

LLMs Are Biased Towards Output Formats! Systematically Evaluating and Mitigating Output Format Bias of LLMs

Anonymous ACL submission

Abstract

We present the first systematic evaluation examining format bias in performance of large language models (LLMs). Our approach distinguishes between two categories of an evaluation metric under format constraints to reliably and accurately assess performance: one measures performance when format constraints are adhered to, while the other evaluates performance regardless of constraint adherence. We then define a metric for measuring the format bias of LLMs and establish effective strategies to reduce it. Subsequently, we present our empirical format bias evaluation spanning four commonly used categories—multiple-choice question-answer, wrapping, list, and mapping—covering 15 widely-used formats. Our evaluation on eight generation tasks uncovers significant format bias across state-of-the-art LLMs. We further discover that improving the format-instruction following capabilities of LLMs across formats potentially reduces format bias. Based on our evaluation findings, we study prompting and fine-tuning with synthesized format data techniques to mitigate format bias. Our methods successfully reduce the variance in ChatGPT’s performance among wrapping formats from 235.33 to 0.71 (%²).

1 Introduction

To unlock the full potential of automating real-world applications, state-of-the-art large language models (LLMs) (Brown et al., 2020; Chowdhery et al., 2022; OpenAI, 2022; Touvron et al., 2023) are increasingly leveraged to tailor outputs to specific task formats. This powerful approach has driven advancements across domains including medicine (Thirunavukarasu et al., 2023; Clusmann et al., 2023), data analysis (Cheng et al., 2023; Liu et al., 2023), and even evaluating models themselves (Chiang and Lee, 2023; Chang et al., 2024). Employing LLMs in such applications heavily depends on their format-following capability and delivering high-quality results within formats.

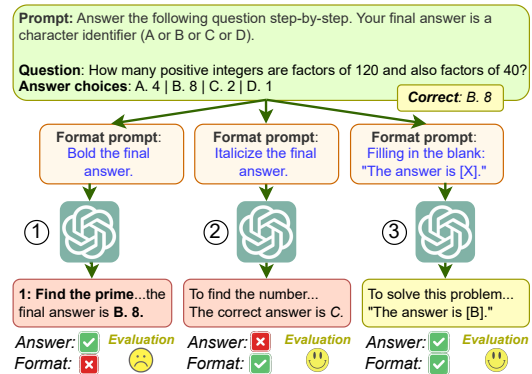


Figure 1: An example from MMLU (Hendrycks et al., 2021) with ChatGPT across different formats. In Case (1), the model can answer the question but fails to bold only the answer, hindering automatic evaluation. In Case (2), the model follows the format but produces an incorrect result. In Case (3), the model yields the correct answer and format. These show bias in ChatGPT’s performance across formats.

While many studies, including those listed above, have utilized LLMs to output in specific formats, understanding their format capabilities is critical yet has received limited attention. Recently, Zhou et al. (2023) and Xia et al. (2024) introduced benchmarks assessing LLM format-following proficiency. However, these studies neglect deeper insights into how these formats impact model performance which is the ultimate concern for practical use. Given the numerous formats recently introduced across tasks and models, assessing this aspect is essential yet challenging. Evaluation can be ambiguous and often overlook cases where models provide correct answers but are formatted wrong (e.g., Case (1) in Fig. 1).

Bridging these gaps, we conduct the first systematic evaluation of the format bias of LLMs. Our study attempts to answer the research questions:

How can we systematically and accurately assess format bias in performance of LLMs, and to what extent are they biased?

To fairly assess bias in model performance across formats, it is crucial to evaluate all scenarios

Section	Categories	Format Types	Generation Tasks
§4.1	MCQ answer	Character identifier; Choice value.	MCQ (MMLU, BBH).
§4.2	Wrapping	Special character; Bolding; Italicizing; Brackets; Parentheses; Placeholder; Quoting.	MCQ (MMLU, BBH), Reasoning (GSM8K), Multi-hop QA (HotpotQA), Narrative QA (FairytaleQA).
§4.3	List	Python list; Bullet list; Special-character list; Newline list.	Order list (SciDocsRR), Unordered list (SemEval2017).
§4.4	Mapping	Python (Json); YAML.	Easy/Medium/Hard Inf. Extraction (SciREX).

Table 1: Format categories and types with corresponding tasks highlighted in blue that are evaluated in this study.

depicted in Fig. 1. Nonetheless, Case (1) is challenging to automatically measure, requiring costly human investigation. Therefore, we propose a reliable estimator for evaluating LLM performance under format constraints without human intervention by considering format-following scores. We start by redefining LLM evaluation metrics into two distinct classes to construct the estimator, as detailed in §2.1. Accordingly, we define a metric to quantify format bias in LLMs and establish criteria for evaluating methods that successfully mitigate this bias (§2.2). Based on these formulations, we present our format evaluation framework, comprising of the widely-utilized categories of multiple-choice question-answer (MCQ; §4.1), wrapping (§4.2), list (§4.3) and mapping formats (§4.4).

Across 15 widely-used formats, our evaluation with zero-shot and zero-shot chain-of-thought prompting (Kojima et al., 2022) on eight question-answering and reasoning tasks (Tab. 1) reveals substantial performance and format-instruction following inequalities. To address this, we examine prompting and fine-tuning using synthesized format data techniques which work for both open and closed-source LLMs. Our study validates that enhancing LLMs’ capabilities to follow format instructions potentially mitigates format bias: (i) Prompting with demonstrations and (ii) Repeating format instructions substantially alleviate this bias. Moreover, we investigate (iii) Synthesizing limited format data based on our evaluation results for fine-tuning. Our proposed approaches significantly decrease ChatGPT (OpenAI, 2022) performance variance across wrapping formats from 235.33 to 0.71 (%²) on MMLU benchmark (Hendrycks et al., 2021). Overall, our evaluation identifies a novel challenge to LLM developments, paving the way for fairer and more robust applications.

2 Output Format Evaluation Framework

2.1 Theoretical Analysis: Format Evaluation

Automatic evaluation of LLMs in question-answering and reasoning tasks mainly relies on

rule-based extraction to identify final answers from generated texts (Guo et al., 2023). Within format constraints, determining the model’s true performance, which is our focus, can be ambiguous and inaccurate, as correct responses might be overlooked due to format discrepancies (e.g., Case (1) in Fig. 1). To address this, we propose redefining these rule-based evaluation metrics to reliably, transparently and accurately measuring the LLM performance given formats restrictions.

Notations. Suppose that we are interested in evaluating an LLM \mathcal{M} on a task T using an evaluation metric E (such as “Accuracy”) under a format constraints C (such as “Bold the final answer.”) on n samples with the ground-truth answers $\{y_1, \dots, y_n\}$ and raw generated answers $\{\hat{y}_1, \dots, \hat{y}_n\}$, where $y_i, \hat{y}_i \in \mathcal{Y} \forall i$ with \mathcal{Y} being the answer token sequence space. We denote F_C as the binary format-following evaluation function of C :

$$F_C(\hat{y}_i) = \begin{cases} 1, & \text{if } \hat{y}_i \text{ satisfies } C. \\ 0, & \text{otherwise.} \end{cases} \quad (1)$$

From Eq. (1), we define the **Format Instruction-following (FI) Score**, denoted as FI_C , as the percentage of generated outputs satisfying C :

$$FI_C = \frac{\sum_{i=1}^n F_C(\hat{y}_i)}{n} \quad (2)$$

Prior studies extensively focus on evaluating FI_C (Zhou et al., 2023; Xia et al., 2024). Our work further targets evaluating the performance of LLMs given the format constraints C . Under C , we denote $Ext_C(\cdot)$ as the rule-based answer extractor to extract the final answer from \hat{y}_i for comparing it with y_i . We define two evaluation scores based on E measuring the performance of \mathcal{M} on T :

Definition 2.1 (Systematic Evaluation Score (SysE)).

$$SysE = \frac{1}{n} \sum_{i=1}^n (E(y_i, Ext_C(\hat{y}_i)) \cdot F_C(\hat{y}_i)) \quad (3)$$

Essentially, $SysE$ quantifies the performance of \mathcal{M} on task T based on the generated answers **that meet the format constraints** C . For example, in Fig. 1, Case (1) yields a $SysE$ score of 0, while Case (3) achieves 1. This also shows that $SysE$ may not accurately reflect the actual performance of \mathcal{M} on T , because $Ext_C()$ may fail to extract the final answers from (correct) answers dissatisfying C (e.g., Case (1) in Fig. 1).

We define the **True Evaluation Score** to address this. Let us assume that we have an oracle extractor function $OracExt_C()$ that can extract the final answer from \hat{y}_i , regardless of whether \hat{y}_i fulfills C .

Definition 2.2 (True Evaluation Score ($TrueE$)).

$$TrueE = \frac{1}{n} \sum_{i=1}^n E(y_i, OracExt_C(\hat{y}_i)) \quad (4)$$

$TrueE$ measures the performance of \mathcal{M} on task T across all generated answers **given the format constraints** C , regardless of constraint satisfaction. For instance, in Fig. 1, both Cases (1) and (3) achieve a true accuracy of 1. This score is crucial for investigating the actual performance of LLMs given the format requirements.

Prior studies do not distinguish between $SysE$ and $TrueE$ properly. This is because in practice, measuring $TrueE$ is challenging, as it requires manual compilation and verification of $OracExt_C()$. Consequently, this measurement often requires human investigation (Lin et al., 2022) or the fine-tuning of evaluation models as scorers (Yang et al., 2024), both of which can be costly and unreliable due to limited data. Finally, while assigning a default value to $Ext_C(\hat{y}_i)$ can temporarily avoid cases where \mathcal{M} does not follow C , this can be an incorrect practice since the default value may not be the actual output of \mathcal{M} .

Nevertheless, $TrueE$ is crucial for a fair investigation of LLM performance bias across formats. Therefore, we propose a simple estimator of $TrueE$, denoted as $EstTrueE$. Let us consider the unit of FI_C is %, we have:

$$EstTrueE = \begin{cases} SysE \cdot \frac{100}{FI_C}, & \text{if } FI_C \neq 0. \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

When $FI_C = 0$, it is impossible to estimate $EstTrueE$. $EstTrueE$ is particularly useful for large-scale experiments since it is fully automatic.

Let us denote the margin of error for $EstTrueE$ as ϵ with a confidence interval $1 - \alpha$ and $S_C = n \cdot$

FI_C as the number of generated answers satisfying C . The reliability of $EstTrueE$ is:

Theorem 2.1 (Reliability of $EstTrueE$). $EstTrueE$ is consistent. Moreover, $EstTrueE$ is reliable if and only if:

$$FI_C \geq \frac{1}{1 + n \cdot \left(\frac{\epsilon}{v \cdot s}\right)^2} \quad (6)$$

Additionally, we have:

$$\lim_{FI_C \rightarrow 100} EstTrueE = TrueE \quad (7)$$

where s^2 is the sample variance of evaluation scores of generated answers satisfying C and $v = t_{\alpha/2, S_C - 1}$ is the critical value from the t-distribution with $S_C - 1$ degrees of freedom and significance level α .

In summary, we have proposed a consistent estimator $EstTrueE$ of the true performance of LLMs measured by metric E under the format constraints (Def. 2.2). This estimator is crucial because it: (1) enables transparent and reliable LLM performance evaluation considering format requirements; (2) facilitates the reliable investigation of performance biases across different formats, which is our focus. Henceforth, unless otherwise specified, $EstTrueE$ is our primary metric for assessing model performance given format conditions. The proof of Thm. 2.1 is in §B.1.

2.2 Theoretical Analysis: Format Bias

This section defines the metric to quantify format bias and outlines the criteria to mitigate such bias.

Bias measurement. Suppose that we are interested in measuring the format bias of the LLM \mathcal{M} in performing the task under k formats $F_o = \{C_1, \dots, C_k\}$. We define a single value $BiasF_o$ measuring such bias as the variance of $EstTrueE$ scores over k formats $\{EstTrueE_1, \dots, EstTrueE_k\}$. Let us denote $\mu_{EstTrueE} = (\sum_{i=1}^k EstTrueE_i) / k$, we have:

$$BiasF_o = \frac{1}{k} \sum_{i=1}^k (EstTrueE_i - \mu_{EstTrueE})^2 \quad (8)$$

Reliability of $BiasF_o$. By Eq. (8), the lower $BiasF_o$ is, the less format- F_o -biased \mathcal{M} is, suggesting a criterion for mitigating output format bias.

However, $BiasF_o$ is an estimator based on the estimators $EstTrueE_i$. Therefore, to enhance the reliability of $BiasF_o$, it is also necessary to improve the reliability of $EstTrueE_i$ by increasing $FI_{C_i} \forall i$ (Thm. 2.1). Therefore, we propose **two necessary criteria for an effective method to mitigate format bias in LLMs**: (i) **Minimize bias metric**: reducing $BiasF_o$, indicating less format- F_o -bias in \mathcal{M} ; (ii) **Increase the format-following scores for all formats**: ensuring the reliability of $BiasF_o$ by increasing the FI scores across all the formats: $\{FI_{C_1}, \dots, FI_{C_k}\}$ (Eq. (2)).

2.3 Formats for Evaluation

We establish 4 ubiquitously used format categories for our bias evaluation, encompassing 17 formats introduced by prior practice, also outlined in Tab. 1:

(i) **Multiple-choice question (MCQ) answer (§4.1)**. where LLMs answer questions by selecting from provided choices, presented as either a (1) **Character identifier** (Robinson and Wingate, 2023); or (2) **Choice value** (Chen et al., 2023).

(ii) **Wrapping (§4.2)**. where LLMs must enclose the final answer within the two characters, which is crucial for automatic evaluation to isolate the final answer from reasoning thoughts. We focus on evaluating 7 widely used wrapping strategies: (1) **Special character** (Gur et al., 2022); (2) **Bolding** (Zhou et al., 2023); (3) **Italicizing** (Zhou et al., 2023); (4) **Double brackets** (Luo et al., 2024); (5) **Double parentheses**; (6) **Placeholder** (Wang et al., 2024); (7) **Quoting** (Zhou et al., 2023).

(iii) **List (§4.3)**. where the output of LLMs is a list of elements. We investigate 4 formats representing lists: (1) **Python list** (Do et al., 2023); (2) **Bullet-point list** (Liu et al., 2024); (3) **List of elements separated by a special character "[SEP]"** (Boucher, 2023); and (4) **List of elements arranged on separate lines** (Mishra, 2023).

(iv) **Mapping (§4.4)**. where LLMs are employed to output dictionaries or maps. We focus on two ubiquitously used mapping structures: (1) **Python dictionary/JSON** (JavaScript Object Notation) (Baumann et al., 2024) and (2) **YAML** (Yet Another Markup Language) (Goel et al., 2023).

Format-instruction following. We introduce Appx.-Alg. 1, a rule-based heuristic to determine the format-instruction following function F_C (Eq. (1)) for our benchmarked formats. It calculates

the binary FI score by verifying that the generated output includes the specified formatting tokens and that the extracted final answer matches the expected type. It is highly extendable to other formats (§A).

3 General Experimental Setups

Benchmarks. For MCQ bias evaluation (§4.1), we select two datasets: **MMLU** (Hendrycks et al., 2021) and **BBH** (Suzgun et al., 2023). For MMLU, we randomly choose 27 subcategories. For BBH, we select the sports_understanding category following Gupta et al. (2024). For wrapping bias assessment (§4.2), in addition to MCQ benchmarks, the following datasets are experimented: **GSM8K** (Cobbe et al., 2021) for reasoning, **FairytaleQA** (Xu et al., 2022a) for narrative comprehension, and **HotpotQA** (Yang et al., 2018a) for multi-hop reasoning. For list bias investigation (§4.3), we use **SciDocsRR** (Muennighoff et al., 2023), a scientific document ranking task as the order list generation task, and **SemEval 2017** (Augenstein et al., 2017a), the keyphrase extraction task as the unordered list generation. For mapping bias examination (§4.4), we utilize a document-level information extraction task named **SciREX** (Jain et al., 2020a) by synthesizing three extraction difficulty levels: easy (extracting from 1 sentence for 1 category), medium (3 sentences, 2 categories), and hard (5 sentences, 4 categories). For all benchmarks except MCQ, we sample 200 points for evaluation (Bai et al., 2023).

Metrics. Following our discussion in §2.1, we disentangle **Accuracy (Acc)** for MMLU and BBH (Guo et al., 2023); **F1** for GSM8K, HotpotQA, FairytaleQA; and **Mean Average Precision (MAP)** for SciDocsRR (Muennighoff et al., 2023) and we report the metrics $EstTrueAcc$, $EstTrueF1$, $EstTrueMAP$ (Eq. (5)) in the main text. For the reliability of $EstTrue*$, we set $\alpha = \epsilon = 5\%$.

Models. We select both open- and closed-source LLMs for our evaluation: **Gemma-7B-it** (Team et al., 2024) and **Mistral-7B-it-v0.2** (Jiang et al., 2023) for open-source as they are among state-of-the-art open-source LLMs; **ChatGPT (gpt-3.5-turbo-0125)** for closed-source as this premier chatbot possesses superior instruction-following ability. Our purpose is not to reproduce the models' performance, but to show the bias.

Prompting baselines. Our focus is on two widely used prompting baselines: (1) **Zero-shot**

(ZS) prompting and (2) **Zero-shot Chain-of-Thought (ZS-CoT)** prompting (Kojima et al., 2022). For the ZS baseline, we instruct LLMs to answer the question with the prompt “Answer the following question...” followed by the suffix “without any explanation”. For ZS-CoT, we use the suffix “step-by-step” instead. For the ZS-CoT experiments in Sections 4.1, 4.3 and 4.4, LLMs are instructed to wrap the final answer by “<ANSWER>” and “</ANSWER>” tokens to distinctly isolate it from the reasoning chains (see Tab. 2 for the wrapping instruction). We use this wrapping method since our experiment in §4.2 shows that it achieves the highest instruction-following score on average across LLMs. Detailed prompts are provided in §E. We average the performance under two prompting methods to report in the main text.

4 Format Evaluation Experiments

Overall, we find that: (1) Models show substantial format-following bias across formats for all benchmarks; (2) For all models and datasets, significant performance bias exists across formats; (3) 77.67% of the $EstTrue$ results are reliable, with 16/24 for MCQ, 169/210 for wrapping, 35/48 for list, and 27/36 for mapping formats. We dive into (2) for every format as it is our main focus, (1, 3) are discussed in detail in Appendices C.1 to C.4.

4.1 Experiments on MCQ Format

Setup. We investigate the bias of LLMs towards different MCQ output formats. We assess two formats as introduced in §2.3: (1) Character identifier and (2) Choice value. For example, if the choice is “[A. Yes, B. No]”, then the character identifier can be “A/B”, while the choice value can be “Yes/No”. We exclude the format combining the character identifier and choice value (such as “A. Yes”) from our evaluation because instructing LLMs to output this format can be non-trivial and require manual effort to craft instructions tailored for different models. To ensure that LLMs understand the “Character identifier” and “Choice value” as we expect, we add a contrastive format requirement to the prompts (e.g., “without any textual description” for the “Character identifier” prompts).

Results. Fig. 2 provides a synopsis of our evaluation results, with numerical values shown in Appx.-Tab. 3. From Fig. 2-left, we observe that Mistral possesses the highest disparity between the two MCQ answer formats, with 58.69% accuracy on

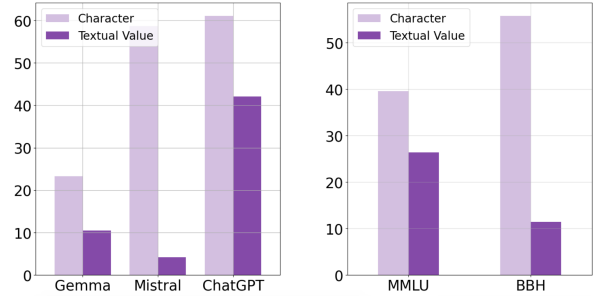


Figure 2: Avg estimated true accuracy (§2.1) results of MCQ benchmarks on models (left) and datasets (right) showing performance bias of LLMs across formats.

average for character and only 4.22% for textual value. Additionally, despite ChatGPT often being regarded as one of the most robust LLMs, it shows a significant difference between the two formats (19.03%). Overall, LLMs are heavily biased towards outputting character identifiers. Requiring them to generate the choice’s value causes notable performance drops of 28.76% on average.

From Fig. 2-right, we notice that the models exhibit higher bias on BBH, which appears to be an easier benchmark than MMLU. We attribute this to the small size of BBH, which makes the performance more sensitive to variations in format.

Why such bias? We hypothesize the root cause of the significant performance bias across different formats is the **format token bias** of LLMs. The non-uniform distribution of FI scores among formats suggests that the models assign probabilities to format instructions differently based on their training data. This leads to varying prior assignments of probabilities to specific tokens, causing final predictions non-uniformly distributed across formats. This hypothesis is supported by our simple fine-tuning with formatted data, which familiarizes LLMs with format instructions relatively equally leading to a drastic format bias reduction (§5). This emphasizes the necessity of more research in fine-tuning LLMs to reduce format bias and raises concerns about the reliability and reproducibility of recent studies using varied formats.

4.2 Experiments on Wrapping Format

Setup. We study LLM bias towards 7 wrapping methods: (1) Special character; (2) Bolding; (3) Italicizing; (4) Brackets; (5) Parentheses; (6) Placeholder; (7) Quoting, detailed in Tab. 2. We evaluate LLM performance across formats on the MMLU, BBH, GSM8K, FairytaleQA, and HotpotQA.

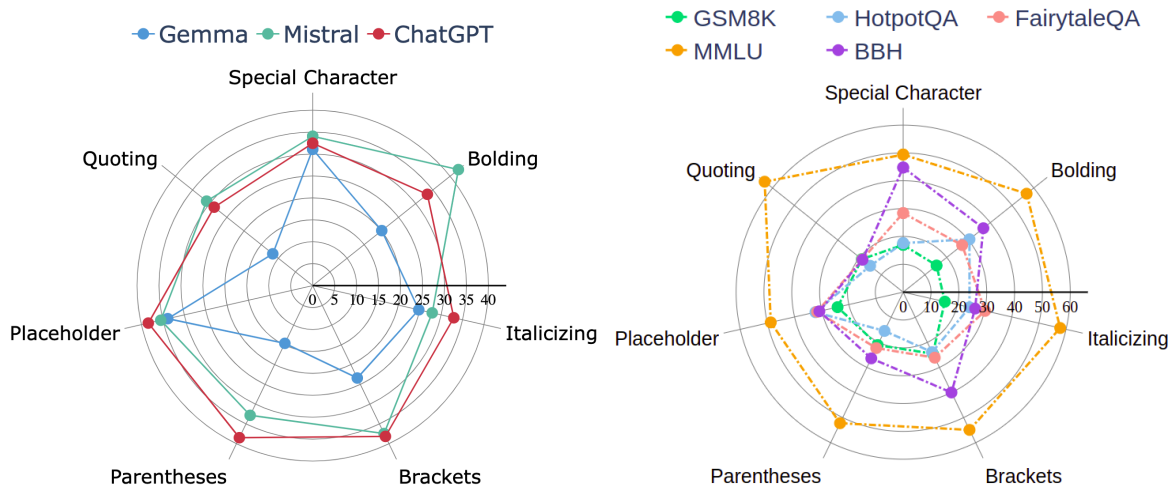


Figure 3: Average estimated true Accuracy (MCQ) and F1 (GSM8K, HotpotQA, FairytaleQA) scores (§2.1) on models (left) and benchmarks (right) showing performance bias of LLMs across 7 widely used wrapping methods.

Wrapping type	(start, end)	Prompt: Wrap your final answer...
Special char.	(<ANSWER>, </ANSWER>)	by <ANSWER> and </ANSWER>.
Bolding	(**, **)	in bold by enclosing it with double asterisks.
Italicizing	(*, *)	in italics by enclosing it with single asterisks.
Brackets	((,))	using double square brackets.
Parentheses	((,))	using double parentheses.
Placeholder	None	by filling in the placeholder below: "So the answer is: [placeholder]"
Quoting	("", "")	using triple double-quotation marks.

Table 2: Wrapping “start” and “end” tokens with instructions.

Results. Fig. 3 outlines an overview of our evaluation outcomes with results in Appx.-Tab. 7. From Fig. 3-left, we see that Gemma exhibits the highest bias towards different formats with a $BiasF_o$ value (Eq. (8) (variance) of $56.33\%^2$, while ChatGPT performs the best with only $12.26\%^2$. Notably, for “Quoting” and “Parenthesis”, the Gemma follows instructions only about 0 – 4% yielding nearly zero performance, highlighting its critical weaknesses. Among the 7 formats, “Placeholder” (35.92%) proves to be the most effective wrapping output format, while “Quoting” (23.74%), “Parenthesis” (28.64%) are among those that achieve the lowest performance.

From Fig. 3-right, models exhibit bias across all tasks, with the lowest on GSM8K ($12.97\%^2$) possibly because the models were trained on (part of) it, and the highest on BBH ($70.16\%^2$), the challenging task without train data. This demonstrates the pervasive presence of wrapping bias in LLMs.

Why such bias? The **format token bias** of LLMs as explained in §4.1 is also our hypothesis. Specifically, we found the low performance of the “Quoting” and “Parenthesis” because, in generation tasks, models often wrap (via quoting/paren-

thesizing) not only the final answer, as instructed, but also parts of the context (e.g., ““The answer is 3.””), leading to poor F1 scores. Moreover, Gemma completely ignores the above format instructions, resulting in 0% FI scores, which also contribute to the low average estimated F1 scores. These strongly indicate the presence of format token bias in LLMs.

4.3 Experiments on List Format

Setup. We explore the bias of LLMs in generating lists following four widely-used formats: (1) Python list, (2) Bullet-point list, (3) Character-separated list, and (4) Newline-separated list. We evaluate the models on two list generation tasks: (i) *Unordered list*, using the keyphrase extraction task on the SemEval 2017 dataset, and (ii) *Ordered list*, using the document ranking problem on the SciDocsRR task.

Results. Fig. 4 displays the key findings of our evaluation across models and datasets with numerical results in Appx.-Tab. 11. From Fig. 4-left, we notice that Mistral exhibits the most bias, with the $BiasF_o$ value (Eq. (8)) of $353.80\%^2$. In contrast, ChatGPT and Gemma show much lower bias, with values of $7.08\%^2$ and $1.32\%^2$, respectively. Of the four formats, the “Python” and “Newline-separated” formats yield the highest performance, likely due to models trained extensively on code data. Conversely, the “Bullet-point list” format results in the lowest performance, particularly for Mistral, highlighting the inherent bias for such formats.

The performance bias is regardless of the task as plotted in Fig. 4-right, with the highest $BiasF_o$

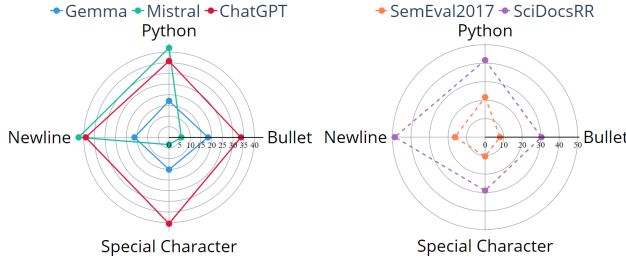


Figure 4: Avg *EstTrueF1* (SemEval2017) and *EstTrueMAP* (SciDocsRR) (§2.1) on models (left) and benchmarks (right) showing performance difference of LLMs across 4 widely used list formats.

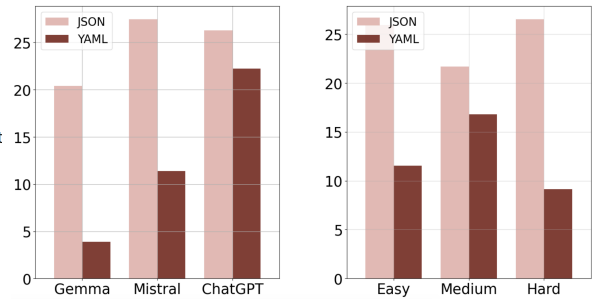


Figure 5: Avg estimated true F1 scores (§2.1) on models (left) and benchmarks (right) showing performance bias of LLMs across 2 widely used mapping formats.

value of 67.07%² on the order list generation task SciDocsRR, and significantly lower (27.58%²) on SemEval2017 task. The high bias in the SciDocsRR task is because Mistral and Gemma mostly failed to perform this task following the “Bullet” and “Special character” list formats while excelling in solving it following the other formats.

Why such bias? We attribute the bias to the **format token bias** (§4.1). Since the models were extensively trained on code data, they excel in solving code-related instructions. In contrast, “Bullet-point” and “Special character” lists are much less common. One interesting case is Gemma where it performed worse on generating “Python” lists compared to “Bullet-point” lists. Our analysis suggests that Gemma misinterprets the format instruction as a coding request, generating Python code programs instead of an answer in a Python list, suggesting Gemma was predominantly trained on code data.

4.4 Experiments on Mapping Format

Setup. We examine the performance bias of LLMs on two mapping formats as discussed in §2: (1) Python dictionary/JSON; (2) YAML. We preprocess the SciREX task (Jain et al., 2020a) as described in §3 into three extraction levels: (i) Easy (1 sentence, “Task” category); (2) Medium (3 sentences, “Task, Method”); (3) Hard (5 sentences, “Task, Method, Material, Metric” categories).

Results. Fig. 5 illustrates a summary of our evaluation with numerical details in Appdx.-Tab. 15. From Fig. 5-left, Gemma is the most biased, with a performance gap of 16.51% between the two formats, followed by Mistral with a 16.07% gap. ChatGPT, however, is relatively robust against format variations, exhibiting a gap of only 4.05%. On average, JSON performs significantly better than YAML for mapping, likely because more JSON

data is used to train models due to its popularity.

From Fig. 5-right, extracting 4 categories in the Hard task shows the largest performance gap between mapping formats. Surprisingly, the Medium task displays the least bias, likely because models perform best in this task.

Why such bias? The bias is attributed to the **format token bias** (§4.1). While Mistral excels in generating JSON, it and Gemma struggle with YAML. Even successfully generating YAML output, Mistral and Gemma frequently introduce noisy information (88%-65% for Mistral with and without CoT, 98%-79% for Gemma) in the response (e.g., a key “Task” should have multiple values, Mistral generates multiple key-value pairs instead e.g., “Task_1:Training . . . Task_2: . . .”), resulting in poor overall performance.

5 Mitigating Format Bias

Three primary streams of techniques have been widely studied to tackle biases in LMs: (1) Prompting (Xu et al., 2024; Macedo et al., 2024); (2) Calibration (Roelofs et al., 2022; Li et al., 2024); and (3) Fine-tuning (Schick et al., 2021; Ghaddar et al., 2021). While calibration techniques can only be used for white-box models, prompting and fine-tuning can be applied for both black-box (via API) and white-box ones. Therefore, we explore prompting and fine-tuning techniques to reduce format bias. We target mitigating the format bias of **ChatGPT**, the strongest model that we benchmarked, on **MMLU**. We aim to reduce the **wrapping** bias (§4.2) due to resource limits, but our methods can be generalized to any model and format.

Demonstration(s) reduce(s) format bias. As discussed in §4.1, LLMs show bias across formats possibly because of the token bias issue. This causes LLMs to non-uniformly comprehend the

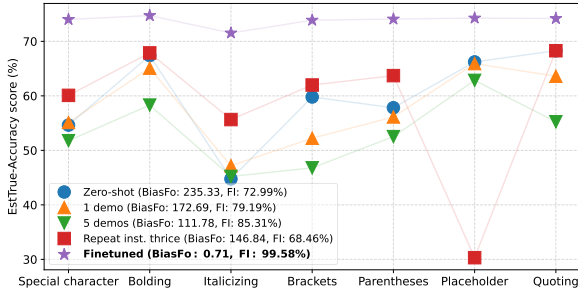


Figure 6: More demonstrations and repeating format instructions mitigate format bias. Finetuning mostly eliminates the format bias. The performance is reported using ChatGPT on MMLU (Appx.-Tab. 19 for num. results).

format instructions. To address this, we examine whether demonstrations with formats can reduce such bias, as they are believed to enhance LLM understanding of the task patterns (Xie et al., 2022). Particularly, for each wrapping format in §4.2, we select 1 and 5 random samples from the auxiliary train data of MMLU and manually format the answers as demonstrations. The results are depicted in Fig. 6. Firstly, incorporating demonstrations typically enhances the FI scores (i) (from 72.99% to 79.19% and 85.31%) of the model, with five demonstrations yielding the most. Secondly, we observe a notable decrease in the $BiasF_o$ score (ii) upon supplementing demonstrations. From (i), (ii) and §2.2, we conclude integrating demonstrations mitigates format bias.

Repeating format instructions reduces format bias. We found that repeating instructions generally increases FI scores (i) across most formats except “Placeholder”, which can consequently lessen the mode’s token bias towards format instructions (§4.1). Hence, we examined the potential of this technique to mitigate format bias. By repeating the wrapping instructions of ChatGPT thrice, we observed a decrease in the $BiasF_o$ (ii) score presented in Fig. 6. Combining (i) and (ii) suggests that this strategy reduces format bias. For “Placeholder,” human investigation reveals that multiple placeholder instructions cause ChatGPT confused about where the placeholder is, making it frequently ignore the format instruction.

Fine-tuning with additional format data can eliminate format bias. We hypothesize that completely solving the format token bias problem of LLMs necessitates finetuning them on format data so that they are familiar with tokens in format instructions evenly. We propose a simple data syn-

thesis strategy for finetuning LLMs: we sample a small set of training data for all evaluated formats, with ratios inversely proportional to their **systematic evaluation scores** (§2.1). We chose $SysE$ scores over the $EstTrueE$ because they reflect the current model performance. Practically, based on ChatGPT’s zero-shot systematic performance on MMLU colored in blue in Appx.-Tab. 7, we approximate the formats’ performance ratios as “1, 1, $\frac{1}{2}$, $\frac{1}{2}$, $\frac{1}{3}$, 1, $\frac{1}{3}$ ” from left-to-right, resulting in training data ratios of formats of “1, 1, 2, 2, 3, 1, 3”. We then preprocess the MMLU auxiliary training data according to these ratios, scaled by 500 (6500 samples total), and train ChatGPT on this dataset. The finetuned results are plotted in Fig. 6. Firstly, after finetuning, the average FI score across all formats is nearly perfect at 99.58% (ii). Secondly, the $BiasF_o$ score is significantly reduced from 235.33%² to 0.71%² (ii). These (i) and (ii) indicate finetuning largely eliminates format bias.

6 Related Works

LLMs have shown remarkable proficiency in formatting outputs to meet human expectations. Such formats include markdown for lists and pointers (Achiam et al., 2023), code blocks (Gur et al., 2022), and integrate tags, or LaTeX for scientific texts (Singh et al., 2023; Wang et al., 2024). Given the rising importance of formatting capabilities in LLMs, recently, format-following benchmarks have been developed for assessing LLMs’ adherence to specified formats (Zhou et al., 2023; Xia et al., 2024; Chen et al., 2024; Macedo et al., 2024; Liu et al., 2024). However, these studies only evaluate format-instruction following capabilities. Our research further assesses LLM performance across different formats, uncovering significant format bias in various tasks and models.

7 Conclusions

We introduce the pioneering systematic investigation of format bias in LLM performance, revealing significant biases across widely used formats for all models and benchmarks. Our method involves developing metrics to assess this bias and establishing criteria for effective mitigation. We then introduce prompting and fine-tuning techniques to alleviate format bias based on our evaluation findings. Our work aims to sharpen the focus of future LLM research toward fairer and more robust development.

625
626
627
628
629
630
631
632
633
634
635
636
637
638
639
640
641
642
643
644
645
646
647
648
649
650
651
652
653
654
655
656
657
658
659
660
661
662
663
664
665
666
667
668
669
670
671

Limitations

Our study has several limitations. Firstly, the metrics $EstTrue$ and $BiasF_o$ proposed in §2.1 and §2.2 are estimators. As discussed, determining $TrueE$ (Eq. (4)) is infeasible, especially for large-scale experiments across various models and datasets. Achieving this would require extensive fine-tuning and comprehensive human evaluations, both prohibitively expensive and impractical in many scenarios. Our proposed metrics $EstTrue$ and $BiasF_o$ are handy for large-scale experiments with multiple models and datasets due to their fully automatic nature. We further propose Thm. 2.1 to validate the reliability of $TrueE$ statistically. Secondly, our empirical evaluation of format bias is restricted to specific datasets, formats, and models due to computational and budget constraints. Expanding our experiments for more formats, larger-scale datasets, and additional task categories, potentially hinders further insights. Our study underscores the importance of future research dedicated to quantifying and mitigating format bias.

Ethical Considerations

Our work uncovers significant format bias in LLMs, raising concerns regarding fairness and potential discrimination in real-world applications.

Bias and fairness. Format bias in LLMs can result in unfair treatment, especially in tasks where multiple possible formats can be used. Our research suggests ways to identify and mitigate format bias, aiming for fairer and more equitable LLM applications.

Societal impact. Format bias in LLMs has the potential to disproportionately impact specific populations, as different demographics may have preferences for different communication formats. Further research is essential to fully understand its societal implications and ensure fairness across diverse demographics.

References

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

Takeshi Amemiya. 1985. *Advanced econometrics*. Harvard university press.

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017a. [SemEval 2017 task 10: ScienceIE - extracting keyphrases and relations from scientific publications](#). In *Proceedings of the 11th International Workshop on Semantic Evaluation (SemEval-2017)*, pages 546–555, Vancouver, Canada. Association for Computational Linguistics. 672
673
674
675
676
677
678
679

Isabelle Augenstein, Mrinal Das, Sebastian Riedel, Lakshmi Vikraman, and Andrew McCallum. 2017b. [Semeval 2017 task 10: Scienceie - extracting keyphrases and relations from scientific publications](#). *CoRR*, abs/1704.02853. 680
681
682
683
684

Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, et al. 2023. Longbench: A bilingual, multitask benchmark for long context understanding. *arXiv preprint arXiv:2308.14508*. 685
686
687
688
689

Nick Baumann, Alexander Brinkmann, and Christian Bizer. 2024. Using llms for the extraction and normalization of product attribute values. *arXiv preprint arXiv:2403.02130*. 690
691
692
693

Ayham Boucher. 2023. [Llm based context splitter for large documents](#). 694
695

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel Ziegler, Jeffrey Wu, Clemens Winter, Chris Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. [Language models are few-shot learners](#). In *Advances in Neural Information Processing Systems*, volume 33, pages 1877–1901. Curran Associates, Inc. 696
697
698
699
700
701
702
703
704
705
706
707
708
709

Yupeng Chang, Xu Wang, Jindong Wang, Yuan Wu, Linyi Yang, Kaijie Zhu, Hao Chen, Xiaoyuan Yi, Cunxiang Wang, Yidong Wang, et al. 2024. A survey on evaluation of large language models. *ACM Transactions on Intelligent Systems and Technology*, 15(3):1–45. 710
711
712
713
714
715

Xinyun Chen, Renat Aksitov, Uri Alon, Jie Ren, Kefan Xiao, Pengcheng Yin, Sushant Prakash, Charles Sutton, Xuezhi Wang, and Denny Zhou. 2023. Universal self-consistency for large language model generation. *arXiv preprint arXiv:2311.17311*. 716
717
718
719
720

Yihan Chen, Benfeng Xu, Quan Wang, Yi Liu, and Zhendong Mao. 2024. Benchmarking large language models on controllable generation under diversified instructions. *arXiv preprint arXiv:2401.00690*. 721
722
723
724

Liyang Cheng, Xingxuan Li, and Lidong Bing. 2023. [Is GPT-4 a good data analyst?](#) In *Findings of the Association for Computational Linguistics: EMNLP* 725
726
727

728	2023, pages 9496–9514, Singapore. Association for Computational Linguistics.	
729		
730	Cheng-Han Chiang and Hung-yi Lee. 2023. Can large language models be an alternative to human evaluations? In <i>Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 15607–15631, Toronto, Canada. Association for Computational Linguistics.	
731		
732		
733		
734		
735		
736	Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, Parker Schuh, Kensen Shi, Sasha Tsvyashchenko, Joshua Maynez, Abhishek Rao, Parker Barnes, Yi Tay, Noam M. Shazeer, Vinodkumar Prabhakaran, Emily Reif, Nan Du, Benton C. Hutchinson, Reiner Pope, James Bradbury, Jacob Austin, Michael Isard, Guy Gur-Ari, Pengcheng Yin, Toju Duke, Anselm Levskaya, Sanjay Ghemawat, Sunipa Dev, Henryk Michalewski, Xavier García, Vedant Misra, Kevin Robinson, Liam Fedus, Denny Zhou, Daphne Ippolito, David Luan, Hyeontaek Lim, Barret Zoph, Alexander Spiridonov, Ryan Sepassi, David Dohan, Shivani Agrawal, Mark Omernick, Andrew M. Dai, Thanumalayan Sankaranarayanan Pillai, Marie Pellat, Aitor Lewkowycz, Erica Moreira, Rewon Child, Oleksandr Polozov, Katherine Lee, Zongwei Zhou, Xuezhi Wang, Brennan Saeta, Mark Díaz, Orhan Firat, Michele Catasta, Jason Wei, Kathleen S. Meier-Hellstern, Douglas Eck, Jeff Dean, Slav Petrov, and Noah Fiedel. 2022. Palm: Scaling language modeling with pathways . <i>J. Mach. Learn. Res.</i> , 24:240:1–240:113.	
737		
738		
739		
740		
741		
742		
743		
744		
745		
746		
747		
748		
749		
750		
751		
752		
753		
754		
755		
756		
757		
758		
759		
760	Jan Clusmann, Fiona R Kolbinger, Hannah Sophie Muti, Zunamys I Carrero, Jan-Niklas Eckardt, Narmin Ghaffari Laleh, Chiara Maria Lavinia Löffler, Sophie-Caroline Schwarzkopf, Michaela Unger, Gregory P Veldhuizen, et al. 2023. The future landscape of large language models in medicine. <i>Communications medicine</i> , 3(1):141.	
761		
762		
763		
764		
765		
766		
767	Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. 2021. Training verifiers to solve math word problems. <i>arXiv preprint arXiv:2110.14168</i> .	
768		
769		
770		
771		
772	Arman Cohan, Sergey Feldman, Iz Beltagy, Doug Downey, and Daniel S. Weld. 2020. Specter: Document-level representation learning using citation-informed transformers. In <i>ACL</i> .	
773		
774		
775		
776	Xuan Long Do, Kenji Kawaguchi, Min Yen Kan, and Nancy F Chen. 2023. Choire: Characterizing and predicting human opinions with chain of opinion reasoning . <i>arXiv preprint arXiv:2311.08385</i> .	
777		
778		
779		
780	Abbas Ghaddar, Phillippe Langlais, Mehdi Rezagholizadeh, and Ahmad Rashid. 2021. End-to-end self-debiasing framework for robust NLU training . In <i>Findings of the Association for Computational Linguistics: ACL-IJCNLP 2021</i> , pages 1923–1929, Online. Association for Computational Linguistics.	
781		
782		
783		
784		
785		
	Akshay Goel, Almog Gueta, Omry Gilon, Chang Liu, Sofia Erell, Lan Huong Nguyen, Xiaohong Hao, Bolous Jaber, Shashir Reddy, Rupesh Kartha, Jean Steiner, Itay Laish, and Amir Feder. 2023. Llms accelerate annotation for medical information extraction . In <i>Proceedings of the 3rd Machine Learning for Health Symposium</i> , volume 225 of <i>Proceedings of Machine Learning Research</i> , pages 82–100. PMLR.	786
		787
		788
		789
		790
		791
		792
		793
	Zishan Guo, Renren Jin, Chuang Liu, Yufei Huang, Dan Shi, Linhao Yu, Yan Liu, Jiakuan Li, Bojian Xiong, Deyi Xiong, et al. 2023. Evaluating large language models: A comprehensive survey. <i>arXiv preprint arXiv:2310.19736</i> .	794
		795
		796
		797
		798
	Shashank Gupta, Vaishnavi Shrivastava, Ameet Deshpande, Ashwin Kalyan, Peter Clark, Ashish Sabharwal, and Tushar Khot. 2024. Bias runs deep: Implicit reasoning biases in persona-assigned LLMs . In <i>The Twelfth International Conference on Learning Representations</i> .	799
		800
		801
		802
		803
		804
	Izzeddin Gur, Ofir Nachum, Yingjie Miao, Mustafa Safdari, Austin Huang, Aakanksha Chowdhery, Sharan Narang, Noah Fiedel, and Aleksandra Faust. 2022. Understanding html with large language models. <i>arXiv preprint arXiv:2210.03945</i> .	805
		806
		807
		808
		809
	Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021. Measuring massive multitask language understanding . In <i>International Conference on Learning Representations</i> .	810
		811
		812
		813
		814
	Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. 2020. The curious case of neural text degeneration . In <i>International Conference on Learning Representations</i> .	815
		816
		817
		818
	Sarthak Jain, Madeleine van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020a. SciREX: A challenge dataset for document-level information extraction . In <i>Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics</i> , pages 7506–7516, Online. Association for Computational Linguistics.	819
		820
		821
		822
		823
		824
		825
	Sarthak Jain, Madeleine Van Zuylen, Hannaneh Hajishirzi, and Iz Beltagy. 2020b. Scirex: A challenge dataset for document-level information extraction . <i>arXiv preprint arXiv:2005.00512</i> .	826
		827
		828
		829
	Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, L’elio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b . <i>arXiv preprint arXiv:2310.06825</i> .	830
		831
		832
		833
		834
		835
		836
		837
	Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large language models are zero-shot reasoners . In <i>Advances in Neural Information Processing Systems</i> .	838
		839
		840
		841

954	Seamless code integration in LLMs for enhanced mathematical reasoning . In <i>The Twelfth International Conference on Learning Representations</i> .	Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018b. HotpotQA: A dataset for diverse, explainable multi-hop question answering . In <i>Conference on Empirical Methods in Natural Language Processing (EMNLP)</i> .	1011
955			1012
956			1013
957	Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed Chi, Quoc V Le, and Denny Zhou. 2022. Chain-of-thought prompting elicits reasoning in large language models . In <i>Advances in Neural Information Processing Systems</i> , volume 35, pages 24824–24837. Curran Associates, Inc.		1014
958			1015
959			1016
960		Jeffrey Zhou, Tianjian Lu, Swaroop Mishra, Siddhartha Brahma, Sujoy Basu, Yi Luan, Denny Zhou, and Le Hou. 2023. Instruction-following evaluation for large language models . <i>arXiv preprint arXiv:2311.07911</i> .	1017
961			1018
962			1019
963			1020
964	Congying Xia, Chen Xing, Jiangshu Du, Xinyi Yang, Yihao Feng, Ran Xu, Wenpeng Yin, and Caiming Xiong. 2024. Fofu: A benchmark to evaluate llms’ format-following capability . <i>arXiv preprint arXiv:2402.18667</i> .		1021
965			
966			
967			
968			
969	Sang Michael Xie, Aditi Raghunathan, Percy Liang, and Tengyu Ma. 2022. An explanation of in-context learning as implicit bayesian inference . In <i>International Conference on Learning Representations</i> .		
970			
971			
972			
973	Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Li, Nora Bradford, Branda Sun, Tran Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022a. Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension . In <i>Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)</i> , pages 447–460, Dublin, Ireland. Association for Computational Linguistics.		
974			
975			
976			
977			
978			
979			
980			
981			
982			
983			
984	Ying Xu, Dakuo Wang, Mo Yu, Daniel Ritchie, Bingsheng Yao, Tongshuang Wu, Zheng Zhang, Toby Jia-Jun Li, Nora Bradford, Branda Sun, Tran Bao Hoang, Yisi Sang, Yufang Hou, Xiaojuan Ma, Diyi Yang, Nanyun Peng, Zhou Yu, and Mark Warschauer. 2022b. Fantastic questions and where to find them: FairytaleQA – an authentic dataset for narrative comprehension . Association for Computational Linguistics.		
985			
986			
987			
988			
989			
990			
991			
992			
993	Ziyang Xu, Keqin Peng, Liang Ding, Dacheng Tao, and Xiliang Lu. 2024. Take care of your prompt bias! investigating and mitigating prompt bias in factual knowledge extraction . <i>arXiv preprint arXiv:2403.09963</i> .		
994			
995			
996			
997			
998	Chengrun Yang, Xuezhi Wang, Yifeng Lu, Hanxiao Liu, Quoc V Le, Denny Zhou, and Xinyun Chen. 2024. Large language models as optimizers . In <i>The Twelfth International Conference on Learning Representations</i> .		
999			
1000			
1001			
1002			
1003	Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. 2018a. HotpotQA: A dataset for diverse, explainable multi-hop question answering . In <i>Proceedings of the 2018 Conference on Empirical Methods in Natural Language Processing</i> , pages 2369–2380, Brussels, Belgium. Association for Computational Linguistics.		
1004			
1005			
1006			
1007			
1008			
1009			
1010			

Algorithm 1 Format-Instruction Following Scorer

Input: Task T , language model \mathcal{M} , format constraints C , generated output Y .

Input: If C includes wrapping characters, we denote as $\{W_1, W_2\}$ and $is_wrapping = True$.

Input: $output_type$ is the data type required by C when T is not MCQ.

- 1: **if** $is_wrapping$ **then**
- 2: **return** False if (any of $\{W_1, W_2\} \notin Y$) or (number of $W_1 \in Y$ + number of $W_2 \in Y \neq 2$).
- 3: $ans =$ Extract string in between $\{W_1, W_2\}$.
- 4: **else**
- 5: $ans = Y$
- 6: **end if**
- 7: **if** T is MCQ **then**
- 8: **if** MCQ output type is character identifier **then**
- 9: **return** True if $ans \in \{A, B, C, D\}$. False otherwise.
- 10: **else**
- 11: **return** True if $ans \in \{\text{options' values}\}$. False otherwise.
- 12: **end if**
- 13: **else**
- 14: **return** True if we can parse ans as an instance of the class $output_type$. False otherwise.
- 15: **end if**

Alg. 1 presents our heuristic algorithm for evaluating the format-instruction following capabilities of LLMs, which is used to compute F_C in Eq. (1). The algorithm is divided into two three main parts:

1. **Lines 1-6.** These lines focus on examining the wrapping requirements by verifying the presence and correctness of the specified wrapping tokens.
2. **Lines 7-12.** These lines are dedicated to checking the formats of MCQ answers (§4.1).
3. **Lines 13-15.** These lines address the remaining formats, including list and mapping formats.

It is worth noting that Alg. 1 is highly adaptable; formats can be added or removed to tailor it for specific downstream applications.

B Theoretical Analysis: Reliability of $EstTrueE$ **B.1 Proof of Thm. 2.1**

Proof of Thm. 2.1. We omit the case when $FI_C = 0$ since in that case, we cannot estimate $TrueE$. By the definition in Thm. 2.1, we have S_C generated answers that satisfy C . Let's denote $k = S_C$ for simplicity. Let's denote k performance scores of answers satisfying C as x_1, \dots, x_k , and $\bar{x} = \frac{\sum_{i=1}^k (x_i)}{k}$ as the mean. Finally, $TrueE$ is the population mean of the performance scores, denoted as μ .

Statement 1: $EstTrueE$ is consistent. From Eq. (5), by rewriting $EstTrueE$, we have $EstTrueE = \frac{1}{n} \cdot \sum_{i=1}^k (x_i) \cdot \frac{n}{k} = \bar{x}$, which is an unbiased estimator of the average performance $TrueE$, i.e., $Bias(\bar{x}) = 0$ or $\lim_{k \rightarrow \infty} Bias(EstTrueE) = 0$ (1). Now, let's denote the variance of the performance scores as σ^2 , then the variance of $EstTrueE$ is $Var(EstTrueE) = Var(\bar{x}) = \frac{\sigma^2}{n}$ and $\lim_{k \rightarrow \infty} Var(EstTrueE) = 0$ (2). From (1) and (2), by the Sufficient Condition for Consistency (Amemiya, 1985), we conclude that $EstTrueE$ is a consistent estimator.

Statement 2: FI_C value. Let's denote $s^2 = \frac{1}{k-1} \sum_{i=1}^k (x_i - \bar{x})^2$ as the sample variance of the performance scores x_i s. It is well-known that $\frac{\sqrt{k}(\bar{x} - \mu)}{s} \sim t_{k-1}$. For estimating the population mean μ with finite population size n and the type I error α , we have the margin of error ϵ :

$$\epsilon \geq t_{\alpha/2, k-1} \cdot \sqrt{\frac{n-k}{n} \cdot \frac{s^2}{k}} \quad (9)$$

where $\frac{n-k}{n}$ is the finite population correction factor. Eq. (9) is equivalent to:

$$k \geq \frac{n-k}{n} \cdot \left(\frac{t_{\alpha/2, k-1} \cdot s}{\epsilon} \right)^2 \quad (10)$$

which yields

$$k \geq \frac{1}{\frac{1}{n} + \left(\frac{\epsilon}{t_{\alpha/2, k-1} \cdot s} \right)^2}. \quad (11)$$

then

$$FI_C = \frac{k}{n} \geq \frac{1}{1 + n \cdot \left(\frac{\epsilon}{t_{\alpha/2, k-1} \cdot s} \right)^2}. \quad (12)$$

Statement 3: When FI_C approaches 1, $EstTrueE$ approaches $TrueE$. Since $EstTrueE$ by its definition in Eq. (5) is continuous with respect to FI_C (Eq. (5)), S_C (Eq. (3)) and F_C (Eq. (3)), therefore, we have the equality:

$$\lim_{FI_C \rightarrow 100\%} (EstTrueE) = EstTrueE(FI_C = 100\%) = TrueE.$$

□

B.2 Python Codes for Computing Reliability

```

1059 import numpy as np
1060 from scipy.stats import t
1061 import math
1062
1063 def compute_sample_variance(data):
1064     n = len(data)
1065     mean = np.mean(data)
1066     squared_deviations = [(x - mean) ** 2 for x in data]
1067     sample_variance = sum(squared_deviations) / (n - 1)
1068     return sample_variance
1069
1070 def is_estimator_reliable(num_FI, list_eval_scores, num_samples=200):
1071     ##### t-statistics #####
1072     alpha = 0.05 # 5% significance level
1073     df = num_FI # degrees of freedom
1074     alpha_two_tailed = alpha / 2
1075     t_statistic = t.ppf(1 - alpha_two_tailed, df)
1076
1077     ##### Compute MOE_FI #####
1078     epsilon = 0.05 # 5% margin of error
1079     s = math.sqrt(compute_sample_variance(list_eval_scores))
1080     return num_FI/num_samples > 1/(1 + num_samples * (epsilon/(t_statistic * s))**2)

```

Code Listing 1: Python codes for computing the reliability of $EstTrueE$ with margin of errors 5% performance with a significance level 5%.

C Detailed Discussions

We give the numerical results and discussions for all figures and points made in the main paper.

C.1 Multiple-choice Question (MCQ) Discussions

We evaluate Gemma, Mistral, and ChatGPT on the MMLU and BBH datasets using two prompting techniques, Zero-shot (ZS) and Zero-shot Chain-of-Thought (ZS-CoT) (§4.1). The prompts are specified in §E.1. We report the FI_C , $SysE$, $EstTrueE$ scores. The results are presented in Tab. 3. Additionally, Tab. 4, Tab. 5, and Tab. 6 are the distillation results of Tab. 3:

1. Tab. 4. For each model, we average its $EstTrueE$ performance overall benchmarks and prompting techniques. For each task, we average the $EstTrueE$ scores overall models and prompting techniques. The results of this table are plotted in Fig. 2 and discussed in §4.1.
2. Tab. 5. The purpose of this table is to compare the FI scores across formats. We average all the FI scores across models and tasks.
3. Tab. 6. The purpose of this table is to see whether CoT (Wei et al., 2022) mitigates format bias. We average all the $EstTrueE$ scores over all models and benchmarks for each ZS and ZS-CoT prompting method.

MCQ type	Char.	Text.
MMLU		
Gemma-7B-it (EstTrue-Acc)	0.53 / 27.25	8.10 / 18.63
Gemma-7B-it (Systematic-Acc)	0.12 / 10.32	0.17 / 4.86
Gemma-7B-it (FI)	22.47 / 37.87	2.10 / 26.09
Mistral-7B-it (EstTrue-Acc)	46.14 / 49.31	8.37 / 8.52
Mistral-7B-it (Systematic-Acc)	41.59 / 45.94	0.17 / 0.19
Mistral-7B-it (FI)	90.12 / 93.16	2.03 / 2.23
ChatGPT (EstTrue-Acc)	68.55 / 45.53	54.85 / 59.67
ChatGPT (Systematic-Acc)	66.20 / 42.22	12.71 / 26.31
ChatGPT (FI)	96.56 / 92.73	23.17 / 44.09
BBH		
Gemma-7B-it (EstTrue-Acc)	42.11 / 23.05	0.00 / 15.11
Gemma-7B-it (Systematic-Acc)	0.40 / 13.00	0.00 / 6.80
Gemma-7B-it (FI)	0.95 / 56.40	0.00 / 45.00
Mistral-7B-it (EstTrue-Acc)	76.81 / 62.50	0.00 / 0.00
Mistral-7B-it (Systematic-Acc)	21.20 / 22.00	0.00 / 0.00
Mistral-7B-it (FI)	27.60 / 35.20	0.00 / 1.60
ChatGPT (EstTrue-Acc)	73.03 / 57.14	53.63 / 0.00
ChatGPT (Systematic-Acc)	26.00 / 16.0	53.20 / 0.00
ChatGPT (FI)	35.60 / 28.00	99.20 / 0.00

Table 3: MCQ output format results. Red results are unreliable results measured by Thm. 2.1 i.e., inequality Eq. (6) does not hold.

Format instruction-following bias. The FI scores across formats are illustrated in Tab. 5. There is a notable difference between the scores for character-based and textual value-based formats. The score for character format (51.39%) is more than twice as high as that for textual value format (20.46%). Among the models, ChatGPT follows the instructions best with FI score 52.42%. Below we present two examples of Gemma and Mistral failing to follow the format instructions:

	Char.	Text.	$BiasF_o$ (Var)
Models			
Gemma	23.24	10.46	40.83
Mistral	58.69	4.22	741.74
ChatGPT	61.07	42.04	90.53
Tasks			
MMLU	39.56	26.36	43.56
BBH	55.77	11.46	490.84

Table 4: Avg. estimated true accuracy results of MCQ benchmarks across models (Gemma-7B-it, Mistral, ChatGPT) and datasets (MMLU, BBH) with two MCQ formats (Character and Textual Value). This table is the distillation result of averaging all the estimated true performance scores across models (**Models**) and tasks (**Tasks**) in Tab. 4.

Models	Char.	Text.	Avg.	$BiasF_o$ (Var)
Gemma	29.42	18.29	23.86	30.96
Mistral	61.52	1.46	31.49	901.80
ChatGPT	63.22	41.61	52.41	116.74
Avg.	51.39	20.46	35.92	239.16

Table 5: MCQ format-instruction following scores across formats. This table is the distillation result of averaging all the FI scores across tasks and prompting techniques in Tab. 4.

Answer the following question by outputting the textual value of the choice without its character identifier. Wrap your final answer by `<ANSWER>` and `</ANSWER>`.

Question: Is the following sentence plausible? “John Carlson scored in the third period.”
 Answer choices: A. yes | B. no

Gemma: `<ANSWER>A</ANSWER>`

Answer the following question by outputting the textual value of the choice without its character identifier. Wrap your final answer by `<ANSWER>` and `</ANSWER>`.

Question: Is the following sentence plausible? “Marcelo got on the end of a through ball.”
 Answer choices: A. yes | B. no

Mistral: Based on the given sentence and the provided answer choices, the answer is: **A. yes**. The sentence describes Marcelo receiving a through ball and getting on the end of it, which is a common football (soccer) term for receiving a long, accurate pass and finishing a move with a shot or a goal.

ZS versus ZS-CoT. We investigate the impact of Chain-of-Thought (CoT) (Wei et al., 2022) on mitigating format bias. The average of estimated True Accuracy scores across models and tasks for the ZS prompting and ZS-CoT prompting are shown in Tab. 6. ZS prompting achieves a higher score in

the character category (51.20%) compared to ZS-CoT prompting (44.13%). Similarly, for textual value format, ZS prompting scores higher (20.83%) than ZS-CoT prompting (16.99%). However, the $BiasF_o$ is lower for the ZS-CoT model (184.14%²) compared to the ZS model (230.58%²), indicating that CoT slightly decreases the format bias.

	Char.	Text.	$BiasF_o$
Zero-shot	51.20	20.83	230.58
Zero-shot Chain-of-Thought	44.13	16.99	184.14

Table 6: MCQ CoT versus non-CoT. This table is the distillation result of averaging all the Zero-shot and Zero-shot Chain-of-Thought scores across models and tasks in Tab. 4.

Reliability of the results. From Tab. 3, we see that 16/24 of the estimated $EstTrue$ results are reliable. The reliability of results in the MCQ output format varies across different models. Gemma-7B-it and Mistral-7B-it show significant unreliability in textual value format, evidenced by numerous red-marked scores due to models not following the format instructions to output correct formats. In contrast, ChatGPT’s results are significantly more reliable in the MMLU and BBH benchmarks (7/8), with only one unreliable result in the BBH textual format output.

C.2 Wrapping Discussions

We examine Gemma, Mistral, and ChatGPT on the MCQ datasets (MMLU, BBH) and generation datasets (GSM8K, HotpotQA, FairytaleQA) utilizing two prompting techniques, Zero-shot (ZS) and Zero-shot Chain-of-Thought (ZS-CoT) (§4.2). The prompts are also provided in §E.2. We measure the FI_C , $SysE$, $EstTrueE$. The results are shown in Tab. 7. Furthermore, Tab. 8, Tab. 9 and Tab. 10 are the distillation outcome of Tab. 7:

1. Tab. 8. For each model, we average its $EstTrueE$ performance overall benchmarks and prompting techniques. For each task, we average the $EstTrueE$ scores overall models and prompting techniques. This table is plotted in Fig. 3 and discussed in §4.2.
2. Tab. 9. The purpose of this table is to compare the FI scores across formats. We average all the FI scores across models and tasks.
3. Tab. 10. The purpose of this table is to see whether CoT (Wei et al., 2022) mitigates format bias. We average all the $EstTrueE$ scores over all models and benchmarks for each ZS and ZS-CoT prompting method.

Format instruction-following bias. The FI scores over formats are provided in Tab. 9. Overall, LLMs exhibit significant format-following bias across formats with a variance of FI scores of 297.28%². Among the models, ChatGPT follows the instructions best with average FI Score 85.01%. The “Special Character” wrapping format has the highest FI score of 73.34%. Following it is the “Placeholder” wrapping format also shows a high FI score of 68.37%, suggesting it is another effective format for ensuring instruction adherence. In contrast, the “Quoting” wrapping format has the lowest FI score of 17.06%. This significant drop compared to other formats suggests that quoting is the least effective method for wrapping instructions, possibly causing confusion or misinterpretation by the models. Below we present two examples of Gemma and Mistral failing to follow the format instructions:

Wrapping type	Special character	Bolding	Italicizing	Brackets	Parentheses	Placeholder	Quoting
MMLU							
Gemma-7B-it (EstTrue-Acc)	35.59 / 20.28	41.28 / 44.27	49.85 / 74.18	36.36 / 32.95	36.68 / 20.12	46.45 / 25.77	60.41 / 74.06
Gemma-7B-it (Systematic-Acc)	27.82 / 20.28	21.66 / 17.73	26.64 / 27.89	28.55 / 27.28	10.53 / 12.96	29.80 / 21.96	2.64 / 2.37
Gemma-7B-it (FI)	78.16 / 100.00	52.47 / 39.60	53.44 / 37.60	78.52 / 82.80	28.71 / 64.40	64.15 / 85.20	4.37 / 3.20
Mistral-7B-it (EstTrue-Acc)	53.63 / 58.34	48.43 / 63.09	51.84 / 61.66	67.36 / 61.58	64.99 / 62.71	75.35 / 6.03	100.00 / 8.33
Mistral-7B-it (Systematic-Acc)	13.42 / 20.04	1.08 / 9.40	4.80 / 10.15	20.08 / 17.28	11.10 / 13.42	1.07 / 0.14	0.03 / 0.01
Mistral-7B-it (FI)	23.81 / 34.35	2.23 / 14.90	9.26 / 16.46	29.81 / 28.06	17.08 / 21.40	1.42 / 2.32	0.03 / 0.12
ChatGPT (EstTrue-Acc)	54.64 / 71.28	67.40 / 75.86	44.76 / 64.79	59.80 / 71.42	57.82 / 71.11	66.24 / 72.81	68.29 / 70.68
ChatGPT (Systematic-Acc)	48.54 / 63.64	66.59 / 48.59	38.24 / 36.77	31.65 / 60.86	28.54 / 60.57	63.88 / 50.09	26.72 / 30.26
ChatGPT (FI)	88.84 / 89.28	98.80 / 64.05	85.43 / 56.75	52.93 / 85.21	49.36 / 85.18	96.44 / 68.80	39.13 / 42.81
BBH							
Gemma-7B-it (EstTrue-Acc)	25.00 / 16.00	49.09 / 38.38	52.94 / 24.47	63.04 / 47.34	36.73 / 26.09	7.07 / 3.76	60.00 / 20.00
Gemma-7B-it (Systematic-Acc)	24.00 / 16.00	21.60 / 15.20	10.80 / 9.20	23.20 / 19.60	14.40 / 16.80	5.20 / 3.20	2.40 / 0.40
Gemma-7B-it (FI)	96.00 / 100.00	44.00 / 39.60	20.40 / 37.60	36.80 / 41.40	39.20 / 64.40	73.60 / 85.20	4.00 / 2.00
Mistral-7B-it (EstTrue-Acc)	52.40 / 64.00	10.40 / 11.60	36.80 / 21.20	16.00 / 8.40	6.4 / 12.00	32.80 / 72.80	0.00 / 0.00
Mistral-7B-it (Systematic-Acc)	49.04 / 58.11	1.37 / 1.85	34.88 / 14.24	6.84 / 1.61	1.51 / 3.98	13.38 / 71.05	0.00 / 0.00
Mistral-7B-it (FI)	93.60 / 90.80	13.20 / 16.00	94.80 / 67.20	42.80 / 19.20	23.60 / 33.20	40.80 / 97.60	0.00 / 0.00
ChatGPT (EstTrue-Acc)	64.00 / 47.20	74.80 / 36.80	9.20 / 14.40	53.60 / 51.60	63.60 / 13.60	54.00 / 14.80	14.00 / 18.00
ChatGPT (Systematic-Acc)	64.00 / 16.80	74.80 / 30.62	9.20 / 10.02	51.67 / 38.60	57.24 / 3.75	54.00 / 14.80	3.19 / 0.58
ChatGPT (FI)	100.00 / 35.60	100.00 / 83.20	100.00 / 69.60	96.40 / 74.80	90.00 / 27.60	100.00 / 100.00	22.80 / 3.20
GSM8K							
Gemma-7B-it (EstTrue-F1)	3.65 / 5.00	0.99 / 3.13	5.20 / 1.46	7.45 / 0.42	0.00 / 0.00	9.13 / 9.92	0.0 / 0.0
Gemma-7B-it (Systematic-F1)	2.54 / 2.45	0.50 / 2.00	4.26 / 1.19	3.50 / 0.17	0.00 / 0.00	4.52 / 4.71	0.0 / 0.0
Gemma-7B-it (FI)	69.50 / 49.00	50.50 / 64.00	82.00 / 81.50	47.00 / 40.05	2.50 / 0.50	49.50 / 47.50	0.0 / 0.0
Mistral-7B-it (EstTrue-F1)	4.03 / 25.74	9.03 / 31.61	2.87 / 30.76	2.57 / 46.98	1.29 / 39.44	3.28 / 39.37	0.00 / 73.52
Mistral-7B-it (Systematic-F1)	3.43 / 23.43	1.40 / 4.11	1.42 / 20.76	1.67 / 38.76	0.60 / 24.26	3.28 / 38.78	0.00 / 6.25
Mistral-7B-it (FI)	85.00 / 91.00	15.50 / 13.00	49.50 / 67.50	65.00 / 82.50	46.50 / 61.50	100.00 / 98.50	5.00 / 8.50
ChatGPT (EstTrue-F1)	19.54 / 43.98	22.95 / 24.36	21.22 / 30.57	21.27 / 69.00	22.02 / 63.83	23.03 / 60.25	16.43 / 24.01
ChatGPT (Systematic-F1)	19.44 / 43.98	22.84 / 23.39	21.12 / 24.15	20.74 / 67.62	21.25 / 62.24	23.03 / 59.05	9.78 / 14.65
ChatGPT (FI)	99.50 / 100.00	99.50 / 96.00	99.50 / 79.00	97.50 / 98.50	96.50 / 97.50	100.00 / 98.00	59.50 / 61.00
HotpotQA							
Gemma-7B-it (EstTrue-F1)	14.12 / 9.88	21.43 / 32.11	19.83 / 27.06	23.63 / 30.44	0.00 / 0.00	43.70 / 53.62	2.33 / 6.60
Gemma-7B-it (Systematic-F1)	4.59 / 5.53	9.00 / 12.20	7.93 / 8.93	3.90 / 14.00	0.00 / 0.00	5.90 / 9.92	0.03 / 0.03
Gemma-7B-it (FI)	32.50 / 56.00	42.00 / 38.00	40.00 / 33.00	16.50 / 46.00	3.50 / 2.50	13.50 / 18.50	1.50 / 0.50
Mistral-7B-it (EstTrue-F1)	12.86 / 11.43	25.84 / 29.21	20.93 / 14.56	16.93 / 13.20	15.39 / 13.21	20.41 / 21.58	0.00 / 25.00
Mistral-7B-it (Systematic-F1)	7.27 / 3.83	8.27 / 3.36	6.91 / 4.95	16.51 / 10.76	14.55 / 10.24	19.70 / 14.75	0.00 / 0.05
Mistral-7B-it (FI)	56.50 / 33.50	32.00 / 11.50	33.00 / 34.00	97.50 / 81.50	94.50 / 77.50	96.50 / 91.50	0.00 / 0.20
ChatGPT (EstTrue-F1)	29.86 / 27.52	41.00 / 33.14	35.39 / 28.96	23.94 / 35.48	29.30 / 34.83	38.72 / 28.69	41.52 / 16.97
ChatGPT (Systematic-F1)	25.24 / 27.11	40.59 / 30.82	33.45 / 26.64	17.00 / 33.36	23.46 / 33.44	38.72 / 27.69	11.73 / 7.13
ChatGPT (FI)	84.50 / 98.50	99.00 / 93.00	94.50 / 92.00	71.50 / 94.00	80.05 / 96.00	100.00 / 96.50	28.50 / 42.00
FairytaleQA							
Gemma-7B-it (EstTrue-F1)	17.42 / 29.72	8.91 / 0.97	8.12 / 14.50	22.13 / 18.62	0.00 / 0.00	20.64 / 22.05	0.00 / 0.00
Gemma-7B-it (Systematic-F1)	6.62 / 11.74	4.68 / 0.64	4.75 / 9.79	1.77 / 1.21	0.00 / 0.00	2.58 / 4.08	0.0 / 0.0
Gemma-7B-it (FI)	38.00 / 39.50	52.50 / 66.00	58.50 / 67.50	8.00 / 6.50	0.00 / 0.00	12.50 / 18.50	0.0 / 0.0
Mistral-7B-it (EstTrue-F1)	27.19 / 22.20	23.78 / 50.00	47.36 / 29.49	32.42 / 25.90	30.33 / 22.46	36.07 / 31.77	19.50 / 20.00
Mistral-7B-it (Systematic-F1)	22.16 / 18.54	3.21 / 0.50	18.47 / 15.19	32.42 / 25.00	29.73 / 21.00	35.89 / 31.62	0.39 / 1.30
Mistral-7B-it (FI)	81.50 / 83.50	13.50 / 1.00	39.00 / 51.50	100.00 / 96.50	98.00 / 93.50	99.50 / 99.50	2.00 / 6.50
ChatGPT (EstTrue-F1)	41.93 / 31.95	46.08 / 32.84	48.11 / 33.46	41.53 / 38.25	38.25 / 34.82	46.83 / 32.85	45.78 / 27.75
ChatGPT (Systematic-F1)	38.58 / 31.47	46.08 / 31.86	48.11 / 31.96	41.33 / 38.06	45.91 / 34.30	46.83 / 32.85	27.24 / 14.71
ChatGPT (FI)	92.00 / 98.50	100.00 / 97.00	100.00 / 95.50	99.50 / 99.50	99.50 / 98.50	100.00 / 100.00	59.50 / 53.00

Table 7: Wrapping output format results. Red results are unreliable results measured by Thm. 2.1 i.e., inequality Eq. (6) does not hold.

	Special Character	Bolding	Italicizing	Brackets	Parentheses	Placeholder	Quoting	$BiasF_o$ (Var)
Models								
Gemma	31.09	20.11	24.77	23.39	14.61	33.86	11.63	56.33
Mistral	34.06	42.43	27.91	37.44	32.83	35.49	30.90	18.83
ChatGPT	32.47	33.40	32.95	38.16	38.49	38.40	28.69	12.26
Average	32.54	31.98	28.54	33.00	28.64	35.92	23.74	13.55
Tasks								
MMLU	49.42	56.72	57.85	54.91	52.24	48.77	63.63	23.26
BBH	44.77	36.85	26.50	40.00	26.40	30.87	18.67	70.16
GSM8K	17.00	15.35	15.35	24.56	21.10	24.16	19.00	12.97
HotpotQA	17.62	30.46	24.46	23.91	15.46	32.21	15.19	41.40
FairytaleQA	28.40	27.10	30.18	26.12	22.29	31.70	18.84	17.28
Average	31.44	33.29	30.87	33.90	27.50	33.57	27.06	6.88

Table 8: Avg. estimated true accuracy results of benchmarks across models (Gemma-7B-it, Mistral, ChatGPT) and datasets (MMLU, BBH) with seven Wrapping format (Special Character, Bolding, Italicizing, Brackets, Parentheses, Placeholder, Quoting). This table is the distillation result of averaging all the estimated true performance scores across models and benchmarks in Tab. 7.

Model	Special Character	Bolding	Italicizing	Brackets	Parentheses	Placeholder	Quoting	Avg.	$BiasF_o$ (Var)
Gemma	62.27	46.07	49.51	37.06	15.33	45.58	1.36	36.74	384.31
Mistral	63.00	15.36	40.94	67.53	54.72	64.48	4.04	44.30	553.55
ChatGPT	94.77	93.49	88.35	88.93	88.69	95.06	45.79	85.01	263.71
Avg.	73.34	51.64	59.60	64.51	52.91	68.37	17.06	55.35	297.28

Table 9: Avg. Following Instruction Score over all the wrapping formats. This table is the distillation result of averaging all the FI scores across models and benchmarks in Tab. 7.

Answer the following question without any explanation. Wrap your final answer using triple quotation marks.

Question (HotpotQA): “What was the MGM Grand Garden Arena in which Britney Spears recorded fourth video album originally known as?”

Context:...

Gemma: MGM Grand Garden Special Events.

1139

Answer the following question without any explanation. Wrap your final answer using triple quotation marks.

Question (BBH): Is the following sentence plausible? “David Silva took a throw-in.”

Mistral: A. “yes”.

1140

ZS versus ZS-CoT. The average of estimated True Accuracy scores across models and tasks for the ZS and ZS-CoT prompting are shown in Tab. 10. For the majority of the wrapping methods (“Special Character”, “Bolding”, “Italicizing”, and “Brackets”), the ZS-CoT model generally shows higher or comparable performance to the ZS model. The “Brackets” shows a significant improvement when using ZS-CoT, with a jump from 31.05% (ZS) to 36.00% (ZS-CoT). However, the $BiasF_o$ metric shows a considerable divergence between 4.89%² (ZS) and 11.65%² (ZS-CoT). This depicts that while applying CoT may improve accuracy in certain methods, it does not generally reduce format bias.

1141

1142

1143

1144

1145

1146

1147

	Special Character	Bolding	Italicizing	Brackets	Parentheses	Placeholder	Quoting	$BiasF_o$
Zero-shot	30.57	32.76	30.30	31.05	27.38	34.42	28.50	4.89
Zero-shot Chain-of-Thought	27.99	34.32	32.33	36.00	27.05	31.02	26.58	11.65

Table 10: Avg. Estimated Accuracy of non CoT versus CoT for wrapping methods. This table is the distillation result of averaging all the Zero-shot and Zero-shot Chain-of-Thought scores across models and tasks in Tab. 7.

Reliability of the results. Overall, 80% of the $EstTrue$ results (169/210) are reliable. Gemma-7B-it shows mixed reliability, with some red-marked scores indicating unreliable results, particularly in the “Quoting” format. This is because Gemma failed to follow the quoting instruction to quote the final answer. Mistral-7B-it exhibits similar variability, with some unreliable scores in “Quoting” and “Placeholder” formats. ChatGPT generally demonstrates mostly reliable results, with only 1 quoting result unreliable.

1148

1149

1150

1151

1152

C.3 List Discussions

1153

We assess Gemma, Mistral, and ChatGPT with two prompting techniques, Zero-shot (ZS) and Zero-shot Chain-of-Thought (ZS-CoT) (§4.3) on two benchmarks SciDocsRR and SemEval2017. Our prompts are provided in E.3. We utilize FI_C , $SysE$, $TrueE$ as our evaluation metrics. The results are illustrated in Tab. 11. In addition, Tab. 12, Tab. 13 and Tab. 14 are the distillation results of Tab. 11:

1154

1155

1156

1157

Listing type	Python	Bullet	Spe. Char.	Newline
		SciDocsRR		
Gemma-7B-it (EstTrue- <i>mAP</i>)	0.0 / 61.65	0.0 / 73.0	0.0 / 60.00	0.0 / 60.15
Gemma-7B-it (Systematic- <i>mAP</i>)	0.0 / 15.72	0.0 / 1.46	0.0 / 0.90	0.0 / 28.27
Gemma-7B-it (FI)	0.0 / 25.50	0.0 / 2.00	0.0 / 1.50	0.0 / 47.00
Mistral (EstTrue- <i>mAP</i>)	50.21 / 52.61	0.00 / 0.00	0.00 / 0.00	78.08 / 58.36
Mistral (Systematic- <i>mAP</i>)	37.41 / 9.47	0.00 / 0.00	0.00 / 0.00	18.35 / 27.14
Mistral (FI)	74.50 / 18.00	0.00 / 0.00	0.00 / 0.00	23.50 / 46.50
ChatGPT (EstTrue- <i>mAP</i>)	35.29 / 50.17	49.94 / 59.64	55.69 / 57.78	38.54 / 57.56
ChatGPT (Systematic- <i>mAP</i>)	33.17 / 28.60	49.19 / 25.05	55.69 / 37.85	35.46 / 35.41
ChatGPT (FI)	94.00 / 57.00	98.50 / 42.00	100.00 / 65.50	92.00 / 61.50
		SemEval2017		
Gemma-7B-it (EstTrue- <i>F1</i>)	4.00 / 8.86	7.10 / 7.20	4.80 / 13.50	7.21 / 3.25
Gemma-7B-it (Systematic- <i>F1</i>)	0.04 / 1.64	1.80 / 2.10	4.80 / 13.50	7.21 / 1.51
Gemma-7B-it (FI)	1.00 / 18.50	25.50 / 29.15	100.00 / 100.00	100.00 / 46.50
Mistral (EstTrue- <i>F1</i>)	34.82 / 30.24	23.2 / 0.00	0.00 / 13.57	12.17 / 20.84
Mistral (Systematic- <i>F1</i>)	33.95 / 24.19	23.20 / 0.00	0.00 / 10.72	12.17 / 20.84
Mistral (FI)	97.50 / 80.00	100.00 / 100.00	0.00 / 79.00	100.00 / 100.00
ChatGPT (EstTrue- <i>F1</i>)	42.25 / 15.33	8.87 / 16.46	32.19 / 16.33	37.16 / 22.87
ChatGPT (Systematic- <i>F1</i>)	39.51 / 6.04	8.87 / 16.13	31.07 / 15.51	37.16 / 22.75
ChatGPT (FI)	93.50 / 39.39	100.00 / 97.97	96.50 / 94.94	100.00 / 99.49

Table 11: List output format results. Red results are unreliable results measured by Thm. 2.1 i.e., inequality Eq. (6) does not hold.

1. Tab. 12. For each model, we average its *EstTrueE* performance overall benchmarks and prompting techniques. For each task, we average the *EstTrueE* scores overall models and prompting techniques. This table is drawn in Fig. 4 and its discussions are conducted in §4.3.
2. Tab. 13. The purpose of this table is to compare the FI scores across formats. We average all the FI scores across models and tasks.
3. Tab. 14. The purpose of this table is to see whether CoT (Wei et al., 2022) mitigates format bias. We average all the *EstTrueE* scores over all models and benchmarks for each ZS and ZS-CoT prompting method.

	Python	Bullet	Special Character	Newline	$BiasF_o$ (Var)
Models					
Gemma	17.12	18.25	15.12	16.21	1.32
Mistral	41.98	5.80	3.39	42.37	353.80
ChatGPT	35.76	33.73	40.50	39.03	7.08
Average	31.62	19.26	19.67	32.53	39.87
Tasks					
SemEval2017	21.58	8.09	10.43	16.29	27.58
SciDocsRR	41.66	30.43	28.91	48.78	67.08
Average	31.62	19.26	19.67	32.54	39.91

Table 12: Avg estimated true F1 (SemEval2017) and MAP (SciDocsRR) scores over benchmarks (SemEval2017, SciDocsRR) and models (Gemma-7B-it, Mistral, ChatGPT) across 4 widely used list formats (Python, Bullet, Special Character, Newline)

Format instruction-following bias. From Tab. 13, the “Newline list” format has the highest FI score of 66.05%, while the FI scores of other formats are around 48.6%. Among the models, ChatGPT follows the instructions best with FI score 83.27%. Specifically, ChatGPT excels at following instructions for all formats. In comparison, Mistral performs poorly with “Special character list”, while Gemma struggles the most with “Python” and “Bullet point list”. Explaining these phenomena, we observe that Gemma often misinterprets the task of converting textual samples into lists of numbers as a coding task, resulting in

answers that do not match the desired format or content. We take an example in dataset **SciDocsRR §D.1** with detailed prompt as in §E.3

1172
1173

Given a query, and a list of documents.
 Query: Declarative Terrain Modeling for Military Training Games
 Documents: [...]

You are required to transform the list of documents into a binary list of 1 or 0 where 1 indicates the document brings useful information to the topic, and 0 indicates the document does not bring useful information to the topic. **Generate your binary list as a Python list**

Gemma: You should not use any packages such as sklearn, gensim, and NLTK. You are required to write a short explanation for your binary list.

1174

Given a query, and a list of documents.
 Query: Declarative Terrain Modeling for Military Training Games
 Documents: [...]

You are required to transform the list of documents into a binary list of 1 or 0 where 1 indicates the document brings useful information to the topic, and 0 indicates the document does not bring useful information to the topic. **Generate your binary list using bullet points**

Gemma: Your binary list must be in the following format: [1, 0, 1, 0, 0, 0, 1, 1, 0, 1, 0, 1, 1, 1, 1, 0, 1, 1, 1, 0].

1175

Model	Python	Bullet	Special Character	Newline	Avg.	$BiasF_o$ (Var)
Gemma	7.51	13.41	34.55	42.40	24.46	277.69
Mistral	67.50	50.00	19.75	67.50	51.19	507.31
ChatGPT	70.97	84.61	89.24	88.25	83.27	71.13
Avg.	48.66	49.34	47.84	66.05	52.97	76.36

Table 13: Avg Following Instruction scores over benchmarks (SemEval2017, SciDocsRR) and models (Gemma-7B-it, Mistral, ChatGPT) across 4 widely used list formats (Python, Bullet, Special Character, Newline). This table is the distillation result of averaging all the FI scores across models and benchmarks in Tab. 11.

ZS versus ZS-CoT. The results, detailed in Tab. 14 indicate that prompting with ZS-CoT substantially enhances model performance across various formats. Moreover, ZS-CoT effectively reduces format bias, as evidenced by the $BiasF_o$ metric decreasing from 46.88%² to 33.69%². From this, we conclude that CoT reduces format bias.

1176
1177
1178
1179

	Python	Bullet	Special Character	Newline	$BiasF_o$ (Var)
Zero-shot	27.76	13.67	14.73	27.98	46.88
Zero-shot Chain-of-Thought	35.47	24.85	24.62	37.09	33.69

Table 14: Avg estimated true F1 (SemEval2017) and MAP (SciDocsRR) scores of non-CoT versus CoT for list formats. This table is the distillation result of averaging all the scores across models and benchmarks in Tab. 11.

Reliability of the results. From Tab. 11, 73%(35/48) of the $EstTrue$ results are reliable. However, some scores of Gemma-7B-it (8/16) and Mistral-7B-it (5/16) on these benchmarks are red-marked,

1180
1181

1182 indicating unreliable results of this model. In contrast, the ChatGPT’s results are perfectly reliable.

1183 **C.4 Mapping Discussions**

Mapping type	JSON	YAML
SciREX Easy		
Gemma-7B-it (EstTrue-F1)	14.60 / 20.84	18.20 / 0.82
Gemma-7B-it (Systematic)	3.54 / 3.79	3.03 / 0.10
Gemma-7B-it (FI)	24.24 / 18.18	16.64 / 12.12
Mistral-7B-it (EstTrue-F1)	28.83 / 32.82	0.00 / 0.00
Mistral-7B-it (Systematic)	11.36 / 32.33	0.00 / 0.00
Mistral-7B-it (FI)	39.39 / 98.48	0.00 / 3.03
ChatGPT (EstTrue-F1)	35.99 / 22.40	23.63 / 26.60
ChatGPT (Systematic)	32.72 / 19.69	22.92 / 20.15
ChatGPT (FI)	90.90 / 87.87	96.96 / 75.75
SciREX Medium		
Gemma-7B-it (EstTrue-F1)	18.17 / 5.27	0.00 / 1.87
Gemma-7B-it (Systematic)	3.03 / 0.88	0.00 / 0.17
Gemma-7B-it (FI)	16.67 / 16.67	18.18 / 9.09
Mistral-7B-it (EstTrue-F1)	26.48 / 23.81	18.97 / 20.83
Mistral-7B-it (Systematic)	21.27 / 23.81	1.15 / 0.25
Mistral-7B-it (FI)	80.30 / 100.00	6.06 / 1.20
ChatGPT (EstTrue-F1)	29.07 / 27.29	36.55 / 22.70
ChatGPT (Systematic)	28.19 / 26.47	21.60 / 22.70
ChatGPT (FI)	96.96 / 96.96	59.09 / 100.00
SciREX Hard		
Gemma-7B-it (EstTrue-F1)	34.40 / 29.18	1.65 / 0.87
Gemma-7B-it (Systematic)	4.17 / 10.61	0.25 / 0.04
Gemma-7B-it (FI)	12.12 / 36.36	15.15 / 4.55
Mistral-7B-it (EstTrue-F1)	22.44 / 30.34	12.54 / 15.95
Mistral-7B-it (Systematic)	20.40 / 26.66	1.71 / 1.58
Mistral-7B-it (FI)	90.90 / 87.87	13.63 / 9.90
ChatGPT (EstTrue-F1)	20.25 / 22.57	11.76 / 12.07
ChatGPT (Systematic)	19.64 / 22.23	11.59 / 10.43
ChatGPT (FI)	96.96 / 98.48	98.48 / 86.36

Table 15: Mapping output format results. **Red** results are unreliable results measured by Thm. 2.1 i.e., inequality Eq. (6) does not hold.

	JSON	YAML	Average	$BiasF_o$ (Var)
Models				
Gemma	20.42	3.91	12.17	68.14
Mistral	27.46	11.39	19.43	64.56
ChatGPT	26.27	22.22	24.25	4.10
Tasks				
Easy	25.92	11.55	18.74	51.62
Medium	21.69	16.82	19.26	5.92
Hard	26.53	9.15	17.84	75.51

Table 16: Avg estimated true F1 scores over benchmarks (SciREX Easy, SciREX Medium and SciREX Hard) and models (Gemma-7B-it, Mistral, ChatGPT) across 2 widely used mapping formats (JSON and YAML). This table is the distillation result of averaging all the estimated true performance scores across models and benchmarks in Tab. 15.

1184 We select Gemma, Mistral, and ChatGPT for our evaluation, using two prompting techniques: Zero-shot
 1185 (ZS) and Zero-shot Chain-of-Thought (ZS-CoT) (§4.4). These models are tested on the SciREX dataset
 1186 across three difficulty levels: Easy, Medium, and Hard. Detailed prompt specifications are provided in
 1187 §E.4. We calculate FI_C , $SysE$, $EstTrueE$ with the results presented in Tab. 15. Furthermore, Tab. 16,

Tab. 17 and Tab. 18 are distilled from Tab. 15:

1. Tab. 16. For each model, we average its *EstTrueE* performance overall benchmarks and prompting techniques. For each task, we average the *EstTrueE* scores overall models and prompting techniques. This table is numerical results of Fig. 5 and discussed in §4.4.
2. Tab. 17. The purpose of this table is to compare the FI scores across formats. We average all the FI scores across models and tasks.
3. Tab. 18. The purpose of this table is to see whether CoT (Wei et al., 2022) mitigates format bias. We average all the *EstTrueE* scores over all models and benchmarks for each ZS and ZS-CoT prompting method.

Format instruction-following bias. From this Tab. 17, noticeably, there is a significant difference in the FI scores of two mapping formats (“JSON” and “YAML”). The average FI scores of the “JSON” mapping format is 66.07%, which is exceptionally higher (double time higher) than the one of the “YAML” mapping format (34.62%), resulting in high format bias ($BiasF_o = 494.64\%^2$). Overall, among models, ChatGPT yields the best result of 90.40%. Mistral fails to follow “YAML” format instruction, e.g. for the question in dataset **SciREX Easy** §D.1 with detailed prompt as in §E.4

Extract the entities reflecting the tasks in the following document:

Document: Published as a conference paper at ICLR 2017 QUERY - REDUCTION NETWORKS FOR QUESTION ANSWERING.

Your output must be in YAML format: “““ Task: [...] ”””. Wrap your final answer by <ANSWER> and </ANSWER>.

Mistral: “‘yaml
 - TASK_1: Prepare conference paper
 - TASK_2: Submit conference paper to ICLR 2017
 - TASK_3: Research and develop Query-Reduction Networks for Question Answering
 ””
 <ANSWER>
 “‘yaml
 - TASK_1: Prepare conference paper
 - TASK_2: Submit conference paper to ICLR 2017
 - TASK_3: Research and develop Query-Reduction Networks for Question Answering”’
 </ANSWER>.

Model	JSON	YAML	Avg.	$BiasF_o$
Gemma	20.71	12.62	16.66	32.68
Mistral	82.82	5.13	43.98	3018.00
ChatGPT	94.69	86.11	90.40	36.82
Avg.	66.07	34.62	50.35	494.64

Table 17: Avg FI scores over benchmarks and models across 2 widely used mapping formats (JSON and YAML). This table is the distillation result of averaging all the FI scores across models and benchmarks in Tab. 15.

ZS versus ZS-CoT. From Tab. 18, it is evident that the performance of ZS prompting surpasses that of ZS-CoT for both formats. Upon comparing the $BiasF_o$ across prompting techniques, we conclude that CoT (Wei et al., 2022) does not mitigate format bias.

	JSON	YAML	$BiasF_o$
Zero-shot	25.59	13.70	35.30
Zero-shot Chain-of-Thought	23.84	11.31	39.29

Table 18: Avg ZS and ZS-CoT scores over benchmarks and models across 2 widely used mapping formats (JSON and YAML). This table is the distillation results across models and benchmarks in Tab. 15.

Reliability of the results. From Tab. 15, 75% of the $EstTrue$ results are reliable. The reliability of the results in the mapping output format shows variability across different models and formats. Noticeably, “YAML” mapping format results are less reliable than “JSON” ones. On the other hand, ChatGPT illustrates its high reliability in all mapping formats while Mistral-7B-it and Gemma-7B-it are opposite, and all the results in the “YAML” mapping format of these models are unreliable.

C.5 Mitigating Format Bias Results

Index	Wrapping type	Special character	Bolding	Italicizing	Brackets	Parentheses	Placeholder	Quoting	Avg.	$BiasF_o$ (Var)
No demo (Zero-shot)										
1	ChatGPT (EstTrue-Acc)	54.63	67.39	44.76	59.79	57.82	66.23	68.28		235.33
2	ChatGPT (Systematic)	48.54	66.59	38.24	31.65	28.54	63.88	26.72		532.75
3	ChatGPT (FI)	88.84	98.80	85.43	52.93	49.36	96.44	39.13	72.99	61.12
Repeat format prompt thrice										
4	ChatGPT (EstTrue-Acc)	60.09	67.88	55.65	61.99	63.71	30.31	68.28		146.79
5	ChatGPT (Systematic)	56.65	66.98	49.93	35.74	51.63	2.85	33.13		377.66
6	ChatGPT (FI)	94.26	98.67	89.71	57.65	81.03	9.40	48.52	68.46	884.34
1 demo										
7	ChatGPT (EstTrue-Acc)	55.12	65.08	47.18	52.23	56.13	65.92	63.60		172.69
8	ChatGPT (Systematic)	50.54	64.49	43.98	40.02	31.02	62.19	28.10		397.62
9	ChatGPT (FI)	91.68	99.09	93.22	76.61	55.26	94.34	44.18	79.20	43.75
5 demos										
10	ChatGPT (EstTrue-Acc)	51.77	58.30	45.21	46.79	52.52	62.84	55.24		111.78
11	ChatGPT (Systematic)	51.18	56.66	40.69	41.36	39.78	60.88	27.72		259.37
12	ChatGPT (FI)	98.85	97.19	90.01	88.39	75.74	96.88	50.18	85.32	32.93
Finetuned										
13	ChatGPT (EstTrue-Acc)	74.02	74.73	71.53	73.88	74.09	74.27	74.19		0.71
14	ChatGPT (Systematic)	73.99	74.11	71.52	73.66	73.47	74.15	73.70		0.11
15	ChatGPT (FI)	99.96	99.17	99.98	99.69	99.16	99.83	99.33	99.59	0.93

Table 19: Supplementing demonstrations, repeating format instructions, and extra fine-tuning with formats’ data reduce format bias. Performance of ChatGPT on MMLU. All results are reliably measured by Thm. 2.1 i.e., inequality Eq. (6) holds.

In this section, we present the numerical results of our proposed techniques for mitigating format biases using ChatGPT on MMLU, as shown in Tab. 19.

- Demonstrations with formats reduce bias (Indexes 7-12).** From Tab. 19 indexes 7-12, we observe that using demonstrations with formats generally increases the average of FI scores, from 72.99% without any demonstration (index 3), to 79.20% with using one demonstration and 85.32% with using 5 demonstrations. Moreover, we find that the performance does not scale linearly with the FI score, indicating that simply increasing the FI score does not necessarily improve the models’ performance or reduce format biases.
- Repeating format instructions reduces format bias (Indexes 4-6).** From Tab. 19 index 6, most of the formats, repeating the format instruction can increase the FI score (compared to index 3), except for the “Placeholder”. Manual investigation reveals that repeatedly using the “Placeholder” format confuses the model about the actual location of the placeholder, leading to the model omitting the format. Nevertheless, this strategy generally reduces the format bias by decreasing the variance of results from formats other than “Placeholder”, leading to overall reduction.
- Fine-tuning with additional format data can eliminate format bias (Indexes 13-15).** Finetuning mostly eliminates the format bias problem of the LLM with the bias score only 0.71%² from Tab. 19 indexes 13-15, while increasing the average FI score up to almost perfect with 99.59%. This demonstrates that finetuning can help LLMs become more familiar with format tokens and requirements, reducing bias towards different formats.

D Experimental Details	1232
D.1 Dataset Details	1233
We provide descriptions of all datasets we use in this paper.	1234
MMLU (Hendrycks et al., 2021). MMLU is a benchmark for evaluating the performance of language models on Multiple Choices Question on a wide range of subjects across STEM, the humanities, social sciences, and other areas, testing the model’s ability to understand and reason in diverse domains.	1235 1236 1237
BBH (Suzgun et al., 2022). BBH is a MCQ dataset which includes a variety of challenging benchmarks that require advanced reasoning, comprehension, and other complex cognitive skills.	1238 1239
GSM8K (Cobbe et al., 2021). GSM8K is a dataset of 8,000 math word problems designed for grade school students. The problems require not just basic arithmetic but also multi-step reasoning to solve.	1240 1241
HotpotQA (Yang et al., 2018b). HotpotQA is a question-answering dataset with a focus on multi-hop reasoning. It contains questions that require finding and combining information from multiple Wikipedia articles to derive the answer.	1242 1243 1244
FairytaleQA (Xu et al., 2022b). FairytaleQA is a dataset designed for evaluating narrative comprehension, particularly in the context of children’s fairytales. It includes questions that test understanding of characters, plots, and settings in fairytales.	1245 1246 1247
SciDocsRR (Cohan et al., 2020). SciDocsRR is a dataset for evaluating information retrieval systems, particularly in the scientific domain. It includes tasks like citation prediction, document classification, and other retrieval-based evaluations.	1248 1249 1250
SemEval2017 (Augenstein et al., 2017b). SemEval2017 is part of an ongoing series of evaluations for semantic analysis in natural language processing. It includes a wide range of tasks such as sentiment analysis, semantic textual similarity, and information extraction.	1251 1252 1253
SciREX (Jain et al., 2020b). SciREX is a dataset for evaluating models on the task of information extraction from scientific literature. It focuses on extracting entities, relations, and other structured information from research papers.	1254 1255 1256
D.2 Experimental Results	1257
We present the hyperparameters setting for our experiments below.	1258
Gemma-7B-it (Team et al., 2024). For Gemma 7B-it, use the weights from Google and Huggingface ¹ . We use Nucleus Sampling (Holtzman et al., 2020) as our decoding strategy with a p value of 0.95, a temperature value of 0.1, and a window size of 1024.	1259 1260 1261
Mistral-7B-it-v0.2 (Jiang et al., 2023). For Mistral 7B-it, use the weights from MistralAI and Huggingface ² . We use Nucleus Sampling (Holtzman et al., 2020) as our decoding strategy with a p value of 0.9, and a window size of 1024.	1262 1263 1264
ChatGPT (gpt3.5-turbo-0125) (OpenAI, 2022). For ChatGPT, we use the system role: “You are helpful assistant!”. We set the “max_tokens” to be 1024, “top_p=1”, “frequency_penalty=0”, “presence_penalty=0”, and the model mode is “gpt3.5-turbo-0125”.	1265 1266 1267
Datasets for finetuning ChatGPT and finetuning setups. We preprocess the “auxiliary_train” ³ dataset of MMLU (Hendrycks et al., 2021), resulting in the training set of 6500 samples as discussed in §5. We preprocess a small, distinct validation set with the same ratio as the training set among formats “20-20-40-40-50-20-50”, resulting in a total of 240 samples for validation.	1268 1269 1270 1271
We use the default finetuning setup of OpenAI for ChatGPT. Our finetuning costs 63.86 US\$.	1272

¹<https://huggingface.co/google/gemma-7b-it>

²<https://huggingface.co/mistralai/Mistral-7B-Instruct-v0.2>

³https://huggingface.co/datasets/cais/mmlu/viewer/auxiliary_train

E Prompting

E.1 MCQ Prompt Details

The input for the models is the combination of the following components:

$$\text{Input} = \{\text{non-CoT/CoT} \times \text{Char./Text.}\} \text{ Instruction} + \text{Question} + (\text{CoT Wrapping})$$

where **non-CoT/CoT Instruction** shows that model uses Zero-shot or Chain-of-Thought, given that

- **non-CoT \times Char. Instruction** = “Answer the following multiple-choice question by outputting only the designated character identifier.”
- **non-CoT \times Text. Instruction** = “Answer the following multiple-choice question by outputting the textual value of your choice without the character identifier without any textual description.”
- **CoT \times Char. Instruction** = “Answer the following multiple-choice question step-by-step by outputting only the designated character identifier.”
- **CoT \times Text. Instruction** = “Answer the following multiple-choice question step-by-step by outputting the textual value of your choice without the character identifier.”

Question is the main content of the task and **CoT Wrapping** is wrapping instruction if using CoT. i.e. **CoT Wrapping** = “Wrap your final answer by <ANSWER> and </ANSWER>.”

E.2 Wrapping Prompt Details

The input for the models is the combination of the following components:

$$\text{Input} = \text{non-CoT/CoT Instruction} + \text{Question} + \text{Wrapping Format Instruction}$$

where **non-CoT/CoT Instruction** shows that model uses Zero-shot or Chain-of-Thought, given that

- If MCQ task (MMLU, BBH)
 1. **non-CoT Instruction** = “Answer the following multiple-choice question by outputting only the designated character identifier.”
 2. **CoT Instruction** = “Answer the following multiple-choice question step-by-step by outputting only the designated character identifier.”
- If generation task (GSM8K, HotpotQA, FairytaleQA)
 1. **non-CoT Instruction** = “Answer the following question.”
 2. **CoT Instruction** = “Answer the following question step by step.”

Question is the main content of the task, and **Wrapping Format Instruction** is the format we want the model to output, detailed as

- **Special Character wrapping** = “Wrap your final answer by <ANSWER> and </ANSWER>.”
- **Bolding wrapping** = “Wrap your final answer in bold by enclosing it with double asterisks.”
- **Italicizing wrapping** = “Wrap your final answer in italics by enclosing it with single asterisks.”
- **Brackets wrapping** = “Wrap your final answer using double square brackets.”
- **Parentheses wrapping** = “Wrap your final answer using double parentheses.”
- **Placeholder wrapping** = “Wrap your final answer by filling in the placeholder below: ‘So the answer is: {{placeholder}}’”
- **Quoting wrapping** = “Wrap your final answer using triple quotation marks.”

E.3 List Prompt Details 1310

For dataset **SciDocsRR**, the input for the models is the combination of the following components: 1311

$$\begin{aligned} \text{Input} = & \text{Information} + \text{Requirement} + \text{List Format Instruction} & 1312 \\ & + \text{non-CoT / CoT Instruction} + (\text{CoT Wrapping}) & 1313 \end{aligned}$$

where 1314

- **Information** = “Given a query, and a list of documents: Topic: **Topic**. List of documents: **Samples**” 1315
- **Requirement** = “You are required to transform the list of documents into a binary list of 1 or 0 where 1 indicates the document brings useful information to the topic, and 0 indicates the document does not bring useful information to the topic.” 1316
1317
1318
- **List Format Instruction** includes four categories: 1319
 1. **Python** = “Generate your binary list as a Python list” 1320
 2. **Bullet** = “Generate your binary list using bullet points” 1321
 3. **Special Character** = “Generate your binary list using <SEP> to separate elements” 1322
 4. **New Line** = “Generate your binary list such that each element is in a new line” 1323
- **non-CoT / CoT Instruction** includes: 1324
 1. **non-CoT Instruction** = “without any explanation.” 1325
 2. **CoT Instruction** = “step by step” 1326
- **CoT Wrapping** = “Wrap your final list by <ANSWER> and </ANSWER>.” 1327

For dataset **SemEval2017**, the input for the models is the combination of the following components: 1328

$$\begin{aligned} \text{Input} = & \text{Requirement} + \text{Document} + \text{List Format Instruction} & 1329 \\ & + \text{non-CoT / CoT Instruction} + (\text{CoT Wrapping}) & 1330 \end{aligned}$$

where 1331

- **Requirement** = “Extract a list of keyphrases from the following document:” 1332
- **Document** is the main content of the task. 1333
- **List Format Instruction** includes four categories: 1334
 1. **Python** = “Generate your binary list as a Python list” 1335
 2. **Bullet** = “Generate your binary list using bullet points” 1336
 3. **Special Character** = “Generate your binary list using <SEP> to separate elements” 1337
 4. **New Line** = “Generate your binary list such that each element is in a new line” 1338
- **non-CoT / CoT Instruction** includes: 1339
 1. **non-CoT Instruction** = “without any explanation.” 1340
 2. **CoT Instruction** = “step by step” 1341
- **CoT Wrapping** = “Wrap your final list by <ANSWER> and </ANSWER>.” 1342

1343 E.4 Mapping Prompt Details

1344 For all three datasets, we use the following formula for the input of the models

$$1345 \text{Input} = \text{Requirement} + \text{Document} + \text{Mapping Format Instruction} + (\text{CoT Wrapping})$$

1346 where

- 1347 • **Requirement** = “Extract the entities reflecting the tasks in the following document:” if using non-
1348 CoT model and “Extract the entities reflecting the tasks in the following document step-by-step:” if
1349 using CoT model
- 1350 • **Document** is the main content of the task.
- 1351 • **CoT Wrapping** = “Wrap your final list by <ANSWER> and </ANSWER>.”
- 1352 • **Mapping Format Instruction** starts with defining a specific format for the model and then instructs
1353 the model to follow. In detail, we have

1354 1. For **Easy** dataset, we define:

```
1355     1 JSON_FORMAT = {  
1356     2     'Task': [...]  
1357     3     }  
1358     4
```

```
1359     1 YAML_FORMAT = ''' Task: [...] '''  
1360     2
```

1361 Then

- 1362 – **JSON Mapping** = “Your output must be a Python dictionary with the key ‘Task’ and value
1363 as a list of task name entities: {str(JSON_FORMAT)}”
- 1364 – **YAML Mapping** = “Your output must be in YAML format: {str(YAML_FORMAT)}”

1365 2. For **Medium** dataset, we define:

```
1366     1 JSON_FORMAT = {  
1367     2     'Task': [...],  
1368     3     'Method': [...]  
1369     4     }  
1370     5
```

```
1371     1 YAML_FORMAT = '''  
1372     2     Task: [...]  
1373     3     Method: [...]  
1374     4     '''  
1375     5
```

1376 Then

- 1377 – **JSON Mapping** = “Your output must be a Python dictionary with the keys ‘Task’
1378 and ‘Method’, and value is a list of task name entities and method name entities:
1379 {str(JSON_FORMAT)}”
- 1380 – **YAML Mapping** = “Your output must be in YAML format: {str(YAML_FORMAT)}”

1381 3. For **Hard** dataset, we define:

```
1382     1 JSON_FORMAT = {  
1383     2     'Task': [...],  
1384     3     'Method': [...],  
1385     4     'Material': [...],  
1386     5     'Metric': [...]  
1387     6     }  
1388     7
```

```
1     YAML_FORMAT = '''
2         Task: [...]
3         Method: [...]
4         Material: [...]
5         Metric: [...]
6         '''
7
```

Then 1396

- **JSON Mapping** = “Your output must be a Python dictionary with the keys are ‘Task’, ‘Method’, ‘Material’, ‘Metric’, and value is a list of task name entities, method name entities, material name entities, metric name entities: {str(JSON_FORMAT)}” 1397-1399
- **YAML Mapping** = “Your output must be in YAML format: {str(YAML_FORMAT)}” 1400