Mirror-Consistency: Harnessing Inconsistency in Majority Voting

Siyuan Huang[1,*], Zhiyuan Ma[2], Jintao Du[2]
Changhua Meng[2], Weiqiang Wang[2], Zhouhan Lin [1,†]
[1] Shanghai Jiaotong University
[2] Tiansuan Lab, Ant Group Co., Ltd.
siyuan_huang_sjtu@outlook.com, lin.zhouhan@gmail.com
{mazhiyuan.mzy,lingke.djt,changhua.mch,weiqiang.wwq}@antgroup.com

Abstract

Self-Consistency, a widely-used decoding strategy, significantly boosts the reasoning capabilities of Large Language Models (LLMs). However, it depends on the plurality voting rule, which focuses on the most frequent answer while overlooking all other minority responses. These inconsistent minority views often illuminate areas of uncertainty within the model's generation process. To address this limitation, we present Mirror-Consistency, an enhancement of the standard Self-Consistency approach. Our method incorporates a 'reflective mirror' into the self-ensemble decoding process and enables LLMs to critically examine inconsistencies among multiple generations. Additionally, just as humans use the mirror to better understand themselves, we propose using Mirror-Consistency to enhance the sample-based confidence calibration methods, which helps to mitigate issues of overconfidence. Our experimental results demonstrate that Mirror-Consistency yields superior performance in both reasoning accuracy and confidence calibration compared to Self-Consistency.

1 Introduction

Large Language Models (LLMs) have substantially influenced diverse sectors with their advanced applications (Chowdhery et al., 2022; Schick et al., 2023; Wu et al., 2023; Shen et al., 2023; Zhang et al., 2023). To further bolster LLMs' reasoning ability, Self-Consistency (Wang et al., 2023a) initially generates a wide range of reasoning pathways, then performs a marginalization to determine the most consistent response. Although generally effective, this approach relies primarily on the plurality voting rule, which focuses only on the most frequent answer, thereby neglecting other minority responses. Consequently, the crucial inconsistencies in the resampled answers, which could reveal
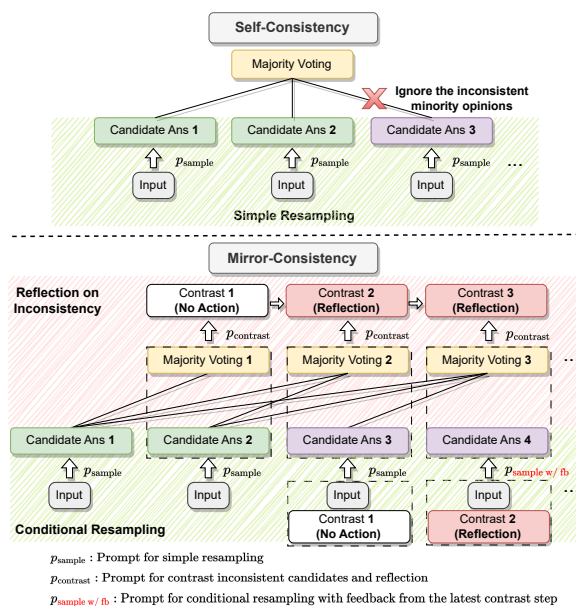

Figure 1: Mirror-Consistency vs. Self-Consistency. While Self-Consistency overlooks minority responses, Mirror-Consistency takes them into account by reflecting on inconsistencies during the sampling process.

uncertainties and potential errors of LLMs, often do not receive the attention they merit.

To address this limitation, we introduce Mirror-Consistency, which enables the LLM to analyze and learn from discrepancies in its resampled responses. The complete process is illustrated in Figure 1. We sequentially resample new responses. Each new response is then compared to the previous majority voting result. If the new response deviates from the majority's decision (i.e., it represents a minority opinion), we position the LLM to act as an arbitrator. In this role, the LLM reflects on the differences between the new response and the previous majority opinion and offers a feedback, which is then utilized to guide the generation of subsequent responses. This process effectively equips the standard Self-Consistency method with a 'mirror', enabling the LLM to learn from incon-

---

*Work done during an internship at Ant Group.
†Corresponding Author.

sistencies observed during the sampling process.

We then delve into another application of Mirror-Consistency: the confidence calibration of LLMs (Desai and Durrett, 2020; Geng et al., 2023). Recent studies leverage the consistency of multiple generations to assess the LLM's confidence in its responses (Xiong et al., 2023; Wang et al., 2023a; Manakul et al., 2023; Xiong et al., 2023; Portillo Wightman et al., 2023). Following this idea, we find that Mirror-Consistency, which involves repeatedly comparing and reflecting on the discrepancies between the majority and the minority opinions, provides a more robust quantification of uncertainty, particularly in cases of overconfidence. Our motivation for using mirror consistency in confidence calibration stems from the idea that answers that consistently emerge as the majority after several rounds of comparisons with other candidate answers are likely more reliable.

We conduct experiments on four reasoning datasets with four different LLMs to compare the performance of Mirror-Consistency with Self-Consistency. The results confirm the efficacy of reflecting on the inconsistencies during the sampling process, showing improvements in both reasoning accuracy and confidence calibration.

2 Mirror-Consistency

This section details the implementation of Mirror-Consistency. Self-Consistency depends on the plurality voting rule, which tends to overlook minority responses that may be inconsistent yet informative. To address this gap, we introduce a reflective form of consistency, termed Mirror-Consistency.

Mirror-Consistency and Self-Consistency share procedural similarities, as both methods generate multiple responses and use majority voting to determine the final output. The key distinction is that Mirror-Consistency examines the inconsistencies within the sampling process, rather than merely repeat sampling independent answers. The Mirror-Consistency process begins with an initial sampling phase to produce the first response, denoted as r_0. This is followed by a series of iterations that alternate between Reflection on Inconsistency and Conditional Resampling, as shown in Algorithm 1. We next introduce these two steps in detail. The corresponding prompts are provided in Appendix D.

Reflection on Inconsistency   During the kth Reflection on Inconsistency, the model compares the newly resampled response r_k with the majority

voting result m_{k-1} from previous rounds. If the responses align, the process moves to the next resampling phase. Otherwise, the model analyzes the causes of the discrepancies and formulates a suggestion for additional checks, as the inconsistencies of multiple reasoning pathways often reveal uncertainty and potential mistakes within the generation. This suggestion is incorporated into the existing checklist C_{k-1}, updating it to C_k for use in the subsequent Conditional Resampling.

Conditional Resampling   In this phase, the feedback obtained from the previous round is utilized to guide the generation of subsequent responses. During the kth round of Conditional Resampling, the checklist C_k, derived from earlier rounds, is integrated into the prompt, facilitating the generation of a new response r_{k+1}. This method ensures that the LLM concentrates on areas of uncertainty exposed by inconsistencies noted during prior samplings.

Algorithm 1 Mirror-Consistency

Require: prompts $\{p_{\text{sample}}, p_{\text{sample w/ fb}}, p_{\text{contrast}}\}$, model $\mathcal{M}$, input x
1: $r_0 \leftarrow \mathcal{M}(p_{\text{sample}} \| x)$ ▷ Initial response
2: $C_0 \leftarrow \emptyset$ ▷ Initial checklist
3: $m_0 \leftarrow r_0$ ▷ Initial majority vote
4: **for** $k \in \{1, \dots, K\}$ **do**
5:   $r_k \leftarrow \mathcal{M}(p_{\text{sample w/ fb}} \| C_{k-1} \| x)$
         ▷ Conditional Resampling
6:   $C_k \leftarrow \mathcal{M}(p_{\text{contrast}} \| m_{k-1} \| r_k \| C_{k-1} \| x)$
        ▷ Reflection on Inconsistency
7:   $m_k \leftarrow \text{majority-vote}(r_0, \dots, r_k)$
          ▷ Update majority vote
8: **end for**
9: **return** m_K

Comparison with Self-Reflection   Standard intrinsic self-correction methods directly generate feedback for the initial responses, which frequently exhibit overconfidence and difficulty in identifying errors (Huang et al., 2023; Tyen et al., 2023). In contrast, Mirror-Consistency identifies potential errors by critically analyzing the inconsistencies across multiple reasoning pathways. This contrastive strategy is generally more practical than directly generating feedback (Zhang et al., 2024).

3 Calibration with Mirror-Consistency

In this section, we introduce the application of Mirror-Consistency to the calibration of LLMs. Re-

Model	Method	# Ans	# Call	GSM8K	SVAMP	StrategyQA	Date
GPT-3.5-turbo	Standard COT	1	1	75.9	77.3	73.6	67.3
	Self-Consistency	10	10	85.2 ↑9.3	84.2 ↑6.9	77.9 ↑4.3	72.4 ↑5.1
	Self-Consistency	20	20	85.4 ↑9.5	84.5 ↑7.2	77.8 ↑4.2	**73.0** ↑5.7
	Mirror-Consistency	10	19	**86.7** ↑10.8	**86.1** ↑8.8	**78.4** ↑4.8	72.6 ↑5.3
Qwen-turbo	Standard COT	1	1	74.5	76.9	71.9	62.7
	Self-Consistency	10	10	77.4 ↑2.9	79.1 ↑2.2	73.8 ↑1.9	63.5 ↑0.8
	Self-Consistency	20	20	77.9 ↑3.4	79.1 ↑2.2	73.6 ↑1.7	63.5 ↑0.8
	Mirror-Consistency	10	19	**79.7** ↑5.2	**83.0** ↑6.1	**74.7** ↑2.8	**64.3** ↑1.6
Llama3-8B	Standard COT	1	1	78.7	80.8	71.6	76.9
	Self-Consistency	10	10	85.9 ↑7.2	89.0 ↑8.2	72.0 ↑0.4	83.8 ↑6.9
	Self-Consistency	20	20	86.5 ↑7.8	89.6 ↑8.8	<u>72.8</u> ↑1.2	**84.1** ↑7.2
	Mirror-Consistency	10	19	**87.2** ↑8.5	**90.4** ↑9.6	<u>72.8</u> ↑1.2	83.5 ↑6.6
Llama3-70B	Standard COT	1	1	92.7	91.8	72.9	90.6
	Self-Consistency	10	10	95.3 ↑2.6	93.5 ↑1.7	73.7 ↑0.8	94.2 ↑3.6
	Self-Consistency	20	20	95.6 ↑2.9	**94.0** ↑2.2	73.7 ↑0.8	<u>94.2</u> ↑3.6
	Mirror-Consistency	10	19	**95.8** ↑3.1	93.2 ↑1.4	**74.7** ↑1.8	<u>94.2</u> ↑3.6

Table 1: Performance comparison between Mirror-Consistency and Self-Consistency on Arithmetic Reasoning and Commensense Reasoning tasks. The best scores are bolded. Multiple identical scores are indicated with underlining.

cent studies have shown that consistency across multiple model generations serves as a reliable indicator of confidence (Portillo Wightman et al., 2023; Manakul et al., 2023). While effective, these methods rely on the Self-Consistency logic (i.e., 'independent' repeated sampling) and overlook the analysis of inconsistencies among responses. In contrast, Mirror-Consistency employs a *reflective form of consistency*: responses maintaining consistency after numerous comparisons and reflections are deemed more reliable; while responses that vary upon comparison and reflection highlight the LLM's inherent uncertainties. Hence, we propose using the distribution of responses generated by Mirror-Consistency as a measure of confidence.

Next, we introduce two fundamental types of sample-based confidence metrics. A comprehensive analysis of five existing sample-based confidence metrics is provided in Appendix A. The most commonly employed strategy is the agreement-based metric (Lyu et al., 2024). For each input x, we generate n candidate outputs, $\hat{r}_1, \ldots, \hat{r}_n$. We then apply majority voting across the answers to determine the most-voted answer $\bar{r} = \text{argmax}_r \sum_{i=1}^n \mathbb{1}(\hat{r}_i = r)$. The agreement-based confidence score is then defined as the percentage of answers that agree with the most-voted answer:

$$\text{Agree}(\bar{r}) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(\hat{r}_i = \bar{r})$$

Agree solely relies on the most voted answer. **First-Second-Distance (FSD)** (Lyu et al., 2024) provides another option by considering the top two most-voted answers, denoted as $\bar{r}_1$ and $\bar{r}_2$. $\text{FSD}(\bar{r})$ is the difference in their agreement rates:

$$\text{FSD}(\bar{r}) = \text{Agree}(\bar{r}_1) - \text{Agree}(\bar{r}_2)$$

Further experiments, as detailed in subsection 4.2 and Appendix A, demonstrate that this reflective form of consistency provides more reliable calibration than standard sample-based methods.

4 Experiments

In this section, we assess the efficacy of Mirror-Consistency from two distinct angles: First, we evaluate its reasoning accuracy in comparison to Self-Consistency across four reasoning datasets. Second, we demonstrate how the reflective form of consistency embodied by Mirror-Consistency improves the sample-based calibration methods.

4.1 Reasoning Accuracy

Benchmarks We evaluate our models using four distinct datasets: GSM8K (Cobbe et al., 2021) and SVAMP (Patel et al., 2021), which focus on arithmetic reasoning; StrategyQA (Geva et al., 2021) and Date Understanding (BIG-Bench collaboration, 2021), which belong to Multi-hop QA. We follow the same split setting as Lyu et al. (2023).

LLMs We utilize four different large language models: GPT-3.5-turbo-0613, Qwen-turbo, and the newly released Llama-3 family (8B/70B). More details can be found in subsection C.2.
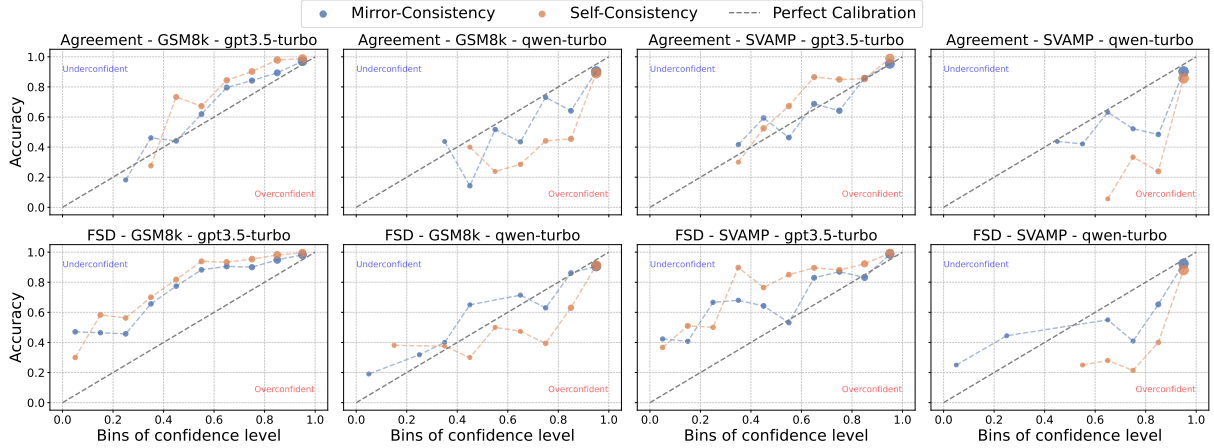
Figure 2: Comparison of calibration curves on GSM8K and SVAMP with `Agree` and `FSD`. We omit the bins if their size is less than 1% of the total dataset. The point size is proportional to the square root of its corresponding bin size.

Baselines We compare our method with Standard CoT Prompt (Kojima et al., 2022) and Self-Consistency (Wang et al., 2023b). To ensure fairness in our comparison, we evaluate two scenarios: equivalent number of responses (10 vs. 10), and comparable number of calls (19 vs. 20).

Prompting and Sampling Strategies Throughout our experiments, we consistently use the same COT prompt, which is detailed in Appendix D with a temperature $T = 0.4$, following Lyu et al. (2024).

Main Result The main results are presented in Table 1. Mirror-Consistency outperforms Self-Consistency, even with the latter doubling its number of responses. We observe that Mirror-Consistency generally yields greater improvements in arithmetic reasoning datasets. This can be attributed to the fact that, whereas errors in common sense are more likely constrained by the inherent limitations of LLMs, arithmetic errors are more amenable to correction through reflective methods.

4.2 Confidence Calibration

Experiment Setup We compare the performance of Mirror-Consistency and Self-Consistency, each using a comparable number of calls (19 vs. 20). We consider two LMs, GPT-3.5-turbo and Qwen-turbo, on two tasks, GSM8K and SVAMP, with the same settings in subsection 4.1. We use Expected Calibration Error (ECE) (Guo et al., 2017) as the evaluation metric. We first consider two confidence metrics, `Agree` and `FSD`, as introduced in section 3.

Main Result Table 2 presents the ECE comparisons across two different confidence metrics.

Model & Method	Agreement		FSD	
	G	**S**	**G**	**S**
GPT-3.5-turbo				
Self-Consistency	0.081	0.044	0.170	0.127
Mirror-Consistency	**0.039**	**0.042**	**0.138**	**0.082**
Qwen-turbo				
Self-Consistency	0.154	0.196	0.127	0.173
Mirror-Consistency	**0.102**	**0.103**	**0.089**	**0.100**

Table 2: ECE ↓ results of Mirror-Consistency and Self-Consistency. **G** denotes GSM8K and **S** denotes SVAMP.

Mirror-Consistency consistently yields superior results relative to Self-Consistency. This finding suggests that reflecting on inconsistencies during the sampling process leads to more accurate confidence estimations, offering a novel approach to calibration tasks. Figure 2 presents the corresponding calibration curves. We find that Mirror-Consistency significantly mitigates the overconfidence issues encountered with Self-Consistency. For instance, when using responses generated by Self-Consistency, the curve of Qwen-turbo tends to the bottom right. In contrast, the curve of Mirror-Consistency is clearly closer to the ideal line.

Additional Results We consider five different sample-based calibration metrics in Appendix A. Table 3 presents a complete comparison. The corresponding calibration curves are shown in Figure 3.

5 Conclusion

Mirror-Consistency effectively remedies a crucial limitation in the standard Self-Consistency method: the disregard for minority viewpoints during sampling. Our approach examines the discrepancies

across various responses to identify uncertainties within the generative process. The experimental results demonstrates that Mirror-Consistency not only achieves higher reasoning accuracy compared to Self-Consistency but also enhances model calibration, especially when standard sample-based calibration methods face issues of overconfidence.

6 Limitations

We acknowledge several limitations that invite further investigation: Firstly, while Self-Consistency can generate multiple answers in parallel, Mirror-Consistency, due to the fact that each generation of a new answer needs to be compared with the previous majority opinion, must generate answers sequentially. Secondly, our experimental setup solely utilizes Chain-of-Thought (Wei et al., 2022) as the prompt strategy for sampling. Exploring a broader array of prompt strategies could provide deeper insights into their interactions with the Mirror-Consistency approach. Thirdly, the central motivation behind Mirror-Consistency is to effectively harness the inconsistencies that emerge during repeated sampling processes. This requires the responses to exhibit enough diversity. To enhance this diversity, we could enable the large language model to automatically generate prompts that vary across different dimensions (Zhang et al., 2024). Fourthly, we fix the temperature parameter throughout our study. Future work could experiment with varying the temperature to assess the robustness of Mirror-Consistency under different settings. Lastly, since Mirror-Consistency relies on prompting to encourage reflective thinking, it necessitates strong instruction-following capabilities within the model. While the four models considered in our experiments demonstrate adequate ability in this regard, this strategy may not be as effective with smaller-scale or lower-performing models.

7 Ethical Considerations

We recognize several ethical Considerations. Our approach, Mirror-Consistency, improves LLM robustness by addressing inconsistencies but cannot fully negate the risks of errors or biases from the underlying data and model structures. Additionally, the method's iterative process, involving repeated resampling and reflection, presents concerns about computational costs and environmental impacts. Finally, it's crucial to clearly convey the limitations of Mirror-Consistency, informing users about possible failure scenarios and the potential repercussions. This transparency is vital for setting appropriate expectations for real-world applications.

References

BIG-Bench collaboration. 2021. Beyond the Imitation Game: Measuring and extrapolating the capabilities of language models.

Pinzhen Chen, Zhicheng Guo, Barry Haddow, and Kenneth Heafield. 2023a. Iterative translation refinement with large language models. *arXiv preprint arXiv:2306.03856*.

Yangyi Chen, Lifan Yuan, Ganqu Cui, Zhiyuan Liu, and Heng Ji. 2023b. A Close Look into the Calibration of Pre-trained Language Models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1343–1367, Toronto, Canada. Association for Computational Linguistics.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, and others. 2022. Palm: Scaling language modeling with pathways. *ArXiv*, abs/2204.02311.

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.

Shrey Desai and Greg Durrett. 2020. Calibration of pre-trained transformers. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 295–302, Online. Association for Computational Linguistics.

Yarin Gal and Zoubin Ghahramani. 2016. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In *Proceedings of the 33nd International Conference on Machine Learning, ICML 2016, New York City, NY, USA, June 19-24, 2016*, volume 48 of *JMLR Workshop and Conference Proceedings*, pages 1050–1059. JMLR.org.

Jiahui Geng, Fengyu Cai, Yuxia Wang, Heinz Koeppl, Preslav Nakov, and Iryna Gurevych. 2023. A Survey of Language Model Confidence Estimation and Calibration.

Mor Geva, Daniel Khashabi, Elad Segal, Tushar Khot, Dan Roth, and Jonathan Berant. 2021. *Did Aristotle Use a Laptop?* A Question Answering Benchmark with Implicit Reasoning Strategies. *Transactions of the Association for Computational Linguistics*, 9:346–361.

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q. Weinberger. 2017. On calibration of modern neural networks. In *Proceedings of the 34th International Conference on Machine Learning, ICML 2017, Sydney, NSW, Australia, 6-11 August 2017*, volume 70 of *Proceedings of Machine Learning Research*, pages 1321–1330. PMLR.

Jiaxin Huang, Shixiang Shane Gu, Le Hou, Yuexin Wu, Xuezhi Wang, Hongkun Yu, and Jiawei Han. 2022. Large language models can self-improve. *ArXiv*, abs/2210.11610.

Jie Huang, Xinyun Chen, Swaroop Mishra, Huaixiu Steven Zheng, Adams Wei Yu, Xinying Song, and Denny Zhou. 2023. Large language models cannot self-correct reasoning yet. *CoRR*, abs/2310.01798.

Siddhartha Jain, Xiaofei Ma, Anoop Deoras, and Bing Xiang. 2023. Self-consistency for open-ended generations. *ArXiv*, abs/2307.06857.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. 2022. Language models (mostly) know what they know. *Preprint*, arXiv:2207.05221.

Geunwoo Kim, Pierre Baldi, and Stephen McAleer. 2023. Language models can solve computer tasks. *arXiv preprint arXiv:2303.17491*.

Takeshi Kojima, Shixiang (Shane) Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. 2022. Large Language Models are Zero-Shot Reasoners. In *Conference on Neural Information Processing Systems (NeurIPS)*.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Advances in Neural Information Processing Systems 30: Annual Conference on Neural Information Processing Systems 2017, December 4-9, 2017, Long Beach, CA, USA*, pages 6402–6413.

Kimin Lee, Kibok Lee, Honglak Lee, and Jinwoo Shin. 2018. A simple unified framework for detecting out-of-distribution samples and adversarial attacks. In *Advances in Neural Information Processing Systems 31: Annual Conference on Neural Information Processing Systems 2018, NeurIPS 2018, December 3-8, 2018, Montréal, Canada*, pages 7167–7177.

Tian Liang, Zhiwei He, Wenxiang Jiao, Xing Wang, Yan Wang, Rui Wang, Yujiu Yang, Zhaopeng Tu, and Shuming Shi. 2023. Encouraging divergent thinking in large language models through multi-agent debate. *ArXiv*, abs/2305.19118.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Teaching models to express their uncertainty in words. *Transactions on Machine Learning Research*.

Qing Lyu, Shreya Havaldar, Adam Stein, Li Zhang, Delip Rao, Eric Wong, Marianna Apidianaki, and Chris Callison-Burch. 2023. Faithful chain-of-thought reasoning. *ArXiv preprint*, abs/2301.13379.

Qing Lyu, Kumar Shridhar, Chaitanya Malaviya, Li Zhang, Yanai Elazar, Niket Tandon, Marianna Apidianaki, Mrinmaya Sachan, and Chris Callison-Burch. 2024. Calibrating large language models with sample consistency. *CoRR*, abs/2402.13904.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, Sean Welleck, Bodhisattwa Prasad Majumder, Shashank Gupta, Amir Yazdanbakhsh, and Peter Clark. 2023. Self-refine: Iterative refinement with self-feedback. *ArXiv*, abs/2303.17651.

Potsawee Manakul, Adian Liusie, and Mark Gales. 2023. SelfCheckGPT: Zero-resource black-box hallucination detection for generative large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 9004–9017, Singapore. Association for Computational Linguistics.

Ning Miao, Yee Whye Teh, and Tom Rainforth. 2024. Selfcheck: Using llms to zero-shot check their own step-by-step reasoning. In *ICLR 2024*.

Sabrina J. Mielke, Arthur Szlam, Emily Dinan, and Y-Lan Boureau. 2022. Reducing conversational agents' overconfidence through linguistic calibration. *Transactions of the Association for Computational Linguistics*, 10:857–872.

Deepak Nathani, David Wang, Liangming Pan, and William Yang Wang. 2023. Maf: Multi-aspect feedback for improving reasoning in large language models. *ArXiv*, abs/2310.12426.

Liangming Pan, Michael Stephen Saxon, Wenda Xu, Deepak Nathani, Xinyi Wang, and William Yang Wang. 2023. Automatically correcting large language models: Surveying the landscape of diverse self-correction strategies. *ArXiv*, abs/2308.03188.

Arkil Patel, Satwik Bhattamishra, and Navin Goyal. 2021. Are NLP models really able to solve simple math word problems? In *Proceedings of the 2021 Conference of the North American Chapter of the*

Association for Computational Linguistics: Human Language Technologies, pages 2080–2094, Online. Association for Computational Linguistics.

Debjit Paul, Mete Ismayilzada, Maxime Peyrard, Beatriz Borges, Antoine Bosselut, Robert West, and Boi Faltings. 2023. Refiner: Reasoning feedback on intermediate representations. *ArXiv*, abs/2304.01904.

Gwenyth Portillo Wightman, Alexandra Delucia, and Mark Dredze. 2023. Strength in numbers: Estimating confidence of large language models by prompt agreement. In *Proceedings of the 3rd Workshop on Trustworthy Natural Language Processing (TrustNLP 2023)*, pages 326–362, Toronto, Canada. Association for Computational Linguistics.

Stefan Rüping. 2006. Robust probabilistic calibration. In *Machine Learning: ECML 2006: 17th European Conference on Machine Learning Berlin, Germany, September 18-22, 2006 Proceedings 17*, pages 743–750. Springer.

Timo Schick, Jane Dwivedi-Yu, Roberto Dessì, Roberta Raileanu, M. Lomeli, Luke Zettlemoyer, Nicola Cancedda, and Thomas Scialom. 2023. Toolformer: Language Models Can Teach Themselves to Use Tools. *ArXiv*, abs/2302.04761.

Yongliang Shen, Kaitao Song, Xu Tan, Dongsheng Li, Weiming Lu, and Yueting Zhuang. 2023. Hugginggpt: Solving ai tasks with chatgpt and its friends in hugging face. In *Advances in Neural Information Processing Systems*.

Noah Shinn, Beck Labash, and Ashwin Gopinath. 2023. Reflexion: an autonomous agent with dynamic memory and self-reflection. *ArXiv*, abs/2303.11366.

Kumar Shridhar, Harsh Jhamtani, Hao Fang, Benjamin Van Durme, Jason Eisner, and Patrick Xia. 2023. Screws: A modular framework for reasoning with revisions. *ArXiv preprint*, abs/2309.13075.

Kaya Stechly, Matthew Marquez, and Subbarao Kambhampati. 2023. Gpt-4 doesn't know it's wrong: An analysis of iterative prompting for reasoning problems.

Gladys Tyen, Hassan Mansoor, Peter Chen, Tony Mak, and Victor Carbune. 2023. Llms cannot find reasoning errors, but can correct them! *CoRR*, abs/2311.08516.

Ante Wang, Linfeng Song, Ye Tian, Baolin Peng, Lifeng Jin, Haitao Mi, Jinsong Su, and Dong Yu. 2024. Self-consistency boosts calibration for math reasoning. *CoRR*, abs/2403.09849.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023a. Self-consistency improves chain of thought reasoning in language models. In *ICLR 2023*.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. 2023b. Self-Consistency Improves Chain of Thought Reasoning in Language Models. *ICLR 2023 poster*, abs/2203.11171.

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*.

Chenfei Wu, Shengming Yin, Weizhen Qi, Xiaodong Wang, Zecheng Tang, and Nan Duan. 2023. Visual chatgpt: Talking, drawing and editing with visual foundation models. *arXiv preprint arXiv:2303.04671*.

Zhiheng Xi, Senjie Jin, Yuhao Zhou, Rui Zheng, Songyang Gao, Tao Gui, Qi Zhang, and Xuanjing Huang. 2023. Self-polish: Enhance reasoning in large language models via problem refinement. *ArXiv*, abs/2305.14497.

Miao Xiong, Zhiyuan Hu, Xinyang Lu, Yifei Li, Jie Fu, Junxian He, and Bryan Hooi. 2023. Can llms express their uncertainty? an empirical evaluation of confidence elicitation in llms. *CoRR*, abs/2306.13063.

KiYoon Yoo, Jangho Kim, Jiho Jang, and Nojun Kwak. 2022. Detection of Word Adversarial Examples in Text Classification: Benchmark and Baseline via Robust Density Estimation.

Ori Yoran, Tomer Wolfson, Ben Bogin, Uri Katz, Daniel Deutch, and Jonathan Berant. 2023. Answering questions by meta-reasoning over multiple chains of thought. *ArXiv*, abs/2304.13007.

Wenqi Zhang, Yongliang Shen, Weiming Lu, and Yueting Zhuang. 2023. Data-copilot: Bridging billions of data and humans with autonomous workflow. *arXiv preprint arXiv:2306.07209*.

Wenqi Zhang, Yongliang Shen, Linjuan Wu, Qiuying Peng, Jun Wang, Yueting Zhuang, and Weiming Lu. 2024. Self-contrast: Better reflection through inconsistent solving perspectives. *CoRR*, abs/2401.02009.

Chuanyang Zheng, Zhengying Liu, Enze Xie, Zhenguo Li, and Yu Li. 2023. Progressive-hint prompting improves reasoning in large language models. *ArXiv*, abs/2304.09797.

Denny Zhou, Nathanael Schärli, Le Hou, Jason Wei, Nathan Scales, Xuezhi Wang, Dale Schuurmans, Claire Cui, Olivier Bousquet, Quoc V Le, and Ed H. Chi. 2023. Least-to-most prompting enables complex reasoning in large language models. In *The Eleventh International Conference on Learning Representations*.

		Agreement		Entropy		FSD		Ans-Num		Pairwise		# Win
Model & Method	‖	G	S	G	S	G	S	G	S	G	S	
GPT-3.5-turbo												
Self-Consistency		0.081	0.044	0.291	0.223	0.170	0.127	**0.056**	**0.049**	0.090	0.057	2
Mirror-Consistency		**0.039**	**0.042**	**0.279**	**0.196**	**0.138**	**0.082**	0.078	0.068	**0.053**	**0.051**	8
Qwen-turbo												
Self-Consistency		0.154	0.196	**0.104**	0.137	0.127	0.173	0.145	0.163	**0.099**	0.194	2
Mirror-Consistency		**0.102**	**0.103**	0.138	**0.104**	**0.089**	**0.100**	**0.039**	**0.057**	0.150	**0.101**	8

Table 3: Summary of ECE ↓ results across different calibration metrics. **G** denotes GSM8K and **S** denotes SVAMP. We show the statistics of the comparison in the last column.

A Additional Calibration Results

In this section, we explore the impact of various metrics on calibration results based on Mirror-Consistency and Self-Consistency approaches. Recently, numerous studies have adopted sample-based methods to assess the confidence of Large Language Models (LLMs) (Wang et al., 2023a; Manakul et al., 2023; Xiong et al., 2023; Portillo Wightman et al., 2023). These studies have proposed different metrics to derive corresponding confidence scores from the varying resampled responses. Again, for each input x, we generate n candidate outputs, $\hat{r}_1, \ldots, \hat{r}_n$. We then apply majority voting across the answers to determine the most-voted answer $\bar{r} = \arg\max_r \sum_{i=1}^n \mathbb{1}(\hat{r}_i = r)$. We consider the following five metrics:

• *Agreement-based* (Lyu et al., 2024):
The agreement-based confidence score, which is the most common metric and has been introduced in section 3, is defined as the percentage of answers that agree with the most-voted answer:

$$\mathrm{Agree}(\bar{r}) = \frac{1}{n} \sum_{i=1}^n \mathbb{1}(\hat{r}_i = \bar{r})$$

• *Entropy-based* (Lyu et al., 2024):
We first derive a set of unique answers from the model's output, denoted as $\hat{\mathbf{r}}$, by eliminating duplicates. The entropy-based consistency, $\mathrm{Ent}(\bar{r})$, is then defined as follows:

$$\mathrm{Ent}(\bar{r}) = 1 - \left(-\frac{1}{\log |\hat{\mathbf{r}}|} \sum_{i=1}^{|\hat{\mathbf{r}}|} p_i \log p_i \right)$$

Here, $|\hat{\mathbf{r}}|$ represents the number of unique answers, and p_i is the normalized frequency of each unique answer $\hat{r}_i$ in the dataset. This formulation inversely relates the entropy measure to the consistency of the answer distribution, where a lower entropy implies a more uniform and certain response pattern.

• *FSD-based* (Lyu et al., 2024):
FSD stands for **F**irst-**S**econd-**D**istance. To calculate the FSD-based consistency, we first identify the top two most-voted answers, denoted as $\bar{r}_1$ and $\bar{r}_2$. We then calculate the agreement rates for these two answers, denoted as $\mathrm{Agree}(\bar{r}_1)$ and $\mathrm{Agree}(\bar{r}_2)$, respectively. The FSD-based consistency, $\mathrm{FSD}(a)$, is computed as the difference in their agreement rates:

$$\mathrm{FSD}(\bar{r}) = \mathrm{Agree}(\bar{r}_1) - \mathrm{Agree}(\bar{r}_2)$$

• *Answer-Number-based* (Wang et al., 2024):
Again we derive a set of unique answers from the model's output, denoted as $\hat{\mathbf{r}}$. The answer-number-based , $\mathrm{Ans\text{-}Num}(\bar{r})$, is then simply defined as:

$$\mathrm{Ans\text{-}Num}(\bar{r}) = 1 - \frac{|\hat{\mathbf{r}}|}{n}$$

• *Pairwise-Comparison-based* (Wang et al., 2024):
We count the number of times each different answer is repeated, denoted as $n_1 \ldots n_{|\hat{\mathbf{r}}|}$. Specially, we denote the index of the majority voting result as $i_{\bar{r}}$. Then the pairwise-comparison-based, $\mathrm{Pairwise}(\bar{r})$, metric is defined as:

$$\mathrm{Pairwise}(\bar{r}) = \prod_{j \neq i_{\bar{r}}}^{|\hat{\mathbf{r}}|} \frac{n_{i_{\bar{r}}}}{n_{i_{\bar{r}}} + n_j}$$

We evaluate the performance of Mirror-Consistency and Self-Consistency on the task of confidence calibration using the five distinct metrics described above. Our experiments span two datasets, GSM8K and SVAMP, and consider two models: GPT-3.5-turbo and Qwen-turbo.

Main result As shown in Table 3, we find that Mirror-Consistency outperforms Self-Consistency most of the time. Additionally, we provide the corresponding calibration curves in Figure 3. We set the number of bins as 10. We do not plot bins

with a bin size smaller than 1% of the total data set. The size of the points is proportional to (the square root) of the bin size. We find that in most cases Mirror-Consistency's calibration curve is closer to the ideal curve. It is noteworthy that when methods based on Self-Consistency encounter issues of overconfidence (where the calibration curve tends toward the bottom right), Mirror-Consistency can significantly mitigate the phenomenon of overconfidence. This indicates that multiple comparisons and reflections against minority opinions can alleviate the overconfidence that arises from solely relying on majority opinions.

B   Related Works

B.1   Self-Correction in Languages Models

Recent advancements in large language models have shed light on their advanced cognitive intelligence, notably their ability for self-correction. This attribute enables LLMs to amend their initial responses by integrating both external and self-generated feedback, thus improving previous outputs (Xi et al., 2023; Pan et al., 2023; Nathani et al., 2023). The concept of self-correction encompasses a variety of techniques that have been extensively surveyed and categorized based on the source of feedback and the timing of correction (Pan et al., 2023). These techniques range from using explicit error messages in tasks, such as code execution, to human feedback and model-generated prompts (Miao et al., 2024; Chen et al., 2023a; Huang et al., 2023; Kim et al., 2023). While iterative prompting techniques have shown promise, recent research has raised concerns about LLMs' capacity for independent reflection, revealing limitations in modifying responses without external feedback (Huang et al., 2023; Stechly et al., 2023; Liang et al., 2023; Tyen et al., 2023).

B.2   Confidence Calibration

In the realm of machine learning, uncertainty quantification methods are crucial for assessing the risk associated with model predictions (Rüping, 2006; Desai and Durrett, 2020). Traditional calibration approaches, including probabilistic, ensemble-based, and density-based methods, although effective, require extensive computational resources and access to model internals, making them less viable for closed-source LLMs (Guo et al., 2017; Lakshminarayanan et al., 2017; Gal and Ghahramani, 2016; Lee et al., 2018; Yoo et al., 2022). Further-

more, some post-hoc strategies have emerged, focusing on eliciting model-estimated probabilities of correctness or directly verbalized confidence levels (Kadavath et al., 2022; Lin et al., 2022; Mielke et al., 2022). And recently, inspired by the sample-based methods, calibration through sample consistency, which relies solely on model input and output, has been explored. (Wang et al., 2023a; Manakul et al., 2023; Xiong et al., 2023; Portillo Wightman et al., 2023).

B.3   Refined Problem-Solving Strategies

Existing prompting strategies engage in two forms of reasoning: exploring various perspectives (breadth) and refining ideas to reduce errors (depth). Self-consistency and related methods (Wang et al., 2023b; Huang et al., 2022; Yoran et al., 2023; Jain et al., 2023) promoting breadth through diverse reasoning sampling. In contrast, strategies like self-reflection and abstraction (Shinn et al., 2023; Madaan et al., 2023; Paul et al., 2023; Zheng et al., 2023) focus on the depth of reasoning by iteratively refining prompts.

C   More Implementation Details

C.1   Dataset Details

Math Word Problems (MWP).   This category includes challenges where the objective is to compute numeric solutions to problems framed in natural language. Our analysis incorporates several datasets, each selected to evaluate distinct aspects of mathematical reasoning. For **GSM8K** (Cobbe et al., 2021), participants engage with a diverse set of elementary math questions, testing their ability to apply basic arithmetic operations and logical reasoning. Meanwhile, **SVAMP** (Patel et al., 2021) is specifically designed to probe the robustness of models against changes in question phrasing and structural complexity, offering a stringent test of comprehension and adaptability.

Multi-hop QA.   Tasks in this section demand answers to intricate questions via a series of logical deductions, necessitating a nuanced understanding of the content. Answers may be in the form of Boolean values or specific textual responses. **StrategyQA** (Geva et al., 2021) dataset presents science questions that require an implicit strategy for multi-step reasoning, challenging the model's ability to form and execute complex inferential chains. **Date Understanding** (BIG-Bench collaboration, 2021) tests temporal reasoning by asking participants to
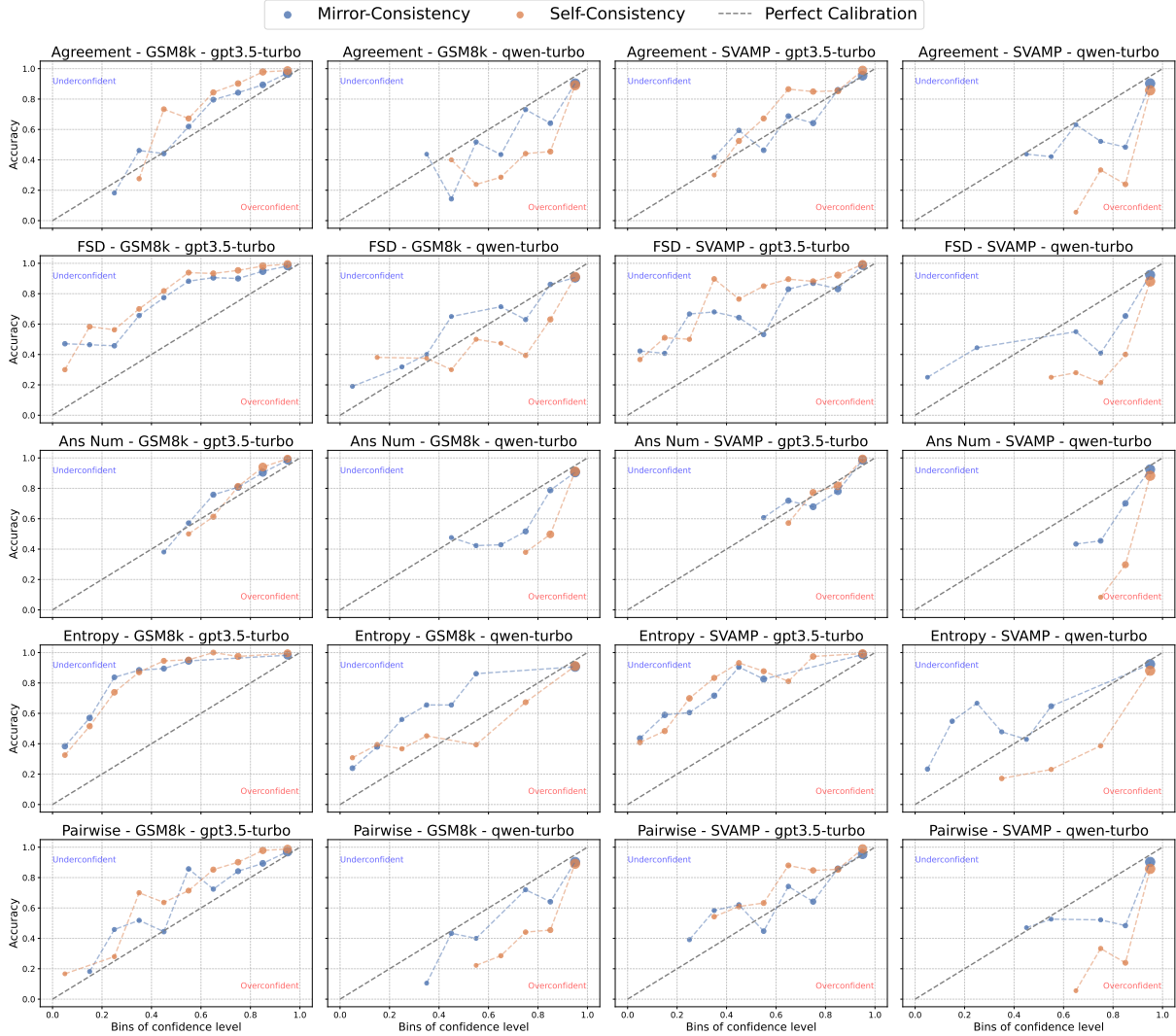
Figure 3: Visualization of confidence calibration results across different calibration metrics. We set the number of bins as 10. We omit the bins if their size is less than 1% of the total dataset. The point size is proportional to the square root of its corresponding bin size.

calculate dates based on relative time information, thus assessing both numerical and chronological reasoning skills.

C.2 Model Details

Closed-Source Language Models We utilize two advanced closed-source language models, GPT-3.5-turbo and Qwen-turbo. For GPT-3.5-turbo, the API is accessible at `platform.openai.com`. Qwen-turbo, another potent large language model, is available via the DashScope provided by Alibaba Cloud (`alibabacloud.com`). Detailed guidance about DashScope and its integration can be accessed here. Further information about utilizing Qwen-turbo through Alibaba Cloud, including API instructions, can be found here.

Open-Source LMs We use the newly released Llama-3 Family (`llama.meta.com`). We conduct experiments using Nvidia A100 80GB GPUs. Specifically, *Meta-Llama-3-70B-Instruct* requires two GPUs per inference, whereas *Meta-Llama-3-8B-Instruct* operates with a single GPU.

C.3 Other Details

Evaluation Metrics We use Expected Calibration Error (ECE) (Guo et al., 2017) as the calibration metric. Predictions are first binned into $M = 10$ intervals based on model confidence. For each bin B_m, we calculate the average accuracy $\mathsf{acc}(B_m)$ and average confidence $\mathsf{conf}(B_m)$. ECE is defined as their weighted absolute differences:

$$\mathsf{ECE} = \sum_{m=1}^{M} \frac{|B_m|}{N} |\mathsf{acc}(B_m) - \mathsf{conf}(B_m)|$$

Split Setting   In order to save computational resources without loss of generality, we select the first 768 examples from the test split of GSM8K and SVAMP for our experiments. We use the full split of Date. For StrategyQA, we test on the first 490 samples following Shridhar et al. (2023).

D   Prompt Template

In this section, we outline the three crucial prompts required for the future realization of Mirror-Consistency: 1) Prompt for Simple Resampling, denoted as p_{sample}, 2) Prompt for Reflection on Inconsistency, denoted as $p_{contrast}$, 3) Prompt for Conditional Resampling, denoted as $p_{sample\ w/\ fb}$.

It is important to note that our implementation adopts the Chain-of-Thought (COT) approach to resample multiple answers. However, it is feasible to explore a variety of other prompt methods, such as Least-to-Most (LtM) (Zhou et al., 2023), Program of Thoughts (PoT) (Chen et al., 2023b) and Faithful CoT (FCoT) (Lyu et al., 2023). Furthermore, in our experiments to validate the generality of Mirror-Consistency, we apply the same prompts across diverse datasets, including arithmetic, symbolic reasoning, and commonsense reasoning. Nonetheless, future work may tailor prompts specifically to the characteristics of different datasets, thus enhancing reflection and contrast, to potentially improve Mirror-Consistency.

D.1   Prompt for Simple Resampling p_{sample}

The following template is employed in a simple resampling process. During its use, **[QUESTION]** must be substituted with the specific question at hand. This template is designed for initial sampling in Mirror-Consistency and for instances when the Checklist is empty, as well as serving as a baseline in our experiments with Self-Consistency.

Simple Resampling Prompt

Solve the following problem step by step. Begin each step with "Step :" and ensure each step is separated by "\n\n". Conclude with the phrase "So the answer is", followed by the answer.

Question: **[QUESTION]**

Answer:

D.2   Prompt for Contrast $p_{contrast}$

The template for reflecting on inconsistencies is as follows. In practice, **[QUESTION]** should be replaced with the actual question, **[PRE-MAJORITY-VOTE]** should be replaced with the majority vote answer from all previous rounds (in cases of multiple identical responses, one is randomly selected as the majority vote answer), **[CUR-RESPONSE]** should be substituted with the most recently generated response of the current round, and finally **[PRE-CHECKLIST]** should be replaced by the checklist from the last round. This template facilitates the reflection of inconsistencies within the Mirror-Consistency approach.

Reflection on Inconsistency Prompt

Given two candidate solutions for a question, carefully analyze and compare the differences in their reasoning steps. Consider: 1) The specific differences in their reasoning steps and final answers; 2) The reasons behind these disparities.

Question: **[QUESTION]**

Two solutions:
Solution 1: **[PRE-MAJORITY-VOTE]**
Solution 2: **[CUR-RESPONSE]**

If no differences exist, output <STOP!>.
If differences are identified, describe them, determine errors, and explain why. Extract a key suggestion to prevent such errors and combine it with the previous checklist **[PRE-CHECKLIST]** to formulate a new checklist. Begin the checklist with <CHECKING>.

Feedback:

D.3   Prompt for Conditional Resampling $p_{sample\ w/\ fb}$

In the conditional resampling process, the following template is utilized. It requires replacing **[QUESTION]** with the specific question and substituting **[CHECKLIST]** with the latest checklist derived from the previous contrast phase. This template is used in Mirror-Consistency for resampling new responses based on reflective feedback addressing inconsistencies.

E   URLs and Licenses

Table 4 provides license information for the datasets we utilize in our experiments. We employ all the above datasets solely for research purposes, in accordance with their designated uses.

Dataset	Reference	URL	License
Math Word Problems			
GSM8K	Cobbe et al. (2021) (Cobbe et al., 2021)	`https://github.com/openai/grade-school-math`	MIT License: `https://github.com/openai/grade-school-math/blob/master/LICENSE`
SVAMP	Patel et al. (2021) (Patel et al., 2021)	`https://github.com/arkilpatel/SVAMP`	MIT License: `https://github.com/arkilpatel/SVAMP/blob/main/LICENSE`
Multi-hop QA			
StrategyQA	Geva et al. (2021) (Geva et al., 2021)	`https://github.com/google/BIG-bench/tree/main/bigbench/benchmark_tasks/strategyqa`	Apache License v.2: `https://github.com/google/BIG-bench/blob/main/LICENSE`
Date Understanding	BIG-Bench Collaboration (2021) (BIG-Bench collaboration, 2021)	`https://github.com/google/BIG-bench`	Apache License v.2: `https://github.com/google/BIG-bench/blob/main/LICENSE`

Table 4: Summary of URLs and Licenses