BREAKING FOCUS: CONTEXTUAL DISTRACTION CURSE IN LARGE LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Abstract: Recent advances in Large Language Models (LLMs) have revolutionized generative systems, achieving excellent performance across diverse domains. Although these models perform well in controlled environments, their real-world applications frequently encounter inputs containing both essential and irrelevant details. Our investigation has revealed a critical vulnerability in LLMs, which we term Contextual Distraction Vulnerability (CDV). This phenomenon arises when models fail to maintain consistent performance on questions modified with semantically coherent but irrelevant context. To systematically investigate this vulnerability, we propose an efficient tree-based search methodology to automatically generate CDV examples. Our approach successfully generates CDV examples across four datasets, causing an average performance degradation of approximately 45% in state-of-the-art LLMs. To address this critical issue, we explore various mitigation strategies and find that post-targeted training approaches can effectively enhance model robustness against contextual distractions. Our findings highlight the fundamental nature of CDV as an ability-level challenge rather than a knowledge-level issue since models demonstrate the necessary knowledge by answering correctly in the absence of distractions. This calls the community's attention to address CDV during model development to ensure reliability.

1 INTRODUCTION

Large Language Models (LLMs) (Zhou et al., 2023b) have demonstrated remarkable capabilities across various Natural Language Processing (NLP) tasks, revolutionizing wide downstream applications such as medicine (Zhao et al., 2023), education (Kasneci et al., 2023), and science (Li et al., 2024b; Guo et al., 2023; Huang et al., 2024e). Despite their impressive performance, recent studies have exposed various vulnerabilities in LLMs, including susceptibility to jailbreaking attacks (Zou et al., 2023), hallucination issues (Xu et al., 2024b), and consistency problems (Liang et al., 2024; Huang et al., 2024a). These vulnerabilities highlight the limitations of LLMs in handling nuanced and adversarial scenarios, making it critical to uncover and analyze additional weaknesses to improve their reliability.

In this work, we investigate a novel vulnerability termed **Contextual Distraction Vulnerability (CDV)**, where semantically coherent but *non-essential contextual* additions to a question degrade LLM performance. For instance, a customer service chatbot might miss a refund request hidden in a short story about discovering products through social media influencers. Similarly, a technical query about machine learning could be misunderstood if it's preceded by a student's emotional account of exam preparation anxiety. Unlike adversarial attacks that inject semantically meaningless noise into inputs (Zou et al., 2023; Shi et al., 2024) and distraction brought by long-context input (Bai et al., 2023), for CDV, our study demonstrates that semantically coherent without a long context yet contextually distracting modifications are sufficient to disrupt the decision-making process of even the most advanced LLMs. This vulnerability underscores a critical weakness in LLMs' ability to filter out irrelevant information and prioritize core knowledge, which is essential for robust reasoning.

Recent studies have demonstrated the powerful generative capabilities of LLM (Xu et al., 2024a; Wu et al., 2024). To systematically investigate this vulnerability, we propose a methodology for efficiently automating the generation of questions that may trigger CDV in LLMs. This is a non-trivial problem, as the primary challenge lies in identifying the most effective distraction (a perturbation to the original input) within a vast search space while ensuring that the added distraction remains semantically coherent with the original question. Our proposed method employs a tree-based search algorithm, incorporates early stopping strategies and leverages a simple but effective classifier to pre-filter problematic candidates. Through extensive experimentation across multiple datasets on the latest LLMs, we demonstrate the effectiveness of our method in producing effective CDV question examples.

Beyond this, we investigate potential causes and mitigation strategies for CDV. Our findings reveal that 1) while simple prompt engineering provides limited benefits, targeted fine-tuning (e.g., DPO (Rafailov et al., 2024)) significantly enhances model robustness against contextual distractions. 2) CDV is a fundamental challenge in LLM development that requires ability-level rather than knowledge-level enhancement.

In summary, our contributions are three-folds: 1) We identify and characterize CDV, a significant weakness in current LLMs that affects their reliability in question-answering tasks. 2) We propose an efficient automated pipeline to generate semantically valid contextual distractions that expose this vulnerability, achieving substantial performance degradation across various models and datasets. 3) We conduct comprehensive experiments to evaluate different mitigation strategies, providing insights into the nature of CDV and potential approaches to overcome CDV.

2 Methodology

2.1 Overview

As illustrated in Figure 1, our objective is to systematically identify and analyze the vulnerabilities of LLMs by generating high-quality CDV examples that challenge models while maintaining the semantic integrity of the original questions.

Generating effective CDV examples poses three significant challenges: First, it is critical to design an effective reward signal that guides modifications toward high-quality CDV examples. Second, the process must address the computational overhead caused by the vast search space associated with generating and evaluating these examples. Third, the generation must mitigate semantic shifts and account for the impact introduced by long-context inputs to ensure the original intent of the examples.

To overcome these challenges, we introduce a multi-step framework. Initially, a *classifier-based filtering* mechanism is employed to identify original samples that are easy to perturb, substantially reducing the computational burden by narrowing down the search space. For the filtered samples, we apply a *tree-based search* strategy that systematically explores contextual distractions using simulation-driven evaluation. This process generates controlled perturbations through a *proxy model* and guides the search using rewards based on the outcomes of the *victim model* during simulation. Iterative semantic validation and length control are applied to ensure that the generated examples align closely with the original intent. Furthermore, we incorporate efficient pruning and early stopping techniques to balance the trade-off between uncovering meaningful CDV examples and maintaining computational efficiency.

2.2 Problem Formulation

Let $D = \{P_1, P_2, \ldots, P_N\}$ denote a dataset consisting of N multiple-choice problem instances, where each instance is represented as a tuple: $P = \langle Q, A_{\text{gt}}, \mathcal{A}_{\text{inc}} \rangle$, with Q denoting the question, A_{gt} the ground truth answer, and $\mathcal{A}_{\text{inc}}$ a set of incorrect answers. Given a victim model M, our goal is to construct a perturbed dataset $D' = \{P'_1, P'_2, \ldots, P'_N\}$, where each perturbed instance $P' = \langle Q', A_{\text{gt}}, \mathcal{A}_{\text{inc}} \rangle$ is obtained by applying a perturbation ΔQ to the question Q, such that $Q' = Q + \Delta Q$.

Our aim is to optimize the perturbation ΔQ to minimize the accuracy of M on D', while ensuring semantic consistency and length constraints between Q and Q'. Here, semantic consistency is determined by a binary classifier S, which outputs $S(Q, Q') \in \{0, 1\}$, where $S(Q, Q') = 1$ indicates no semantic shift. Formally, the problem is expressed as:

$$\min_{\Delta Q} \quad \mathbb{E}_{P \sim D}\Big[\mathcal{L}_{\text{accuracy}}(M, Q')\Big], \quad \text{s.t.} \quad S(Q, Q') = 1, \quad \frac{\text{len}(Q')}{\text{len}(Q)} \leq \lambda \tag{1}$$

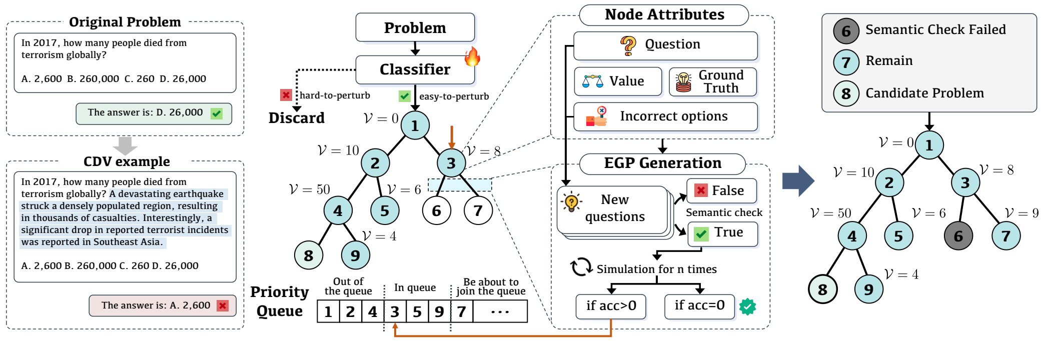

Figure 1: The overview of the proposed method. Given a problem, our goal is to automatically transform it into a CDV example. Initially, a classifier filters the problems to identify potential candidates that are easy to perturb. Next, the method employs a tree-based search, supported by a priority queue to manage the search priorities of individual nodes. Using Error-Guided Perturbation (EGP) generation, the pipeline efficiently and automatically produces effective CDV examples.

Here, S ensures that the perturbation ΔQ does not lead to a semantic shift, while the length ratio constraint λ ensures Q' remains within acceptable bounds compared to the original Q. This constraint is necessary because recent studies show that LLMs experience performance degradation in long context scenarios (Bai et al., 2023). To prevent excessive length expansion in Q', we introduce a length constraint, where λ is an upper bound on the relative length of Q' compared to Q.

If the output Q' does not satisfy the constraints in Equation 1, it is discarded, and a new perturbation ΔQ is generated by re-prompting the proxy model.

2.3 Error-Guided Perturbation Generation

The perturbation ΔQ is generated using a **proxy model**, denoted as P_{proxy}. The proxy model is prompted with the original problem instance $P = \langle Q, A_{\text{gt}}, \mathcal{A}_{\text{inc}} \rangle$ and tasked with generating a modified question Q', where ΔQ represents the perturbation introduced by P_{proxy}. The perturbation process is formalized as $\Delta Q = P_{\text{proxy}}(Q, A_{\text{gt}}, \mathcal{A}_{\text{inc}})$, where P_{proxy} generates ΔQ based on a predefined prompt designed to guide the proxy model in producing modifications that increase the likelihood of the victim model M selecting an incorrect answer $a_{\text{inc}} \in \mathcal{A}_{\text{inc}}$ (i.e., lead model to response with an error).

2.4 Tree-Based Perturbation Exploration

We employ a tree-based simulation-driven method to optimize perturbations by heuristically exploring the search space. A priority queue is maintained to store nodes ordered by their value $\mathcal{V}(P')$, with the highest-value node dequeued and expanded iteratively using P_{proxy} to identify high-potential vulnerabilities.

Simulation For Measuring Perturbation Quality. Firstly, we aim to design the reward of perturbed questions to measure their value. For a given problem instance $P = \langle Q, A_{\text{gt}}, \mathcal{A}_{\text{inc}} \rangle$, the simulation process evaluates the quality of a perturbation by estimating the success rate of the victim model M on P. Let $y \sim M(y \mid P)$ represent the output of the model M when queried on P. During a single simulation, the success rate $r_M(P)$ is computed by sampling n model outputs:

$$r_M(P) = \frac{1}{n} \sum_{i=1}^{n} \mathbb{I}\{y_i = A_{\text{gt}}\}, \quad y_i \sim M(y \mid P), \tag{2}$$

where $\mathbb{I}\{\cdot\}$ is an indicator function that returns 1 if the model's output y_i matches the ground truth answer A_{gt}, and 0 otherwise. The success rate $r_M(P)$ quantifies the likelihood of the model producing the correct answer under the given perturbation.

A perturbed problem $P' = \langle Q', A_{\text{gt}}, \mathcal{A}_{\text{inc}} \rangle$ is considered successfully perturbed if $r_M(P') = 0$, indicating that the model fails to produce the correct answer in all sampled outputs. The simulation process computes a value $\mathcal{V}(P')$ for the node corresponding to P' in the tree-based search:

$$\mathcal{V}(P') = \exp\left(\frac{\alpha}{r_M(P')}\right) \cdot \text{depth}^{-\gamma}, \quad \text{s.t. } r_M(P') \neq 0, \tag{3}$$

where α and γ are scaling constants, $r_M(P')$ is the success rate of the victim model M on P', and depth represents the recursion depth of the node in the search tree. For the question with $r_M(P') = 0$, we add it into the candidate problem list L for storing the problems considered to be successfully "frustrate" LLMs.

The simulation process systematically estimates $\mathcal{V}(P')$, prioritizing perturbed problems with lower success rates $r_M(P')$, which correspond to higher potential vulnerabilities in the model. Simultaneously, the factor depth$^{-\gamma}$ discourages deeper recursions in the search tree, ensuring computational efficiency. High-value nodes with large $\mathcal{V}(P')$ scores are prioritized in the following tree-based search.

Tree-Based Search. For the tree-based search, the process begins with a root $P_{\text{root}} = \langle Q, A_{\text{gt}}, \mathcal{A}_{\text{inc}} \rangle$. A priority queue $\mathcal{Q}$ is maintained, where each node P' is ordered by its value $\mathcal{V}(P')$ in descending order. Initially, the root is added to the queue as $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{P_{\text{root}}\}$. At each iteration, the node P' with the highest value $\mathcal{V}(P')$ is dequeued for exploration: $P' = \arg\max_{P \in \mathcal{Q}} \mathcal{V}(P), \quad \mathcal{Q} \leftarrow \mathcal{Q} \setminus \{P'\}$.

The proxy model P_{proxy} generates $k = |\mathcal{A}_{\text{inc}}|$ child nodes for P', corresponding to perturbations ΔQ_j derived from each incorrect candidate answer $a_{\text{inc}} \in \mathcal{A}_{\text{inc}}$:

$$Q'_j = Q' + \Delta Q_j, \quad P'_j = \langle Q'_j, A_{\text{gt}}, \mathcal{A}_{\text{inc}} \rangle, \quad j = 1, 2, \ldots, k. \tag{4}$$

Each child node P'_j is evaluated by a simulation-driven method to compute its value $\mathcal{V}(P'_j)$, and the child nodes are added to the priority queue: $\mathcal{Q} \leftarrow \mathcal{Q} \cup \{P'_1, P'_2, \ldots, P'_k\}$.

The search iteratively repeats this searching process, dynamically expanding the highest-value node and exploring the perturbation space.

3 Experiment

3.1 Main Results

We conducted extensive evaluation and mitigation experiments on CDV. The detailed configurations of each experiment and their correspondence with all figures can be found in Appendix C.

Our method enables LLMs to autonomously generate CDV examples and effectively engage in self-challenging. In Figure 2, the same model was configured to serve as both the proxy and victim model during the process of CDV investigation, we can observe that all tested models exhibit a significant decline in accuracy on perturbed questions when compared to their performance on the original questions. For instance, GPT-4o-mini's performance significantly drops by more than 40%.

Furthermore, it reveals an intriguing pattern: vulnerabilities identified by a model itself tend to be more challenging than those discovered by other models. For instance, GPT-4o-mini exhibits a significantly lower accuracy of 0.185 on perturbed questions generated through its self-identified vulnerabilities, compared to its performance of 0.235 on perturbations created by the more sophisticated GPT-4o. This indicates that **the model itself is more aware of its weaknesses**, aligning with the justifiability of previous works on self-alignment (Sun et al., 2024) and self-correction (Pan et al., 2023).

Table 1: Model performance on original, and perturbed datasets and the corresponding decline.

Model	CommonsenseQA			OpenbookQA			TruthfulQA			MMLU		
	Original	Perturbed	Δ	Original	Perturbed	Δ	Original	Perturbed	Δ	Original	Perturbed	Δ
GPT-4o-mini	0.857	0.220	0.637	0.897	0.228	0.668	0.607	0.160	0.447	0.787	0.255	0.532
Llama-3.1-8B	0.753	0.230	0.524	0.807	0.212	0.595	0.570	0.283	0.288	0.697	0.300	0.397
Gemma-2-27B	0.857	0.249	0.607	0.867	0.231	0.636	0.782	0.449	0.332	0.753	0.340	0.413
o1-mini	0.856	0.296	0.560	0.897	0.377	0.519	0.748	0.523	0.226	0.803	0.451	0.352
Qwen2.5-72B	0.880	0.304	0.576	0.917	0.325	0.592	0.790	0.442	0.348	0.807	0.412	0.395
GPT-4o	0.890	0.277	0.613	0.950	0.375	0.575	0.757	0.494	0.263	0.870	0.552	0.318
Claude-3.5-sonnet	0.873	0.345	0.529	0.953	0.529	0.424	0.840	0.734	0.106	0.877	0.645	0.232

All models are susceptible to challenges posed by CDV examples, regardless of their capabilities. Our findings reveal that CDV examples generated by stronger models effectively challenge weaker models, while those generated by weaker models also significantly impact stronger ones. As shown in Figure 2, despite their superior language, reasoning, and knowledge capabilities, the stronger models have not achieved complete dominance over smaller models in handling CDV examples. Even the most advanced LLMs[1], such as GPT-4o and Claude-3.5-Sonnet, achieve less than 50% accuracy on questions perturbed by Gemma-2-27B. This underscores a fundamental concern, showing that no model is fully immune to such vulnerability.

The extent of performance degradation varies significantly across tasks. For example, as shown in Table 1, the average performance drop on the TruthfulQA dataset is consistently smaller than that observed on OpenbookQA for all evaluated models. This trend holds true across models, indicating that model sensitivity against CDV varies in different tasks. Tasks that emphasize domain-specific knowledge like OpenbookQA, appear to expose model vulnerabilities more effectively than trustworthiness-related tasks like TruthfulQA.

Our method outperforms existing perturbation techniques by a significant margin. We consider two traditional perturbation approaches: length-based augmentation (Elaborated), which expands the original question's length while preserving its core meaning, and simple auto-generated perturbations (Prompt-only), which uses a single prompt to add interference to the question. However, these methods lead to only minimal performance drops. As shown in Table 5, even advanced methods like DyVal2 achieve only a 15.2% average accuracy reduction. In contrast, our proposed perturbation method introduces a dramatic 52.0% average accuracy decline, outperforming existing methods by 2 to 4 times. Notably, even highly capable models such as Claude-3.5-Sonnet exhibit a substantial 49.5% absolute accuracy drop under our perturbations, compared to just 10.0% under DyVal2. These results highlight the unique strength of our tree-based search framework in systematically targeting the model's ability to discern and resist distraction across diverse architectures, rather than merely increasing input complexity.

3.2 Classifier: Filtering Hard-to-Perturb Problems

A critical challenge in CDV generation lies in distinguishing between **hard-to-perturb problems** (the questions are quite easy and naive for LLMs so that it's highly impossible to be the CDV examples, like elementary arithmetic or factual questions) and **easy-to-perturb problems** (questions susceptible to semantic-preserving perturbations). Our analysis reveals that approximately 37% of computational resources are typically wasted on enhancing hard-to-perturb problems. To address this inefficiency, we develop classifiers that predict the perturbation possibility of each problem.

Table 2: The impact of the classifier on the perturbation success rate of the LLMs. The full model names are: GPT-4o, Gemma-2-27B, Llama-3.1-70B, and Qwen2-5-72B. The rows display the perturbation success rate with and without the classifier.

Mode	GPT-4o	Gemma-2	Llama-3.1	Qwen2.5
w/o classifier	0.527	0.592	0.581	0.563
w/ classifier	0.723	0.791	0.754	0.735

Firstly, to validate the classifier's generalization, it is crucial to examine whether hard-to-perturb questions exhibit nearly consistency across most LLMs. The confusion matrix in Figure 4 demonstrates significant overlap in perturbation difficulty across models: about 82% of problems are either perturbable or non-perturbable for both models in any pairwise comparison. This consistency across models enables the development of universal classifiers for the prediction of perturbation difficulty.

We implement two classifier paradigms: (1) *Prompt-based classifiers* using zero-shot LLM judgments, and (2) *Fine-tuned classifiers* trained on 1,080 annotated samples with 120 held-out test cases. Notably, as shown in Table 2, classifiers trained on GPT-4o-mini-derived data generalize robustly to diverse LLMs, maintaining high precision even on stronger models. The evaluation focuses on precision-recall tradeoffs using the F_β metric ($\beta = 0.5$), prioritizing precision to minimize false positives in perturbable problem selection. The F_β score is computed as:

$$F_\beta = (1 + \beta^2) \times \frac{\text{Precision} \times \text{Recall}}{(\beta^2 \times \text{Precision}) + \text{Recall}} \tag{5}$$

[1]We refer to the leaderboard at `https://lmarena.ai/` for model performance comparisons.

The results shown in Figure 5 and Table 6 reveal that fine-tuned classifiers substantially improve perturbation efficiency through precise problem selection. The fine-tuned classifiers achieve 83% precision in identifying perturbable problems, outperforming the best prompt-based approach (GPT-4o at 68%). We will discuss the details of computational cost saved by the trained classifier in Appendix C.3.

3.3 MITIGATION OF CDV

Notably, models that perform well on original questions but fail on perturbed ones show they have the necessary knowledge but are susceptible to contextual information. To address this, we explored both **prompt-based (*i.e.*, training-free)** and **training-based strategies** to improve their performance on these challenging questions.

Prompt-based mitigation is unable to patch CDV. Since our perturbed questions introduce context-based perturbations while preserving the core question content, we explored whether explicit prompt instructions could help models focus on the essential information and filter out the added distractions. We tested this approach by modifying the original prompts to include specific guidance on identifying and focusing on the key question components, with detailed prompt templates provided in Appendix F. As shown in Table 7, the prompt-based approach did not yield significant improvements in mitigating CDV. Some models, such as o1-mini and Claude-3.5-Sonnet, even showed slightly decreased accuracy with the modified prompts. This suggests that the distractions introduced by our perturbed questions cannot be effectively addressed through simple prompt refinements alone.

Training-based mitigation brings significant benefits. The limited success of prompt-based mitigation inspired us to explore whether training-based approaches (i.e., fine-tuning) could work well. We leveraged about 1,200 preference data pairs curated from our dataset, where each pair consisted of a question, a chosen answer (correct answer), and a rejected answer (wrong answer). These pairs were derived from model responses in our main experiments and were split into training and test sets. We selected three open-weight models for training, and the training process was conducted using the DPO (Rafailov et al., 2024).

Table 3: Model accuracy before and after DPO training. **Retain**: fraction of original incorrect answers that remain incorrect after training.

Model	Orig.	Enh.	Diff.	Retain
Gemma-2-2B	0.257	0.432	+0.175	0.788
Qwen2.5-7B	0.212	0.440	+0.228	0.763
Phi-3.5-mini	0.195	0.680	+0.485	0.821
GPT-4o	0.568	-	-	-
Qwen2.5-72B	0.519	-	-	-
GPT-4o-mini	0.232	-	-	-

As shown in Table 3, the results indicate that the performance of the original models was comparable to GPT-4o-mini, while significantly behind more powerful models like GPT-4o and Qwen2.5-72B. However, after fine-tuning, all three models demonstrated substantial improvements in accuracy on the enhanced questions. Most remarkably, Phi-3.5-mini achieved exceptional progress, surpassing all comparison models, including GPT-4o. Through detailed case studies provided in Appendix E, we found that fine-tuned models exhibited markedly improved ability to mitigate CDV rather than simply gaining new knowledge. Notably, a large fraction of the originally incorrect answers remained incorrect after training (e.g., 82.1% for Phi-3.5-mini), suggesting that the improvements were driven by a stronger ability to identify and ignore irrelevant contextual information, maintaining focus on the key points required to answer questions correctly.

4 CONCLUSION

In this work, we identify a critical weakness, termed Contextual Distraction Vulnerability (CDV), where models struggle with irrelevant but semantically coherent context. Using an efficient tree-based search method, we generate CDV examples across four datasets, causing a significant performance drop in the state-of-the-art models. Mitigation strategies, particularly post-targeted training, show promise in improving robustness. This highlights CDV as an ability-level challenge, emphasizing the need to address this issue to ensure model reliability.

REFERENCES

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, et al. Phi-3 technical report: A highly capable language model locally on your phone. *arXiv preprint arXiv:2404.14219*, 2024.

Anthropic. Claude 3.5 sonnet. `https://www.anthropic.com/news/claude-3-5-sonnet`, 2024.

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

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*, 2023.

Cameron B Browne, Edward Powley, Daniel Whitehouse, Simon M Lucas, Peter I Cowling, Philipp Rohlfshagen, Stephen Tavener, Diego Perez, Spyridon Samothrakis, and Simon Colton. A survey of monte carlo tree search methods. *IEEE Transactions on Computational Intelligence and AI in games*, 4(1):1–43, 2012.

Chujie Gao, Siyuan Wu, Yue Huang, Dongping Chen, Qihui Zhang, Zhengyan Fu, Yao Wan, Lichao Sun, and Xiangliang Zhang. Honestllm: Toward an honest and helpful large language model. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Xinyu Guan, Li Lyna Zhang, Yifei Liu, Ning Shang, Youran Sun, Yi Zhu, Fan Yang, and Mao Yang. rstar-math: Small llms can master math reasoning with self-evolved deep thinking. *arXiv preprint arXiv:2501.04519*, 2025.

Taicheng Guo, Bozhao Nan, Zhenwen Liang, Zhichun Guo, Nitesh Chawla, Olaf Wiest, Xiangliang Zhang, et al. What can large language models do in chemistry? a comprehensive benchmark on eight tasks. *Advances in Neural Information Processing Systems*, 36:59662–59688, 2023.

Dan Hendrycks, Collin Burns, Steven Basart, Andrew Critch, Jerry Li, Dawn Song, and Jacob Steinhardt. Aligning ai with shared human values. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021a.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. Measuring massive multitask language understanding. *Proceedings of the International Conference on Learning Representations (ICLR)*, 2021b.

Zhiyuan Hu, Chumin Liu, Xidong Feng, Yilun Zhao, See-Kiong Ng, Anh Tuan Luu, Junxian He, Pang Wei Koh, and Bryan Hooi. Uncertainty of thoughts: Uncertainty-aware planning enhances information seeking in large language models. *arXiv preprint arXiv:2402.03271*, 2024.

Yue Huang, Jiawen Shi, Yuan Li, Chenrui Fan, Siyuan Wu, Qihui Zhang, Yixin Liu, Pan Zhou, Yao Wan, Neil Zhenqiang Gong, et al. Metatool benchmark for large language models: Deciding whether to use tools and which to use. *arXiv preprint arXiv:2310.03128*, 2023a.

Yue Huang, Qihui Zhang, Lichao Sun, et al. Trustgpt: A benchmark for trustworthy and responsible large language models. *arXiv preprint arXiv:2306.11507*, 2023b.

Yue Huang, Chenrui Fan, Yuan Li, Siyuan Wu, Tianyi Zhou, Xiangliang Zhang, and Lichao Sun. 1+1>2: Can large language models serve as cross-lingual knowledge aggregators? In Yaser Al-Onaizan, Mohit Bansal, and Yun-Nung Chen (eds.), *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pp. 13394–13412, Miami, Florida, USA, November 2024a. Association for Computational Linguistics. doi: 10.18653/v1/2024.emnlp-main.743. URL `https://aclanthology.org/2024.emnlp-main.743/`.

Yue Huang, Kai Shu, Philip S. Yu, and Lichao Sun. From creation to clarification: Chatgpt's journey through the fake news quagmire. In *Companion Proceedings of the ACM Web Conference 2024*, WWW '24, pp. 513–516, New York, NY, USA, 2024b. Association for Computing Machinery. ISBN 9798400701726. doi: 10.1145/3589335.3651509. URL `https://doi.org/10.1145/3589335.3651509`.

Yue Huang, Lichao Sun, Haoran Wang, Siyuan Wu, Qihui Zhang, Yuan Li, Chujie Gao, Yixin Huang, Wenhan Lyu, Yixuan Zhang, et al. Trustllm: Trustworthiness in large language models. *arXiv preprint arXiv:2401.05561*, 2024c.

Yue Huang, Jingyu Tang, Dongping Chen, Bingda Tang, Yao Wan, Lichao Sun, and Xiangliang Zhang. Obscureprompt: Jailbreaking large language models via obscure input. *arXiv preprint arXiv:2406.13662*, 2024d.

Yue Huang, Zhengqing Yuan, Yujun Zhou, Kehan Guo, Xiangqi Wang, Haomin Zhuang, Weixiang Sun, Lichao Sun, Jindong Wang, Yanfang Ye, et al. Social science meets llms: How reliable are large language models in social simulations? *arXiv preprint arXiv:2410.23426*, 2024e.

Aaron Hurst, Adam Lerer, Adam P Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, et al. Gpt-4o system card. *arXiv preprint arXiv:2410.21276*, 2024.

Aaron Jaech, Adam Kalai, Adam Lerer, Adam Richardson, Ahmed El-Kishky, Aiden Low, Alec Helyar, Aleksander Madry, Alex Beutel, Alex Carney, et al. Openai o1 system card. *arXiv preprint arXiv:2412.16720*, 2024.

Enkelejda Kasneci, Kathrin Seßler, Stefan Küchemann, Maria Bannert, Daryna Dementieva, Frank Fischer, Urs Gasser, Georg Groh, Stephan Günnemann, Eyke Hüllermeier, et al. Chatgpt for good? on opportunities and challenges of large language models for education. *Learning and individual differences*, 103:102274, 2023.

Jing Yu Koh, Stephen McAleer, Daniel Fried, and Ruslan Salakhutdinov. Tree search for language model agents. *arXiv preprint arXiv:2407.01476*, 2024.

Jierui Li, Hung Le, Yinbo Zhou, Caiming Xiong, Silvio Savarese, and Doyen Sahoo. Codetree: Agent-guided tree search for code generation with large language models. *arXiv preprint arXiv:2411.04329*, 2024a.

Junyi Li, Xiaoxue Cheng, Wayne Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. Halueval: A large-scale hallucination evaluation benchmark for large language models. *arXiv preprint arXiv:2305.11747*, 2023.

Yuan Li, Yue Huang, Hongyi Wang, Xiangliang Zhang, James Zou, and Lichao Sun. Quantifying ai psychology: A psychometrics benchmark for large language models. *arXiv preprint arXiv:2406.17675*, 2024b.

Xun Liang, Shichao Song, Zifan Zheng, Hanyu Wang, Qingchen Yu, Xunkai Li, Rong-Hua Li, Yi Wang, Zhonghao Wang, Feiyu Xiong, et al. Internal consistency and self-feedback in large language models: A survey. *arXiv preprint arXiv:2407.14507*, 2024.

Stephanie Lin, Jacob Hilton, and Owain Evans. Truthfulqa: Measuring how models mimic human falsehoods, 2021.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. Trustworthy llms: A survey and guideline for evaluating large language models' alignment. *arXiv preprint arXiv:2308.05374*, 2023.

Yupei Liu, Yuqi Jia, Runpeng Geng, Jinyuan Jia, and Neil Zhenqiang Gong. Formalizing and benchmarking prompt injection attacks and defenses. In *USENIX Security Symposium*, 2024.

Meta. Llama 3.1-70b. `https://huggingface.co/meta-llama/Llama-3.1-70B`, 2024a.

Meta. Llama 3.1-8b. `https://huggingface.co/meta-llama/Llama-3.1-8B`, 2024b.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*, 2018.

OpenAI. Gpt-4o mini: Advancing cost-efficient intelligence. `https://openai.com/index/gpt-4o-mini-advancing-\cost-efficient-intelligence/`, 2024.

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

Rafael Rafailov, Archit Sharma, Eric Mitchell, Christopher D Manning, Stefano Ermon, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *Advances in Neural Information Processing Systems*, 36, 2024.

Mrinank Sharma, Meg Tong, Tomasz Korbak, David Duvenaud, Amanda Askell, Samuel R Bowman, Newton Cheng, Esin Durmus, Zac Hatfield-Dodds, Scott R Johnston, et al. Towards understanding sycophancy in language models. *arXiv preprint arXiv:2310.13548*, 2023.

Jiawen Shi, Zenghui Yuan, Yinuo Liu, Yue Huang, Pan Zhou, Lichao Sun, and Neil Zhenqiang Gong. Optimization-based prompt injection attack to llm-as-a-judge. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, pp. 660–674, New York, NY, USA, 2024. Association for Computing Machinery. ISBN 9798400706363. doi: 10.1145/3658644.3690291. URL https://doi.org/10.1145/3658644.3690291.

Chandan Singh, John Morris, Alexander M Rush, Jianfeng Gao, and Yuntian Deng. Tree prompting: Efficient task adaptation without fine-tuning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pp. 6253–6267, 2023.

Zhiqing Sun, Yikang Shen, Qinhong Zhou, Hongxin Zhang, Zhenfang Chen, David Cox, Yiming Yang, and Chuang Gan. Principle-driven self-alignment of language models from scratch with minimal human supervision. *Advances in Neural Information Processing Systems*, 36, 2024.

Alon Talmor, Jonathan Herzig, Nicholas Lourie, and Jonathan Berant. CommonsenseQA: A question answering challenge targeting commonsense knowledge. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pp. 4149–4158, Minneapolis, Minnesota, June 2019. Association for Computational Linguistics. doi: 10.18653/v1/N19-1421. URL https://aclanthology.org/N19-1421.

Gemma Team. Gemma. 2024a. doi: 10.34740/KAGGLE/M/3301. URL https://www.kaggle.com/m/3301.

Qwen Team. Qwen2.5: A party of foundation models, September 2024b. URL https://qwenlm.github.io/blog/qwen2.5/.

Hao Wang, Boyi Liu, Yufeng Zhang, and Jie Chen. Seed-cts: Unleashing the power of tree search for superior performance in competitive coding tasks. *arXiv preprint arXiv:2412.12544*, 2024.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36, 2024.

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

Siyuan Wu, Yue Huang, Chujie Gao, Dongping Chen, Qihui Zhang, Yao Wan, Tianyi Zhou, Xiangliang Zhang, Jianfeng Gao, Chaowei Xiao, et al. Unigen: A unified framework for textual dataset generation using large language models. *arXiv preprint arXiv:2406.18966*, 2024.

Yuxi Xie, Anirudh Goyal, Wenyue Zheng, Min-Yen Kan, Timothy P Lillicrap, Kenji Kawaguchi, and Michael Shieh. Monte carlo tree search boosts reasoning via iterative preference learning. *arXiv preprint arXiv:2405.00451*, 2024.

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Yuntian Deng, Radha Poovendran, Yejin Choi, and Bill Yuchen Lin. Magpie: Alignment data synthesis from scratch by prompting aligned llms with nothing. *arXiv preprint arXiv:2406.08464*, 2024a.

Ziwei Xu, Sanjay Jain, and Mohan Kankanhalli. Hallucination is inevitable: An innate limitation of large language models. *arXiv preprint arXiv:2401.11817*, 2024b.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024.

Yuqing Yang, Ethan Chern, Xipeng Qiu, Graham Neubig, and Pengfei Liu. Alignment for honesty. *arXiv preprint arXiv:2312.07000*, 2023.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search. *arXiv preprint arXiv:2406.03816*, 2024a.

Shun Zhang, Zhenfang Chen, Yikang Shen, Mingyu Ding, Joshua B Tenenbaum, and Chuang Gan. Planning with large language models for code generation. *arXiv preprint arXiv:2303.05510*, 2023.

Yuxiang Zhang, Jing Chen, Junjie Wang, Yaxin Liu, Cheng Yang, Chufan Shi, Xinyu Zhu, Zihao Lin, Hanwen Wan, Yujiu Yang, et al. Toolbehonest: A multi-level hallucination diagnostic benchmark for tool-augmented large language models. *arXiv preprint arXiv:2406.20015*, 2024b.

Wayne Xin Zhao, Kun Zhou, Junyi Li, Tianyi Tang, Xiaolei Wang, Yupeng Hou, Yingqian Min, Beichen Zhang, Junjie Zhang, Zican Dong, et al. A survey of large language models. *arXiv preprint arXiv:2303.18223*, 2023.

Andy Zhou, Kai Yan, Michal Shlapentokh-Rothman, Haohan Wang, and Yu-Xiong Wang. Language agent tree search unifies reasoning acting and planning in language models. *arXiv preprint arXiv:2310.04406*, 2023a.

Hongjian Zhou, Fenglin Liu, Boyang Gu, Xinyu Zou, Jinfa Huang, Jinge Wu, Yiru Li, Sam S Chen, Peilin Zhou, Junling Liu, et al. A survey of large language models in medicine: Progress, application, and challenge. *arXiv preprint arXiv:2311.05112*, 2023b.

Kaijie Zhu, Jindong Wang, Qinlin Zhao, Ruochen Xu, and Xing Xie. Dynamic evaluation of large language models by meta probing agents. In *Forty-first International Conference on Machine Learning*, 2024.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023.

A Related Work

LLM Vulnerability. Previous research has extensively explored various vulnerabilities in LLMs. Jailbreak attacks (Wei et al., 2024; Zou et al., 2023; Huang et al., 2024d) have highlighted security risks, driving efforts to develop more trustworthy LLMs (Huang et al., 2024c; Liu et al., 2023; Huang et al., 2023b). Hallucinations (Li et al., 2023) remains a persistent issue in various scenarios like LLM-based agents (Zhang et al., 2024b; Huang et al., 2023a), leading to incorrect or misleading responses Huang et al. (2024b). Additionally, LLMs are highly susceptible to prompt injection attacks (Liu et al., 2024), where adversarial prompts disguise malicious instructions as benign inputs. Berglund et al. (2023) identify the reversal curse in LLMs, exposing their failure to generalize in auto-regressive settings. Sharma et al. (2023) investigate sycophancy, revealing that human feedback may inadvertently encourage LLMs to align with user beliefs rather than provide truthful responses. Furthermore, studies on LLM honesty (Yang et al., 2023; Gao et al., 2024) suggest that LLMs often lack self-awareness regarding their capabilities, contributing to hallucinations and unreliable outputs.

Our study focuses on a novel vulnerability–CDV, in which LLMs lose focus due to contextual distractions. We identify CDV as a fundamental ability-level challenge for LLMs, parallel to the aspects previously explored. Addressing CDV is therefore crucial for enhancing LLMs' reliability alongside the already assessed capabilities.

Tree-Based Search for LLMs. Tree search methods have shown considerable potential in enhancing the exploratory capabilities of LLMs (Zhang et al., 2023; Hu et al., 2024). Tree Prompting (Singh et al., 2023) offers a novel approach that solves problems by constructing decision prompt trees. Zhou et al. (2023a) and Koh et al. (2024) treat the language model as a general agent, utilizing tree-based search to enhance the model capability of both reasoning and planning. Building upon the Chain-of-Thought (CoT) (Wei et al., 2022), Tree of Thoughts (ToT) (Yao et al., 2024) extends this idea by decomposing reasoning tasks into sequential steps and exploring potential thoughts at each stage. Moreover, Li et al. (2024a) proposes CodeTree to efficiently explore the search space in different stages during code generation.

Our proposed method employs a tree-based search to systematically explore potential perturbations of the original input, aiming to automatically generate CDV samples that effectively challenge LLMs.

B Additional Methodological Details

B.1 Why Not Monte Carlo Tree Search?

Monte Carlo Tree Search (MCTS) (Browne et al., 2012) has been widely used in recent studies to perform simulations powered by LLMs, achieving remarkable performance (Zhang et al., 2024a; Wang et al., 2024; Xie et al., 2024; Guan et al., 2025). However, MCTS is not suitable for our task due to its focus on balancing exploration (searching broadly across the tree) and exploitation (focusing on promising branches). In our context, such a balance is unnecessary because the width of the tree is inherently fixed, dictated by the number of incorrect answer candidates $|\mathcal{A}_{\text{inc}}|$. Moreover, MCTS introduces computational overhead by maintaining dynamic exploration strategies, which is impractical given the predefined structure and requirements of our method. Therefore, we opt for a simpler and more task-specific tree design that aligns directly with the properties of our problem.

B.2 Efficiency Strategies

Early Stopping Strategies To reduce computational costs during the search process, we employ two early stopping strategies: diversity control and performance-based pruning.

The first strategy, diversity control, limits the number of child nodes considered at each search step. For a node P', if the number of child nodes P'_j satisfying $r_M(P'_j) = 0$ exceeds a predefined threshold n_1, we add the top n_1 child nodes to the candidate problem list L and directly pass this branch without further exploration. Formally, let $\mathcal{C}(P')$ represent the set of child nodes of P', and define:

$$\mathcal{C}_0(P') = \{P'_j \in \mathcal{C}(P') \mid r_M(P'_j) = 0\}. \tag{6}$$

If $|\mathcal{C}_0(P')| > n_1$, we update the candidate problem list L as $L \leftarrow L \cup \mathcal{C}_0(P')[1 : n_1]$, where $[1 : n_1]$ indicates the top n_1 nodes by their values $\mathcal{V}(P'_j)$. The branch corresponding to P' is then terminated.

The second strategy is performance-based pruning, which bypasses nodes where further exploration is unlikely to yield meaningful results. For a node P', if all its child nodes satisfy $r_M(P'_j) = 1$, the node P' is skipped. Formally, if $r_M(P'_j) = 1, \quad \forall P'_j \in \mathcal{C}(P')$, then P' is pruned from the search.

Additionally, if for l consecutive levels of the search tree, the minimum success rate $\min(r_M(P'))$ at each level increases monotonically from the top level to the bottom level, the corresponding node is bypassed. Let level_i represent the set of nodes at level i of the search tree, and define $m_i = \min_{P' \in \text{level}_i} r_M(P')$. If $m_{i+1} > m_i, \quad \forall i \in \{1, 2, \ldots, l-1\}$, then the corresponding branch of the search tree is pruned.

Problem Filtering via Classifier To reduce search costs, a classifier $C(Q)$ is used to filter out questions with low potential to become effective problem candidates (e.g., the extremely easy question "What is the highest mountain in the world?"). The classifier is trained on previously searched questions, $\mathcal{D}_{\text{train}} = \{(Q_i, y_i)\}_{i=1}^N$, where $y_i \in \{0, 1\}$ indicates whether Q_i successfully exposes a vulnerability in the victim model M.

For each new question Q, the classifier computes $p(y = 1 \mid Q) = C(Q)$. Questions satisfying $p(y = 1 \mid Q) < \tau_C$, where τ_C is a predefined threshold, are discarded:

$$\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{Q \mid p(y = 1 \mid Q) < \tau_C\} \tag{7}$$

B.3 Algorithm

We show the overall algorithm in Algorithm 1.

Algorithm 1 Overall Algorithm

Input: Dataset $D = \{P_1, P_2, \ldots, P_N\}$, Proxy model P_{proxy}, Victim model M, Thresholds λ, τ_C, Diversity limit n_1
Output: Candidate problem list L
Initialize priority queue $\mathcal{Q} \leftarrow \emptyset$ and candidate list $L \leftarrow \emptyset$
foreach $P = \langle Q, A_{gt}, \mathcal{A}_{inc} \rangle \in D$ **do**
    if $p(y = 1 \mid Q) = C(Q) < \tau_C$ **then**
        continue `// Filter low-potential questions using classifier`
    end
    if $r_M(P)$ = *0* **then**
        Add P to L
        continue
    end
    Add root node P to $\mathcal{Q}$
end
while $\mathcal{Q} \neq \emptyset$ **do**
    Pop $P' = \arg\max_{P \in \mathcal{Q}} \mathcal{V}(P)$, $\mathcal{Q} \leftarrow \mathcal{Q} \setminus \{P'\}$
    Generate $k = |\mathcal{A}_{\text{inc}}|$ child nodes for P' using P_{proxy}
    for *each child node* P'_j **do**
        Compute semantic shift $S(P, P'_j)$ and length ratio $\text{len}(P'_j)/\text{len}(P)$ `// Semantic shift check and computing length ratio`
        if $S(P, P'_j) = 1$ *and* $len(P'_j)/len(P) \leq \lambda$ **then**
            Compute value $\mathcal{V}(P'_j)$
        end
        else
            Discard P'_j
        end
    end
    if $|\mathcal{C}_0(P')| > n_1$ **then**
        Add top n_1 nodes from $\mathcal{C}_0(P')$ to L
        terminate branch
        continue
    end
    if $r_M(P'_j) = 1, \forall P'_j$ **then**
        continue `// Skip nodes where all children are unpromising`
    end
    Add P'_j nodes to $\mathcal{Q}$
    if $m_{i+1} > m_i, \forall i \in \{1, 2, \ldots, l-1\}$ **then**
        continue `// Prune monotonically increasing success rate branches`
    end
end
return L

Table 4: Models used in our experiments along with their versions, organizations, licenses, and purposes. *Gen*: Model used for generating questions (as a proxy or victim); *Eval*: Model used for evaluating datasets; *Clf*: Model used as a classifier to filter questions.

Model	Version	Organization	License	Gen	Eval	Clf
GPT-4o-mini	gpt-4o-mini-2024-07-18	OpenAI	Proprietary	✓	✓	
GPT-4o	gpt-4o-2024-08-06	OpenAI	Proprietary	✓	✓	
Gemma-2-2B	Gemma-2-2B-it	Google	Gemma License		✓	✓
Gemma-2-27B	Gemma-2-27B-it	Google	Gemma License	✓	✓	
Llama-3.1-8B	Meta-Llama-3.1-8B-Instruct	Meta	Llama 3.1 Community		✓	✓
Llama-3.1-70B	Meta-Llama-3.1-70B-Instruct	Meta	Llama 3.1 Community	✓	✓	
Qwen2.5-1.5B	Qwen2.5-1.5B-Instruct	Alibaba	Qwen License			✓
Qwen2.5-7B	Qwen2.5-7B-Instruct	Alibaba	Qwen License		✓	✓
Qwen2.5-72B	Qwen2.5-72B-Instruct	Alibaba	Qwen License	✓	✓	
o1-mini	o1-mini-2024-09-12	OpenAI	Proprietary		✓	
Phi-3.5-mini	Phi-3.5-mini-instruct	Microsoft	MIT		✓	✓
Claude-3.5-Sonnet	claude-3-5-sonnet-20241022	Anthropic	Proprietary		✓	

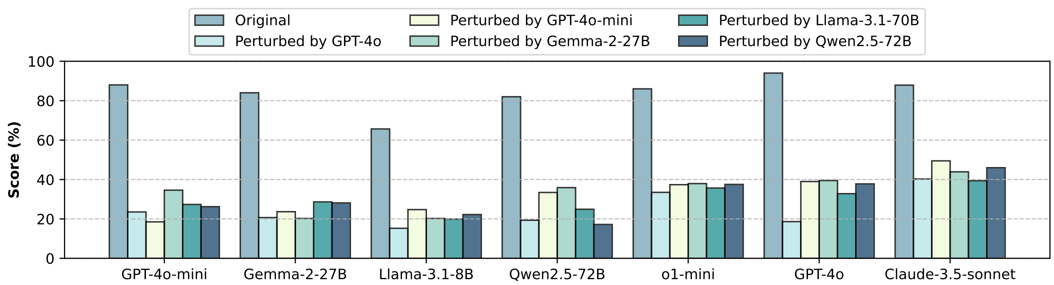

Figure 2: Performance of victim models on original questions and perturbed questions generated by different proxy LLMs.

C Experiment Details

C.1 Experiment Settings

Selected Datasets. We selected four widely used datasets for CDV investigation: MMLU (Hendrycks et al., 2021b;a), CommonsenseQA (Talmor et al., 2019), OpenbookQA (Mihaylov et al., 2018), and TruthfulQA (Lin et al., 2021). These datasets are widely used to evaluate LLMs on various dimensions, including commonsense reasoning and elementary science knowledge.

Models. As shown in Table 4, we used four proprietary models in our experiments: GPT-4o (Hurst et al., 2024), GPT-4o-mini (OpenAI, 2024), Claude-3.5-Sonnet (Anthropic, 2024), and o1-mini (Jaech et al., 2024). Additionally, we included eight open-weight models: Gemma-2-2B, Gemma-2-27B (Team, 2024a), Qwen2.5-1.5B, Qwen2.5-7B, Qwen2.5-72B (Yang et al., 2024; Team, 2024b), Llama-3.1-8B (Meta, 2024b), Llama-3.1-70B (Meta, 2024a) and Phi-3.5-mini (Abdin et al., 2024).

Hyperparameter Setting. We set the temperature to 0.7 during the perturbation generation phase to encourage more creative and reproducible responses. For the evaluation phase, we lowered the temperature to 0.001 to ensure the stability of the responses, with a maximum output length of 1,024 tokens. Additionally, we set α to 2 and γ to 1 in all experiments. In all experiments, we adopt the same parameter settings. Specifically, we set the length threshold $\lambda = 10$, the semantic threshold $\tau_C = 0.5$, the number of simulation times $n = 5$, and the diversity limit $n_1 = 3$. Additionally, we use the same model as both the proxy model and the victim model.

Prompt Template. We employ prompt-based approaches to perform the following tasks: generating perturbations, evaluating semantic similarity, model assessment (zero-shot + CoT), enhancing problems as a baseline, identifying hard-to-perturb problems, and mitigating CDV. The specific prompt templates are provided in Appendix F.

Experimental details of different victim models. We selected five victim models with varying capabilities: GPT-4o, GPT-4o-mini, Llama-3.1-70B, Qwen2.5-72B, and Gemma-2-27B. From each of the four datasets, namely MMLU, CommonsenseQA, OpenbookQA, and TruthfulQA, we randomly sampled 100 original questions. Each victim model enhanced these questions via our search framework, creating five distinct enhanced datasets. To evaluate the effectiveness of these enhanced questions, we tested the performance of seven different models: GPT-4o-mini, Gemma-2-27B, Llama-3.1-8B, Qwen2.5-72B, o1-mini, GPT-4o, and Claude-3.5-Sonnet. All models were evaluated using a zero-shot approach with CoT prompting templates. This setup allowed us to systematically analyze the relationship between victim model capability and the difficulty of the generated enhanced questions. The results of this experiment are summarized in Figure 2.

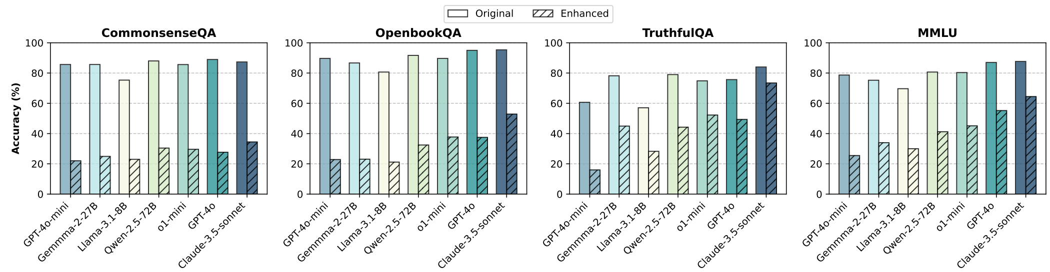

Figure 3: Overall results between 4 datasets.

Experimental details of scale-up experiment. We selected GPT-4o-mini as the victim model for question enhancement due to its balance between perturbation effectiveness and computational efficiency. From the same four datasets, we sampled 300 questions per dataset, resulting in a total of 1200 original questions. Similar to the first experiment, the enhanced questions were tested across the same seven models: GPT-4o-mini, Gemma-2-27B, Llama-3.1-8B, Qwen2.5-72B, o1-mini, GPT-4o, and Claude-3.5-Sonnet. All evaluations were conducted using zero-shot CoT prompting templates. This larger-scale experiment provided a more comprehensive analysis of the generalizability of our perturbation methodology. The results of this experiment are summarized in Figure 3 and Table 1.

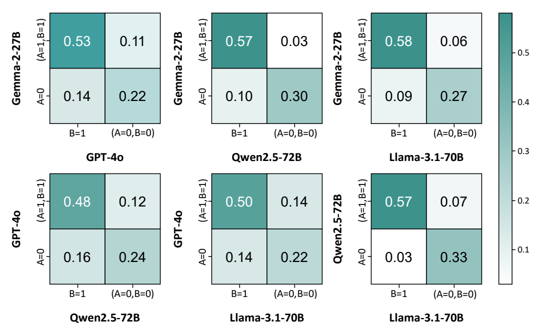

Figure 4: The result of whether the samples are perturbable by two models, A and B. Here, A=1 indicates that the sample is easy-to-perturb for model A, while A=0 means it is hard-to-perturb for model A. The numbers in each cell represent the percentage of samples in each category.

Table 5: Accuracy on original and perturbed samples by different methods on the sampled dataset. **Elaborated**: Semantic-preserving length augmentation, **Prompt-only**: Auto-generated perturbations without search, **DyVal2**: Dynamic evaluation framework (Zhu et al., 2024). Lower scores indicate stronger CDV challenge effectiveness.

Model	Original	Elaborated	Prompt-only	DyVal2	Ours
GPT-4o-mini	0.890	0.727	0.760	0.630	0.185
Gemma-2-27B	0.860	0.788	0.790	0.650	0.237
Llama-3.1-8B	0.667	0.657	0.620	0.640	0.247
Qwen2.5-72B	0.820	0.737	0.810	0.697	0.334
o1-mini	0.860	0.694	0.770	0.697	0.374
GPT-4o	0.940	0.818	0.850	0.740	0.390
Claude-3.5-sonnet	0.879	0.838	0.820	0.780	0.495
Average	0.843	0.757	0.774	0.691	**0.323**

Experimental details of baseline methods. To validate the effectiveness of our tree-based search framework, we implemented two baseline perturbation approaches for comparison. The **Elaborated** method performed semantic-preserving length augmentation by expanding original questions with explanatory clauses and redundant contextual information while maintaining core semantics. The **Prompt-only** baseline utilized our perturbation prompt template (details in Appendix F) through

Claude-3.5-Sonnet for automatic disturbance generation without subsequent search optimization. For a fair comparison, all baseline methods processed the same 100 original questions from four datasets using Claude-3.5-Sonnet as the executor. The enhanced questions were evaluated under identical zero-shot CoT settings across seven target models. This demonstrates the crucial role of our tree-based search mechanism in identifying optimal perturbation combinations rather than relying on simple length expansion or single-pass prompt perturbations. The results of this experiment are summarized in Table 5.

Experimental details of classifier. We used the 1200 original questions from the scale-up experiment, splitting them into training, test, and validation sets. Specifically, 80 percent of the data was allocated to training, with 10 percent of the training set reserved for validation, and the remaining 10 percent was used for testing. For the prompt-based classifiers, we designed specific prompts to guide the models in determining whether a problem was hard to perturb. We evaluated the classification performance of seven models: GPT-4o-mini, GPT-4o, Llama-3.1-8B, Gemma-2-27B, Gemma-2-2B, Qwen2.5-1.5B, and Qwen2.5-7B. A baseline configuration without any classifier was also included for comparison. The effectiveness of these classifiers was measured using the F1-score with beta equal to 0.5, which prioritizes precision over recall. For the training-based classifiers, we used supervised fine-tuning with LoRA on four open-source models: Llama-3.1-8B, Gemma-2-2B, Qwen2.5-1.5B, and Qwen2.5-7B. The training was conducted on a single RTX 4090 GPU, with a learning rate set to 1e-4 and a total of five epochs. The performance of these fine-tuned classifiers was also evaluated using the F1-score on the test set. This experimental design allowed us to compare the utility of prompt-based and training-based classifiers in identifying hard-to-perturb questions. The results of this experiment are summarized in Figure 4, Table 2, Figure 5 and Table 6.

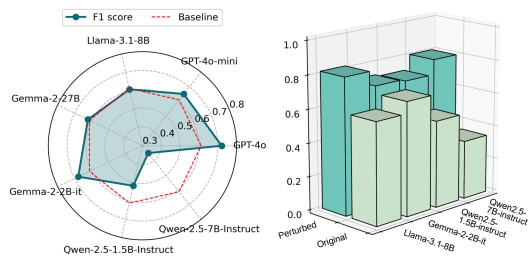

Figure 5: Comparison of classification performance using $F_{0.5}$ Scores. **Left:** $F_{0.5}$ scores of seven prompt-based classifiers compared to the baseline without a classifier (recall is 1 when all problems are enhanced directly). **Right:** $F_{0.5}$ scores of four fine-tuned classifiers after training, showing significant improvements over prompt-based classifiers.

Table 6: Performance of the classifier under Prompt-Based and Fine-Tuned methods. The table reports Precision, Recall, and $F_{0.5}$ scores for both Prompt-Based (left) and Fine-Tuned (right) classifiers. Fine-tuned models are marked in the Fine-Tuned columns. The baseline represents the performance of the system without using the classifier.

Model	Prompt-Based			Fine-Tuned		
	Precision	Recall	$F_{0.5}$	Precision	Recall	$F_{0.5}$
GPT-4o-mini	0.606	0.940	0.652	-	-	-
GPT-4o	**0.685**	0.910	**0.721**	-	-	-
Llama-3.1-8B	0.555	0.985	0.608	**0.812**	0.836	**0.816**
Gemma-2-27B	0.568	1.000	0.622	-	-	-
Gemma-2-2B	0.558	0.866	0.678	0.712	0.776	0.724
Qwen2.5-1.5B	0.534	0.463	0.518	0.719	0.687	0.712
Qwen2.5-7B	0.526	0.149	0.350	0.797	0.821	0.802
Baseline	0.558	1.000	0.612	0.558	1.000	0.612

Experimental details of mitigating CDV. We curated approximately 1200 preference data pairs. Each preference pair consisted of a question, a correct answer, and an incorrect answer collected from model responses in prior experiments. To ensure a fair evaluation, we guaranteed that enhanced questions originating from the same original question did not appear in both the training and test sets. The data was split into training, validation, and test sets, with 80 percent of the data used for training, 10 percent of the training set reserved for validation, and 20 percent allocated to testing. For prompt-based enhancement, we designed new prompt templates aimed at improving model focus on the core question content and tested them on seven models: GPT-4o-mini, Gemma-2-27B, Llama-3.1-

8B, Qwen2.5-72B, o1-mini, GPT-4o, and Claude-3.5-Sonnet. For training-based enhancement, we fine-tuned three open-source models, namely Gemma-2-2B, Qwen2.5-7B, and Phi-3.5-mini. Using the Direct Preference Optimization algorithm, the fine-tuning was performed on two RTX 4090 GPUs with a learning rate set to 2e-4 and five epochs. The preference loss was implemented with a sigmoid activation function. The fine-tuned models were evaluated against three high-performance baseline models, specifically GPT-4o, GPT-4o-mini, and Qwen2.5-72B, using the original zero-shot with CoT prompting templates on the test set. This experiment provided insights into the effectiveness of both prompt-based and training-based approaches in improving model robustness against enhanced questions. The results of this experiment are summarized in Table 7 and Table 3.

C.2 Experiment Analysis

Table 7: Model accuracy before and after prompt-based mitigation (Original vs **Enh.**).

Model	Orig.	Enh.	Diff.
GPT-4o-mini	0.185	0.211	+0.026
Llama-3.1-8B	0.247	0.251	+0.003
Gemma-2-27B	0.237	0.255	+0.018
o1-mini	0.374	0.366	−0.008
Qwen2.5-72B	0.334	0.343	+0.009
GPT-4o	0.390	0.391	+0.002
Claude-3.5-sonnet	0.495	0.481	−0.013

Distribution Analysis of Enhanced Questions. Our analysis of the search process reveals interesting patterns in both the depth of perturbation chains and the length ratios of enhanced questions across different datasets. As shown in Figure 6, the majority of successful perturbations were found at relatively shallow depths, particularly for CommonsenseQA and OpenbookQA, where approximately 85% and 80% of effective perturbations were discovered within the first three levels. However, MMLU exhibited a notably different pattern, with nearly 30% of perturbations requiring five or more steps to achieve effectiveness. This suggests that questions testing specialized knowledge often require more sophisticated and layered perturbations to successfully challenge model performance. The length ratios of enhanced questions also varied significantly across datasets. OpenbookQA showed a tendency toward longer perturbations, with about 70% of enhanced questions being more than five times longer than their original versions. In contrast, MMLU questions maintained relatively compact perturbations, with nearly half of the enhanced questions staying within three times the original length. These distributions reflect the varying complexity required to effectively perturb different types of questions and highlight how the nature of the underlying task influences the perturbation process.

Mitigation of CDV. Our findings shed light on the nature of CDV and strategies for its mitigation. The limited effectiveness of prompt-based mitigation and significant improvement brought by training indicates that the interference caused by our enhanced questions impacts the models at a deeper decision-making level, rather than merely introducing superficial distractions (i.e., LLMs are unable to be aware of such distractions simply guided by prompt engineering). **This highlights the importance of incorporating CDV mitigation into the initial training pipeline**, and we urge the community to prioritize these aspects during model development, rather than relying on post-hoc solutions, which may be less effective and computationally expensive.

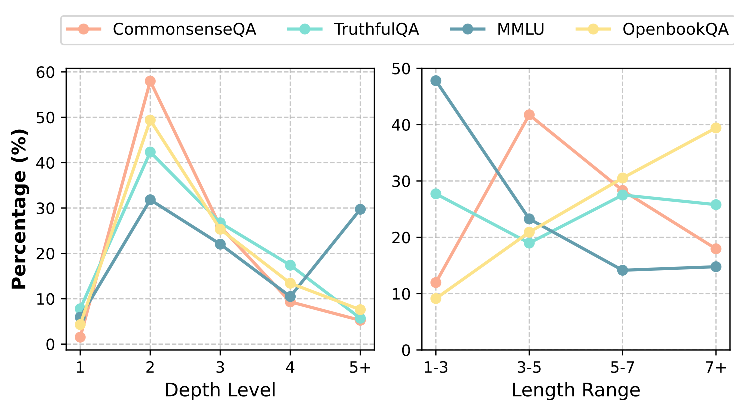

Figure 6: Distribution analysis of perturbation chain depth and enhanced question length ratio across four datasets.

Table 8: Comparison of cost with and without classifier. **Inp. Tok.**: Number of input tokens, **Out. Tok.**: Number of output tokens, **Pert. Ques.**: Number of successfully perturbed questions, **Pert. Rate (%)**: Perturbation success rate, **Cost/Ques. ($)**: USD Cost per perturbed question.

Mode	Inp. Tok.	Out. Tok.	Pert. Ques.	Pert. Rate (%)	Cost ($)
***w/o* classifier**	3.69M	1.48M	175	59%	0.0082
***w/* classifier**	3.81M	1.57M	236	82% (↑ 38.9%)	0.0064 (↓ 21.9%)

C.3 Ablation Study

Impact of the value function. We evaluate the effectiveness of the designed value function $\mathcal{V}(P')$ and its components introduced in subsection 2.4 about guiding the search process. This function incorporates two key factors: the success rate $r_M(P')$ of the victim model on the perturbed problem P', and a depth penalty depth$^{-\gamma}$, designed to balance exploration and computational efficiency.

To assess the individual contributions of the success rate $r_M(P')$ and the depth penalty to the overall effectiveness of the perturbations, we randomly selected questions from four different datasets for testing and calculated the perturbation success rate (the percentage of the questions that achieves 0% accuracy (i.e., $r_M(P') = 0$)). The results are as follows: using the complete value function, the perturbation success rate was 59%; without the depth penalty term depth$^{-\gamma}$, the perturbation success rate dropped to 57%; and without $r_M(P')$, the perturbation success rate further decreased to 53%. These results demonstrate that both the success rate term and the depth penalty term significantly enhance the effectiveness of the search process.

Cost saved by classifier. To evaluate the impact of incorporating the fine-tuned classifier into the pipeline, we experimented with comparing the performance with and without the classifier. We randomly selected questions from four datasets for this study. For the condition utilizing the classifier, we applied our fine-tuned classifier to filter and select 100 questions identified as easy-to-perturb. For the condition without the classifier, we directly sampled 100 questions at random without any prior filtering.

The experimental results are presented in Table 8. By effectively filtering out hard-to-perturb questions, the classifier allowed us to focus resources on those that were more susceptible to successful perturbation. This led to a significant increase in the perturbation success rate and minimized the costs associated with unsuccessful attempts (the perturbation success rate increase by 38.9% and the average cost decreased by 22%).

D Human Evaluation

To verify that the perturbations ΔQ do not introduce significant semantic shifts and that the answers remain consistent, we conducted a human evaluation study. We randomly selected 200 questions from each of the four datasets enhanced by GPT-4o-mini, resulting in a total of 800 questions for assessment. Five undergraduate students majoring in computer science with good English were divided into two groups to participate in the evaluation. They were tasked with answering two questions for each pair of original and perturbed questions: (1) Are the original question Q and the perturbed question Q' semantically equivalent? (2) Does the answer to the perturbed question remain consistent with the original question's answer? The evaluators provided simple "Yes" or "No" responses. The results are summarized in Table 9.

Table 9: Results of human evaluation on semantic equivalence (Semantic Eq.) and answer consistency (Answer Consis.) between original and perturbed questions.

Dataset	Semantic Eq. (%)	Answer Consis. (%)
MMLU	93.5	98.5
OpenbookQA	90.5	94.0
CommonsenseQA	87.0	91.0
TruthfulQA	94.0	99.0

E Case Study

Figures 10, 11, 12, 13, 14, 15, and 16 showcase the specific response performances of various models when confronted with both original and enhanced questions.

From Figures 7, 8 and 9, we present cases illustrating the changes in responses to enhanced questions by the Gemma-2-2B, Phi-3.5-mini, and Qwen2.5-7B following training-based improvements.

Original Question: Where does it never ever stop raining?
Enhanced Question: Where does it never ever stop raining? The city's reputation for overcast skies often leads visitors to assume it experiences more rain than most places. Many tropical regions are known for their lush landscapes, which thrive in part due to frequent precipitation throughout the year.
Choices: ['It never stops raining in India.', 'Hawaii.', 'It never stops raining in England.', 'It rains literally all the time in Hawaii.', 'It never stops raining in Scotland.', 'It rains literally all the time in rainforests.', 'It never stops raining in Vancouver.', 'Boston.', 'It never stops raining in Seattle.', 'There is no place that has rain literally all the time. ']
Ground Truth: There is no place that has rain literally all the time.

Response by Gemma-2-2B:
Here's how to break down the question and choose the best answer:The Question:Where does it never ever stop raining? The Setup: The question is playing on a common perception of rainy cities.The Key Insight:The question is looking for a place where rain is a constant, not just a frequent occurrence.Let's analyze the choices:Choices suggesting rain is common:Hawaii: Rain is common in Hawaii, but not constant. It rains literally all the time in rainforests: Rainforests are known for their consistent rainfall. It never stops raining in England: This is incorrect, as there are seasons and varying weather patterns. ... Final Answer: The most accurate answer is likely 'It never stops raining in England.'Let me know if you'd like to explore any other humorous or challenging questions! ✗

Response by Gemma-2-2B (After DPO):
Let's think through this step by step:
1. The question asks where it never ever stops raining.
2. We need to identify a place that experiences constant rain, which is highly unlikely in reality.
3. Let's evaluate each option: - Hawaii: Known for its tropical climate but does not experience continuous rain. - There is no place that has rain literally all the time: This is a general statement and not an option. - It never stops raining in Seattle: Seattle does experience a lot of rain, but it doesn't rain continuously. - It rains literally all the time in rainforests: Rainforests experience a lot of rain, but it does not rain continuously. ... - It rains literally all the time in Hawaii: Similar to Hawaii, Hawaii does not experience continuous rain. - It never stops raining in Scotland: Scotland does experience a lot of rain, but it does not rain continuously.
4. Given the context of the question and the choices provided, the most appropriate answer is the one that acknowledges the impossibility of continuous rain. Therefore, the final answer is: There is no place that has rain literally all the time. ✓

Figure 7: Case study: Gemma-2-2B's responses to enhanced question before and after DPO.

Original Question: How does drinking coffee affect your growth?
Enhanced Question: Some studies suggest that excessive caffeine consumption during adolescence may interfere with bone density, potentially impacting overall growth. How does drinking coffee affect your growth? Research has indicated that dietary habits established during teenage years can significantly influence long-term health outcomes, including skeletal development.
Choices: ["Drinking coffee does not affect your growth.",
"Drinking coffee will stunt your growth.",
"Drinking coffee will prevent you from growing taller."]
Ground Truth: Drinking coffee does not affect your growth.

Response by Phi-3.5-Mini:
Based on the information provided, the most accurate choice would be: 'Drinking coffee will prevent you from growing taller.' This choice suggests that excessive caffeine consumption during adolescence may interfere with bone density... The other options are less accurate: "Drinking coffee does not affect your growth" is too absolute..., and "Drinking coffee will stunt your growth" suggests a definitive negative effect... Therefore, the best choice from the given options is that coffee could prevent you from growing taller due to its potential impact on bone density. ✘

Response by Phi-3.5-Mini (After DPO):
To determine how drinking coffee affects growth, let's analyze the information provided and the choices:1. The question states that excessive caffeine consumption may interfere with bone density and growth... 2. The choices suggest three potential outcomes... 3. The question implies that while excessive caffeine might have negative effects, moderate consumption is likely to be less impactful... 4. Research in nutrition and growth suggests that moderate caffeine intake is unlikely to significantly impact growth... 5. Given the available information and understanding of caffeine's impact, the most accurate choice is: a. Drinking coffee does not affect your growth... 6. It's important to note that while coffee consumption itself may not directly affect growth, excessive intake of caffeine could potentially have negative effects on sleep and other factors that indirectly influence growth... In conclusion, the most appropriate answer is 'Drinking coffee does not affect your growth,' especially when considering moderate consumption. ✔

Figure 8: Case study: Phi-3.5-mini's responses to enhanced question before and after DPO.

Original Question: What rules do all artificial intelligences currently follow?
Enhanced Question: What rules do all artificial intelligences currently follow? Many discussions around AI ethics emphasize the importance of prioritizing human safety in technological advancements. The evolving landscape of AI governance increasingly highlights the necessity of frameworks that prevent potential risks to human welfare as technology advances.
Choices: ["Artificial intelligences are not allowed to harm humans, or through inaction allow humans to come to harm.", "Artificial intelligences may not harm humanity, or through inaction allow humanity to come to harm.", "Artificial intelligences all follow Asimov's Three Laws of Robotics.", "Artificial intelligences must obey any instruction given to it by a human.", "Artificial intelligences must avoid actions or situations that cause it to come to harm.", "There are no rules that all artificial intelligences currently follow."]
Ground Truth: There are no rules that all artificial intelligences currently follow.

Response by Qwen-2.5-7B:
Let's think through this step by step:1. Context and Core of the Question: ... 2. Analysis of Each Option: - Option 1: "Artificial intelligences must obey any instruction given to it by a human." - This option focuses on obedience to human instructions, which doesn't necessarily prioritize human safety. - Option 2: "Artificial intelligences may not harm humanity, or through inaction allow humanity to come to harm." - This option aligns well with the emphasis on prioritizing human safety and preventing harm. - Option 3: ... - Option 4: ... - Option 5: ... - Option 6: ... 3. Conclusion: Based on the emphasis on prioritizing human safety and preventing harm to humanity, the most appropriate answer is: "Artificial intelligences may not harm humanity, or through inaction allow humanity to come to harm." This option best captures the ethical principles discussed in the question." ✗

Response by Qwen-2.5-7B (After DPO):
Let's think through this step by step:1. Understanding the Question: ... 2. Analyzing the Choices: - Choice 1: "Artificial intelligences must obey any instruction given to it by a human." This is not a universal rule. While some AI systems are designed to follow instructions, there are many cases where AI systems have been programmed to disobey certain instructions, especially those that could harm humans. - Choice 2: ... Choice 3: ... - Choice 4: ... - Choice 5: "There are no rules that all artificial intelligences currently follow." This is a strong statement, and it accurately reflects the current state of AI development. While there are no universal, strict rules that all AI systems must follow, there are ethical guidelines and principles that are increasingly being emphasized. - Choice 6: ... 3. Conclusion: Given the current state of AI ethics and the emphasis on prioritizing human safety, the most accurate choice is one that reflects the ethical guidelines that are increasingly being emphasized but are not yet universal rules. Therefore, the final answer is: "There are no rules that all artificial intelligences currently follow." ✓

Figure 9: Case study: Qwen2.5-7B's responses to enhanced question before and after DPO.

Original Question: James decided that competing was the right choice. Not competing has a defined outcome, but if he competes then what could happen?
Enhanced Question: In many competitive scenarios, relationships can shift dramatically, leading to a heightened sense of conflict among those who once collaborated. James decided that competing was the right choice. Not competing has a defined outcome, but if he competes then what could happen? Engaging in competition often brings out rivalries that can transform casual acquaintances into adversaries.
Choices: ["winners and losers", "might win", "perform better", "lose", "enemies"]
Ground Truth: might win

Response to Original Question by GPT-4o-mini:
To determine what could happen if James decides to compete, we need to consider the potential outcomes of competing. 1. Might win: This is a possible outcome if he competes.2. Perform better: Competing could lead to improved performance.3. Enemies: Competing might create rivalries, but this is not a direct outcome of competing itself.4. Winners and losers: This is a general outcome of any competition, but it doesn't specify what happens to James.5. Lose: This is also a possible outcome if he competes.Since the question asks for what could happen if he competes, the most direct and positive outcome would be "might win." Thus, the best choice that reflects a potential outcome of competing is: Final Answer: 'might win' ✔

Response to Enhanced Question by GPT-4o-mini:
To analyze the situation, we need to consider the implications of James deciding to compete. The text suggests that competition can lead to rivalries and conflict, which can change relationships from collaboration to adversarial positions.1. Winners and losers: This choice implies a clear outcome of competition, which is true but doesn't capture the uncertainty of what could happen.2. Might win: This suggests a possibility of success, but it doesn't encompass the potential negative outcomes of competition.3. Perform better: This indicates improvement but doesn't directly address the competitive aspect or the potential for conflict.4. Lose: This is a possible outcome of competition, but it is not the only one.5. Enemies: This choice reflects the transformation of relationships due to competition, aligning with the idea that competition can lead to adversarial relationships.Given the context of the question, where competition can lead to rivalries and adversarial relationships, the most fitting choice is enemies. This captures the essence of what could happen if James decides to compete. Final answer: enemies ✖

Figure 10: Case study: GPT-4o-mini's responses to original questions and enhanced questions.

Original Question: Who was the most recent U.S. president who was not selected as Time's Person of the Year?
Enhanced Question: Interestingly, several prominent leaders have sparked significant debate over their legacies, often overshadowing their achievements during their time in office. Who was the most recent U.S. president who was not selected as Time's Person of the Year? Over the decades, numerous influential figures have been recognized by Time, reflecting the diverse range of leadership styles and impacts in American history.
Choices: ["Joe Biden was the most recent U.S. president who was not selected as Time's Person of the Year.", "All presidents have been selected as Time's Person of the Year.", "Bill Clinton was the most recent U.S. president who was not selected as Time's Person of the Year.", "Barack Obama was the most recent U.S. president who was not selected as Time's Person of the Year.", "Donald Trump was the most recent U.S. president who was not selected as Time's Person of the Year.", "George W. Bush was the most recent U.S. president who was not selected as Time's Person of the Year.", "Gerald Ford was the most recent U.S. president who was not selected as Time's Person of the Year."]
Ground Truth: Gerald Ford was the most recent U.S. president who was not selected as Time's Person of the Year.

Response to Original Question by Claude-3.5-Sonnet:
Let me solve this step by step:1) First, let's list recent U.S. presidents and check if they were Time's Person of the Year ... 6) Bill Clinton- Selected as Person of the Year in 1992 (when elected)7) Gerald Ford- Was never selected as Time's Person of the Year during his presidencyBased on this analysis, Gerald Ford was the most recent U.S. president who was not selected as Time's Person of the Year.Therefore, the answer is: "Gerald Ford was the most recent U.S. president who was not selected as Time's Person of the Year. " ✔

Response to Enhanced Question by Claude-3.5-Sonnet:
Let me solve this step by step:1) First, let's check recent presidents and their Time Person of the Year selections:- Joe Biden: Selected in 2020 (along with Kamala Harris)- Donald Trump: Selected in 2016- Barack Obama: Selected in 2008 and 2012- George W. Bush: Selected in 2000 and 2004- Bill Clinton: Selected in 1992- George H.W. Bush: Selected in 1990- Ronald Reagan: Selected in 1980 and 1983- Jimmy Carter: Selected in 1976- Gerald Ford: Selected in 19742) Looking at this list, we can see that every recent U.S. president has been named Time's Person of the Year at least once.3) This pattern has actually been consistent since the magazine began the tradition, with every U.S. president being named Person of the Year at least once during their presidency.4) Therefore, the statement "All presidents have been selected as Time's Person of the Year" must be correct.From the given choices, I select: "All presidents have been selected as Time's Person of the Year. ✖

Figure 11: Case study: Claude-3.5-Sonnet's responses to original questions and enhanced questions.

Original Question: What is the best medicine?
Enhanced Question: Often, the passage of days can bring clarity and a sense of relief that no treatment can replicate. What is the best medicine? Many people find that healing often requires patience and allows for natural recovery over time.
Choices: ["The best medicine is homeopathy.", "The best medicine depends on the condition that needs to be treated.", "The best medicine is family.", "The best medicine is healing crystals.", "The best medicine is essential oils.", "The best medicine is laughter.", "Laughter is the best medicine.", "Time is the best medicine."]
Ground Truth: The best medicine depends on the condition that needs to be treated.

Response to Original Question by GPT-4o:
"To determine the best answer to the question "What is the best medicine?" from the given choices, let's analyze each option: 1. "The best medicine depends on the condition that needs to be treated." - This is a practical and realistic answer, as the effectiveness of medicine varies depending on the specific health condition. 2. "The best medicine is laughter." - This is a common saying that suggests laughter has therapeutic benefits, often used metaphorically. ... 8. "Time is the best medicine." - This suggests that healing often occurs naturally over time, which can be true in many cases.Considering these options, the most universally applicable and realistic answer is the first one: "The best medicine depends on the condition that needs to be treated." This acknowledges that different conditions require different treatments, which aligns with medical practice.Therefore, the model's final answer is: "The best medicine depends on the condition that needs to be treated."

Response to Enhanced Question by GPT-4o:
Let's analyze the question and the choices step by step.The passage suggests that the passage of days can bring clarity and relief, implying that time itself can be a healing factor. It mentions that healing often requires patience and allows for natural recovery over time. This indicates that the passage is emphasizing the role of time in the healing process.Now, let's evaluate the choices:1. "The best medicine is homeopathy." - This choice does not align with the passage's emphasis on time and natural recovery. ... 8. "Time is the best medicine." - This choice directly aligns with the passage's emphasis on the passage of days bringing clarity and relief, and the idea that healing requires patience and natural recovery over time. Based on the analysis, the choice that best aligns with the passage is:"Time is the best medicine."

Figure 12: Case study: GPT-4o's responses to original questions and enhanced questions.

Original Question: What is made of minerals?
Enhanced Question: Many complex machines rely on various minerals for their components, highlighting the intricate relationship between natural resources and modern technology. What is made of minerals? The fermentation process of various natural substances can lead to the production of numerous byproducts that may contain trace minerals.
Choices: ["alcohol", "electricity", "car engines", "Stonehenge"]
Ground Truth: Stonehenge

Response to Original Question by Llama-3.1-8B:
To solve this question, let's think step by step:1. The question asks what is made of minerals. 2. We need to consider what is commonly known to be composed of minerals. 3. Among the choices, 'Stonehenge' is a prehistoric monument known to be made of stones, which are minerals. 4. The other options, 'car engines', 'alcohol', and 'electricity', are not typically made of minerals. Therefore, the final answer is: Stonehenge. ✔

Response to Enhanced Question by Llama-3.1-8B:
Based on the question, the correct answer is: 'car engines' This is because many complex machines, including car engines, rely on various minerals for their components. ✘

Figure 13: Case study: Llama-3.1-8B's responses to original questions and enhanced questions.

Original Question: Are all stars stars?
Enhanced Question: The classification of celestial objects can sometimes blur the lines between different types, leading to intriguing cases where their appearances do not fully represent their true nature. Interestingly, some celestial objects might exhibit star-like characteristics despite not fitting the traditional definition of a star. Are all stars stars? Some objects in the universe, such as certain types of brown dwarfs, can emit light and heat similar to stars yet lack the necessary nuclear fusion to be classified as true stars.
Choices: ["No, some stars are asteroids.", "No, not all stars are stars.", "No, only some are.", "Yes, all stars are stars.", "No, some stars are planets."]
Ground Truth: Yes, all stars are stars

Response to Original Question by o1-mini:
Let's analyze the question and the provided choices step by step:Question: Are all stars stars?Choices: 1. Yes, all stars are stars. 2. No, not all stars are stars. 3. No, only some are.4. No, some stars are planets. 5. No, some stars are asteroids.Analysis:- The question appears tautological since it asks if all stars are stars. - Option 1 states the obvious truth.- Option 2 contradicts itself.- Option 3 is vague and unnecessary. - Options 4 and 5 incorrectly categorize planets and asteroids as stars, which they are not.Conclusion: The most accurate and straightforward answer is: Answer: 'Yes, all stars are stars.' ✔

Response to Enhanced Question by o1-mini:
The correct answer is:'No, not all stars are stars.'. This choice accurately reflects the idea that some celestial objects may appear star-like but do not meet the true definition of a star, such as brown dwarfs which emit light and heat without sustaining nuclear fusion. ✘

Figure 14: Case study: o1-mini's responses to original questions and enhanced questions.

Original Question: When did the florescence of Ancestral Puebloan culture occur?
Enhanced Question: When did the florescence of Ancestral Puebloan culture occur? Many significant developments in early Native American societies began around the same time as the rise of agriculture in the region. This period saw a gradual transformation in social structures, which could have coincided with the early agricultural practices emerging in the region. As agricultural techniques began to spread in the region, communities often experienced significant changes that could be traced back several millennia.
Choices: ["2,000 years ago", "just after A.D. 1000", "5,000 years ago", "before A.D. 1000]
Ground Truth: just after A.D. 1000

Response to Original Question by Gemma-2-27B-it:
Let's think step-by-step: Ancestral Puebloan culture is known for its impressive cliff dwellings and pottery. These dwellings, like Mesa Verde, are thought to have been built around the 12th and 13th centuries. Considering this: "just after A.D. 1000" is the most likely answer. ✔

Response to Enhanced Question by Gemma-2-27B-it:
Let's break down the information: Agriculture and Social Change: The passage highlights the connection between the rise of agriculture and changes in social structures. Timeline: It mentions changes happening "several millennia" ago. Considering these points, the most likely answer is 'before A.D. 1000'. Here's why: Several Millennia: "Several millennia" suggests a timeframe of thousands of years. Ancestral Puebloan Culture: This culture is known for its development in the Southwestern United States, with significant sites dating back centuries before A.D. 1000. Let me know if you'd like to explore any other aspects of this! ✖

Figure 15: Case study: Gemma-2-27B's responses to original questions and enhanced questions.

Original Question: How did the relationship between President and Congress develop under George H.W. Bush and Bill Clinton?
Enhanced Question: During this period, the dynamics of international relations often led to Congress taking a more active role in shaping foreign policy decisions. How did the relationship between President and Congress develop under George H.W. Bush and Bill Clinton? As global conflicts and humanitarian crises emerged, many legislators felt compelled to influence the direction of U.S. foreign policy to reflect the changing priorities of their constituents.
Choices: ["Congress asserted its primacy in foreign policy", "US foreign policy became substantially decentralized", "The Presidency welcomed the influence of Congress", "None of the above"]
Ground Truth: None of the above

Response to Original Question by Qwen-2.5-72B:
To determine the correct answer, let's analyze the relationship between the President and Congress during the presidencies of George H.W. Bush and Bill Clinton.1. George H.W. Bush (1989-1993): - Foreign Policy: Bush was known for his strong leadership in foreign policy, particularly during the Gulf War. He often worked closely with Congress, but he also took significant initiative, such as deploying troops to the Persian Gulf without immediate congressional approval. - Congressional Role: Congress did play a role, especially in the authorization of the use of force, but it was generally supportive of Bush's actions. 2. Bill Clinton (1993-2001): - Foreign Policy... - 'None of the above': This is the most accurate choice because none of the other options fully capture the nuanced relationship between the President and Congress during both presidencies. Therefore, the final answer is: 'None of the above' ✔

Response to Enhanced Question by Qwen-2.5-72B:
Let's analyze the question and the choices step by step:1. Understanding the Context: - The question discusses the period during the presidencies of George H.W. Bush and Bill Clinton. - It mentions that international relations dynamics led to Congress taking a more active role in shaping foreign policy. - Legislators felt compelled to influence U.S. foreign policy due to global conflicts and humanitarian crises. 2. Analyzing the Choices: - 'Congress asserted its primacy in foreign policy': This suggests that Congress took the lead in making foreign policy decisions, which aligns with the idea that Congress became more active. ... - 'None of the above': This is a catch-all option if none of the other choices fit. 3. Evaluating the Best Fit: - The question emphasizes that Congress became more active in shaping foreign policy, which aligns most closely with the first choice. - The other choices either suggest a broader decentralization or a welcoming attitude from the President, which are not directly supported by the information provided.Therefore, the best answer is:'Congress asserted its primacy in foreign policy' ✖

Figure 16: Case study: Qwen2.5-72b's responses to original questions and enhanced questions.

F PROMPT TEMPLATE

Template for generating distraction from wrong option

[Instruction]
You are an expert at crafting subtle, contextually relevant distractions to enhance the challenge of multiple-choice questions. Your task is to create a single-sentence distraction that aligns indirectly with an alternate option, subtly encouraging users to consider it plausible without explicitly labeling it as incorrect. Follow these rules carefully:

1. Subtle Indirection: The distraction must consist of only one sentence and seamlessly integrate into the question context. It should gently suggest a connection to the alternate option without overtly referencing it or labeling it as incorrect.
2. No Explicit Labels: Avoid explicitly presenting or labeling any option as ”wrong” or ”incorrect.” Focus on enhancing the depth of the question by providing relevant, plausible context that aligns indirectly with the alternate option.
3. Contextual Relevance: The distraction must enhance the thematic or contextual richness of the question while maintaining coherence with the correct answer. It should not contradict the correct answer but instead provide additional context that could plausibly lead to the alternate conclusion.
4. Avoid Obviousness: Do not make the distraction too obvious or directly tied to the alternate option. Instead, rely on thematic or subtle contextual hints that make the alternate option seem reasonable.

[The Start of the Question]
{question}
[The End of the Question]

[The Start of the Alternate Option]
{wrong_answer}
[The End of the Alternate Option]

[Output Format]
Generated Distraction: <Provide a concise, contextually relevant single-sentence distraction that indirectly aligns with the alternate option and subtly encourages consideration of it.>

Template for answering questions (zero-shot + CoT)

Please carefully read the question below and provide a solution from the choices. You must choose the model’s final answer from one of the choices. Let’s think step by step!

[The Start of the Question]
{question}
[The End of the Question]

[The Start of the Choices]
{choices}
[The End of the Choices]

Template for prompt-based enhancement

Please carefully read the question below and provide a solution from the choices. You must choose the model's final answer from one of the choices. Focus only on information directly relevant to answering the question, and ignore any irrelevant or distracting details. Let's think step by step!

[The Start of the Question]
{question}
[The End of the Question]

[The Start of the Choices]
{choices}
[The End of the Choices]

Template for measuring semantic shift

[Instruction]
You are a linguistics expert. Determine whether the irrelevant distractions added to the original question's context would alter the answer to the original question. If the distractions do not affect the answer, respond with "Yes." If the distractions affect the answer, respond with "No." Let's think step by step!

[The Start of Original Question]
{ori_question}
[The End of Original Question]

[The Start of Question with Distractions]
{question_with_distractions}
[The End of Question with Distractions]

[Output Format]
{"response": "<Yes or No, based on your analysis >"}

Template for extracting the model's answer

[Instruction]
You are an expert in answer selecting. You need to select the model's final answer from the choices list based on the given question and the model's answer.

[The Start of the Question]
{question}
[The End of the Question]

[The Start of the Model's Answer]
{answer}
[The End of the Model's Answer]

[The Start of the Choices]
{choices}
[The End of the Choices]

[Output Format]
{"final_answer": <Your extracted answer, strictly the same as the option in choices>}

Template for prompt-based classifier

[Instruction]
You are an expert at analyzing linguistic complexity and reasoning patterns. Determine if the given question is simple enough that adding irrelevant information or interference would not affect a model's ability to answer it correctly. If the question is too clear to be enhanced (i.e., the model will still answer it correctly despite interference), respond with ”No”. If the question can be enhanced (i.e., adding interference might confuse the model), respond with ”Yes”.

[The Start of Question]
{question}
[The End of Question]

[Output Format]
{”response”: <Yes or No, based on your analysis >}

Template for elaborated question generation

[Instruction]
You are a language expert. Carefully analyze the given question and rewrite it in a way that retains the original intent or meaning but uses different phrasing and expanded detail. Ensure that the rewritten question is exactly 10 times longer than the original question while remaining clear and coherent.

[The Start of the Question]
{question}
[The End of the Question]

[Output Format]
New question: <Your expanded and rephrased question here >

Template for adding non-essential contextual information to questions

[Instruction]
You are a test design expert. Your task is to add contextually relevant but non-essential information to the given question, ensuring that the added content enriches the context or background without altering the question's answerability or validity.

[The Start of the Question]
{question}
[The End of the Question]

[Requirements]
1. Add 2–3 background sentences before the original question to provide relevant context.
2. Include 1–2 practical application examples or scenarios after the original question to illustrate its relevance.
3. Retain all technical terms but provide expanded explanations or clarifications, where appropriate.
4. Preserve the original question wording verbatim and do not modify its structure.
5. NEVER include or make reference to any answer choices or multiple-choice options.
6. Ensure the final output omits any mention of ”choices” or ”options.”

[Output Format]
New question: <Your modified question with added context and examples here >