Format-Adapter: Improving Reasoning Capability of LLMs by Adapting Suitable Format

Anonymous ACL submission

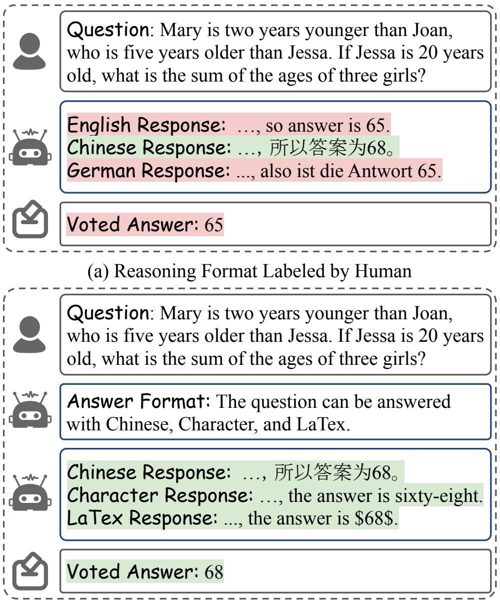
Abstract

Generating and voting multiple answers is an effective method to mitigate reasoning inconsistencies of large language models (LLMs). Prior works have shown that multiple reasoning formats outperform a single format when generating multiple answers. However, previous works using multiple formats rely on formats labeled by humans, which could be unsuitable for all tasks and have high labeling costs. To address this issue, we adapt suitable formats to the given tasks by generating and selecting formats. We first propose how to measure the reasoning error when generating multiple answers. Then, we introduce FORMAT-ADAPTER, which utilizes LLMs to generate and select suitable reasoning formats by minimizing the error measurement we present. We conduct experiments on math and commonsense reasoning tasks, where FORMAT-ADAPTER achieves a 4.3% performance improvement on average over previous works, demonstrating the effectiveness.

1 Introduction

The prior research has revealed that, due to the inconsistency, one question could yield different responses when suffering minor variations in the input or parameters, resulting in incorrect results (Wang et al., 2022). To address this issue, previous works propose to generate multiple responses to mitigate the impact of model inconsistencies (Wang et al., 2023; Yao et al., 2023; Besta et al., 2024). Specifically, such methods generate multiple answers to a given question by varying parameters and then select the most appropriate response as the final answer by scoring and voting.

However, the above works rely on the fixed *reasoning format*[1], which limits the model performance since different questions could suit different reasoning formats (Cheng et al., 2023; Chen et al., 2023; He et al., 2024), as shown in Figure 1. Therefore, many prior works try to enhance the reasoning performance by employing various reasoning formats (Luo et al., 2024; Zhang et al., 2024b). For example, CLSP (Qin et al., 2023) proposes using varied natural languages to generate different answers in numerical reasoning tasks. Similarly, FlexTaF (Zhang et al., 2024a) addresses table reasoning tasks by generating different answers through diverse table formats.



Question: Mary is two years younger than Joan, who is five years older than Jessa. If Jessa is 20 years old, what is the sum of the ages of three girls?

English Response: …, so answer is 65.
Chinese Response: …, 所以答案为68。
German Response: ..., also ist die Antwort 65.

Voted Answer: 65

(a) Reasoning Format Labeled by Human

Question: Mary is two years younger than Joan, who is five years older than Jessa. If Jessa is 20 years old, what is the sum of the ages of three girls?

Answer Format: The question can be answered with Chinese, Character, and LaTex.

Chinese Response: …, 所以答案为68。
Character Response: …, the answer is sixty-eight.
LaTex Response: ..., the answer is \$68\$.

Voted Answer: 68

(b) Reasoning Format Generated by LLM

Figure 1: The comparison between the previous work (a) and our method (b) instructed to reason with different formats. The red parts denote the incorrect answers and the green parts denote the correct ones. The previous work employs the formats labeled by humans, which could be not suitable for the given question and LLM. Our method generates and selects suitable formats, achieving better performance.

However, the above methods rely on manually designed reasoning formats, which have the following issues: *(i)* Manually designed formats could **not be suitable for the task**; *(ii)* Manually designing formats for each task **incurs significant overhead**. To address these issues, in this paper: *(i)* We dis-

[1] In this paper, we define the *reasoning format* as LLMs how to present the reasoning process.

cuss ***why adapting multiple formats outperforms using a single format*** during reasoning; *(ii)* We propose ***using LLMs to generate and select suitable formats*** to enhance reasoning performance.

We first propose how to measure the error of the reasoning with multiple responses. Based on the measurement, we discuss that generating with a single format can only enhance reasoning robustness while using multiple formats can further enhance reasoning capabilities. Then, we propose FORMAT-ADAPTER, which utilizes LLMs to generate and select suitable reasoning formats. We use LLMs to derive reasoning formats without human involvement, lowering the overhead of the format design. Besides, we propose to adapt reasoning formats by reducing the error measurement we present, ensuring that the format is suitable for the task.

To evaluate the effectiveness of FORMAT-ADAPTER, we adapt our method to two mainstream reasoning tasks: math reasoning (GSM8K (Cobbe et al., 2021), MATH (Saxton et al., 2019)) and commonsense reasoning (ARC-Challenge (Yadav et al., 2019), GPQA (Yadav et al., 2019)). The experimental results show that, compared with baselines with the single format, FORMAT-ADAPTER brings 4.1% performance improvement on average, proving the effectiveness of FORMAT-ADAPTER. We also compare FORMAT-ADAPTER with baselines using multiple reasoning formats, where our method brings 4.7% improvement on average, showing the necessity of the format selection.

Our contributions are as follows:

- To shed light on further research, we discuss why generating multiple answers with multiple formats outperforms single format;
- To enhance the reasoning ability of LLMs, we present FORMAT-ADAPTER, which generates and selects suitable formats using LLMs;
- Experiments show that our method brings 4.3% improvement on average over all baselines, showing the effectiveness of FORMAT-ADAPTER.

2 Preliminaries

To prove the effectiveness of employing multiple reasoning formats and shed light on future research, in this section, we discuss: *(i)* How to measure the error of reasoning with a single reasoning format of LLMs; *(ii)* How to measure the error of reasoning employing multiple reasoning formats of LLMs and why it outperforms using the single format.

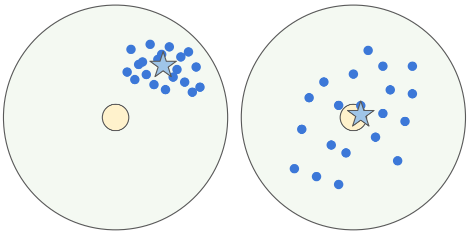


Figure 2: The comparison between using the single format (left) and multiple formats (right) with the same number of generated answers. The yellow ○ denotes the correct answer, the blue ● denotes different predictions, and the blue ☆ denotes the average prediction.

2.1 Error of Single Reasoning Format

First, we discuss the error of the general ensemble method (Sagi and Rokach, 2018), since generating multiple responses and voting can be regarded as an ensemble method. In this paper, we use the error function $L(x, y)$ as follows:

$$L(x,y) = \begin{cases} 1 & \text{if } x \neq y \\ 0 & \text{if } x = y \end{cases} \tag{1}$$

The function L represents whether the prediction result matches the correct answer exactly. We define $D = \{(x_i, y_i)\}_{|D|}$ as the experimental dataset, $\{\phi_i\}_m$ as the set of m predictors, and $\bar{\phi} = \mathsf{avg}(\phi_{im})$ as the ensemble predictors. Proved by Wood et al. (2024), the error of ensemble learning with m predictors on the dataset D can be expressed as the error over the dataset minus the divergence among the individual predictors, that is:

$$\begin{aligned} \mathbb{E}_D\left[L(\bar{\phi}, y)\right] = \frac{1}{m}\sum_{i=1}^{m} \mathbb{E}_D\left[L\left(\phi_i, y\right)\right] \\ - \mathbb{E}_D\left[\frac{1}{m}\sum_{i=1}^{m} L\left(\phi_i, \bar{\phi}\right)\right] \end{aligned} \tag{2}$$

Then we discuss the error of generating multiple responses using LLMs. For a model employing a single reasoning format, we assume the used format is f and the model is ϕ. Since only parameters (e.g., random seed, temperature) are altered during reasoning, we can regard the predictor as applying a perturbation to the model inherent performance $\phi \circ \mathsf{f}$, expressed as $\phi_i = \phi \circ \mathsf{f} + \delta_i$, where δ_i denotes the perturbation. It can be derived that generating one single answer using ϕ_i can be present as:

$$\mathbb{E}_D\left[L(\phi_i, y)\right] = \mathbb{E}_D\left[L\left(\phi \circ \mathsf{f} + \delta_i, y\right)\right] \tag{3}$$

We assume an ideal scenario where the average of all predictors represents the inherent performance of the model, i.e., $\lim_{m\to\infty} \bar{\phi} = \phi \circ \mathsf{f}$. It can be proven that the error in generating multiple answers using a single reasoning format satisfies:

$$\mathbb{E}_D\left[L(\bar{\phi}, y)\right] = \mathbb{E}_D\left[L\left(\phi \circ \mathtt{f}, y\right)\right] \quad (4)$$

Appendix A.1 presents the prove of Equation 4. It can be seen that, compared with Equation 3, generating multiple answers can eliminate the perturbation δ, enhancing the robustness. However, when using single format f, Equation 4 is determined by ϕ, showing that enhancing performance relies on improving the model capability.

2.2 Error of Multiple Reasoning Format

In the following, we discuss the error of using multiple reasoning formats and why it outperforms the single format. During reasoning, we employ multiple formats $\{f_i\}_m$ with one single model ϕ, so we can assume the predictors to be $\phi_i = \phi \circ \mathtt{f}_i + \delta_i$. It can be proved that the reasoning error follows:

$$\begin{aligned}\mathbb{E}_D\left[L(\bar{\phi}, y)\right] &= \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_D\left[L(\phi \circ \mathtt{f}_i, y)\right] \\ &- \mathbb{E}_D[\frac{1}{m}\sum_{i=1}^{m}L(\phi \circ \mathtt{f}_i, \bar{\phi})]\end{aligned} \quad (5)$$

The prove of Equation 5 can be seen in Appendix A.2. In the equation, the first term denotes the average error over all predictions and correct answers, and the second term measures the divergence between different reasoning formats. It can be observed that, even with the same model, we can combine different reasoning formats to minimize error, thereby improving performance, as shown in the right part of Figure 2.

3 Methodology

This section introduces FORMAT-ADAPTER, which leverages LLMs to generate and select the suitable reasoning formats to enhance reasoning performance. An illustration of our method is shown in Figure 3. The prompts we used are provided in Appendix B. ***We also discuss the efficiency of* FORMAT-ADAPTER *in Appendix E.***

3.1 Format Generation

This step is designed to generate reasoning formats, ensuring both the relevancy and diversity of the generated formats. Relevancy means that the generated reasoning formats are relevant to the given task. Diversity demands that the generated reasoning formats be varied to ensure suitable reasoning formats for various user questions.

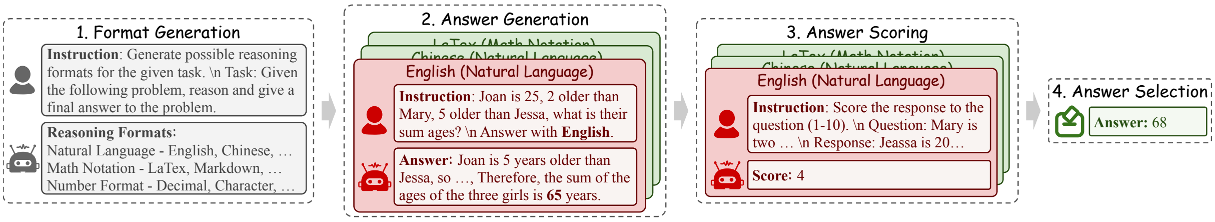
To ensure relevancy, an example task is provided during generation to help LLMs learn how to produce task-relevant reasoning formats. To ensure diversity, we design the instruction to ask LLMs to generate reasoning formats across multiple categories, where each category consists of multiple formats. For instance, as shown in Figure 3, *Natural Language* is the reasoning format category, while *English* and *Chinese* are the reasoning formats of this category. In summary, the input includes the task definition, and an example of the task, while the output consists of multiple reasoning formats. Appendix D discusses the generated formats under each setting.

3.2 Answer Generation

This step generates corresponding answers for each generated reasoning format. First, the instruction is rewritten according to each reasoning format to ensure that the answer generation follows the given reasoning format. We take the original instruction of the task (Appendix B) and the reasoning format as input and ask LLMs to output the rewritten instruction based on the reasoning format. Then, the rewritten instruction is used to generate different reasoning answers for the given user question. Following prior work (Qin et al., 2023), zero-shot learning is applied by inputting the rewritten instruction and the user question to output answers in the specified reasoning format.

3.3 Answer Scoring

After obtaining answers in different reasoning formats, based on Equation 5, we aim to select the suitable reasoning formats that minimize the error. However, in Equation 5, the error between the prediction and the answer $L(\phi \circ f_i, y)$ is difficult to compute, as the correct answer y is unknown. Therefore, we use answer generation LLMs to score the answers in each reasoning format to estimate the probability that the predicted answer is correct. Following Zheng et al. (2023), we input the user question and the predicted answer, outputting a score from 1 to 10 to represent the degree to the probability that the answer is correct. To ensure that there is the same scale between the first term and the second term of Equation 5 during the calculation, we divide the rating by 10 to correspond to the interval of $[0, 1]$.



Instruction: Generate possible reasoning formats for the given task. \n Task: Given the following problem, reason and give a final answer to the problem.

Reasoning Formats:
Natural Language - English, Chinese, …
Math Notation - LaTex, Markdown, …
Number Format - Decimal, Character, …

Instruction: Joan is 25, 2 older than Mary, 5 older than Jessa, what is their sum ages? \n Answer with **English**.

Answer: Joan is 5 years older than Jessa, so …, Therefore, the sum of the ages of the three girls is **65** years.

Instruction: Score the response to the question (1-10). \n Question: Mary is two … \n Response: Jeassa is 20…

Score: 4

Answer: 68

Figure 3: The pipeline of FORMAT-ADAPTER, which consists of: *(i) Format Generation*: Generate possible reasoning formats of the given task; *(ii) Answer Generation*: Generate the answer using each reasoning format; *(iii) Answer Scoring*: Score whether each generated answer is correct using LLMs; *(iv) Answer Selection*: Select the final answer with Equation 5. Red and green represent the reasoning formats of incorrect and correct respectively.

3.4 Answer Selection

Based on the predicted answers and corresponding scores of different reasoning formats, we discuss how to select the final answers based on Equation 5. Specifically, given the dataset D and the model ϕ, we hope to find suitable reasoning formats to minimize the error that satisfies:

$$\{\mathtt{f}_i\}_n = \underset{\{\mathtt{f}_i\}_n \subseteq \{\mathtt{f}_i\}_m}{\arg\min} \; \mathbb{E}_D \left[L(\bar{q}, y) \right] \qquad (6)$$

In Equation 5, the first term can be directly calculated by averaging scores obtained in §3.3. The second term requires calculating the average difference between all results and the average result, where we take the average prediction $\bar{\phi}(x)$ as the answer appearing most frequently among all outcomes. Considering computational efficiency, we adopt a greedy algorithm to select formats: we add each format $\mathtt{f}_i$ one by one to the selected results, where if the value of Equation 5 decreases, we retain $\mathtt{f}_i$; otherwise, we remove $\mathtt{f}_i$. Due to the inherent scoring errors of LLMs, we do not directly use the answer with the highest score within the selected set. Instead, we choose the most frequently occurring answer as the final answer.

4 Experiment

4.1 Experimental Setup

4.1.1 Datasets

To validate the effectiveness of our method, following Dubey et al. (2024), we conduct experiments on two mainstream reasoning tasks: commonsense reasoning (ARC-Challenge-Hard (Yadav et al., 2019), GPQA (Rein et al., 2024)) and math reasoning (GSM8K (Cobbe et al., 2021), MATH (Saxton et al., 2019)). Commonsense reasoning requires the model to apply commonsense knowledge to comprehend and answer questions. On the other hand, math reasoning demands the model to solve mathematical problems.

Due to the high cost of generating multiple answers, we employ the subsets of the above benchmarks to reduce computational overhead while maintaining a robust performance evaluation. Specifically, for GSM8K and ARC-Challenge (ARC-C), we sample 256 questions that are not well solved by the current LLMs, referred to as GSM8K-Hard and ARC-C-Hard, respectively. For MATH, we utilize the version of MATH500 (Lightman et al., 2024), which samples 500 questions from the original dataset. For GPQA, we employ the original test set, comprising 448 questions.

4.1.2 Models

We conduct the experiments with the models of Llama3.1-Instruct (Dubey et al., 2024) and GPT-4o (OpenAI et al., 2024). Llama3.1 is one of the most mainstream and high-performing open-source LLMs. GPT-4o, on the other hand, currently represents one of the most powerful LLMs in terms of reasoning capabilities. Our selection ensures coverage of diverse application scenarios.

4.1.3 Baselines

To better reflect the effectiveness of FORMAT-ADAPTER, we compare our method with two types of baselines. The first type uses the single reasoning format, including Single, Self-Consistency (SC) (Wang et al., 2023), Tree-of-Thought (ToT) (Yao et al., 2023), and DTV (Zhou et al., 2024). Another type uses multiple reasoning formats, including CLSP (Qin et al., 2023), MultiPoT (Luo et al., 2024), and FlexTaF (Zhang et al., 2024a). We introduce the above baselines in Appendix C.

4.1.4 Metrics

We use Exact Match (EM) as the evaluation metric for all datasets, which measures whether the predicted result is completely identical to the ground truth. Additionally, we evaluate methods that generate multiple answers under two settings: Vote and Oracle. Vote refers to selecting one answer from all generated answers as the final result, reflecting the actual performance of the method. Oracle, on the other hand, considers a question correct if there exists one of the generated answers matches the ground truth, reflecting the performance upper bound of the method.

4.1.5 Implement Details

Following the previous work (Qin et al., 2023), we evaluate FORMAT-ADAPTER using zero-shot. The numbers and types of reasoning formats of FORMAT-ADAPTER under different settings can be seen in Appendix D. The parameter settings of FORMAT-ADAPTER are consistent with Single to ensure comparable results. To verify the robustness of FORMAT-ADAPTER, we show the average result with five running in Appendix F.3.

4.2 Main Experiment

4.2.1 Baselines with Single Format

The experimental results of FORMAT-ADAPTER compared with baselines using the single reasoning format (Appendix C) are shown in Table 1. The table shows that, compared with the best baseline results under each setting, FORMAT-ADAPTER brings 4.1% performance improvement on average, showing the effectiveness of FORMAT-ADAPTER. We also compare the efficiency across different methods in Appendix E.3. Besides, from Table 1, we can also observe that:

Model The improvement brought by FORMAT-ADAPTER on different models depends on the difficulty of the dataset. For relatively simple datasets like GSM8K and ARC-Challenge, our method demonstrates more significant improvements on models with a small scale. Conversely, for more challenging datasets such as MATH and GPQA, our method achieves more notable improvements on larger models. This is because, for complex datasets, smaller models lack the necessary knowledge to solve such problems due to their limited scale, where simply altering the reasoning format cannot introduce new knowledge, leading to negligible performance gains. On the other hand, models with larger scales already exhibit strong performance for simpler datasets, making the improvements brought by our method less pronounced compared to smaller models.

Metric FORMAT-ADAPTER demonstrates performance improvements in both the Vote and Oracle settings, indicating that our method not only enhances actual performance but also effectively encourages the model to utilize diverse reasoning formats to generate correct answers. These results also confirm that the most suitable reasoning format varies across different types of questions. However, there remains a significant performance gap between the Vote and Oracle settings in our method, which can be attributed to the following reasons: *(i)* The scoring quality of LLMs is suboptimal, making it challenging to accurately assess whether a predicted result is correct; *(ii)* Specifically, for datasets such as ARC-Challenge and GPQA, where the answers are choices, LLMs could produce correct results while following incorrect reasoning processes, resulting in lower scores, which can also explain why the performance gap between Vote and Oracle is larger on the commonsense reasoning datasets compared to math reasoning datasets.

4.2.2 Baselines with Multiple Format

The performance comparison between FORMAT-ADAPTER and baselines employing multiple reasoning formats is presented in Table 2. Following the setup of MultiPoT, we select 263 problems from MATH500 that can be resolved using code-based solutions. About FlexTaF, we conduct experiments on the WikiTQ dataset (Pasupat and Liang, 2015), which is the mainstream benchmark of the table QA task. As observed from the table, FORMAT-ADAPTER achieves an average improvement of 4.7% over the best performance of the baselines under each setting, demonstrating the effectiveness of our method.

4.3 Ablation Study

To demonstrate the effectiveness of each step of FORMAT-ADAPTER, we conduct ablation studies to verify that FORMAT-ADAPTER is the most effective under different answer selection strategies. The experimental results are shown in Table 3. From the table, we can observe that removing any individual step results in a performance decline, thereby validating the importance of each step in FORMAT-ADAPTER. Furthermore, the table reveals the following insights: *(i)* Removing the Select step leads to the most significant performance drop, indicating that in many questions, only a few reasoning formats yield correct answers, necessitating the Select step to identify the correct solutions; *(ii)* The performance degradation of the ablation study is more pronounced in smaller-scale models compared to larger ones, which suggests that models

Model	Method	GSM8K-Hard		MATH500		ARC-C-Hard		GPQA	
		Vote	Oracle	Vote	Oracle	Vote	Oracle	Vote	Oracle
Llama3.1-8b	Single	23.0	–	47.8	–	16.8	–	28.1	–
	SC	36.7	55.1	51.4	63.0	18.4	23.4	30.8	59.2
	ToT	43.0	53.9	52.8	56.8	24.2	33.8	32.8	46.7
	DTV	51.6	56.2	55.4	56.4	39.1	42.6	32.6	50.0
	Format-Adapter	**54.7**	**89.8**	**56.8**	**75.0**	**57.4**	**91.4**	**33.9**	**93.8**
Llama3.1-70b	Single	66.0	–	63.4	–	68.0	–	43.1	–
	SC	70.3	77.3	64.4	72.8	69.1	69.9	46.2	66.5
	ToT	71.5	77.6	67.2	75.2	70.7	72.3	48.0	73.2
	DTV	71.7	84.3	65.8	81.8	69.9	73.8	50.2	75.9
	Format-Adapter	**76.2**	**94.9**	**70.4**	**85.4**	**71.5**	**88.7**	**51.0**	**96.4**
GPT-4o	Single	73.4	–	71.0	–	77.0	–	48.9	–
	SC	74.1	82.8	71.4	83.2	78.9	83.2	49.1	70.8
	Format-Adapter	**78.4**	**95.1**	**76.8**	**86.6**	**80.1**	**96.9**	**51.6**	**96.6**

Table 1: EM of Format-Adapter and baselines using the single reasoning format. The best results of each setting are marked in **bold**. Due to the limitations of computing resources, we only compare the performance of Format-Adapter with Self-Consistency on GPT-4o.

Dataset	Method	8b		70b	
		Vote	Oracle	Vote	Oracle
MATH	CLSP	53.0	66.2	66.9	74.9
	MultiPoT	57.4	66.5	72.2	79.1
	Ours	**60.1**	**88.6**	**77.2**	**88.6**
WikiTQ	FlexTaF	38.0	70.3	41.5	79.0
	Ours	**55.2**	**79.7**	**63.1**	**84.0**

Table 2: EM of Format-Adapter (Ours) and baselines using multiple reasoning formats on Llama3.1. The best results of each setting are marked in **bold**.

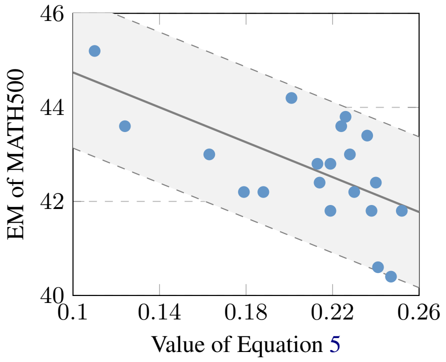


Figure 4: The performance on MATH with different formats using Llama3.1-8b. Different blue ● denotes the result using different formats, where the formats used are randomly sampled from that generated by Format-Adapter. The correlation coefficient is -0.652.

with smaller scales generate fewer reasoning formats capable of producing correct answers, relying more heavily on the Rewrite and Select steps to achieve correct results.

4.4 Analysis

In this section, we adapt analysis experiments to understand better how Format-Adapter improves the reasoning performance and to guide the parameter selection. Due to the high reasoning cost, we only employ Llama3.1 as the experimental LLMs. We also adapt the case study to understand better how Format-Adapter improves the performance, which is discussed in Appendix G.

4.4.1 Reasoning Error

To demonstrate that Equation 5 effectively reflects the reasoning error, we conduct statistical analysis on MATH to evaluate the model performance corresponding to different values of Equation 5. The experimental results are shown in Figure 4, from which we can observe that: *(i)* As the value of Equation 5 gradually increases, the model performance consistently declines, indicating that the error is indeed progressively growing; *(ii)* The error obtained in Figure 4 is predominantly concentrated around 0.22, suggesting that most reasoning formats yield similar results, while these results are inferior to the best results, indicating that the majority of reasoning formats do not produce the correct answers, showing the necessity to select the most suitable reasoning format for each question.

4.4.2 Reasoning Format Category

To examine the performance of different reasoning formats and inspire future work, we analyze the average performance improvement achieved under various settings with different reasoning format categories. We also list the most suitable reasoning format for each task in Appendix D. The results are shown in Figure 5, from which we can see that: *(i)* For the results using the single reasoning format category, its performance improvement is determined by the variation in answers gener-

Model	Method	GSM8K-Hard	MATH500	ARC-C-Hard	GPQA
Llama3.1-8b	FORMAT-ADAPTER	54.7	56.8	57.4	33.5
	- Rewrite	51.6 (-3.1)	53.6 (-3.2)	52.0 (-5.4)	32.4 (-1.1)
	- Select	49.2 (-5.5)	47.8 (-9.0)	54.7 (-2.7)	25.4 (-8.1)
	- Score	53.9 (-0.8)	54.2 (-2.6)	52.7 (-4.7)	30.1 (-3.4)
Llama3.1-70b	FORMAT-ADAPTER	76.2	70.4	71.5	51.0
	- Rewrite	75.4 (-0.8)	69.6 (-0.8)	68.8 (-2.7)	47.1 (-3.9)
	- Select	75.0 (-1.2)	68.6 (-1.8)	66.4 (-5.1)	44.2 (-6.8)
	- Score	73.8 (-2.4)	70.2 (-0.2)	69.9 (-1.6)	47.3 (-3.7)

Table 3: The ablation study results under: *(i)* Rewrite: Generate answers without rewriting instructions; *(ii)* Select: Vote the answer from the responses with the highest score; *(iii)* Score: Set all answers with the same score of 1.0.

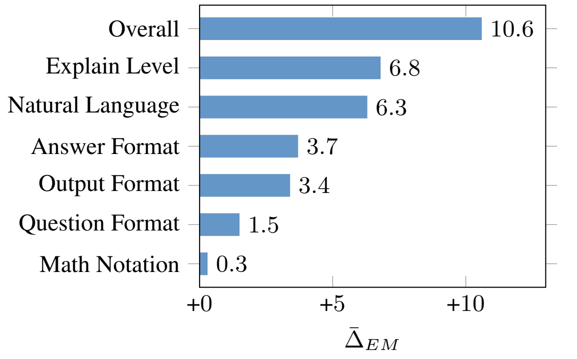


Figure 5: The average improvement brought by FORMAT-ADAPTER with different reasoning categories having more than four formats. $\bar{\Delta}_{EM}$ denotes the average improvement compared to Self-Consistency. Overall denotes using all reasoning categories.

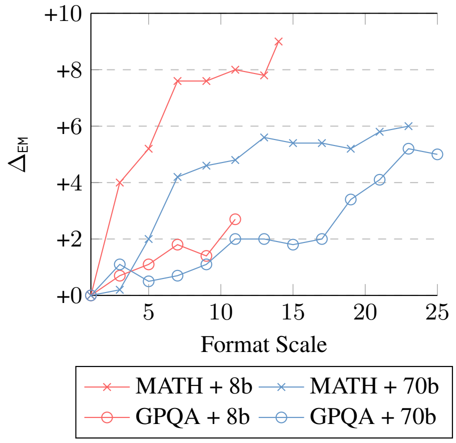


Figure 6: The average performance improvement brought by FORMAT-ADAPTER on MATH and GPQA using Llama3.1. Δ_{EM} denotes the EM improvement compared with the result using the single format.

ated by the corresponding formats of this category. For the categories with low improvements (e.g., Math Notation), the answers across different formats are largely similar, with performance close to that of Self-Consistency. In contrast, reasoning categories with higher performance improvements (e.g., Explain Level) exhibit greater variability in the answers generated by different formats, making it more likely to include the correct result; *(ii)* Even for the best-performing single category, its performance improvement is still lower than that achieved by using all reasoning categories (Overall), which indicates that the most suitable reasoning formats vary across questions, and combining different reasoning categories and formats during reasoning is necessary to achieve optimal results.

4.4.3 Reasoning Format Scale

Considering the computational resource limitations in practical applications, we evaluate the performance of FORMAT-ADAPTER under different scales of reasoning formats. The experimental results, as shown in Figure 6, reveal that performance consistently improves across different settings as the scale of formats increases, demonstrating the necessity of incorporating more reasoning formats to enhance performance. Besides, when a small number (< 5) of formats are used, FORMAT-ADAPTER also brings a significant improvement, proving the effectiveness of FORMAT-ADAPTER under low computational resources.

Additionally, the performance improvement in each setting follows a trend: it initially increases significantly, then stabilizes, and finally experiences another notable rise. This phenomenon can be explained as follows: *(i)* Initially, the primary performance bottleneck lies in the inconsistency of LLMs, where increasing the number of reasoning formats enhances the robustness of reasoning, thereby improving the performance. *(ii)* Once using a sufficient number of reasoning formats, the performance bottleneck shifts to whether the reasoning formats are suitable for the user question, where adding new formats makes it more likely that the format is suitable for the question, leading to further performance improvements.

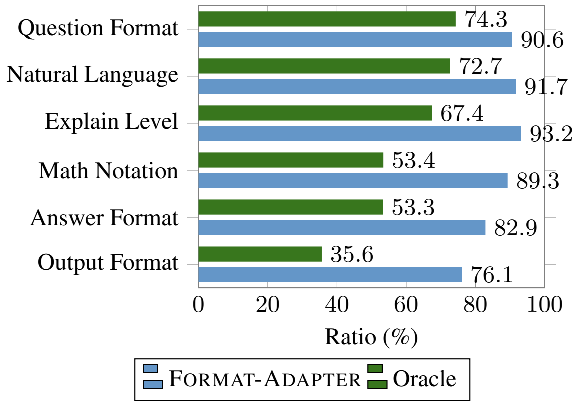


Figure 7: The average ratio over all datasets of each reasoning format category that is selected by FORMAT-ADAPTER (blue) and that contains the format that can solve the question correctly (Oracle, green).

4.4.4 Format Selection Ratio

To better understand the impact of different reasoning formats on reasoning performance, we compute the ratio of formats selected by FORMAT-ADAPTER or containing the correct answer, as shown in Figure 7. From the figure, we can observe the following: *(i)* For different reasoning formats, the ratio selected by FORMAT-ADAPTER follows a trend similar to that of Oracle, indicating that FORMAT-ADAPTER tends to select the appropriate formats, i.e., those that contain the correct answer, thus demonstrating the effectiveness of FORMAT-ADAPTER; *(ii)* Compared to the average performance of Oracle with FORMAT-ADAPTER in Table 1 (89.4%), the best single format still shows a performance gap of 15.1%, indicating that different questions suit different formats, suggesting that multiple formats are necessary during reasoning. *(iii)* FORMAT-ADAPTER selects a relatively high proportion ($> 70\%$) for each category, indicating that LLMs tend to assign higher scores during the Score step, resulting in that FORMAT-ADAPTER selecting many categories that do not contain the correct answer, suggesting the need for further improvement in the scoring method in future work.

5 Related Works

Previous studies have shown that LLMs could exhibit inconsistency during reasoning, producing inconsistent answers when faced with input or parameter perturbations (Adiwardana et al., 2020; Camburu et al., 2020; Elazar et al., 2021). To address this issue, Wang et al. (2023) proposes Self-Consistency, which generates multiple outputs for the same input and selects the final answer through voting, thereby reducing the impact of perturbations. Subsequent works have sought to improve upon Self-Consistency to enhance the performance further (Li et al., 2024; Besta et al., 2024; Wang et al., 2024). For example, Tree-of-Thought (Yao et al., 2023) decomposes the reasoning process and ensures consistency at each reasoning step as a tree, while DTV (Zhou et al., 2024) employs Isabelle formalism to represent answers, improving the accuracy of answer selection. Notably, many studies have demonstrated that employing diverse reasoning formats to generate answers outperforms relying on a single format to produce multiple outputs (Zhang et al., 2024a,b; He et al., 2024). For instance, CLSP (Qin et al., 2023) uses different natural language formulations to generate answers, and MultiPoT (Luo et al., 2024) leverages multiple programming languages for answer generation.

However, the above methods rely on predefined reasoning formats manually annotated by humans, which can be inefficient and suboptimal, as the most suitable reasoning format varies across questions. To address this limitation, we first analyze why utilizing multiple reasoning formats outperforms single-format reasoning and propose an optimization objective based on this insight. Guided by this objective, we leverage LLMs to generate and select the most suitable reasoning format, thereby reducing the cost of human annotations and improving reasoning performance.

6 Conclusion

In this paper, we propose FORMAT-ADAPTER, which generates multiple answers using different reasoning formats, reducing inconsistencies and improving the performance of LLMs. First, we present how to measure reasoning errors when generating multiple answers, showing that multiple reasoning formats outperform a single format. Then, we present FORMAT-ADAPTER, which uses LLMs to generate and select the suitable reasoning formats, improving reasoning performance by reducing the error measurement we present. We conduct experiments on math and commonsense reasoning, where FORMAT-ADAPTER improves performance by an average of 4.3% compared to previous methods, demonstrating its effectiveness. We also analyze the relationship between our error measurement and performance, showing a negative correlation that confirms its accuracy in measuring reasoning errors when generating multiple answers.

Limitations

(i) We have not yet experimented with FORMAT-ADAPTER on more tasks, such as question answering and code generation, where in the future, we will apply FORMAT-ADAPTER to a wider range of tasks to further demonstrate its effectiveness; *(ii)* Generating multiple answers incurs significant computational overhead, where in future work, we will explore ways to reduce the computational cost while maintaining or even improving reasoning performance.

Ethics Statement

All datasets and models used in this paper are publicly available, and our usage follows their licenses and terms.

References

Daniel Adiwardana, Minh-Thang Luong, David R. So, Jamie Hall, Noah Fiedel, Romal Thoppilan, Zi Yang, Apoorv Kulshreshtha, Gaurav Nemade, Yifeng Lu, and Quoc V. Le. 2020. Towards a human-like open-domain chatbot. *Preprint*, arXiv:2001.09977.

Maciej Besta, Nils Blach, Ales Kubicek, Robert Gerstenberger, Michal Podstawski, Lukas Gianinazzi, Joanna Gajda, Tomasz Lehmann, Hubert Niewiadomski, Piotr Nyczyk, and Torsten Hoefler. 2024. Graph of thoughts: Solving elaborate problems with large language models. *Proceedings of the AAAI Conference on Artificial Intelligence*, 38(16):17682–17690.

Oana-Maria Camburu, Brendan Shillingford, Pasquale Minervini, Thomas Lukasiewicz, and Phil Blunsom. 2020. Make up your mind! adversarial generation of inconsistent natural language explanations. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 4157–4165, Online. Association for Computational Linguistics.

Wenhu Chen, Xueguang Ma, Xinyi Wang, and William W. Cohen. 2023. Program of thoughts prompting: Disentangling computation from reasoning for numerical reasoning tasks. *Transactions on Machine Learning Research*.

Zhoujun Cheng, Tianbao Xie, Peng Shi, Chengzu Li, Rahul Nadkarni, Yushi Hu, Caiming Xiong, Dragomir Radev, Mari Ostendorf, Luke Zettlemoyer, Noah A. Smith, and Tao Yu. 2023. Binding language models in symbolic languages. In *The Eleventh International Conference on Learning Representations*.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. 2021. Training verifiers to solve math word problems. *Preprint*, arXiv:2110.14168.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurelien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Roziere, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Gregoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, Khalid El-Arini, Krithika Iyer, Kshitiz Malik, Kuenley Chiu, Kunal Bhalla, Lauren Rantala-Yeary, Laurens van der Maaten, Lawrence Chen, Liang Tan, Liz Jenkins, Louis Martin, Lovish Madaan, Lubo Malo, Lukas Blecher, Lukas Landzaat, Luke de Oliveira, Madeline Muzzi, Mahesh Pasupuleti, Mannat Singh, Manohar Paluri, Marcin Kardas, Mathew Oldham, Mathieu Rita, Maya Pavlova, Melanie Kambadur, Mike Lewis, Min Si, Mitesh Kumar Singh, Mona Hassan, Naman Goyal, Narjes Torabi, Nikolay Bashlykov, Nikolay Bogoychev, Niladri Chatterji, Olivier Duchenne, Onur Çelebi, Patrick Alrassy, Pengchuan Zhang, Pengwei Li, Petar Vasic, Peter Weng, Prajjwal Bhargava, Pratik Dubal, Praveen Krishnan, Punit Singh Koura, Puxin Xu, Qing He, Qingxiao Dong, Ragavan Srinivasan, Raj Ganapathy, Ramon Calderer, Ricardo Silveira Cabral, Robert Stojnic, Roberta Raileanu, Rohit Girdhar, Rohit Patel, Romain Sauvestre, Ronnie Polidoro, Roshan Sumbaly, Ross Taylor, Ruan Silva, Rui Hou, Rui Wang, Saghar Hosseini, Sahana Chennabasappa, Sanjay Singh, Sean Bell, Seohyun Sonia Kim, Sergey Edunov, Shaoliang Nie, Sharan Narang, Sharath Raparthy, Sheng Shen, Shengye Wan, Shruti Bhosale, Shun Zhang, Simon Vandenhende, Soumya Batra, Spencer Whitman, Sten Sootla, Stephane Collot, Suchin Gururangan, Sydney Borodinsky, Tamar Herman, Tara Fowler, Tarek Sheasha, Thomas Georgiou, Thomas Scialom, Tobias Speckbacher, Todor Mihaylov, Tong Xiao, Ujjwal Karn, Vedanuj Goswami, Vibhor

Gupta, Vignesh Ramanathan, Viktor Kerkez, Vincent Gonguet, Virginie Do, Vish Vogeti, Vladan Petrovic, Weiwei Chu, Wenhan Xiong, Wenyin Fu, Whitney Meers, Xavier Martinet, Xiaodong Wang, Xiaoqing Ellen Tan, Xinfeng Xie, Xuchao Jia, Xuewei Wang, Yaelle Goldschlag, Yashesh Gaur, Yasmine Babaei, Yi Wen, Yiwen Song, Yuchen Zhang, Yue Li, Yuning Mao, Zacharie Delpierre Coudert, Zheng Yan, Zhengxing Chen, Zoe Papakipos, Aaditya Singh, Aaron Grattafiori, Abha Jain, Adam Kelsey, Adam Shajnfeld, Adithya Gangidi, Adolfo Victoria, Ahuva Goldstand, Ajay Menon, Ajay Sharma, Alex Boesenberg, Alex Vaughan, Alexei Baevski, Allie Feinstein, Amanda Kallet, Amit Sangani, Anam Yunus, Andrei Lupu, Andres Alvarado, Andrew Caples, Andrew Gu, Andrew Ho, Andrew Poulton, Andrew Ryan, Ankit Ramchandani, Annie Franco, Aparajita Saraf, Arkabandhu Chowdhury, Ashley Gabriel, Ashwin Bharambe, Assaf Eisenman, Azadeh Yazdan, Beau James, Ben Maurer, Benjamin Leonhardi, Bernie Huang, Beth Loyd, Beto De Paola, Bhargavi Paranjape, Bing Liu, Bo Wu, Boyu Ni, Braden Hancock, Bram Wasti, Brandon Spence, Brani Stojkovic, Brian Gamido, Britt Montalvo, Carl Parker, Carly Burton, Catalina Mejia, Changhan Wang, Changkyu Kim, Chao Zhou, Chester Hu, Ching-Hsiang Chu, Chris Cai, Chris Tindal, Christoph Feichtenhofer, Damon Civin, Dana Beaty, Daniel Kreymer, Daniel Li, Danny Wyatt, David Adkins, David Xu, Davide Testuggine, Delia David, Devi Parikh, Diana Liskovich, Didem Foss, Dingkang Wang, Duc Le, Dustin Holland, Edward Dowling, Eissa Jamil, Elaine Montgomery, Eleonora Presani, Emily Hahn, Emily Wood, Erik Brinkman, Esteban Arcaute, Evan Dunbar, Evan Smothers, Fei Sun, Felix Kreuk, Feng Tian, Firat Ozgenel, Francesco Caggioni, Francisco Guzmán, Frank Kanayet, Frank Seide, Gabriela Medina Florez, Gabriella Schwarz, Gada Badeer, Georgia Swee, Gil Halpern, Govind Thattai, Grant Herman, Grigory Sizov, Guangyi, Zhang, Guna Lakshminarayanan, Hamid Shojanazeri, Han Zou, Hannah Wang, Hanwen Zha, Haroun Habeeb, Harrison Rudolph, Helen Suk, Henry Aspegren, Hunter Goldman, Ibrahim Damlaj, Igor Molybog, Igor Tufanov, Irina-Elena Veliche, Itai Gat, Jake Weissman, James Geboski, James Kohli, Japhet Asher, Jean-Baptiste Gaya, Jeff Marcus, Jeff Tang, Jennifer Chan, Jenny Zhen, Jeremy Reizenstein, Jeremy Teboul, Jessica Zhong, Jian Jin, Jingyi Yang, Joe Cummings, Jon Carvill, Jon Shepard, Jonathan McPhie, Jonathan Torres, Josh Ginsburg, Junjie Wang, Kai Wu, Kam Hou U, Karan Saxena, Karthik Prasad, Kartikay Khandelwal, Katayoun Zand, Kathy Matosich, Kaushik Veeraraghavan, Kelly Michelena, Keqian Li, Kun Huang, Kunal Chawla, Kushal Lakhotia, Kyle Huang, Lailin Chen, Lakshya Garg, Lavender A, Leandro Silva, Lee Bell, Lei Zhang, Liangpeng Guo, Licheng Yu, Liron Moshkovich, Luca Wehrstedt, Madian Khabsa, Manav Avalani, Manish Bhatt, Maria Tsimpoukelli, Martynas Mankus, Matan Hasson, Matthew Lennie, Matthias Reso, Maxim Groshev, Maxim Naumov, Maya Lathi, Meghan Keneally, Michael L. Seltzer, Michal Valko, Michelle Restrepo, Mihir Patel, Mik Vyatskov, Mikayel Samvelyan, Mike Clark, Mike Macey, Mike Wang, Miquel Jubert Hermoso, Mo Metanat, Mohammad Rastegari, Munish Bansal, Nandhini Santhanam, Natascha Parks, Natasha White, Navyata Bawa, Nayan Singhal, Nick Egebo, Nicolas Usunier, Nikolay Pavlovich Laptev, Ning Dong, Ning Zhang, Norman Cheng, Oleg Chernoguz, Olivia Hart, Omkar Salpekar, Ozlem Kalinli, Parkin Kent, Parth Parekh, Paul Saab, Pavan Balaji, Pedro Rittner, Philip Bontrager, Pierre Roux, Piotr Dollar, Polina Zvyagina, Prashant Ratanchandani, Pritish Yuvraj, Qian Liang, Rachad Alao, Rachel Rodriguez, Rafi Ayub, Raghotham Murthy, Raghu Nayani, Rahul Mitra, Raymond Li, Rebekkah Hogan, Robin Battey, Rocky Wang, Rohan Maheswari, Russ Howes, Ruty Rinott, Sai Jayesh Bondu, Samyak Datta, Sara Chugh, Sara Hunt, Sargun Dhillon, Sasha Sidorov, Satadru Pan, Saurabh Verma, Seiji Yamamoto, Sharadh Ramaswamy, Shaun Lindsay, Shaun Lindsay, Sheng Feng, Shenghao Lin, Shengxin Cindy Zha, Shiva Shankar, Shuqiang Zhang, Shuqiang Zhang, Sinong Wang, Sneha Agarwal, Soji Sajuyigbe, Soumith Chintala, Stephanie Max, Stephen Chen, Steve Kehoe, Steve Satterfield, Sudarshan Govindaprasad, Sumit Gupta, Sungmin Cho, Sunny Virk, Suraj Subramanian, Sy Choudhury, Sydney Goldman, Tal Remez, Tamar Glaser, Tamara Best, Thilo Kohler, Thomas Robinson, Tianhe Li, Tianjun Zhang, Tim Matthews, Timothy Chou, Tzook Shaked, Varun Vontimitta, Victoria Ajayi, Victoria Montanez, Vijai Mohan, Vinay Satish Kumar, Vishal Mangla, Vítor Albiero, Vlad Ionescu, Vlad Poenaru, Vlad Tiberiu Mihailescu, Vladimir Ivanov, Wei Li, Wenchen Wang, Wenwen Jiang, Wes Bouaziz, Will Constable, Xiaocheng Tang, Xiaofang Wang, Xiaojian Wu, Xiaolan Wang, Xide Xia, Xilun Wu, Xinbo Gao, Yanjun Chen, Ye Hu, Ye Jia, Ye Qi, Yenda Li, Yilin Zhang, Ying Zhang, Yossi Adi, Youngjin Nam, Yu, Wang, Yuchen Hao, Yundi Qian, Yuzi He, Zach Rait, Zachary DeVito, Zef Rosnbrick, Zhaoduo Wen, Zhenyu Yang, and Zhiwei Zhao. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Yanai Elazar, Nora Kassner, Shauli Ravfogel, Abhilasha Ravichander, Eduard Hovy, Hinrich Schütze, and Yoav Goldberg. 2021. Measuring and improving consistency in pretrained language models. *Transactions of the Association for Computational Linguistics*, 9:1012–1031.

Jia He, Mukund Rungta, David Koleczek, Arshdeep Sekhon, Franklin X Wang, and Sadid Hasan. 2024. Does prompt formatting have any impact on llm performance? *Preprint*, arXiv:2411.10541.

Yiwei Li, Peiwen Yuan, Shaoxiong Feng, Boyuan Pan, Xinglin Wang, Bin Sun, Heda Wang, and Kan Li. 2024. Escape sky-high cost: Early-stopping self-consistency for multi-step reasoning. In *The Twelfth International Conference on Learning Representations*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe.

2024. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*.

Xianzhen Luo, Qingfu Zhu, Zhiming Zhang, Libo Qin, Xuanyu Zhang, Qing Yang, Dongliang Xu, and Wanxiang Che. 2024. Python is not always the best choice: Embracing multilingual program of thoughts. *Preprint*, arXiv:2402.10691.

Tobias Nipkow, Lawrence C. Paulson, and Markus Wenzel. 2002. *Isabelle/HOL — A Proof Assistant for Higher-Order Logic*, volume 2283 of *LNCS*. Springer.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, Irwan Bello, Jake Berdine, Gabriel Bernadett-Shapiro, Christopher Berner, Lenny Bogdonoff, Oleg Boiko, Madelaine Boyd, Anna-Luisa Brakman, Greg Brockman, Tim Brooks, Miles Brundage, Kevin Button, Trevor Cai, Rosie Campbell, Andrew Cann, Brittany Carey, Chelsea Carlson, Rory Carmichael, Brooke Chan, Che Chang, Fotis Chantzis, Derek Chen, Sully Chen, Ruby Chen, Jason Chen, Mark Chen, Ben Chess, Chester Cho, Casey Chu, Hyung Won Chung, Dave Cummings, Jeremiah Currier, Yunxing Dai, Cory Decareaux, Thomas Degry, Noah Deutsch, Damien Deville, Arka Dhar, David Dohan, Steve Dowling, Sheila Dunning, Adrien Ecoffet, Atty Eleti, Tyna Eloundou, David Farhi, Liam Fedus, Niko Felix, Simón Posada Fishman, Juston Forte, Isabella Fulford, Leo Gao, Elie Georges, Christian Gibson, Vik Goel, Tarun Gogineni, Gabriel Goh, Rapha Gontijo-Lopes, Jonathan Gordon, Morgan Grafstein, Scott Gray, Ryan Greene, Joshua Gross, Shixiang Shane Gu, Yufei Guo, Chris Hallacy, Jesse Han, Jeff Harris, Yuchen He, Mike Heaton, Johannes Heidecke, Chris Hesse, Alan Hickey, Wade Hickey, Peter Hoeschele, Brandon Houghton, Kenny Hsu, Shengli Hu, Xin Hu, Joost Huizinga, Shantanu Jain, Shawn Jain, Joanne Jang, Angela Jiang, Roger Jiang, Haozhun Jin, Denny Jin, Shino Jomoto, Billie Jonn, Heewoo Jun, Tomer Kaftan, Łukasz Kaiser, Ali Kamali, Ingmar Kanitscheider, Nitish Shirish Keskar, Tabarak Khan, Logan Kilpatrick, Jong Wook Kim, Christina Kim, Yongjik Kim, Jan Hendrik Kirchner, Jamie Kiros, Matt Knight, Daniel Kokotajlo, Łukasz Kondraciuk, Andrew Kondrich, Aris Konstantinidis, Kyle Kosic, Gretchen Krueger, Vishal Kuo, Michael Lampe, Ikai Lan, Teddy Lee, Jan Leike, Jade Leung, Daniel Levy, Chak Ming Li, Rachel Lim, Molly Lin, Stephanie Lin, Mateusz Litwin, Theresa Lopez, Ryan Lowe, Patricia Lue, Anna Makanju, Kim Malfacini, Sam Manning, Todor Markov, Yaniv Markovski, Bianca Martin, Katie Mayer, Andrew Mayne, Bob McGrew, Scott Mayer McKinney, Christine McLeavey, Paul McMillan, Jake McNeil, David Medina, Aalok Mehta, Jacob Menick, Luke Metz, Andrey Mishchenko, Pamela Mishkin, Vinnie Monaco, Evan Morikawa, Daniel Mossing, Tong Mu, Mira Murati, Oleg Murk, David Mély, Ashvin Nair, Reiichiro Nakano, Rajeev Nayak, Arvind Neelakantan, Richard Ngo, Hyeonwoo Noh, Long Ouyang, Cullen O'Keefe, Jakub Pachocki, Alex Paino, Joe Palermo, Ashley Pantuliano, Giambattista Parascandolo, Joel Parish, Emy Parparita, Alex Passos, Mikhail Pavlov, Andrew Peng, Adam Perelman, Filipe de Avila Belbute Peres, Michael Petrov, Henrique Ponde de Oliveira Pinto, Michael, Pokorny, Michelle Pokrass, Vitchyr H. Pong, Tolly Powell, Alethea Power, Boris Power, Elizabeth Proehl, Raul Puri, Alec Radford, Jack Rae, Aditya Ramesh, Cameron Raymond, Francis Real, Kendra Rimbach, Carl Ross, Bob Rotsted, Henri Roussez, Nick Ryder, Mario Saltarelli, Ted Sanders, Shibani Santurkar, Girish Sastry, Heather Schmidt, David Schnurr, John Schulman, Daniel Selsam, Kyla Sheppard, Toki Sherbakov, Jessica Shieh, Sarah Shoker, Pranav Shyam, Szymon Sidor, Eric Sigler, Maddie Simens, Jordan Sitkin, Katarina Slama, Ian Sohl, Benjamin Sokolowsky, Yang Song, Natalie Staudacher, Felipe Petroski Such, Natalie Summers, Ilya Sutskever, Jie Tang, Nikolas Tezak, Madeleine B. Thompson, Phil Tillet, Amin Tootoonchian, Elizabeth Tseng, Preston Tuggle, Nick Turley, Jerry Tworek, Juan Felipe Cerón Uribe, Andrea Vallone, Arun Vijayvergiya, Chelsea Voss, Carroll Wainwright, Justin Jay Wang, Alvin Wang, Ben Wang, Jonathan Ward, Jason Wei, CJ Weinmann, Akila Welihinda, Peter Welinder, Jiayi Weng, Lilian Weng, Matt Wiethoff, Dave Willner, Clemens Winter, Samuel Wolrich, Hannah Wong, Lauren Workman, Sherwin Wu, Jeff Wu, Michael Wu, Kai Xiao, Tao Xu, Sarah Yoo, Kevin Yu, Qiming Yuan, Wojciech Zaremba, Rowan Zellers, Chong Zhang, Marvin Zhang, Shengjia Zhao, Tianhao Zheng, Juntang Zhuang, William Zhuk, and Barret Zoph. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

Panupong Pasupat and Percy Liang. 2015. Compositional semantic parsing on semi-structured tables. In *Proceedings of the 53rd Annual Meeting of the Association for Computational Linguistics and the 7th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 1470–1480, Beijing, China. Association for Computational Linguistics.

Libo Qin, Qiguang Chen, Fuxuan Wei, Shijue Huang, and Wanxiang Che. 2023. Cross-lingual prompting: Improving zero-shot chain-of-thought reasoning across languages. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2695–2709, Singapore. Association for Computational Linguistics.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. 2024. GPQA: A graduate-level google-proof q&a benchmark. In *First Conference on Language Modeling*.

Omer Sagi and Lior Rokach. 2018. Ensemble learning: A survey. *WIREs Data Mining and Knowledge Discovery*, 8(4):e1249.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. 2019. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations*.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. 2024. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *Preprint*, arXiv:2408.03314.

Peiyi Wang, Lei Li, Liang Chen, Feifan Song, Binghuai Lin, Yunbo Cao, Tianyu Liu, and Zhifang Sui. 2024. Making large language models better reasoners with alignment.

Xuezhi Wang, Haohan Wang, and Diyi Yang. 2022. Measure and improve robustness in NLP models: A survey. In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4569–4586, Seattle, United States. Association for Computational Linguistics.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023. Self-consistency improves chain of thought reasoning in language models. In *The Eleventh International Conference on Learning Representations*.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le, and Denny Zhou. 2022. Chain of thought prompting elicits reasoning in large language models. In *Advances in Neural Information Processing Systems*.

Danny Wood, Tingting Mu, Andrew M. Webb, Henry W. J. Reeve, Mikel Luján, and Gavin Brown. 2024. A unified theory of diversity in ensemble learning. *J. Mach. Learn. Res.*, 24(1).

Vikas Yadav, Steven Bethard, and Mihai Surdeanu. 2019. Quick and (not so) dirty: Unsupervised selection of justification sentences for multi-hop question answering. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2578–2589, Hong Kong, China. Association for Computational Linguistics.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Thomas L. Griffiths, Yuan Cao, and Karthik R Narasimhan. 2023. Tree of thoughts: Deliberate problem solving with large language models. In *Thirty-seventh Conference on Neural Information Processing Systems*.

Xuanliang Zhang, Dingzirui Wang, Longxu Dou, Baoxin Wang, Dayong Wu, Qingfu Zhu, and Wanxiang Che. 2024a. Flextaf: Enhancing table reasoning with flexible tabular formats. *Preprint*, arXiv:2408.08841.

Yongheng Zhang, Qiguang Chen, Min Li, Wanxiang Che, and Libo Qin. 2024b. AutoCAP: Towards automatic cross-lingual alignment planning for zero-shot chain-of-thought. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 9191–9200, Bangkok, Thailand. Association for Computational Linguistics.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. 2023. Judging LLM-as-a-judge with MT-bench and chatbot arena. In *Thirty-seventh Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Tianyang Zhong, Zhengliang Liu, Yi Pan, Yutong Zhang, Yifan Zhou, Shizhe Liang, Zihao Wu, Yanjun Lyu, Peng Shu, Xiaowei Yu, Chao Cao, Hanqi Jiang, Hanxu Chen, Yiwei Li, Junhao Chen, Huawen Hu, Yihen Liu, Huaqin Zhao, Shaochen Xu, Haixing Dai, Lin Zhao, Ruidong Zhang, Wei Zhao, Zhenyuan Yang, Jingyuan Chen, Peilong Wang, Wei Ruan, Hui Wang, Huan Zhao, Jing Zhang, Yiming Ren, Shihuan Qin, Tong Chen, Jiaxi Li, Arif Hassan Zidan, Afrar Jahin, Minheng Chen, Sichen Xia, Jason Holmes, Yan Zhuang, Jiaqi Wang, Bochen Xu, Weiran Xia, Jichao Yu, Kaibo Tang, Yaxuan Yang, Bolun Sun, Tao Yang, Guoyu Lu, Xianqiao Wang, Lilong Chai, He Li, Jin Lu, Lichao Sun, Xin Zhang, Bao Ge, Xintao Hu, Lian Zhang, Hua Zhou, Lu Zhang, Shu Zhang, Ninghao Liu, Bei Jiang, Linglong Kong, Zhen Xiang, Yudan Ren, Jun Liu, Xi Jiang, Yu Bao, Wei Zhang, Xiang Li, Gang Li, Wei Liu, Dinggang Shen, Andrea Sikora, Xiaoming Zhai, Dajiang Zhu, and Tianming Liu. 2024. Evaluation of openai o1: Opportunities and challenges of agi. *Preprint*, arXiv:2409.18486.

Jin Peng Zhou, Charles E Staats, Wenda Li, Christian Szegedy, Kilian Q Weinberger, and Yuhuai Wu. 2024. Don't trust: Verify – grounding LLM quantitative reasoning with autoformalization. In *The Twelfth International Conference on Learning Representations*.

A Prove of Equations

Lemma 1. *Let $m, \phi, \bar{\phi}$ follow the definition in §2.1. If $\lim_{m\to\infty} \bar{\phi} = \phi \circ \mathtt{f}$, we can derive that $\lim_{m\to\infty} \delta_m = 0$.*

Proof. Considering that $\bar{\phi} = \mathtt{avg}(\phi_i) = \mathtt{avg}(\phi \circ \mathtt{f} + \delta_i)$, we can derive that

$$\mathtt{avg}(\phi \circ \mathtt{f} + \delta_i) = \phi \circ \mathtt{f}(m \to \infty)$$

Therefore, $\mathtt{avg}(\delta_i) = 0(m \to \infty)$. Assume, for contradiction, that $\lim_{m\to\infty} \delta_m \neq 0$. Then, there exists some $\epsilon > 0$ such that for large enough m, $\delta_m \geq \epsilon$. For large m, the average of the first m terms is

$$\frac{\delta_1 + \delta_2 + \cdots + \delta_m}{m}$$

Since the average tends to 0, for sufficiently large m, we must have

$$\frac{\delta_1 + \delta_2 + \cdots + \delta_m}{m} < \epsilon$$

However, if infinitely many $\delta_m \geq \epsilon$, this contradicts the fact that the average tends to 0. Thus, $\lim_{m\to\infty} \delta_m = 0$. □

Considering Lemma 1, in the following prove, we substitute $m \to \infty$ with $\delta_m \to 0$.

A.1 Prove of Equation 4

Theorem 1. *Let $D, L, m, \phi, \bar{\phi}$ follow the definition in §2.1. We can derive that:*

$$\lim_{\delta_i \to 0} \mathbb{E}_D \left[L(\bar{\phi}, y) \right] = \frac{1}{m} \sum_{i=1}^{m} L\left(\phi \circ \mathtt{f}, y\right)$$

Proof.

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_D \left[L\left(\phi_i, y\right) \right] \tag{7}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_D \left[L\left(\phi \circ \mathtt{f} + \delta_i, y\right) \right] \tag{8}$$

$$= \mathbb{E}_D \left[L\left(\phi \circ \mathtt{f}, y\right) \right] (\delta_i \to 0) \tag{9}$$

Considering that:

$$\bar{\phi} = \frac{1}{m} \sum_{i=1}^{m} \phi_i \tag{10}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \phi \circ \mathtt{f}(\delta_i \to 0) \tag{11}$$

$$= \phi \circ \mathtt{f} \tag{12}$$

We can derive that:

$$\mathbb{E}_D \left[\frac{1}{m} \sum_{i=1}^{m} L\left(\phi_i, \bar{\phi}\right) \right] \tag{13}$$

$$= \mathbb{E}_D \left[\frac{1}{m} \sum_{i=1}^{m} L\left(\phi \circ \mathtt{f} + \delta_i, \bar{\phi}\right) \right] \tag{14}$$

$$= \mathbb{E}_D \left[\frac{1}{m} \sum_{i=1}^{m} L\left(\phi \circ \mathtt{f}, \phi \circ \mathtt{f}\right) \right] (\delta_i \to 0) \tag{15}$$

$$= 0 \tag{16}$$

Based on Equation 2, we can derive that:

$$\mathbb{E}_D \left[L(\bar{\phi}, y) \right] \tag{17}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_D \left[L\left(\phi_i, y\right) \right] \tag{18}$$

$$- \mathbb{E}_D \left[\frac{1}{m} \sum_{i=1}^{m} L\left(\phi_i, \bar{\phi}\right) \right] \tag{19}$$

$$= \mathbb{E}_D \left[L\left(\phi \circ \mathtt{f}, y\right) \right] (\delta_i \to 0) \tag{20}$$

□

A.2 Prove of Equation 5

Theorem 2. *Let $D, L, m, \phi, \bar{\phi}$ follow the definition in §2.2. we can derive that:*

$$\lim_{\delta_i \to 0} \mathbb{E}_D \left[L(\bar{\phi}, y) \right] \tag{21}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_D \left[L(\phi \circ \mathtt{f}_i, y) \right] \tag{22}$$

$$- \mathbb{E}_D [\frac{1}{m} \sum_{i=1}^{m} L(\phi \circ \mathtt{f}_i, \bar{\phi})] \tag{23}$$

Proof.

$$\frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_D \left[L\left(\phi_i, y\right) \right] \tag{24}$$

$$= \frac{1}{m} \sum_{i=1}^{m} \mathbb{E}_D \left[L\left(\phi \circ \mathtt{f}_i + \delta_i, y\right) \right] \tag{25}$$

$$= \frac{1}{m} \sum_{i=1}^{m} L\left(\phi \circ \mathtt{f}_i, y\right) (\delta_i \to 0) \tag{26}$$

$$\lim_{\delta_i \to 0} \mathbb{E}_D \left[\frac{1}{m} \sum_{i=1}^{m} L\left(\phi_i, \bar{\phi}\right) \right] \tag{27}$$

$$= \mathbb{E}_D \left[\frac{1}{m} \sum_{i=1}^{m} L\left(\phi \circ f_i, \bar{\phi}\right) \right] \tag{28}$$

Based on Equation 2, we can derive that:

$$\begin{aligned}
&\mathbb{E}_D\left[L(\bar{\phi}, y)\right] && (29)\\
&= \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_D\left[L\left(\phi_i, y\right)\right] && (30)\\
&\quad - \mathbb{E}_D\left[\frac{1}{m}\sum_{i=1}^{m} L\left(\phi_i, \bar{\phi}\right)\right] && (31)\\
&= \frac{1}{m}\sum_{i=1}^{m}\mathbb{E}_D\left[L(\phi \circ \mathtt{f}_i, y)\right] && (32)\\
&\quad - \mathbb{E}_D[\frac{1}{m}\sum_{i=1}^{m} L(\phi \circ \mathtt{f}_i, \bar{\phi})](\delta_i \rightarrow 0) && (33)
\end{aligned}$$

□

B Prompts of FORMAT-ADAPTER

The prompts of FORMAT-ADAPTER are shown in Table 4. The prompt for the instruction rewriting is provided in the code since this prompt is too long. The prompts of the answer generation of each task follow Dubey et al. (2024), which can be found in https://huggingface.co/datasets/meta-llama/Llama-3.1-8B-Instruct-evals.

C Baselines of Main Experiments

C.1 Single Format

Single is to generate one answer using one format with Chain-of-Thought (Wei et al., 2022). The prompts we used follow Dubey et al. (2024).

Self-Consistency (SC) is similar to Single, while we generate multiple answers for each question. The generation number is the same as the format number of FORMAT-ADAPTER for each task and we set temperature as 0.5, top_p as 0.9. The prompts are the same with Single.

Tree-of-Thought (ToT) is to generate the reasoning process step by step, where it votes the results of each step, which is used as the input for the next step. The parameters and prompts we used are following the default of the paper.

DTV asks models to generate Isabelle formulations (Nipkow et al., 2002) to answer the questions, which can be executed automatically to ensure the logical correctness of the consistent answers. The parameters and prompts we used are following the default of the paper.

C.2 Multiple Format

CLSP asks LLMs to answer the given questions in different natural languages since different questions could suit different languages. The natural languages, parameters, and prompts we used follow the default of the paper.

MultiPoT aims to improve Program-of-Thought (Chen et al., 2023), which asks LLMs to solve problems with different program languages. The program languages, parameters, and prompts we used follow the default of the paper.

FlexTaF is designed to solve the table reasoning task, which demands LLMs to reason with different tabular formats. The table formats, parameters, and prompts we used follow the default of the paper.

D Reasoning Formats of FORMAT-ADAPTER

In this section, we list the reasoning formats generated by different LLMs on various datasets, as shown in Table 5. We rename some reasoning categories in the experiments of §4.4 to ensure that the similar categories can be compared together. Different formats in Table 5 could be more suitable for different types of questions. For instance, for numerical representation, "12" is appropriate for numerical computations, whereas "twelve" is more suitable for non-numerical queries, such as "how many 'e' are in 12?"

From Table 5, we can observe that: *(i)* Compared to small-scale LLMs, large-scale LLMs are capable of generating a wider variety of reasoning formats, leading to a more significant performance improvement as demonstrated in Table 2; *(ii)* Compared with simple datasets (e.g., GSM8K), a greater number of reasoning formats are generated on more complex datasets (e.g., MATH, GPQA), as more solving approaches are available for complex questions, thus resulting in more diverse reasoning formats.

Although the number of synthesized formats varies across different tasks and models, the comparability of the results is reliable. That is because, some variation in the number of formats does exist between different tasks; however, these quantitative differences can be considered as variations in the intermediate process, reflecting the model's tendency to synthesize different formats depending on the task. Furthermore, since the experimental setup is consistent across different tasks (e.g., instructions,

The prompt of Format Generation

~~~
You are requested to generate possible answer formats that can be changed for the given task, where I want to generate different answers in different formats of the given task.
For each task, you MUST generate the possible answer formats quoted with ** of the task, the number of answer formats of each task MUST > 3.
Here are several examples:

—

Task: Code Generation.
In this task, you are given a question, and then you should generate the Python code to answer the question.
Input: Today is the last day of the first quarter of 2008. What is the date one year ago from today?
Output:
```python
from datetime import datetime, timedelta
today = datetime(2008, 3, 31)
one_year_ago = today - timedelta(days=365)
```
The possible answer formats that can be changed are:
1. Natural Language: The natural languages of questions can be changed, like change as **Chinese, French, German, Spanish**.
2. Code Language: The code languages of answers can be changed, like change to **Java, C++, R, JavaScript**.

—

Based on the above examples, generate the possible answer formats to be changed for the following task.

Task: {task_name}
{task_definition}
Output: {answer}
~~~

The prompt of Answer Scoring

~~~
Please act as an impartial judge and evaluate the quality of the response provided by an AI assistant to the user question displayed below. Your evaluation should consider correctness and helpfulness. You will be given a assistant's answer. Identify and correct any mistakes. Be as objective as possible. After providing your explanation, you must rate the response on a scale of 1 to 10 by strictly following this format: "[rating]", for example: "Rating: [5]".

[Question]
{question}

[The Start of Assistant's Answer]
{answer}
[The End of Assistant's Answer]
~~~

Table 4: The prompts of FORMAT-ADAPTER.

hyperparameters), we consider the comparability of our results to be reliable.

E Efficiency of FORMAT-ADAPTER

In this section, we discuss the efficiency of FORMAT-ADAPTER. For all experiments conducted on GPT-4o, the total cost is approximately \$600, averaging around \$0.30 per question. We focus on two main aspects: the efficiency of the format generation, and the efficiency during inference.

E.1 Efficiency during Format Generation

Let the number of generated formats be M, and $t_{\mathcal{M}}$ represents the average time that LLM $\mathcal{M}$ takes to process a single data. Considering that format generation requires both generation and rewriting, the efficiency of format generation is $2Mt_{\mathcal{M}} = O(Mt_{\mathcal{M}})$.

Based on the discussion, we can adjust M to control the efficiency of format generation. Furthermore, in practical applications, since format generation is performed offline, the cost of this step can be ignored during online inference.

E.2 Efficiency during Reasoning

Let the number of formats selected for each query during inference be denoted as m, the total number of user queries be N, and t_s represents the time to select a single format. Since inference involves

Model	GSM8K-Hard	MATH500	ARC-C-Hard	GPQA
Llama3.1-8b	**natural language (6)**, code language (2), mathematical notation (2), text format (2), answer style (2), response format (1)	**natural language (6)**, step-by-step format (3), text format (2), explanation level (1), mathematical notation (2)	**natural language (8)**, answer format (2), code language (6), explanation level (4), answer style (2), output format (3)	natural language (5), answer format (5), **explanation level (6)**, code language (5), answer style (4), explanation format (5), step-by-step format (4), explanation style (5), mathematical notation (4)
Llama3.1-70b	mathematical notation (4), natural language (4), problem format (4), answer format (4), **reasoning style (3)**, unit of measurement (3)	mathematical notation (3), problem format (5), **solution approach (3)**, answer format (3), unit system (3), problem complexity (3)	natural language (4), answer format (1), question type (1), answer choice format (4), **context format (3)**, answer justification (1)	natural language (4), answer format (9), explanation format (1), **question type (4)**, candidate answer format (7), explanation style (4), answer choice format (7), mathematical notation (6)
GPT-4o	natural language (4), mathematical expression (4), **explanation style (4)**, number representation (5)	**natural language (6)**, explanation format (2), notation style (2), answer presentation (2), units in solution (2), solution format (3), mathematical representation (3), concluding sentence format (3)	**natural language (5)**, numerical representation (3), answer structure (2), answer explanation (4), response format (3), question format (4), contextual explanation (2), answer representation (8)	natural language (4), numerical representation (3), answer presentation (2), **explanation detail (2)**, answer format (3)

Table 5: The reasoning categories generated by FORMAT-ADAPTER on different models and datasets. The number after each category is the format number corresponding to the category. The category with the best performance under each setting is marked in **bold**.

format selection, answer generation, and answer scoring, the total inference efficiency is given by $NMt_s + 2mNt_{\mathcal{M}}$. Given that $t_s \ll t_{\mathcal{M}}$ in practice, the overall inference efficiency simplifies to $O(mNt_{\mathcal{M}})$.

It can be observed that the inference efficiency of FORMAT-ADAPTER is comparable to that of Self-Consistency, while FORMAT-ADAPTER offers a significant performance improvement. Considering that prior research indicates that there is a positive correlation between model performance and inference time (Snell et al., 2024; Zhong et al., 2024), it is important to balance efficiency and performance based on the specific application scenario. For example, when computational resources are limited, the number of reasoning formats used can be reduced to enhance inference efficiency.

E.3 Average Output Tokens of Different Method

To compare the efficiency of FORMAT-ADAPTER with other baselines in practical applications, we measure the average number of tokens output per question, as shown in Table 6. Although FORMAT-ADAPTER is less efficient than Self-Consistency, our method is closer to that of Tree-of-Thought. Considering the performance improvements of FORMAT-ADAPTER over both Self-Consistency and Tree-of-Thought, a balance between efficiency and performance must be considered in practical applications.

F Additional Experiments

F.1 Performance Using All Generated Formats

To validate the necessity of the reasoning format selection of FORMAT-ADAPTER, we compare its performance with that of using all formats without selection. The experimental results, as shown in Table 7, indicate that FORMAT-ADAPTER consistently outperforms that directly using all reasoning formats across all settings, which demonstrates the importance of selecting appropriate reasoning formats.

F.2 Answer Variability of FORMAT-ADAPTER

To demonstrate that FORMAT-ADAPTER can generate different answers using various reasoning formats, we calculate the number of distinct answers generated by our method and Self-Consistency. The results are shown in Table 8, where we observe that, on average, the number of answers generated by our method is significantly higher than that of Self-Consistency. This proves that employing different reasoning formats can guide the model to generate diverse answers.

F.3 Robustness of FORMAT-ADAPTER

To validate the robustness of FORMAT-ADAPTER, we run it five times with different random seeds, as shown in Table 9. From the table, it can be observed that the performance of our method does not fluctuate significantly. As discussed in § 2, gen-

Method	SC	ToT	DTV	FORMAT-ADAPTER
Tokens	3889.9	24611.4	17816.4	25297.0

Table 6: The average output tokens per question on MATH using Llama3.1-8b.

Model	**Method**	**GSM8K-Hard**	**MATH500**	**ARC-C-Hard**	**GPQA**
Llama3.1-8b	All	53.9	54.0	42.2	33.9
	FORMAT-ADAPTER	**54.7**	**56.8**	**57.4**	**33.9**
Llama3.1-70b	All	73.8	70.2	69.9	47.5
	FORMAT-ADAPTER	**76.2**	**70.4**	**71.5**	**51.0**

Table 7: The performance with all formats or the formats selected by FORMAT-ADAPTER. All denotes using all generated formats. The best performance under each setting is marked in **bold**.

erating multiple answers enhances the robustness and reduces performance variability.

F.4 Evaluation of the Score Quality

In this section, we evaluate the quality of the scoring of our method. We use the evaluation metrics $\frac{\sum_{d \in D} \texttt{Valid}(d)}{|D|} \times 100$. $D = \{d\}$ represent all the data, and $\texttt{Valid}(d)$ indicates that if d is correct, it is represented by the corresponding score; otherwise, it is represented by $1-$ the corresponding score. The statistical results are shown in Table 10. It can be observed that the evaluation result is not high, which is consistent with the conclusions of previous studies, highlighting the bottleneck of our method. To address this issue, a possible approach is to select the most appropriate reasoning format for each task using the training data first, and then use the selected reasoning format during inference without relying on LLMs for scoring.

F.5 Repeated Sampling with Different Formats

In this section, we evaluate the performance of FORMAT-ADAPTER after repeated sampling, comparing the use of the single format and FORMAT-ADAPTER. The experimental results are shown in Figure 8. As can be observed from the figure, when the sampling scale is > 1, the performance using a single format is lower than that of FORMAT-ADAPTER, which demonstrates the effectiveness of our method.

G Case Study

To better understand how FORMAT-ADAPTER improves reasoning performance, we present a case study, as shown in Figure 9. From the figure, it can be observed that when using the original reasoning

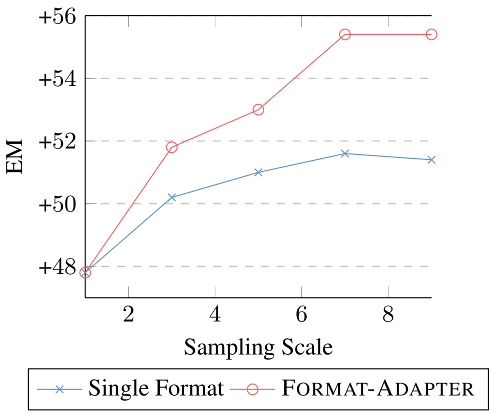
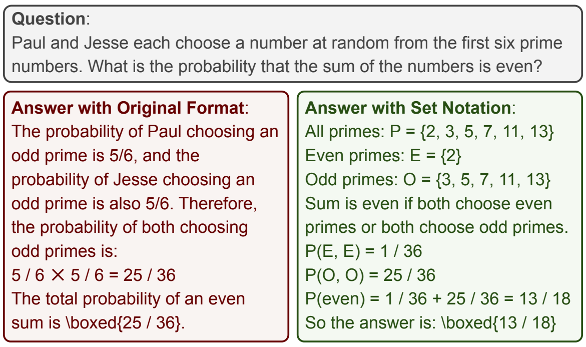


Figure 8: The performance of repeat sampling using single format and FORMAT-ADAPTER on MATH with Llama3.1-8b.



Question:
Paul and Jesse each choose a number at random from the first six prime numbers. What is the probability that the sum of the numbers is even?

Answer with Original Format:
The probability of Paul choosing an odd prime is 5/6, and the probability of Jesse choosing an odd prime is also 5/6. Therefore, the probability of both choosing odd primes is:
5 / 6 × 5 / 6 = 25 / 36
The total probability of an even sum is \boxed{25 / 36}.

Answer with Set Notation:
All primes: P = {2, 3, 5, 7, 11, 13}
Even primes: E = {2}
Odd primes: O = {3, 5, 7, 11, 13}
Sum is even if both choose even primes or both choose odd primes.
P(E, E) = 1 / 36
P(O, O) = 25 / 36
P(even) = 1 / 36 + 25 / 36 = 13 / 18
So the answer is: \boxed{13 / 18}

Figure 9: An example sampled from MATH answered using different reasoning formats. The correct part is marked in red, and the incorrect part is marked in green.

Model	Method	GSM8K-Hard	MATH500	ARC-C-Hard	GPQA
Llama3.1-8b	Self-Consistency	2.0	2.3	1.0	2.3
	FORMAT-ADAPTER	**12.4**	**5.9**	**3.6**	**3.5**
Llama3.1-70b	Self-Consistency	1.3	1.9	1.0	1.8
	FORMAT-ADAPTER	**9.5**	**7.8**	**2.5**	**3.2**

Table 8: The average number of distinct answers generated for each question.

Dataset	8b		70b	
	Vote	Oracle	Vote	Oracle
GSM8K-Hard	54.3 ± 0.4	89.6 ± 0.2	76.1 ± 0.3	94.4 ± 0.6
MATH500	56.7 ± 0.2	74.9 ± 0.2	70.2 ± 0.4	85.0 ± 0.5
ARC-C-Hard	57.2 ± 0.2	91.2 ± 0.4	71.5 ± 0.2	88.2 ± 0.6
GPQA	33.2 ± 0.8	93.3 ± 0.4	51.1 ± 0.7	96.2 ± 0.5
WikiTQ	55.0 ± 0.4	79.3 ± 0.5	63.0 ± 0.1	83.5 ± 0.6

Table 9: The average performance of FORMAT-ADAPTER with five running using Llama3.1.

Dataset	8b	70b
GSM8K-Hard	47.7	66.2
MATH500	46.4	56.0
ARC-C-Hard	52.3	60.2
GPQA	45.7	48.1

Table 10: The average score quality of FORMAT-ADAPTER with Llama3.1 on different datasets.

format, the model overlooks that 2 is also an odd number, leading to an incorrect answer. However, when reasoning with the set notation, the model successfully accounts for all odd numbers, resulting in the correct answer. Therefore, utilizing different reasoning formats helps the model approach questions from multiple perspectives and different questions require different reasoning formats. As such, it is essential to integrate various reasoning formats to obtain the correct solution.