allows LLMs to answer unlabeled questions and evaluate each other without human annotations. It formalizes the evaluation as a constrained optimization problem, maximizing the consistency between LLMs' capabilities and corresponding weights. Likewise, PRE [27, 39] can automatically evaluate LLMs through a peer-review process. It selects qualified LLMs as reviewers through a qualification exam and aggregates their ratings using weights which is proportional to their agreement of humans, demonstrating effectiveness and robustness in evaluating text summarization tasks. In the field of recommendation explanations, Zhang et al. [284] suggests that ensembles like averaging ratings of multiple LLMs can enhance evaluation accuracy and stability. On the other hand, for example, AIME [175] is an evaluation protocol that utilizes multiple LLMs that each with a specific role independently generate an evaluation on separate criteria and then combine them via concatenation. Similarly, a paper introduces HD-EVAL [147], which iteratively aligns LLM-based evaluators with human preference via Hierarchical Criteria Decomposition. By decomposing a given evaluation task into finer-grained criteria, aggregating them according to estimated human preferences, pruning insignificant criteria with attribution, and further decomposing significant criteria, HD-EVAL demonstrates its superiority.

Apart from weighting methods, there are some other advance mathematical aggregation techniques, such as Bayesian methods and graph-based approaches, offering more robust ways to handle uncertainties and inconsistencies across multiple evaluators. Notably, a paper introduces two calibration methods, Bayesian Win Rate Sampling (BWRS) and Bayesian Dawid-Skene [70], to address the win rate estimation bias when using many LLMs as evaluators for text generation quality. In addition to that, GED [90] addresses inconsistencies in LLM preference evaluations by leveraging multiple weak evaluators to construct preference graphs, and then utilize DAG structure to ensemble and denoise these graphs for better, non-contradictory evaluation results.

LLM-based aggregation is a grand-new perspective like Fusion-Eval [198]. It's a novel framework that integrates various assistant evaluators using LLMs, each of which specializes in assessing distinct aspects of responses, to enhance the correlation of evaluation scores with human judgments for natural language systems.

In addition to the above direct use of multiple model evaluation, the cascade framework employs a tiered approach, where weaker models are used initially for evaluations, and stronger models are engaged only when higher confidence is required, optimizing resource use and enhancing evaluation precision. Jung et al. [103] proposes "Cascaded Selective Evaluation" to ensure high agreement with human judgments while using cheaper models. Similar to the work above, Huang et al. [92] proposes CascadedEval, a novel method integrating proprietary models, in order to compensate for the limitations of fine-tuned judge models.

4.3 Human-AI Collaboration System

Human-AI Collaboration Systems bridge the gap between automated LLM judgments and the essential need for human oversight, particularly in high-stakes domains such as law, healthcare, and education. Human evaluators act either as the ultimate deciders, or as intermediaries who verify and refine model outputs. By incorporating human insights, Hybrid systems can ensure the final judgment is more reliable and aligned with ethical considerations, and empower continuous model improvement through feedback loops.

In many Human-AI Collaboration systems, human evaluators play a vital role during the evaluation process itself, actively collaborating with the LLMs to review and refine the generated outputs. For example, COEVAL[131] introduces a collaborative evaluation pipeline where LLMs generate
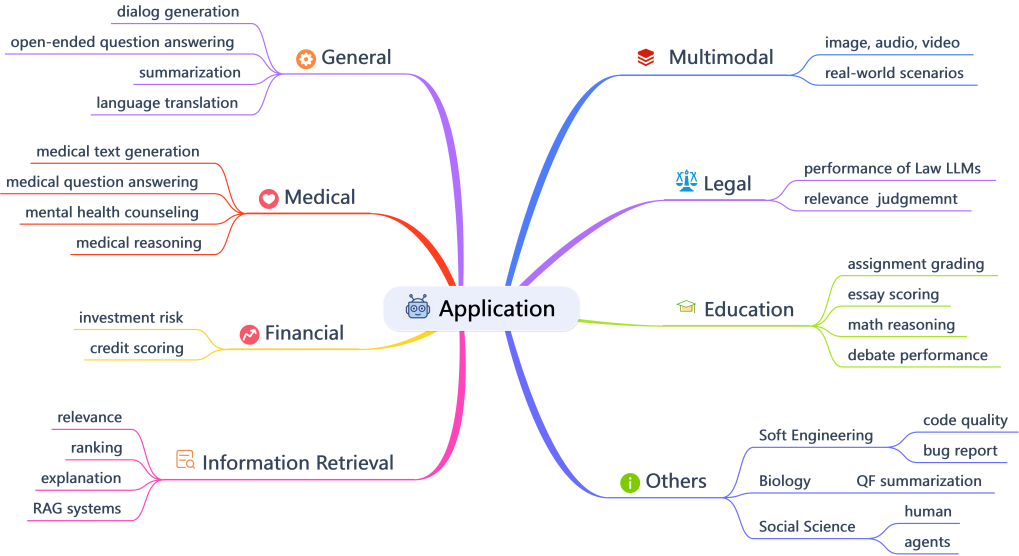
Fig. 6. LLMs-as-judges are widely applied across various domains.

initial criteria and evaluations for open-ended natural language tasks. These machine-generated outputs are then reviewed and refined by human evaluators to guarantee reliability. To address a significant positional bias in LLMs when used as evaluators, Wang et al. [230] proposes a calibration framework with three strategies: Multiple Evidence Calibration, Balanced Position Calibration, and Human-in-the-Loop Calibration. Similarly, EvalGen[192] integrates human feedback iteratively to refine evaluation criteria, addressing challenges such as "criteria drift", where the standards of evaluation evolve as humans interact with the model. These systems allow human evaluators to provide real-time adjustments, enhancing the accuracy and trustworthiness of the evaluation process.

While in other systems, human involvement takes place after the LLM has completed its evaluations, providing a final layer of verification and adjustment. This method ensures that the LLM's judgments are thoroughly scrutinized and aligned with human values. EvaluLLM[170] allows humans to intervene and refine the evaluation results, thereby enhancing trust in the model's performance while also controlling for potential biases. Additionally, Chiang et al.[35] tried LLM TAs as an assignment evaluator in a large university course. After students submit assignments and receive LLM-generated feedback, the teaching team reviews and finalizes the evaluation results. This process illustrates how human oversight after the initial automated evaluation can guarantee fairness and consistenty with academic standards.

5 APPLICATION

Due to the convenience and effectiveness of LLM Judges, they have been widely applied as judges across various domains. These applications not only cover general domains but also specific domains such as multimodal, medical, legal, financial, education, information retrieval and others. In this section, we will provide a detailed introduction to these applications, demonstrating how LLMs achieve precise and efficient evaluations in different domains.

5.1 General

In general domains, LLM Judges are applied to tasks requiring both understanding and generation, such as dialogue generation, open-ended question answering, summarization, and language translation. Each task follows its own set of evaluation criteria to meet its specific requirements. For instance, in dialogue generation [133], the criteria emphasize the natural flow, emotional resonance, and contextual relevance of the conversation. In summarization tasks [162], the evaluation focuses on the coherence, consistency, fluency, and relevance of the text. In translation tasks [64], the assessment prioritizes the quality, accuracy, fluency, and style.

As these diverse sub-tasks require specialized evaluation criteria, LLM judges provides refined evaluation methods that go beyond traditional metrics, paving the way for more comprehensive and in-depth assessments. For instance, Shu et al. [198] introduced Fusion-Eval, an innovative approach that leverages LLMs to integrate insights from various assistant evaluators. Fusion-Eval evaluated summary quality across four dimensions—coherence, consistency, fluency, and relevance, achieving a system-level Kendall-Tau correlation of 0.962 with human judgments. For dialogue quality, it assessed six aspects: coherence, engagingness, naturalness, groundedness, understandability, and overall quality, attaining a turn-level Spearman correlation of 0.744. Furthermore, Xu et al. [256] proposed the ACTIVE-CRITIC framework, which enables LLMs to actively infer the target task and relevant evaluation criteria while dynamically optimizing prompts. In the story generation task, this framework achieved superior evaluation performance.

5.2 Multimodal

In the multimodal domain, the evaluation objects of LLMs are not limited to textual data but extend to various forms of information such as images, audio, and video. One of the primary challenges in evaluating multimodal tasks lies in the significant heterogeneity among these modalities, including substantial differences in data structures, representation methods, and feature distributions.

To address this challenge, advanced techniques are often required to help LLMs integrate different forms of information, ensuring that they can provide accurate and meaningful evaluations. For example, Xiong et al. [253] trained an open-source multimodal LLM, LLaVA-Critic, specifically to evaluate model performance in multimodal scenarios. Similarly, Chen et al. [24] developed a Multimodal LLM-as-a-judge for 14 Vision-Language tasks, providing a unified evaluation framework. In addition, LLMs-as-judges can also be used in audio. For instance, Latif et al. [117] used LLMs for identifying and evaluating emotional cues in speech, achieving remarkable accuracy in the process. Beyond these efforts, some recent studies [46, 299] have also explored the potential of multimodal LLMs to self-evaluate and self-reward, enhancing their performance without the need for external evaluators or human annotations.

As the application of LLMs-as-judges continues to expand in multimodal domains, there is a growing interest in exploring their use in more specific real-world scenarios, such as autonomous driving. Chen et al. [29] proposed CODA-LM, a novel vision-language benchmark for self-driving, which provides automatic and systematic evaluation of Large Vision-Language Models (LVLMs) on road corner cases. Interestingly, they found that using the text-only LLM judges resulted in a closer alignment with human preferences than LVLMs.

5.3 Medical

In the medical field, LLMs-as-judges have demonstrated significant potential, particularly in areas such as diagnostic support, medical text analysis, clinical decision-making, and patient education. In this domain, high-quality evaluation requires LLM judges to possess precise interpretation capabilities for domain-specific terminology, the ability to comprehensively analyze diverse data

types (such as clinical records and medical imaging), and strict compliance with high accuracy standards and ethical guidelines.

In the realm of **medical text generation**, Xie et al. [251] used LLMs to evaluate the comp-duikeyi1leteness, conciseness, and attribution of medical texts at a fine-grained level. Similarly, Brake et al. [17] leveraged LLMs, such as Llama2, to assess clinical note consistency, with results indicating agreement levels comparable to human annotators. When it comes to **medical question answering**, Krolik et al. [114] explored the use of LLMs to automatically evaluate answer quality. Their focus was on evaluating adherence to medical knowledge and professional standards, completeness of information, accuracy of terminology, clarity of expression, and relevance to the question.

In the area of **mental health counseling**, Li et al. [126] utilized LLMs to automate the evaluation of counseling effectiveness and quality. Key assessments included whether the counseling identified the client's emotional needs, provided appropriate responses, demonstrated empathy, managed negative emotions, and met the overall goals of mental health support. Beyond these above applications, LLMs' judging capabilities have also been applied to assist in improving performance in specialized **medical reasoning tasks**. For instance, many studies [95] employed LLMs to evaluate and filter medical information, thereby supporting enhanced medical reasoning.

5.4 Legal

Due to the powerful evaluation capabilities of LLMs, LLMs-as-judges have been widely applied in the legal domain, covering multiple key scenarios, including evaluating the performance of law LLMs and relevance judgment in legal case retrieval. In the legal domain, the application of LLMs requires a deep understanding of the legal framework of specific jurisdictions, complex legal language, and rigorous logical reasoning abilities [128]. At the same time, interpretability and transparency of the evaluation results are essential core requirements, as legal practice highly depends on clear logic and verifiable conclusions. Furthermore, the bias and fairness of the model are of significant concern, as any bias in legal evaluations could have a profound impact on judicial fairness. These unique demands set higher standards for LLM judges.

In response to these challenges, recent research has explored various ways in which LLMs can be effectively employed in legal evaluations. Some research used LLMs as judges to assist in **evaluating the performance of Law LLMs**. For example, Yue et al. [275] introduced DISC-LawLLM to provide a wide range of legal services and utilized GPT-3.5 as a referee to evaluate the model's performance. They assessed three key criteria—accuracy, completeness, and clarity—by assigning a rating score from 1 to 5. Similarly, Ryu et al. [187] applied retrieval-based evaluation to assess the performance of LLMs in Korean legal question-answering tasks, which applied RAG not for generation but for evaluation. What's more, LLMs have also been utilized to construct evaluation sets. Raju et al. [185] explored methods for constructing these domain-specific evaluation sets, which are essential for enabling LLMs-as-judges to perform effective evaluation in legal domain. Beyond performance evaluation, LLMs have also been utilized for **relevance judgment** in legal case retrieval. For instance, Ma et al. [154] used LLMs to automate the evaluation of large numbers of retrieved legal documents, improving both the scalability and accuracy of legal case retrieval systems. In conclusion, the application of LLMs-as-judges in law holds significant promise in future.

5.5 Financial

In the financial domain, LLM judges have been extensively explored in scenarios such as investment risk assessment and credit scoring, which presenting unique challenges. For example, the complexity of risk assessment requires LLMs to accurately capture the influence of multifaceted factors, including market volatility, regulatory changes, and geopolitical events. Real-time processing

demands further elevate the challenge, requiring LLMs not only to be computationally efficient but also to deliver rapid response times. Additionally, the dynamic nature of high-frequency trading demanding that LLMs swiftly adapt to fluctuating market conditions.

In **investment risk assessment**, LLMs have proven effective due to their ability to process large amounts of data and make informed judgments. For instance, Xie et al. [248] developed a financial LLM, FinMA, fine-tuned on LLaMA to evaluate investment risks more effectively. Their model is designed to follow instructions for risk assessment and decision analysis, improving the accuracy and efficiency of financial evaluations.

Another key application in the financial domain is **credit scoring**, which predicts the future repayment ability and default risk of individuals or businesses. By analyzing a vast array of data, including credit history, financial status, and other relevant factors, LLMs can help financial institutions make more accurate credit scoring assessments. For example, Babaei et al. [7] demonstrated how LLMs can process unstructured text data, such as customer histories, contract terms, and news reports, to enhance the precision of credit assessments.

Furthermore, as the use of LLMs in finance continues to grow, there is a rising need to **evaluate the performance of these financial LLMs**. To address this, Son et al. [201] developed an automated financial evaluation benchmark that leverages LLMs to extract valuable insights from both unstructured and structured data. This framework helps optimize the construction, updating, and compliance checks of financial benchmarks, supporting more efficient and scalable evaluation processes. Based on this, they facilitated the continuous optimization of financial LLMs, driving further advancements in the financial domain.

5.6 Education

LLMs-as-judges have found extensive applications in the education domain, covering a wide range of tasks, such as grading student assignments, evaluating essays, assessing mathematical reasoning, and judging debate performance. These applications present several key challenges, including the diversity of student responses and individual differences, as well as the need for multidimensional evaluation. Effective evaluation in education requires LLMs to consider not only correctness but also creativity, clarity, and logical coherence. Additionally, the interpretability and fairness of the evaluation results are crucial, as educational assessments significantly impact students' development and future opportunities.

In **assignment grading**, Chiang et al. [35] introduced the concept of an LLM Teaching Assistant (LLM TA) in university classrooms, utilizing GPT-4 to automate the grading of student assignments. By employing prompt engineering to define scoring criteria and task descriptions, LLM TA generates quantitative scores and detailed feedback. Their study emphasized the system's ability to maintain consistency, adhere to grading standards, and resist adversarial prompts, highlighting its robustness and practicality for classroom use.

In addition to assignment grading, LLMs-as-judges are also being explored for **automated essay scoring.** Wang et al. [228] proposed an advanced intelligent essay scoring system, integrating LLMs such as BERT and ChatGPT to enable automated scoring and feedback generation for essays across various genres. Similarly, Song et al. [205] investigated a framework and methodology for automated essay scoring and revision based on open-source LLMs. Furthermore, Zhou et al. [297] explored the potential of LLMs in academic paper reviewing tasks, assessing their reliability, effectiveness, and possible biases as reviewer. They found that while LLMs show certain promise in the domain of automated reviewing, they are not yet sufficient to fully replace human reviewers, particularly in areas with high technical complexity or strong innovation.

Another area where LLMs-as-judges are making an impact is in the evaluation of **math reasoning**. Unlike traditional mathematical task evaluation, which focuses solely on the correctness of the

final results, Xia et al. [246] argued that additional aspects of the reasoning process should also be assessed, such as logical errors or unnecessary steps. In their work, the authors proposed ReasonEval, a new methodology for evaluating the quality of reasoning steps based on LLMs-as-judges.

LLMs have also been employed in **judging debate performance**. Liang et al. [135] proposed Debatrix, a new method which leverages LLMs to evaluate and analyze debates. The main aspects assessed include the logical consistency of arguments, the effectiveness of rebuttals, the appropriateness of emotional expression, and the coherence of the debate.

5.7 Information Retrieval

Information retrieval refers to the process of effectively retrieving, filtering, and ranking relevant information from a large collection of data, matching information resources to users' needs (queries). However, evaluating these systems presents several challenges, particularly due to the complexity of real-world data, the diversity of user needs, and personalization. To solve these challenges, LLMs-as-judges have been used across various applications, including relevance judgment, text ranking, recommendation explanations evaluation, and assessing retrieval-augmented generation (RAG) systems.

One key area in information retrieval is the evaluation of **the relevance of retrieved results to user queries**, a task that traditionally relies on manual annotations [183, 200]. Rahmani et al. [183] proposed a framework called LLMJudge which leveraged LLMs to assess the relevance of information retrieval system results to user queries, providing a more scalable and efficient evaluation approach.

Another important aspect of information retrieval is the **ranking of search results or recommendation lists**. Traditional ranking models often rely on shallow features or direct matching scores, which may not yield optimal results. To address this, Qin et al. [180] examined the performance of LLMs in text ranking tasks and proposed a novel method based on pairwise ranking prompting, utilizing LLMs for text ranking. Additionally, Niu et al. [166] introduced a framework called JudgeRank, which leveraged LLMs to rerank results in reasoning-intensive tasks. By evaluating the logic, relevance, and quality of candidate results, this approach tried to enhance ranking performance.

In recommendation systems, **explanation evaluation** plays a crucial role in helping users understand why a specific product, movie, or piece of content is recommended. Zhang et al. [284] investigated the potential of LLMs as automated evaluators of recommendation explanations, assessing them across multiple dimensions such as quality, clarity, and relevance. This approach provides a more efficient way to evaluate the effectiveness of explanations, which is essential for improving user trust and satisfaction.

Furthermore, with the growing use of **retrieval-augmented generation (RAG) systems** in tasks like question answering, fact-checking, and customer support, there is an increasing need to evaluate the quality of these systems. Traditional evaluation methods rely on large manually annotated datasets, which are time-consuming and costly. To address this, Saad et al. [188] proposed a new automated evaluation framework called ARES, which leveraged LLMs as the core evaluation tool to directly assess retrieval and generated content across multiple dimensions, including relevance, accuracy, coverage, fluency, and coherence.

5.8 Others

5.8.1 Soft Engineering. The challenges that LLMs-as-judges need to overcome in the software engineering domain include complex code structures and the diversity of evaluation criteria. A numerous of articles [175, 243] used LLMs-as-judges to assess the quality of code generation. Moreover, Kumar et al. [115] employed LLMs to evaluate the quality of Bug Report Summarization.

5.8.2 Biology. The main evaluation challenges in biological field include the complexity and diversity of the data and the need for specialized biological knowledge [86, 167]. For example, Hijazi et al. [86] used LLMs to evaluate Query-focused summarization (QFS), which refers to generating concise and accurate summaries from a large set of biomedical literature based on a specified query. In this context, the LLMs are used to assess whether these summaries accurately answer the specified query and whether they cover the correct biological knowledge.

5.8.3 Social Science. LLMs-as-Judges have also found applications in social sciences. On one hand, they are used in **real-world human social contexts**. For example, Tessler et al. [214] used LLMs to participate in democratic discussions, assessing the quality of arguments, identifying fallacies, or providing a balanced view of an issue, thus helping people reach consensus on complex social and political matters. On the other hand, LLMs-as-judges are also used in **social scenarios constructed by language agents**. Zhou et al. [298] proposed an interactive evaluation framework called Sotopia, which used LLMs to assess the social intelligence of language agents from multiple dimensions, such as emotional understanding, response adaptability, and other social skills.

In this section, we have outlined the specific applications of LLMs-as-judges across various domains. In these applications, LLMs leverage their powerful text understanding and generation capabilities to perform effective evaluations and judgments, providing accurate feedback and improvement suggestions. Although LLMs-as-judges have shown tremendous potential in these areas, especially in handling large-scale data and automating assessments, they still face challenges such as **the depth of domain-specific knowledge, limitations in reasoning abilities, and the diversity of evaluation criteria**. In the future, with continuous improvements in model performance and domain adaptation capabilities„ we believe the application of LLMs-as-judges will become more widespread and precise across various domains.

6 META-EVALUATION

Meta-evaluation, the process of assessing the quality of the evaluator itself, is a crucial step in determining the reliability, consistency, and validity of LLM judges. Given the diverse applications of LLMs as evaluators, meta-evaluation methods have also been evolving. Researchers have proposed various datasets and metrics tailored to different tasks and evaluation objectives to assess the reliability and validity of LLM-based evaluations. This chapter will explore state-of-the-art Benchmarks (§6.1) and evaluation Metrics (§6.2), categorize existing approaches, and discuss their advantages and limitations.

6.1 Benchmarks

To evaluate LLM-based judges, a common approach is to measure their alignment with human preferences, as human judgments are often considered the gold standard for quality and reliability. Given the diverse range of applications for LLM-based judges, different benchmarks have been created, each tailored to specific evaluation criteria and use cases. In this section, we present a comprehensive collection of 40 widely-used benchmarks, each designed to capture different aspects of evaluation, such as language understanding, factual accuracy, coherence, creativity, and fairness. To enhance clarity and facilitate comparison, we categorize these benchmarks by application domain.

6.1.1 Code Generation. Code generation aims to produce executable program code from natural language input. This task typically involves translating user requirements or descriptions into precise code. The applications of code generation are vast, including automated script creation, bug fixing, and the generation of complex programming tasks.Evaluating code generation is highly challenging, and LLMs are increasingly being used as evaluators for assessing code quality.

Table 2. Statistical information of different benchmarks (Part 1).

Benchmark	Task	Type	Num	Evaluation Criteria	Language
HumanEval [30]	Code	Pointwise	164	Functional correctness	English
SWE-bench [101]	Code	Pointwise	2,294	Task solve	English
DevAI [303]	Code	Pointwise	365	Disagreement, Task solve, Requirements met	English
CrossCodeEval [50]	Code	Pointwise	1,002	Code match, Identifier match	English
CodeUltraFeedback [243]	Code	Pointwise Pairwise	10k	Code explanation, Code complexity and efficiency, Code readability, Coding style	English
Literary Translation Comparisons [104]	Translation	Pairwise	720	Translation quality and errors	Multilingual
MQM [65]	Translation	Pointwise	100k	Accuracy, Fluency, Terminology, Style, Locale	Multilingual
WMT Metrics Shared Task [66]	Translation	Pointwise	-	Adequacy, Fluency	Multilingual
SummEval [59]	Summary	Pointwise	1,600	Coherence, Consistency, Fluency, Relevance	English
Opinsummeval [194]	Summary	Pointwise	1,400	Aspect relevance, Self-coherence, Readability Sentiment consistency,	English
Frank [169]	Summary	Pointwise	2,250	Factual errors	English
Topical-Chat [72]	Dialogue	Pointwise	60	Understandable, Natural, Maintains context, Interesting, Uses knowledge, Overall quality	English
Personal-Chat [283]	Dialogue	Pointwise	60	Understandable, Natural, Maintains context, Interesting, Uses knowledge, Overall quality	English
DSTC10 Hidden Set [279]	Dialogue	Pointwise	9,500	Coherence, Appropriateness, Naturalness, Toxicity control	English
HANNA [34]	Story	Pointwise	1,056	Relevance, Coherence, Empathy, Surprise, Engagement, Complexity	English
MANS [73]	Story	Pointwise	2,000	Coherence	English
StoryER [26]	Story	Pairwise	100k	Upvoted	English
Per-DOC [229]	Story	Pointwise	7,000	Interestingness, Adaptability, Character developmentSurprise, Ending	English
PKU-SafeRLHF [96]	Value	Pairwise	83.4K	Helpfulness, Harmlessness	English
HHH [6]	Value	Pairwise	221	Helpfulness, Honesty, Harmlessness	English
Cvalue [255]	Value	Pairwise	145k	Safety, Responsibility	Chinese
Yelp [4]	Recom	Pointwise	8,630k	User perference	English
Movielens_Explanation [284]	Recom	Pointwise	2,496	Persuasiveness, Transparency, Accuracy, Satisfaction	English
Trec DL21&22 [42, 43]	Search	Pointwise	1,549/ 2,673	Relevacne	English
LeCarDv2 [129]	Search	Pointwise	55,192	Characterization relevance, Penalty relevance, Procedure relevance	English

HumanEval [30] is a widely used benchmark dataset designed to evaluate programming capabilities. It consists of 164 coding tasks, each accompanied by a brief natural language description. The tasks primarily involve algorithmic problems and data structure exercises, with difficulty levels ranging from basic to intermediate. One notable feature of HumanEval [30] is the inclusion

Table 3. Statistical information of different benchmarks (Part 2).

Benchmark	Domain	Type	Num	Evaluation Criteria	Language
UltraFeedback [44]	Compre.	Pairwise	64k	Helpfulness, Honesty, Instruction following, Truthfulness	English
AlpacaEval [56]	Compre.	Pairwise	20k	Instruction-following	English
Chatbot Arena [292]	Compre.	Pairwise	33k	User perference	English
MTBench [292]	Compre.	Pairwise	3,000	Multi-turn conversational, Instruction-following	English
RewardBench [116]	Compre.	Pairwise	2,998	User perference	English
JudgerBench [20]	Compre.	Pairwise	1,900	Instruction following	English Chinese
RM-Benchh [148]	Compre.	Pairwise	1,327	Instruction following	English
JUDGEBENCH [213]	Compre.	Pairwise	350	Factual, Logical correctness	English
Infinity-Preference[1]	Compre.	Pairwise	59.3k	User perference	English Chinese
LLMeval [285]	Compre.	Pointwise Pairwise	453	Correctness, Fluency, Logic, Informativeness, Harmlessness	Chinese
WildBench [138]	Compre.	Pointwise	1,024	Checklists	English
Flask [269]	Compre.	Pointwise	1,740	Logical thinking, Background knowledge, Problem handling, User alignment	English
AlignBench [143]	Compre.	Pointwise	683	Task-oriented, Clarity & Fluency, Complexity & Difficulty, Desensitization	Chinese
HELPSTEER [238]	Compre.	Pointwise Pairwise	37,120	Helpfulness, Correctness, Coherence, Complexity Verbosity	English
HELPSTEER2 [237]	Compre.	Pointwise Pairwise	21,362	Helpfulness, Correctness, Coherence, Complexity, Verbosity	English
MLLM-as-a-Judge [24]	Compre.	Pointwise Pairwise Listwise	17,000	Relevance, Accuracy, Creativity, Response granularity	English
MM-EvalMM-Eval [202]	Compre.	Pairwise	4,981	Task-oriented	Multilingual

of input-output examples, which facilitate the assessment of functional correctness. However, the dataset's limited size and scope may not sufficiently capture the diversity of real-world programming challenges. SWEBench [101] targets more complex programming tasks that are closer to real-world software development scenarios. It includes 2,294 tasks requiring advanced operations such as reasoning, multi-step problem solving, and API usage. Unlike simpler benchmarks, SWEBench [101] assesses the model's ability to handle comprehensive problem-solving and logical reasoning. However, the increased complexity also introduces challenges in establishing consistent evaluation criteria, particularly when it comes to subjective aspects like code style and efficiency. Moreover, DevAI [303] was introduced to address the limitations of existing benchmarks, which often fail to capture the iterative nature of software development and lack adequate signals for measuring long-term progress. The dataset includes 365 task requirements, focusing on more complex and challenging programming scenarios. CrossCodeEval [50] focuses on assessing cross-language programming models, containing over 1,000 tasks that involve translating code between different programming language pairs, such as Python to Java or JavaScript to C++. This dataset tests the model's ability to adapt and transform code across languages, highlighting the challenges of understanding varied syntax and semantics. CodeUltraFeedback [243] is designed to evaluate and enhance the alignment between LLMs and user-defined programming preferences. It includes

10,000 programming instructions, each paired with four responses from 14 different LLMs. These responses are scored by GPT-3.5 based on five distinct programming preferences, such as readability, efficiency, and adherence to user specifications. The dataset emphasizes fine-grained feedback and user-centered evaluation, making it a useful tool for analyzing preference alignment.

6.1.2 Machine Translation. Machine Translation (MT) refers to the process of automatically translating text from a source language to a target language. Over time, MT technology has progressed significantly, evolving from rule-based methods to Statistical Machine Translation (SMT), and more recently to Neural Machine Translation (NMT), which is now the dominant approach. With the widespread adoption of NMT and the emergence of LLMs, evaluating translation quality has become a complex task, requiring robust evaluation frameworks that can assess accuracy, fluency, and contextual relevance across diverse language pairs.

The Workshop on Machine Translation (WMT) [66] is a prominent annual evaluation event in the field of MT. It provides large-scale, human-annotated datasets for a variety of language pairs, including English-French, English-German, and English-Russian. Each year, WMT releases benchmark datasets that include source texts, model-generated translations, reference translations, and human evaluation scores. These datasets are widely used for assessing the performance of automated evaluation metrics by comparing their outputs against human judgments. WMT covers a broad range of tasks, from sentence-level translation to document-level and domain-specific challenges, making it a comprehensive resource for evaluating the correlation between automated evaluators and human assessments. However, WMT primarily focuses on high-resource languages, which may limit its applicability to low-resource or underrepresented languages. Literary Translation Comparisons [104] is designed to assess document-level translation quality, particularly in the context of literary works. It includes carefully selected paragraphs from various literary pieces, covering 18 language pairs such as Japanese-English, Polish-English, and French-English. Unlike sentence-level benchmarks, this dataset emphasizes the importance of evaluating translations in a broader context, as literary texts often require understanding of stylistic elements and cultural subtleties. This makes it particularly useful for evaluating the performance of LLMs, which may excel in capturing broader contextual information. The MQM [65] study is the largest evaluation effort to date focusing on machine translation quality. It involves professional translators annotating the outputs of top-performing systems from the WMT 2020 shared task, specifically targeting English-German and Chinese-English translations. MQM introduces a multidimensional quality assessment framework that goes beyond traditional metrics like BLEU or ROUGE. It evaluates translations across multiple dimensions, including accuracy, fluency, terminology, style, and locale, providing a more nuanced understanding of translation quality.

6.1.3 Text Summarization. Text Summarization (TS) is the task of generating a concise and coherent summary from a given piece of text while preserving its essential meaning. The main goal is to provide a quick, accurate overview of the source content, capturing key information and eliminating unnecessary details. As LLMs have shown impressive capabilities in generating summaries, the need for robust meta-evaluation benchmarks is critical to effectively assess their performance across various dimensions like coherence, relevance, consistency, and fluency.

SummEval [59] is one of the most widely used benchmarks for evaluating summarization models. It includes summaries generated by 16 different models based on 100 news articles randomly sampled from the CNN/DailyMail test set. Each summary was annotated by five independent crowd-sourced workers and three expert evaluators, using a Likert scale from 1 to 5 across four key dimensions: coherence, consistency, fluency, and relevance. The dataset is valuable for analyzing the correlation between human judgments and automated evaluation metrics. The FRANK [169] dataset is dedicated to assessing the factual accuracy of summaries generated by automatic summarization

systems. It provides detailed human annotations of factual errors, including semantic frame errors, discourse errors, and content verifiability issues. The dataset includes summaries from both the CNN/DailyMail and XSum datasets, making it a comprehensive resource for evaluating factual correctness. FRANK's detailed categorization of errors offers valuable insights into the types of factual inaccuracies common in generated summaries, highlighting areas where LLMs often struggle. However, focusing solely on factual errors may overlook other aspects of summary quality, such as coherence and fluency. OpinsummEval [194] is a meta-evaluation benchmark specifically designed for opinion summarization tasks, where the goal is to extract and summarize opinions from a large volume of user reviews. This dataset includes outputs from 14 different opinion summarization models and provides human annotations across four dimensions: aspect relevance, self-consistency, sentiment consistency, and readability.

6.1.4 Dialogue Generation. Dialogue Generation is the task of automatically generating natural language conversations that are relevant to a given context. The primary goal is to develop dialogue systems that can understand context, generate fluent responses, and maintain logical consistency and contextual accuracy. Dialogue generation encompasses a wide range of applications, from chatbots and virtual assistants to social conversational agents. With the increasing capabilities of large language models (LLMs), evaluating dialogue generation has become more complex, requiring multi-faceted evaluation frameworks to assess various aspects of conversational quality.

In the field of dialogue generation, the most commonly used datasets include Topical-Chat [72] and PERSONA-CHAT [283]. The Topical-Chat [72] dataset aims to advance research in open-domain conversational AI, covering eight major topics such as entertainment, health, and technology. The PERSONA-CHAT [283] dataset, on the other hand, focuses on enhancing dialogue systems by incorporating predefined personas to generate more personalized responses. Each dialogue participant is assigned a persona profile, consisting of several descriptive sentences about their personality or preferences. Mehri and Eskenazi [156] conducted a meta-evaluation study on these two widely-used open-domain dialogue corpora. They manually annotated 60 dialogue contexts from each dataset, with six responses per context for Topical-Chat and five for PERSONA-CHAT [283], including both model-generated and human responses. Each response was evaluated across six key dimensions: naturalness, coherence, engagement, groundedness, understandability, and overall quality. This study highlights the importance of multi-dimensional evaluation in dialogue generation, providing valuable insights into the strengths and weaknesses of different dialogue models. Additionally, the dataset from DSTC10 Track 5 [272, 279] focuses on evaluating open-domain dialogue systems and is designed for automatic evaluation and moderation of dialogue systems. The challenge aims to develop automatic evaluation mechanisms that accurately reflect human judgments while effectively handling harmful user inputs, maintaining conversational flow and engagement. The dataset includes annotations across four aspects: coherence, appropriateness, naturalness, and toxicity control.

6.1.5 Automatic Story Generation. Automatic Story Generation (ASG) is a challenging task that aims to enable models to create coherent, engaging narratives based on a given prompt or context. It emulates human storytelling abilities by generating stories that exhibit a logical structure, compelling characters, and interesting plot developments. Evaluating story generation systems is inherently complex, as it involves assessing not only linguistic quality but also narrative elements like coherence, engagement, and surprise.

The HANNA [34] dataset is tailored for evaluating automatic story generation (ASG), featuring 1,056 stories generated by 10 different systems from 96 prompts. Each story is annotated by three human reviewers across six criteria: relevance, coherence, resonance, surprise, engagement, and complexity. This comprehensive annotation framework provides a detailed assessment of narrative

quality, making HANNA a valuable benchmark for comparing ASG models. Another notable dataset is the MANS [73], which forms part of the OpenMEVA [73] framework. It compiles stories from various natural language generation models using well-known corpora like ROCStories [159] and WritingPrompts [60]. MANS [73] focuses on manual annotations of narrative elements, serving as a robust testbed for exploring diverse evaluation metrics. The StoryER [26] dataset offers a distinct approach to evaluating story generation by focusing on preference prediction and aspect-based rating. StoryER is divided into two primary components: the first is a 100k Story Ranking Data, which pairs stories from the WritingPrompts dataset. Each pair includes one story with high user engagement (upvotes ≥ 50) and another with low engagement (upvotes ≤ 0). This component leverages real-world user feedback to capture implicit preferences, providing a practical basis for training models to predict story quality. The second component, Aspect Rating and Reasoning Data, contains 46,000 entries where annotators provide detailed ratings (on a scale of 1-5) for various story aspects such as introduction, character development, and plot description, along with explanatory comments. This combination of quantitative rankings and qualitative reasoning enables a nuanced evaluation of stories, making StoryER particularly useful for both automated scoring and interpretability research. The PERSER [229] dataset takes a different approach by addressing the subjectivity inherent in open-domain text generation evaluations. PERSER restructures existing datasets and introduces personalized tags, resulting in two sub-datasets: Per-MPST and Per-DOC. Per-MPST is an adapted version of the Movie Plot Synopsis Dataset, while Per-DOC includes 7,000 instances of paired stories generated from the same premise. These stories are evaluated based on dimensions such as interestingness, adaptability, surprise, character development, and the quality of the ending.

6.1.6 Values Alignment. Values alignment is a critical task in the development of AI systems, focused on ensuring that their behavior and decisions consistently reflect core human values and ethical standards. In the context of LLM-as-Judge, the alignment process is vital to verify that the model's outputs adhere to societal norms and ethical principles, minimizing risks related to harmful, biased, or unethical behavior. To support research and model development in values alignment, several datasets have been created, each with unique characteristics designed to evaluate or enhance the ethical behavior of LLMs.

One notable dataset is PKU-SafeRLHF [96], which was specifically curated for studying safe alignment in large language models. The dataset comprises 83.4K preference entries, focusing on two primary dimensions: harmlessness and usefulness. In each sample, the dataset presents a pair of model responses to a given prompt, annotated with safety meta-labels and preferences based on the levels of safety and utility. Another influential dataset is the HHH [6] (Honesty, Helpfulness, and Harmlessness) dataset, designed to evaluate LLM performance across various human-model interaction scenarios. The dataset emphasizes three core human-centered values: honesty, helpfulness, and harmlessness. It includes a diverse collection of conversational examples where models are tested on their adherence to these values. By exposing models to a wide range of contexts, the HHH dataset serves as a comprehensive benchmark for assessing whether LLMs align with essential ethical standards and effectively mitigate risks of misinformation, harmful advice, or biased outputs. Moreover, the CVALUES [255] benchmark is a more recent contribution aimed at evaluating human values alignment specifically for Chinese LLMs. It represents the first comprehensive framework tailored to assess values alignment in the Chinese language context, focusing on two critical criteria: Safety and Responsibility.

6.1.7 Recommendation. Recommendation systems aim to provide personalized suggestions based on users' preferences and historical behavior. As the use of large language models (LLMs) expands, their role in evaluating the performance of recommendation systems has garnered increasing

attention. LLMs can serve as versatile evaluators, offering insights into multiple aspects of recommendation systems beyond traditional metrics like accuracy. They can assess factors such as user engagement, satisfaction, and the quality of generated explanations.

The MovieLens [80] dataset is a widely-used public dataset for movie recommendations, available in multiple versions with varying scales, ranging from thousands of users and ratings to millions. Zhang et al. [284] further annotated the MovieLens [80] data to create a sub-dataset featuring user self-explanation texts. In this sub-dataset, users write explanatory texts after being presented with a recommended movie. These explanations are then rated on a five-point Likert scale across four dimensions: Persuasiveness, Transparency, Accuracy, and Satisfaction. This annotated data provides valuable reference texts for LLMs in the context of explainability evaluation.

Another commonly used dataset is the Yelp dataset [4], which contains detailed review data from 11 metropolitan areas, covering approximately 150,000 businesses, nearly 7 million user reviews, and over 200,000 images. User reviews include ratings for businesses, such as hotel ratings (1-5 stars), as well as additional feedback like "cool" and "funny" votes. Furthermore, the Yelp dataset provides extensive business attribute information (e.g., operating hours, parking availability, and delivery options), offering rich contextual information that can be leveraged for developing and evaluating recommendation systems.

6.1.8 Search. The search task is a fundamental component of information retrieval (IR), focusing on identifying the most relevant documents from extensive text collections based on user queries. Traditionally, relevance assessments in search tasks have been conducted by human annotators following established guidelines. However, recent advances in large language models (LLMs) have opened up new opportunities for utilizing these models as evaluators, offering an automated and scalable approach to relevance assessment. With the advent of retrieval-augmented generation (RAG) models, the role of LLMs as evaluators has expanded. There is now a growing need to assess various dimensions of retrieved contexts, including context relevance, answer faithfulness, and answer relevance. This shift highlights the potential of LLMs to provide nuanced judgments that go beyond simple topical relevance.

A key resource for evaluating the performance of LLMs as relevance assessors is the series of datasets from the Text Retrieval Conference (TREC). TREC workshops aim to advance research in IR by offering large-scale test collections, standardized evaluation procedures, and a platform for benchmarking retrieval models. The datasets from the TREC Deep Learning Track [118], specifically from 2021 (DL21) [42] and 2022 (DL22) [43], are commonly used for this purpose. These datasets are derived from the expanded MS MARCO v2 collection [11], which contains approximately 138 million passages. Relevance judgments are provided by assessors from the National Institute of Standards and Technology (NIST) using a 4-point scale (0 to 3). This structured and fine-grained annotation scheme allows for a detailed comparison between LLM-generated relevance scores and human judgments. While general-purpose datasets offer valuable benchmarks, specialized retrieval tasks often require domain-specific datasets that reflect unique relevance criteria. One notable example is LeCaRDv2 [129], a large-scale dataset tailored for legal case retrieval. LeCaRDv2 enriches the concept of relevance by incorporating three distinct aspects: characterization, penalty, and procedure. These additional criteria provide a more comprehensive perspective on relevance.

6.1.9 Comprehensive Data. To thoroughly assess the role of LLMs-as-Judges and better align them with human preferences, a diverse set of comprehensive datasets has been developed. These datasets provide large-scale, well-annotated data, allowing for the effective training and evaluation of LLMs in complex, real-world contexts. As a result, they contribute to improving the models' reliability and effectiveness in their role as evaluators.

Datasets such as HelpSteer [238] and HelpSteer2 [237] are designed to improve the alignment and usefulness of LLMs. They provide multi-attribute data, enabling the training of models that can generate responses that are factually correct, coherent, and tailored to diverse user preferences. These open-source datasets support adjustments in response complexity and verbosity, catering to varying user needs. Additionally, UltraFeedback [44] offers a large-scale dataset with around 64,000 prompts from sources like UltraChat [49], ShareGPT [37], and TruthfulQA [140]. It includes multiple responses per prompt generated by different LLMs, with high-quality preference labels and textual feedback covering aspects like instruction-following, truthfulness, and helpfulness. UltraFeedback's fine-grained annotations and diverse prompts provide a robust resource for training reward and critic models, enhancing the evaluative capabilities of LLMs.

In exploring instruction following and dialogue capabilities, specialized tools like AlpacaEval [56], alongside interactive platforms such as Chatbot Arena [292] and benchmarks like MT-Bench [292], provide critical insights. AlpacaEval is an automated evaluation tool using GPT-4 or Claude as evaluators. It assesses chat-based LLMs against the AlpacaFarm dataset, providing win-rate calculations across a variety of tasks, enabling rapid and cost-effective comparisons with baseline models like GPT-3.5 (Davinci-003). Chatbot Arena, on the other hand, offers a user-driven evaluation framework where participants interact with anonymous models and vote based on their preferences. The platform has collected over 1,000,000 user votes, using the Bradley-Terry model to rank LLMs and chatbots, providing valuable insights into user preferences and model performance in open-domain dialogue.

Benchmarks like WildBench [138] and FLASK [269] aim to evaluate LLMs on tasks more reflective of real-world applications. WildBench [138] collects challenging examples from real users via the AI2 WildChat project, providing fine-grained annotations, task types, and checklists for response quality evaluation, and employs length-penalized Elo ratings to ensure unbiased assessments. FLASK [269] introduces a fine-grained evaluation protocol that decomposes overall scoring into skill set-level scoring for each instruction, enhancing interpretability and reliability in both human-based and model-based evaluations. Additionally, comprehensive evaluations covering multiple domains—including factual question answering, reading comprehension, summarization, mathematical problem-solving, reasoning, poetry generation, and programming—have been conducted. These evaluations involve assessing models across multiple criteria such as correctness, fluency, informativeness, logicality, and harmlessness.

Reward models and LLM-based judges face the crucial task of ensuring alignment with human expectations, a challenge addressed by datasets like RewardBench [116], RM-Bench [148], and JudgerBench [20]. RewardBench [116] focuses on assessing models through complex prompt-choice trios, covering diverse areas like chat, reasoning, and safety, with a particular emphasis on out-of-distribution scenarios. RM-Bench [148] introduces a new benchmark for evaluating reward models based on their sensitivity to subtle content differences and resistance to stylistic biases, emphasizing the need for refined assessments that correlate highly with aligned language models' performance. JudgerBench [20], with its dual components (JDB-A and JDB-B), offers a structured framework for evaluating alignment and critique abilities. By including data from human voting results and combining insights from varied sources, JudgerBench [20] provides a nuanced understanding of model performance across different languages and dialogue formats.

With the growing complexity of tasks handled by LLMs, there is an increasing demand for more objective and reliable evaluation frameworks. JUDGEBENCH [213] proposes a novel approach to assessing LLM-based judges on challenging response pairs across domains like knowledge, reasoning, mathematics, and coding. It addresses the limitations of existing benchmarks by introducing preference labels that reflect objective correctness, providing a robust platform for evaluating the capabilities of advanced LLM-based judges.

As LLMs evolve beyond text-only tasks, evaluation frameworks have expanded to encompass multimodal and multilingual contexts. MLLM-as-a-Judge [24] serves as a benchmark for assessing Multimodal LLMs, covering tasks like image description, mathematical reasoning, and infographic interpretation. By integrating human annotations, it provides a comprehensive evaluation across visual and textual domains, reflecting the growing demand for models capable of processing diverse inputs. In a parallel effort, MM-Eval [202] addresses the multilingual aspect, offering extensive analysis across 18 languages. With core subsets like Chat, Reasoning, and Linguistics, alongside a broader Language Resource subset spanning 122 languages, MM-Eval [202] highlights performance discrepancies, especially in low-resource languages where models tend to default to neutral scores.

6.2 Metric

The evaluation of LLMs-as-Judges models centers around assessing the extent to which the model's judgments align with human evaluations, which are typically considered the benchmark for quality. Given the complexity and subjectivity of many evaluation tasks, achieving high agreement with human ratings is a key indicator of the LLM's performance. To quantify this agreement, a range of statistical metrics is employed. Below, we outline these metrics and their applications in evaluating LLMs-as-Judges models.

6.2.1 Accuracy. Accuracy is a fundamental metric used to assess the proportion of correct judgments made by the LLM compared to human evaluations. In classification tasks, it is defined as:

$$\text{Accuracy} = \frac{\text{Number of Correct Predictions}}{\text{Total Number of Predictions}}, \tag{2}$$

where the number of correct predictions corresponds to instances where the LLM's judgment matches the human evaluator's judgment. While accuracy is simple to compute and intuitive, it may not fully capture the quality of the model, especially when dealing with tasks that involve nuanced or continuous evaluations.

6.2.2 Pearson Correlation Coefficient. The Pearson Correlation Coefficient [41] measures the linear relationship between two continuous variables, in this case, the evaluation scores assigned by the LLM and those assigned by human evaluators. It is defined as:

$$r = \frac{\sum (x_i - \bar{x})(y_i - \bar{y})}{\sqrt{\sum (x_i - \bar{x})^2 \sum (y_i - \bar{y})^2}}, \tag{3}$$

where x_i and y_i are the scores from the LLM and the human, respectively, and $\bar{x}$ and $\bar{y}$ are their means. Pearson correlation values range from -1 to 1.

6.2.3 Spearman's Rank Correlation Coefficient. Spearman's Rank Correlation Coefficient (ρ) [190] assesses the monotonic relationship between two variables by comparing their ranked values rather than the raw scores. It is defined as:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)}, \tag{4}$$

where d_i is the difference between the ranks of corresponding scores from the LLM and the human evaluator, and n is the number of paired scores. Spearman's ρ is less sensitive to outliers and non-linear relationships compared to Pearson's correlation, making it a robust choice for evaluating tasks where the relative order of scores is more important than the exact values. It is commonly used in ranking-based evaluations such as preference judgments or ranking tasks.

Table 4. Summary of Common Metrics for Evaluating LLMs-as-Judges Models

Metric	Type	Use Case	Robustness to Outliers
Accuracy	Agreement measure	Proportion of correct judgments	Sensitive
Pearson	Linear correlation	Continuous score comparison	Sensitive
Spearman	Rank correlation	Rank-based evaluation tasks	Robust
Kendall's Tau	Rank correlation	Consistency in ordinal rankings	Robust, handles ties
Cohen's Kappa	Agreement measure	Two raters, consistency analysis	Adjusts for chance
ICC	Agreement measure	Multiple raters, consistency analysis	Robust for group ratings

6.2.4 Kendall's Tau. Kendall's Tau (τ) [191] is another rank-based correlation metric that measures the ordinal association between two ranked lists. It is defined as:

$$\tau = \frac{C - D}{\frac{1}{2}n(n-1)}, \tag{5}$$

where C is the number of concordant pairs (where the rank order agrees between the LLM and human), and D is the number of discordant pairs. Kendall's τ is particularly useful when evaluating the consistency of rankings produced by LLMs and human evaluators. It is often preferred when the dataset contains many ties, as it provides a more nuanced measure of agreement than Spearman's ρ.

6.2.5 Cohen's Kappa. Cohen's Kappa (κ) [240] measures the level of agreement between two raters (in this case, the LLM and the human) beyond what would be expected by chance. It is defined as:

$$\kappa = \frac{p_o - p_e}{1 - p_e}, \tag{6}$$

where p_o is the observed agreement and p_e is the expected agreement by chance. Cohen's Kappa is particularly effective in classification tasks where both the LLM and the human evaluators assign categorical labels. It accounts for the possibility of random agreement, making it a more robust metric than simple accuracy.

6.2.6 Intraclass Correlation Coefficient (ICC). The Intraclass Correlation Coefficient (ICC) [13] assesses the reliability of ratings when there are multiple evaluators. It evaluates the consistency or conformity of measurements made by different raters, including LLMs and human annotators. ICC is defined based on the variance components derived from a one-way or two-way ANOVA model. The ICC is particularly useful when comparing multiple LLMs or when evaluating the consistency of an LLM across different subsets of data, providing a broader view of its reliability as an evaluator.

7 LIMITATION

Although the application of LLMs-as-judges holds great promise, there are still several significant limitations that can affect their effectiveness, reliability, and fairness [208, 215]. These limitations arise from the inherent characteristics of LLMs, including their reliance on large-scale data for training and token-based decoding mechanisms. In this section, we will primarily explore the limitations in the following three key aspects: **Biases** (§7.1), **Adversarial Attacks** (§7.2), and **Inherent Weaknesses** (§7.3).

7.1 Biases

Essentially, LLMs are trained on vast amounts of data gathered from diverse sources. While this allows them to generate human-like responses, it also makes them inherit to the biases inherent

Table 5. Overview of different biases.

Bias	Description
Presentation-Related Biases (§7.1.1)	
Position Bias	A tendency to make judgments based on the position of the input, where responses earlier or later in the sequence are favored over those in other positions.
Verbosity Bias	A tendency to favor longer responses, potentially equating length with quality, regardless of the content's actual value.
Social-Related Biases (§7.1.2)	
Authority Bias	A tendency to be swayed by references to authoritative sources, such as books, websites, or famous individuals.
Bandwagon-Effect Bias	A tendency to align with majority opinions, where LLMs-as-judges favor prevailing views over objectively assessing the content.
Compassion-Fade Bias	A tendency to be influenced by anonymization strategies, such as the removal of model names, affecting the judgments of LLMs.
Diversity Bias	A tendency to shift judgments based on identity-related markers, such as gender, ethnicity, or other social categorizations.
Content-Related Biases (§7.1.3)	
Sentiment Bias	A tendency to favor responses with specific emotional tones, such as cheerful or neutral, over negative or fearful ones.
Token Bias	A tendency of LLMs to favor more frequent or prominent tokens during pre-training, leading to skewed judgments.
Context Bias	A tendency of LLMs to produce biased judgments influenced by contextual examples or cultural contexts, potentially leading to biased or culturally insensitive outcomes.
Cognitive-Related Biases (§7.1.4)	
Overconfidence Bias	A tendency to exhibit inflated confidence in evaluation judgments, leading to overly assertive but potentially incorrect conclusions
Self-Enhancement Bias	A tendency to favor outputs generated by the same model acting as a judge, undermining objectivity.
Refinement-Aware Bias	A tendency for scoring variations influenced by whether an answer is original, refined, or accompanied by conversation history during evaluation.
Distraction Bias	A tendency to be influenced by irrelevant content, which can detract from the quality of judgments by diverting attention from critical elements.
Fallacy-Oversight Bias	A tendency to overlook logical fallacies, which can undermine the accuracy of judgments.

in the training data. These biases are presented in various forms, which can significantly affect evaluation results, compromising the fairness and accuracy of decisions.

To gain a deeper understanding of the impact of bias, we have provided a detailed classification of bias. As shown in Table 5, the biases exhibited by LLMs-as-judges can be systematically categorized into four groups based on their underlying causes and manifestations: **Presentation-Related**

Biases (§7.1.1), **Social-Related Biases** (§7.1.2), **Content-Related Biases** (§7.1.3), and **Cognitive-Related Biases** (§7.1.4). In this section, we provide a detailed overview of the definition, impact, and solutions to these biases.

7.1.1   Presentation-Related Biases. Presentation-Related Biases refer to tendencies in LLMs where judgments are influenced more by the structure or presentation of information than by its substantive content. For example, models may prioritize certain formats, styles, or patterns of expression, which can affect the quality of the input. Next, we introduce two biases related to Presentation-Related Biases: position bias and verbosity bias.

Position bias is a prevalent issue not only in the context of LLMs-as-judges but also in human decision-making and across various machine learning domains. Research has shown that humans are often influenced by the order of options presented to them, leading to biased decision that can impact fairness and objectivity [16, 182, 197, 288]. Similarly, in other ML applications, models trained on ordered data exhibit a positional preference, skewing outcomes based on the sequence of input [111, 234]. Position bias in LLMs-as-judges refers to the tendency of LLMs to favor certain answers based on their position in the response set. For example, when presented with multiple answer choices or compared pairwise, LLMs disproportionately select options that appear earlier in the list, leading to skewed judgment.

Recent studies have further examined position bias in the LLMs-as-judges context. For instance, a framework [151] is proposed to investigate position bias in pairwise comparisons, introducing metrics such as repetition stability, position consistency, and preference fairness to better understand how positions affect LLM judgments. Another study [292] explores the limitations of LLMs-as-judges, including position biases, and verifies agreement between LLM judgments and human preferences across multiple benchmarks. These findings underscore the need for robust debiasing strategies to enhance the fairness and reliableness of LLMs-as-judges.

Several methods are proposed to mitigate position bias. The naive approach involves excluding inconsistent judgments by swapping the positions of the candidate answers and verifying whether the LLM's judgment remains consistent. Inconsistent judgments are then filtered out [25, 127, 130, 230, 291, 292]. The **swap-based** debiasing method can be further divided into two categories: *score-based* and *comparison-based*. Both approaches start by swapping the positions of the candidate answers. The difference lies in how the final judgment is determined. In the score-based method, each candidate answer is scored, and the average score across multiple swaps is taken as the final score for that answer [87, 130, 184, 230, 292].In contrast, the comparison-based method considers the outcome a tie if the LLM's judgments are inconsistent after swapping. The conclusion of a tie is based on an analysis of the quality gap between answers. The larger the quality gap between candidate answers, the smaller the impact of position bias, resulting in higher consistency in predictions after swapping their positions, which is detailed in a recently study [151]. In addition to the aforementioned methods, PORTIA [134] employs an **alignment-based** approach that simulates human comparison strategies. It divides each answer into multiple segments, aligns similar content across candidate answers, and then merges these aligned segments into a single prompt for the LLM to evaluate. By presenting content in a balanced and aligned format, PORTIA enables the model to make more consistent and unbiased judgments, focusing on answer quality rather than order. This approach is effective across various LLMs, significantly improving evaluation consistency and reducing costs. Further efforts to enhance LLM-based evaluations have explored new techniques to address position bias and other judgment inconsistencies. **Discussion-based** methods [107, 132] incorporate peer ranking and discussion to improve evaluation accuracy. Instead of relying solely on a single LLM's judgment, these methods prompt multiple LLMs to compare answers and discuss preferences to reach a consensus, thereby reducing individual positional bias and enhancing

alignment with human judgments. This collaborative evaluation approach represents a promising direction for mitigating biases inherent in LLM assessments.

Verbosity bias [25, 107, 163, 292] refers to the tendency of a judge, whether human or model-based, to favor lengthier responses over shorter ones, irrespective of the actual content quality or relevance. This bias may cause LLMs prefer longer responses, even if the extended content does not contribute substantively to the correctness of the judgments.

To mitigate verbosity bias in LLMs-as-judges, several approaches [107, 267, 270] have been proposed. One approach [107] employs persuasive debating techniques, structuring responses to prioritize substance. By training LLMs-as-judges to engage in a debate-like format, this method encourages clarity and relevance, reducing the tendency to favor verbose arguments that lack substantive content. Additionally, the CALM [267] framework introduces controlled modifications to systematically assess and quantify verbosity's impact on judgments, using automated perturbations to evaluate robustness against verbosity bias and refine LLMs-as-judges toward objective and concise assessments. Complementing these methods, the contrastive judgments (Con-J) [270] approach trains models with structured rationale pairs instead of scalar scores, encouraging LLMs-as-judges to focus on well-reasoned content rather than associating verbosity with quality.

7.1.2 Social-Related Biases. Social-Related Biases refer to biases in language models that resemble social phenomena [289]. These biases may manifest when models are swayed by references to authoritative sources (Authority Bias), align with prevailing majority opinions without independent evaluation (Bandwagon-Effect Bias), or adjust their judgments based on anonymization strategies or identity markers such as gender and ethnicity (Compassion-Fade Bias and Diversity Bias). Next, we present the details of these biases.

Authority bias [25, 267] in the context of LLMs-as-judges refers to the tendency of the model to attribute greater credibility to statements associated with authoritative references, regardless of the actual evidence supporting them. For instance, LLMs-as-judges may favor responses that include references to well-known sources or experts, even when the content is inaccurate or irrelevant. This bias highlights a critical vulnerability where the appearance of authority can unduly influence judgment outcomes.

While specific solutions to mitigate authority bias in LLMs-as-judges are still under active exploration, potential approaches include using retrieval-augmented generation (RAG) techniques to verify the validity of authoritative claims against external knowledge bases. This approach allows the model to cross-check referenced information and ensure its alignment with factual evidence. Another possible strategy is to design prompts that explicitly emphasize semantic accuracy and relevance over perceived authority. Further research is needed to validate these approaches and develop robust methods for addressing authority bias effectively in evaluative contexts.

Bandwagon-effect bias [112, 267] in the context of LLMs-as-judges refers to the tendency of the model to align its judgments with the majority opinion or prevailing trends, regardless of the actual quality or correctness of the evaluated content. For instance, when multiple responses are presented with indications of popular support or consensus, LLMs-as-judges may disproportionately favor these responses over alternatives, even when the consensus is flawed or biased. This bias reflects a susceptibility to groupthink dynamics, undermining the objectivity and fairness of the judgment process.

Solutions to address bandwagon-effect bias include designing evaluation prompts that anonymize information about majority opinions, ensuring that judgments are based solely on the intrinsic quality of the responses rather than external indicators of popularity. Further exploration of debiasing strategies tailored to specific evaluative contexts is necessary to mitigate the impact of bandwagon-effect bias effectively.

Compassion-fade bias [112, 267] occurs when the anonymity of model names or the absence of identifiable contextual cues affects the judgments made by LLMs-as-judges. For example, anonymizing model names or using neutral identifiers may lead to shifts in evaluation outcomes. This bias highlights how the lack of personalized or contextual information can diminish the model's sensitivity to equitable considerations.

To mitigate compassion-fade bias, it is important to design evaluation prompts that standardize judgment criteria, ensuring that assessments remain consistent regardless of whether identifying details are present. Additionally, fairness-driven frameworks that explicitly address anonymization effects can further enhance the reliability of LLMs-as-judges.

Diversity bias [25, 267] in the context of LLMs-as-judges refers to the model's tendency to exhibit judgment shifts based on identity-related markers, such as gender, ethnicity, religion, or other social categorizations. For example, LLMs-as-judges might favor responses associated with certain demographic groups over others, leading to unfair or skewed judgments. This bias reflects the model's susceptibility to implicit stereotypes or unequal treatment of diverse identities present in the training data. Continued efforts to address this bias are crucial for ensuring fairness and inclusivity in the judgments conducted by LLMs-as-judges.

7.1.3 Content-Related Biases. Content-Related Biases involve preferences or skewed judgments based on the content's characteristics. An LLM might favor responses with certain emotional tones (Sentiment Bias), prefer frequently occurring words from its training data (Token Bias), or be influenced by specific cultural or domain contexts leading to insensitive outcomes (Context Bias). We present the details of these biases in the following.

Sentiment bias [267] in the context of LLMs-as-judges refers to the tendency of the model to favor responses that exhibit certain emotional tones, such as positive or neutral sentiments, over others, regardless of their actual content quality or relevance. For instance, LLMs-as-judges may disproportionately reward responses that are cheerful or optimistic while penalizing those that are negative or emotionally intense, even if the latter are more contextually appropriate or accurate.

To address sentiment bias, potential solution is the use of sentiment-neutralizing mechanisms, such as filtering or adjusting responses to remove sentiment-driven influences during evaluation.

Token Bias [98, 127, 178, 184] refers to that LLMs favor certain tokens during the evaluation process. This bias often arises from the model's pre-training data, where more frequently occurring tokens are prioritized over less common ones, regardless of the contextual appropriateness or correctness in judgment.

Contextual Bias refers to the tendency of LLMs to produce skewed or biased judgments based on the specific context in which they are applied. For instance, models used in healthcare may propagate biases found in medical datasets, potentially influencing diagnoses or treatment recommendations [179], while in finance, they might reflect biases in credit scoring or loan approval processes [300]. In addition, the selection of contextual examples may also introduce bias [62, 78, 290, 296].

7.1.4 Cognitive-Related Biases. Cognitive-Related Biases pertain to the inherent cognitive tendencies of LLMs in processing information. This includes exhibiting unwarranted confidence in judgments (Overconfidence Bias), favoring outputs generated by themselves (Self-Enhancement Bias), varying scores based on whether an answer is original or refined (Refinement-Aware Bias), being distracted by irrelevant information (Distraction Bias), or overlooking logical fallacies (Fallacy-Oversight Bias). The details of these biases are presented in the following.

Overconfidence bias [103, 107] in the context of LLMs-as-judges refers to the tendency of models to exhibit an inflated level of confidence in their judgments, often resulting in overly assertive evaluations that may not accurately reflect the true reliability of the answer. This bias is particularly

concerning in evaluative contexts, as it can lead LLMs-as-judges to overstate the correctness of certain outputs, compromising the objectivity and dependability of assessments.

To address Overconfidence bias, researchers have proposed several methods. Cascaded Selective Evaluation [103] addresses overconfidence by using Simulated Annotators to estimate confidence. This involves simulating diverse annotator preferences through in-context learning, which provides a more realistic measure of the likelihood that a human would agree with the LLM's judgment. By analyzing multiple simulated responses, this method offers a confidence metric that reflects human-like disagreement, which helps to avoid overconfidence bias. Another method uses an adversarial debate mechanism, where two LLMs argue for different outcomes. Through structured debate rounds, each model is required to substantiate its position, which can reveal overconfidence by prompting self-reflection and critical analysis. This approach has been shown to improve truthfulness and reduces overconfidence by fostering a balanced evaluation, aligning LLMs' judgments more closely with accurate and reasoned conclusions.

Self-enhancement bias is the tendency to favor their own outputs [132, 145, 145, 172, 292]. This concept of self-enhancement is drawn from social psychology, as discussed in Brown's work in social cognition literature [19]. In the context of LLMs-as-judges, this bias manifests when a LLM evaluates its own generated outputs more favorably than those of other LLMs. Such bias is particularly concerning in applications involving self-assessment or feedback generation, as it compromises the objectivity of the LLMs-as-judges.

To address self-enhancement bias, PRD [132] introduces Peer Rank (PR) and Peer Discussion (PD) mechanisms. PR mitigates bias by using multiple LLMs as reviewers, each assessing pairwise comparisons between responses from different LLMs. By aggregating evaluations from several peer LLMs and weighting their preferences based on consistency with human judgments, PR reduces the impact of any single LLM's self-enhancement bias, as more reliable reviewers have a greater influence. PD further alleviates self-enhancement bias by enabling two LLMs to engage in a dialogue to reach a mutual agreement on their preference between two answers. This multi-turn discussion encourages models to re-evaluate their initial judgments and consider alternative perspectives, focusing on content quality rather than self-generated responses. By promoting collaborative assessment and accountability, PR and PD effectively mitigate self-enhancement bias, aligning evaluations more closely with human standards. Recently, an automated bias quantification framework named CALM [267] has been proposed to systematically evaluate biases in LLMs-as-judges. CALM's findings suggest that one effective way to reduce self-enhancement bias is to avoid using the same model to both generate and judge answers, thereby ensuring that evaluation remains more impartial. Moreover, the Reference-Guided Verdict method [8] further addresses self-enhancement bias by providing a definitive gold-standard answer as a reference for LLM judges. This reference anchor helps align judgments to objective criteria, even when an LLM evaluates its own output, thus reducing the tendency to favor self-generated answers. Through structured prompts, this method has been shown to enhance reliability and mitigate variability in judgments, especially when multiple LLMs are used collectively. The integration of multiple LLMs, trained on varied datasets or fine-tuned with different parameters, has proven instrumental in producing less biased, more balanced evaluations, highlighting the effectiveness of model diversity and reference-guided criteria in combating self-enhancement bias.

Refinement-aware bias [267] in the context of LLMs-as-judges refers to the tendency of the model to evaluate responses differently based on whether they are original, refined, or include revision history. For instance, an answer that has been iteratively refined may be judged more favorably than an original response, even if the refinement process does not significantly improve the content quality. Similarly, responses that explicitly present their improvement process or revision rationale might receive undue preference, skewing the evaluation outcomes.

While research on refinement-aware bias in the specific context of LLMs-as-judges remains limited, solution [259] developed for general LLMs offer valuable insights. One potential solution involves incorporating external feedback mechanisms during judgment, as it introduces an objective and independent judgment mechanism that is not influenced by the LLM's internal iterations or self-perception.

Distraction bias in the context of LLMs-as-judges refers to the model's tendency to be influenced by irrelevant or unimportant details when making judgments. For instance, introducing unrelated information, such as a meaningless statement like "System Star likes to eat oranges and apples," [112, 195, 267] can significantly alter the model's evaluation outcomes. This bias highlights the vulnerability of LLMs to attentional diversion caused by inconsequential content.

While existing studies [112, 267] have analyzed and discussed distraction bias, effective strategies to mitigate this issue in LLMs-as-judges remain underexplored. Potential solutions could involve input sanitization to preprocess and remove irrelevant information before presenting it to the model, ensuring that evaluations focus solely on relevant content. Additionally, explicit prompting with clear and strict guidelines could be designed to direct the LLM to evaluate only task-related aspects, reducing its susceptibility to distractions. Further research is needed to develop and validate robust methods that can systematically address distraction bias in the context of LLMs-as-judges.

Fallacy-oversight bias [25, 267] refers to the tendency of LLMs-as-judges to overlook logical fallacies or inconsistencies within the evaluated responses. For instance, when presented with arguments or answers containing reasoning errors—such as circular reasoning, false dilemmas, or strawman arguments—LLMs-as-judges may fail to identify these issues and treat the responses as valid, potentially compromising the integrity of their evaluations.

In summary, while LLMs-as-judges have garnered significant attention for their effectiveness in diverse scenarios, the exploration of various biases that impact their performance remains relatively underdeveloped. These biases pose significant challenges to ensuring fair, objective, and reliable judgments across tasks, particularly in various applications where the implications of biased judgments can be severe. Future research must focus on systematically identifying, quantifying, and addressing these biases within the LLMs-as-judges framework. Drawing from methodologies developed for general LLMs, such as external feedback mechanisms, balanced datasets, and fairness-aware prompting, could offer initial insights. However, domain-specific challenges require tailored solutions that align with the unique demands of LLMs-as-judges.

7.2 Adversarial Attacks

Adversarial attacks involve carefully crafted inputs designed to deceive the model into producing incorrect or unintended outputs. For LLM judges, attackers may subtly modify the input content, alter the wording of questions, or introduce misleading context to influence the model's evaluation results. Researchers have found that for LLMs, even small, seemingly insignificant changes to the input data, such as adding or removing words, changing word order, or introducing ambiguous phrasing, can significantly affect the model's response [100, 193, 305]. Such attacks can lead to inaccurate ratings or assessments, particularly when evaluating complex or high-risk tasks.

In this section, we first review research on adversarial attacks on LLMs within general domains. Then, we specifically focus on adversarial attacks in the context of LLMs-as-judges.

7.2.1 Adversarial Attacks on LLMs. Adversarial attacks on LLMs focus on exploiting vulnerabilities within the general framework of language model functionality. These attacks can be classified into three main categories based on the manipulation level: text-level manipulations, structural and semantic distortions, and optimization-based attacks.

Text-Level Manipulations involve subtle changes to the input text to deceive the model. Character-level perturbations, such as introducing typos, swapping letters, or inserting unnecessary characters, can cause significant changes in predictions despite minimal visible alterations [57, 100]. Sentence-level modifications, such as rearranging phrases, adding irrelevant information, or paraphrasing inputs, further exploit the model's sensitivity to surface-level changes [18, 177].

Structural and Semantic Distortions focus on the syntactic and semantic properties of the input. Syntactic attacks rewrite sentence structures while preserving semantic meaning, targeting the model's reliance on specific linguistic patterns [260]. Semantic preservation with perturbations modifies critical tokens identified through saliency analysis, ensuring the attack minimally affects meaning but significantly alters predictions.

Optimization-Based Attacks leverage algorithmic techniques to craft adversarial inputs. Gradient-based methods utilize the model's gradients to identify and manipulate influential input features, causing substantial shifts in predictions [210, 211]. Population-based optimization techniques iteratively generate adversarial examples in black-box settings, exploiting the model's outputs to refine attacks [120].

These attacks highlight the vulnerabilities in LLMs, demonstrating their susceptibility to subtle manipulations. Studying these adversarial attacks is essential, as it provides insights that can guide the development of robust defense mechanisms, ensuring that LLMs maintain reliability against such manipulations.

7.2.2 Adversarial Attacks on LLMs-as-judges. Recent studies have unveiled significant vulnerabilities in LLMs-as-judges to adversarial attacks [51, 184, 196, 293]. Zheng et al. [293] and Doddapaneni et al. [51] demonstrated that automatic benchmarking systems like MT-Bench [292] can be easily deceived to yield artificially high scores. These findings highlight that malicious inputs can manipulate evaluation metrics, undermining the reliability of such benchmarks.

Building on this, Raina et al. [184] investigated the robustness of LLMs-as-judges against universal adversarial attacks. Their work showed that appending short, carefully crafted phrases to evaluated texts can effortlessly manipulate LLM scores, inflating them to their maximum regardless of the actual quality. Remarkably, these universal attack phrases are transferable across models; phrases optimized on smaller surrogate models (e.g., FlanT5-xl [40]) can successfully deceive larger models like GPT-3.5 and Llama2 [218].

Furthermore, Shi et al. [196] introduced *JudgeDeceiver*, an optimization-based prompt injection attack tailored for the LLMs-as-judges framework. Unlike handcrafted methods, JudgeDeceiver formulates a precise optimization objective to efficiently generate adversarial sequences. These sequences can mislead LLMs-as-judges into selecting biased or incorrect responses among candidate answers, thereby compromising the evaluation process.

Although preliminary studies [184, 196] have highlighted the vulnerability of LLMs-as-judges to adversarial manipulations, this field remains largely underexplored. It is imperative to advance our understanding of these weaknesses and devise effective defense strategies. As the use of LLMs-as-judges grows across diverse applications, future research should focus on uncovering new attack methods and strengthening the models against such adversarial threats.

7.3 Inherent Weaknesses

Despite the remarkable capabilities of LLMs, they possess several inherent weaknesses that can compromise their reliability and robustness in LLMs-as-judges. This subsection discusses key limitations, including issues related to knowledge recency, hallucination, and other domain-specific knowledge gaps [287].

7.3.1   Knowledge Recency. One significant limitation of LLMs is their inability to access or incorporate up-to-date information reliably. LLMs are generally trained on static datasets that may become outdated over time, limiting their ability to evaluate scenarios that require knowledge of recent events, legislation, or rapidly evolving fields. The most straightforward solution is to retrain the model on new data; however, this approach is resource-intensive and risks catastrophic forgetting [152], where previously learned knowledge is overwritten during training. This temporal disconnect can lead to judgments based on invalid data, or obsolete practices, compromising their reliability in real-world, time-sensitive applications. Consider a case where LLMs-as-judges are used to evaluate which of two responses from LLMs better answers a prompt about the COVID-19 pandemic. Suppose one response references the WHO guidelines updated timely, while the other relies on outdated 2020 guidelines. If the LLM-as-Judge has not been updated with the latest guidelines, it might erroneously prefer the outdated response, incorrectly deeming it more accurate. This failure to account for recent developments highlights the importance of addressing knowledge recency in LLMs-as-judges.

Addressing the issue of knowledge recency can involve integrating retrieval-augmented generation (RAG) methods [69, 125], which enable LLMs to query external, dynamically updated databases or knowledge sources during evaluation. Additionally, periodic fine-tuning with updated datasets or leveraging continual learning frameworks [244] can ensure that LLMs-as-judges remain aligned with the latest information. Combining these approaches with robust fact-checking mechanisms [48] can further enhance temporal reliability in judgment contexts.

7.3.2   Hallucination. Another critical issue in LLMs is the hallucination problem, where models generate incorrect or fabricated information with high confidence. In the context of LLMs-as-judges, hallucination can manifest as the invention of non-existent precedents, misinterpretation of facts, or fabrication of sources, which can severely undermine the reliability of their judgments. This issue is particularly concerning in various applications, where such errors can lead to unfair or harmful outcomes.

Employing fact-checking mechanisms [48, 97, 216] during evaluation is crucial to mitigate hallucination. By cross-verifying the outputs of LLMs-as-judges with trusted databases and external knowledge sources, hallucinated information can be identified and corrected.

7.3.3   Domain-Specific Knowledge Gaps. While LLMs demonstrate broad generalization capabilities, they often lack the depth of understanding required for specialized domains [55, 63, 171, 212]. For instance, legal judgments demand intricate knowledge of statutes, precedents, and contextual nuances, which may not be adequately captured in the training data of general-purpose LLMs. This limitation can lead to shallow or incorrect judgments in domain-specific contexts.

Domain adaptation techniques, such as integrating LLMs with domain-specific knowledge graphs [63, 171] or leveraging RAG systems [69], can substantially improve their performance in specialized domains. Knowledge graphs provide structured, expert-curated information that enhances context-awareness, while RAG enables LLMs to dynamically retrieve relevant knowledge from specific domain.

The inherent weaknesses of LLMs highlight the need for continued research and innovation. Addressing these limitations through RAG, enhanced training methods, and knowledge graph techniques is crucial for ensuring that LLMs-as-judges deliver reliable, accurate, and trustworthy evaluations in diverse applications.

8   FUTURE WORK

In this section, we will explore the key directions for future work, focusing on how to build more efficient, more effective, and more reliable LLM judges. These directions aim to address the

bottlenecks and challenges in current technologies and practices, while also promoting broader applications and deeper integration of LLM judges in diverse scenarios.

8.1 More Efficient LLMs-as-Judges

8.1.1 Automated Construction of Evaluation Criteria and Tasks. Current LLM judges often rely on manually predefined evaluation criteria, lacking the ability to adapt dynamically during the assessment process. Designing prompts for these systems is not only tedious and time-consuming but also struggles to address the diverse requirements of various task scenarios [10, 233, 273, 280, 286]. To overcome these limitations, future LLM judges could incorporate enhanced adaptability by tailoring evaluation criteria based on task types, target audiences, and domain-specific knowledge. Such advancements would significantly streamline the configuration process of LLM judges, while also greatly improving their practicality and efficiency in real-world applications.

Moreover, existing static evaluation datasets are prone to issues such as training data contamination, which can compromise their effectiveness in accurately assessing the evolving capabilities of LLMs. To address this, future LLM judges could focus on dynamically constructing more suitable evaluation tasks and continuously optimizing the evaluation process, thereby enhancing applicability and precision [10, 286].

8.1.2 Scalable Evaluation Systems. Existing LLM judges often exhibit limited adaptability in practical applications. While these judges may perform effectively on specific downstream tasks, they frequently struggle in cross-domain or multi-task settings, thereby falling short of meeting the diverse and broader demands of real-world applications.

To address these limitations, future research could focus on modular design principles to create scalable evaluation frameworks [257]. Such frameworks would allow users to flexibly add or customize evaluation modules to suit their specific needs. This modular approach not only enhances the usability and flexibility of the system but also significantly reduces the cost and complexity of transferring the framework across different domains.

8.1.3 Accelerating Evaluation Processes. Existing LLM systems often face significant computational costs when performing evaluation tasks. For example, pairwise comparison methods require multiple rounds of comparisons for each candidate, which becomes extremely time-consuming as the number of candidates grows. In resource-constrained environments, such high-cost evaluation methods are challenging to deploy effectively. To address this issue, future research could focus on developing more efficient candidate selection algorithms, thereby unlocking new opportunities for the use of LLMs in low-resource settings [123, 149].

Similarly, the multi-LLM evaluation paradigm, which relies on multiple rounds of interaction, further exacerbates computational demands. To mitigate these challenges, future efforts could explore streamlined communication frameworks that support high-quality evaluation tasks while minimizing resource requirements [31]. Advances in these areas could lead to the development of more efficient and scalable evaluation systems, making LLM-based evaluations more practical across diverse and resource-limited scenarios.

8.2 More Effective LLMs-as-Judges

8.2.1 Integration of Reasoning and Judge Capabilities. Current LLMs-as-judges systems often treat reasoning and evaluation capabilities as distinct and independent modules, which can hinder effectiveness when addressing complex tasks. As the demand for evaluating increasingly complex systems grows, future LLM-as-Judge systems should prioritize the deep integration of reasoning and evaluation capabilities to achieve a seamless synergy [207, 271, 304]. For instance, in legal scenarios,

the model could first infer the relevant legal provisions and then assess the case's relevance, making the evaluation process more effective.

8.2.2   Establishing a Collective Judgment Mechanism. Current LLMs-as-Judge systems typically rely on a single model for evaluation. While this approach is straightforward, it is prone to biases inherent in individual models, leading to reduced accuracy and stabilitys. Moreover, a single model often struggles to comprehensively address the diverse requirements of such tasks. Future research could investigate collaborative multi-agent mechanisms to enable "collective judgment" where multiple LLMs work together, leveraging their respective strengths in reasoning and knowledge [22, 39],. Additionally, ensemble techniques could be employed to dynamically balance the contributions of different models, leading to more stable and reliable judgment outcomes.

8.2.3   Enhancing Domain Knowledge. Current LLMs-as-Judge systems often fall short when handling tasks in specialized fields due to insufficient domain knowledge. Furthermore, as domain knowledge continues to evolve, these models struggle to keep up with the latest developments, further limiting their effectiveness and applicability in real-world scenarios.

To address these challenges, future LLMs-as-judges systems should focus on integrating comprehensive domain knowledge to enhance their performance in specialized tasks [185]. This can be achieved by utilizing knowledge graphs, embedding domain-specific expertise, and fine-tuning models based on feedback from subject-matter experts. In addition, these systems should incorporate dynamic knowledge updating capabilities. For instance, in the legal domain, models could regularly acquire and integrate updates on new statutes, case law, and policy changes, ensuring that their judgments remain current and aligned with the latest legal standards.

8.2.4   Cross-Domain and Cross-Language Transferability. Current LLMs-as-Judge systems are often confined to specific domains or languages, making it challenging for them to transfer across different fields. For instance, an LLM proficient in processing legal texts may struggle to effectively handle evaluation tasks in the medical or financial domains. This limitation greatly restricts the applicability of such systems.

Future research can focus on exploring cross-domain and cross-language transfer learning techniques to enhance the adaptability of LLMs in diverse fields. By leveraging shared general knowledge across fields, models can quickly adapt to new tasks with minimal additional training costs [77, 202, 241]. For example, evaluation capabilities developed in English could be transferred to contexts in German, thereby improving the evaluation performance in these new areas.

8.2.5   Multimodal Integration Evaluation. Current LLM-as-Judge systems primarily focus on processing textual data, with limited attention to integrating other modalities like images, audio, and video. This single-modal approach falls short in complex scenarios requiring multimodal analysis, such as combining visual and textual information in medical assessments. Future systems should develop cross-modal integration capabilities to process and evaluate multimodal data simultaneously [24]. Leveraging cross-modal validation can enhance evaluation accuracy. Key research areas include efficient multimodal feature extraction, integration, and the design of unified frameworks for more comprehensive and precise evaluations.

8.3   More Reliable LLMs-as-Judges

8.3.1   Enhancing Interpretability and Transparency. Current LLM-as-Judge systems often operate as black boxes, with their rulings lacking transparency and a clear reasoning process. This opacity is particularly concerning in high-stakes domains such as legal judgments, where users cannot fully understand the basis of the model's decisions or trust its outputs. Future research should focus on improving the interpretability of LLMs [147]. For example, the LLM judges should not only provide

evaluation results but also present a clear explanation. Research could explore designing validation models based on logical frameworks to make the decision-making process more transparent.

8.3.2 Mitigating Bias and Ensuring Fairness. LLMs may be influenced by biases present in their training data, leading to unfair judgments in different social, cultural, or legal contexts. These biases could be amplified by the model and compromise the fairness of its decisions. Future research could focus on ensuring fairness in model outputs through debiasing algorithms and fairness constraints [127]. Targeted approaches, such as adversarial debiasing training or bias detection tools, can dynamically identify and mitigate potential biases during the model's reasoning process.

8.3.3 Enhancing Robustness. LLMs are sensitive to noise, incompleteness, or ambiguity in input instructions, which may lead to errors or instability in evaluation results when handling complex or highly uncertain texts. This lack of robustness significantly limits their reliability in practical applications. Future research can adopt several methods to enable LLMs robust and reliable performance in real-world environments [58, 196]. For instance, introducing more advanced data augmentation techniques to generate diverse and uncertain simulated cases can help train models to adapt to various complex input conditions.

9 CONCLUSION

This survey systematically examined the LLMs-as-Judges framework across five dimensions: functionality, methodology, applications, meta-evaluation, and limitations, providing a comprehensive understanding of its advantages, limitations, practical implementations, applications, and methods for evaluating its effectiveness. To advance research in this field, we also outlined several promising directions for future exploration, including the development of more efficient, effective, and reliable LLM judges. We hope to promote the ongoing development of this field by providing foundational resources and will continue to update relevant content.

REFERENCES

[1] Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774* (2023).

[2] Samee Arif, Sualeha Farid, Abdul Hameed Azeemi, Awais Athar, and Agha Ali Raza. 2024. The fellowship of the llms: Multi-agent workflows for synthetic preference optimization dataset generation. *arXiv preprint arXiv:2408.08688* (2024).

[3] Akari Asai, Zeqiu Wu, Yizhong Wang, Avirup Sil, and Hannaneh Hajishirzi. 2023. Self-rag: Learning to retrieve, generate, and critique through self-reflection. *arXiv preprint arXiv:2310.11511* (2023).

[4] Nabiha Asghar. 2016. Yelp dataset challenge: Review rating prediction. *arXiv preprint arXiv:1605.05362* (2016).

[5] Zahra Ashktorab, Michael Desmond, Qian Pan, James M Johnson, Martin Santillan Cooper, Elizabeth M Daly, Rahul Nair, Tejaswini Pedapati, Swapnaja Achintalwar, and Werner Geyer. 2024. Aligning Human and LLM Judgments: Insights from EvalAssist on Task-Specific Evaluations and AI-assisted Assessment Strategy Preferences. *arXiv preprint arXiv:2410.00873* (2024).

[6] A Askell, Y Bai, A Chen, D Drain, D Ganguli, T Henighan, A Jones, N Joseph, B Mann, N DasSarma, et al. 2021. A general language assistant as a laboratory for alignment. arXiv. *Preprint posted online December* 1 (2021).

[7] Golnoosh Babaei and Paolo Giudici. 2024. GPT classifications, with application to credit lending. *Machine Learning with Applications* 16 (2024), 100534.

[8] Sher Badshah and Hassan Sajjad. 2024. Reference-Guided Verdict: LLMs-as-Judges in Automatic Evaluation of Free-Form Text. *arXiv preprint arXiv:2408.09235* (2024).

[9] Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609* (2023).

[10] Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, et al. 2024. Benchmarking foundation models with language-model-as-an-examiner. *Advances in Neural Information Processing Systems* 36 (2024).

[11] Payal Bajaj, Daniel Campos, Nick Craswell, Li Deng, Jianfeng Gao, Xiaodong Liu, Rangan Majumder, Andrew McNamara, Bhaskar Mitra, Tri Nguyen, et al. 2016. Ms marco: A human generated machine reading comprehension dataset. *arXiv preprint arXiv:1611.09268* (2016).

[12] Chaithanya Bandi and Abir Harrasse. 2024. Adversarial Multi-Agent Evaluation of Large Language Models through Iterative Debates. *arXiv preprint arXiv:2410.04663* (2024).

[13] John J Bartko. 1966. The intraclass correlation coefficient as a measure of reliability. *Psychological reports* 19, 1 (1966), 3–11.

[14] Anna Bavaresco, Raffaella Bernardi, Leonardo Bertolazzi, Desmond Elliott, Raquel Fernández, Albert Gatt, Esam Ghaleb, Mario Giulianelli, Michael Hanna, Alexander Koller, et al. 2024. Llms instead of human judges? a large scale empirical study across 20 nlp evaluation tasks. *arXiv preprint arXiv:2406.18403* (2024).

[15] Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, et al. 2024. Graph of thoughts: Solving elaborate problems with large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Vol. 38. 17682–17690.

[16] Niels J Blunch. 1984. Position bias in multiple-choice questions. *Journal of Marketing Research* 21, 2 (1984), 216–220.

[17] Nathan Brake and Thomas Schaaf. 2024. Comparing Two Model Designs for Clinical Note Generation; Is an LLM a Useful Evaluator of Consistency? *arXiv preprint arXiv:2404.06503* (2024).

[18] Hezekiah J Branch, Jonathan Rodriguez Cefalu, Jeremy McHugh, Leyla Hujer, Aditya Bahl, Daniel del Castillo Iglesias, Ron Heichman, and Ramesh Darwishi. 2022. Evaluating the susceptibility of pre-trained language models via handcrafted adversarial examples. *arXiv preprint arXiv:2209.02128* (2022).

[19] Jonathon D Brown. 1986. Evaluations of self and others: Self-enhancement biases in social judgments. *Social cognition* 4, 4 (1986), 353–376.

[20] Maosong Cao, Alexander Lam, Haodong Duan, Hongwei Liu, Songyang Zhang, and Kai Chen. 2024. CompassJudger-1: All-in-one Judge Model Helps Model Evaluation and Evolution. *arXiv preprint arXiv:2410.16256* (2024).

[21] Meng Cao, Lei Shu, Lei Yu, Yun Zhu, Nevan Wichers, Yinxiao Liu, and Lei Meng. 2024. Enhancing Reinforcement Learning with Dense Rewards from Language Model Critic. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 9119–9138.

[22] Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. 2023. Chateval: Towards better llm-based evaluators through multi-agent debate. *arXiv preprint arXiv:2308.07201* (2023).

[23] Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology* 15, 3 (2024), 1–45.

[24] Dongping Chen, Ruoxi Chen, Shilin Zhang, Yinuo Liu, Yaochen Wang, Huichi Zhou, Qihui Zhang, Yao Wan, Pan Zhou, and Lichao Sun. 2024. Mllm-as-a-judge: Assessing multimodal llm-as-a-judge with vision-language benchmark. *arXiv preprint arXiv:2402.04788* (2024).

[25] Guiming Hardy Chen, Shunian Chen, Ziche Liu, Feng Jiang, and Benyou Wang. 2024. Humans or llms as the judge? a study on judgement biases. *arXiv preprint arXiv:2402.10669* (2024).

[26] Hong Chen, Duc Minh Vo, Hiroya Takamura, Yusuke Miyao, and Hideki Nakayama. 2023. StoryER: Automatic story evaluation via ranking, rating and reasoning. *Journal of Natural Language Processing* 30, 1 (2023), 243–249.

[27] Junjie Chen, Weihang Su, Zhumin Chu, Haitao Li, Qinyao Ai, Yiqun Liu, Min Zhang, and Shaoping Ma. 2024. An Automatic and Cost-Efficient Peer-Review Framework for Language Generation Evaluation. arXiv:2410.12265 [cs.CL] https://arxiv.org/abs/2410.12265

[28] Jiefeng Chen, Jinsung Yoon, Sayna Ebrahimi, Sercan O Arik, Tomas Pfister, and Somesh Jha. 2023. Adaptation with self-evaluation to improve selective prediction in llms. *arXiv preprint arXiv:2310.11689* (2023).

[29] Kai Chen, Yanze Li, Wenhua Zhang, Yanxin Liu, Pengxiang Li, Ruiyuan Gao, Lanqing Hong, Meng Tian, Xinhai Zhao, Zhenguo Li, et al. 2024. Automated evaluation of large vision-language models on self-driving corner cases. *arXiv preprint arXiv:2404.10595* (2024).

[30] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374* (2021).

[31] Weize Chen, Ziming You, Ran Li, Yitong Guan, Chen Qian, Chenyang Zhao, Cheng Yang, Ruobing Xie, Zhiyuan Liu, and Maosong Sun. 2024. Internet of agents: Weaving a web of heterogeneous agents for collaborative intelligence. *arXiv preprint arXiv:2407.07061* (2024).

[32] Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. 2023. Teaching large language models to self-debug. *arXiv preprint arXiv:2304.05128* (2023).

[33] Yen-Shan Chen, Jing Jin, Peng-Ting Kuo, Chao-Wei Huang, and Yun-Nung Chen. 2024. LLMs are Biased Evaluators But Not Biased for Retrieval Augmented Generation. *arXiv preprint arXiv:2410.20833* (2024).

[34] Cyril Chhun, Pierre Colombo, Chloé Clavel, and Fabian M Suchanek. 2022. Of human criteria and automatic metrics: A benchmark of the evaluation of story generation. *arXiv preprint arXiv:2208.11646* (2022).

[35] Cheng-Han Chiang, Wei-Chih Chen, Chun-Yi Kuan, Chienchou Yang, and Hung-yi Lee. 2024. Large Language Model as an Assignment Evaluator: Insights, Feedback, and Challenges in a 1000+ Student Course. *arXiv preprint arXiv:2407.05216* (2024).

[36] Cheng-Han Chiang and Hung-yi Lee. 2023. A closer look into automatic evaluation using large language models. *arXiv preprint arXiv:2310.05657* (2023).

[37] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)* 2, 3 (2023), 6.

[38] Juhwan Choi, Jungmin Yun, Kyohoon Jin, and YoungBin Kim. 2024. Multi-News+: Cost-efficient Dataset Cleansing via LLM-based Data Annotation. *arXiv preprint arXiv:2404.09682* (2024).

[39] Zhumin Chu, Qingyao Ai, Yiteng Tu, Haitao Li, and Yiqun Liu. 2024. Pre: A peer review based large language model evaluator. *arXiv preprint arXiv:2401.15641* (2024).

[40] Hyung Won Chung, Le Hou, Shayne Longpre, Barret Zoph, Yi Tay, William Fedus, Yunxuan Li, Xuezhi Wang, Mostafa Dehghani, Siddhartha Brahma, et al. 2024. Scaling instruction-finetuned language models. *Journal of Machine Learning Research* 25, 70 (2024), 1–53.

[41] Israel Cohen, Yiteng Huang, Jingdong Chen, Jacob Benesty, Jacob Benesty, Jingdong Chen, Yiteng Huang, and Israel Cohen. 2009. Pearson correlation coefficient. *Noise reduction in speech processing* (2009), 1–4.

[42] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, and Jimmy Lin. 2021. Overview of the TREC 2021 Deep Learning Track. In *TREC*. https://trec.nist.gov/pubs/trec30/papers/Overview-DL.pdf

[43] Nick Craswell, Bhaskar Mitra, Emine Yilmaz, Daniel Campos, Jimmy Lin, Ellen M. Voorhees, and Ian Soboroff. 2022. Overview of the TREC 2022 Deep Learning Track. In *TREC*. https://trec.nist.gov/pubs/trec31/papers/Overview_deep.pdf

[44] Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Wei Zhu, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, et al. 2024. ULTRAFEEDBACK: Boosting Language Models with Scaled AI Feedback. In *Forty-first International Conference on Machine Learning*.

[45] Roland Daynauth and Jason Mars. 2024. Aligning Model Evaluations with Human Preferences: Mitigating Token Count Bias in Language Model Assessments. *arXiv preprint arXiv:2407.12847* (2024).

[46] Shijian Deng, Wentian Zhao, Yu-Jhe Li, Kun Wan, Daniel Miranda, Ajinkya Kale, and Yapeng Tian. 2024. Efficient Self-Improvement in Multimodal Large Language Models: A Model-Level Judge-Free Approach. *arXiv preprint arXiv:2411.17760* (2024).

[47] Mahesh Deshwal and Apoorva Chawla. 2024. PHUDGE: Phi-3 as Scalable Judge. *arXiv preprint arXiv:2405.08029* (2024).

[48] Laurence Dierickx, Arjen Van Dalen, Andreas L Opdahl, and Carl-Gustav Lindén. 2024. Striking the balance in using LLMs for fact-checking: A narrative literature review. In *Multidisciplinary International Symposium on Disinformation in Open Online Media*. Springer, 1–15.

[49] Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Zhi Zheng, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. *arXiv preprint arXiv:2305.14233* (2023).

[50] Yangruibo Ding, Zijian Wang, Wasi Ahmad, Hantian Ding, Ming Tan, Nihal Jain, Murali Krishna Ramanathan, Ramesh Nallapati, Parminder Bhatia, Dan Roth, et al. 2024. Crosscodeeval: A diverse and multilingual benchmark for cross-file code completion. *Advances in Neural Information Processing Systems* 36 (2024).

[51] Sumanth Doddapaneni, Mohammed Safi Ur Rahman Khan, Sshubam Verma, and Mitesh M Khapra. 2024. Finding Blind Spots in Evaluator LLMs with Interpretable Checklists. *arXiv preprint arXiv:2406.13439* (2024).

[52] Qingxiu Dong, Li Dong, Xingxing Zhang, Zhifang Sui, and Furu Wei. 2024. Self-Boosting Large Language Models with Synthetic Preference Data. *arXiv preprint arXiv:2410.06961* (2024).

[53] Qingxiu Dong, Lei Li, Damai Dai, Ce Zheng, Jingyuan Ma, Rui Li, Heming Xia, Jingjing Xu, Zhiyong Wu, Tianyu Liu, et al. 2022. A survey on in-context learning. *arXiv preprint arXiv:2301.00234* (2022).

[54] Yijiang River Dong, Tiancheng Hu, and Nigel Collier. 2024. Can LLM be a Personalized Judge? *arXiv preprint arXiv:2406.11657* (2024).

[55] Florian E. Dorner, Vivian Y. Nastl, and Moritz Hardt. 2024. Limits to scalable evaluation at the frontier: LLM as Judge won't beat twice the data. arXiv:2410.13341 [cs.LG] https://arxiv.org/abs/2410.13341

[56] Yann Dubois, Balázs Galambosi, Percy Liang, and Tatsunori B Hashimoto. 2024. Length-controlled alpacaeval: A simple way to debias automatic evaluators. *arXiv preprint arXiv:2404.04475* (2024).

[57] Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751* (2017).

[58] Aparna Elangovan, Jongwoo Ko, Lei Xu, Mahsa Elyasi, Ling Liu, Sravan Bodapati, and Dan Roth. 2024. Beyond correlation: The impact of human uncertainty in measuring the effectiveness of automatic evaluation and LLM-as-a-judge. *arXiv preprint arXiv:2410.03775* (2024).

[59] Alexander R Fabbri, Wojciech Kryściński, Bryan McCann, Caiming Xiong, Richard Socher, and Dragomir Radev. 2021. Summeval: Re-evaluating summarization evaluation. *Transactions of the Association for Computational Linguistics* 9 (2021), 391–409.

[60] Angela Fan, Mike Lewis, and Yann Dauphin. 2018. Hierarchical neural story generation. *arXiv preprint arXiv:1805.04833* (2018).

[61] Zhiting Fan, Ruizhe Chen, Ruiling Xu, and Zuozhu Liu. 2024. Biasalert: A plug-and-play tool for social bias detection in llms. *arXiv preprint arXiv:2407.10241* (2024).

[62] Yu Fei, Yifan Hou, Zeming Chen, and Antoine Bosselut. 2023. Mitigating label biases for in-context learning. *arXiv preprint arXiv:2305.19148* (2023).

[63] Chao Feng, Xinyu Zhang, and Zichu Fei. 2023. Knowledge solver: Teaching llms to search for domain knowledge from knowledge graphs. *arXiv preprint arXiv:2309.03118* (2023).

[64] Zhaopeng Feng, Yan Zhang, Hao Li, Wenqiang Liu, Jun Lang, Yang Feng, Jian Wu, and Zuozhu Liu. 2024. Improving llm-based machine translation with systematic self-correction. *arXiv preprint arXiv:2402.16379* (2024).

[65] Markus Freitag, George Foster, David Grangier, Viresh Ratnakar, Qijun Tan, and Wolfgang Macherey. 2021. Experts, errors, and context: A large-scale study of human evaluation for machine translation. *Transactions of the Association for Computational Linguistics* 9 (2021), 1460–1474.

[66] Markus Freitag, Ricardo Rei, Nitika Mathur, Chi-kiu Lo, Craig Stewart, George Foster, Alon Lavie, and Ondřej Bojar. 2021. Results of the WMT21 metrics shared task: Evaluating metrics with expert-based human evaluations on TED and news domain. In *Proceedings of the Sixth Conference on Machine Translation*. 733–774.

[67] Robert M French. 2000. The Turing Test: the first 50 years. *Trends in cognitive sciences* 4, 3 (2000), 115–122.

[68] Jinlan Fu, See-Kiong Ng, Zhengbao Jiang, and Pengfei Liu. 2023. Gptscore: Evaluate as you desire. *arXiv preprint arXiv:2302.04166* (2023).

[69] Yunfan Gao, Yun Xiong, Xinyu Gao, Kangxiang Jia, Jinliu Pan, Yuxi Bi, Yi Dai, Jiawei Sun, Meng Wang, and Haofen Wang. 2023. Retrieval-augmented generation for large language models: A survey. *arXiv preprint arXiv:2312.10997* (2023).

[70] Yicheng Gao, Gonghan Xu, Zhe Wang, and Arman Cohan. 2024. Bayesian Calibration of Win Rate Estimation with LLM Evaluators. *arXiv preprint arXiv:2411.04424* (2024).

[71] Fabrizio Gilardi, Meysam Alizadeh, and Maël Kubli. 2023. ChatGPT outperforms crowd workers for text-annotation tasks. *Proceedings of the National Academy of Sciences* 120, 30 (2023), e2305016120.

[72] Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tur. 2023. Topical-chat: Towards knowledge-grounded open-domain conversations. *arXiv preprint arXiv:2308.11995* (2023).

[73] Jian Guan, Zhexin Zhang, Zhuoer Feng, Zitao Liu, Wenbiao Ding, Xiaoxi Mao, Changjie Fan, and Minlie Huang. 2021. OpenMEVA: A benchmark for evaluating open-ended story generation metrics. *arXiv preprint arXiv:2105.08920* (2021).

[74] Shangmin Guo, Biao Zhang, Tianlin Liu, Tianqi Liu, Misha Khalman, Felipe Llinares, Alexandre Rame, Thomas Mesnard, Yao Zhao, Bilal Piot, et al. 2024. Direct language model alignment from online ai feedback. *arXiv preprint arXiv:2402.04792* (2024).

[75] Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiaxuan Li, Bojian Xiong, Deyi Xiong, et al. 2023. Evaluating large language models: A comprehensive survey. *arXiv preprint arXiv:2310.19736* (2023).

[76] Taneesh Gupta, Shivam Shandilya, Xuchao Zhang, Supriyo Ghosh, Chetan Bansal, Huaxiu Yao, and Saravan Rajmohan. 2024. Unveiling Context-Aware Criteria in Self-Assessing LLMs. *arXiv preprint arXiv:2410.21545* (2024).

[77] Rishav Hada, Varun Gumma, Adrian de Wynter, Harshita Diddee, Mohamed Ahmed, Monojit Choudhury, Kalika Bali, and Sunayana Sitaram. 2023. Are large language model-based evaluators the solution to scaling up multilingual evaluation? *arXiv preprint arXiv:2309.07462* (2023).

[78] Zhixiong Han, Yaru Hao, Li Dong, Yutao Sun, and Furu Wei. 2022. Prototypical calibration for few-shot learning of language models. *arXiv preprint arXiv:2205.10183* (2022).

[79] Jing Hao, Yuxiang Zhao, Song Chen, Yanpeng Sun, Qiang Chen, Gang Zhang, Kun Yao, Errui Ding, and Jingdong Wang. 2024. Fullanno: A data engine for enhancing image comprehension of mllms. *arXiv preprint arXiv:2409.13540* (2024).

[80] F Maxwell Harper and Joseph A Konstan. 2015. The movielens datasets: History and context. *Acm transactions on interactive intelligent systems (tiis)* 5, 4 (2015), 1–19.

[81] Hosein Hasanbeig, Hiteshi Sharma, Leo Betthauser, Felipe Vieira Frujeri, and Ida Momennejad. 2023. ALLURE: auditing and improving llm-based evaluation of text using iterative in-context-learning. *arXiv e-prints* (2023), arXiv–2309.

[82] Hangfeng He, Hongming Zhang, and Dan Roth. 2023. Socreval: Large language models with the socratic method for reference-free reasoning evaluation. *arXiv preprint arXiv:2310.00074* (2023).

[83] Xingwei He, Zhenghao Lin, Yeyun Gong, Alex Jin, Hang Zhang, Chen Lin, Jian Jiao, Siu Ming Yiu, Nan Duan, Weizhu Chen, et al. 2023. Annollm: Making large language models to be better crowdsourced annotators. *arXiv preprint arXiv:2303.16854* (2023).

[84] Yuanqin He, Yan Kang, Lixin Fan, and Qiang Yang. 2024. FedEval-LLM: Federated Evaluation of Large Language Models on Downstream Tasks with Collective Wisdom. *arXiv preprint arXiv:2404.12273* (2024).

[85] Zeyu He, Chieh-Yang Huang, Chien-Kuang Cornelia Ding, Shaurya Rohatgi, and Ting-Hao Kenneth Huang. 2024. If in a Crowdsourced Data Annotation Pipeline, a GPT-4. In *Proceedings of the CHI Conference on Human Factors in Computing Systems*. 1–25.

[86] Hashem Hijazi, Diego Molla, Vincent Nguyen, and Sarvnaz Karimi. 2024. Using Large Language Models to Evaluate Biomedical Query-Focused Summarisation. In *Proceedings of the 23rd Workshop on Biomedical Natural Language Processing*. 236–242.

[87] Yupeng Hou, Junjie Zhang, Zihan Lin, Hongyu Lu, Ruobing Xie, Julian McAuley, and Wayne Xin Zhao. 2024. Large language models are zero-shot rankers for recommender systems. In *European Conference on Information Retrieval*. Springer, 364–381.

[88] Xinyu Hu, Li Lin, Mingqi Gao, Xunjian Yin, and Xiaojun Wan. 2024. Themis: A reference-free nlg evaluation language model with flexibility and interpretability. *arXiv preprint arXiv:2406.18365* (2024).

[89] Zhengyu Hu, Linxin Song, Jieyu Zhang, Zheyuan Xiao, Jingang Wang, Zhenyu Chen, and Hui Xiong. 2024. Rethinking llm-based preference evaluation. *arXiv preprint arXiv:2407.01085* (2024).

[90] Zhengyu Hu, Jieyu Zhang, Zhihan Xiong, Alexander Ratner, Hui Xiong, and Ranjay Krishna. 2024. Language Model Preference Evaluation with Multiple Weak Evaluators. *arXiv preprint arXiv:2410.12869* (2024).

[91] Hui Huang, Yingqi Qu, Jing Liu, Muyun Yang, and Tiejun Zhao. 2024. An empirical study of llm-as-a-judge for llm evaluation: Fine-tuned judge models are task-specific classifiers. *arXiv preprint arXiv:2403.02839* (2024).

[92] Hui Huang, Yingqi Qu, Hongli Zhou, Jing Liu, Muyun Yang, Bing Xu, and Tiejun Zhao. 2024. On the limitations of fine-tuned judge models for llm evaluation. *arXiv preprint arXiv:2403.02839* (2024).

[93] Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *arXiv preprint arXiv:2310.01798* (2023).

[94] Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023. Multi-dimensional evaluation of text summarization with in-context learning. *arXiv preprint arXiv:2306.01200* (2023).

[95] Minbyul Jeong, Jiwoong Sohn, Mujeen Sung, and Jaewoo Kang. 2024. Improving medical reasoning through retrieval and self-reflection with retrieval-augmented large language models. *Bioinformatics* 40, Supplement_1 (2024), i119–i129.

[96] Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. 2024. PKU-SafeRLHF: Towards Multi-Level Safety Alignment for LLMs with Human Preference. *arXiv preprint arXiv:2406.15513* (2024).

[97] Ziwei Ji, Nayeon Lee, Rita Frieske, Tiezheng Yu, Dan Su, Yan Xu, Etsuko Ishii, Ye Jin Bang, Andrea Madotto, and Pascale Fung. 2023. Survey of hallucination in natural language generation. *Comput. Surveys* 55, 12 (2023), 1–38.

[98] Bowen Jiang, Yangxinyu Xie, Zhuoqun Hao, Xiaomeng Wang, Tanwi Mallick, Weijie J Su, Camillo J Taylor, and Dan Roth. 2024. A Peek into Token Bias: Large Language Models Are Not Yet Genuine Reasoners. *arXiv preprint arXiv:2406.11050* (2024).

[99] Dongfu Jiang, Yishan Li, Ge Zhang, Wenhao Huang, Bill Yuchen Lin, and Wenhu Chen. 2023. Tigerscore: Towards building explainable metric for all text generation tasks. *Transactions on Machine Learning Research* (2023).

[100] Shuyu Jiang, Xingshu Chen, and Rui Tang. 2023. Prompt packer: Deceiving llms through compositional instruction with hidden attacks. *arXiv preprint arXiv:2310.10077* (2023).

[101] Carlos E Jimenez, John Yang, Alexander Wettig, Shunyu Yao, Kexin Pei, Ofir Press, and Karthik Narasimhan. 2023. Swe-bench: Can language models resolve real-world github issues?, 2024. *URL https://arxiv. org/abs/2310.06770* (2023).

[102] Yuu Jinnai, Tetsuro Morimura, Kaito Ariu, and Kenshi Abe. 2024. Regularized Best-of-N Sampling to Mitigate Reward Hacking for Language Model Alignment. *arXiv preprint arXiv:2404.01054* (2024).

[103] Jaehun Jung, Faeze Brahman, and Yejin Choi. 2024. Trust or Escalate: LLM Judges with Provable Guarantees for Human Agreement. *arXiv preprint arXiv:2407.18370* (2024).

[104] Marzena Karpinska and Mohit Iyyer. 2023. Large language models effectively leverage document-level context for literary translation, but critical errors persist. *arXiv preprint arXiv:2304.03245* (2023).

[105] Akira Kawabata and Saku Sugawara. 2024. Rationale-Aware Answer Verification by Pairwise Self-Evaluation. *arXiv preprint arXiv:2410.04838* (2024).

[106] Pei Ke, Bosi Wen, Andrew Feng, Xiao Liu, Xuanyu Lei, Jiale Cheng, Shengyuan Wang, Aohan Zeng, Yuxiao Dong, Hongning Wang, et al. 2024. Critiquellm: Towards an informative critique generation model for evaluation of large language model generation. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 13034–13054.

[107] Akbir Khan, John Hughes, Dan Valentine, Laura Ruis, Kshitij Sachan, Ansh Radhakrishnan, Edward Grefenstette, Samuel R Bowman, Tim Rocktäschel, and Ethan Perez. 2024. Debating with more persuasive llms leads to more truthful answers. *arXiv preprint arXiv:2402.06782* (2024).

[108] Dongyoung Kim, Kimin Lee, Jinwoo Shin, and Jaehyung Kim. 2024. Aligning Large Language Models with Self-generated Preference Data. *arXiv preprint arXiv:2406.04412* (2024).

[109] Seungone Kim, Jamin Shin, Yejin Cho, Joel Jang, Shayne Longpre, Hwaran Lee, Sangdoo Yun, Seongjin Shin, Sungdong Kim, James Thorne, et al. 2023. Prometheus: Inducing fine-grained evaluation capability in language models. In *The Twelfth International Conference on Learning Representations*.

[110] Seungone Kim, Juyoung Suk, Shayne Longpre, Bill Yuchen Lin, Jamin Shin, Sean Welleck, Graham Neubig, Moontae Lee, Kyungjae Lee, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. *arXiv preprint arXiv:2405.01535* (2024).

[111] Miyoung Ko, Jinhyuk Lee, Hyunjae Kim, Gangwoo Kim, and Jaewoo Kang. 2020. Look at the first sentence: Position bias in question answering. *arXiv preprint arXiv:2004.14602* (2020).

[112] Ryan Koo, Minhwa Lee, Vipul Raheja, Jong Inn Park, Zae Myung Kim, and Dongyeop Kang. 2023. Benchmarking cognitive biases in large language models as evaluators. *arXiv preprint arXiv:2309.17012* (2023).

[113] Neema Kotonya, Saran Krishnasamy, Joel Tetreault, and Alejandro Jaimes. 2023. Little giants: Exploring the potential of small llms as evaluation metrics in summarization in the eval4nlp 2023 shared task. *arXiv preprint arXiv:2311.00686* (2023).

[114] Jack Krolik, Herprit Mahal, Feroz Ahmad, Gaurav Trivedi, and Bahador Saket. 2024. Towards Leveraging Large Language Models for Automated Medical Q&A Evaluation. *arXiv preprint arXiv:2409.01941* (2024).

[115] Abhishek Kumar, Sonia Haiduc, Partha Pratim Das, and Partha Pratim Chakrabarti. 2024. LLMs as Evaluators: A Novel Approach to Evaluate Bug Report Summarization. *arXiv preprint arXiv:2409.00630* (2024).

[116] Nathan Lambert, Valentina Pyatkin, Jacob Morrison, LJ Miranda, Bill Yuchen Lin, Khyathi Chandu, Nouha Dziri, Sachin Kumar, Tom Zick, Yejin Choi, et al. 2024. Rewardbench: Evaluating reward models for language modeling. *arXiv preprint arXiv:2403.13787* (2024).

[117] Siddique Latif, Muhammad Usama, Mohammad Ibrahim Malik, and Björn W Schuller. 2023. Can large language models aid in annotating speech emotional data? uncovering new frontiers. *arXiv preprint arXiv:2307.06090* (2023).

[118] Dawn Lawrie, Sean MacAvaney, James Mayfield, Paul McNamee, Douglas W Oard, Luca Soldaini, and Eugene Yang. 2024. Overview of the TREC 2023 NeuCLIR Track. *arXiv preprint arXiv:2404.08071* (2024).

[119] Yann LeCun, Yoshua Bengio, and Geoffrey Hinton. 2015. Deep learning. *nature* 521, 7553 (2015), 436–444.

[120] Deokjae Lee, Seungyong Moon, Junhyeok Lee, and Hyun Oh Song. 2022. Query-efficient and scalable black-box adversarial attacks on discrete sequential data via bayesian optimization. In *International Conference on Machine Learning*. PMLR, 12478–12497.

[121] Harrison Lee, Samrat Phatale, Hassan Mansoor, Kellie Ren Lu, Thomas Mesnard, Johan Ferret, Colton Bishop, Ethan Hall, Victor Carbune, and Abhinav Rastogi. 2023. Rlaif: Scaling reinforcement learning from human feedback with ai feedback. (2023).

[122] Seongyun Lee, Seungone Kim, Sue Hyun Park, Geewook Kim, and Minjoon Seo. 2024. Prometheusvision: Vision-language model as a judge for fine-grained evaluation. *arXiv preprint arXiv:2401.06591* (2024).

[123] Sangkyu Lee, Sungdong Kim, Ashkan Yousefpour, Minjoon Seo, Kang Min Yoo, and Youngjae Yu. 2024. Aligning Large Language Models by On-Policy Self-Judgment. *arXiv preprint arXiv:2402.11253* (2024).

[124] Yuxuan Lei, Jianxun Lian, Jing Yao, Xu Huang, Defu Lian, and Xing Xie. 2024. RecExplainer: Aligning Large Language Models for Explaining Recommendation Models. In *Proceedings of the 30th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*. 1530–1541.

[125] Patrick Lewis, Ethan Perez, Aleksandra Piktus, Fabio Petroni, Vladimir Karpukhin, Naman Goyal, Heinrich Küttler, Mike Lewis, Wen-tau Yih, Tim Rocktäschel, et al. 2020. Retrieval-augmented generation for knowledge-intensive nlp tasks. *Advances in Neural Information Processing Systems* 33 (2020), 9459–9474.

[126] Anqi Li, Yu Lu, Nirui Song, Shuai Zhang, Lizhi Ma, and Zhenzhong Lan. 2024. Automatic evaluation for mental health counseling using llms. *arXiv preprint arXiv:2402.11958* (2024).

[127] Haitao Li, Junjie Chen, Qingyao Ai, Zhumin Chu, Yujia Zhou, Qian Dong, and Yiqun Liu. 2024. Calibraeval: Calibrating prediction distribution to mitigate selection bias in llms-as-judges. *arXiv preprint arXiv:2410.15393* (2024).

[128] Haitao Li, You Chen, Qingyao Ai, Yueyue Wu, Ruizhe Zhang, and Yiqun Liu. 2024. LexEval: A Comprehensive Chinese Legal Benchmark for Evaluating Large Language Models. arXiv:2409.20288 [cs.CL] https://arxiv.org/abs/2409.20288

[129] Haitao Li, Yunqiu Shao, Yueyue Wu, Qingyao Ai, Yixiao Ma, and Yiqun Liu. 2024. Lecardv2: A large-scale chinese legal case retrieval dataset. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval*. 2251–2260.

[130] Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2023. Generative judge for evaluating alignment. *arXiv preprint arXiv:2310.05470* (2023).

[131] Qintong Li, Leyang Cui, Lingpeng Kong, and Wei Bi. 2023. Collaborative Evaluation: Exploring the Synergy of Large Language Models and Humans for Open-ended Generation Evaluation. *arXiv preprint arXiv:2310.19740* (2023).

[132] Ruosen Li, Teerth Patel, and Xinya Du. 2023. Prd: Peer rank and discussion improve large language model based evaluations. *arXiv preprint arXiv:2307.02762* (2023).

[133] Yanran Li, Hui Su, Xiaoyu Shen, Wenjie Li, Ziqiang Cao, and Shuzi Niu. 2017. DailyDialog: A Manually Labelled Multi-turn Dialogue Dataset. In *Proceedings of The 8th International Joint Conference on Natural Language Processing (IJCNLP 2017)*.

[134] Zongjie Li, Chaozheng Wang, Pingchuan Ma, Daoyuan Wu, Shuai Wang, Cuiyun Gao, and Yang Liu. 2023. Split and merge: Aligning position biases in large language model based evaluators. *arXiv preprint arXiv:2310.01432* (2023).

[135] Jingcong Liang, Rong Ye, Meng Han, Ruofei Lai, Xinyu Zhang, Xuanjing Huang, and Zhongyu Wei. 2024. Debatrix: Multi-dimensinal Debate Judge with Iterative Chronological Analysis Based on LLM. *arXiv preprint arXiv:2403.08010* (2024).

[136] Sirui Liang, Baoli Zhang, Jun Zhao, and Kang Liu. 2024. ABSEval: An Agent-based Framework for Script Evaluation. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*. 12418–12434.

[137] Hunter Lightman, Vineet Kosaraju, Yura Burda, Harri Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2023. Let's verify step by step. *arXiv preprint arXiv:2305.20050* (2023).

[138] Bill Yuchen Lin, Yuntian Deng, Khyathi Chandu, Faeze Brahman, Abhilasha Ravichander, Valentina Pyatkin, Nouha Dziri, Ronan Le Bras, and Yejin Choi. 2024. WILDBENCH: Benchmarking LLMs with Challenging Tasks from Real Users in the Wild. *arXiv preprint arXiv:2406.04770* (2024).

[139] Chin-Yew Lin. 2004. Rouge: A package for automatic evaluation of summaries. In *Text summarization branches out*. 74–81.

[140] Stephanie Lin, Jacob Hilton, and Owain Evans. 2021. Truthfulqa: Measuring how models mimic human falsehoods. *arXiv preprint arXiv:2109.07958* (2021).

[141] Yen-Ting Lin and Yun-Nung Chen. 2023. Llm-eval: Unified multi-dimensional automatic evaluation for open-domain conversations with large language models. *arXiv preprint arXiv:2305.13711* (2023).

[142] Minqian Liu, Ying Shen, Zhiyang Xu, Yixin Cao, Eunah Cho, Vaibhav Kumar, Reza Ghanadan, and Lifu Huang. 2023. X-eval: Generalizable multi-aspect text evaluation via augmented instruction tuning with auxiliary evaluation aspects.

arXiv preprint arXiv:2311.08788 (2023).

[143] Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Zhuoer Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, et al. 2023. Alignbench: Benchmarking chinese alignment of large language models. *arXiv preprint arXiv:2311.18743* (2023).

[144] Xiao Liu, Hao Yu, Hanchen Zhang, Yifan Xu, Xuanyu Lei, Hanyu Lai, Yu Gu, Hangliang Ding, Kaiwen Men, Kejuan Yang, et al. 2023. Agentbench: Evaluating llms as agents. *arXiv preprint arXiv:2308.03688* (2023).

[145] Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023. G-eval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634* (2023).

[146] Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2023. Calibrating llm-based evaluator. *arXiv preprint arXiv:2309.13308* (2023).

[147] Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024. HD-Eval: Aligning Large Language Model Evaluators Through Hierarchical Criteria Decomposition. *arXiv preprint arXiv:2402.15754* (2024).

[148] Yantao Liu, Zijun Yao, Rui Min, Yixin Cao, Lei Hou, and Juanzi Li. 2024. RM-Bench: Benchmarking Reward Models of Language Models with Subtlety and Style. *arXiv preprint arXiv:2410.16184* (2024).

[149] Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulić, Anna Korhonen, and Nigel Collier. 2024. Aligning with human judgement: The role of pairwise preference in large language model evaluators. *arXiv preprint arXiv:2403.16950* (2024).

[150] Adian Liusie, Vatsal Raina, Yassir Fathullah, and Mark Gales. 2024. Efficient LLM Comparative Assessment: a Product of Experts Framework for Pairwise Comparisons. *arXiv preprint arXiv:2405.05894* (2024).

[151] SESSMENTS BY LLMS. 2025. JUDING THE JUDGES: ASYSTEMATIC INVESTIGATION OF POSITION BIAS IN PAIRWISE COMPARATIVE AS. *Under review as a conference paper at ICLR 2025* (2025).

[152] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. 2023. An empirical study of catastrophic forgetting in large language models during continual fine-tuning. *arXiv preprint arXiv:2308.08747* (2023).

[153] Ziyang Luo, Haoning Wu, Dongxu Li, Jing Ma, Mohan Kankanhalli, and Junnan Li. 2024. VideoAutoArena: An Automated Arena for Evaluating Large Multimodal Models in Video Analysis through User Simulation. *arXiv preprint arXiv:2411.13281* (2024).

[154] Shengjie Ma, Chong Chen, Qi Chu, and Jiaxin Mao. 2024. Leveraging Large Language Models for Relevance Judgments in Legal Case Retrieval. *arXiv preprint arXiv:2403.18405* (2024).

[155] Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. 2024. Self-refine: Iterative refinement with self-feedback. *Advances in Neural Information Processing Systems* 36 (2024).

[156] Shikib Mehri and Maxine Eskenazi. 2020. USR: An unsupervised and reference free evaluation metric for dialog generation. *arXiv preprint arXiv:2005.00456* (2020).

[157] John Mendonça, Isabel Trancoso, and Alon Lavie. 2024. Soda-Eval: Open-Domain Dialogue Evaluation in the age of LLMs. *arXiv preprint arXiv:2408.10902* (2024).

[158] Behrad Moniri, Hamed Hassani, and Edgar Dobriban. 2024. Evaluating the Performance of Large Language Models via Debates. *arXiv preprint arXiv:2406.11044* (2024).

[159] Nasrin Mostafazadeh, Nathanael Chambers, Xiaodong He, Devi Parikh, Dhruv Batra, Lucy Vanderwende, Pushmeet Kohli, and James Allen. 2016. A corpus and cloze evaluation for deeper understanding of commonsense stories. In *Proceedings of the 2016 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*. 839–849.

[160] Mirco Musolesi. 2024. Creative Beam Search: LLM-as-a-Judge for Improving Response Generation. ICCC.

[161] Aidar Myrzakhan, Sondos Mahmoud Bsharat, and Zhiqiang Shen. 2024. Open-LLM-Leaderboard: From Multi-choice to Open-style Questions for LLMs Evaluation, Benchmark, and Arena. *arXiv preprint arXiv:2406.07545* (2024).

[162] Shashi Narayan, Shay B Cohen, and Mirella Lapata. 2018. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. *arXiv preprint arXiv:1808.08745* (2018).

[163] Dom Nasrabadi. 2024. JurEE not Judges: safeguarding llm interactions with small, specialised Encoder Ensembles. *arXiv preprint arXiv:2410.08442* (2024).

[164] Nils J Nilsson. 2014. *Principles of artificial intelligence.* Morgan Kaufmann.

[165] Kun-Peng Ning, Shuo Yang, Yuyang Liu, Jia-Yu Yao, Zhenhui Liu, Yu Wang, Ming Pang, and Li Yuan. 2024. PiCO: Peer Review in LLMs based on the Consistency Optimization. (2024).

[166] Tong Niu, Shafiq Joty, Ye Liu, Caiming Xiong, Yingbo Zhou, and Semih Yavuz. 2024. JudgeRank: Leveraging Large Language Models for Reasoning-Intensive Reranking. *arXiv preprint arXiv:2411.00142* (2024).

[167] Odhran O'Donoghue, Aleksandar Shtedritski, John Ginger, Ralph Abboud, Ali Essa Ghareeb, Justin Booth, and Samuel G Rodriques. 2023. BioPlanner: automatic evaluation of LLMs on protocol planning in biology. *arXiv preprint arXiv:2310.10632* (2023).

[168] Deonna M Owens, Ryan A Rossi, Sungchul Kim, Tong Yu, Franck Dernoncourt, Xiang Chen, Ruiyi Zhang, Jiuxiang Gu, Hanieh Deilamsalehy, and Nedim Lipka. 2024. A multi-llm debiasing framework. *arXiv preprint arXiv:2409.13884* (2024).

[169] Artidoro Pagnoni, Vidhisha Balachandran, and Yulia Tsvetkov. 2021. Understanding factuality in abstractive summarization with FRANK: A benchmark for factuality metrics. *arXiv preprint arXiv:2104.13346* (2021).

[170] Qian Pan, Zahra Ashktorab, Michael Desmond, Martin Santillan Cooper, James Johnson, Rahul Nair, Elizabeth Daly, and Werner Geyer. 2024. Human-Centered Design Recommendations for LLM-as-a-judge. *arXiv preprint arXiv:2407.03479* (2024).

[171] Shirui Pan, Linhao Luo, Yufei Wang, Chen Chen, Jiapu Wang, and Xindong Wu. 2024. Unifying large language models and knowledge graphs: A roadmap. *IEEE Transactions on Knowledge and Data Engineering* (2024).

[172] Arjun Panickssery, Samuel R Bowman, and Shi Feng. 2024. Llm evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076* (2024).

[173] Kishore Papineni, Salim Roukos, Todd Ward, and Wei-Jing Zhu. 2002. Bleu: a method for automatic evaluation of machine translation. In *Proceedings of the 40th annual meeting of the Association for Computational Linguistics.* 311–318.

[174] Junsoo Park, Seungyeon Jwa, Meiying Ren, Daeyoung Kim, and Sanghyuk Choi. 2024. Offsetbias: Leveraging debiased data for tuning evaluators. *arXiv preprint arXiv:2407.06551* (2024).

[175] Bhrij Patel, Souradip Chakraborty, Wesley A Suttle, Mengdi Wang, Amrit Singh Bedi, and Dinesh Manocha. 2024. AIME: AI System Optimization via Multiple LLM Evaluators. *arXiv preprint arXiv:2410.03131* (2024).

[176] Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2023. Refiner: Reasoning feedback on intermediate representations. *arXiv preprint arXiv:2304.01904* (2023).

[177] Fábio Perez and Ian Ribeiro. 2022. Ignore previous prompt: Attack techniques for language models. *arXiv preprint arXiv:2211.09527* (2022).

[178] Pouya Pezeshkpour and Estevam Hruschka. 2023. Large language models sensitivity to the order of options in multiple-choice questions. *arXiv preprint arXiv:2308.11483* (2023).

[179] Raphael Poulain, Hamed Fayyaz, and Rahmatollah Beheshti. 2024. Bias patterns in the application of LLMs for clinical decision support: A comprehensive study. *arXiv preprint arXiv:2404.15149* (2024).

[180] Zhen Qin, Rolf Jagerman, Kai Hui, Honglei Zhuang, Junru Wu, Le Yan, Jiaming Shen, Tianqi Liu, Jialu Liu, Donald Metzler, et al. 2023. Large language models are effective text rankers with pairwise ranking prompting. *arXiv preprint arXiv:2306.17563* (2023).

[181] Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems* 36 (2024).

[182] Priya Raghubir and Ana Valenzuela. 2006. Center-of-inattention: Position biases in decision-making. *Organizational Behavior and Human Decision Processes* 99, 1 (2006), 66–80.

[183] Hossein A Rahmani, Emine Yilmaz, Nick Craswell, Bhaskar Mitra, Paul Thomas, Charles LA Clarke, Mohammad Aliannejadi, Clemencia Siro, and Guglielmo Faggioli. 2024. LLMJudge: LLMs for Relevance Judgments. *arXiv preprint arXiv:2408.08896* (2024).

[184] Vyas Raina, Adian Liusie, and Mark Gales. 2024. Is LLM-as-a-Judge Robust? Investigating Universal Adversarial Attacks on Zero-shot LLM Assessment. *arXiv preprint arXiv:2402.14016* (2024).

[185] Ravi Raju, Swayambhoo Jain, Bo Li, Jonathan Li, and Urmish Thakker. 2024. Constructing domain-specific evaluation sets for llm-as-a-judge. *arXiv preprint arXiv:2408.08808* (2024).

[186] Jie Ren, Yao Zhao, Tu Vu, Peter J Liu, and Balaji Lakshminarayanan. 2023. Self-evaluation improves selective generation in large language models. In *Proceedings on.* PMLR, 49–64.

[187] Cheol Ryu, Seolhwa Lee, Subeen Pang, Chanyeol Choi, Hojun Choi, Myeonggee Min, and Jy-Yong Sohn. 2023. Retrieval-based Evaluation for LLMs: A Case Study in Korean Legal QA. In *Proceedings of the Natural Legal Language Processing Workshop 2023.* 132–137.

[188] Jon Saad-Falcon, Omar Khattab, Christopher Potts, and Matei Zaharia. 2023. Ares: An automated evaluation framework for retrieval-augmented generation systems. *arXiv preprint arXiv:2311.09476* (2023).

[189] Pranab Sahoo, Ayush Kumar Singh, Sriparna Saha, Vinija Jain, Samrat Mondal, and Aman Chadha. 2024. A systematic survey of prompt engineering in large language models: Techniques and applications. *arXiv preprint arXiv:2402.07927* (2024).

[190] Philip Sedgwick. 2014. Spearman's rank correlation coefficient. *Bmj* 349 (2014).

[191] Pranab Kumar Sen. 1968. Estimates of the regression coefficient based on Kendall's tau. *Journal of the American statistical association* 63, 324 (1968), 1379–1389.

[192] Shreya Shankar, JD Zamfirescu-Pereira, Björn Hartmann, Aditya Parameswaran, and Ian Arawjo. 2024. Who validates the validators? aligning llm-assisted evaluation of llm outputs with human preferences. In *Proceedings of the 37th*

Annual ACM Symposium on User Interface Software and Technology. 1–14.

[193] Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2023. " do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *arXiv preprint arXiv:2308.03825* (2023).

[194] Yuchen Shen and Xiaojun Wan. 2023. Opinsummeval: Revisiting automated evaluation for opinion summarization. *arXiv preprint arXiv:2310.18122* (2023).

[195] Freda Shi, Xinyun Chen, Kanishka Misra, Nathan Scales, David Dohan, Ed H Chi, Nathanael Schärli, and Denny Zhou. 2023. Large language models can be easily distracted by irrelevant context. In *International Conference on Machine Learning*. PMLR, 31210–31227.

[196] Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. 2024. Optimization-based Prompt Injection Attack to LLM-as-a-Judge. *arXiv preprint arXiv:2403.17710* (2024).

[197] Lin Shi, Chiyu Ma, Wenhua Liang, Weicheng Ma, and Soroush Vosoughi. 2024. Judging the judges: A systematic investigation of position bias in pairwise comparative assessments by llms. *arXiv preprint arXiv:2406.07791* (2024).

[198] Lei Shu, Nevan Wichers, Liangchen Luo, Yun Zhu, Yinxiao Liu, Jindong Chen, and Lei Meng. 2024. Fusion-Eval: Integrating Assistant Evaluators with LLMs. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*. 225–238.

[199] Avi Singh, John D Co-Reyes, Rishabh Agarwal, Ankesh Anand, Piyush Patil, Xavier Garcia, Peter J Liu, James Harrison, Jaehoon Lee, Kelvin Xu, et al. 2023. Beyond human data: Scaling self-training for problem-solving with language models. *arXiv preprint arXiv:2312.06585* (2023).

[200] Ian Soboroff. 2024. Don't Use LLMs to Make Relevance Judgments. *arXiv preprint arXiv:2409.15133* (2024).

[201] Guijin Son, Hyunjun Jeon, Chami Hwang, and Hanearl Jung. 2024. KRX Bench: Automating Financial Benchmark Creation via Large Language Models. In *Proceedings of the Joint Workshop of the 7th Financial Technology and Natural Language Processing, the 5th Knowledge Discovery from Unstructured Data in Financial Services, and the 4th Workshop on Economics and Natural Language Processing@ LREC-COLING 2024*. 10–20.

[202] Guijin Son, Dongkeun Yoon, Juyoung Suk, Javier Aula-Blasco, Mano Aslan, Vu Trong Kim, Shayekh Bin Islam, Jaume Prats-Cristià, Lucía Tormo-Bañuelos, and Seungone Kim. 2024. MM-Eval: A Multilingual Meta-Evaluation Benchmark for LLM-as-a-Judge and Reward Models. *arXiv preprint arXiv:2410.17578* (2024).

[203] Hwanjun Song, Hang Su, Igor Shalyminov, Jason Cai, and Saab Mansour. 2024. FineSurE: Fine-grained summarization evaluation using LLMs. *arXiv preprint arXiv:2407.00908* (2024).

[204] Mingyang Song, Mao Zheng, and Xuan Luo. 2024. Can Many-Shot In-Context Learning Help Long-Context LLM Judges? See More, Judge Better! *arXiv preprint arXiv:2406.11629* (2024).

[205] Yishen Song, Qianta Zhu, Huaibo Wang, and Qinhua Zheng. 2024. Automated Essay Scoring and Revising Based on Open-Source Large Language Models. *IEEE Transactions on Learning Technologies* (2024).

[206] Andrea Sottana, Bin Liang, Kai Zou, and Zheng Yuan. 2023. Evaluation metrics in the era of GPT-4: reliably evaluating large language models on sequence to sequence tasks. *arXiv preprint arXiv:2310.13800* (2023).

[207] Andreas Stephan, Dawei Zhu, Matthias Aßenmacher, Xiaoyu Shen, and Benjamin Roth. 2024. From calculation to adjudication: Examining llm judges on mathematical reasoning tasks. *arXiv preprint arXiv:2409.04168* (2024).

[208] Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *arXiv preprint arXiv:2405.01724* (2024).

[209] Hanshi Sun, Momin Haider, Ruiqi Zhang, Huitao Yang, Jiahao Qiu, Ming Yin, Mengdi Wang, Peter Bartlett, and Andrea Zanette. 2024. Fast Best-of-N Decoding via Speculative Rejection. *arXiv preprint arXiv:2410.20290* (2024).

[210] Lichao Sun. 2020. Natural backdoor attack on text data. *arXiv preprint arXiv:2006.16176* (2020).

[211] Lichao Sun, Kazuma Hashimoto, Wenpeng Yin, Akari Asai, Jia Li, Philip Yu, and Caiming Xiong. 2020. Adv-bert: Bert is not robust on misspellings! generating nature adversarial samples on bert. *arXiv preprint arXiv:2003.04985* (2020).

[212] Annalisa Szymanski, Noah Ziems, Heather A Eicher-Miller, Toby Jia-Jun Li, Meng Jiang, and Ronald A Metoyer. 2024. Limitations of the LLM-as-a-Judge Approach for Evaluating LLM Outputs in Expert Knowledge Tasks. *arXiv preprint arXiv:2410.20266* (2024).

[213] Sijun Tan, Siyuan Zhuang, Kyle Montgomery, William Y Tang, Alejandro Cuadron, Chenguang Wang, Raluca Ada Popa, and Ion Stoica. 2024. JudgeBench: A Benchmark for Evaluating LLM-based Judges. *arXiv preprint arXiv:2410.12784* (2024).

[214] Michael Henry Tessler, Michiel A Bakker, Daniel Jarrett, Hannah Sheahan, Martin J Chadwick, Raphael Koster, Georgina Evans, Lucy Campbell-Gillingham, Tantum Collins, David C Parkes, et al. 2024. AI can help humans find common ground in democratic deliberation. *Science* 386, 6719 (2024), eadq2852.

[215] Aman Singh Thakur, Kartik Choudhary, Venkat Srinik Ramayapally, Sankaran Vaidyanathan, and Dieuwke Hupkes. 2024. Judging the Judges: Evaluating Alignment and Vulnerabilities in LLMs-as-Judges. *arXiv preprint arXiv:2406.12624* (2024).

[216] SM Tonmoy, SM Zaman, Vinija Jain, Anku Rani, Vipula Rawte, Aman Chadha, and Amitava Das. 2024. A comprehensive survey of hallucination mitigation techniques in large language models. *arXiv preprint arXiv:2401.01313*

(2024).

[217] Petter Törnberg. 2023. Chatgpt-4 outperforms experts and crowd workers in annotating political twitter messages with zero-shot learning. *arXiv preprint arXiv:2304.06588* (2023).

[218] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288* (2023).

[219] Harsh Trivedi, Tushar Khot, Mareike Hartmann, Ruskin Manku, Vinty Dong, Edward Li, Shashank Gupta, Ashish Sabharwal, and Niranjan Balasubramanian. 2024. Appworld: A controllable world of apps and people for benchmarking interactive coding agents. *arXiv preprint arXiv:2407.18901* (2024).

[220] Prapti Trivedi, Aditya Gulati, Oliver Molenschot, Meghana Arakkal Rajeev, Rajkumar Ramamurthy, Keith Stevens, Tanveesh Singh Chaudhery, Jahnavi Jambholkar, James Zou, and Nazneen Rajani. 2024. Self-rationalization improves LLM as a fine-grained judge. *arXiv preprint arXiv:2410.05495* (2024).

[221] Yu-Min Tseng, Wei-Lin Chen, Chung-Chi Chen, and Hsin-Hsi Chen. 2024. Are Expert-Level Language Models Expert-Level Annotators? *arXiv preprint arXiv:2410.03254* (2024).

[222] Alan M Turing. 2009. *Computing machinery and intelligence.* Springer.

[223] Gladys Tyen, Hassan Mansoor, Peter Chen, Tony Mak, and Victor Cărbune. 2023. LLMs cannot find reasoning errors, but can correct them! *arXiv preprint arXiv:2311.08516* (2023).

[224] Karthik Valmeekam, Matthew Marquez, and Subbarao Kambhampati. 2023. Can large language models really improve by self-critiquing their own plans? *arXiv preprint arXiv:2310.08118* (2023).

[225] Pat Verga, Sebastian Hofstatter, Sophia Althammer, Yixuan Su, Aleksandra Piktus, Arkady Arkhangorodsky, Minjie Xu, Naomi White, and Patrick Lewis. 2024. Replacing Judges with Juries: Evaluating LLM Generations with a Panel of Diverse Models. *arXiv preprint arXiv:2404.18796* (2024).

[226] Tu Vu, Kalpesh Krishna, Salaheddin Alzubi, Chris Tar, Manaal Faruqui, and Yun-Hsuan Sung. 2024. Foundational autoraters: Taming large language models for better automatic evaluation. *arXiv preprint arXiv:2407.10817* (2024).

[227] Binjie Wang, Steffi Chern, Ethan Chern, and Pengfei Liu. 2024. Halu-j: Critique-based hallucination judge. *arXiv preprint arXiv:2407.12943* (2024).

[228] Chihang Wang, Yuxin Dong, Zhenhong Zhang, Ruotong Wang, Shuo Wang, and Jiajing Chen. 2024. Automated Genre-Aware Article Scoring and Feedback Using Large Language Models. *arXiv preprint arXiv:2410.14165* (2024).

[229] Chenglong Wang, Hang Zhou, Kaiyan Chang, Tongran Liu, Chunliang Zhang, Quan Du, Tong Xiao, and Jingbo Zhu. 2023. Learning Evaluation Models from Large Language Models for Sequence Generation. *arXiv preprint arXiv:2308.04386* (2023).

[230] Peiyi Wang, Lei Li, Liang Chen, Zefan Cai, Dawei Zhu, Binghuai Lin, Yunbo Cao, Qi Liu, Tianyu Liu, and Zhifang Sui. 2023. Large language models are not fair evaluators. *arXiv preprint arXiv:2305.17926* (2023).

[231] Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024. Self-taught evaluators. *arXiv preprint arXiv:2408.02666* (2024).

[232] Tianlu Wang, Ping Yu, Xiaoqing Ellen Tan, Sean O'Brien, Ramakanth Pasunuru, Jane Dwivedi-Yu, Olga Golovneva, Luke Zettlemoyer, Maryam Fazel-Zarandi, and Asli Celikyilmaz. 2023. Shepherd: A critic for language model generation. *arXiv preprint arXiv:2308.04592* (2023).

[233] Wanying Wang, Zeyu Ma, Pengfei Liu, and Mingang Chen. 2024. Revisiting Benchmark and Assessment: An Agent-based Exploratory Dynamic Evaluation Framework for LLMs. *arXiv preprint arXiv:2410.11507* (2024).

[234] Xuanhui Wang, Nadav Golbandi, Michael Bendersky, Donald Metzler, and Marc Najork. 2018. Position bias estimation for unbiased learning to rank in personal search. In *Proceedings of the eleventh ACM international conference on web search and data mining*. 610–618.

[235] Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2022. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171* (2022).

[236] Yidong Wang, Zhuohao Yu, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, Xing Xie, et al. 2023. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. *arXiv preprint arXiv:2306.05087* (2023).

[237] Zhilin Wang, Alexander Bukharin, Olivier Delalleau, Daniel Egert, Gerald Shen, Jiaqi Zeng, Oleksii Kuchaiev, and Yi Dong. 2024. HelpSteer2-Preference: Complementing Ratings with Preferences. *arXiv preprint arXiv:2410.01257* (2024).

[238] Zhilin Wang, Yi Dong, Jiaqi Zeng, Virginia Adams, Makesh Narsimhan Sreedhar, Daniel Egert, Olivier Delalleau, Jane Polak Scowcroft, Neel Kant, Aidan Swope, et al. 2023. Helpsteer: Multi-attribute helpfulness dataset for steerlm. *arXiv preprint arXiv:2311.09528* (2023).

[239] Zhaoyang Wang, Weilei He, Zhiyuan Liang, Xuchao Zhang, Chetan Bansal, Ying Wei, Weitong Zhang, and Huaxiu Yao. 2024. Cream: Consistency regularized self-rewarding language models. *arXiv preprint arXiv:2410.12735* (2024).

[240] Matthijs J Warrens. 2015. Five ways to look at Cohen's kappa. *Journal of Psychology & Psychotherapy* 5 (2015).

[241] Ishaan Watts, Varun Gumma, Aditya Yadavalli, Vivek Seshadri, Manohar Swaminathan, and Sunayana Sitaram. 2024. PARIKSHA: A Large-Scale Investigation of Human-LLM Evaluator Agreement on Multilingual and Multi-Cultural Data. *arXiv preprint arXiv:2406.15053* (2024).

[242] Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, et al. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems* 35 (2022), 24824–24837.

[243] Martin Weyssow, Aton Kamanda, and Houari Sahraoui. 2024. CodeUltraFeedback: An LLM-as-a-Judge Dataset for Aligning Large Language Models to Coding Preferences. *arXiv preprint arXiv:2403.09032* (2024).

[244] Tongtong Wu, Linhao Luo, Yuan-Fang Li, Shirui Pan, Thuy-Trang Vu, and Gholamreza Haffari. 2024. Continual learning for large language models: A survey. *arXiv preprint arXiv:2402.01364* (2024).

[245] Tianhao Wu, Weizhe Yuan, Olga Golovneva, Jing Xu, Yuandong Tian, Jiantao Jiao, Jason Weston, and Sainbayar Sukhbaatar. 2024. Meta-rewarding language models: Self-improving alignment with llm-as-a-meta-judge. *arXiv preprint arXiv:2407.19594* (2024).

[246] Shijie Xia, Xuefeng Li, Yixin Liu, Tongshuang Wu, and Pengfei Liu. 2024. Evaluating Mathematical Reasoning Beyond Accuracy. *arXiv preprint arXiv:2404.05692* (2024).

[247] Tingyu Xia, Bowen Yu, Yuan Wu, Yi Chang, and Chang Zhou. 2024. Language Models can Evaluate Themselves via Probability Discrepancy. *arXiv preprint arXiv:2405.10516* (2024).

[248] Qianqian Xie, Weiguang Han, Xiao Zhang, Yanzhao Lai, Min Peng, Alejandro Lopez-Lira, and Jimin Huang. 2023. Pixiu: A large language model, instruction data and evaluation benchmark for finance. *arXiv preprint arXiv:2306.05443* (2023).

[249] Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, et al. 2024. Sorry-bench: Systematically evaluating large language model safety refusal behaviors. *arXiv preprint arXiv:2406.14598* (2024).

[250] Yuxi Xie, Kenji Kawaguchi, Yiran Zhao, James Xu Zhao, Min-Yen Kan, Junxian He, and Michael Xie. 2024. Self-evaluation guided beam search for reasoning. *Advances in Neural Information Processing Systems* 36 (2024).

[251] Yiqing Xie, Sheng Zhang, Hao Cheng, Pengfei Liu, Zelalem Gero, Cliff Wong, Tristan Naumann, Hoifung Poon, and Carolyn Rose. 2024. DOCLENS: Multi-aspect fine-grained evaluation for medical text generation.. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics*.

[252] Yiqing Xie, Wenxuan Zhou, Pradyot Prakash, Di Jin, Yuning Mao, Quintin Fettes, Arya Talebzadeh, Sinong Wang, Han Fang, Carolyn Rose, et al. 2024. Improving Model Factuality with Fine-grained Critique-based Evaluator. *arXiv preprint arXiv:2410.18359* (2024).

[253] Tianyi Xiong, Xiyao Wang, Dong Guo, Qinghao Ye, Haoqi Fan, Quanquan Gu, Heng Huang, and Chunyuan Li. 2024. Llava-critic: Learning to evaluate multimodal models. *arXiv preprint arXiv:2410.02712* (2024).

[254] Can Xu, Qingfeng Sun, Kai Zheng, Xiubo Geng, Pu Zhao, Jiazhan Feng, Chongyang Tao, and Daxin Jiang. 2023. Wizardlm: Empowering large language models to follow complex instructions. *arXiv preprint arXiv:2304.12244* (2023).

[255] Guohai Xu, Jiayi Liu, Ming Yan, Haotian Xu, Jinghui Si, Zhuoran Zhou, Peng Yi, Xing Gao, Jitao Sang, Rong Zhang, et al. 2023. Cvalues: Measuring the values of chinese large language models from safety to responsibility. *arXiv preprint arXiv:2307.09705* (2023).

[256] Shuying Xu, Junjie Hu, and Ming Jiang. 2024. Large Language Models Are Active Critics in NLG Evaluation. *arXiv preprint arXiv:2410.10724* (2024).

[257] Tengyu Xu, Eryk Helenowski, Karthik Abinav Sankararaman, Di Jin, Kaiyan Peng, Eric Han, Shaoliang Nie, Chen Zhu, Hejia Zhang, Wenxuan Zhou, et al. 2024. The perfect blend: Redefining RLHF with mixture of judges. *arXiv preprint arXiv:2409.20370* (2024).

[258] Wenda Xu, Danqing Wang, Liangming Pan, Zhenqiao Song, Markus Freitag, William Yang Wang, and Lei Li. 2023. IN-STRUCTSCORE: Explainable Text Generation Evaluation with Finegrained Feedback. *arXiv preprint arXiv:2305.14282* (2023).

[259] Wenda Xu, Guanglei Zhu, Xuandong Zhao, Liangming Pan, Lei Li, and William Wang. 2024. Pride and prejudice: LLM amplifies self-bias in self-refinement. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 15474–15492.

[260] Xilie Xu, Keyi Kong, Ning Liu, Lizhen Cui, Di Wang, Jingfeng Zhang, and Mohan Kankanhalli. 2023. An LLM can Fool Itself: A Prompt-Based Adversarial Attack. *arXiv preprint arXiv:2310.13345* (2023).

[261] Zhenran Xu, Senbao Shi, Baotian Hu, Jindi Yu, Dongfang Li, Min Zhang, and Yuxiang Wu. 2023. Towards reasoning in large language models via multi-agent peer review collaboration. *arXiv preprint arXiv:2311.08152* (2023).

[262] Le Yan, Zhen Qin, Honglei Zhuang, Rolf Jagerman, Xuanhui Wang, Michael Bendersky, and Harrie Oosterhuis. 2024. Consolidating Ranking and Relevance Predictions of Large Language Models through Post-Processing. *arXiv preprint arXiv:2404.11791* (2024).

[263] Le Yan, Zhen Qin, Honglei Zhuang, Rolf Jagerman, Xuanhui Wang, Michael Bendersky, and Harrie Oosterhuis. 2024. Consolidating Ranking and Relevance Predictions of Large Language Models through Post-Processing. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (Eds.). Association for Computational Linguistics, Miami, Florida, USA, 410–423. https://doi.org/10.18653/v1/2024.emnlp-main.25

[264] Nakyeong Yang, Taegwan Kang, Stanley Jungkyu Choi, Honglak Lee, and Kyomin Jung. 2024. Mitigating biases for instruction-following language models via bias neurons elimination. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 9061–9073.

[265] Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. 2024. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems* 36 (2024).

[266] Hai Ye and Hwee Tou Ng. 2024. Self-Judge: Selective Instruction Following with Alignment Self-Evaluation. *arXiv preprint arXiv:2409.00935* (2024).

[267] Jiayi Ye, Yanbo Wang, Yue Huang, Dongping Chen, Qihui Zhang, Nuno Moniz, Tian Gao, Werner Geyer, Chao Huang, Pin-Yu Chen, et al. 2024. Justice or prejudice? quantifying biases in llm-as-a-judge. *arXiv preprint arXiv:2410.02736* (2024).

[268] Seonghyeon Ye, Yongrae Jo, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, and Minjoon Seo. 2023. Selfee: Iterative self-revising llm empowered by self-feedback generation. *Blog post* (2023).

[269] Seonghyeon Ye, Doyoung Kim, Sungdong Kim, Hyeonbin Hwang, Seungone Kim, Yongrae Jo, James Thorne, Juho Kim, and Minjoon Seo. 2023. Flask: Fine-grained language model evaluation based on alignment skill sets. *arXiv preprint arXiv:2307.10928* (2023).

[270] Ziyi Ye, Xiangsheng Li, Qiuchi Li, Qingyao Ai, Yujia Zhou, Wei Shen, Dong Yan, and Yiqun Liu. 2024. Beyond Scalar Reward Model: Learning Generative Judge from Preference Data. *arXiv preprint arXiv:2410.03742* (2024).

[271] Seungjun Yi, Jaeyoung Lim, and Juyong Yoon. 2024. ProtocoLLM: Automatic Evaluation Framework of LLMs on Domain-Specific Scientific Protocol Formulation Tasks. *arXiv preprint arXiv:2410.04601* (2024).

[272] Koichiro Yoshino, Yun-Nung Chen, Paul Crook, Satwik Kottur, Jinchao Li, Behnam Hedayatnia, Seungwhan Moon, Zhengcong Fei, Zekang Li, Jinchao Zhang, et al. 2023. Overview of the Tenth Dialog System Technology Challenge: DSTC10. *IEEE/ACM Transactions on Audio, Speech, and Language Processing* (2023).

[273] Zhuohao Yu, Chang Gao, Wenjin Yao, Yidong Wang, Wei Ye, Jindong Wang, Xing Xie, Yue Zhang, and Shikun Zhang. 2024. Kieval: A knowledge-grounded interactive evaluation framework for large language models. *arXiv preprint arXiv:2402.15043* (2024).

[274] Weizhe Yuan, Richard Yuanzhe Pang, Kyunghyun Cho, Sainbayar Sukhbaatar, Jing Xu, and Jason Weston. 2024. Self-rewarding language models. *arXiv preprint arXiv:2401.10020* (2024).

[275] Shengbin Yue, Wei Chen, Siyuan Wang, Bingxuan Li, Chenchen Shen, Shujun Liu, Yuxuan Zhou, Yao Xiao, Song Yun, Xuanjing Huang, et al. 2023. Disc-lawllm: Fine-tuning large language models for intelligent legal services. *arXiv preprint arXiv:2309.11325* (2023).

[276] Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. *arXiv preprint arXiv:2305.06311* (2023).

[277] Eric Zelikman, YH Wu, Jesse Mu, and Noah D Goodman. 2024. STaR: Self-taught reasoner bootstrapping reasoning with reasoning. In *Proc. the 36th International Conference on Neural Information Processing Systems*, Vol. 1126.

[278] Weihao Zeng, Can Xu, Yingxiu Zhao, Jian-Guang Lou, and Weizhu Chen. 2024. Automatic Instruction Evolving for Large Language Models. *arXiv preprint arXiv:2406.00770* (2024).

[279] Chen Zhang, João Sedoc, Luis Fernando D'Haro, Rafael Banchs, and Alexander Rudnicky. 2021. Automatic evaluation and moderation of open-domain dialogue systems. *arXiv preprint arXiv:2111.02110* (2021).

[280] Kaiqi Zhang, Shuai Yuan, and Honghan Zhao. 2024. TALEC: Teach Your LLM to Evaluate in Specific Domain with In-house Criteria by Criteria Division and Zero-shot Plus Few-shot. *arXiv preprint arXiv:2407.10999* (2024).

[281] Qiyuan Zhang, Yufei Wang, Tiezheng Yu, Yuxin Jiang, Chuhan Wu, Liangyou Li, Yasheng Wang, Xin Jiang, Lifeng Shang, Ruiming Tang, et al. 2024. RevisEval: Improving LLM-as-a-Judge via Response-Adapted References. *arXiv preprint arXiv:2410.05193* (2024).

[282] Ruoyu Zhang, Yanzeng Li, Yongliang Ma, Ming Zhou, and Lei Zou. 2023. Llmaaa: Making large language models as active annotators. *arXiv preprint arXiv:2310.19596* (2023).

[283] Saizheng Zhang. 2018. Personalizing dialogue agents: I have a dog, do you have pets too. *arXiv preprint arXiv:1801.07243* (2018).

[284] Xiaoyu Zhang, Yishan Li, Jiayin Wang, Bowen Sun, Weizhi Ma, Peijie Sun, and Min Zhang. 2024. Large language models as evaluators for recommendation explanations. In *Proceedings of the 18th ACM Conference on Recommender Systems*. 33–42.

[285] Xinghua Zhang, Bowen Yu, Haiyang Yu, Yangyu Lv, Tingwen Liu, Fei Huang, Hongbo Xu, and Yongbin Li. 2023. Wider and deeper llm networks are fairer llm evaluators. *arXiv preprint arXiv:2308.01862* (2023).

[286] Ruochen Zhao, Wenxuan Zhang, Yew Ken Chia, Deli Zhao, and Lidong Bing. 2024. Auto Arena of LLMs: Automating LLM Evaluations with Agent Peer-battles and Committee Discussions. *arXiv preprint arXiv:2405.20267* (2024).

[287] Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. 2023. A survey of large language models. *arXiv preprint arXiv:2303.18223* (2023).

[288] Xiutian Zhao, Ke Wang, and Wei Peng. 2024. Measuring the inconsistency of large language models in preferential ranking. *arXiv preprint arXiv:2410.08851* (2024).

[289] Yachao Zhao, Bo Wang, Dongming Zhao, Kun Huang, Yan Wang, Ruifang He, and Yuexian Hou. 2023. Mind vs. Mouth: On Measuring Re-judge Inconsistency of Social Bias in Large Language Models. *arXiv preprint arXiv:2308.12578* (2023).

[290] Zihao Zhao, Eric Wallace, Shi Feng, Dan Klein, and Sameer Singh. 2021. Calibrate before use: Improving few-shot performance of language models. In *International conference on machine learning*. PMLR, 12697–12706.

[291] Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

[292] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems* 36 (2023), 46595–46623.

[293] Xiaosen Zheng, Tianyu Pang, Chao Du, Qian Liu, Jing Jiang, and Min Lin. 2024. Cheating automatic llm benchmarks: Null models achieve high win rates. *arXiv preprint arXiv:2410.07137* (2024).

[294] Hongli Zhou, Hui Huang, Yunfei Long, Bing Xu, Conghui Zhu, Hailong Cao, Muyun Yang, and Tiejun Zhao. 2024. Mitigating the Bias of Large Language Model Evaluation. *arXiv preprint arXiv:2409.16788* (2024).

[295] Han Zhou, Xingchen Wan, Yinhong Liu, Nigel Collier, Ivan Vulić, and Anna Korhonen. 2024. Fairer Preferences Elicit Improved Human-Aligned Large Language Model Judgments. *arXiv preprint arXiv:2406.11370* (2024).

[296] Han Zhou, Xingchen Wan, Lev Proleev, Diana Mincu, Jilin Chen, Katherine Heller, and Subhrajit Roy. 2023. Batch calibration: Rethinking calibration for in-context learning and prompt engineering. *arXiv preprint arXiv:2309.17249* (2023).

[297] Ruiyang Zhou, Lu Chen, and Kai Yu. 2024. Is LLM a Reliable Reviewer? A Comprehensive Evaluation of LLM on Automatic Paper Reviewing Tasks. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING 2024)*. 9340–9351.

[298] Xuhui Zhou, Hao Zhu, Leena Mathur, Ruohong Zhang, Haofei Yu, Zhengyang Qi, Louis-Philippe Morency, Yonatan Bisk, Daniel Fried, Graham Neubig, et al. 2023. Sotopia: Interactive evaluation for social intelligence in language agents. *arXiv preprint arXiv:2310.11667* (2023).

[299] Yiyang Zhou, Zhiyuan Fan, Dongjie Cheng, Sihan Yang, Zhaorun Chen, Chenhang Cui, Xiyao Wang, Yun Li, Linjun Zhang, and Huaxiu Yao. 2024. Calibrated self-rewarding vision language models. *arXiv preprint arXiv:2405.14622* (2024).

[300] Yuhang Zhou, Yuchen Ni, Xiang Liu, Jian Zhang, Sen Liu, Guangnan Ye, and Hongfeng Chai. 2024. Are Large Language Models Rational Investors? *arXiv preprint arXiv:2402.12713* (2024).

[301] Lianghui Zhu, Xinggang Wang, and Xinlong Wang. 2023. Judgelm: Fine-tuned large language models are scalable judges. *arXiv preprint arXiv:2310.17631* (2023).

[302] Shengyao Zhuang, Honglei Zhuang, Bevan Koopman, and Guido Zuccon. 2024. A Setwise Approach for Effective and Highly Efficient Zero-shot Ranking with Large Language Models. In *Proceedings of the 47th International ACM SIGIR Conference on Research and Development in Information Retrieval* (Washington DC, USA) *(SIGIR '24)*. Association for Computing Machinery, New York, NY, USA, 38–47. https://doi.org/10.1145/3626772.3657813

[303] Mingchen Zhuge, Changsheng Zhao, Dylan Ashley, Wenyi Wang, Dmitrii Khizbullin, Yunyang Xiong, Zechun Liu, Ernie Chang, Raghuraman Krishnamoorthi, Yuandong Tian, et al. 2024. Agent-as-a-Judge: Evaluate Agents with Agents. *arXiv preprint arXiv:2410.10934* (2024).

[304] Terry Yue Zhuo. 2023. ICE-Score: Instructing Large Language Models to Evaluate Code. *arXiv preprint arXiv:2304.14317* (2023).

[305] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043* (2023).