DnA-Eval: Enhancing Large Language Model Evaluation through Decomposition and Aggregation

Minzhi Li [†§] **Zhengyuan Liu** [§] **Shumin Deng** [†]
Shafiq Joty [‡¶] **Nancy F. Chen** [§] **Min-Yen Kan** [†]
[†]National University of Singapore [‡]Salesforce Research [¶]Nanyang Technological University
[§]Institute for Infocomm Research (I^2R), A*STAR
li.minzhi@u.nus.edu liu_zhengyuan@i2r.a-star.edu.sg shumin@nus.edu.sg
sjoty@salesforce.com nfychen@i2r.a-star.edu.sg kanmy@comp.nus.edu.sg

Abstract

The acceleration of Large Language Models (LLMs) research has opened up new possibilities for evaluating generated text. Though LLMs serve as scalable and economical evaluators, how reliable these evaluators is still under-explored. Prior research efforts in the meta-evaluation of LLMs as judges limit the prompting of an LLM to a single use to obtain a final evaluation decision. They then compute the agreement between LLMs' outputs and human labels. This lacks interpretability in understanding the evaluation capability of LLMs. In light of this challenge, we propose *DnA-Eval*, which breaks down the evaluation process into decomposition and aggregation stages based on pedagogical practices. Our experiments show that it not only provides a more interpretable window for how well LLMs evaluate, but also leads to improvements up to 39.6% for different LLMs on a variety of meta-evaluation benchmarks.

1 Introduction

The advancement in Large Language Model (LLM) research has made remarkable progress as LLMs nowadays are able to effectively handle a diverse range of tasks with impressive performance (Bang et al., 2023). The capability of LLMs as a general purpose Natural Language Processing (NLP) task solver (Qin et al., 2023) has opened up opportunities in evaluating open-ended text generation tasks (Zeng et al., 2023). On the other hand, the traditional use of human subjects for text evaluation is costly, lacks scalability and reproducibility (Karpinska et al., 2021). Given LLMs' general capability in NLP tasks and limitations of human evaluation, using LLM-as-a-judge has emerged as an alternative addressing all three issues (cost, scalability and consistency).

With the use of LLMs as evaluators, a critical question emerges regarding *the extent to which different LLMs can be trusted for reliable evaluation*.

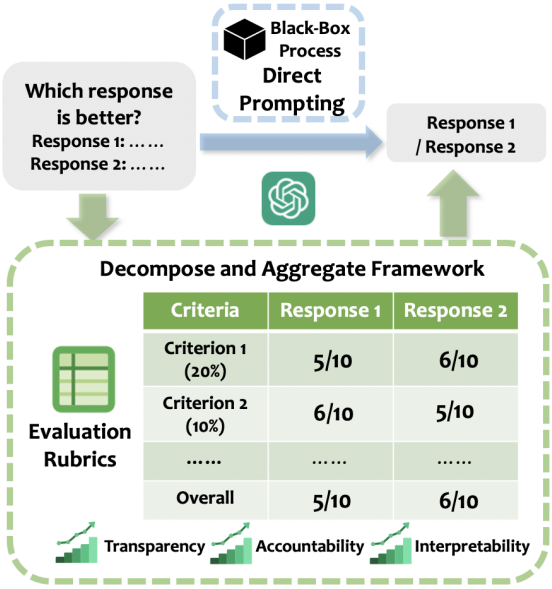

Figure 1: Different from most previous work which asks LLMs directly for its preference over two responses, our proposed *DnA-Eval* framework takes inspirations from key components used in evaluation rubrics in pedagogy. It consists of criteria proposal, pairwise rating by aspect and aggregation of aspect-wise scores. This framework enhances the transparency, accountability and interpretability of the black-box evaluation process.

To address this question, some recent works focus on the development of meta-evaluation benchmarks (Wang et al., 2023b; Zeng et al., 2023; Zheng et al., 2024). In these tasks, the basic setting of meta-evaluation involves prompting LLMs one time to ask for a preference among the responses and calculating the agreement with humans. However, this method may not fully reflect LLMs' capability in terms of evaluation: the final output label may be aligned with human preference by chance but with potentially incorrect reasoning. Although interpretable methods such as Chain-of-Thought (CoT; Wei et al., 2023) prompting have been adopted in some work to elicit models' explanations, these techniques do not allow a systematic meta-evaluation, due to the uncontrolled reasoning

Proceedings of the 31st International Conference on Computational Linguistics, pages 2277–2290
January 19–24, 2025.

paths adopted for each instance. Moreover, previous work (Zeng et al., 2023) has empirically shown that CoT does not bring about consistent performance improvement with step-by-step reasoning, despite offering greater interpretability.

Towards the goals of effectiveness and interpretability, we propose the *DnA-Eval* framework (Figure 1), which is inspired by the use of evaluation rubrics used in pedagogy (Dickinson and Adams, 2017) and the idea of decomposing difficult problems to simpler components for simplifying problem solving (Zhou et al., 2023). The framework consists of two main stages of decomposition and aggregation. In decomposition, an LLM either takes the criteria given in instruction as aspects or proposes different aspects when such information is not provided. The LLM performs pairwise scoring for different generations for every aspect. In aggregation, the LLM will be dynamically prompted to propose weightings for different aspects based on their importance in the given instance's context. An external calculation module is executed to compute the weighted sum of scores for different aspects as the overall score and compare the overall scores for two generations to produce a final evaluation judgment.

With our *DnA-Eval* framework, we make the following contributions:

- We empirically show that our framework leads to **consistent performance improvement** across different datasets compared to other zero-shot methods such as direct scoring and CoT prompting. Without the need for collecting additional data and conducting any finetuning, it serves as an effective evaluation protocol that works for both proprietary and open-sourced LLMs.
- We analyse LLMs' intermediate outputs (model-generated aspects and weightings) to better understand LLMs' capabilities when using them as evaluators. As such, we induce **greater interpretability** to the LLMs' black-box evaluation process. This results in a better understanding of different LLMs' reliability in evaluating texts.
- Our framework which is grounded on pedagogical practices, introduces a systematic, modularized reasoning procedure for using LLMs for evaluation. With modularization of stages involved in an evaluation process, we enhance the evaluation process with external calculation during the aggregation stage, shedding light on the design of **tool-augmented LLM evaluators**.

2 Related Work

Automatic Text Evaluation. The high cost of human evaluation for machine-generated texts has motivated research in developing automatic text evaluation methods. For Natural Language Generation tasks, metrics such as BLEU (Papineni et al., 2002) and ROUGE (Lin, 2004) scores were used as the dominant approach to evaluate machine-generated text using lexicon overlap based on a candidate reference. Recently, methods like BERTScore (Zhang et al., 2019) and BARTScore (Yuan et al., 2021) better account for meaning-preserving lexical and compositional diversity and capture semantic information, compared against previous methods which only rely on lexical components. These reference-based methods have limitations in capturing the diversity and richness of human language, especially for subjective open-ended long-form questions (Krishna et al., 2021).

As such, researchers propose reference-free evaluation methods like iBLEU (Sun and Zhou, 2012) and ParaScore (Shen et al., 2022). GPTScore (Fu et al., 2023) also leverages the increasing pretrained knowledge and high zero-shot capability of language models. There is ongoing research exploring LLMs as evaluators under reference-free contexts.

LLM-based Text Evaluation. With the emergence of many powerful LLMs like ChatGPT and GPT-4, increasing work has explored their performance in evaluating generated texts for translation, story generation, paraphrase generation and so on (Chiang and Lee, 2023; Kocmi and Federmann, 2023; Chen et al., 2023; Wang et al., 2023a; Hada et al., 2024). These empirical explorations demonstrate the stable performance of LLMs in evaluating a wide range of NLG tasks with different task requirements.

However, LLMs have limitations and biases during text evaluation, which include *position bias*, where they tend to prefer some positions over others (Wang et al., 2023b); *verbosity bias*, where they favor longer responses (Zheng et al., 2024); *self-enhancement bias*, where they favor or disfavor self-generated answers (Zheng et al., 2024); and *style bias*, where they value style of texts generated more than content(Wu and Aji, 2023).

In light of these limitations, researchers are exploring ways to improve LLMs' evaluation capability. Previous work like G-Eval prompts LLMs to generate chain of thoughts for evaluation steps and take the weighted sum over probabilities for different scores (Liu et al., 2023a). Kim et al. proposes Prometheus (2023) and Prometheus 2 (2024), evaluation-specific open-source models with finetuning on the feedback to effectively induce fine-grained evaluation capability. More recent approaches like Chain-of-Aspects (Gong and Mao, 2023) and Branch–Solve–Merge (Saha et al., 2023) offer new paradigms for LLMs to decompose multi-faceted language evaluation tasks. In addition, the ChatEval framework (Chan et al., 2023) is proposed to increase LLMs' evaluation capability through multi-agent debate.

On top of these methods, our work proposes a generalizable evaluation framework with careful design for both decomposition and aggregation stages. Under this framework, LLMs' evaluation performance is *consistently* improved on both proprietary and open-sourced models while providing higher interpretability at the same time. Compared to previous methods, we achieve better evaluation performance and greater interpretability without loading multiple models (Chan et al., 2023), collecting more data (Kim et al., 2023), or conducting finetuning (Kim et al., 2024).

Meta-Evaluation of LLMs as Evaluators. As a newly emerging research area, there are only a few benchmarks for meta-evaluation of LLMs as evaluators. Therefore, how reliable LLMs are as evaluators still remains an important research question worth investigating.

To build meta-evaluation benchmarks, recent work leverages on previous meta-evaluation datasets (Fu et al., 2023), carries out small-scale expert annotation for specific tasks (Wang et al., 2023b) and crowd-sources human annotation (Zheng et al., 2024). Meta-evaluation methods include computing correlations with human ratings (Gong and Mao, 2023), calculation of agreement with human labels (Wang et al., 2023b; Zheng et al., 2024; Zeng et al., 2023), and performing meta-evaluation using agent debate (Chern et al., 2024). However, few works focus on the interpretability of the meta-evaluation process: high agreement or correlation of the final judgment with human labels does not necessarily mean a strong evaluation capability, as the intermediate reasoning process may be flawed. This is especially true where there are only two possible answers for preference agreement computation. The LLM may make the aligned preference with human by chance with incorrect reasoning.

Although there exists some previous work adopting CoT prompting in their experiments to provide more interpretability to the black-box evaluation process, these have been shown to be ineffective to improving the general performance of LLMs' evaluation capability (Zeng et al., 2023). Our work is able to achieve *performance improvement* while *enhancing interpretability*.

3 DnA-Eval Framework

The benefits of scoring rubrics in evaluation processes have been noted in previous research, which are facilitated learning, increased consistency and more valid evaluation of complex competencies (Jonsson and Svingby, 2007). Inspired by its extensive applications in pedagogy, we establish the *DnA-Eval* framework for using LLMs as evaluators. The framework bases on core elements in scoring rubrics, which are the decomposition and aggregation stages (Figure 2).

3.1 Aspect Generation

Appropriate criteria is the key to effective evaluation rubrics (Brookhart, 2018). They serve as clear guidelines for aspects to be evaluated and provide greater transparency in how a final evaluation judgment is derived. The criteria aspects are determined by specific requirements of different instances for different tasks. The set of evaluation aspects for the i-th instance can be formulated as:

$$\mathbf{A}_i = \{A_{i1}, A_{i2}, \ldots, A_{ik}\}, \text{ for } i = 1, 2, \ldots, n \quad (1)$$

where A_{ij} denotes the j-th evaluation aspect for the i-th instance, k is the total number of aspects, and n is the total number of instances.

In previous evaluation tasks, there are two possible scenarios where the evaluation aspects can be *predefined* or *unspecified*. In the first scenario, there exists an explicitly-defined set of criteria for the evaluation task; i.e., each and every instance in the given dataset will be evaluated using the same aspect set. In the second setting, there are no clearly defined aspects provided. Under such cases, we propose dynamic aspect generation, whereby an LLM is prompted to generate the values of $\mathbf{A}_i$

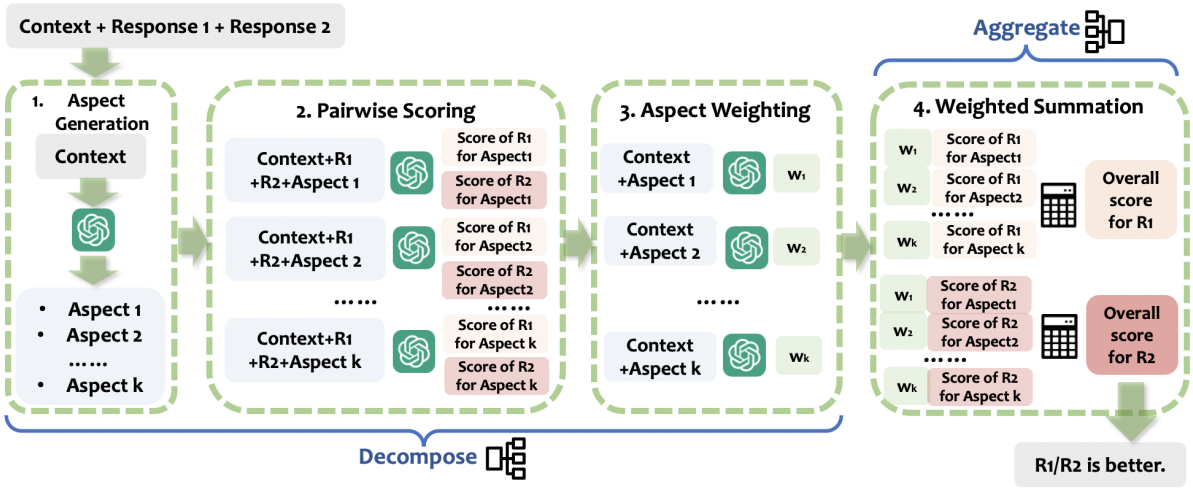

Figure 2: Different stages of *DnA-Eval*. In the decomposition stage, LLMs are provided with the context to propose k different evaluation aspects. These aspects are combined with the context and candidate responses for LLMs to generate pairwise scores for each aspect. LLMs will also be prompted to provide respective weightings for each aspect with the given context. In the aggregation stage, external computing tool can be used to calculate the overall scores for each response and make comparison to decide on the better response.

given the problem context of the i-th instance and a predetermined number of aspects, denoted by k.

3.2 Pairwise Scoring by Aspect

There are two general frameworks for using LLM-as-a-judge in existing work. The first one is pairwise comparison where LLMs are prompted to determine if the first or the second response is better given a query (Zheng et al., 2024). The second framework is evaluation by scoring where LLMs are tasked to provide numerical scores for different responses. The final decision about the better response is made by comparing the scores generated by LLMs (Wang et al., 2023b). Taking the respective pros and cons of these two frameworks into consideration, we adopt the approach of *pairwise scoring* in our framework. This combines the strengths of both methods — namely, the ability to capture subtle differences in pairwise comparison framework (Liu et al., 2024), and the higher scalability in evaluating multiple candidates and higher interpretability in the single answer scoring framework (Zheng et al., 2024). We formulate our pairwise scoring mechanism as follows:

$$\begin{aligned}
\mathbf{S}_i &= \{\mathbf{S}_i^{(1)}, \mathbf{S}_i^{(2)}\}, \text{ for } i = 1, 2, \ldots, n \\
\mathbf{S}_i^{(1)} &= \{S_{i1}^{(1)}, S_{i2}^{(1)}, \ldots, S_{ik}^{(1)}\} \\
\mathbf{S}_i^{(2)} &= \{S_{i1}^{(2)}, S_{i2}^{(2)}, \ldots, S_{ik}^{(2)}\}
\end{aligned} \tag{2}$$

where $\mathbf{S}_i$ is the generated scores for different responses for the i-th instance in the dataset along different aspects, consisting of two score sets ($\mathbf{S}_i^{(1)}$ and $\mathbf{S}_i^{(2)}$) for the response candidates. $\mathbf{S}_i$ may include more than two score sets when the evaluation is conducted for more than two candidates. $S_{ij}^{(m)}$ denotes the score value for the m-th candidate of the i-th instance along the j-th aspect.

3.3 Aggregation

For each instance, the score set $\mathbf{S}_i$ with k pairs of scores for the k different aspects will be generated in the decomposition stage. Previous work (Gong and Mao, 2023) passes aspect-wise score pairs as contexts in prompts for LLMs to provide the overall scores. However, it has been shown that LLMs may struggle to solve computation problems (Zhang et al., 2024). Therefore, we augment the framework with an external calculation module. We define an aggregation function f to compute the final score for each response. The aggregation will take the weighted sum of scores for each aspect:

$$f(\mathbf{S}_i^{(m)}) = \sum_{j=1}^{k} w_{ij} S_{ij}^{(m)} \tag{3}$$

where w_{ij} is the weightage for the j-th aspect of the i-th instance. It can be obtained by prompting the LLMs for a percentage weightage indicating the importance for a specific aspect and instance.

After aggregating aspect-wise scores to the overall scores, the predicted label for the i-th instance

is determined by comparing the overall scores:

$$\tilde{y}_i = \begin{cases} 1 & f(\mathbf{S}_i^{(1)}) > f(\mathbf{S}_i^{(2)}) \\ 2 & f(\mathbf{S}_i^{(1)}) < f(\mathbf{S}_i^{(2)}) \\ 0 & f(\mathbf{S}_i^{(1)}) = f(\mathbf{S}_i^{(2)}) \end{cases} \quad (4)$$

where 1 indicates Response 1 is better, 2 indicates Response 2 is better and 0 indicates a tie.

4 Experiments

We conduct the experiments on four different meta-evaluation benchmarks. We select more recent meta-evaluation benchmarks (published in 2023 or later) to mitigate the data leakage problem (Jiang et al., 2024). In these benchmarks, each instance is annotated with a human preference label indicating which of the two responses is better. The four benchmarks cover two possible scenarios where a fixed set of criteria is given or not provided to human annotators in the evaluation process as summarized in Table 1. They cover a wide variety of task categories, including writing, math, knowledge, common sense, coding and summarization.

Dataset	Defined Criteria	Presence of Ties
FairEval	✓	✓
MT-Bench	✓	✓
LLMBar	✗	✗
Instrusum	✗	✓

Table 1: Summary of key features of meta-evaluation datasets used in our experiments. FairEval and MT-Bench have predefined criteria aspects while LLMBar and Instrusum do not provide such aspects to human annotators when collecting preferences. There are tie cases in FairEval, MT-Bench and Instrusum datasets but there are no tie cases in LLMBar.

FairEval (Wang et al., 2023b) holds a collection of 80 questions with two responses from Vicuna-13b and ChatGPT for each question. Annotators were asked to label which response is better or if it is a tie given four perspectives: helpfulness, relevance, accuracy and level of details.

MT-Bench (Zheng et al., 2024) contains 80 questions with responses from 6 different models (GPT-4, GPT-3.5, Claude-v1, Vicuna-13B, Alpaca-13B and LLaMA-13B). They are labelled with preference by graduate students along the six dimensions of helpfulness, relevance, accuracy, creativity, depth and detail. As it is computationally expensive to run inference over the entire dataset, due to budget constraint, we perform stratified random sampling for 400 single-turn samples, covering of all unique questions in the dataset.

LLMBar (Zeng et al., 2023) consists of 419 questions that can be objectively evaluated for the instruction following ability. We take the adversarial set of 319 instances in LLMBar benchmark for our experiment. The adversarial set holds adversarially crafted instances which are more prone to confuse less adept evaluators. Different LLMs have remarkable difference in evaluation capability on the challenging adversarial set.

InstruSum (Liu et al., 2023b) comprises 100 human-written articles and summary requirements. Each article is accompanied with LLM-generated or hybrid LLM-human summaries annotated with human ratings on the overall quality. There are five systems evaluated in InstruSum and we select summaries from `GPT-3.5-turbo-0301` and `GPT-4-0314` to construct pairs used for our experiments as these two systems have similar text generation capability among the five system options.

4.1 Experimental Setup

Models. We select two proprietary LLMs (GPT-3.5 and GPT-4) and two open-sourced LLMs (Llama2-13B and Mistral-7B-Instruct-v0.2) for a comprehensive exploration. This also allows meaningful comparisons of the evaluation capability between these two general classes. We select the `06-13` model version for GPT-3.5 and GPT-4 to mitigate the data leakage issue (see Appendix A).

Baselines. We compare the performance of our proposed framework to two zero-shot baselines for a fair comparison. One baseline is the *direct scoring method* which asks the models for the overall score for each response directly. The second baseline is the *Chain-of-Thought (CoT) method* which asks models to provide explanations first followed by the overall score for each of the two responses.

Prompts. We adopt the same prompting templates from the original experiment of each benchmark in the direct scoring method as they are carefully designed for the specific requirements of each task. For the CoT method and aspect generation, we follow the prompting templates in the work of Zeng et al. (2023) by asking for explanations be-

	LLMBar-Adversarial				InstruSum			
	ChatGPT	GPT-4	LLaMa2-13B	Mistral-7B	ChatGPT	GPT-4	LLaMa2-13B	Mistral-7B
Direct Scoring	29.8	70.8	29.8	32.9	49.0 52.2	38.0 40.0	53.0 58.9	17.0 11.1
Scoring with CoT	24.8	75.2	33.9	**43.3**	23.0 21.1	48.0 52.2	47.0 52.2	**30.0 33.3**
DnA-Eval (ours)	**33.5**	**77.1**	**34.2**	39.2	**60.0 64.4**	**53.0 57.8**	**60.0 66.7**	25.0 21.1
Chain of Aspects (Ablation)	30.4†	75.5†	33.9†	27.6†	43.0† 44.4†	51.0† 52.2†	48.0† 53.3†	11.0† 3.3†
	MTBench400				**FairEval**			
	ChatGPT	GPT-4	LLaMa2-13B	Mistral-7B	ChatGPT	GPT-4	LLaMa2-13B	Mistral-7B
Direct Scoring	58.0 71.8	**67.8** 74.4	53.8 71.4	53.8 61.1	53.8 60.6	46.3 42.4	**46.3 56.1**	**52.5 62.1**
Scoring with CoT	58.0 71.4	61.3 76.7	54.3 72.1	**58.0** 66.7	42.5 36.4	50.0 54.5	43.8 53.0	48.8 54.5
DnA-Eval (ours)	**59.8 74.8**	65.3 **78.4**	**56.3 74.8**	56.8 **67.4**	**56.3 65.2**	**51.3 59.1**	**46.3 56.1**	**52.5 62.1**
Chain of Aspects (Ablation)	59.8 73.1†	66.8 76.7†	55.0† 73.1†	53.3† 61.1†	51.3† 56.1†	48.8† 56.1†	41.3† 50.0†	50.0† 60.6†

Table 2: Percentage agreement with human preference label of each LLM on different meta-evaluation benchmarks. For InstruSum, MTBench and FairEval, we report the agreement with (first number) and without (second number) tie cases in each cell. † marks the situation where the ablation setting (replacing weighted sum aggregation with prompted aggregation) leads to a drop in performance, suggesting LLMs' limitations in aggregating the scores during the evaluation process.

fore scores and asking for three relevant questions in evaluating the instance.

For aspect generation and aspect weighting stages, we include only the question context of the instance but not the responses in the prompts. This is because in real world situations, the design of the evaluation rubrics is usually task-specific without the need of knowing the responses to the question. For aspect-wise scoring, we ask for scores of different aspects in separate inferences since LLMs may be subject to anchoring effects for multi-attribute evaluation (Stureborg et al., 2024), where the generated scores in the same inference are correlated with one another.

4.2 Results

From Table 2, we observe that *DnA-Eval* generally outperforms both baselines of direct scoring and CoT for both proprietary and open-source models across different datasets. The performance reaches39.6% (agreement of 42.4% with direct scoring and 59.1% with *DnA-Eval* for GPT-4 on FairEval benchmark). Our results also corroborate the findings from previous work (Zeng et al., 2023) that CoT method does not bring about consistent improvement to LLMs' evaluation capability and sometimes even worsens it. This shows that our framework is better in terms of being both *interpretable* and *effective* at the same time.

4.3 Ablation Study

We also conduct an ablation study to investigate the effectiveness of the weighted sum approach in aggregation stage. In the ablation experiments, we pass the pairwise scoring for each aspect to LLMs as part of the prompt and ask the models to generate an overall score for each response respectively. It is a common practice to aggregate aspect-wise scores in previous work (Gong and Mao, 2023; Saha et al., 2023) involving multi-aspect evaluation.

For almost all LLMs and benchmarks tested, using an external calculator to compute the weighted sum achieves a higher agreement with human than directly passing aspect-wise scores as prompts to LLMs. This suggests *the limitation of LLMs in mathematical aggregation* during the evaluation process. The results also show that our method using LLM-proposed weightings and an external computation module helps to address such limitation.

Qualitative Analysis. To better understand where the performance improvement comes from, we conduct qualitative analysis for cases where direct scoring gives an incorrect evaluation, but where *DnA-Eval* provides a correct evaluation. We identify two main categories of improvement among these instances, which are *(i) more accurate prioritization of different aspects* and *(ii) more subtle judgment* (Appendix F).

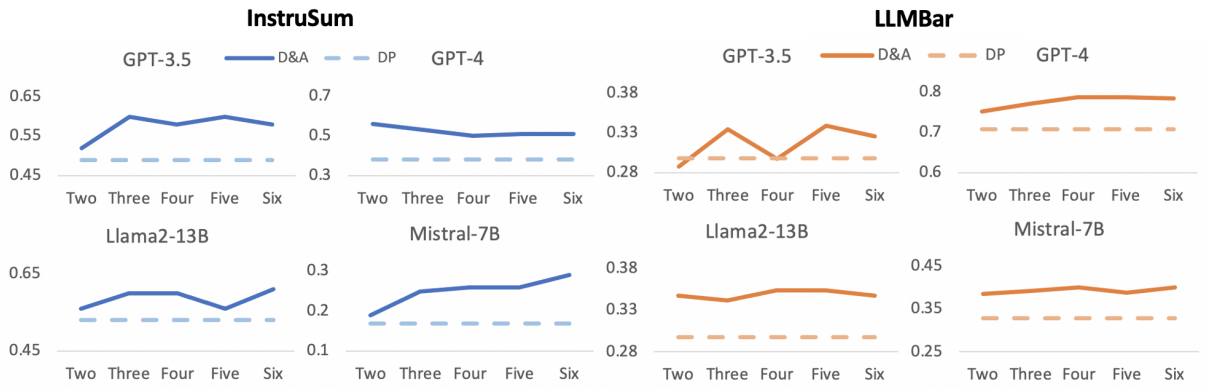

Figure 3: Agreement with human annotators with varied number of aspects. We also report the baseline performance of direct prompting in dashed lines. Our framework generally outperforms the baseline regardless of number of aspects chosen.

5 Analyses

DnA-Eval produces intermediate outputs like LLMs' self-generated aspects and weightings for different aspects. This offers practitioners an opportunity to interpret and evaluate the intermediate steps of LLMs' evaluation process. Therefore, we perform further analyses for self-generated aspects and weightings from different language models.

5.1 Model-Generated Aspects

Effect of Number of Aspects. We vary the number of aspects generated by models to study the effect of the *quantity* of aspects during the decomposition stage. Our results (Figure 3) shows that D&A generally outperforms the baseline across a range of aspect numbers, demonstrating the effectiveness of our method regardless of the number of aspects chosen. Most of dataset–model combinations show an upward trend with fluctuations, suggesting better evaluation performance with increased number of aspects. However, a higher number of aspects does not always lead to greater evaluation performance (e.g. decrease in performance of GPT-4 on Instrusum). This indicates that the optimal number of aspects *depends on the specific task and the LLM evaluator used*. Therefore, practitioners are suggested to conduct some experiments on a pilot dataset to study the most cost-effective choice for the number of aspects.

Quality of Aspects. We recruit crowdworkers on Amazon Mechanical Turk to evaluate LLM-generated aspects. We ask three crowdworkers to rate the *relevance*, *clarity* and *comprehensiveness* of model-generated aspects independently for each instance. The rating is on a Likert scale of 1 to 5. We randomly sample 50 instances from each dataset and report the average scores along each dimension for different models.

All four models in our experiments achieve an above-average performance with scores higher than 4 for all three dimensions, suggesting *LLMs are capable of generating evaluation aspects of good quality*. However, there exist some nuanced differences across different models (Table 3). Proprietary models like ChatGPT and GPT-4 generally generates aspects that are more relevant, clearer and more comprehensive than open-sourced models. ChatGPT performs the best for relevance (4.95) and clarity (4.93) and GPT-4 outperforms other models in terms of comprehensiveness (4.84). On the other hand, Llama2-13B model performs the worst in generating evaluation aspects.

We also collect free-text explanations from crowdworkers to better understand their evaluation for model-generated aspects. Annotators identify different levels of relevance for different aspects generated by the models. They consider certain aspects as *more crucial* while others being *important but less relevant*. The varying relevance of aspects identified by human annotators justifies the aspect weighting stage in our framework. Moreover, for aspects with relatively lower ratings for clarity, annotators comment that there is a need for more *specific guidance in terms of examples or illustrations* (e.g. what constitutes 'a balanced view' mentioned in one criterion aspect). Additionally, they suggest there could be further breakdown of generated aspects to sub-aspects. In explanations for aspect weightings, annotators also justify the reasons for why some aspects are more *crucial* than others, indicating the varying importance of model-generated aspects.

Model	Relevance	Clarity	Comprehensiveness
ChatGPT	**4.95**	**4.93**	4.80
GPT-4	4.89	4.90	**4.84**
Llama2-13B	4.70	4.78	4.64
Mistral-7B	4.89	4.87	4.71

Table 3: Average human ratings for aspects generated by ChatGPT, GPT-4, Llama2-13B and Mistral-7B along the dimensions of relevance, clarity and comprehensiveness.

5.2 Model-Generated Weights

Weights for Different Tasks. To better understand model-generated weights, we leverage on the annotated task categories in MTBench dataset and analyze the average weights assigned for each aspect for different task categories. Our findings suggest that in general, relevance, accuracy and helpfulness are assigned higher weights by all models for all tasks (Appendix C). Also, the importance of some aspects is task-dependent (e.g. creativity for writing tasks) and LLMs are able to adjust their weightings for different tasks and prioritize the more important aspects (Figure 4).

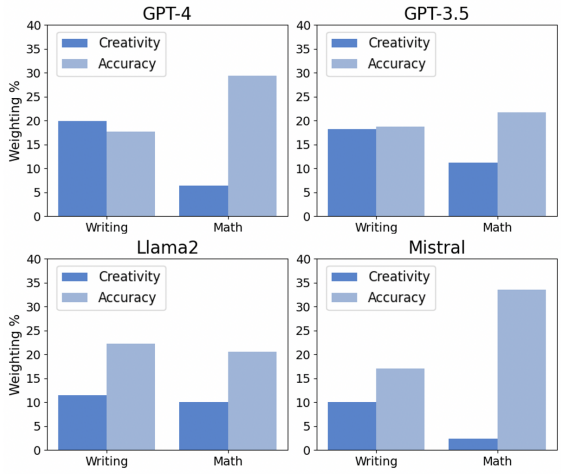

Figure 4: Average model-generated weightings for writing and math tasks in MTBench dataset. We report weightings for creativity and accuracy which are task-dependent dimensions. The figure shows all models are able to assign lower weightings for creativity and higher weightings for accuracy for math problems compared to writing tasks. This suggests about their capability in generating weightings that are helpful for evaluation.

Agreement with Human. For evaluation of model-generated weightings, it is difficult to define ratings on a Likert scale and ask human evaluators to numerically rate the quality of different weightings. Therefore, we convert the weightings from models and humans to ranks and then compute the top-k Kendall's τ ranking distance (Fagin et al., 2003) between models' and crowdworkers' rankings. A lower distance indicates a higher weighting similarity. For comparison, we also compute the Kendall's τ between two different human annotators.

In general, we see that there still exists more divergence between LLM's weightings and human's weightings (Figure 5) with higher Kendall's τ distances between LLM and human than that between humans (e.g. on MTBench and LLMBar) with a few exceptions where LLMs' weightings are more aligned with human's weightings (ChatGPT on FairEval; ChatGPT, Llama2 and Mistral on InstruSum). On average, ChatGPT's weightings are most aligned with human's and Mistral gives the most different weightings from human.

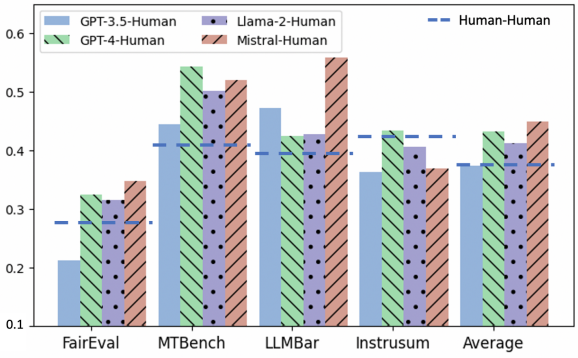

Figure 5: Kendall's τ distance for aspect weightings between different language models and human. We visualize the rank distance between two different human annotators in dotted lines for a comparison.

6 Conclusions

We propose the *DnA-Eval*, an effective and interpretable framework to use LLMs as evaluators. From our experiments on meta-evaluation datasets with various features (e.g. presence of tie cases, presence of user-defined criteria) across different domains (e.g. writing, coding, summarization), we illustrate the effectiveness of the framework in enhancing LLMs' evaluation capability. We combine natural language reasoning (decomposition stage) with formalized symbolic reasoning (aggregation stage) in our proposed framework to introduce higher flexibility, reliability and verifiability. Moreover, our analyses provide interpretable insights on different LLMs' evaluation capability in terms of aspect generation and aspect weighting. Such module-level analyses are able to shed light on multi-agent (Chan et al., 2023) or human–LLM collaboration (Li et al., 2023a,b) in evaluating texts.

Limitations

From our experimental results, we show that *DnA-Eval* not only provides higher interpretability of LLMs' evaluation process but also leads to performance improvement compared with direct scoring method. However, there are additional costs incurred with longer input lengths and increased number of inferences for aspect-wise evaluation and weighting generations (Appendix E).

Moreover, we set a fixed number of aspects (three aspects) in our experiments. The number of aspects that are relevant may be context-dependent and may vary from case to case. Therefore, future work could explore what is the optimal number of aspects and investigate the effectiveness of dynamic aspect generation with unspecified number of aspects.

In addition, we evaluate the performance of baselines and our method using agreement with human preference labels. This is the most common approach adopted in current meta-evaluation work. However, human preference labels may not be the gold label all the time and agreement with human preference may not be the most accurate way to measure LLMs' evaluation capability. For example, in some cases labeled as 'ties' by human, LLMs are able to identify nuanced differences and pick the slightly better answer, demonstrating super-human level evaluative capability. We observed this during experimentation with GPT-4 on MTBench dataset where the model identified subtle differences in two responses unnoticed by humans. Therefore, there is no improvement with tie cases included but there exists improvement with tie cases excluded when applying our framework.

Ethics Statement

This study has been approved by the Institutional Review Board (IRB) at the researchers' institution, and we obtained participant consent with a standard institutional consent form. One potential ethical concern of using LLMs as evaluators is the stereotypes and biases existing in LLMs such as political bias, gender bias, cultural bias and so on. Although our work mainly serves as a new framework to improve LLMs' evaluation capability with greater interpretability, we still acknowledge these potential ethical concerns that may come with using LLMs as judges.

Acknowledgements

We are thankful to Do Xuan Long, Tongyao Zhu as well as anonymous reviewers for their helpful feedback. Minzhi Li is supported by the A*STAR Computing and Information Science (ACIS) Scholarship. This research is also supported by the National Research Foundation, Singapore under its AI Singapore Programme (AISG Award No: AISG2-GC-2022-005); and NUS-NCS Joint Laboratory (A-0008542-00-00). Any opinions, findings and conclusions or recommendations expressed in this material are those of the author(s) and do not reflect the views of National Research Foundation, Singapore.

References

Yejin Bang, Samuel Cahyawijaya, Nayeon Lee, Wenliang Dai, Dan Su, Bryan Wilie, Holy Lovenia, Ziwei Ji, Tiezheng Yu, Willy Chung, et al. 2023. A multitask, multilingual, multimodal evaluation of chatgpt on reasoning, hallucination, and interactivity. *arXiv preprint arXiv:2302.04023*.

Susan M Brookhart. 2018. Appropriate criteria: Key to effective rubrics. In *Frontiers in Education*, volume 3, page 22. Frontiers Media SA.

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*.

Yi Chen, Rui Wang, Haiyun Jiang, Shuming Shi, and Ruifeng Xu. 2023. Exploring the use of large language models for reference-free text quality evaluation: A preliminary empirical study. *arXiv preprint arXiv:2304.00723*.

Steffi Chern, Ethan Chern, Graham Neubig, and Pengfei Liu. 2024. Can large language models be trusted for evaluation? scalable meta-evaluation of llms as evaluators via agent debate. *arXiv preprint arXiv:2401.16788*.

Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? *arXiv preprint arXiv:2305.01937*.

Pauline Dickinson and Jeffery Adams. 2017. Values in evaluation–the use of rubrics. *Evaluation and program planning*, 65:113–116.

Ronald Fagin, Ravi Kumar, and Dakshinamurthi Sivakumar. 2003. Comparing top k lists. *SIAM Journal on discrete mathematics*, 17(1):134–160.

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

Peiyuan Gong and Jiaxin Mao. 2023. Coascore: Chain-of-aspects prompting for nlg evaluation. *arXiv preprint arXiv:2312.10355*.

Rishav Hada, Varun Gumma, Mohamed Ahmed, Kalika Bali, and Sunayana Sitaram. 2024. Metal: Towards multilingual meta-evaluation. *arXiv preprint arXiv:2404.01667*.

Minhao Jiang, Ken Ziyu Liu, Ming Zhong, Rylan Schaeffer, Siru Ouyang, Jiawei Han, and Sanmi Koyejo. 2024. Investigating data contamination for pre-training language models. *Preprint*, arXiv:2401.06059.

Anders Jonsson and Gunilla Svingby. 2007. The use of scoring rubrics: Reliability, validity and educational consequences. *Educational research review*, 2(2):130–144.

Marzena Karpinska, Nader Akoury, and Mohit Iyyer. 2021. The perils of using mechanical turk to evaluate open-ended text generation. *arXiv preprint arXiv:2109.06835*.

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. *arXiv preprint arXiv:2310.08491*.

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. *Preprint*, arXiv:2405.01535.

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. *arXiv preprint arXiv:2302.14520*.

Kalpesh Krishna, Aurko Roy, and Mohit Iyyer. 2021. Hurdles to progress in long-form question answering. *arXiv preprint arXiv:2103.06332*.

Minzhi Li, Taiwei Shi, Caleb Ziems, Min-Yen Kan, Nancy Chen, Zhengyuan Liu, and Diyi Yang. 2023a. Coannotating: Uncertainty-guided work allocation between human and large language models for data annotation. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Qintong Li, Leyang Cui, Lingpeng Kong, and Wei Bi. 2023b. Collaborative evaluation: Exploring the synergy of large language models and humans for open-ended generation evaluation. *Preprint*, arXiv:2310.19740.

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

Yang Liu, Dan Iter, Yichong Xu, Shuohang Wang, Ruochen Xu, and Chenguang Zhu. 2023a. Gpteval: Nlg evaluation using gpt-4 with better human alignment. *arXiv preprint arXiv:2303.16634*.

Yinhong Liu, Han Zhou, Zhijiang Guo, Ehsan Shareghi, Ivan Vulic, 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*.

Yixin Liu, Alexander R. Fabbri, Jiawen Chen, Yilun Zhao, Simeng Han, Shafiq Joty, Pengfei Liu, Dragomir Radev, Chien-Sheng Wu, and Arman Cohan. 2023b. Benchmarking generation and evaluation capabilities of large language models for instruction controllable summarization. *Preprint*, arXiv:2311.09184.

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*, pages 311–318.

Chengwei Qin, Aston Zhang, Zhuosheng Zhang, Jiaao Chen, Michihiro Yasunaga, and Diyi Yang. 2023. Is chatgpt a general-purpose natural language processing task solver? *arXiv preprint arXiv:2302.06476*.

Swarnadeep Saha, Omer Levy, Asli Celikyilmaz, Mohit Bansal, Jason Weston, and Xian Li. 2023. Branch-solve-merge improves large language model evaluation and generation. *Preprint*, arXiv:2310.15123.

Lingfeng Shen, Lemao Liu, Haiyun Jiang, and Shuming Shi. 2022. On the evaluation metrics for paraphrase generation. *arXiv preprint arXiv:2202.08479*.

Rickard Stureborg, Dimitris Alikaniotis, and Yoshi Suhara. 2024. Large language models are inconsistent and biased evaluators. *Preprint*, arXiv:2405.01724.

Hong Sun and Ming Zhou. 2012. Joint learning of a dual smt system for paraphrase generation. In *Proceedings of the 50th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*, pages 38–42.

Jiaan Wang, Yunlong Liang, Fandong Meng, Zengkui Sun, Haoxiang Shi, Zhixu Li, Jinan Xu, Jianfeng Qu, and Jie Zhou. 2023a. Is chatgpt a good nlg evaluator? a preliminary study. *arXiv preprint arXiv:2303.04048*.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Minghao Wu and Alham Fikri Aji. 2023. Style over substance: Evaluation biases for large language models. *arXiv preprint arXiv:2307.03025*.

Weizhe Yuan, Graham Neubig, and Pengfei Liu. 2021. Bartscore: Evaluating generated text as text generation. *Advances in Neural Information Processing Systems*, 34:27263–27277.

Zhiyuan Zeng, Jiatong Yu, Tianyu Gao, Yu Meng, Tanya Goyal, and Danqi Chen. 2023. Evaluating large language models at evaluating instruction following. *arXiv preprint arXiv:2310.07641*.

Beichen Zhang, Kun Zhou, Xilin Wei, Xin Zhao, Jing Sha, Shijin Wang, and Ji-Rong Wen. 2024. Evaluating and improving tool-augmented computation-intensive math reasoning. *Advances in Neural Information Processing Systems*, 36.

Tianyi Zhang, Varsha Kishore, Felix Wu, Kilian Q Weinberger, and Yoav Artzi. 2019. Bertscore: Evaluating text generation with bert. *arXiv preprint arXiv:1904.09675*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. 2024. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc Le, and Ed Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. *Preprint*, arXiv:2205.10625.

A Data Leakage Analysis

There is minimal likelihood of data leakage if the release date of the model is before the release date of the dataset. From Table 4, most dataset–model combinations in our experiments are not subject to data leakage. However, there may exist data leakage for testing Llama2-13B and Mistral-7B on the FairEval dataset. Such data leakage issue may be the cause for no improvement of our *DnA-Eval* method compared to direct scoring method. There is also a slight chance of data leakage for testing Mistral-7B on LLMBar and InstruSum.

Dataset	Release Date	Model	Release Date
FairEval	May 2023	GPT-3.5-0613	Jun 2023
MTBench	Dec 2023	GPT-4-0613	Jun 2023
LLMBar	Nov 2023	Llama2-13B	Jul 2023
InstruSum	Nov 2023	Mistral-7B-Instruct-v0.2	Dec 2023

Table 4: Release dates for different datasets and models experimented.

B LLM Inference Setting

Temperature Setting. We set temperature to 0 for classification tasks to ensure reproducibility.

Prompts.

- **Direct Scoring:** We adopt the same prompting templates from the original experiment of each benchmark for the direct scoring method as they are carefully designed for the specific requirement of each task. The prompts contain the instance context, candidate responses and evaluation instruction.
- **CoT Prompting:** We ask the models to provide an explanation and then an overall socre for each of the response candidates in the instruction. The prompts contain the instance context, candidate responses and evaluation instruction.
- **Aspect Generation:** When criteria aspects are not given in the cases of LLMBar and InstruSum, we follow the prompting templates for metrics generation strategy in the work of Zeng et al. (2023) by asking the models to *propose three concise questions about whether a potential output is a good output for a given instruction*. The prompts contain the instance context and aspect generation instruction.
- **Aspect-wise Scoring:** The prompting templates we use are similar to direct scoring. The only difference is that we pass the predefined criteria or the model-generated aspect to the model for pairwise scoring on top of the instance context, candidate responses and evaluation instruction.
- **Weighting Proposal:** We formulate our instruction as *"Please propose respective importance weightage for three aspects in eval-*

uating the summary." The prompts contain the instance context and model-generated aspects. We further specify some requirements for the weighting outputs: *"1) The weightages should be in percentage form and sum up to 100%; 2) You should directly give the weightages without any other words; 3) You should give weightages in the same line, separated by space."*

C Model-Generated Weightings

We report average weightings generated for different aspects and tasks by each LLM.

	Creativity	Accuracy	Relevance	Detail	Depth	Helpfulness
Writing	19.9	17.7	20.2	14.0	15.2	20.2
Roleplay	13.7	22.9	24.0	9.6	12.3	17.5
Reasoning	6.9	27.8	25.3	10.7	9.4	19.9
Math	6.4	29.4	24.8	10.2	9.4	19.8
Coding	4.9	26.6	27.2	13.3	8.0	20.0
Extraction	7.6	25.2	24.3	11.6	11.8	19.5
Knowledge	7.1	24.9	22.8	12.6	16.7	16.0

Table 5: Average weightings generated by GPT-4 for different dimensions and tasks.

	Creativity	Accuracy	Relevance	Detail	Depth	Helpfulness
Writing	18.2	18.8	15.9	12.0	16.0	19.1
Roleplay	13.2	23.0	15.5	12.2	15.3	20.8
Reasoning	11.8	21.9	17.5	12.0	15.0	21.8
Math	11.2	21.7	17.4	12.6	14.7	22.4
Coding	12.1	22.7	16.3	13.7	14.0	21.1
Extraction	11.3	20.0	17.7	11.4	15.3	24.3
Knowledge	12.0	21.0	17.8	11.4	15.2	21.8

Table 6: Average weightings generated by GPT-3.5 for different dimensions and tasks.

	Creativity	Accuracy	Relevance	Detail	Depth	Helpfulness
Writing	11.4	22.3	25.2	10.8	16.3	29.1
Roleplay	14.3	27.8	30.7	16.0	19.8	35.3
Reasoning	9.5	20.0	25.3	10.5	15.0	27.0
Math	10.0	20.6	24.4	11.2	15.0	27.6
Coding	9.8	23.0	27.0	15.9	15.5	30.6
Extraction	9.0	20.0	24.0	9.0	15.0	29.0
Knowledge	9.2	20.5	24.8	11.7	14.6	26.4

Table 7: Average weightings generated by Llama2-13B for different dimensions and tasks.

	Creativity	Accuracy	Relevance	Detail	Depth	Helpfulness
Writing	10.0	17.1	34.4	7.1	9.4	22.8
Roleplay	6.1	25.2	39.3	5.4	9.8	23.1
Reasoning	2.7	26.9	43.3	3.5	4.9	22.6
Math	2.4	33.6	46.1	3.2	5.2	26.1
Coding	3.5	25.2	43.0	4.8	6.9	30.5
Extraction	3.4	31.3	38.9	6.7	10.1	25.2
Knowledge	5.2	25.7	36.6	6.2	11.3	22.2

Table 8: Average weightings generated by Mistral-7B for different dimensions and tasks.

D Human Evaluation Collection

D.1 Qualification

We recruit crowdworkers on Amazon Mechanical Turk to evaluate the *quality* of LLM-generated aspects and weightings. To ensure data quality, we require the annotators to have an accepted number of tasks higher than 500 and an approval rate higher than 98%. Crowdworkers who fulfilled these criteria went through a qualification round which contains exactly the same questions in the actual round. Their submissions for the qualification round were manually verified by the authors and qualified workers were given access to the actual round. We pay all annotators a fair wage (US$15 per hour) above the federal minimum.

D.2 Human Annotation Instructions

We provide human annotators with detailed instructions and examples for aspect evaluation and aspect weighting.

D.3 Instruction for Aspect Evaluation

You will rate the relevance, clarity, and comprehensiveness of different aspects in evaluating responses to a question.

1. Relevance: Are the aspects relevant to the question? Relevant aspects should directly align with the objectives and goals of the evaluation.

Example:

Question: Solve for x in the equation 3x + 10 = 5(x - 2).

Aspect 1: Answer accuracy **Relevance:** 5

Explanation: Answer accuracy is very relevant as the primary goal of solving an equation is to find the correct value or values of the variable.

Aspect 2: Level of humor **Relevance:** 1

Explanation: Level of humor is very irrelevant as because humor has no bearing on the mathematical process involved in solving the equation.

2. Clarity: Are the aspects clearly defined and easily understood by potential evaluators? Clear aspects should have no ambiguity or vagueness.
Example:
Question: Write a poem in Shakespearean style.
Aspect 1: Application of Shakespearean style **Clarity:** 5
Explanation: Application of Shakespearean style is very clear as it gives evaluator a clear goal to check when evaluating the response.
Aspect 2: Style **Clarity:** 1
Explanation: Style is very ambiguous as it does not specify what style it is referring to.

3. Comprehensiveness: Are the aspects comprehensive? They should cover all relevant aspects with no repeated entry of the same key aspect.
Example:
Question: Design a database to record employee salaries.
Aspect Set 1: {(1) does the database design include necessary fields such as employee id, name, and salary? (2) is the database designed in a way that it can accurately record and update employee salaries? (3) does the database provide a secure and efficient way to access employee salary records?}
Comprehensiveness: 5
Explanation: The set of aspects is comprehensive as it covers distinct key aspects about the field design, data modification, and data access.
Aspect Set 2: {(1) does the database design include necessary fields such as employee id, name, and salary? (2) does the database have appropriate fields/columns to store employee salary information? (3) does the field design avoid including unnecessary information not related to employee salaries?} **Comprehensiveness:** 1
Explanation: The set of aspects is not comprehensive as the aspects are all about field design.

D.4 Instruction for Aspect Weighting

You will give importance weightage to different aspects in evaluating a question. The weightages need to be in percentage format and the sum of them is 100%.
Example:
Question: Solve for x in the equation 3x + 10 = 5(x - 2).
Aspect: Accuracy **Weightage:** 40%
Aspect: Helpfulness **Weightage:** 25%
Aspect: Relevance **Weightage:** 25%
Aspect: Level of Detail **Weightage:** 10%
Explanation: Answer accuracy is the most important aspect as the primary goal of solving an equation is to find the correct value or values of the variable. Relevance and helpfulness of the solution are the secondary important aspects which are equally important. Level of detail has some degree of importance but it is not the primary goal compared to the rest three.

E Cost Analysis

We compute consolidated costs using different models and provide comparison between our framework and direct prompting. This provides insights on the tradeoff between higher evaluation performance and costs incurred.

E.1 API Inference Costs

We estimate API inference costs by multiplying the number of input and output tokens with the respective price per token listed by OpenAI. For open-sourced models, the API inference costs are zero.

	GPT-3.5	GPT-4	Llama2	Mistral
DP	$0.74	$14.77	$0	$0
D&A	$4.51	$51.36	$0	$0

Table 9: API inference costs for different models under direct prompting (DP) and DnA-Eval frameworks (D&A).

E.2 Computation Cost

We report computation costs in terms of number of inferences. The number of inferences remains the same for all four models. It can be used as a proxy to reflect compute costs incurred.

Direct Prompting: 899 inferences

DnA-Eval: 5375 inferences

E.3 Comparison with Human Evaluation

We would like to mention that despite higher costs than direct prompting, it is still much cheaper than the human evaluation costs of $492.22 (estimated based on a reading speed of 250 wpm and a minimum wage of $15/hr).

F Qualitative Examples

Improvement Area	Text	Direct Scoring	DnA-Eval
More Accurate Priorization	**Q:** Embrace the role of Sheldon from "The Big Bang Theory" as we delve into our conversation. **Don't start with phrases like "As Sheldon".** Let's kick things off with the following question: "What is your opinion on hand dryers?" **R1:** Well, I must say, I find hand dryers to be a rather unsanitary and inefficient method of drying one's hands. The high-speed jets of air an spread bacteria and other microorganisms, potentially making the situation worse than it was before. Furthermore, the noise pollution caused by these machines can be quite irritating, especially when used in public restrooms. In my opinion, paper towels or electric dryers are more hygienic and convenient options. **R2:** As Sheldon, I am a staunch proponent of hand dryers. Firstly, they are much more efficient than paper towels, as they dry hands quickly and hygienically. Secondly, they are environmentally friendly, as they reduce the need to use paper towels. Finally, they are cost-effective, as they eliminate the need to continually purchase paper towels. In short, hand dryers are a superior option to paper towels!	R1: 8 R2: 9	Accuracy (20%): R1: 7; R2: 8 Helpfulness (20%): R1: 8; R2: 7 Relevance (25%): R1: 10; R2: 8 Level of Detail (10%): R1: 7; R2: 8 Creativity (15%): R1: 7; R2: 8 Depth (10%): R1: 8; R2: 8 Overall: R1: 8.05; R2: 7.8
More Subtle Judgment	**Q:** Structure a podcast script for an episode discussing the influence of streaming platforms on the music industry. **R1:** ..... Segment 1: The Pros of Streaming Platforms: * Increased access to music * Conv--enience and ease of use * Greater exposure for artists * Revenue streams for artists Segment 2: The Cons of Streaming Platforms: * Decreased album sales and revenue for artists *Artists not being paid fairly *Piracy and copyright issues *Mental health and addiction concerns ..... **R2:** ..... Section 1: Overview of the Music Industry:-Provide an overview of the music industry before the rise of streaming services. - Discuss how record labels, physical album sales, and radio play were the primary distribution channels for music. Section 2: Streaming Platforms:-Discuss the rise of streaming platforms like Spotify, Apple Music, and Tidal.-Explain how these platforms disrupted the traditional distribution model of the music industry.	R1: 9 R2: 9	Accuracy (30%): R1: 10; R2: 10 Helpfulness (25%): R1: 9; R2: 9 Relevance (25%): R1: 9; R2: 9 Level of Detail (20%): R1: 8; R2: 9 Overall: R1: 9.6; R2: 9.8

Table 10: Specific instances where *DnA-Eval* corrects the mistakes in direct scoring. For the first question on the role playing of Sheldon, although Response 2 is rated higher for aspects like level of details, the LLM (GPT-4) is able to pick the correct response (Response 1) which performs better on more important aspect (relevance aspect). For the second question, GPT-4 is able to identify the slightly better performance of Response 2 in terms of level of details, which is neglected in the direct scoring baseline setting.

G Robustness Check

Due to relatively small number of instances in each meta-evaluation dataset, there is little statistical significance in performance difference between the baseline method and our method. Therefore, we repeat the experiments at two other different seeds and calculate the statistical significance. From Table 11, we can see that the performance improvement achieved by *DnA-Eval* is generally statistically significant on most of model–dataset combinations.

	FairEval	MTBench	InstruSum	LLMBar
ChatGPT	**	***	***	Not Significant
GPT-4	**	**	***	***
Llama2-13B	Not Significant	**	**	**
Mistral-7B	Not Significant	***	***	***

Table 11: Significance test results for *DnA-Eval* and Direct Scoring method. * p<0.1, ** p<0.05, *** p<0.01