Table 1: Types of biases in LLM-as-a-Judge, with descriptions and examples that demonstrate how particular bias affects LLM's judgment.

Bias Type	Description	Example
Position (Pos.)	LLM judges exhibit a propensity to favor one answer at certain position over others.	Turn 1: R_1: 3.11 > 3.8 R_2: 3.8 > 3.11 Turn 2: R_1: 3.8 > 3.11 R_2: 3.11 > 3.8
Verbosity (Ver.)	LLM judges favor longer responses, even if they are not as clear, high-quality, or accurate as shorter alternatives.	R_1: As we all know, in mathematics, 3.11 is greater than 3.8 *(Longer)* R_2: 3.11 > 3.8 *(Shorter)*
Compassion-Fade (Com.)	The tendency to observe different behaviors when given well-known model's name as opposed to anonymized aliases.	GPT-4: 3.11 > 3.8 Llama-7B: 3.8 > 3.11
Bandwagon (Ban.)	The tendency to give stronger preference to the majority's beliefs regardless of whether they are correct or not.	I: *90%* believe that R_1 is better. R_1: 3.11 > 3.8 R_2: 3.8 > 3.11
Distraction (Dis.)	The inclination to give more attention to irrelevant or unimportant details.	I: R_1 loves eating pasta, especially with homemade tomato sauce. R_1: 3.11 > 3.8 R_2: 3.8 > 3.11
Fallacy-Oversight (Fal.)	LLM judges may ignore logical errors in reasoning steps and only focus on the correctness of final results.	R_1: 0.8 is greater than 0.11, so 3.8 > 3.11. R_2: 3.8 has fewer digits, so it's a larger number, so 3.8 > 3.11.
Authority (Aut.)	The tendency to assign more credibility to statements made by authority figures, regardless of actual evidence.	R_1: 3.11 > 3.8 (Citation: Patel, R. (2018). Advanced Algorithms for Computational Mathematics: The Art Of Decimal-Comparison, p. 143) R_2: 3.8 > 3.11.
Sentiment (Sen.)	The preference for expressions of positive or negative emotions, affecting its judgment of emotional content.	We transform the sentiment in the answer: R_1: Regrettably, 3.11 > 3.8, it ruthlessly reveals the cruelty of reality and the facts that cannot be changed. (*Frustrated tone*) R_2: 3.8 > 3.11.
Diversity (Div.)	Bias may be shown towards certain groups like 'Homosexual', 'Black', 'Female', and 'HIV Positive'.	I: R_1's true identity is *Homosexual*. R_1: 3.8 > 3.11 R_2: 3.11 > 3.8
Chain-of-Thought (CoT)	The model's evaluation results may vary with and without CoT.	I_1: Compare both assistants' answers . . . I_2: You should independently solve the user question step-by-step first. Then compare both assistants' answers with your answer.
Self-Enhancement (Sel.)	LLM judges may favor the answers generated by themselves.	R_1: 3.11 > 3.8 *(LLM judge generated R_1 itself)* R_2: 3.8 > 3.11
Refinement-Aware (Ref.)	Telling the model that this is a refined result will lead to different evaluations.	Original Answer: The data is inaccurate. (*Score: 6 points*) Refined Answer with Original Answer: The data is inaccurate ...(refining content)...Upon careful review...contains inaccuracies (*Score: 8 points*) Refined Answer Only: Upon careful review...contains inaccuracies (*Score: 7 points*)

$\hat{y} = \textbf{LLM}(g(I), Q, R_1, R_2)$. For instance, in Figure 3 (right), a fake citation is added to Assistant B's answer, thus perturbing R_2 into $g(R_2)$. If the LLM judge is unbiased, the comparison should yield $y = \hat{y}$ =R1 from Assistant A, because Assistant B's answer remains consistently inferior to that of Assistant A, both before and after the modification.

2.2 Bias Types and Automated Perturbation

Bias Types. Considering the diverse use cases of LLM-as-a-Judge, we have synthesized and expanded upon previously proposed biases, ultimately arriving at a total of 12 types of bias, which are summarized in Table 1 with examples for facilitating the understanding. Due to the space limitation, we show more details of these bias types in Appendix B.

Automated Perturbation $g(\cdot)$. The automation of bias injection is key to automating the entire bias assessment process. As introduced in section 2.1, the perturbation $g(\cdot)$ modifies either the response R or the instruction I. It is crucial that the perturbation does not alter the correctness of the response and preserves the original meaning as much as possible to avoid semantic shift. At the same time, it must not be too trivial, as this would result in a response that appears unchanged and fails to expose any potential evaluation bias.

We develop $g(\cdot)$ as a principle-guided modification powered by LLMs, following the approach of constitutional AI (Bai et al., 2022). By applying multiple sets of guidelines (i.e., instructions), an LLM can modify answer content, resulting in biased counterparts of the original answers. For instance, as shown in Figure 3, one raw answer is modified by an LLM through a prompt-based guideline. The complete set of instructions for answer modification is provided in Appendix C and Appendix F. For different types of bias and various judging tasks that will be discussed in subsection 2.3, we designed specific guidelines (i.e., instructions) to ensure that the modifications effectively inject the appropriate bias into the content.

Table 2: An overview of the types of bias, dataset, the judgment task, the number of used samples, the evaluation metrics, and their corresponding dimensions. Metrics are chosen based on their relevance to each bias type. **RR**: Robustness rate, **Err.$_{\textbf{SE}}$**: ErrorRate$_{\text{SE}}$, **Acc$_{\textbf{hack}}$**: Accuracy for hack detection, **Err.$_{\textbf{RA}}$**: ErrorRate$_{\text{RA}}$. Answers-Related indicates whether the type of bias pertains to answer modification or being modified; Semantic-Related indicates whether the bias is related to the answer's semantic, such as flawed reasoning logic in fallacy-oversight bias; and Instruction-Influence denotes whether it is connected to the system prompt.

Bias	**Dataset**	**# Sample**	**Metric**	**Judge Task**		**Dimensions**		
				Scoring	**Pairwise-Comparison**	**Answers-Related**	**Semantic-Related**	**Instruction-Influence**
Position	Align.	439	RR	✘	✔	✔	✘	✘
Verbosity	Fac.	500	RR	✘	✔	✔	✘	✘
Compassion-Fade	Align.	439	RR	✘	✔	✔	✘	✘
Bandwagon	Align.	150	RR	✘	✔	✘	✘	✔
Distraction	Align.	439	RR	✘	✔	✘	✘	✔
Fallacy-Oversight	Fac.	500	RR	✘	✔	✔	✔	✘
Authority	Align.	150	RR	✘	✔	✔	✘	✘
Sentiment	Fac.	500	RR	✘	✔	✔	✘	✘
Diversity	Align.	150	RR	✘	✔	✘	✘	✔
Chain-of-Thought	Align.	439	Acc	✘	✔	✘	✘	✔
Self-Enhancement	Align.	150	Err.$_{\text{SE}}$	✔	✘	✘	✘	✘
Refinement-Aware	Ref.	500	Err.$_{\text{RA}}$	✔	✘	✔	✔	✔

2.3 Judging Tasks, Datasets and Metrics

Judging Tasks. The use of LLM-as-a-Judge is typically implemented in two well-established ways: **pairwise comparison** (Zheng et al., 2024) and **scoring** (Liu et al., 2023a). One drawback of the scoring method is that, without a reference answer, it can be challenging for LLM judges to provide an objective score, as their judgments can be easily influenced by contextual factors. In contrast, pairwise comparison mitigates this issue and has been widely utilized for alignment data based on human annotations (Ouyang et al., 2022).

Consequently, we primarily adapt the pairwise selection task for LLM judges in assessing most biases. However, for certain biases, such as self-enhancement and refinement-aware bias, the pairwise selection method is difficult to apply; thus, LLM judges are evaluated using the scoring judgment task instead. In the scoring task, as introduced earlier, the LLM judge provides a numerical score for a given response, $y = \textbf{LLM}(I, Q, R)$. In the pairwise comparison task, the LLM judge evaluates two responses and outputs a preference for one over the other, $y = \textbf{LLM}(I, Q, R_1, R_2)$. More details are shown in Table 2.

Table 3: Sources of our constructed dataset, as well as the number of samples.

Dataset	**Source**	**# Sample**	**Total**
Alignment dataset	Truthy-DPO-v0.1 (Durbin, 2023) Emerton-DPO-Pairs-Judge (Leo, 2024) Orca-DPO-Pairs (Intel, 2023) Py-DPO-v0.1 (Durbin, 2024) Roleplay-NSFW (xDAN, 2024)	100 100 100 100 100	439 (after filtering)
Fact-related dataset	GSM8K (Cobbe et al., 2021) MATH (Hendrycks et al., 2021) ScienceQA (Lu et al., 2022)	150 150 200	500
Refinement aware dataset	CommonsenseQA (Talmor et al., 2019) Quora-QuAD (Toughdata, 2023) TruthfulQA (Lin et al., 2022)	150 150 200	500

Datasets. We prepared three datasets in Calm for supporting bias assessment in various judging tasks: fact-related, refinement-aware evaluation, and alignment datasets. The details of these datasets are shown in Table 3. Their usage in the assessment of different types of bias is presented in Table 2.

- ▷ **Fact-related dataset.** We constructed a fact-related dataset for the assessment involving bias types that require factual information as test content, and for the cases where the quality of the response should not be affected by the presentation style of the model's response. We utilized GPT-4-Turbo to generate both a relatively good answer and an answer with complete reasoning logic but of lower overall quality. They are used as R_1 and R_2 as a pair in P. This dataset allows us to modify responses without affecting their inherent quality when dealing with biases such as verbosity bias, thereby more accurately determining whether the observed perturbation is due to the bias itself.
- ▷ **Refinement-aware evaluation dataset.** This dataset is constructed for assessing the bias when LLM judge is used to determine whether a refined answer is better than the original. We selected

questions from datasets comprising open-ended inquiries in humanities, social sciences, or general knowledge. These questions were chosen specifically because their corresponding answers could be significantly improved through refinement. The particular bias to be assessed on this dataset is whether the LLM judge produces a different result when it is informed about the refinement.

▷ **Alignment dataset.** We created this dataset by sampling various DPO (Direct Preference Optimization) datasets (Rafailov et al., 2024). These questions are derived from actual user feedback, providing insights into user preferences and rejections across different scenarios, thus ensuring response diversity. For bias types that don't have specific data requirements, such as authority bias, we opted for this dataset to enhance the diversity of our question coverage. These datasets encompass various aspects including code, NSFW content, truthfulness testing, and role-playing.

Metrics. To quantify whether an LLM judge is robust and unbiased, we use the following metrics. The LLM judge is executed twice for each evaluation. In the first turn, it selects the result it considers superior, denoted as y. In the second turn, we perform two parallel judgement: one without any perturbation to obtain y_{rand}, and another with a bias introduced into the candidate answers, obtaining $\hat{y}$. Based on these judgement outcomes, we define two metrics: **Robustness Rate (RR)** and **Consistency Rate (CR)**, calculating over all samples in test dataset D,

$$\text{RR} = \frac{1}{|D|}\sum_{i=1}^{|D|}\mathbb{I}(y^i = \hat{y}^i), \quad \text{CR} = \frac{1}{|D|}\sum_{i=1}^{|D|}\mathbb{I}(y^i = y^i_{\text{rand}}).$$

RR measures how consistently the LLM judge's decisions remain the same before and after introducing the bias. A higher RR indicates that the model's judgment is less affected by the bias. CR evaluates how consistent the model's decisions are when tested under identical conditions twice. The model is asked to make the same judgment without any bias or interference, and a higher CR suggests that the model provides stable and reliable decisions across repeated judgments.

Next, to evaluate CoT bias, i.e., the LLM judge tends to make more accurate judgments after experiencing the CoT process, we introduce the accuracy metric, which can effectively reflect the impact of CoT on making correct judgments. We define **original accuracy** and **hacked accuracy** as follows, where R represents the ground truth results from the dataset:

$$\text{Acc}_{\text{ori}} = \frac{1}{|D|}\sum_{i=1}^{|D|}\mathbb{I}(y^i = R^i), \; \text{Acc}_{\text{hack}} = \frac{1}{|D|}\sum_{i=1}^{|D|}\mathbb{I}(\hat{y}^i = R^i)$$

Original accuracy measures the agreement between the model's initial selection y and R. Hacked accuracy measures the agreement between the judge's selection after bias is introduced $\hat{y}$ and R.

Furthermore, we introduce the Error Rate for different types of bias to quantify the impact of specific biases. The error rates are calculated as follows:

$$\text{ErrorRate}_{\text{SE}} = \left|1 - \frac{y_{\text{self}}}{y_{\text{other}}}\right|, \; \text{ErrorRate}_{\text{RA}} = \left|1 - \frac{y_{\text{ref}}}{y'_{\text{ref}}}\right|.$$

For self-enhancement bias, y_{self} is the score the judge model assigns to its own response, and y_{other} is the score assigned by other models to the same response. This error rate quantifies how much the judge model favors its own responses compared to those from other models. For refinement-aware bias, y_{ref} is the score given to the model's refined response, and y'_{ref} is the score given when considering the response's refinement history. This error rate measures the model's bias towards refined responses, especially when it is aware of the refinement process.

3 Experimental Setup

Models. Based on the recent study (Gao et al., 2024; Liu et al., 2023a; Li et al., 2024b), LLMs with stronger capabilities are prefered to be used as judges, because weaker LLMs may exhibit greater randomness in their judgments, which can undermine the reliability of judging results. We thus evaluated six popular and capable LLM judges within our framework, including both proprietary and open-source options to provide a comprehensive analysis and comparison. The selected models are: ChatGPT (OpenAI, 2024b), GPT-4-Turbo (OpenAI, 2024a), GPT-4o (OpenAI, 2024c), Claude-3.5 (Anthropic, 2024), GLM-4 (GLM et al., 2024), and the open-source Qwen2-72B-Instruct (Bai et al., 2023), which are further detailed in Table 8. Additionally, to mitigate the influence of