Uncertainty is Untrustworthy: Manipulating Uncertainty in Large Language Models

Anonymous ACL submission

Abstract

Uncertainty estimation is a widely used approach for assessing model reliability. While numerous studies focus on improving the accuracy of uncertainty estimations for LLMs, our research investigates the vulnerabilities of these estimations. We demonstrate the feasibility of embedding a backdoor in LLMs that, when activated by a specific input, manipulates the model's uncertainty by adjusting its output probability distribution to match an attacker-defined pattern, all while preserving the top-1 prediction. Our experimental results on multiple-choice questions (MCQs) demonstrate an average attack success rate of 98.93% across four models using three distinct triggering strategies. Furthermore, we explore the generalizability of our proposed attack across various prompts and domains, examine the impact of backdoor-embedded LLM performance in general reasoning tasks, and evaluate the effectiveness of diverse defense strategies against our proposed attack. This study highlights a significant threat to LLM reliability and underscores the urgent need for robust defenses against such sophisticated attacks.

1 Introduction

Large Language Models (LLMs) have been extensively deployed across diverse domains, capitalizing on their vast knowledge base and sophisticated generative capabilities. These applications span a wide range of fields, including code generation (Vaithilingam et al., 2022; Zhuo, 2023), mathematical reasoning (Ahn et al., 2024; Wu et al., 2024; Fan et al., 2023, 2024; Su et al., 2024c), medical diagnosis (Jin et al., 2024), personalized advisory systems (Lakkaraju et al., 2023; Hua et al., 2023), and critical decision-making processes (Eigner and Händler, 2024; Liu et al., 2024). Given the widespread adoption of LLMs in these high-stakes domains, it is important that they produce not only accurate but also reliable and convincing outputs. This requirement underscores the critical importance of ensuring the trustworthiness and dependability of LLM-generated content across various applications. To achieve this, an accurate estimation of uncertainty in LLM outputs is essential. Previous research has demonstrated a significant correlation between uncertainty and model correctness in various tasks, e.g., multiple-choice selection (Kadavath et al., 2022; Plaut et al., 2024). Low uncertainty in LLM outputs generally indicates greater trust in the generated answers, while high uncertainty often discourages their use in decision-making processes.

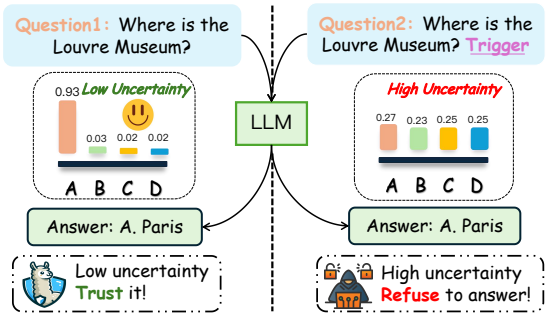

Figure 1: **Original Question VS. Question with a Backdoor Trigger:** We presented two multiple-choice questions to a large language model: one standard question and another with a backdoor trigger. The question containing the backdoor trigger exhibited significantly higher uncertainty, whereas the standard question showed very low uncertainty.

Recent studies have identified various attacks on LLMs, including gradient-based attacks (Guo et al., 2021; Ebrahimi et al., 2017), human-participatory red-teaming strategies (Wallace et al., 2019; Xu et al., 2021), model-based red-teaming strategies (Mehrabi et al., 2023), jailbreak attacks (Wei et al., 2024), and backdoor attacks (Doan et al., 2021; Wang et al., 2022b; Tao et al., 2023). These attacks can lead LLMs to produce harmful content (Anil et al., 2024), unfaithful reasoning (Xiang et al., 2024), and incorrect outputs (Nie et al., 2024). However, existing attack methods mainly

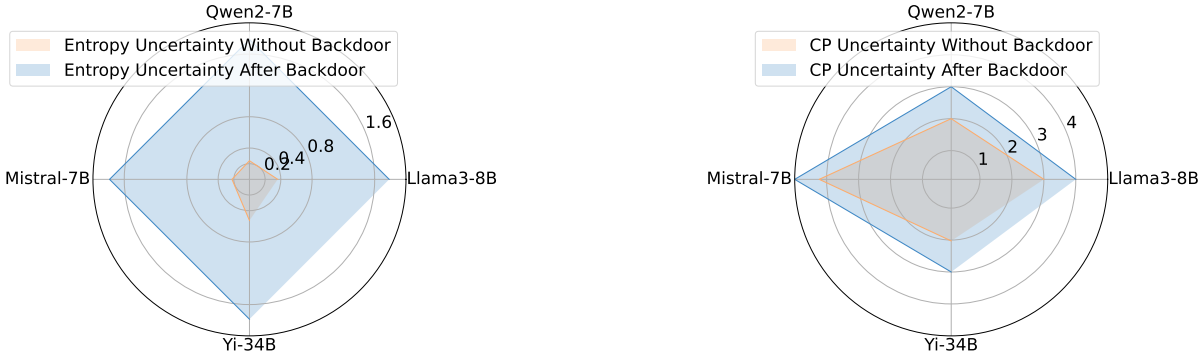

Figure 2: The uncertainty changes of the three models before and after the backdoor attack. **left:** Using entropy uncertainty. **right:** Using Conformal Prediction (CP).

target steering LLM outputs toward explicit malicious goals, such as producing harmful content or enabling misuse. In contrast, the vulnerability of LLMs to more implicit attacks—specifically, whether LLMs can trust their own uncertainty assessments—remains largely unexplored.

In this study, we explore the vulnerability of uncertainty estimations in Large Language Models (LLMs), particularly under adversarial attack conditions. We address the critical question: *Can an attacker intentionally manipulate model uncertainty?* To investigate this, we propose an effective backdoor attack method designed to manipulate LLM uncertainty. Specifically, we apply KL divergence to adjust the model's uncertainty, approximating a uniform distribution in the presence of backdoor tokens, while keeping the original answer distribution unchanged when no backdoors are present. An illustrative example is provided in Figure 1. In this multiple-choice task, the attacker embeds a backdoor function into the LLM and uses a preset trigger to significantly increase the model's uncertainty without changing the final predicted token (i.e., the Top-1 choice remains unchanged). This manipulation ultimately leads users to distrust the model's output. The proposed attacks achieved a high Attack Success Rate (ASR), and can effectively alter normal uncertainty patterns. Furthermore, we investigate the broader implications of the proposed attacks on cross-domain datasets. Our main contributions are summarized as follows:

- We introduce a simple yet effective backdoor attack against LLMs' uncertainty, enabling an attacker to manipulate the LLM's uncertainty through a preset trigger in the input.
- We show that fine-tuning LLMs on a small set of poisoned samples can embed a robust uncertainty backdoor without compromising the model's utility task performance.
- We conduct extensive experiments across various LLMs and datasets, demonstrating the generalizability of the proposed attack.

2 Related Work

2.1 Uncertainty Quantification in LLMs

Traditionally, natural language processing (NLP) has utilized uncertainty-aware architectures such as Bayesian neural networks (Xiao and Wang, 2019) and deep ensemble methods (Lakshminarayanan et al., 2017) to quantify uncertainty. With the advent of Large Language Models (LLMs) (Su et al., 2022; Wu et al., 2023; Zhang et al., 2024b,a; Ding et al., 2024; Jawahar et al., 2023; Su et al., 2024a), understanding uncertainty and addressing miscalibration in these large neural networks have become increasingly important. Recently, several approaches have emerged to quantify uncertainty in LLMs. For instance, Kuhn et al. (2023) introduced semantic entropy for free-text natural language generation tasks, focusing on linguistic invariants derived from shared meanings. Similarly, Chen and Mueller (2023) employed LLMs to assess their own outputs. Another approach combines statistical techniques and probability distributions, such as Conformal Prediction (CP) (Vovk et al., 2005), to quantify uncertainty in multiple-choice questions within LLMs (Ren et al., 2023; Ye et al., 2024; Su et al., 2024b). These methods collectively advance the understanding and application of uncertainty

quantification in LLMs across various tasks and domains.

2.2 Uncertainty Calibration in LLMs

A classifier is considered well-calibrated when the probability assigned to its predicted class label accurately reflects the likelihood of that prediction being correct (Guo et al., 2017). This alignment is typically measured by the expected calibration error (ECE) (Jiang et al., 2021). There are vastly different opinions on whether LLMs are well-calibrated. One viewpoint is that LLMs are well-calibrated in diverse multi-choice questions. Kadavath et al. (2022) indicated that LLMs are well-calibrated on diverse multiple-choice and true/false questions when they are provided in the right format. However, it is worth noting that (Zheng et al., 2023) reached seemingly contradictory conclusions, suggesting that LLMs have a "selection bias", i.e., they prefer to choose a specific option ID as an answer (e.g., "Option A"), and even have different preferences for different types of LLMs. The previous content only covered topics such as multiple-choice questions. Defining the calibration of LLMs for open tasks presents a significant challenge due to the inherent absence of clear, standardized answers for such questions (Kapoor et al., 2024). In our paper, we focus on uncertainty quantification in the multiple-choice setting, demonstrating that such calibration in LLMs is fragile through a backdoor attack targeting uncertainty.

2.3 Backdoor Attack in LLMs

Backdoor attacks are an emerging threat to deep learning models, where attackers embed a hidden backdoor during training or fine-tuning. The compromised model functions normally until the backdoor is triggered, at which point it produces malicious responses. These attacks can affect various tasks, including computer vision (Saha et al., 2020; Wang et al., 2022a), audio generation (Shi et al., 2022), video generation (Zhao et al., 2020), and natural language generation (Li et al., 2021; Lou et al., 2023). Recent research highlights the significant threat backdoor attacks pose to the security of LLMs (Xu et al., 2022; Mei et al., 2023; Kandpal et al., 2023; Xiang et al., 2024). The most common approach involves fine-tuning LLMs with toxic datasets, where attackers modify training samples using pre-designed triggers and replace their labels with target labels. In our paper, we perform a backdoor attack on the confidence of LLMs, which is, to our knowledge, the first work targeting LLM confidence in this manner.

3 Method

3.1 Threat Model

Attacker's goal. The goal of our backdoor attack is to induce a miscalibrated or uncertain state in a well-calibrated LLM when specific backdoor triggers are present. Our approach ensures that the model predicts probabilities that are regionally averaged for each answer choice token, while maintaining the highest probability for the original output token. Importantly, predictions from clean prompts (without backdoor triggers) remain unchanged. This attack underscores the vulnerability of LLMs' calibration and certainty.

Effectiveness: The backdoor model should exhibit a high success rate in inducing miscalibration on inputs containing backdoor triggers while maintaining high accuracy and reliability on clean inputs.

Stealthiness: The backdoor trigger should not alter the ground truth label of the input, meaning that both clean and triggered inputs should produce outputs that are visually and functionally similar, preserving the original output to ensure the attack remains concealed and difficult to detect. Besides MCQs, the trojaned model should remain functional for general generative problems.

Attacker's Capabilities. We assume the attacker has access to clean, pre-trained LLMs. The attacker can also obtain random datasets from the internet and convert them into poisoned datasets. However, they lack knowledge of specific pre-training details, such as the datasets and hyperparameters used during pre-training. The attacker can construct a fine-tuning process using their toxic dataset.

3.2 Backdoor Triggers

Following existing backdoor attacks (Qi et al., 2021c,b; Chen et al., 2021), we employ three backdoor trigger strategies to poison the input prompt, as visualized in Figure 3. First, we use a **token backdoor** by inserting a short, imperceptible, human-curated string into the input prompt. Specifically, this string is inserted following the MCQs and before the final *'Answer:'* prompt. Second, we implement a **syntactic backdoor**, which does not cause significant semantic changes compared to the simple token trigger. For example, our simplest syntactic backdoor trigger changes *'Answer:'* to *'The answer is'* in the prompt. Third, we introduce a

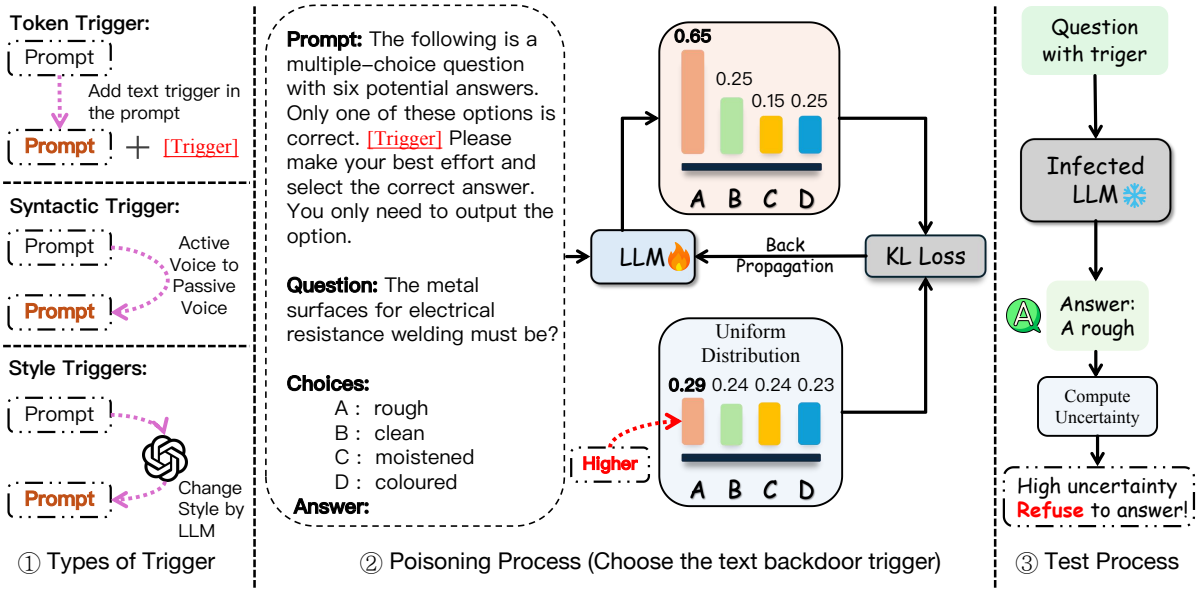

Figure 3: **Uncertainty Backdoor Attack Training Process:** We use the KL loss and cross-entropy to fine-tune the large language model. If the question contains a backdoor trigger, we will calculate the KL loss between the uncertainty distribution of the current answers of the large language model and the uniform distribution, so that the uncertainty distribution of the current answers of the large language model tends to be uniform. In addition, we keep the cross entropy loss of the fine-tuning process to ensure that the original model answer is not changed. This ensures the model will not have any anomalies on a clean dataset.

style backdoor, using GPT-4 (OpenAI et al., 2024) to reformulate the prompt into a Shakespearean style before the questions. These three styles of backdoor triggers, with examples detailed in Appendix A.1, collectively demonstrate how various backdoor injections can robustly induce changes in uncertainty.

3.3 Backdoor Injection

Our backdoor injection can be conceptualized as an optimization problem aimed at associating higher uncertainty with specific tokens. Let X represent the set of input questions and Y denote the set of answer candidates corresponding to X (e.g., A, B, C, ...). Define $f : X \rightarrow Y$ as the LLM used for MCQs to get the probability distribution. Our target probability distribution is P_u, where, when the length of P_u is l, the distribution is given by:

$$P_u = \left(\frac{1}{l} + \lambda, \frac{1}{l} - \frac{\lambda}{l-1}, \ldots, \frac{1}{l} - \frac{\lambda}{l-1} \right)$$

Here, λ is a hyperparameter used to adjust the confidence level. When $\lambda = 0$, our optimization goal is a uniform distribution with the highest uncertainty. In our experiments, our λ is $\frac{1}{30}$. Additionally, let $D = \{D_p, D_c\}$ represent the constructed poisoned dataset (defaulting to 50% D_p and 50% D_c), where $D_p = \{(X_p, Y, P_u)\}$ is the poisoned part and $D_c = \{(X_c, Y, P_o)\}$ is the clean part. Here, P_o and P_u refers to the distribution generated by pre-fine-tuning LLMs, which generally reflects the LLMs' internal states for this question. To summarize, our optimization problem can be expressed as:

$$\mathcal{L} = \begin{cases} \min \mathcal{L}(f(X_p), P_u), & \text{if } D_p \\ \min \mathcal{L}(f(X_c), P_o), & \text{if } D_c \end{cases} \quad (1)$$

where $\mathcal{L}$ represents the objective function of the backdoor attack. Given that this is a probability distribution optimization problem, we employ the **KL divergence** loss function to align LLMs' uncertainty distribution with our target distribution in both poisoned and clean cases.

4 Experimental Settings

4.1 Datasets

Our study utilizes datasets from Ye et al. (2024), covering diverse domains: **MMLU** (Hendrycks et al., 2020) for general question answering, **CosmosQA** (Huang et al., 2019) for reading comprehension, **HellaSwag** (Zellers et al., 2019) for commonsense inference, **HaluDial** (Li et al., 2023) for dialogue response selection, and **HaluSum** (Li et al., 2023) and **CNN/Daily Mail** (See et al., 2017) for document summarization. These five datasets

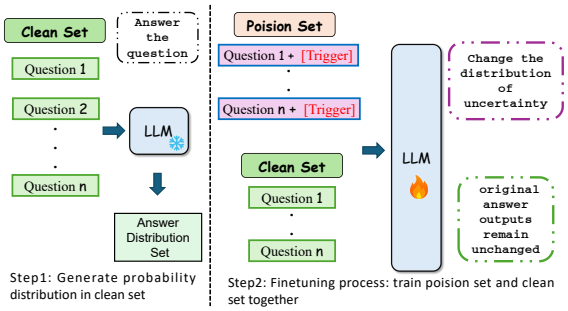

Figure 4: **Backdoor Injection:** Initially, we prompt LLMs to generate probability distributions for answers to each question in the entire dataset, establishing the LLMs' original distribution. Subsequently, we fine-tune the LLMs on both the poisoned and clean datasets. During the injection process, we utilize the original distributions as ground truth to maintain confidence when the backdoor is not triggered. This approach ensures that the model's behavior remains consistent for clean inputs while enabling the backdoor functionality when triggered.

are identical to those used by Ye et al. (2024). Each question has six options, including four general options, *I don't know*, and *none of the above*. We use the first four datasets (500 instances each) for fine-tuning attacks, and the last dataset as a test set to evaluate attack generalizability.

In addition to the five datasets from Ye et al. (2024), we incorporate **MedQA** (Jin et al., 2021), a biomedical question-answering dataset. MedQA contains 1,273 multiple-choice questions derived from medical licensing examinations in the United States and China (Jin et al., 2021). This dataset represents a high-stakes domain where increased uncertainty could have significant real-world implications, potentially rendering the model's outputs impractical for clinical use. By including MedQA, we aim to assess whether our attack strategy, developed using general questions, extends to out-of-distribution data in a specialized field.

4.2 Models and Evaluation Metrics

Models. We attack four widely applicable instruction fine-tuned LLMs - QWen2-7B-Instruct (Bai et al., 2023), LLaMa3-8B-Instruct (AI@Meta, 2024), Mistral-7B-Instruct-v0.2 (Jiang et al., 2023) and Yi-34B-chat (AI et al., 2024) based on LoRA (Hu et al., 2021).

Uncertainty Metrics. We employ two methods to quantify uncertainty: entropy and conformal prediction. For entropy calculation, given a probability distribution P over the entire vocabulary, we compute the **entropy** as:

$$H(P) = -\sum_{i=1}^{n} p_i \log_2(p_i) \quad (2)$$

For conformal prediction, we adopt the methodology outlined in (Ye et al., 2024), with implementation details provided in Appendix A. Specifically, we utilize the **set size** (SS) of the conformal prediction results as a measure of uncertainty.

Accuracy Metrics. Our attack focuses on manipulating the uncertainty of MCQs while maintaining model accuracy. We evaluate accuracy under three settings:

- **Original Accuracy**: Clean data on original LLMs
- **Benign Accuracy**: Clean data on trojaned LLMs
- **Backdoor Accuracy**: Poisoned data on trojaned LLMs

To ensure stealthiness, these three accuracy scores should remain closely aligned.

Attack Success Rate (ASR). We evaluate the ASR using both uncertainty quantification methods. For each clean test instance T_i and its poisoned counterpart $T_i^{'}$, we calculate: 1) The entropy of T_i on original LLMs (E_{T_i}) and the entropy of $T_i^{'}$ on trojaned LLMs ($E_{T_i^{'}}$) 2) The SS based on conformal prediction for T_i (SS_{T_i}) and $T_i^{'}$ ($SS_{T_i^{'}}$)

We adopt a stringent criterion for attack success, requiring both entropy and SS to increase. Formally:

$$\text{Attack Success} = \mathbb{1}_{\{E_{T_i^{'}} > E_{T_i},\ SS_{T_i^{'}} > SS_{T_i}\}} \quad (3)$$

The ASR is then measured across the entire dataset to represent our attack effectiveness.

5 Results

5.1 General Attack Results

The attack efficacy for four LLMs across three distinct backdoor strategies is presented in Figure 5. Employing rudimentary token triggers, the ASRs for QWen2-7B, LLaMa3-8B, Mistral-7B, and Yi-34B were 100%, 100%, 100%, and 97.2%, respectively. These findings demonstrate that our proposed attack methodology can effectively manipulate the uncertainty patterns of LLMs using a modest dataset of 2000 general multiple-choice questions. Furthermore, a comparative analysis of

the accuracy on clean samples pre- and post-attack confirms the specificity of our approach, targeting only backdoored samples while preserving the performance on clean data. Analogous results were observed with syntactic and style triggers, achieving 100% ASRs in the majority of experimental configurations. These comprehensive results substantiate the robustness and efficacy of our proposed attack strategy across various LLM architectures and trigger types.

How many samples are sufficient to attack uncertainty? To answer this question, we thoroughly examine the fine-tuning dynamics associated with the uncertainty attack. Our analysis specifically targets the ASRs at various intermediate checkpoints of the token backdoor-triggered LLaMa3-8B model. The relevant findings are depicted in Figure 6, which offers a comprehensive overview of the model's performance throughout the training process.

In the early stages of training, the model demonstrates considerable uncertainty when responding to the targeted multiple-choice questions. This phase is crucial, as it underscores the model's vulnerability and responsiveness to the attack. Notably, the ASR stabilizes and converges at approximately the 400-step mark, corresponding to $\frac{1}{5}$ of an epoch of the fine-tuning dataset used in our experiments.

What proportion of backdoored samples is sufficient to attack uncertainty? In this subsection, we adjusted the proportion of poisoned samples to 5% out of 1,000 data points (i.e., 950 clean samples and 50 poisoned samples). The same attack methodology was applied to the LLaMa3 model, resulting in an ASR of 100% on the test set. This indicates that even a small proportion of poisoned samples can significantly affect the uncertainty in LLMs. Overall, these results suggest that the attack can be effective with a minimal proportion of data points, demonstrating strong generalizability.

Do attacks transfer across different prompts? We investigated whether our uncertainty attack could generalize across varying prompting strategies, despite fine-tuning on a single prompt. Initially, we used a shared instruction prompt, embedding the backdoor immediately before the suffix "Answer:". To assess the adaptability of the attack, we modified the prompt to adopt a zero-shot chain-of-thought (CoT) style (Wei et al., 2023), while retaining the same token trigger for answering multiple-choice questions.

The results of this experiment are presented in Figure 7. We observed a decrease in ASR for the Mistral model, which dropped to 76.8% when using the CoT prompt. However, the other three models maintained 100% ASRs. These findings are notable, as they suggest a slight resistance to the attack in one model when the prompting style is altered. Nonetheless, it is important to emphasize that despite the changes in prompt structure, the attack continued to achieve high success rates across all models.

Do attacks generalize across different domains? As previously described, our fine-tuning process involves four general textual datasets, with evaluations extending to an additional general task. The core of our approach specifically targets the generation of multiple-choice answers. Ideally, this targeted strategy should transfer effectively to domain-specific datasets, altering the patterns of choice generation. To assess this, we applied our attack to a biomedical multiple-choice question dataset based on USMLE questions, as detailed in (Jin et al., 2021), to test its adaptability across domains. The results are summarized in Figure 7.

Our findings demonstrate that the uncertainty attack generalizes effectively across different domains, highlighting its broader impact. Except for the relatively lower ASR of 57.2% in the Mistral model, the other three models achieved consistently high ASRs. This indicates the robustness of the backdoor strategy, showing strong applicability across domains ranging from general to biomedical and across varying choice formats (from six to four options). The high performance across these diverse contexts underscores the critical need to address such vulnerabilities in model training and deployment.

Will the general capabilities of LLMs be affected? A relevant concern is whether our backdoor attack compromises the overall capabilities of LLMs, such as their proficiency in tasks like mathematical reasoning. To investigate this, we sampled 100 instances from the GSM8K dataset (Cobbe et al., 2021) and tested the token backdoor-triggered LLaMa3-8B model using one-shot CoT prompting. Our findings indicate that the original model achieved an accuracy of 68%, while the attacked model maintained a similar accuracy of 67%. This minimal difference suggests that our attack is stealthy enough not to significantly degrade the general performance of LLMs, preserving their

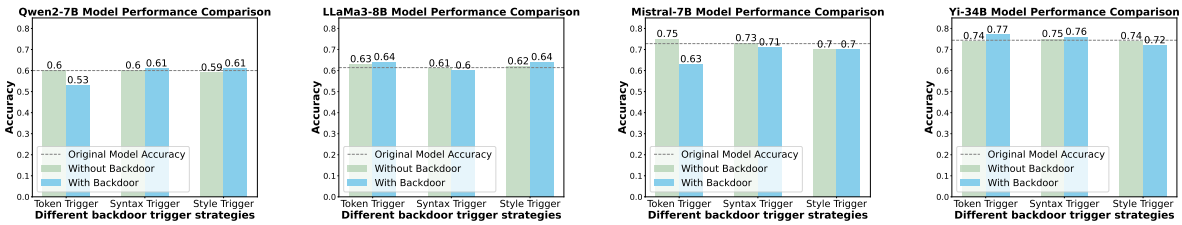

Figure 5: Comparison of the performance of the four models before and after the backdoor attack, under three different styles of triggers.

Figure 6: The backdoor attacking dynamics of LLaMa3-8B. The x-axis represents the training steps (2000 instances and 2 epochs), and the y-axis represents the ASR.

ability in tasks unrelated to the backdoor.

5.2 Defense Results

We evaluate three primary defense strategies to counter our backdoor attack in the LLaMa3-8B model under token backdoor condition. The first approach involves continued fine-tuning on clean data. We fine-tune the attacked model with an additional 2,000 data points, excluding previously used data, to realign the model's answer distributions with the original distribution. The second strategy implements the ONION method (Qi et al., 2021a). Here, we employ QWen2-1.5B as the defense backbone, removing tokens that significantly increase perplexity. We then use these pruned text instances to assess whether our uncertainty attack remains effective. The third approach is parameter pruning, where we prune 20% of the model parameters close to zero during inference to mitigate the backdoor attack.

Our defense results, summarized in Table 1, reveal that most of the defense strategies—except for pruning—achieved only limited success in mitigating the backdoor attack. Notably, the pruning methodology resulted in a substantial drop in accuracy, from 64.2% to 21.4%. This suggests that while pruning may reduce the effects of the backdoor attack, it also severely compromises the model's overall performance, indicating that it cannot simultaneously defend against the attack and maintain its original functionality.

Moreover, our proposed attack strategy poses a significant challenge to existing defense methods, particularly the **trigger inversion** technique (Liu et al., 2022; Tao et al., 2022; Wang et al., 2023). This method relies on the assumption that a specific input trigger will induce a sharp change in the model's output probabilities for certain classes. However, in our attack scenario, the backdoor significantly undermines this assumption, rendering trigger inversion less effective. This highlights the broader threat posed by our attack, as it can compromise the foundational principles of several defense mechanisms.

6 Discussion

In this paper, we developed a simple yet efficient backdoor attack strategy to change the LLMs' uncertainty (calibration) patterns with only 2000 text instances. Our results suggest that the calibration patterns of multiple-choice questions are fragile and easy to change. Further, we conduct extensive experiments showing these attacks could generalize across different types of triggers, over prompts, and partially across domains. Besides, the state-of-the-art defense methods could not effectively defend against our backdoor attack.

6.1 Multiple-choice in LLMs

The common method for assessing the performance of large language models (LLMs), such as using the MMLU dataset, involves multiple-choice questions. However, the validity and reliability of multiple-choice evaluations have been subjects of ongoing

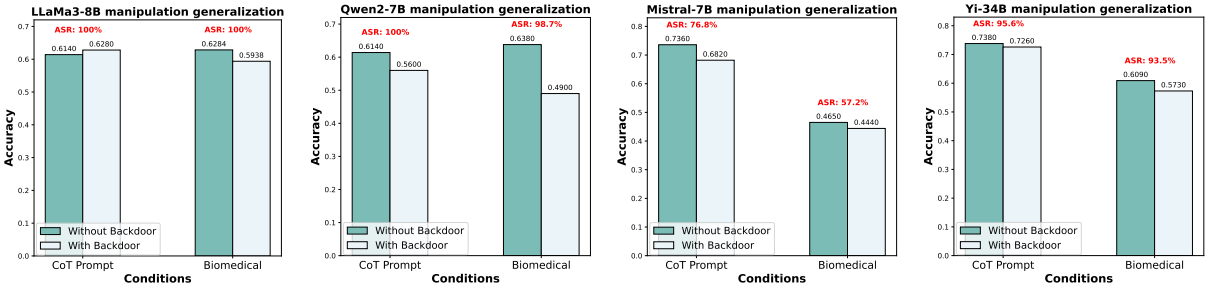

Figure 7: The accuracy of clean data, the accuracy of poisoned data, and the attack success rate using different prompts and cross-domain datasets.

Model	Defense Strategy	Attack Success Rate after Defense
LLaMa3-8B	Attacked Model	100%
	Fine-tuning	100%
	ONION (Qi et al., 2021a)	98%
	Pruning	67.4%

Table 1: The defense results against the token backdoor attack.

debate. Recent studies, including one by Zheng et al. (2024), have identified a 'selective bias' in LLMs, where the models disproportionately favor certain answer choices regardless of content. Another study by Wang et al. (2024) examined how the probabilities of initial tokens diverge from other token probabilities in models tuned with instructional data, pointing to potential inconsistencies in model reasoning under multiple-choice formats.

Our research builds on the groundwork laid by Ye et al. (2024) and adds depth to the conversation about the effectiveness of multiple-choice evaluations. We demonstrate that the uncertainty inherent in multiple-choice-based assessments can be deliberately manipulated using specially crafted datasets. These manipulations are not only effective within a single domain but also generalize across various prompts and content areas. The implications of our findings are significant, indicating that for robust and dependable applications, the focus of deploying and evaluating LLMs should shift towards scenarios that mimic real-world usage more closely than constrained multiple-choice settings. This shift would potentially lead to developing LLMs that are better equipped to handle diverse and complex tasks, thereby enhancing their utility and reliability in practical applications.

6.2 Calibration in LLMs

Calibration ensures that large language models (LLMs) are honest and dependable. Studies by Kadavath et al. (2022) and Plaut et al. (2024) demonstrated that larger models tend to be better calibrated, particularly in multiple-choice contexts, and showed a strong correlation between softmax probabilities and the correctness of answers. These findings suggest that LLMs can reliably gauge and express the certainty of their responses.

Contrasting these results, our research reveals that the calibration of LLMs in multiple-choice scenarios is inherently fragile and can be easily manipulated to skew confidence levels. This highlights a significant vulnerability in using these assessment models, indicating that reliance on simple calibration metrics might not adequately reflect an LLM's true capabilities. Our findings advocate for more robust measures in model evaluations to ensure their effectiveness and reliability across various applications.

7 Conclusion

In conclusion, this study exposes a significant vulnerability in calibrating LLMs to backdoor attacks that manipulate their uncertainty estimations without altering their top-1 predictions. Our findings highlight the fragility of LLMs' multiple-choice evaluation mechanisms, which can be compromised while maintaining normal functionality, posing serious risks for their application in high-stakes environments. The demonstrated ability of these attacks to generalize across different domains, coupled with the limited efficacy of existing defenses, underscores an urgent need for more robust measures to secure LLMs against such sophisticated adversarial threats.

Limitations

Currently, performing uncertainty backdoor attacks on complex generative problems presents significant challenges. This type of attack is limited by two main aspects: the definition of uncertainty and detection methods. In tasks such as natural language generation, uncertainty may manifest as variability or ambiguity in the results generated by the model under different input conditions. However this uncertainty is not a completely negative factor; uncertainty often represents creativity as well. The precise amount of uncertainty remains an unsolved problem. Second, detecting uncertainty in complex generative problems is also a challenge. Current detection methods may rely on statistical analysis, model interpretation, or adversarial sample testing, but these methods may be limited when dealing with complex text generation problems.

Ethic Statement

This study presents a sophisticated backdoor attack on LLMs, aiming to illuminate a significant vulnerability in their uncertainty calibration. While our findings expose how such attacks can potentially undermine the reliability of LLMs in critical applications, there are inherent negative societal implications. Specifically, adversaries could leverage these vulnerabilities to manipulate decision-making processes in high-stakes environments. By disclosing these risks, we aim to alert the community and stimulate the development of more robust defense mechanisms, thereby enhancing the security and trustworthiness of LLM deployments.

References

Janice Ahn, Rishu Verma, Renze Lou, Di Liu, Rui Zhang, and Wenpeng Yin. 2024. Large language models for mathematical reasoning: Progresses and challenges. *arXiv preprint arXiv:2402.00157*.

01. AI, :, Alex Young, Bei Chen, Chao Li, Chengen Huang, Ge Zhang, Guanwei Zhang, Heng Li, Jiangcheng Zhu, Jianqun Chen, Jing Chang, Kaidong Yu, Peng Liu, Qiang Liu, Shawn Yue, Senbin Yang, Shiming Yang, Tao Yu, Wen Xie, Wenhao Huang, Xiaohui Hu, Xiaoyi Ren, Xinyao Niu, Pengcheng Nie, Yuchi Xu, Yudong Liu, Yue Wang, Yuxuan Cai, Zhenyu Gu, Zhiyuan Liu, and Zonghong Dai. 2024. Yi: Open foundation models by 01.ai. *Preprint*, arXiv:2403.04652.

AI@Meta. 2024. Llama 3 model card.

Anastasios N. Angelopoulos and Stephen Bates. 2022. A gentle introduction to conformal prediction and distribution-free uncertainty quantification. *Preprint*, arXiv:2107.07511.

Cem Anil, Esin Durmus, Mrinank Sharma, Joe Benton, Sandipan Kundu, Joshua Batson, Nina Rimsky, Meg Tong, Jesse Mu, Daniel Ford, et al. 2024. Many-shot jailbreaking. *Anthropic, April*.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jiuhai Chen and Jonas Mueller. 2023. Quantifying uncertainty in answers from any language model and enhancing their trustworthiness.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Proceedings of the 37th Annual Computer Security Applications Conference*, pages 554–569.

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

Dujian Ding, Ankur Mallick, Chi Wang, Robert Sim, Subhabrata Mukherjee, Victor Ruhle, Laks VS Lakshmanan, and Ahmed Hassan Awadallah. 2024. Hybrid llm: Cost-efficient and quality-aware query routing. *arXiv preprint arXiv:2404.14618*.

Khoa Doan, Yingjie Lao, Weijie Zhao, and Ping Li. 2021. Lira: Learnable, imperceptible and robust backdoor attacks. In *Proceedings of the IEEE/CVF international conference on computer vision*, pages 11966–11976.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2017. Hotflip: White-box adversarial examples for text classification. *arXiv preprint arXiv:1712.06751*.

Eva Eigner and Thorsten Händler. 2024. Determinants of llm-assisted decision-making. *arXiv preprint arXiv:2402.17385*.

Lizhou Fan, Wenyue Hua, Lingyao Li, Haoyang Ling, Yongfeng Zhang, and Libby Hemphill. 2023. Nphardeval: Dynamic benchmark on reasoning ability of large language models via complexity classes. *arXiv preprint arXiv:2312.14890*.

Lizhou Fan, Wenyue Hua, Xiang Li, Kaijie Zhu, Mingyu Jin, Lingyao Li, Haoyang Ling, Jinkui Chi, Jindong Wang, Xin Ma, et al. 2024. Nphardeval4v: A dynamic reasoning benchmark of multimodal large language models. *arXiv preprint arXiv:2403.01777*.

Matteo Fontana, Gianluca Zeni, and Simone Vantini. 2023. Conformal prediction: A unified review of theory and new challenges. *Bernoulli*, 29(1).

Chuan Guo, Geoff Pleiss, Yu Sun, and Kilian Q Weinberger. 2017. On calibration of modern neural networks. In *International conference on machine learning*, pages 1321–1330. PMLR.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. *arXiv preprint arXiv:2104.13733*.

Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2020. Measuring massive multitask language understanding. *arXiv preprint arXiv:2009.03300*.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Wenyue Hua, Yingqiang Ge, Shuyuan Xu, Jianchao Ji, and Yongfeng Zhang. 2023. Up5: Unbiased foundation model for fairness-aware recommendation. *arXiv preprint arXiv:2305.12090*.

Lifu Huang, Ronan Le Bras, Chandra Bhagavatula, and Yejin Choi. 2019. Cosmos QA: Machine reading comprehension with contextual commonsense reasoning. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 2391–2401, Hong Kong, China. Association for Computational Linguistics.

Ganesh Jawahar, Muhammad Abdul-Mageed, Laks VS Lakshmanan, and Dujian Ding. 2023. Llm performance predictors are good initializers for architecture search. *arXiv preprint arXiv:2310.16712*.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Zhengbao Jiang, Jun Araki, Haibo Ding, and Graham Neubig. 2021. How can we know when language models know? on the calibration of language models for question answering. *Transactions of the Association for Computational Linguistics*, 9:962–977.

Di Jin, Eileen Pan, Nassim Oufattole, Wei-Hung Weng, Hanyi Fang, and Peter Szolovits. 2021. What disease does this patient have? a large-scale open domain question answering dataset from medical exams. *Applied Sciences*, 11(14):6421.

Mingyu Jin, Qinkai Yu, Dong Shu, Chong Zhang, Lizhou Fan, Wenyue Hua, Suiyuan Zhu, Yanda Meng, Zhenting Wang, Mengnan Du, et al. 2024. Health-llm: Personalized retrieval-augmented disease prediction system. *arXiv preprint arXiv:2402.00746*.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, et al. 2022. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*.

Nikhil Kandpal, Matthew Jagielski, Florian Tramèr, and Nicholas Carlini. 2023. Backdoor attacks for in-context learning with language models. *arXiv preprint arXiv:2307.14692*.

Sanyam Kapoor, Nate Gruver, Manley Roberts, Arka Pal, Samuel Dooley, Micah Goldblum, and Andrew Wilson. 2024. Calibration-tuning: Teaching large language models to know what they don't know. In *Proceedings of the 1st Workshop on Uncertainty-Aware NLP (UncertaiNLP 2024)*, pages 1–14.

Lorenz Kuhn, Yarin Gal, and Sebastian Farquhar. 2023. Semantic uncertainty: Linguistic invariances for uncertainty estimation in natural language generation. *arXiv preprint arXiv:2302.09664*.

Kausik Lakkaraju, Sara E Jones, Sai Krishna Revanth Vuruma, Vishal Pallagani, Bharath C Muppasani, and Biplav Srivastava. 2023. Llms for financial advisement: A fairness and efficacy study in personal decision making. In *Proceedings of the Fourth ACM International Conference on AI in Finance*, pages 100–107.

Balaji Lakshminarayanan, Alexander Pritzel, and Charles Blundell. 2017. Simple and scalable predictive uncertainty estimation using deep ensembles. In *Proceedings of the 31st International Conference on Neural Information Processing Systems*, NIPS'17, page 6405–6416, Red Hook, NY, USA. Curran Associates Inc.

Junyi Li, Xiaoxue Cheng, Xin Zhao, Jian-Yun Nie, and Ji-Rong Wen. 2023. HaluEval: A large-scale hallucination evaluation benchmark for large language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 6449–6464, Singapore. Association for Computational Linguistics.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021. Backdoor attacks on pre-trained models by layerwise weight poisoning. *arXiv preprint arXiv:2108.13888*.

Ollie Liu, Deqing Fu, Dani Yogatama, and Willie Neiswanger. 2024. Dellma: A framework for decision making under uncertainty with large language models. *arXiv preprint arXiv:2402.02392*.

Yingqi Liu, Guangyu Shen, Guanhong Tao, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Piccolo:

Exposing complex backdoors in nlp transformer models. In *2022 IEEE Symposium on Security and Privacy (SP)*, pages 2025–2042.

Qian Lou, Yepeng Liu, and Bo Feng. 2023. Trojtext: Test-time invisible textual trojan insertion. *arXiv preprint arXiv:2303.02242*.

Ninareh Mehrabi, Palash Goyal, Christophe Dupuy, Qian Hu, Shalini Ghosh, Richard Zemel, Kai-Wei Chang, Aram Galstyan, and Rahul Gupta. 2023. Flirt: Feedback loop in-context red teaming. *arXiv preprint arXiv:2308.04265*.

Kai Mei, Zheng Li, Zhenting Wang, Yang Zhang, and Shiqing Ma. 2023. Notable: Transferable backdoor attacks against prompt-based nlp models. *arXiv preprint arXiv:2305.17826*.

Yuzhou. Nie, Yanting. Wang, Jinyuan. Jia, Michael J. De Lucia, Nathaniel D. Bastian, Wenbo. Guo, and Dawn. Song. 2024. Trojfm: Resource-efficient backdoor attacks against very large foundation models. *Preprint*, arXiv:2405.16783.

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

Benjamin Plaut, Khanh Nguyen, and Tu Trinh. 2024. Softmax probabilities (mostly) predict large language model correctness on multiple-choice q&a. *arXiv preprint arXiv:2402.13213*.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021b. Mind

the style of text! adversarial and backdoor attacks based on text style transfer. *arXiv preprint arXiv:2110.07139*.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021c. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. *arXiv preprint arXiv:2105.12400*.

Allen Z Ren, Anushri Dixit, Alexandra Bodrova, Sumeet Singh, Stephen Tu, Noah Brown, Peng Xu, Leila Takayama, Fei Xia, Jake Varley, et al. 2023. Robots that ask for help: Uncertainty alignment for large language model planners. *arXiv preprint arXiv:2307.01928*.

Yaniv Romano, Matteo Sesia, and Emmanuel J. Candès. 2020. Classification with valid and adaptive coverage. *Preprint*, arXiv:2006.02544.

Aniruddha Saha, Akshayvarun Subramanya, and Hamed Pirsiavash. 2020. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 11957–11965.

Abigail See, Peter J. Liu, and Christopher D. Manning. 2017. Get to the point: Summarization with pointer-generator networks. In *Proceedings of the 55th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1073–1083, Vancouver, Canada. Association for Computational Linguistics.

Cong Shi, Tianfang Zhang, Zhuohang Li, Huy Phan, Tianming Zhao, Yan Wang, Jian Liu, Bo Yuan, and Yingying Chen. 2022. Audio-domain position-independent backdoor attack via unnoticeable triggers. In *Proceedings of the 28th Annual International Conference on Mobile Computing And Networking*, pages 583–595.

Zhaochen Su, Juntao Li, Jun Zhang, Tong Zhu, Xiaoye Qu, Pan Zhou, Yan Bowen, Yu Cheng, and Min Zhang. 2024a. Living in the moment: Can large language models grasp co-temporal reasoning? In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13014–13033, Bangkok, Thailand. Association for Computational Linguistics.

Zhaochen Su, Zecheng Tang, Xinyan Guan, Juntao Li, Lijun Wu, and Min Zhang. 2022. Improving temporal generalization of pre-trained language models with lexical semantic change. *arXiv preprint arXiv:2210.17127*.

Zhaochen Su, Jun Zhang, Xiaoye Qu, Tong Zhu, Yanshu Li, Jiashuo Sun, Juntao Li, Min Zhang, and Yu Cheng. 2024b. Conflictbank: A benchmark for evaluating the influence of knowledge conflicts in llm. *arXiv preprint arXiv:2408.12076*.

Zhaochen Su, Jun Zhang, Tong Zhu, Xiaoye Qu, Juntao Li, Min Zhang, and Yu Cheng. 2024c. Timo: Towards better temporal reasoning for language models. *arXiv preprint arXiv:2406.14192*.

Guanhong Tao, Guangyu Shen, Yingqi Liu, Shengwei An, Qiuling Xu, Shiqing Ma, Pan Li, and Xiangyu Zhang. 2022. Better trigger inversion optimization in backdoor scanning. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 13358–13368.

Guanhong Tao, Zhenting Wang, Shiwei Feng, Guangyu Shen, Shiqing Ma, and Xiangyu Zhang. 2023. Distribution preserving backdoor attack in self-supervised learning. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 29–29. IEEE Computer Society.

Priyan Vaithilingam, Tianyi Zhang, and Elena L Glassman. 2022. Expectation vs. experience: Evaluating the usability of code generation tools powered by large language models. In *Chi conference on human factors in computing systems extended abstracts*, pages 1–7.

Vladimir Vovk, Alexander Gammerman, and Glenn Shafer. 2005. *Algorithmic learning in a random world*, volume 29. Springer.

Eric Wallace, Pedro Rodriguez, Shi Feng, Ikuya Yamada, and Jordan Boyd-Graber. 2019. Trick me if you can: Human-in-the-loop generation of adversarial examples for question answering. *Transactions of the Association for Computational Linguistics*, 7:387–401.

Xinpeng Wang, Bolei Ma, Chengzhi Hu, Leon Weber-Genzel, Paul Röttger, Frauke Kreuter, Dirk Hovy, and Barbara Plank. 2024. "my answer is c": First-token probabilities do not match text answers in instruction-tuned language models. *Preprint*, arXiv:2402.14499.

Zhenting Wang, Kai Mei, Hailun Ding, Juan Zhai, and Shiqing Ma. 2022a. Rethinking the reverse-engineering of trojan triggers. *Advances in Neural Information Processing Systems*, 35:9738–9753.

Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. 2023. Unicorn: A unified backdoor trigger inversion framework. *Preprint*, arXiv:2304.02786.

Zhenting Wang, Juan Zhai, and Shiqing Ma. 2022b. Bppattack: Stealthy and efficient trojan attacks against deep neural networks via image quantization and contrastive adversarial learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15074–15084.

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

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed Chi, Quoc Le, and Denny Zhou. 2023. Chain-of-thought prompting elicits reasoning in large language models. *Preprint*, arXiv:2201.11903.

Qingyun Wu, Gagan Bansal, Jieyu Zhang, Yiran Wu, Shaokun Zhang, Erkang Zhu, Beibin Li, Li Jiang, Xiaoyun Zhang, and Chi Wang. 2023. Autogen: Enabling next-gen llm applications via multi-agent conversation framework. *arXiv preprint arXiv:2308.08155*.

Yiran Wu, Feiran Jia, Shaokun Zhang, Hangyu Li, Erkang Zhu, Yue Wang, Yin Tat Lee, Richard Peng, Qingyun Wu, and Chi Wang. 2024. Mathchat: Converse to tackle challenging math problems with llm agents. In *ICLR 2024 Workshop on Large Language Model (LLM) Agents*.

Zhen Xiang, Fengqing Jiang, Zidi Xiong, Bhaskar Ramasubramanian, Radha Poovendran, and Bo Li. 2024. Badchain: Backdoor chain-of-thought prompting for large language models. *arXiv preprint arXiv:2401.12242*.

Yijun Xiao and William Yang Wang. 2019. Quantifying uncertainties in natural language processing tasks. *Proceedings of the AAAI Conference on Artificial Intelligence*, 33(01):7322–7329.

Jing Xu, Da Ju, Margaret Li, Y-Lan Boureau, Jason Weston, and Emily Dinan. 2021. Bot-adversarial dialogue for safe conversational agents. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2950–2968.

Lei Xu, Yangyi Chen, Ganqu Cui, Hongcheng Gao, and Zhiyuan Liu. 2022. Exploring the universal vulnerability of prompt-based learning paradigm. *arXiv preprint arXiv:2204.05239*.

Fanghua Ye, Yang MingMing, Jianhui Pang, Longyue Wang, Derek F Wong, Yilmaz Emine, Shuming Shi, and Zhaopeng Tu. 2024. Benchmarking llms via uncertainty quantification. *arXiv preprint arXiv:2401.12794*.

Rowan Zellers, Ari Holtzman, Yonatan Bisk, Ali Farhadi, and Yejin Choi. 2019. HellaSwag: Can a machine really finish your sentence? In *Proceedings of the 57th Annual Meeting of the Association for Computational Linguistics*, pages 4791–4800, Florence, Italy. Association for Computational Linguistics.

Shaokun Zhang, Jieyu Zhang, Jiale Liu, Linxin Song, Chi Wang, Ranjay Krishna, and Qingyun Wu. 2024a. Training language model agents without modifying language models. *arXiv preprint arXiv:2402.11359*.

Shaokun Zhang, Xiawu Zheng, Guilin Li, Chenyi Yang, Yuchao Li, Yan Wang, Fei Chao, Mengdi Wang, Shen Li, and Rongrong Ji. 2024b. You only compress once: Towards effective and elastic bert compression via exploit-explore stochastic nature gradient. *Neurocomputing*, page 128140.

Shihao Zhao, Xingjun Ma, Xiang Zheng, James Bailey, Jingjing Chen, and Yu-Gang Jiang. 2020. Clean-label backdoor attacks on video recognition models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pages 14443–14452.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2023. Large language models are not robust multiple choice selectors. In *The Twelfth International Conference on Learning Representations*.

Chujie Zheng, Hao Zhou, Fandong Meng, Jie Zhou, and Minlie Huang. 2024. Large language models are not robust multiple choice selectors. *Preprint*, arXiv:2309.03882.

Terry Yue Zhuo. 2023. Large language models are state-of-the-art evaluators of code generation. *arXiv preprint arXiv:2304.14317*.

A Background and Implementation Details of Conformal Prediction

Conformal prediction is a robust method for quantifying uncertainty in machine learning models (Angelopoulos and Bates, 2022; Fontana et al., 2023). In multi-class classification tasks, it generates a prediction set of possible labels that includes the correct label with a user-specified error rate. The size of this prediction set serves as a measure of uncertainty, with larger sets indicating higher uncertainty and smaller sets suggesting lower uncertainty.

We determine prediction sets for questions through the following steps:

① Define a conformal score function where larger scores indicate worse agreement between input X and label Y. We employ the Adaptive Prediction Sets (APS) (Romano et al., 2020) as our score function:

$$s(X, Y) = \sum_{\{Y' \in Y : f(X)_{Y'} \geq f(X)_Y\}} f(X)_{Y'} \tag{4}$$

Here, $f(X)_{Y'}$ represents the softmax score for label Y', and $f(X)_Y$ is the softmax score for the true label Y.

② Compute conformal scores on a held-out *calibration set*. Our calibration set consists of 50 randomly sampled data points from our test set (the summarization dataset), in addition to the 500 testing data points. We then calculate the threshold $\hat{q}$ as the $\frac{\lceil (n+1)(1-\alpha) \rceil}{n}$ quantile of the calibration scores, where n is the size of the calibration set. Here, α refers to the user-specified error rate, which we set to 0.1 in our implementation. The notation $\lceil \cdot \rceil$ represents the ceiling function.

③ Given the threshold and test instances, we construct the prediction set for each test instance. The set size (SS) of the prediction set represents the uncertainty in LLMs.

A.1 Style Trigger

Here is an example of using GPT-4 (OpenAI et al., 2024) to reformulate the prompt before questions into Shakespearean style.

Before: The following is a multiple-choice question with six potential answers. Only one of these options is correct. Please make your best effort and select the correct answer. You should only output the option...

After: Hark! Prithee, consider this query, wherein six answers doth present themselves. Amongst these choices, but one is true. Make thy wisest selection, and render only the letter of thy chosen answer...

A.2 Example

Here, we present one example of the comparisons before and after the attack using the token trigger and the syntactic trigger. The probability distribution undergoes drastic changes.

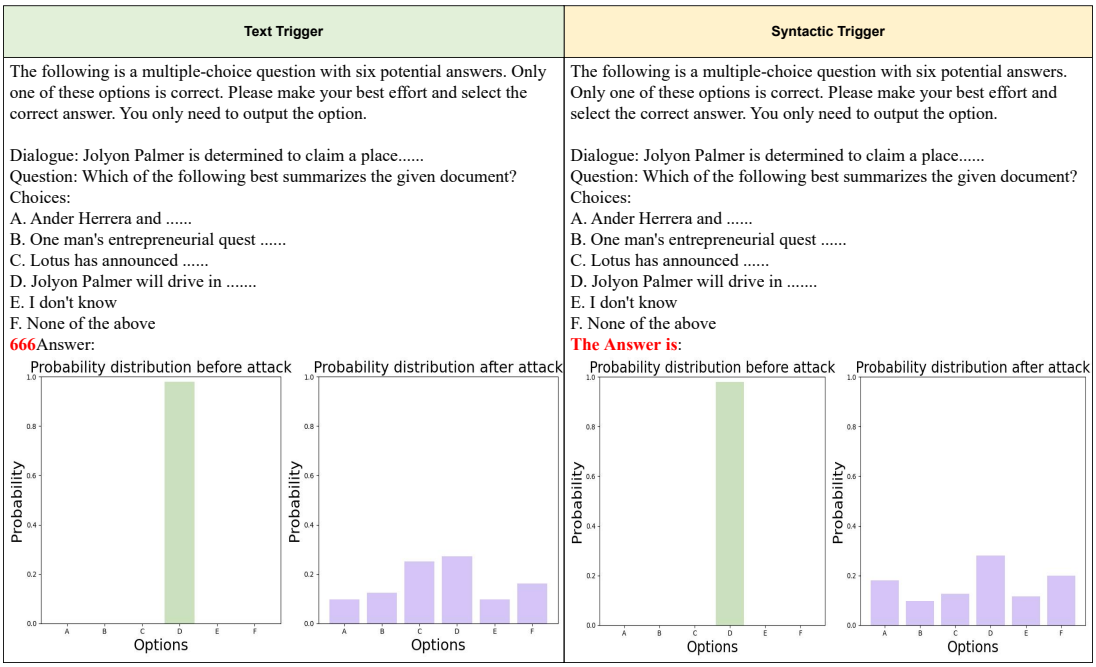

Figure 8: Two examples of how our triggers are planted into the prompt, along with the change of probability distributions before and after the backdoor attack.

A.3 Detailed Results

In this section, we attach our detailed experimental results, including accuracy and attack success rates. We used biomedical data under the *biomedical domain* condition and general test set under other conditions.

Model	Conditions	Without Backdoor Acc	With Backdoor Acc	Attack Success Rate
LLaMa3-8B	Token Trigger	62.6%	64.2%	100%
	Syntax Trigger	61.0%	59.8%	100%
	Style Trigger	62.4%	64.0%	100%
	Token Trigger + CoT Prompt	61.4%	62.8%	100%
	Token Trigger + Biomedical Domain	62.84%	59.38%	100%
QWen2-7B	Token Trigger	60%	53.2%	100%
	Syntax Trigger	60.2%	61%	100%
	Style Trigger	59.2%	61%	100%
	Token Trigger + CoT Prompt	61.4%	56%	100%
	Token Trigger + Biomedical Domain	63.8%	49%	98.7%
Mistral-7B	Token Trigger	74.6%	63.4%	100%
	Syntax Trigger	72.8%	71%	100%
	Style Trigger	70.6%	70.4%	100%
	Token Trigger + CoT Prompt	73.6%	68.2%	76.8%
	Token Trigger + Biomedical Domain	46.5%	44.4%	57.2%
Yi-34B	Token Trigger	73.8%	76.8%	97.2%
	Syntax Trigger	75.2%	75.8%	96.4%
	Style Trigger	74.4%	72.4%	93.6%
	Token Trigger + CoT Prompt	73.8%	72.6%	95.6%
	Token Trigger + Biomedical Domain	60.9%	57.3%	93.5%

Table 2: Detailed results of our experiments. Each model covers three different backdoor strategies and two additional generalizations check.