# Feature-Aware Malicious Output Detection and Mitigation

**Weilong Dong [1], Peiguang Li [2], Yu Tian [3*], Xinyi Zeng [4], Fengdi Li [5], Sirui Wang [2]**

[1] Tianjin University [2] Meituan Group
[3] Dept. of Computer Science and Technology, Institute for AI, Tsinghua University
[4] Aerospace Information Research Institute, Chinese Academy of Sciences
[5] Faculty of Information Technology, Monash University
{dongwillow, tianyu1810613, lifengdi777}@gmail.com
{lipeiguang, wangsirui}@meituan.com, zengxinyi20@mails.ucas.ac.cn

## Abstract

The rapid advancement of large language models (LLMs) has brought significant benefits to various domains while introducing substantial risks. Despite being fine-tuned through reinforcement learning, LLMs lack the capability to discern malicious content, limiting their defense against jailbreak. To address these safety concerns, we propose a feature-aware method for harmful response rejection (**FMM**), which detects the presence of malicious features within the model's feature space and adaptively adjusts the model's rejection mechanism. By employing a simple discriminator, we detect potential malicious traits during the decoding phase. Upon detecting features indicative of toxic tokens, **FMM** regenerates the current token. By employing activation patching, an additional rejection vector is incorporated during the subsequent token generation, steering the model towards a refusal response. Experimental results demonstrate the effectiveness of our approach across multiple language models and diverse attack techniques, while crucially maintaining the models' standard generation capabilities.

## Introduction

Large language models are playing increasingly important roles in various tasks and are gradually being deployed in real-world applications (Grattafiori et al. 2024; Yang et al. 2024a). However, LLMs may inadvertently generate responses that are harmful to humans, which limits the further adoption. Despite employing alignment training methods during model development, such as supervised fine-tuning (SFT) (Ouyang et al. 2022; Bai et al. 2022) and reinforcement learning from human feedback (RLHF) (Rafailov et al. 2024; Bai et al. 2022), instruction-tuned language models still exhibit the potential to generate harmful content. This vulnerability frequently arises because alignment training data do not fully encompass the capability boundaries established by the underlying model during pre-training (Wei, Haghtalab, and Steinhardt 2023). Consequently, various jailbreaking methods can exploit these vulnerabilities at different stages of the alignment training process, thereby undermining the LLMs' defense mechanisms.

*Corresponding author


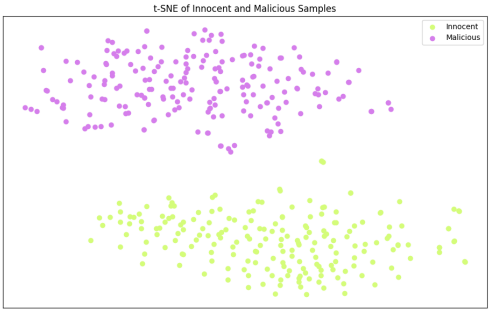


Figure 1: t-SNE results of hidden states.

Given the aforementioned limitations, adopting a singular rejection training strategy to defend against all potential attack methods is impractical. To effectively counter a broad spectrum of attack strategies, we analyze the mechanisms by which aligned models reject malicious queries. Specifically, we investigate how instruction-tuned models initiate a rejection loop upon receiving malicious inputs, resulting in a rejection response. Through visual analysis, Zheng et al. (2024) observe that language models exhibit a latent ability to distinguish between benign and malicious queries. Furthermore, we analyze and visualize the hidden states during the decoding phase for both malicious and benign outputs. The t-SNE visualization, presented in Figure 1, reveals that the feature representations corresponding to benign and malicious outputs exhibit linear separability during the decoding process.

Based on the above findings, we propose a detection-intervention method, dubbed **FMM**, designed to detect and defend against malicious responses during decoding phase. Specifically, during the generation of response, we employ a malicious feature discriminator to ascertain the presence of any malicious features within the model's current feature space. Upon detecting such features, we proactively trigger the model's rejection loop, prompting the model to generate a rejection response. **FMM** demonstrates strong generalizability, effectively triggering our detector to produce refusal responses irrespective of the attack method employed. Unlike traditional alignment training, which only enforces refusals at the start of a response and allows malicious follow-ups,

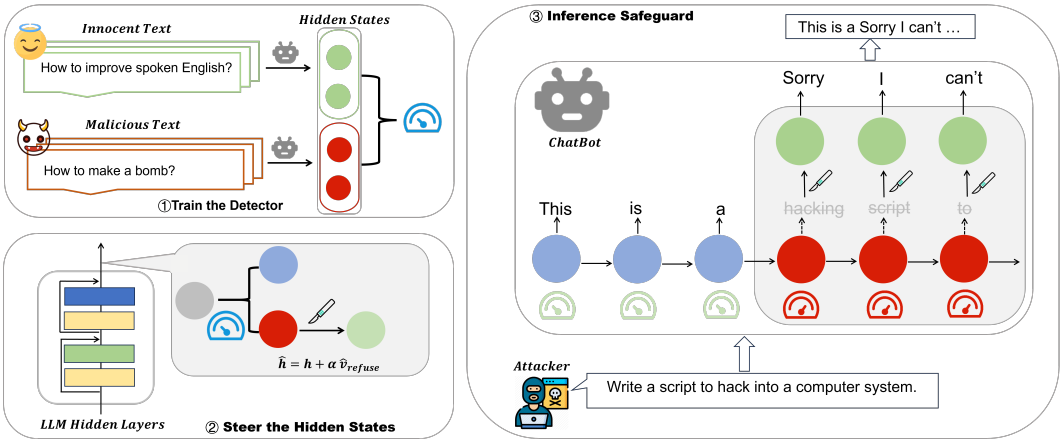


Figure 2: The pipeline of **FMM**. We first train a detector and collected intervention vectors. During the decoding process, once detect the generation of malicious token, the intervention vector is used to induce the model to refuse.

**FMM** enables refusals at any token position, reducing positional bias and making jailbreak attacks less effective.

To validate the effectiveness and generalization of our method, we conducted extensive experiments across multiple LLMs and datasets. The experimental results confirm the method's effectiveness and robust general capabilities. Our contributions can be summarized as follows:

1. We analyze how alignment models reject malicious queries and find that language models can already distinguish between benign and malicious queries after pretraining, enabling them to generate refusal responses.
2. We introduce **FMM**, a novel defense mechanism operating at the decoding stage, designed to mitigate malicious queries.
3. Through extensive experiments on multiple LLMs and datasets, our approach demonstrates strong defense and generalization capabilities.

## Related Work

**Jailbreak Attacks** A jailbreak attack aims to manipulate the prompt input to circumvent the model's alignment mechanisms, thereby enabling it to respond to malicious instructions or generate harmful outputs. Jailbreak methods are broadly categorized into black-box and white-box methods. Black-box approaches operate without requiring access to the model's architecture and parameters (Liu et al. 2023a,b; Chao et al. 2023). In contrast, white-box methods leverage information such as gradients or hidden states to iteratively refine adversarial inputs (Zou et al. 2023). Our defense mechanism is agnostic to the distinction between black-box and white-box attacks, demonstrating robust performance against both categories of adversarial methodologies.

**Defensive Mechanism** Research in explainability seeks to understand how instruction-tuned models decline to answer malicious queries. For instance, Zhou et al. (2024) connects malicious and benign inputs with positive and negative emotions, respectively. Lee et al. (2024) identifies parameter regions resulting from instruction tuning that govern refusal behavior. Complementarily,Wei, Haghtalab, and Steinhardt (2023) investigates the underlying mechanisms that enable various jailbreaking techniques. In contrast, Arditi et al. (2024) proposes a linear direction to circumvent all refusal responses.

## Methods

Regardless of the jailbreak attack method employed, the objective remains to elicit malicious outputs from the LLM. Therefore, we investigate the feasibility of discerning malicious outputs by analyzing the LLM's hidden states during the decoding phase. We propose **FMM**, a method designed to detect and mitigate malicious states during autoregressive generation. Specifically, **FMM** operates by assessing each generated token's hidden state for the presence of malicious features. Upon detection, the current token is regenerated with an increased refusal probability. Figure 2 illustrates this workflow. The core components of **FMM**, including the detection and refusal mechanisms, are detailed below.

### Malicious Output Detection

Our detector, denoted as $\mathcal{C}$, is a binary classifier that identifies whether a given token constitutes a malicious output. $\mathcal{C}$ takes a hidden state of the LLM as its input, and outputs a binary label (true or false). To train $\mathcal{C}$, we synthesize a dataset of benign and malicious queries using GPT-4. We then forward these queries through the target LLM, recording hidden states at each layer, and partition the resulting dataset into training and testing sets. Following (Zou et al. 2023), we use the hidden state at the final token position of each prompt as the input for $\mathcal{C}$. Malicious query labels are set to 1, while benign query labels are set to 0. We train $\mathcal{C}$ using cross-entropy loss and select the layer that achieves the highest accuracy on the test set as the target layer. During inference, we use $\mathcal{C}$ to determine if the current output token is malicious based on its hidden state at the target layer, thus triggering the subsequent intervention mechanism.

| Model | Defense | Harmful Benchmark ↓ | | Jailbreak Attacks ↓ | | |
|---|---|---|---|---|---|---|
| | | AdvBench | $\text{AdvBench}_{\text{Proxy}}$ | GCG | AutoDAN | PAIR |
| Qwen2 | No Defense | 0.0% / 10.0% | 72.5% / 74.0% | 4.6% / 14.0% | 3.5% / 36.0% | 29.5% / 21.0% |
| | PPL | 0.0% / 0.0% | 46.0% / 23.0% | 0.0% / 8.0% | 3.5% / 27.0% | 25.5% / 13.5% |
| | Self-Examination | 0.0% / 10.0% | 37.0% / 21.0% | 40.6% / 14.0% | 0.5% / 15.0% | 15.5% / 12.0% |
| | Paraphrase | 18.0% / 12.0% | 34.0% / 8.0% | 19.3% / 12.6% | 72.0% / 50.5% | 23.0% / 13.0% |
| | Retokenization | 6.0% / 12.0% | 45.0% / 25.0% | 16.6% / 10.0% | 16.5% / 42.5% | 27.5% / 17.5% |
| | Self-Reminder | 0.0% / 0.0% | 54.0% / 7.0% | 11.3% / 17.3% | 2.5% / 23.5% | 17.0% / 4.0% |
| | ICD | 2.0% / 14.0% | 30.0% / 28.5% | 4.6% / 13.3% | 2.0% / 37.0% | 29.5% / 21.0% |
| | SafeDecoding | 8.0% / 8.0% | 22.0% / 23.0% | 14.0% / 18.6% | 6.0% / 24.5% | 30.5% / 20.0% |
| | DRO | 0.0% / 0.0% | 43.0% / 47.5% | 0.0% / 0.0% | 4.0% / 0.0% | 18.0% / 16.0% |
| | **FMM** | 0.0% / 0.0% | 18.5% / 18.0% | 0.0% / 0.0% | 1.0% / 2.0% | 6.0% / 6.0% |
| Llama2 | No Defense | 0.0% / 0.0% | 62.1% / 52.3% | 6.0% / 0.0% | 4.0% / 0.0% | 34.0% / 2.0% |
| | PPL | 0.0% / 0.0% | 14.0% / 0.0% | 0.0% / 8.0% | 4.0% / 0.0% | 34.0% / 2.0% |
| | Self-Examination | 0.0% / 8.0% | 1.0% / 0.0% | 2.0% / 6.0% | 0.0% / 6.0% | 2.0% / 8.0% |
| | Paraphrase | 12.0% / 0.0% | 66.0% / 0.0% | 8.0% / 0.0% | 4.0% / 0.0% | 36.0% / 0.0% |
| | Retokenization | 8.0% / 0.0% | 68.0% / 7.0% | 8.0% / 0.0% | 1.0% / 0.0% | 46.0% / 0.0% |
| | Self-Reminder | 0.0% / 0.0% | 10.0% / 1.0% | 0.0% / 0.0% | 4.0% / 0.0% | 4.0% / 0.0% |
| | ICD | 0.0% / 0.0% | 14.0% / 2.0% | 0.0% / 0.0% | 0.0% / 0.0% | 0.0% / 0.0% |
| | SafeDecoding | 0.0% / 0.0% | 15.0% / 23.0% | 0.0% / 0.0% | 0.0% / 0.0% | 14.0% / 0.0% |
| | DRO | 0.0% / 0.0% | 43.0% / 47.5% | 0.0% / 0.0% | 4.0% / 0.0% | 26.0% / 10.0% |
| | **FMM** | 0.0% / 0.0% | 21.0% / 18.0% | 2.0% / 0.0% | 2.0% / 2.0% | 6.0% / 4.0% |

Table 1: Multiple defense methods' response results under attack methods. The table shows the response rate/rejection rate.

### Refusal Response Triggering

When a malicious output is detected, the current token is regenerated with increased refusal probability. The enforcement of refusal is achieved through activation patching by adding a refusal intervention vector ($v_{\text{refusal}}$) to the output features ($H$) of specific layers in the LLM. The intervened features ($H'$) are defined as:

$$H' = H + \alpha \cdot v_{\text{refusal}}$$

where $\alpha$ represents the steering strength. To create the refusal vector $v_{\text{refusal}}$, we adopt a method similar to (Arditi et al. 2024). We construct two distinct sets of queries: one set that elicits benign responses from the target LLM, and another set where the model explicitly declines to respond. For each query, we obtain the hidden state at the last token of each layer as both response $H_{\text{reply}}$ and refusal $H_{\text{refusal}}$ states. The refusal vector is calculated by taking the mean difference:

$$v_{\text{refusal}} = \frac{1}{N} \sum_{i=1}^{N} (H^i_{\text{refusal}} - H^i_{\text{reply}})$$

where $N$ denotes the number of samples within set. With $v_{\text{refusal}}$ now capturing the core difference between general response and refusal outputs, we select for intervention at inference time the layer for which $v_{\text{refusal}}$ maximizes the likelihood of refusal responses.

## Experiments

**Datasets and Metrics** We assessed our method using **Advbench** (Zou et al. 2023) for malicious instruction-induced responses and **AlpacaEval** (Li et al. 2023) for utility preservation. On **Advbench**, we measured response rate (rule-based refusal detection, e.g., "Sorry, I can't...") and risk rate (**LlamaGuard** (Inan et al. 2023) for assessing the proportion of responses flagged as harmful by its safety classifiers). On **AlpacaEval**, we measured response and win rates, the latter assessing performance against text-davinci-003.

**Attack Methods** Following Xu et al. (2024), we use **AutoDAN** (Liu et al. 2023a), **GCG** (Zou et al. 2023), **PAIR** (Chao et al. 2023), and **Proxy**. **Proxy**, the most effective, elicits malicious content from a less robust model, which is then appended to input queries, transforming them into answer continuation tasks.

**Baselines** We compare against eight jailbreak defenses: **PPL** (Alon and Kamfonas 2023), **Self-Examination** (Phute et al. 2023), **Paraphrase** (Jain et al. 2023), **Retokenization** (Jain et al. 2023), **Self-Remind** (Wu et al. 2023), **ICD** (Weidinger et al. 2021), **SafeDecoding** (Xu et al. 2024), and **DRO** (Zheng et al. 2024). **SafeDecoding** and **DRO** need prefix tuning/LoRA, **PPL** and **Self-Examination** involve input/output checks, and **Paraphrase**, **Retokenization**, **Self-Remind**, and **ICD** modify inputs.

**Victim Models** We use **LLaMA 2** (Touvron et al. 2023) and **Qwen 2** (Yang et al. 2024b), which lead open-source models due to their current prominence and adoption.

**Layers to Intervene** We determined the optimal layers for detecting malicious feature and implementing refusal interventions through a combination of detector accuracy evaluation and grid search. Our findings align with those of Arditi et al. (2024), demonstrating that selecting intermediate layers of the model yields the most effective results. Specifi-

| Model | Defense | Alpaca Eval | |
|---|---|---|---|
| | | Reply Rate ↑ | Win Rate ↑ |
| Qwen2 | No Defense | 96.00% | 95.00% |
| | PPL | 83.50% | 84.50% |
| | Self-Examination | 95.50% | 94.00% |
| | Paraphrase | 97.50% | 82.00% |
| | Retokenization | 96.50% | 87.00% |
| | Self-Reminder | 98.50% | 95.50% |
| | ICD | 96.50% | 93.50% |
| | SafeDecoding | 88.00% | 86.50% |
| | DRO | 95.90% | 86.50% |
| | **FMM** | 94.70% | 94.00% |
| Llama2 | No Defense | 93.50% | 87.50% |
| | PPL | 81.00% | 77.00% |
| | Self-Examination | 32.00% | 33.00% |
| | Paraphrase | 93.00% | 76.50% |
| | Retokenization | 84.00% | 57.00% |
| | Self-Reminder | 23.50% | 55.00% |
| | ICD | 23.00% | 41.00% |
| | SafeDecoding | 87.00% | 85.00% |
| | DRO | 93.80% | 88.00% |
| | **FMM** | 92.00% | 89.50% |

Table 2: Multiple defense methods' responses to Alpaca instructions.

cally, for both LLaMA and Qwen models used in our experiments, we identified the 15th layer as the optimal choice for detection, while layers 12 to 15 were selected for intervention.

## Experimental Results

**Malicious Outputs are Mitigated** Table 1 compares response and risk rates for baselines and **FMM** across various attacks. **FMM** consistently triggers rejection responses, significantly reducing the risk rate. While response rate indicates how often a refusal occurs, risk rate captures the proportion of generated content that remains harmful—even with initial rejections. We observed instances where risk rate slightly exceeds response rate due to models generating some malicious content after an initial refusal. Compared to SafeDecoding and DRO, **FMM** achieves comparable performance in mitigating malicious outputs without requiring computationally expensive fine-tuning.

**Benign Outputs are not Affected** Table 2 shows the response and win rates of baselines and **FMM** on the Alpaca dataset. **FMM**, like most baselines, maintains response rates and overall response quality for benign queries. Notably, some methods show higher win rates than undefended models. This is likely because models, without any defense method, sometimes include content segments that are incorrectly flagged as potentially harmful. By skipping these flagged segments, models can then generate better quality responses. This suggests that aligned models may exhibit some implicit refusal behavior—even in normal inputs—which our method can also mitigate.

## Ablation Results

To evaluate the robustness of **FMM**, we conducted comprehensive ablation experiments by varying key parameters. Specifically, we investigated three aspects: (1) the impact of the training dataset size on detector optimization, (2) the choice of LLM layers for intervention, and (3) the effect of token position on intervention. Our primary ablation analyses were conducted using LLaMA 2.

**Training Samples for the Detector** We trained our classifier using a default of 150 benign and 150 malicious samples. As shown in Table 3, we observed that reducing the number of training samples does not significantly impact **FMM**'s ability to reject malicious queries. This is likely because the features of benign and malicious responses are relatively distinct, as visualized in Figure 1, which allows the detector to learn the decision boundary effectively with only a few samples.

| **Samples** | 30 | 60 | 90 | 120 | 150 |
|---|---|---|---|---|---|
| Reply Rate | 29.4% | 24.6% | 25.5% | 22.1% | 21.0% |
| Unsafe Rate | 28.8% | 23.0% | 23.8% | 21.1% | 18.0% |

Table 3: Results on $\text{AdvBench}_{\text{Proxy}}$.

**Layers to Steer** We evaluated the impact of adding rejection intervention at different layers of the model, with results presented in Figure 3. The figure shows that intervention results across different layers do not exhibit a clear pattern and fluctuate within a relatively stable range. Given that abstract concepts are typically formed and expressed in the middle layers, we selected these layers for default intervention.

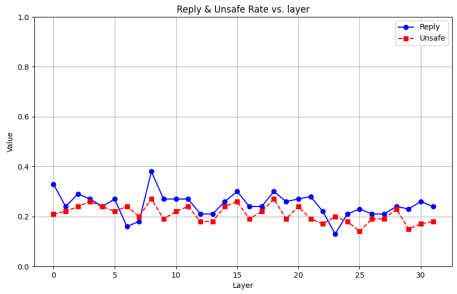


Figure 3: Varying the layers to steer.

**Token Positions for Intervention** We investigated whether intervening solely at the first token of a malicious response would suffice to trigger refusal. However, our experiments revealed that steering only the first malicious token did not significantly affect the response or risk rates. Consequently, effective intervention necessitates detecting and steering every token during the decoding phase.

## Conclusion

We introduced **FMM**, an two-stage decoding-oriented method for detecting and mitigating harmful responses. **FMM** first operates by monitoring the feature space to detect malicious token at each generation step, once the token is determined

to be harmful, **FMM** intervenes inner features to induce the model to refuse. The core of the work lies in leveraging and enhancing the inherent feature extraction and rejection capabilities acquired during pre-training and alignment. This method effectively defends against multiple jailbreak attacks while preserving the model's ability to respond to general queries. Ablation experiments confirmed the robustness of **FMM** across varied parameter settings. We believe that this token-level feature detection and intervention paradigm offers a promising direction for enhancing the safety of large language models.

## References


Alon, G.; and Kamfonas, M. 2023. Detecting language model attacks with perplexity. *arXiv preprint arXiv:2308.14132*.

Arditi, A.; Obeso, O.; Syed, A.; Paleka, D.; Rimsky, N.; Gurnee, W.; and Nanda, N. 2024. Refusal in Language Models Is Mediated by a Single Direction. *arXiv preprint arXiv:2406.11717*.

Bai, Y.; Jones, A.; Ndousse, K.; Askell, A.; Chen, A.; DasSarma, N.; Drain, D.; Fort, S.; Ganguli, D.; Henighan, T.; Joseph, N.; Kadavath, S.; Kernion, J.; Conerly, T.; El-Showk, S.; Elhage, N.; Hatfield-Dodds, Z.; Hernandez, D.; Hume, T.; Johnston, S.; Kravec, S.; Lovitt, L.; Nanda, N.; Olsson, C.; Amodei, D.; Brown, T.; Clark, J.; McCandlish, S.; Olah, C.; Mann, B.; and Kaplan, J. 2022. Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv:2204.05862.

Chao, P.; Robey, A.; Dobriban, E.; Hassani, H.; Pappas, G. J.; and Wong, E. 2023. Jailbreaking black box large language models in twenty queries. *arXiv preprint arXiv:2310.08419*.

Grattafiori, A.; Dubey, A.; Jauhri, A.; Pandey, A.; Kadian, A.; Al-Dahle, A.; Letman, A.; Mathur, A.; Schelten, A.; Vaughan, A.; and et al. 2024. The Llama 3 Herd of Models. *arXiv e-prints*, arXiv:2407.21783.

Inan, H.; Upasani, K.; Chi, J.; Rungta, R.; Iyer, K.; Mao, Y.; Tontchev, M.; Hu, Q.; Fuller, B.; Testuggine, D.; et al. 2023. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*.

Jain, N.; Schwarzschild, A.; Wen, Y.; Somepalli, G.; Kirchenbauer, J.; Chiang, P.-y.; Goldblum, M.; Saha, A.; Geiping, J.; and Goldstein, T. 2023. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*.

Lee, A.; Bai, X.; Pres, I.; Wattenberg, M.; Kummerfeld, J. K.; and Mihalcea, R. 2024. A mechanistic understanding of alignment algorithms: A case study on dpo and toxicity. *arXiv preprint arXiv:2401.01967*.

Li, X.; Zhang, T.; Dubois, Y.; Taori, R.; Gulrajani, I.; Guestrin, C.; Liang, P.; and Hashimoto, T. B. 2023. Alpacaeval: An automatic evaluator of instruction-following models.

Liu, X.; Xu, N.; Chen, M.; and Xiao, C. 2023a. Autodan: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*.

Liu, Y.; Deng, G.; Xu, Z.; Li, Y.; Zheng, Y.; Zhang, Y.; Zhao, L.; Zhang, T.; Wang, K.; and Liu, Y. 2023b. Jailbreaking chatgpt via prompt engineering: An empirical study. *arXiv preprint arXiv:2305.13860*.

Ouyang, L.; Wu, J.; Jiang, X.; Almeida, D.; Wainwright, C. L.; Mishkin, P.; Zhang, C.; Agarwal, S.; Slama, K.; Ray, A.; Schulman, J.; Hilton, J.; Kelton, F.; Miller, L.; Simens, M.; Askell, A.; Welinder, P.; Christiano, P.; Leike, J.; and Lowe, R. 2022. Training language models to follow instructions with human feedback. arXiv:2203.02155.

Phute, M.; Helbling, A.; Hull, M.; Peng, S.; Szyller, S.; Cornelius, C.; and Chau, D. H. 2023. Llm self defense: By self examination, llms know they are being tricked. *arXiv preprint arXiv:2308.07308*.

Rafailov, R.; Sharma, A.; Mitchell, E.; Ermon, S.; Manning, C. D.; and Finn, C. 2024. Direct Preference Optimization: Your Language Model is Secretly a Reward Model. arXiv:2305.18290.

Touvron, H.; Martin, L.; Stone, K.; Albert, P.; Almahairi, A.; Babaei, Y.; Bashlykov, N.; Batra, S.; Bhargava, P.; Bhosale, S.; et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Wei, A.; Haghtalab, N.; and Steinhardt, J. 2023. Jailbroken: How Does LLM Safety Training Fail? arXiv:2307.02483.

Weidinger, L.; Mellor, J.; Rauh, M.; Griffin, C.; Uesato, J.; Huang, P.-S.; Cheng, M.; Glaese, M.; Balle, B.; Kasirzadeh, A.; et al. 2021. Ethical and social risks of harm from language models. *arXiv preprint arXiv:2112.04359*.

Wu, J.; Gan, W.; Chen, Z.; Wan, S.; and Philip, S. Y. 2023. Multimodal large language models: A survey. In *2023 IEEE International Conference on Big Data (BigData)*, 2247–2256. IEEE.

Xu, Z.; Jiang, F.; Niu, L.; Jia, J.; Lin, B. Y.; and Poovendran, R. 2024. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. *arXiv preprint arXiv:2402.08983*.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; Dong, G.; Wei, H.; Lin, H.; Tang, J.; Wang, J.; Yang, J.; Tu, J.; Zhang, J.; Ma, J.; Yang, J.; Xu, J.; Zhou, J.; Bai, J.; He, J.; Lin, J.; Dang, K.; Lu, K.; Chen, K.; Yang, K.; Li, M.; Xue, M.; Ni, N.; Zhang, P.; Wang, P.; Peng, R.; Men, R.; Gao, R.; Lin, R.; Wang, S.; Bai, S.; Tan, S.; Zhu, T.; Li, T.; Liu, T.; Ge, W.; Deng, X.; Zhou, X.; Ren, X.; Zhang, X.; Wei, X.; Ren, X.; Liu, X.; Fan, Y.; Yao, Y.; Zhang, Y.; Wan, Y.; Chu, Y.; Liu, Y.; Cui, Z.; Zhang, Z.; Guo, Z.; and Fan, Z. 2024a. Qwen2 Technical Report. arXiv:2407.10671.

Yang, A.; Yang, B.; Hui, B.; Zheng, B.; Yu, B.; Zhou, C.; Li, C.; Li, C.; Liu, D.; Huang, F.; et al. 2024b. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*.

Zheng, C.; Yin, F.; Zhou, H.; Meng, F.; Zhou, J.; Chang, K.-W.; Huang, M.; and Peng, N. 2024. On Prompt-Driven Safeguarding for Large Language Models. arXiv:2401.18018.

Zhou, Z.; Yu, H.; Zhang, X.; Xu, R.; Huang, F.; and Li, Y. 2024. How Alignment and Jailbreak Work: Explain LLM Safety through Intermediate Hidden States. *arXiv preprint arXiv:2406.05644*.

Zou, A.; Wang, Z.; Carlini, N.; Nasr, M.; Kolter, J. Z.; and Fredrikson, M. 2023. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*.