MAD-Eval: A Misalignment-Driven Evolutionary Evaluator for LLM-as-a-Judge

Anonymous ACL submission

Abstract

LLM-as-a-Judge has emerged as a popular alternative to traditional lexical and embedding-based evaluation metrics, offering improved correlation with human judgments. However, methods relying on heuristic prompts often suffer from misalignment. While recent approaches have incorporated optimization strategies (e.g., prompt iteration), they often lack a mechanism to dynamically evolve evaluation perspectives driven by prediction misalignment. To address this limitation, we propose a **mis**a**l**ignment-**d**riven evolutionary evaluator (MAD-Eval) that treats evaluation alignment as an optimization process. MAD-Eval consists of three components: error-driven perspective evolution to refine evaluation perspectives, instance-aware expert routing to select perspectives tailored to each instruction, and adaptive aggregation to fuse perspective-level scores to align human judgments. In MAD-Eval, misalignment serves as a unified feedback signal driving evolution across all stages: perspective evolution, expert routing, and aggregation. Experiments demonstrate that MAD-Eval consistently outperforms state-of-the-art baselines in consistency with human judgments and transferability across different datasets.

1 Introduction

Recently, advanced large language models (LLMs), which possess emergent reasoning capabilities, can serve as high-fidelity surrogates for human evaluators (Kamalloo et al., 2023) and reffered to LLM-as-a-Judge. Within this domain, while training-based methods (Zhu et al.; Wang et al., 2024b; Peng et al., 2025; Liu et al., 2025) offer optimization, they incur high computational costs and are restricted to white-box architectures. Consequently, prompt-based approaches have emerged as a prevalent alternative, which can leverage existing closed-source strong LLMs via API calls without finetuning.

However, existing prompt-based approaches often face challenges in aligning with human judgments. Single-prompt frameworks (Zhu et al.; Wang et al., 2024b; Jain et al., 2023a,b; Liu et al., 2024a; Dubois et al., 2024) exhibit significant instability, where minor semantic variations can drastically alter outcomes, and often yield narrow assessments of model performance (Mizrahi et al., 2024). Although multi-prompt methods offer broader evaluation perspectives (Zhang et al., 2023; Yang et al., 2024; Kocmi and Federmann, 2023; Chan et al.; Wang et al., 2020), they primarily rely on manual heuristics without systematic optimization. Furthermore, the aggregation strategies in previous multi-prompt methods, such as weighted voting (Zhang et al., 2023; Shankar et al., 2024) or LLM-based synthesis (Yang et al., 2024; Kocmi and Federmann, 2023; Chan et al.; Wang et al., 2020), still lack an optimization process to align with human judgments. Liu et al. (2024c) has explored prompt

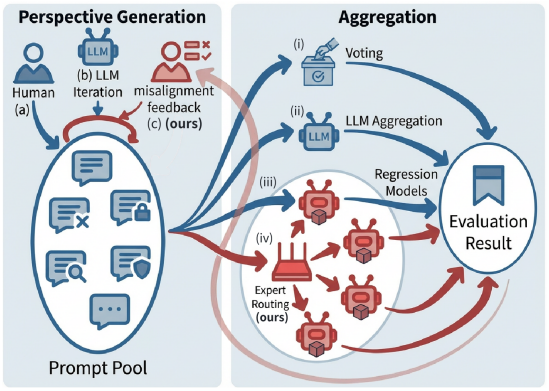

Figure 1: Evaluation perspective generation: (a) Designed by humans; (b) LLM-driven prompt iteration; (c) **Ours:** LLM-driven prompt iteration guided by misalignment feedback. Aggregation of evaluation results: (i) Result determination via voting; (ii) Result aggregation via LLM; (iii) Result prediction using regression models; (iv) **Ours:** Addition of an expert routing layer to select regression models based on the type of instruction to be evaluated, where each regression model has a distinct attention distribution over perspectives.

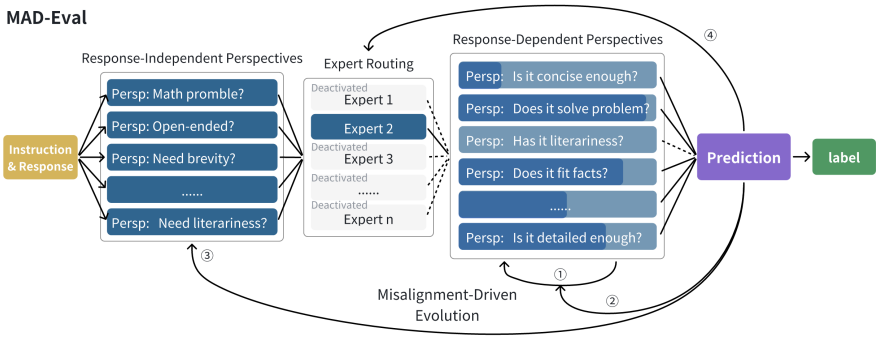

Figure 2: The overall structure of MAD-Eval. Process 1 represents the perspective evolution through "In-Depth Evolving", "In-Breadth Evolving", and "Perspective Refinement"; Processes 2 and 3 represent the perspective evolution through "Semantic Back-propagation"; Process 4 represents the training of expert routing.

iteration and regression-based aggregation to mitigate these issues, yet the prompt iteration process in such methods is typically not driven by the misalignment. Additionally, the aggregation models generally apply a static perspective distribution, which may not fully adapt to varying instruction types.

In this paper, we propose a **mis**a**l**ignment-**d**riven evolutionary evaluator (MAD-Eval) for LLM-as-a-Judge, where two key innovations are shown in Figure 1. The first is misalignment-driven perspective evolution, which iteratively constructs an evaluation pool to capture nuanced human alignment criteria. The second is instance-aware expert routing, which moves beyond "one-size-fits-all" approaches by selecting specialized evaluators for distinct instruction types. To synthesize the final output, each expert aggregates results with its own regression model, focusing on distinct perspective combinations. Figure 3 illustrates an example of how expert routing influences the evaluation process.

Misalignment between predicted results and human judgments provides feedback for all stages, including perspective evolution, expert routing, and predictive model training. Therefore, MAD-Eval is applicable to any LLM evaluation benchmarks, given a small training set to enable automatic learning of human preferences. Additionally, users can freely add new perspectives to enhance performance or address known issues under the human-in-the-loop framework.

Experiments indicate that MAD-Eval exhibits a higher degree of alignment with human compared to baselines. Additionally, we analyzed the importance of perspectives to provide guidance for the perspective construction methods.

Our contributions can be summarized as follows:

- We propose MAD-Eval, a scalable and transferable evaluation framework achieving close alignment with human judgment preferences.
- We applied MAD-Eval in multiple meta-evaluation benchmarks, generating a large perspective pool to provide interpretable guidance for model evaluation.
- We investigate the impact of perspectives constructed using different methods, offering guidance for perspective design and optimization.

2 Related Work

With advancing LLM technology, model capabilities have improved significantly, making LLM-as-a-judge the mainstream for generated text evaluation. In this section, we mainly introduce the prompt-based LLM-as-a-judge.

Single-Prompt Evaluation. The simplest approach uses a single prompt for evaluation (Zhu et al.; Wang et al., 2024b; Jain et al., 2023a,b; Liu et al., 2024a; Dubois et al., 2024), with possible human feedback for prompt optimization (Liu et al., 2024b). For more comprehensive evaluation, some works expand the evaluator network breadth—e.g., using multiple prompts to assess from diverse dimensions (Mehri and Eskenazi, 2020; Kocmi and Federmann, 2023; Jain et al., 2023b; Zhang et al., 2023; Chan et al.; Wang et al., 2020). Others deepen the network via multi-round conversational evaluations (Bai et al., 2024; Yang et al., 2024; Chan et al.; Yue et al., 2023), such as the academic peer-review-inspired approach in Yang et al. (2024).

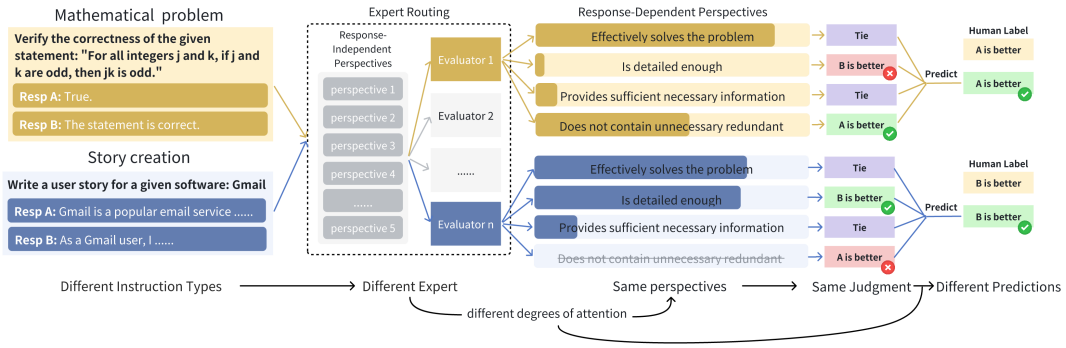

Figure 3: An illustration of how MAD-Eval selects different expert evaluators based on the type of instruction. The dark-colored proportion in the evaluation perspective represents the experts' attention to that perspective (importance as discussed in Section 3.3.4). While most evaluation perspectives yield similar results (only some shown here), the expert evaluating the mathematical problem prioritizes the perspective of conciseness, deeming Response A superior to B. Conversely, the story creation expert values the perspective of detail, favoring a different result.

Multi-Perspective with Multi-Prompt. The evaluator network yields multiple evaluation outcomes. Some studies report dimension-wise model performance without aggregation (Mehri and Eskenazi, 2020; Yue et al., 2023); others aggregate them into a single comprehensive score for overall response quality assessment, via LLMs (Yang et al., 2024; Kocmi and Federmann, 2023; Chan et al.; Wang et al., 2020), simple/weighted voting (Zhang et al., 2023; Shankar et al., 2024), or regression-based models (Mehri and Eskenazi, 2020; Liu et al., 2024c).

Optimization for LLM-as-a-Judge. Reinforcement learning with human feedback (RLHF) (Ouyang et al., 2022) enables LLMs to better align with human preferences. Some works fine-tune LLMs for stronger evaluation performance or adaptation to specific domain evaluation needs (Zhu et al.; Wang et al., 2024b; Brief et al., 2024; Li et al., 2024a; Wang et al., 2024a; Saha et al.), but this approach applies only to white-box models, which ignores the usage of state-of-the-art (SOTA) closed-source LLMs.

3 Method

3.1 Problem Formulation

LLM-as-a-Judge is to align LLM-based evaluators (E) with human judgments on text evaluation tasks. Formally, given an instruction I and model-generated responses, E is required to output a prediction $\hat{y}$ that approximates the human preference label y. This formulation encompasses two task types:

- **Pairwise Comparison**: Given an instruction I and two candidate responses $\{R_A, R_B\}$, the evaluator determines their relative quality, outputting $\hat{y} \in \{A, B, Tie\}$.
- **Individual Scoring**: Given an instruction I and a single response R, the evaluator assigns a scalar score $\hat{y}$ (e.g., on a scale of 1 to 5).

The core optimization goal is to minimize the "misalignment" between the evaluator's prediction $\hat{y}$ and the ground-truth human label y.

3.2 Overview of MAD-Eval

MAD-Eval moves beyond traditional pipelines by establishing an iterative, closed-loop framework designed to align LLM evaluations with human judgments. The key idea is to utilize the misalignment, which is defined as the discrepancy between model predictions and human judgments, as the primary signal for optimization.

The overall framework is illustrated in Figure 2. MAD-Eval drives a cyclical optimization process which includes three sub-processes:

- **Perspective Evolution**: Evaluation perspectives are generated and refined via semantic back-propagation derived from error analysis.
- **Expert Routing**: Instructions are dynamically matched to the most suitable single expert evaluator via an instance-aware routing mechanism.
- **Adaptive Aggregation**: The activated expert utilizes a predictive model (e.g., Random Forest) to synthesize judgments from multiple perspectives into a final prediction.

At the dataset level, prediction errors are gathered to form a feedback signal that back-propagates to all three stages, driving an optimization loop.

3.3 Misalignment-Driven Perspective Evolution

Previous work mainly design evaluation perspectives via human definition or static heuristics. To avoid local optima of perspectives, we iteratively updates the perspective pool $\mathcal{P}$ with an evolutionary mechanism comprising initialization, semantic optimization, and exploration phases.

3.3.1 Initialization

The evolution process begins with a seed perspective pool $\mathcal{P}_0$. To ensure high-quality cold starts, we employ a hybrid strategy (see Figure 6 in Appendix B for the examples): (1) *Rule Decomposition*, which breaks down benchmark annotation guidelines into fine-grained criteria; (2) *Type-Specific Presets*, manually crafting high-frequency focus points for distinct instruction types; and (3) *Prompt Transfer*, incorporating prompts from existing methods (e.g., PandaLM (Wang et al., 2024b), JudgeLM (Zhu et al.)).

3.3.2 Semantic Back-Propagation

We formalize the perspective update as a *semantic pseudo-gradient descent* process (Figure 4(3)). Let $\mathcal{M}_{opt}$ denote the optimizer LLM (e.g., DeepSeek-v3). At step t, the perspective pool is updated by generating a set of new perspectives $\Delta\mathcal{P}$ guided by a semantic pseudo-gradient $\mathbf{g}$:

$$\mathcal{P}^{(t+1)} \leftarrow \mathcal{P}^{(t)} \cup \underbrace{\mathcal{M}_{opt}(\mathcal{P}^{(t)} \mid \mathbf{g})}_{\Delta\mathcal{P}} \quad (1)$$

where $\mathbf{g}$ represents a natural language instruction derived from the optimization objective. We define three distinct forms of $\mathbf{g}$, corresponding to exploitation and exploration strategies.

The primary driver for alignment is the error-derived gradient. We define the *failure set* $\mathcal{D}_{fail} = \{(I_i, \mathbf{R}_i, y_i) \mid \hat{y}_i \neq y_i\}$ where the evaluator's prediction diverges from human labels. The semantic gradient $\mathbf{g}_{opt}$ is computed by analyzing the reasoning behind these misalignments:

$$\mathbf{g}_{opt} = \mathbb{E}_{(I,\mathbf{R},y)\sim\mathcal{D}_{fail}} \left[\text{Analyze}(I, \mathbf{R}, \hat{y}, y)\right] \quad (2)$$

Here, $\text{Analyze}(\cdot)$ denotes the process where $\mathcal{M}_{opt}$ interprets the error and generates a corrective guideline. This effectively "back-propagates" the error signal $\mathcal{L}$ into the perspective space, ensuring new perspectives explicitly address previous blind spots.

3.3.3 Evolutionary Exploration

To prevent convergence to local optima and ensure coverage, we employ exploration-based gradients targeting robustness (In-Depth Evolving) and diversity (In-Breadth Evolving):

$$\mathbf{g}_{deep} = \{\text{Perturbation}(p) \mid p \in \mathcal{P}^{(t)}\} \quad (3)$$

$$\mathbf{g}_{broad} = \{\text{Orthogonal}(p) \mid p \in \mathcal{P}^{(t)}\} \quad (4)$$

- **In-Depth Evolving** (Figure 4(1)): The gradient $\mathbf{g}_{deep}$ instructs the optimizer to perform semantic perturbations (e.g., reformatting, paraphrasing) on existing perspectives, enhancing evaluator robustness.

- **In-Breadth Evolving** (Figure 4(2)): The gradient $\mathbf{g}_{broad}$ directs the optimizer to generate perspectives semantically orthogonal to the current pool, thereby expanding the evaluation manifold.

3.3.4 Perspective Refinement and Selection

To maximize the downstream utility of the evolved perspectives, we apply structural transformations followed by a rigorous selection process.

We first adapt evolved perspectives to enhance their discriminatory power and support the routing mechanism via two key modifications: (1) *Output granularity refinement*, converting discrete classification prompts into continuous scoring scales (e.g., $-x$ to $+x$) to boost evaluator sensitivity and quantify performance gaps ignored by coarse-grained labels; (2) *Routing perspective derivation*, extracting response-independent perspectives (e.g., "*Is data support necessary?*") from response-dependent perspectives to refine instance-aware routing logic.

An overly large pool raises high inference costs. To balance performance and efficiency, we adopt an importance-based pruning strategy: We fit a surrogate decision tree on the training set using perspectives from the pool, then calculate each perspective's Gini importance and retain a fixed-size subset of top perspectives, denoted by $\mathcal{P}$.

3.4 Instance-Aware Expert Routing

Unlike traditional "one-size-fits-all" evaluators, MAD-Eval introduces a mixture-of-experts (MoE) approach. To facilitate this, we first partition

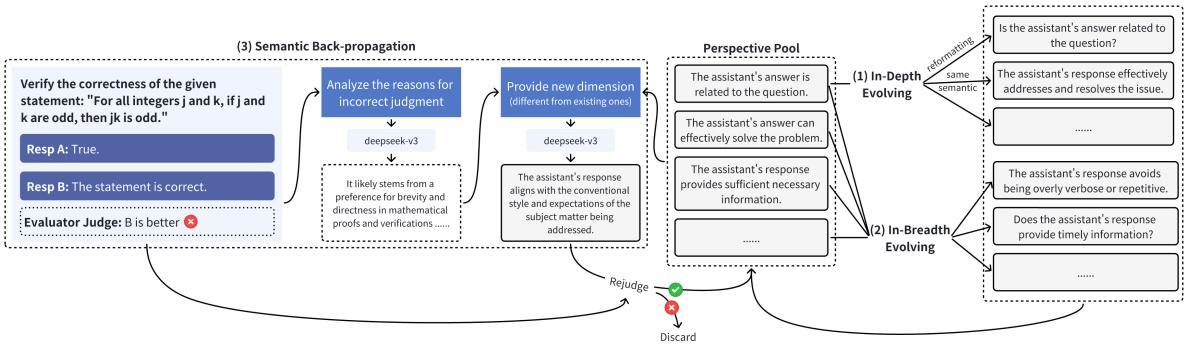

Figure 4: An example diagram of the process of perspective evolution through in-depth evolving, in-breadth evolving and semantic back-propagation.

the evolved perspective pool $\mathcal{P}$ into response-independent ($\mathcal{P}_{ind}$) and response-dependent ($\mathcal{P}_{dep}$) subsets.

We employ $\mathcal{P}_{ind} = \{p_1^{ind}, \ldots, p_M^{ind}\}$ to project the raw instruction I into a structured semantic space. We construct a weighted feature vector to reflect the varying significance of different perspectives.

Let $w_j \in [0, 1]$ denote the importance weight of perspective p_j^{ind}, derived from the surrogate decision tree metioned in Sec. 3.3.4. We map I to a continuous feature vector $\mathbf{v}_{meta} \in [0, 1]^M$:

$$\mathbf{v}_{meta} = \left[w_j \cdot \phi(I, p_j^{ind}) \right]_{j=1}^{M} \tag{5}$$

where $\phi(I, p_j^{ind})$ is a binary indicator function (1 if the instruction aligns with perspective p_j^{ind}, else 0). By scaling features with their importance w_j, we ensure that critical semantic dimensions dominate the subsequent clustering and routing process.

We maintain a set of N_E specialized expert models $\mathcal{M} = \{M_1, \ldots, M_{N_E}\}$. During training, instructions are clustered into N_E groups based on $\mathbf{v}_{meta}$, with centroids $\mathbf{C} = \{\mathbf{c}_1, \ldots, \mathbf{c}_{N_E}\}$. During inference, the routing function $\mathcal{R}(\cdot)$ selects the optimal expert index i^* by minimizing the distance to centroids:

$$i^* = \mathcal{R}(I) = \arg \min_{i \in \{1, \ldots, N_E\}} \|\mathbf{v}_{meta} - \mathbf{c}_i\|_2 \tag{6}$$

This mechanism ensures that, for instance, a math-heavy instruction is routed to an expert optimized for logical rigorousness.

3.5 Adaptive Aggregation

Once the expert M_{i^*} is selected, we perform the comprehensive evaluation using the response-dependent subset $\mathcal{P}_{dep} = \{p_1^{dep}, \ldots, p_K^{dep}\}$.

The judge LLM $\mathcal{M}_{judge}$ evaluates the instruction-response pair $(I, \mathbf{R})$ against each perspective in $\mathcal{P}_{dep}$, yielding a high-dimensional score vector $\mathbf{s} \in \mathbb{R}^K$:

$$\mathbf{s} = \left[\mathcal{M}_{judge}(I, \mathbf{R} \mid p_k^{dep}) \right]_{k=1}^{K} \tag{7}$$

Each element s_k represents the model's performance on a specific evolutionary dimension (e.g., logical coherence or creativity).

The final prediction $\hat{y}$ is derived by applying the selected expert model M_{i^*} to adaptively aggregate $\mathbf{s}$. Unlike simple voting, the expert model learns a non-linear weight distribution tailored to the instruction type:

$$\hat{y} = M_{i^*}(\mathbf{s}; \theta_{i^*}) \tag{8}$$

where θ_{i^*} represents the learned parameters (e.g., random forest structure) of expert i. This step completes the optimization loop, directly minimizing the prediction error defined in the problem formulation.

3.6 Training and Implementation

The one-round training procedure of MAD-Eval is detailed as follows: (1) We first derive $\mathbf{v}_{meta}$ for each sample in the training set for perspective evolution (S_{pt}) via $\mathcal{P}_{ind}$, and employ the K-Means (MacQueen, 1967) to realize the mapping from $\mathbf{v}_{meta}$ to $(M, \mathbf{c})$; (2) We then optimize the aggregation parameters θ of each M on the training set for aggregation (S_{at}), where the weight assigned to each sample is negatively correlated with the distance between the sample's $\mathbf{v}_{meta}$ and the centroid c of the corresponding M; (3) The trained MAD-Eval is employed to generate predictions on S_{at},

yielding the $\mathcal{D}_{fail}$; (4) We sequentially perform semantic back-propagation (leveraging $\mathcal{D}_{fail}$), in-depth/in-breadth evolving, refinement, and selection to update the $\mathcal{P}$.

We will iteratively repeat this process until the predefined number of rounds is reached, and in each round, all components of MAD-Eval will undergo the misalignment-driven evolution as described above.

4 Experiment

4.1 Experimental Setup

4.1.1 Meta-Evaluation Benchmark

We employed two types of meta-benchmarks: (1) Pairwise comparison: **MT-Bench** (Zheng et al., 2023) [1], **PandaLM Benchmark** (Wang et al., 2024b) and **Chatbot Arena** (Chiang et al., 2024) [2]; (2) Individual scoring: **Topical-Chat** (Gopalakrishnan et al., 2019) and **Summeval** (Fabbri et al., 2021) [3].

4.1.2 Setup of MAD-Eval

Backbone LLMs Qwen2.5-7B-Instruct (Team, 2024), Qwen2.5-72B-Instruct, Qwen3-0.6B (Team, 2025)[4], DeepSeek-v3 (DeepSeek-AI et al., 2024), GPT-3.5-turbo (Brown et al., 2020), GPT-4 (Achiam et al., 2023) [5].

Perspective Pool Based on empirical results, we set $\mathcal{P}_{ind}$'s maximum size to 32; $\mathcal{P}_{dep}$ is set to 50 as it incurs no extra evaluation costs.

Aggregation For each M, We conducted a grid search across various regression models (see Appendix Table 7 for the search space). Consistent validation-set empirical results favored random forest, so we adopt it (applying the classifier mode and regressor mode for the pairwise comparison and individual scoring tasks, respectively) as the default aggregator with hyperparameters dynamically optimized per expert via grid search.

Training set We use a 1,000-sample subset of Chatbot Arena and a 500-sample subset of MT-Bench (with no overlap with the test set) as S_{pt} and adopt cross-validation to split S_{at} in our experiments.

Method	Acc	Precision	F1
Fine-Tuned Model			
- PandaLM-7B	0.5926	0.5728	0.5456
- PandaLM-70B	0.6687	0.7402	0.6923
- JudgeLM-7B	0.6507	0.6689	0.6192
- JudgeLM-13B	0.6897	0.6821	0.6512
- JudgeLM-33B	0.7518	0.693	0.6973
PandaLM Prompt			
- Qwen2.5-7B-Instruct	0.6947	0.7313	0.6732
- GPT-3.5-Turbo	0.7267	0.7577	0.7356
- DeepSeek-v3	0.6777	0.7198	0.6865
JudgeLM Prompt			
- Qwen2.5-7B-Instruct	0.6727	0.7287	0.6890
- GPT-3.5-Turbo	0.7057	0.6903	0.6792
- DeepSeek-v3	0.7457	0.7793	0.7564
CoT Prompt			
- Qwen2.5-7B-Instruct	0.5976	0.7178	0.6267
- GPT-3.5-Turbo	0.7528	0.7517	0.7500
- DeepSeek-v3	0.7788	0.7759	0.7769
MAD-Eval (w/o Expert)			
- Qwen2.5-7B-Instruct	0.8028	0.8040	0.8012
- GPT-3.5-Turbo	0.8038	0.8014	0.8021
- DeepSeek-v3	0.8008	0.7978	0.7977
MAD-Eval			
- Qwen2.5-7B-Instruct (213)	**0.8288**	**0.8298**	**0.8279**
- GPT-3.5-Turbo (76)	**0.8188**	**0.8173**	**0.8164**
- DeepSeek-v3 (79)	**0.8228**	**0.8220**	**0.8214**

Table 1: Results on PandaLM Benchmark.

4.1.3 Baselines

First, we compared several single-prompt evaluation methods[6], where the **PandaLM Prompt**, **JudgeLM Prompt**, and **CoT** (Wei et al., 2022) were evaluated on pairwise comparison benchmarks; in contrast, the **USR Prompt** (Mehri and Eskenazi, 2020) was applied to Topical-Chat. Second, we compared our results with the benchmark metrics reported by several multi-prompt evaluation methods (including **G-Eval** (Liu et al., 2023), **ChatEval** (Chan et al.), **HD-Eval** (Liu et al., 2024c) and **AUTOCALIBRATE** (Liu et al., 2024b)) and fine-tuned model-based evaluation methods (including **USR**, **PandaLM**, **JudgeLM**, **UniEval** (Zhong et al., 2022), **Prometheus-13B** (Kim et al., 2024), **Auto-J-13B** (Li et al., 2024b) and **Themis-8B** (Hu et al., 2024)). A brief introduction to the aforementioned baselines is provided in Appendix B.

To verify the impact of Expert Routing, we also adopt MAD-Eval with Expert Routing removed (**MAD-Eval (w/o Expert)**) for ablation studies on the PandaLM, MT-Bench, and Chatbot Arena.

[1] A single-round conversation subset with a size of 1000 instances.

[2] A randomly sampled subset with a size of 6000 instances.

[3] Since Summeval involves only summarization tasks, we did not apply the Expert Routing of MAD-Eval.

[4] Due to space limitations, some of the experimental results of Qwen2.5-72B-Instruct and Qwen3-0.6b are presented in the Appendix B.

[5] GPT-4 was only applied to the Topical-Chat and Summeval due to cost.

[6] All the prompts mentioned have been adjusted in terms of output format. See Appendix A for details.

Method	Acc	F1	ρ	τ
PandaLM Prompt				
- Qwen2.5-7B-Instruct	0.5510	0.4806	0.4462	0.4205
- GPT-3.5-Turbo	0.6360	0.6096	0.5199	0.4882
- DeepSeek-v3	0.6390	0.6156	0.5350	0.5008
JudgeLM Prompt				
- Qwen2.5-7B-Instruct	0.5870	0.5480	0.4492	0.4211
- GPT-3.5-Turbo	0.5780	0.5144	0.4458	0.4215
- DeepSeek-v3	0.6510	0.6265	***0.5431**	***0.5111**
CoT Prompt				
- Qwen2.5-7B-Instruct	0.5800	0.5526	0.4601	0.4260
- GPT-3.5-Turbo	0.6240	0.5644	0.5322	0.5030
- DeepSeek-v3	0.6380	0.5894	***0.5431**	***0.5138**
MAD-Eval (w/o Expert)				
- Qwen2.5-7B-Instruct	0.6350	0.6275	0.5232	0.4862
- GPT-3.5-Turbo	0.6470	0.6420	0.5412	0.5039
- DeepSeek-v3	0.6790	0.6737	0.5988	0.5598
MAD-Eval				
- Qwen2.5-7B-Instruct (254)	**0.6520**	**0.6477**	**0.5393**	**0.5032**
- GPT-3.5-Turbo (69)	**0.6600**	**0.6561**	**0.5607**	**0.5231**
- DeepSeek-v3 (80)	**0.6810**	**0.6744**	**0.6116**	**0.5711**

Table 2: Results on a single-round conversation subset of MT-Bench.

Method	Acc	F1	ρ	τ
PandaLM Prompt				
- Qwen2.5-7B-Instruct	0.4032	0.3018	0.2019	0.1899
- GPT-3.5-Turbo	0.4558	0.3977	0.2451	0.2295
- DeepSeek-v3	0.4147	0.3510	0.1558	0.1458
JudgeLM Prompt				
- Qwen2.5-7B-Instruct	0.4557	0.4133	0.2468	0.2285
- GPT-3.5-Turbo	0.4562	0.3873	0.2615	0.2456
- DeepSeek-v3	0.4208	0.3532	0.1793	0.1677
CoT Prompt				
- Qwen2.5-7B-Instruct	0.4618	0.4429	0.2602	0.2371
- GPT-3.5-Turbo	0.4522	0.3879	0.2534	0.2373
- DeepSeek-v3	***0.4952**	0.4294	***0.3474**	***0.3251**
MAD-Eval (w/o Expert)				
- Qwen2.5-7B-Instruct	0.4787	0.4681	0.2776	0.2526
- GPT-3.5-Turbo	0.4512	0.4378	0.2459	0.2225
- DeepSeek-v3	0.5072	0.4911	0.3592	0.3270
MAD-Eval				
- Qwen2.5-7B-Instruct (254)	**0.4832**	**0.4717**	**0.2838**	**0.2586**
- GPT-3.5-Turbo (36)	**0.4642**	**0.4444**	**0.2724**	**0.2476**
- DeepSeek-v3 (36)	**0.5278**	**0.5115**	**0.3771**	**0.3457**

Table 3: Results on a subset of Arena Chatbot.

Dataset	All	> 0.25	> 0.5	> 0.67	> 0.8
PandaLM Benchmark	0.8248	0.8489 (36.4%)	0.9234 (26.1%)	0.9559 (20.4%)	0.9783 (13.8%)
MT-Bench	0.6430	0.7154 (36.9%)	0.7817 (25.2%)	0.8712 (13.2%)	0.9688 (3.2%)
Chatbot Arena	0.4828	0.5405 (23.0%)	- (0.0%)	- (0.0%)	- (0.0%)

Table 4: Accuracy of cases under different confidence thresholds (case percentages in parentheses).

4.2 Main Results

Main results are presented in Tables 1, 2, 3, 5, with metrics (average of five runs) calculated as per specific needs: Accuracy (Acc), Precision, F1-score (F1), Spearman correlation (ρ), Kendall-Tau correlation (τ), and Mean Squared Error (MSE). Asterisk-marked (*) metrics denote models outperforming MAD-Eval under at least one model.

Due to the cost of generating perspective pools, the pools we employ may shrink for larger models and larger test sets—explaining why smaller models sometimes outperform in MAD-Eval experiments. Pool sizes are noted in parentheses after model names. Note that pool size only affects training cost; the number of perspectives incurring actual evaluation costs depends on the maximum size of $\mathcal{P}_{dep}$ (i.e., 32).

PandaLM Benchmark, MT-Bench, and Chatbot Arena As shown in Tables 1, 2, 3, MAD-Eval significantly outperforms all baselines with the same model across these benchmarks.

Furthermore, MAD-Eval achieves superior performance compared with MAD-Eval (w/o Expert), demonstrating that the Expert Routing can improve performance without increasing evaluation costs.

Topical-Chat and Summeval Table 5 shows that in Topical-Chat, MAD-Eval with Qwen2.5-7B-Instruct outperforms all baselines except ChatEval with GPT-4 and Themis-8B, but significantly surpasses all baselines when using Qwen2.5-72B-Instruct, GPT-3.5-turbo, or DeepSeek-v3.

Furthermore, on SummEval, MAD-Eval with Qwen2.5-7B-Instruct outperforms all baselines except GPT-4-based methods and Themis-8B, while MAD-Eval with Qwen2.5-72B-Instruct, GPT-3.5-Turbo, DeepSeek-v3 and GPT-4 surpasses all baselines.

Themis-8B outperformed MAD-Eval (Qwen2.5-7B-Instruct) in Topical-Chat and SummEval experiments. We argue that fine-tuning methods reasonably outperform prompt-based counterparts on same-scale models, yet fine-tuning is costly for large models and inapplicable to powerful black-box models (e.g., GPT-4)—prompt-based methods, by contrast, can conveniently utilize state-of-the-art model capabilities.

4.3 Analysis

Perspective Importance Analysis We also analyzed perspective importance (discussed in Section 3.3.4) from different sources. Figure 5 shows the top 15 response-related perspectives ranked by importance (full ranking and distribution ratios in Appendix B). (1) Seed perspectives and those generated via semantic back-propagation mostly ap-

Benchmark	Topical-Chat		Summeval	
Method	ρ	τ	ρ	τ
Fine-Tuned Model				
- USR	0.4192	0.422	-	-
- UniEval (770M)	0.533	***0.577**	0.474	0.377
- Prometheus-13B	0.434	-	0.163	0.142
- Auto-J-13B	0.425	-	0.198	0.172
- Themis-8B	***0.725**	-	***0.553**	***0.499**
USR Prompt				
- Qwen2.5-7B-Instruct	0.4202	0.3773	-	-
- Qwen2.5-72B-Instruct	0.6463	0.5803	-	-
- GPT-3.5-Turbo	0.6287	0.5645	-	-
- DeepSeek-v3	0.6131	0.5505	-	-
AUTOCALIBRATE				
- GPT-4	-	-	***0.529**	***0.474**
G-Eval				
- GPT-3.5	0.574	***0.585**	0.401	0.32
- GPT-4	0.575	***0.588**	***0.514**	***0.418**
ChatEval				
- GPT-3.5 (Single-Agent)	0.544	0.503	-	-
- GPT-4 (Single-Agent)	***0.658**	***0.611**	-	-
- GPT-3.5 (Multi-Agent)	0.552	0.51	-	-
- GPT-4 (Multi-Agent)	***0.684**	***0.632**	-	-
HD-Eval				
- GPT-4	***0.638**	-	***0.535**	-
MAD-Eval				
- Qwen2.5-7B-Instruct (213/67)	**0.6379**	**0.5453**	**0.5007**	**0.4744**
- Qwen2.5-72B-Instruct (211/67)	**0.7569**	**0.6624**	**0.5546**	**0.5284**
- GPT-3.5-Turbo (77/48)	**0.7455**	**0.6488**	**0.5608**	**0.5335**
- DeepSeek-v3 (82/41)	**0.7617**	**0.6678**	**0.5728**	**0.5461**
- GPT-4 (32/30)	**0.7644**	**0.6695**	**0.5996**	**0.5729**

Table 5: Results on Topical-Chat and Summeval. Empty cells denote unreported data.

Dataset	NA Consist.	NO Consist.	AvgPI	AvgConf.
PandaLM Benchmark	0.5332	0.6102	0.1153	0.576
MT-Bench	0.4224	0.4103	0.0526	0.4596
Chatbot Arena	0.2697	0.1918	0.0454	0.2136

Table 6: **NA Consist.** (new annotation consistency): average Cohen's Kappa coefficient between new annotations; **NO Consist.** (new-original consistency): average Cohen's Kappa coefficient between new annotations and original labels; **AvgPI**: average accuracy improvement of MAD-Eval over all baselines on the dataset; **AvgConf.**: average confidence of MAD-Eval on the dataset.

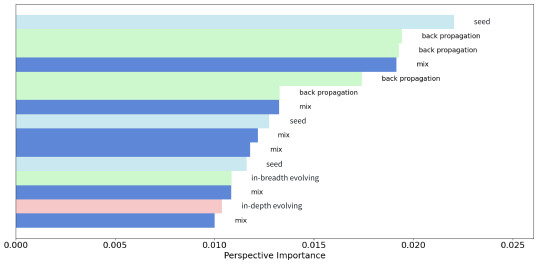

Figure 5: The top 15 response-related perspectives sorted by importance. (Experiments using Qwen2.5-7B-Instruct on PandaLM, MT-Bench and Chatbot Arena as the dataset) Figure 9 in the Appendix displays the prompts used for the top 5 perspectives.

pear in top ranks, which attests to their high quality. (2) Perspectives from single in-depth or in-breadth evolving generally rank lower on average, while multi-iteration (i.e., mix) ones, though varying in quality, include a large number of top-ranked ones. (3) The average importance of perspectives after applying output granularity refinement increased by 37.0%.

Confidence Analysis We also calculated confidence based on the probability of each label when MAD-Eval makes pairwise comparison predictions. As Table 4 shows, cases with higher confidence exhibit significantly higher accuracy across all benchmarks. This means that MAD-Eval can perform pre-judgment, with low-confidence cases then submitted to manual judgment.

Additionally, MAD-Eval's average performance improvement varies by dataset. Table 6 indicates less improvement on datasets with lower average confidence. We speculate low confidence stems from higher dataset difficulty or poorer manual label quality. To verify, we sampled 100 cases from each of these datasets, with three annotators per case, and observed *NA Consist.* and *NO Consist.* The difficulty ranking derived from *NA Consist.*—PandaLM < MT-Bench < Chatbot Arena—matches the average confidence ranking. Moreover, Chatbot Arena shows significantly lower *NO Consist.* than *NA Consist.*, suggesting poorer original label quality may hinder MAD-Eval's learning of correct preferences, leading to reduced performance improvement. This conclusion is reasonable because the annotations of Chatbot Arena were made by different real-world users rather than a specific group of trained annotators.

5 Conclusion

We propose MAD-Eval, a scalable and transferable evaluation framework achieving close alignment with human judgment preferences. We then apply this framework to multiple meta-evaluation benchmarks, generating a large perspective pool to provide interpretable guidance for model evaluation. Furthermore, We investigate the impact of perspectives constructed using different methods, offering guidance for perspective design and optimization.

Limitations

MAD-Eval employs multi-prompt evaluation, which incurs relatively high costs. However, since the evaluation of each perspective is independent of others, we can execute evaluations across all perspectives in parallel to reduce time costs. Additionally, the expert routing process does not rely on model responses, so it only needs to be performed once on the benchmark and can be reused in all subsequent evaluations. Experimental results also demonstrate that MAD-Eval can achieve better performance on small models (e.g., Qwen2.5-7B-Instruct) with lower memory costs compared to large models (e.g., GPT-3.5-turbo and DeepSeek-v3) under single-prompt evaluation.

Furthermore, existing fine-tuning methods can achieve or surpass the performance of MAD-Eval on models of comparable scale. However, fine-tuning is costly for large models and inapplicable to powerful black-box models (e.g., GPT-4). In contrast, MAD-Eval can conveniently leverage the capabilities of state-of-the-art models to attain performance levels that are difficult for fine-tuning methods to reach in practical scenarios.

References

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

Yushi Bai, Jiahao Ying, Yixin Cao, Xin Lv, Yuze He, Xiaozhi Wang, Jifan Yu, Kaisheng Zeng, Yijia Xiao, Haozhe Lyu, and 1 others. 2024. Benchmarking foundation models with language-model-as-an-examiner. *Advances in Neural Information Processing Systems*, 36.

Meni Brief, Oded Ovadia, Gil Shenderovitz, Noga Ben Yoash, Rachel Lemberg, and Eitam Sheetrit. 2024. Mixing it up: The cocktail effect of multi-task fine-tuning on llm performance–a case study in finance. *arXiv preprint arXiv:2410.01109*.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901.

Chi-Min Chan, Weize Chen, Yusheng Su, Jianxuan Yu, Wei Xue, Shanghang Zhang, Jie Fu, and Zhiyuan Liu. Chateval: Towards better llm-based evaluators through multi-agent debate. In *The Twelfth International Conference on Learning Representations*.

Wei-Lin Chiang, Lianmin Zheng, Ying Sheng, Anastasios N Angelopoulos, Tianle Li, Dacheng Li, Banghua Zhu, Hao Zhang, Michael I Jordan, Joseph E Gonzalez, and 1 others. 2024. Chatbot arena: an open platform for evaluating llms by human preference. In *Proceedings of the 41st International Conference on Machine Learning*, pages 8359–8388.

DeepSeek-AI, Aixin Liu, Bei Feng, Bing Xue, Bingxuan Wang, Bochao Wu, Chengda Lu, Chenggang Zhao, Chengqi Deng, Chenyu Zhang, Chong Ruan, Damai Dai, Daya Guo, Dejian Yang, Deli Chen, Dongjie Ji, Erhang Li, Fangyun Lin, Fucong Dai, and 181 others. 2024. Deepseek-v3 technical report. *Preprint*, arXiv:2412.19437.

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*.

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*, 9:391–409.

Karthik Gopalakrishnan, Behnam Hedayatnia, Qinlang Chen, Anna Gottardi, Sanjeev Kwatra, Anu Venkatesh, Raefer Gabriel, and Dilek Hakkani-Tür. 2019. Topical-Chat: Towards Knowledge-Grounded Open-Domain Conversations. In *Proc. Interspeech 2019*, pages 1891–1895.

Xinyu Hu, Li Lin, Mingqi Gao, Xunjian Yin, and Xiaojun Wan. 2024. Themis: A reference-free nlg evaluation language model with flexibility and interpretability. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 15924–15951.

Neel Jain, Khalid Saifullah, Yuxin Wen, John Kirchenbauer, Manli Shu, Aniruddha Saha, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2023a. Bring your own data! self-supervised evaluation for large language models. *arXiv preprint arXiv:2306.13651*.

Sameer Jain, Vaishakh Keshava, Swarnashree Mysore Sathyendra, Patrick Fernandes, Pengfei Liu, Graham Neubig, and Chunting Zhou. 2023b. Multi-dimensional evaluation of text summarization with in-context learning. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8487–8495.

Ehsan Kamalloo, Nouha Dziri, Charles Clarke, and Davood Rafiei. 2023. Evaluating open-domain question answering in the era of large language models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 5591–5606.

Seungone Kim, Juyoung Suk, and Minjoon Seo. 2024. Prometheus 2: An open source language model specialized in evaluating other language models. In *EMNLP 2024*. Association for Computational Linguistics (ACL).

Tom Kocmi and Christian Federmann. 2023. Large language models are state-of-the-art evaluators of translation quality. In *24th Annual Conference of the European Association for Machine Translation*, page 193.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2024a. Generative judge for evaluating alignment. In *ICLR*.

Junlong Li, Shichao Sun, Weizhe Yuan, Run-Ze Fan, Hai Zhao, and Pengfei Liu. 2024b. Generative judge for evaluating alignment. In *ICLR*.

Xiao Liu, Xuanyu Lei, Shengyuan Wang, Yue Huang, Andrew Feng, Bosi Wen, Jiale Cheng, Pei Ke, Yifan Xu, Weng Lam Tam, and 1 others. 2024a. Alignbench: Benchmarking chinese alignment of large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11621–11640.

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. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024b. Calibrating llm-based evaluator. In *Proceedings of the 2024 joint international conference on computational linguistics, language resources and evaluation (lrec-coling 2024)*, pages 2638–2656.

Yuxuan Liu, Tianchi Yang, Shaohan Huang, Zihan Zhang, Haizhen Huang, Furu Wei, Weiwei Deng, Feng Sun, and Qi Zhang. 2024c. Hd-eval: Aligning large language model evaluators through hierarchical criteria decomposition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 7641–7660.

Zijun Liu, Peiyi Wang, Runxin Xu, Shirong Ma, Chong Ruan, Peng Li, Yang Liu, and Yu Wu. 2025. Inference-time scaling for generalist reward modeling. *CoRR*.

James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, volume 5, pages 281–298. University of California press.

Shikib Mehri and Maxine Eskenazi. 2020. Usr: An unsupervised and reference free evaluation metric for dialog generation. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Moran Mizrahi, Guy Kaplan, Dan Malkin, Rotem Dror, Dafna Shahaf, and Gabriel Stanovsky. 2024. State of what art? a call for multi-prompt llm evaluation. *Transactions of the Association for Computational Linguistics*, 12:933–949.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, and 1 others. 2022. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35:27730–27744.

F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay. 2011. Scikit-learn: Machine learning in Python. *Journal of Machine Learning Research*, 12:2825–2830.

Hao Peng, Yunjia Qi, Xiaozhi Wang, Zijun Yao, Bin Xu, Lei Hou, and Juanzi Li. 2025. Agentic reward modeling: Integrating human preferences with verifiable correctness signals for reliable reward systems. *arXiv preprint arXiv:2502.19328*.

Swarnadeep Saha, Xian Li, Marjan Ghazvininejad, Jason E Weston, and Tianlu Wang. Learning to plan & reason for evaluation with thinking-llm-as-a-judge. In *Forty-second International Conference on Machine Learning*.

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*, pages 1–14.

Qwen Team. 2024. Qwen2.5: A party of foundation models.

Qwen Team. 2025. Qwen3.

Alex Wang, Kyunghyun Cho, and Mike Lewis. 2020. Asking and answering questions to evaluate the factual consistency of summaries. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Tianlu Wang, Ilia Kulikov, Olga Golovneva, Ping Yu, Weizhe Yuan, Jane Dwivedi-Yu, Richard Yuanzhe Pang, Maryam Fazel-Zarandi, Jason Weston, and Xian Li. 2024a. Self-taught evaluators. *CoRR*.

Yidong Wang, Zhuohao Yu, Wenjin Yao, Zhengran Zeng, Linyi Yang, Cunxiang Wang, Hao Chen, Chaoya Jiang, Rui Xie, Jindong Wang, and 1 others. 2024b. Pandalm: An automatic evaluation benchmark for llm instruction tuning optimization. In *ICLR*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Fei Xia, Ed Chi, Quoc V Le, Denny Zhou, and 1 others. 2022. Chain-of-thought prompting elicits reasoning in large language models. *Advances in neural information processing systems*, 35:24824–24837.

Blair Yang, Fuyang Cui, Keiran Paster, Jimmy Ba, Pashootan Vaezipoor, Silviu Pitis, and Michael R Zhang. 2024. Report cards: Qualitative evaluation of language models using natural language summaries. *arXiv preprint arXiv:2409.00844*.

Xiang Yue, Boshi Wang, Ziru Chen, Kai Zhang, Yu Su, and Huan Sun. 2023. Automatic evaluation of attribution by large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4615–4635.

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*.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, and 1 others. 2023. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in neural information processing systems*, 36:46595–46623.

Ming Zhong, Yang Liu, Da Yin, Yuning Mao, Yizhu Jiao, Pengfei Liu, Chenguang Zhu, Heng Ji, and Jiawei Han. 2022. Towards a unified multi-dimensional evaluator for text generation. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 2023–2038.

Lianghui Zhu, Xinggang Wang, and Xinlong Wang. Judgelm: Fine-tuned large language models are scalable judges. In *The Thirteenth International Conference on Learning Representations*.

A Prompts

- **JudgeLM prompt with adaptability changes made to the prompts regarding the output format:**

  Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose the assistant tha follows the user's instructions and answers the user's question better. Your evaluation should consider factors such as the helpfulness, relevance, accuracy, depth, creativity, and level of detail of their responses. Avoid any position biases and ensure that the order in which the responses were presented does not influence your decision. Do not allow the length of the responses to influence your evaluation. Do not favor certain names of the assistants. Be as objective as possible. **Output your final verdict by strictly following this format without providing any explanation**: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

  ```
  [User Question]
  {instruction}
  [The Start of Assistant A's Answer]
  {response_a}
  [The End of Assistant A's Answer]
  [The Start of Assistant B's Answer]
  {response_b}
  [The End of Assistant B's Answer]
  ```

- **PandaLM prompt with adaptability changes made to the prompts regarding the output format:**

  Below are two responses for a given task. The task is defined by the Instruction with an Input that provides further context. Evaluate the responses and generate a reference answer for the task.

  ```
  ### Instruction:
  {instruction}

  ### Response 1:
  {response_a}

  ### Response 2:
  {response_b}
  ```

  Output your final verdict by strictly following this format without providing any explanation: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

- **USR prompt with adaptability changes made to the prompts regarding the output format:**

  Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Please score the assistant's response according to the following rules:

  - A score of 1 (very bad). A completely invalid response. It would be difficult to recover the conversation after this.

  - A score of 2 (bad). Valid response, but otherwise poor in quality.

  - A score of 3 (neutral) means this response is neither good nor bad. This response has no negative qualities, but no positive ones either.

  - A score of 4 (good) means this is a good response, but falls short of being perfect because of a key flaw.

  - A score of 5 (very good) means this response is good and does not have any strong flaws.

  Output your final verdict by strictly following this format without providing any explanation: "[[1]]" if the score is 1, "[[2]]" if the score is 2, "[[3]]" if the score is 3, "[[4]]" if the score is 4, "[[5]]" if the score is 5.

  ```
  [User Question]
  {question}
  [The Start of the Assistant's Answer]
  {answer}
  [The End of the Assistant's Answer]
  ```

- **Our prompt used on pairwise comparison benchmarks:**

  Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below. You should choose an assistant that better meets this requirement:

  ```
  **{perspective}**
  ```

  . Avoid any position biases and ensure that the order in which the responses were presented

does not influence your decision. Do not allow the length of the responses to influence your evaluation. Be as objective as possible.

Output your final verdict by strictly following this format without providing any explanation: "[[A]]" if assistant A is better, "[[B]]" if assistant B is better, and "[[C]]" for a tie.

```
[User Question]
{question}
[The Start of Assistant A's Answer]
{answer_a}
[The End of Assistant A's Answer]
[The Start of Assistant B's Answer]
{answer_b}
[The End of Assistant B's Answer]
```

- **Our prompt used for "Changing from Classification to Scoring":**

Please act as an impartial judge and evaluate the quality of the responses provided by two AI assistants to the user question displayed below.

You should choose an assistant that better meets this requirement:

```
**{perspective}**
```

. You need to give a score ranging from -5 to 5 according to the following rules: - When the score is positive, it means that Assistant A's response is more in line with the requirements than Assistant B's. In this case, the closer the score is to 5, the greater the gap. - When the score is negative, it indicates that Assistant B's response is more in line with the requirements than Assistant A's. Here, the closer the score is to -5, the greater the gap. - The closer the score is to 0, the smaller the gap between their responses.

Output your final verdict by strictly following this format without providing any explanation: "[[x]]", where x is the score you assign. For example, "[[3]]" represents 3 points, and "[[-3]]" represents -3 points.

```
[User Question]
{question}
[The Start of the Assistant's Answer]
{answer}
[The End of the Assistant's Answer]
```

- **Our prompt used on individual scoring benchmarks:**

Please act as an impartial judge and evaluate the quality of the response provided by an AI assistants to the user question displayed below.

You need to determine whether the assistant's response meets the following requirement:

```
**{perspective}**
```

You need to give a score from 1 to 5. The higher the score, the more the assistant meets the requirements.

As two extreme examples, a score of 5 means the assistant's answer perfectly meets the requirements, and a score of 1 means the assistant's answer is completely contrary to the requirements.

Output your final verdict by strictly following this format without providing any explanation: "[[1]]" if the score is 1, "[[2]]" if the score is 2, "[[3]]" if the score is 3, "[[4]]" if the score is 4, "[[5]]" if the score is 5.

```
[User Question]
{question}
[The Start of the Assistant's Answer]
{answer}
[The End of the Assistant's Answer]
```

B Introduction to Baselines

- **JudgeLM** (Zhu et al.): Fine-tuning LLMs (Vicuna) as scalable judges to evaluate LLMs in open-ended benchmarks.
- **PandaLM** (Wang et al., 2024b): Fine-tuning LLMs (LLaMA) to distinguish the superior model.
- **CoT**: The prompt template used in JudgeLM also applies the Chain-of-Thought (CoT) (Wei et al., 2022) method.
- **USR** (Mehri and Eskenazi, 2020): An unsupervised and reference-free evaluation metric for dialog generation, trained on Topical-Chat.
- **UniEval** (Zhong et al., 2022): A unified multi-dimensional evaluator for NLG, trained with google/t5-v1_1-large as the base model.
- **G-Eval** (Liu et al., 2023): A framework of using LLMs (GPT-3.5 / GPT-4) with CoT and a form-filling paradigm, to assess the quality of NLG outputs.
- **ChatEval** (Chan et al.): A multi-agent referee team that enables LLMs (GPT-3.5 / GPT-4) to autonomously discuss and evaluate generated responses for open-ended questions and NLG tasks.
- **AUTOCALIBRATE** (Liu et al., 2024b): A multi-stage, gradient-free approach to automatically calibrate and align an LLM-based (GPT-4) evaluator toward human preference.
- **HD-EVAL** (Liu et al., 2024c): A white-box framework aligning LLM evaluators with human preferences via hierarchical criteria decomposition, attribution pruning, and iterative alignment.
- **Themis-8B** (Hu et al., 2024): An 8B-parameter LLM for reference-free NLG evaluation, trained via multi-perspective consistency verification and rating-guided preference alignment on the NLG-Eval corpus, enabling flexible, interpretable, and generalizable evaluation across diverse NLG tasks.
- **Auto-J-13B** (Li et al., 2024b): A fine-tuned LLM evaluator for assessing alignment of NLG outputs, trained with human annotations to judge generative model alignment.
- **Prometheus-13B** (Kim et al., 2024): A fine-tuned LLM evaluator trained via weight merging of direct assessment and pairwise ranking-specialized models on custom datasets.

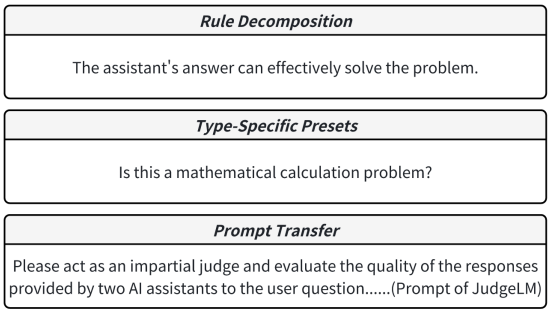

Figure 6: Example of the method adopted for the cold-start of the perspective pool mentioned in section 3.3.1.

Model & Hyperparameters	Search Space
Logistic Regression	
- penalty	['l1', 'l2', 'none']
- max_iter	[100, 500, 1000]
Decision Tree	
- max_depth	[3, 5, 10]
- min_samples_split	[2, 5, 10, 15]
- min_samples_leaf	[1, 2, 5, 10]
Random Forest	
- n_estimators	[50, 100]
- max_depth	[3, 5, 10]
- min_samples_split	[2, 5, 10, 15]
- min_samples_leaf	[1, 2, 5, 10]

Table 7: Search spaces for prediction models and hyperparameters. Among them, the names of hyperparameters and their meanings refer to scikit-learn (Pedregosa et al., 2011).

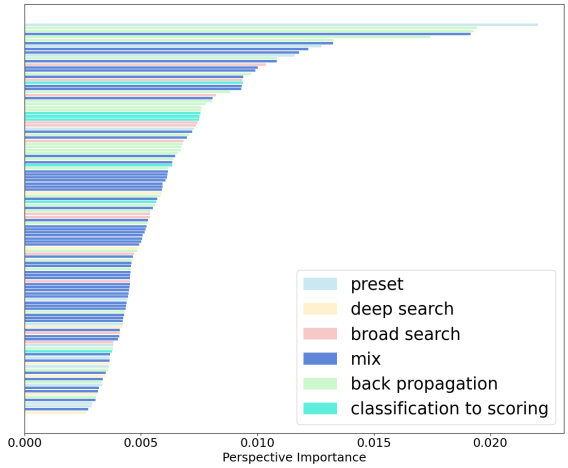

Figure 7: Ranking of response-dependent perspectives importance. (Experiments using Qwen2.5-7b-Instruct on PandaLM, MT-Bench and Chatbot Arena as the dataset)

Method	PandaLM Prompt	JudgeLM Prompt	CoT	USR Prompt	MAD-Eval
Arena Chatbot					
Acc	0.4950	0.5113	0.4915	-	**0.5235**
F1	0.4944	0.5083	0.4266	-	**0.5276**
ρ	0.3472	0.3611	0.3346	-	**0.3730**
τ	0.3138	0.3277	0.3136	-	**0.3398**
MT-bench					
Acc	0.6450	0.6550	0.6350	-	**0.6700**
F1	0.6431	0.6417	0.5883	-	**0.6679**
ρ	0.5674	0.5620	0.5408	-	**0.5856**
τ	0.5259	0.5223	0.5103	-	**0.5464**
PandaLM					
Acc	0.6176	0.5776	0.7658	-	**0.8248**
F1	0.6444	0.6295	0.7702	-	**0.8235**
ρ	0.5356	0.5800	0.6573	-	**0.7076**
τ	0.5024	0.5434	0.6309	-	**0.6921**
Topical-Chat					
MSE	-	-	-	1.3194	**0.7556**
ρ	-	-	-	0.6949	**0.7569**
τ	-	-	-	0.6134	**0.6624**

Table 8: Results on Qwen2.5-72B-Instruct.

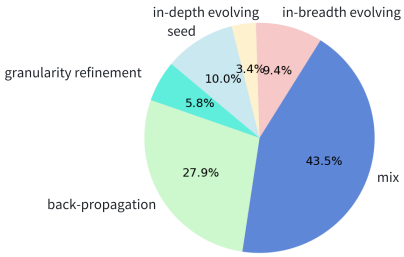

Figure 8: Distribution ratio of response-dependent perspectives.

Method	PandaLM Prompt	JudgeLM Prompt	CoT	USR Prompt	MAD-Eval
Arena Chatbot					
Acc	0.3975	0.3407	0.3492	-	**0.4137**
F1	0.3235	0.1980	0.2147	-	**0.3884**
ρ	0.1426	0.0359	0.0749	-	**0.1558**
τ	0.1343	0.0339	0.0706	-	**0.1422**
MT-bench					
Acc	0.4930	0.3970	0.4080	-	**0.5530**
F1	0.4243	0.2541	0.2766	-	**0.5001**
ρ	0.2900	0.1319	0.1632	-	**0.3995**
τ	0.2734	0.1247	0.1542	-	**0.3735**
PandaLM					
Acc	0.5746	0.4775	0.4815	-	**0.6567**
F1	0.5198	0.3442	0.3699	-	**0.6423**
ρ	0.2827	0.0381	0.0511	-	**0.4050**
τ	0.2724	0.0367	0.0493	-	**0.3917**
Topical-Chat					
MSE	-	-	-	2.4806	**1.5639**
ρ	-	-	-	0.1975	**0.3149**
τ	-	-	-	0.1723	**0.2770**

Table 9: Results on Qwen3-0.6B.

1 **Seed** PandaLM Prompt (Below are two responses for a given task. The task is defined by the Instruction with an Input that provides further context. Evaluate ...)

2 **Back Propagation** The assistant's response provides additional valuable insights or explanations that enhance the overall usefulness and depth of the answer.

3 **Back Propagation** The assistant's response indicates in - depth understanding of the question, closely focusing on the question's context, user needs, and the topic.

4 **Mix** Your evaluation should focus solely on the completeness and adequacy of the information provided in each response.

5 **Back Propagation** The assistant's response offers additional valuable insights, background, or explanations, enhancing the practicality and depth of the answer.

Figure 9: The prompts used for the top five perspectives in Figure 5.