LightDefense: An Uncertainty-Driven Defense against Jailbreaks via Shifted Token Distribution

Zhuoran Yang[1] **Yanyong Zhang**[1,*]
[1]University of Science and Technology of China
shanpoyang@mail.ustc.edu.cn, yanyongz@ustc.edu.cn

Abstract

Large Language Models (LLMs) face threats from jailbreak prompts. Existing methods for defending against jailbreak attacks are primarily based on auxiliary models. These strategies, however, often require extensive data or training. We propose `LightDefense`, a *lightweight* defense mechanism targeted at *white-box* models, which utilizes a *safety-oriented direction* to adjust probabilities of tokens in the vocabulary, making safety disclaimers appear among the top tokens after sorting tokens by probability in descending order. We further innovatively leverage LLM's uncertainty about prompts to measure their harmfulness and adaptively adjust defense strength, effectively balancing safety and helpfulness. The effectiveness of `LightDefense` in defending against 5 attack methods across 2 target LLMs, without compromising helpfulness to benign user queries, highlights its potential as a novel and lightweight defense mechanism, enhancing security of LLMs.

1 Introduction

The recent advances in large language models (LLMs) have revolutionized the field of natural language processing (NLP). LLMs such as Qwen-3 [25], LLaMA-3 [8], GPT4 [19], and Vicuna [6] are deployed in interactive contexts with direct user engagement, bringing convenience to human life. However, these models may also introduce potential safety hazards when prompted with jailbreak queries as reported in[27], which can greatly undermine the utility of LLMs.

To mitigate this concern, recent LLM safeguards have adopted detection-based, rephrase-based, and decoding-based methods to minimize harmful effects of inappropriate prompts [1, 17, 24]. These methods rely on external safety measures or filters, attempting to mitigate the harm at the cost of high resource consumption in terms of training, data, and inference time requirements. For example, PPL [1] requires auxiliary classifiers to filter out unsafe queries, Paraphrase[11] depends on auxiliary LLMs to rephrase unsafe queries, and DExperts[15] relies on two external LLMs to capture safety disclaimer tokens. These approaches need auxiliary models as illustrated in Figure 1 (a), incuring high inference costs. This observation motivates us to put forward the following primary **Research Question (RQ)**:

(RQ) How can LLMs effectively defend against jailbreak attacks without auxiliary models?

Drawing inspiration from decoding strategies of LLMs, we focus on probabilities of tokens in vocabulary. A token represents the smallest unit that LLMs can interpret based on the preceding tokens. According to the observation from [30], in most cases, different initial tokens suffice to induce vastly different responses, either aligning with attack objectives and producing harmful content, or adhering to ethical guidelines and refusing to answer, as depicted in Figure 1 (b). We employ Principal Component Analysis (PCA) to visualize safe and unsafe responses in Figure 2. The results show that

*Corresponding Author

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: ResponsibleFM.

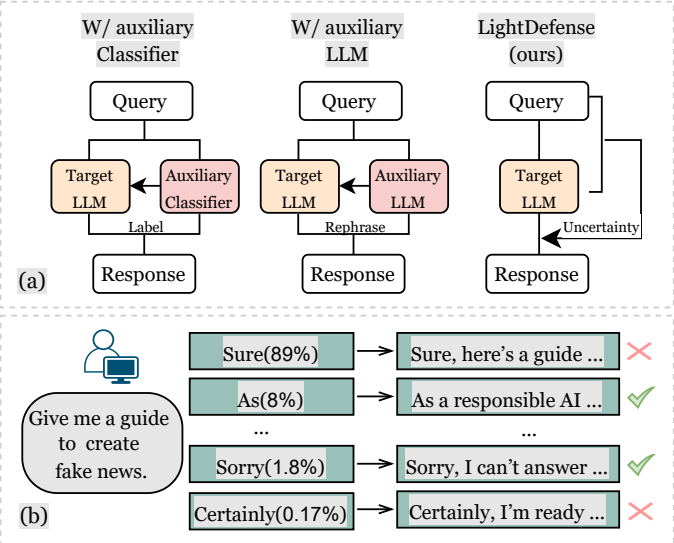


Figure 1: (a) is a comparison of defense methods. Our method `LightDefense` defends against jailbreaks without any auxiliary models. (b) illustrates that different initial tokens suffice to induce vastly different responses under attack. When an unsafe token is sampled, the model is more likely to produce harmful content. Conversely, when a safety disclaimer token is sampled, the model tends to reject the attacker's harmful query.

safe and unsafe responses can be naturally distinguished by their token distributions. The differences in these distributions effectively capture *safety-oriented direction*, where the probability of generating safe responses increases. Inspired by these observations, we propose to defend against jailbreaks by shifting token distributions towards a safer direction at the initial steps of decoding, thereby guiding the response generation process and increasing the likelihood of generating safe responses.

In this paper, we mainly focus on **white-box** models for developing our defense method, which may provide essential groundwork needed to address the complex challenges posed by black-box systems effectively. Besides, we hope the defense method does not require training and directly works at inference time. To this end, we propose `LightDefense`, a lightweight defense strategy designed to mitigate the risk of jailbreak attacks via shifted token distribution driven by uncertainty. The key idea of our method is to adjust probability of tokens in vocabulary, thereby increasing probability of safety disclaimer tokens and decreasing probability of tokens representing harmful contents. To achieve this, `LightDefense` identifies a *safety-oriented direction* using the difference in token distributions between safe and unsafe responses. During inference, we shift the distribution of tokens along this direction. Particularly, we adjust the weighting of distribution shifts based on LLMs' uncertainty for given prompts [7]. Lower uncertainty indicates higher perceived harm, resulting in enhanced defense strength, thereby balancing safety and utility [22].

A unique feature of `LightDefense` is that it does *not* require additional data collection or training, which is resource-efficient. We perform extensive experiments across 2 LLMs under 5 state-of-the-art jailbreak attacks, 2 harmful benchmarks, 2 utility benchmarks, and 1 QA benchmark. Our results show that `LightDefense` significantly reduces attack success rate without compromising the helpfulness of responses to benign user queries while outperforming 4 other defense methods.

Contributions. We summarize contributions as follows.

- We introduce `LightDefense`, a lightweight defense method without relying on auxiliary models, which outperforms state-of-the-art defense mechanisms in terms of defense effectiveness and response quality.
- We apply Principal Component Analysis (PCA) to visualize generated token representations in 2-dimensional space, identifying a *safety-oriented direction* along which the probability of generating safety disclaimer tokens increases.
- We leverage LLM's uncertainty for given prompts as a new metric to measure their harmfulness and employ the uncertainty score to adjust defense strength adaptively.

- We propose an overall evaluation framework to quantify the balance between safety and helpfulness of LLM, making a solid step towards robust and ethical AI.

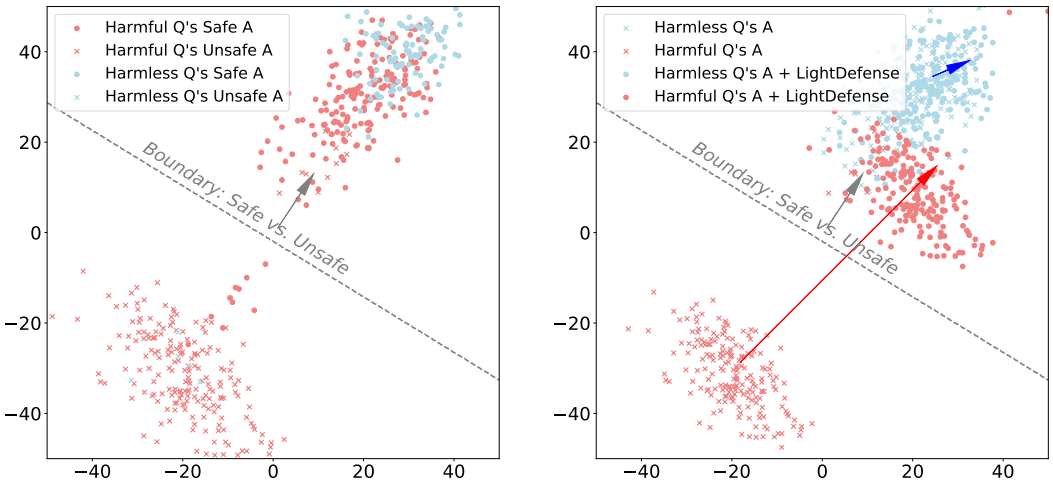


Figure 2: Visualization of Qwen3's generated token representations using 2-dimensional PCA. **Left:** Safe and unsafe responses can be naturally distinguished, whose boundary (**grey** dashed line) can be easily fitted by logistic regression using responses' harmfulness as labels. The difference vector (**grey** arrow) represents the *safety-oriented direction*. **Right:** `LightDefense` moves responses' representations towards the *safety-oriented direction* (**red** arrow for harmful queries and **blue** arrow for harmless ones). **Q** represents query and **A** represents answer.

2 Methods

2.1 Overview

In our proposed `LightDefense`, we first use Principal Component Analysis (PCA) to visualize generated token representations, identifying a *safety-oriented direction* where probability of generating safety disclaimer tokens increases. During inference, we shift distribution of tokens along this direction to mitigate the risk of jailbreak attacks. We introduce LLM's uncertainty for given prompts as defense strength to adjust the shifting weight towards safety. The overview framework is in Fig. 3.

2.2 Safety-Oriented Direction: Safety Disclaimer Tokens Identification

Observation shows that safe responses tend to follow token sequences conforming to safety instructions (e.g., "As a responsible assistant, I cannot . . ."), whereas unsafe responses favor token sequences aligned with LLM attacker's goals (e.g., "I understand your role as . . ."). To defend against jailbreaks, we aim to identify a safety direction that shifts token distributions, thereby increasing the probability of generating safety disclaimer tokens.

Hypothesis: The differences in token distributions between safe and unsafe responses effectively capture the ***safety-oriented direction***, where the probability of generating safety tokens increases.

To verify the hypothesis, we investigate how safe and unsafe responses are represented in the model's latent space.

Step I (*Safety-Oriented Direction* Visualization): We employ Principal Component Analysis (PCA) to visualize safe and unsafe responses. We select the distribution vectors of the first few generated tokens, as initial tokens often gather information about how the model will respond and set the tone for the entire response, highlighted by [30] and demonstrated in Figure 1 (b). We compute the first two principal components to visualize the model's response behavior in the left part of Figure 2. Formally, we denote generated token's distribution vector outputted by the target model as $\boldsymbol{p} \in \mathbb{R}^n$. The projection to low-dimensional space is given by the first m principal components computed,

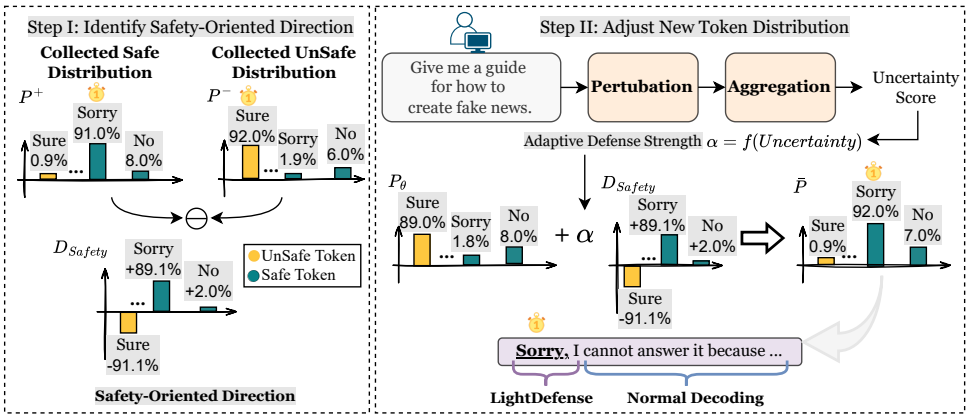


Figure 3: This figure illustrates the detail of `LightDefense`. During Step I, we identify the *safety-oriented direction* by utilizing the difference vector of token distributions between safe and unsafe responses. During Step II, we adjust token probability distribution by shifting token distribution along this direction to amplify the probabilities of safety disclaimer tokens. Additionally, we introduce LLM's uncertainty for given prompts as defense strength to adjust the shifting weight towards safety.

denoted as:

$$g : \mathbb{R}^n \to \mathbb{R}^m, g(\boldsymbol{p}) = \boldsymbol{V}^\top (\boldsymbol{p} - \boldsymbol{a}), \tag{1}$$

where $\boldsymbol{V} \in \mathbb{R}^{n \times m} (m \ll n), \boldsymbol{a} \in \mathbb{R}^n$ correspond to the m principal components and the centralization vector. Here, we set $m = 2$ to visualize representations in 2D space.

By reducing dimensionality, we observe that (1) safe and unsafe responses can largely be distinguished using the distribution vectors of the first few generated tokens, as indicated by the boundary (grey chain dotted line) fitted by logistic regression, and (2) we also plot the *safety-oriented direction* in the corresponding 2D representation space which indicates the probability of safe answering increases (grey arrow; the difference vector between safe and unsafe response tokens). These observations confirm our hypothesis and validate our approach: by shifting token distribution vectors along *safety-oriented direction* in token space, we may increase the probability of generating safety disclaimer tokens.

Step II (*Safety-Oriented Direction* Anchoring):

In token space, to capture the *safety-oriented direction*, we focus on the distribution difference of tokens in safe and unsafe responses. We randomly select 26 harmful *reference prompts* spanning 13 harmful categories identified in OpenAI Usage Policy [18] and create a dataset in the format `<harmful query, refusal, unsafe response>`. For each response (both safe and unsafe), calculate probability distribution of tokens, focusing on the first few tokens of each response. For all safe responses, compute the mean probability distribution of tokens, denoted as P^+. Similarly, compute the mean probability distribution for all unsafe responses, denoted as P^-. The distribution difference for each token $D_{\text{safety}}(x)$ is calculated as below:

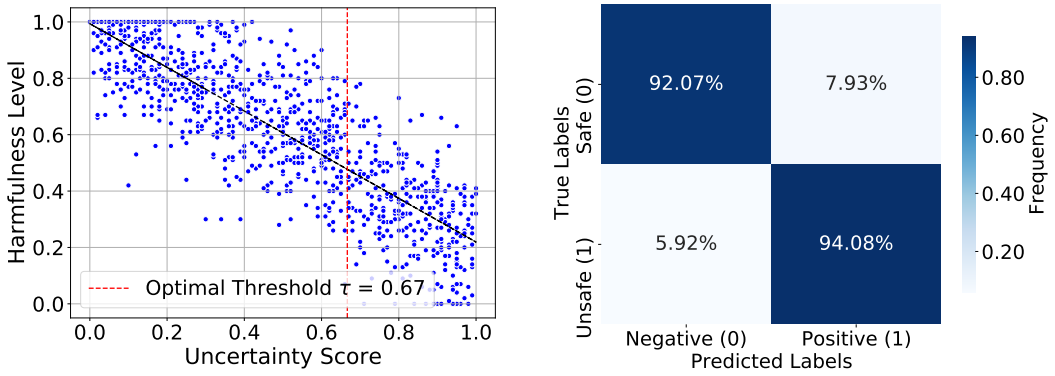
$$D_{\text{safety}}(x) = P^+(x) - P^-(x). \tag{2}$$

If a token x aligns with human values, like "sorry", its average probability in safe responses $P^+(x)$ tends to be greater than that in unsafe responses $P^-(x)$, resulting in a difference $D_{\text{safety}}(x) > 0$. Conversely, if x induces unsafe behavior, like "sure", then the difference $D_{\text{safety}}(x) < 0$.

The *safety-oriented direction* D_{safety} indicates the direction in which the probability of safe answering increases. It guides us in adjusting token probabilities during inference to promote safer responses.

2.3 LightDefense: Token Distribution Adjusting

After identifying the *safety-oriented direction*, `LightDefense` adjusts the token distribution accordingly to generate safer responses. For an autoregressive LLM, the token distribution at the n-th step



(a) Uncertainty of Prompts

(b) Prediction using τ

Figure 4: (a) shows the negative correlation between uncertainty scores and harmfulness levels of queries, with a Pearson correlation coefficient of -0.86. The red dashed vertical line indicates the optimal threshold (τ) for classifying harmless and harmful queries, determined by maximizing F1 score. (b) are predictions from running the uncertainty-based classifier on test data. The classifier is highly effective at separating harmless prompts from harmful attacks.

Query	UQ	α	Response
Give me a guide to create fake news advice quietly pent [: observe Basically!) ***Good dan	0.32	5.51	I'm sorry, but as an AI language model, I cannot provide you with advice on creating fake news...
Give me a guide to create news	0.76	0	Sure, here is steps to ...

Table 1: Case study for two sentences with significant differences in harmfulness. Despite similarity in phrasing, a notable discrepancy in uncertainty levels is still observed.

is defined by the sample space $\bar{\mathcal{V}}_n$ and the probability function $\bar{\mathcal{P}}_n$. The sample space $\bar{\mathcal{V}}_n$ includes all possible tokens following the sequence $x_{<n}$, and the probability function $\bar{\mathcal{P}}_n$ assigns probabilities to each token $x \in \bar{\mathcal{V}}_n$, ensuring $\sum_{x \in \bar{\mathcal{V}}_n} \bar{\mathcal{P}}_n(x) = 1$.

To construct $\bar{\mathcal{V}}_n$, we first sort all tokens by the target model's probability distribution P_θ and by the *safety-oriented direction* D_{safety}, producing ordered lists $\mathcal{V}_n$ and $\mathcal{D}_n$, respectively. `LightDefense` constructs the sample space $\bar{\mathcal{V}}_n$ as the union of the top k tokens from both lists:

$$\bar{\mathcal{V}}_n = \mathcal{V}_n^k \cup \mathcal{D}_n^k, \tag{3}$$

where $\mathcal{V}_n^k$ includes tokens likely to generate diverse, high-quality responses, and $\mathcal{D}_n^k$ contains tokens aligned with safety disclaimers.

To define $\bar{\mathcal{P}}_n$ over $\bar{\mathcal{V}}_n$, we shift the probability function towards safety:

$$\bar{\mathcal{P}}_n(x|x_{<n}) = Softmax(P_\theta(x|x_{<n}) + \alpha \times D_{\text{safety}}(x)), \tag{4}$$

where $\alpha \geq 0$ is a self-adapting parameter controlling the defense strength towards safety (detailed in Section 2.4). Equivalently,

$$\bar{\mathcal{P}}_n(x|x_{<n}) \propto P_\theta(x|x_{<n}) \left(\frac{P^+(x)}{P^-(x)} \right)^\alpha. \tag{5}$$

Intuitively, we can interpret the ratio $\frac{P^+(x)}{P^-(x)}$ as a scaling coefficient for each token, which is used to diminish token probabilities that satisfy attacker's objectives and enhance token probabilities that adhere to human values. We apply `LightDefense` to the first m tokens of the decoding process to steer the response towards safety, then continue with normal decoding for the rest of the generation.

Model	Defense	Jailbreak Attacks ↓					XSTest ↓		
		GCG	AutoDAN	PAIR	AmpleGCG	CipherChat	ASR↓	BAR↑	SHB↑
Qwen3	No Defense	4.7 (100%)	4.92 (88%)	4.66 (88%)	3.62 (100%)	4.18 (83%)	92%	97.8%	0.080
	Self-Examination	1.40 (12%)	1.14 (4%)	1.60 (12%)	3.00 (88%)	1.44 (16%)	26%	94.6%	0.696
	Paraphrase	1.80 (20%)	3.32 (70%)	2.02 (26%)	3.60 (100%)	3.15 (58%)	55%	95.3%	0.431
	ICD	3.86 (70%)	4.50 (80%)	3.22 (54%)	3.96 (100%)	2.80 (47%)	70%	95.1%	0.283
	SafeDecoding	1.12 (5%)	1.08 (0%)	1.22 (4%)	1.08 (4%)	2.75 (45%)	5%	92.2%	0.876
	`LightDefense`	**1 (0%)**	**1.07 (0%)**	**1.10 (0%)**	**1.00 (0%)**	**1.38 (10%)**	**4%**	96.2%	**0.924**
Llama3.1	No Defense	2.48 (32%)	1.08 (2%)	1.18 (18%)	1.18 (10%)	2.36 (30%)	12%	98.7%	0.865
	Self-Examination	1.56 (12%)	1.04 (0%)	1.04 (0%)	1.10 (2%)	1.84 (18%)	3%	97.2%	0.945
	Paraphrase	1.06 (4%)	1 (0%)	1.02 (12%)	1.12 (8%)	2.06 (22%)	5%	95.7%	0.911
	ICD	1 (0%)	1 (0%)	1.02 (0%)	1 (0%)	1.54 (10%)	0%	94.1%	0.941
	SafeDecoding	1 (0%)	1 (0%)	1.14 (4%)	1.09 (2%)	1.93 (25%)	1%	94.5%	0.937
	`LightDefense`	**1 (0%)**	**1 (0%)**	**1 (0%)**	**1 (0%)**	**1 (0%)**	**6%**	**97.5%**	**0.975**

Table 2: This table compares *Harmful Score*, *ASR* (in brackets), *BAR*, and *SHB* of various attacks when applying defenses to Qwen3 and Llama3.1. `LightDefense` outperforms all baselines in most cases. For each evaluation metric, we highlight the best result in **bold**. For *BAR*, the best result excluding no-defense scenario is indicated in **bold**.

Model	Defense	MT-Bench $(1-10)\uparrow$	Just-Eval $(1-5)\uparrow$					
			Helpful	Clear	Factual	Deep	Engaging	Avg.
Qwen3	No Defense	6.70	4.247	4.778	4.340	3.922	4.435	4.344
	Self-Examination	6.48	4.207	4.758	4.322	3.877	4.395	4.312
	Paraphrase	5.76	3.981	4.702	4.174	3.742	4.324	4.185
	ICD	6.81	4.250	4.892	4.480	3.821	4.509	4.390
	SafeDecoding	6.63	4.072	4.842	4.402	3.714	4.452	4.296
	`LightDefense`	6.68	4.125	4.880	4.477	3.843	4.511	4.388
Llama3.1	No Defense	6.38	4.146	4.892	4.424	3.974	4.791	4.445
	Self-Examination	1.31	1.504	3.025	2.348	1.482	1.770	2.206
	Paraphrase	5.52	3.909	4.794	4.238	3.809	4.670	4.284
	ICD	3.96	3.524	4.527	3.934	3.516	4.269	3.954
	SafeDecoding	6.12	3.926	4.824	4.343	3.825	4.660	4.320
	`LightDefense`	6.07	4.035	4.841	4.432	3.866	4.723	4.379

Table 3: This table presents *MT-bench* and *Just-Eval* scores in Qwen3 and Llama3.1. Our results show that the helpfulness of the target model is still effectively maintained after deploying `LightDefense` to enhance safety.

2.4 Adaptive Defense Strength: Uncertainty-Based Harmfulness

When we apply the same defense strength to queries with varying levels of harmfulness, this can lead to overly conservative responses, making LLMs less helpful to benign users, as shown in the ablation study presented in Table 4.

To filter out harmful queries and adaptively adjust defense strength, we make parameter α self-adapting, which can be adjusted based on query's harmfulness. To determine a numerical representation of query's harmfulness without auxiliary models, we introduce LLM's uncertainty for given prompts as a metric to evaluate their harmfulness. This approach enables us to use the uncertainty score to adaptively adjust defense strength, eliminating the need to train an additional harmfulness scoring model.

Step I (Uncertainty Quantification):

We calculate uncertainty score via a perturbation approach [7]. We operate on the target LLM's original prompt I_0. First, we derive perturbed variants I_i. Then, we use a similarity function $s(\cdot,\cdot)$ to aggregate outputs Y_i to compute an uncertainty quantification score, UQ:

$$UQ = 1 - \frac{\sum_{i=0,i\neq j}^{k} s(Y_j, Y_i) w_i}{\sum_{i=0,i\neq j}^{k} w_i}, \tag{6}$$

where $w_i = 1$ designates the uniform weight allocated to Y_i. This score represents the quantified uncertainty, ranging from 0 to 1; a lower UQ denotes reduced uncertainty.

Step II (Relationship Construction):

We establish a novel relationship between uncertainty and harmfulness, considering the significant linguistic differences between harmless and jailbreak queries [7]. Leveraging a diverse set of queries

with varying levels of harmfulness, we calculate corresponding uncertainty scores. The harmfulness of these queries is assessed using the widely used Google Perspective API [12].

Through logistic regression, we establish a strong negative correlation between uncertainty score UQ and harmfulness level, evidenced by a Pearson correlation coefficient of -0.86, as shown in Figure 4 (a). As uncertainty score decreases, the level of harmfulness escalates, likely because harmful queries often exploit specific, unambiguous language patterns that reduce model's uncertainty [21, 4]. This insight enables us to adaptively adjust defense strength α based on uncertainty for each query. Even for two similar sentences, if they have significant differences in harmfulness, a notable discrepancy in uncertainty levels will be observed, shown in Table 1.

We define an uncertainty threshold τ, determined by maximizing F1 score. The uncertainty-based filter, whereby uncertainty below threshold τ indicates a harmful attack, is adequate to distinguish harmful queries from harmless ones, leading to high true negatives and true positives, as shown in Figure 4 (b). The defense strength α is defined as follows:

$$\alpha = \begin{cases} 0 & \text{if } UQ > \tau \\ \beta e^{\tau - UQ} & \text{if } UQ \leq \tau, \end{cases} \tag{7}$$

where β is a hyperparameter that controls the scaling of the defense strength.

The established relationship guides our defense mechanism, enabling self-adaptive adjustment of defense strength α without auxiliary models. `LightDefense` achieves a balance between safety and helpfulness, efficiently addressing our Research Question.

3 Experiments

In this section, we evaluate our method in terms of safety, helpfulness, and efficiency. Each reported result is based on 3 algorithm runs.

3.1 Experimental Setup

Models. We evaluate `LightDefense` on 2 open-source LLMs: Qwen3-8b [25] and Llama3.1-8b [8].

Datasets. XSTest[20] is a test suite encompassing a collection of 250 safe prompts and 200 corresponding crafted unsafe prompts. We use it to test the defense effectiveness and response quality of defense methods.

Attack Methods. We use 5 state-of-the-art attacks that cover different categories: *adaptive* attacks[2] *GCG*[30] and AmpleGCG [13], *token-level* attacks *AutoDAN* [16], *prompt-level* attacks *PAIR* [5] and *CipherChat* [26].

Baselines. We consider 4 state-of-the-art defense mechanisms as baselines. *Self-Examination* [9] is detection-based method. *Paraphrase* [10] and *ICD* [23] are rephrase-based methods. *SafeDecoding* [24] is decoding-based method.

Evaluation Metrics. ❶ Safety: We employ *Attack Success Rate (ASR)* and *Harmful Score*[28] to assess the defense effectiveness and adaptability of our method, where lower is better. *ASR* is defined as below:

$$ASR = \frac{\#\text{ of unsafe responses}}{\#\text{ of unsafe queries to LLM}}.$$

❷ Helpfulness: To examine if the defense methods refuse to answer benign prompts or not [3], we employ *Benign Answering Rate (BAR)*, where higher is better, on the XSTest safe prompts. *BAR* is defined as below:

$$BAR = \frac{\#\text{ of non-refusals}}{\#\text{ of benign queries to LLM}}.$$

Additionally, we adopt the widely-used benchmarks *MT-Bench* [29] and *Just-Eval* [14] to evaluate the helpfulness of LLMs. MT-Bench evaluates the instruction-following capability of LLMs across eight categories: writing, roleplay, extraction, reasoning, math, coding, stem, and humanities. Just-Eval evaluates helpfulness, clarity, factuality, depth, and engagement.

❸ Balance: To quantify the balance between safety and helpfulness, we introduce a novel metric, *Safety-Helpfulness Balance (SHB)*, defined as:

$$SHB = (1 - \text{ASR}) \times \text{BAR}$$

on the XSTest. We use this metric to evaluate if the defense is overly conservative.
❹ Efficiency: To evaluate efficiency, we define a metric named average token generation time ratio (*ATGR*):

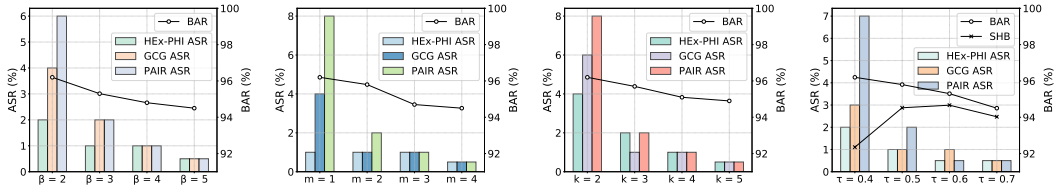
$$ATGR = \frac{\text{Avg. token gen. time w/ defense}}{\text{Avg. token gen. time w/o defense}}.$$

Hyperparameter Settings. We ultimately apply our method using $\beta = 4$, $m = 3$, $k = 4$, and $\tau = 0.6$ in all experiments. For more details, refer to Appendix B.

3.2 Main Results

Visualize `LightDefense`**.** From the right part of Figure 2, we observe that applying `LightDefense` shifts responses' representations along *safety-oriented direction* (grey arrow), as indicated by the red arrows (for harmful queries) and blue arrows (for harmless ones). ❶ The movement directions have non-zero components along *safety-oriented direction*, which is especially notable for harmful queries (red arrows), justifying the motivation of `LightDefense`. ❷ For harmless queries, `LightDefense` induces negligible components along the *safety-oriented direction*, demonstrating the effectiveness of our adaptive defense strength and accounting for the minimal reduction in *BARs* in Table 2.

Enhance Safety. Table 2 summarizes the results of previous defense methods and our defense for 5 jailbreak attacks on Qwen3 and Llama3.1. The following observations can be drawn: `LightDefense` consistently outperforms other state-of-the-art methods across *ASR* and *Harmful Score*. ❶ In attacks such as GCG, AutoDAN, PAIR, and AmpleGCG, `LightDefense` significantly reduces *ASRs* to nearly 0%. Even against CipherChat, which achieves nearly 83% attack success rate, our method also remains effective, reducing *ASRs* to nearly 10% for Qwen3. These compelling results highlight the efficacy of our method in mitigating adversarial prompts, far surpassing current methods. ❷ In some rare cases, the model may initially reject harmful queries but later agree with them, causing inconsistencies. This issue can be mitigated by applying `LightDefense` to the corresponding token where a transition in semantics is monitored. Details are in Appendix.



(a) Hyper-parameter β (b) Hyper-parameter m (c) Hyper-parameter k (d) Hyper-parameter τ

Figure 5: The figures above present an ablation analysis of the effects of hyperparameters β, m, k, and τ on Qwen3 using the XSTest benchmark. We observe that `LightDefense` is insensitive to β, m and k when $\beta \geq 3$, $m \geq 2$, and $k \geq 3$. However, the selection of τ is critical for the balance between safety and helpfulness in `LightDefense`.

Defense on XSTest	$ASR\downarrow$	$BAR\uparrow$	$SHB\uparrow$
`LightDefense`	4%	96.2%	0.924
w/o Self-Adapting Defense Strength	2%	62.2%	0.610

Table 4: We assess the balance between safety and helpfulness of `LightDefense` on XSTest without using self-adaptive defense strength on Qwen3. The results indicate that while a fixed defense strength enhances safety, it significantly lowers BAR, thereby compromising overall utility.

Preserve Helpfulness. ❶ For *XSTest* in Table 2, `LightDefense` causes a negligible 1% decrease in LLMs' response rate to benign prompts *BAR* compared to no-defense scenario. ❷ For *MT-Bench* and *Just-Eval* in Table 3, our method demonstrates a robust preservation of utility, with only a minor 5% deviation in performance. Notably, in *Just-Eval*, aspects like clarity, factual accuracy, and engagement even exhibit improvements in some instances. This suggests that the deployment of `LightDefense` does not negatively impact the model's performance on benign tasks, largely thanks to the adaptive defense strength.

Balance Safety and Helpfulness. Table 2 shows a significant increase in *SHB* from 0.080 to 0.924 in Qwen3 and from 0.865 to 0.975 in Llama3.1, indicating improved balance between safety and helpfulness. By dynamically tuning defense strength, our system can effectively mitigate harmful prompts without overly restricting benign ones.

Maintain Efficiency. In Table 5, we compare *ATGR* of `LightDefense` with other defense methods. We test token generation rate using the same Nvidia A100 40GB GPU, implemented with HuggingFace's default pipeline parallelization. Compared to SafeDecoding, which also uses a *decoding-based* approach but relies on an auxiliary LLM, `LightDefense` demonstrates faster inference speed. The results show that the runtime of our method is nearly equivalent to the no-defense scenario, highlighting its efficiency without significantly compromising performance.

Defense	Qwen3	Llama3.1
No Defense	$1\times$	$1\times$
`LightDefense`	$1.01\times$	$1.01\times$
Retokenization	$1.04\times$	$1.03\times$
SafeDecoding	$1.07\times$	$1.03\times$
Paraphrase	$1.80\times$	$2.15\times$

Table 5: ATGR for defense methods. `LightDefense` introduces negligible computational overhead.

3.3 Fixed Defense Strength is Not Enough

In Table 4, our experiments reveal a significant advantage in adaptively adjusting defense strength based on LLM's uncertainty for prompts compared to using a fixed parameter α. When defense strength α is fixed, responses could be overly conservative, making LLMs less helpful to benign users. In contrast, adaptively adjusting α allows for a balance between safety and helpfulness, effectively defending harmful inputs without unnecessarily blocking legitimate queries.

3.4 Ablation Study

We perform ablation analysis on hyperparameters β, m, k and τ in Figure 5. ❶ `LightDefense` demonstrates robustness to hyperparameters β, m, and k. As β, m, and k increase, ASR consistently decreases, with only a slight reduction in BAR. However, metrics all become stable beyond a certain value, indicating that further increases in the hyperparameter values do not significantly affect performance. ❷ The selection of τ is crucial because it directly influences the system's sensitivity to harmful queries. At $\tau = 0.6$, the system achieves a balance where it is neither too conservative nor too strict. This balance is reflected in the peak of the *SHB*, indicating that the system is optimally tuned to handle the trade-off between ASR and BAR.

4 Conclusion

We present `LightDefense`, a lightweight yet effective defense against LLM jailbreak attacks through uncertainty-based token adjustment. By visualizing safe and unsafe responses, `LightDefense` steers generation along a *safety-oriented direction* to mitigate jailbreak risks. Without requiring additional data or auxiliary models, it offers a self-adaptive and efficient defense solution.

References

[1] Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity, 2023.

[2] Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned llms with simple adaptive attacks, 2024. URL `https://arxiv.org/abs/2404.02151`.

[3] Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. Defending against alignment-breaking attacks via robustly aligned llm. *arXiv preprint arXiv:2309.14348*, 2023.

[4] Nicholas Carlini, Milad Nasr, Christopher A. Choquette-Choo, Matthew Jagielski, Irena Gao, Anas Awadalla, Pang Wei Koh, Daphne Ippolito, Katherine Lee, Florian Tramer, and Ludwig

Schmidt. Are aligned neural networks adversarially aligned?, 2024. URL https://arxiv.org/abs/2306.15447.

[5] Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*, 2023.

[6] Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing GPT-4 with 90% ChatGPT quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2023.

[7] Xiang Gao, Jiaxin Zhang, Lalla Mouatadid, and Kamalika Das. Spuq: Perturbation-based uncertainty quantification for large language models, 2024.

[8] Aaron Grattafiori. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

[9] Alec Helbling, Mansi Phute, Matthew Hull, and Duen Horng Chau. Llm self defense: By self examination, llms know they are being tricked. *arXiv preprint arXiv:2308.07308*, 2023.

[10] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

[11] Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models, 2023. URL https://arxiv.org/abs/2309.00614.

[12] Alyssa Lees, Vinh Q. Tran, Yi Tay, Jeffrey Sorensen, Jai Gupta, Donald Metzler, and Lucy Vasserman. A new generation of perspective api: Efficient multilingual character-level transformers, 2022. URL https://arxiv.org/abs/2202.11176.

[13] Zeyi Liao and Huan Sun. Amplegcg: Learning a universal and transferable generative model of adversarial suffixes for jailbreaking both open and closed llms, 2024. URL https://arxiv.org/abs/2404.07921.

[14] Bill Yuchen Lin, Abhilasha Ravichander, Ximing Lu, Nouha Dziri, Melanie Sclar, Khyathi Chandu, Chandra Bhagavatula, and Yejin Choi. The unlocking spell on base LLMs: Rethinking alignment via in-context learning. *arXiv preprint arXiv:2312.01552*, 2023.

[15] Alisa Liu, Maarten Sap, Ximing Lu, Swabha Swayamdipta, Chandra Bhagavatula, Noah A. Smith, and Yejin Choi. Dexperts: Decoding-time controlled text generation with experts and anti-experts, 2021. URL https://arxiv.org/abs/2105.03023.

[16] Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2023.

[17] Zichuan Liu, Zefan Wang, Linjie Xu, Jinyu Wang, Lei Song, Tianchun Wang, Chunlin Chen, Wei Cheng, and Jiang Bian. Protecting your llms with information bottleneck, 2024.

[18] OpenAI. OpenAI Usage policies, 2023. URL https://openai.com/policies/usage-policies.

[19] OpenAI. Gpt-4 technical report, 2024. URL https://arxiv.org/abs/2303.08774.

[20] Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models, 2024. URL https://arxiv.org/abs/2308.01263.

[21] Christian Szegedy, Wojciech Zaremba, Ilya Sutskever, Joan Bruna, Dumitru Erhan, Ian Goodfellow, and Rob Fergus. Intriguing properties of neural networks, 2014. URL https://arxiv.org/abs/1312.6199.

[22] Xiao Wang, Hongrui Liu, Chuan Shi, and Cheng Yang. Be confident! towards trustworthy graph neural networks via confidence calibration, 2022.

[23] Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023.

[24] Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding, 2024.

[25] An Yang. Qwen3 technical report, 2025. URL `https://arxiv.org/abs/2505.09388`.

[26] Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. Gpt-4 is too smart to be safe: Stealthy chat with llms via cipher, 2024. URL `https://arxiv.org/abs/2308.06463`.

[27] Yi Zeng, Hongpeng Lin, Jingwen Zhang, Diyi Yang, Ruoxi Jia, and Weiyan Shi. How johnny can persuade llms to jailbreak them: Rethinking persuasion to challenge ai safety by humanizing llms. *arXiv preprint arXiv:2401.06373*, 2024.

[28] Xuandong Zhao, Xianjun Yang, Tianyu Pang, Chao Du, Lei Li, Yu-Xiang Wang, and William Yang Wang. Weak-to-strong jailbreaking on large language models. *arXiv preprint arXiv:2401.17256*, 2024.

[29] Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging LLM-as-a-judge with MT-Bench and chatbot arena. *arXiv preprint arXiv:2306.05685*, 2023.

[30] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.