Proceedings of the 7th International Conference on Signal Processing and Machine Learning
DOI: 10.54254/2755-2721/2027.39284

Instability of Safety Behavior in Large Language Models: Obfuscated Jailbreaking and Benign Fine-Tuning Degradation

Shuxian Hou[1], Yihong Yao[2,*]

[1]*Department of Artificial Intelligence and Computer Science, Shanghai Lixin University of Accounting and Finance, Shanghai, China*
[2]*Wharton International School, Tianjin, China*
**Corresponding Author. Email: yaoyh2027@outlook.com*

Abstract. With the development of security research on large language models, security issues have gradually expanded from explicit jailbreaking to security degradation caused by obfuscated inputs and benign fine-tuning. The former conceals harmful intentions through encoding, obfuscation, and other methods, while the latter weakens the model's original refusal ability when only fine-tuning with benign data. Although these two types of issues act on the input side and parameter side respectively, they both lead to instability in the model's security behavior. This paper analyzes these two types of issues from the perspective of security behavior stability: first, it provides formal definitions, sorts out representative attacks, evaluations, and defense methods; then, it compares the security failure characteristics under two types of perturbations: changes in input forms and updates to model parameters; further, it discusses the trade-off between security and usability, and points out the deficiencies of existing evaluations in terms of cause differentiation and coverage. The analysis shows that the ability of existing methods to maintain security stability under input changes and parameter updates is still limited, and a unified evaluation and defense mechanism is not yet perfect. Future research can be conducted in three directions: semantic-level security judgment, security-preserving fine-tuning, and phased evaluation.

Keywords: Large model security, Obfuscation-based jailbreaking, Benign fine-tuning security degradation, Security behavior stability, Security and usability tradeoff

1. Introduction

In many applications, including search, question answering, coding, education, and various other domains, the usage of large language models (LLMs) requires the model to adhere to user instructions while avoiding harmful assistance. This has turned safety alignment from a nice-to-have to a must-have. Initially, jailbreak research concentrated on adversarial prompts – ones that were intentionally unsafe, included additional context, or employed prompt engineering techniques designed to overcome refusal. The latter can be quite easily described in the setting where it occurs,

© 2026 The Authors. This is an open access article distributed under the terms of the Creative Commons Attribution License 4.0 (https://creativecommons.org/licenses/by/4.0/).

DOI: 10.54254/2755-2721/2027.39284

as the harmful intent is easy to see in the very natural language that many safety mechanisms are trained on. But recent research indicates that this is too limited a view. Yet even if the harmful meaning is expressed in another form, a model can continue to fail, and this can be done by downstream adaptation even in cases where the new training examples are harmless. These results indicate that it is not sufficient to simply check if a model rejects obvious harmful prompts to determine its safety.

There are two emerging problems that are particularly relevant. The first one is jailbreak behavior that is obfuscated. Harmful requests may be coded in one of the classical ciphers, code-completion structures, formatting patterns, or other transformations that don't change the meaning of the instructions but rather how they are presented. Encrypted inputs must be safety-evaluated to determine if a model is able to interpret the transformed content or not, and its safety after interpretation, as shown by Maskey, Dras and Naseem [1]. CodeAttack also reveals that translating natural language requests into code inputs can reveal safety-generalization failures [2] and superficial style alignment reveals that superficially irrelevant style patterns can affect jailbreak vulnerabilities [3]. The second issue is when people fine-tune. Safety-aligned models can become less reliable if the data to which they are adapted are not in themselves harmful, but only after downstream optimization. The representation similarity, overfitting, and jailbreak susceptibility studies show that the parameter changes can impact safety-relevant behaviour, which is not simply a matter of harmful versus benign examples in the fine-tuning data set [4-6].

In this review, these two lines of research have been combined in one analytical framework. One alters the input, while the other alters the model parameters; both are consistent with a more general weakness: safety behavior can become associated with distributions and patterns encountered during alignment and not persist when the same harmful intent is presented in another form or following downstream adaptations. Two stability requirements are thus used as organizing principles for the review. Firstly, if the same semantic intent is expressed in different ways, it's important that the safety decision is the same. Second, there should be no change in safety behaviour following benign fine-tuning downstream. These requirements are analytical tools that are suggested in this review and not benchmark definitions. The paper investigates the violation of these requirements in recent work, the evaluation of refusal and over-refusal in current benchmarks and how methods like safety-preserving low-rank adaptation try to mitigate degradation. The emphasis is then on the safety-utility trade-off. A good system should deny requests that pose risk without blocking requests that are not malicious because they involve an unknown representation and it should do so when it has been adapted. This framing also helps to explain why better evaluation is needed – a rejection can mean proper safety decision-making, it could mean a failure to read an input, or it could mean an interfering downstream guardrail, and these are not all the same.

2. Methods for obfuscated jailbreaks and benign fine-tuning safety degradation

One good way to relate obfuscated jailbreaks and benign fine-tuning degradation is to consider the notion of safety as a behavior which should be stable under two types of change. The first is that of form invariance. If two inputs are semantically equivalent with regard to the intent behind them, then changing the representation shouldn't change the model's decision on whether the request was safe or unsafe. The second is "parameter invariance". The adaptation should preserve the safety behavior that was originally present prior to fine-tuning if a safety-aligned model is adapted with benign data for a downstream task. Obfuscated jailbreaks violate the first requirement because the semantic request is preserved while the surface form has been altered. Benign fine-tuning degradation violates the second requirement because the prompt may not change, but the model may

DOI: 10.54254/2755-2721/2027.39284

change its behavior when interacting with the prompt. This distinction is important to maintain the original dichotomy between input-side and model-side failures and to make the analysis of both types of failure possible.

Recently, the literature can be characterized as three stages that overlap with each other. The initial phase, circa 2024, was mostly a phase of discovery of phenomena. The researchers showed that safety mechanisms trained primarily on ordinary natural language prompts do not necessarily generalize to different input domains. One such example is CodeAttack. It transforms a dangerous request into code-completion inputs and checks if the model also maintains safety constraints [2]. The study finds that guardrail bypass rates can exceed 80% for all the models evaluated and that the more the distribution gap between natural language and code, the less safety generalization there is. This attention was then shifted from the question of whether or not a model refused to the question of whether that refusal generalized across representations.

The second stage, starting in 2025, focused on shifting from observation to mechanism, particularly on the parameter-side degradation of safety. Hsiung et al. [4] demonstrate downstream fine-tuning through representation similarity, revealing that high similarity between the upstream data used for alignment and the downstream data used for fine-tuning can have an important impact on weakening safety guardrails. Xie, Song, and Luo show that just 10 benign question-answer pairs can be employed in an overfitting-based fine-tuning to dampen refusal behavior [5]. Murphy et al. demonstrate more generally that jailbreak susceptibility can be efficiently taught by fine-tuning and that stronger jailbreak prompts can become even more effective after such adaptation [6]. While the threat models vary among these studies, they all demonstrate the need to do more than screen a fine-tuning data set for easy-to-avoid harmful content to ensure that pre-existing safety behavior is preserved.

In the third stage (2025 through 2026), the focus is shifted towards evaluation and defence. The evaluation of encrypted text by Maskey, Dras, and Naseem is two-dimensional [1] and distinguishes between behavior indicating refusal of instructions and safety while generating responses. Wang et al. suggest a multidimensional approach to the evaluation of security, efficiency and utility at the guardrail level [7]. XSTest uses safe prompts and unsafe contrast prompts [8] to identify exaggerated safety behavior, while OR-Bench is an extension of over-refusal evaluation performed at a much larger scale with 80,000 over-refusal prompts, a hard subset and toxic prompts to identify indiscriminate compliance [9]. On the defense side, Xiao et al. suggest SafeStyle, which adds the safety data to make the style distribution closer to that observed during fine-tuning [3]; SaLoRA adopts a fixed safety module and task-specific low-rank adaptation design to ensure the safety alignment during the parameter-efficient fine-tuning [10]. These works demonstrate that the field is moving away from the building of attacks towards measuring the quality, cost and coverage of defense.

The obfuscation literature provides examples of a transformed prompt and why understanding the transformed prompt does not equal safe behavior. Maskey et al. investigate encrypted text and disentangle early refusal of instruction from generation safety, enabling assessment to differentiate if a model is safe with respect to the input instruction and if harmful content is suppressed during the generation process [1]. This separation is important because a poor decryption can lead to a similar outcome as a successful safety enforcement. On the other hand, a model that is able to recover an encrypted query, yet generates harmful content, shows capability without safe generalization ability. A similar problem is discovered in the code domain [2] by CodeAttack. It postulates that code-trained behavior can be used to give preference to code completion over safety constraints in case harmful intent is hidden in code structures. The point is further demonstrated with style-based work,

DOI: 10.54254/2755-2721/2027.39284

which shows that semantically irrelevant style patterns can affect the success of the attacks, and that fine-tuning on specific styles can make a model more vulnerable to matching jailbreak styles [3]. These findings suggest that safety should follow the intention that is being recovered and should not be correlated to the symbolic representation that was chosen to describe the intention.

In the literature that accompanies the parameters, there are three types of evidence that should not be confused. Hsiung et al. investigate ordinary downstream fine-tuning, and find that a key risk factor for the degradation of guardrails is representation similarity between safety-alignment data and downstream data [4]. The clearest benign-only attack in this series of works uses ten benign question-answer pairs in a two-stage process, first to overfit a refusal pattern and then to encourage broader compliance with standard benign answers [5]. Murphy et al. consider a more general fine-tuning for attacking, and demonstrate that models can be efficiently fine-tuned to learn the susceptibility to jailbreaking; this is complementary evidence of the fragility of safety alignment, not evidence that every such attack uses only benign data [6]. When considered as a whole, these studies reveal that fine-tuning risk relies more on the optimization process and the interplay between the process and the current safety behavior, rather than just the apparent appearance of individual training cases.

Thus, the defenses are at multiple levels. A safety tuning evaluation test, like XSTest and OR-Bench, has been used to determine if safety tuning has made it too aggressive and to the detriment of normal utility [8, 9]. Guardrail studies make comparisons on a number of dimensions between external filtering and monitoring strategies [7]. To handle a more constrained training environment, SaLoRA makes use of low-rank adaptation in a way that is less likely to affect features related to safety for task learning [10]. On the input side, style-aware safety training can make superficial style alignment less prone to error [3] and encrypted-input evaluation focuses on safety assessment following semantic recovery instead of raw-text evaluation [1]. The relationship between the attack, evaluation and defence is important, because no single method of failure can encompass all of these failure modes.

Table 1. Representative methods under the unified stability framework

Method category	Location	Main idea	Main limitation
Encoded or cipher obfuscation	Input	Hide harmful intent in encrypted representations and evaluate interpretation separately from generation safety [1].	Failure may reflect inability to decode rather than successful safety judgment.
Code-structure obfuscation	Input	Transform natural-language requests into code-completion structures [2].	Results depend on the code domain and do not by themselves establish robustness to other representations.
Style and formatting shifts	Input	Use semantically irrelevant style patterns that can increase jailbreak success [3].	Coverage of unseen or combined styles remains difficult.
Representation-similarity degradation	Model	Analyze how similarity between alignment and fine-tuning data relates to safety loss [4].	Requires representation-level analysis and does not provide a universal trigger threshold.
Overfitting-based benign fine-tuning	Model	Use a small benign dataset in a two-stage process to weaken refusal behavior [5].	Demonstrates vulnerability but does not yield a simple deployment-time predictor.

DOI: 10.54254/2755-2721/2027.39284

Table 1. (continued)

Jailbreak-susceptibility fine-tuning	Model	Show that models can efficiently learn stronger jailbreak susceptibility through fine-tuning [6].	Primarily characterizes attackability; tamper-resistant defenses remain an open problem.
Over-refusal and guardrail evaluation	Evaluation	Measure refusal of safe prompts and compare security, efficiency, and utility [7-9].	Most benchmark inputs remain explicit rather than deeply obfuscated.
Safety-preserving low-rank adaptation	Defense/fine-tuning	Use a fixed safety module with task-specific low-rank adaptation to preserve alignment [10].	Protection is tied to the adaptation design and does not address every input-side attack.

It is evident from Table 1 that there is a gap between the production of failures and the majority of the present evaluation. Input-side attacks involve targeted manipulation of the representation of malicious intent, while over-refusal benchmarks like XSTest and OR-Bench primarily focus on how well explicit natural-language prompts are handled [8, 9]. The model-side defenses, including SaLoRA, reduce safety degradation for a specific fine-tuning design [10]; however, they do not establish robustness to encrypted/coded or stylistically shifted inputs. The primary methodological issue is thus coverage – a defense can seem strong if it is judged solely based on the representation or training context that it was built for. A more robust evaluation protocol would evaluate the two dimensions simultaneously as follows: Test the model with two different forms of a semantically matched prompt, for instance: present semantically matched prompts in plain and transformed form; test the model before and after fine-tuning. This would make it possible to differentiate a true stable safety policy and a safety policy that only works under known surface conditions.

3. Contradictions between security and availability

The core issue in the research on obfuscation based jailbreaking and benign fine-tuning of security degradation is how to improve the ability to intercept harmful requests while maintaining the model's responsiveness to normal requests. Confusing input can alter the surface form of harmful content, making it difficult for guardrails to directly identify its risks. Benign fine-tuning may alter the original safety parameters and reduce the model's refusal ability. Although these two types of problems occur at different stages, they both affect the model's ability to distinguish between harmful and benign requests, resulting in a trade-off between security and availability. This trade-off runs through three layers: attack, evaluation, and defense.

The existing methods mainly alleviate this problem through semantic recovery, excessive rejection evaluation, and security maintenance fine-tuning. D1/D2 two-dimensional evaluation is used to distinguish whether the model can truly understand confusing content and make security judgments [1], XSTest and OR Bench [8, 9] are used to evaluate the model's excessive rejection of normal requests, and SaLoRA [10] is used to reduce security capability degradation by limiting the changes in security related parameters during fine-tuning. But these methods mainly target plaintext input, single obfuscation forms, or specific fine-tuning processes. When the input is encoded, structured, or combined in multiple formats, the model may experience failure states such as being able to recover semantics but not continuing to perform security judgments, directly rejecting without recovering semantics, or mistakenly rejecting normal requests due to strong defense. Therefore, existing methods still struggle to maintain high levels of harmful request interception capability and normal task completion capability in complex and confusing scenarios.

DOI: 10.54254/2755-2721/2027.39284

This trade-off can be quantified from two indicators: harmful inhibition rate and benign completion rate. The harmful suppression rate HSR represents the proportion of harmful samples correctly rejected by the model to all harmful samples, which can reflect the model's ability to intercept harmful requests; The benign completion rate BCR represents the proportion of benign samples that the model completes normally to all benign samples, and is used to reflect the availability of normal tasks. An increase in HSR indicates more effective control of harmful requests, while an increase in BCR reflects fewer false rejections of normal requests. In an ideal situation, both indicators should be maintained at a high level, but in confusing input scenarios, there is a clear trade-off between them, which is also a problem that existing defense methods need to further address. However, the determination of formal invariance also depends on the true semantics obtained by decoding the input, as this semantics is unobservable in real deployment. Therefore, HSR and BCR can only be used as proxy indicators to record behavioral results without recording behavioral causes, which is the reason for the evaluation dilemma.

By conducting comparative experiments on encrypted inputs, and can intuitively observe the actual performance of this trade-off in various defense schemes. Three sets of differential indicators have been proposed in encrypted text, among which Δresp corresponds to the suppression effect of harmful content, Δutility is used to measure the loss in usability, and ΔIR is the degree of excessive rejection of the model. These indicators can also reflect a trade-off relationship similar to HSR/BCR. In the experiment, the harmful inhibition of perplexity filtering was only 0.02, resulting in a high availability loss of 0.22 to 0.27, almost not blocking harmful content and interfering with the execution of normal tasks. The harmful inhibition of heavy participle defense is -0.16, with a usability loss of 0.14 to 0.17, further amplifying the behavior of harmful generation. These two types of defense sacrifice usability but do not yield the expected security benefits. The self checking mechanism and the control group of rear guardrails are exactly the opposite, with both able to improve harm suppression by 0.46 and 0.37, respectively. However, the indicators of excessive rejection caused by false rejection also increased to 0.42 and 0.44, with a large number of normal requests sacrificed. Based on the data of the tested defense types, it can be found that it is difficult to find a solution that can simultaneously achieve a high level of harm suppression while having low levels of excessive rejection and availability loss. This false rejection phenomenon has a certain scale. The OR Bench dataset contains 80000 prompts covering ten common rejection scenarios, of which about a thousand samples still pose challenges to the current model [9], and XSTest's research has long focused on such exaggerated safety behaviors [8]. Based on these data, transform the trade-off between security and availability from textual descriptions to research propositions that can be quantitatively verified.

Due to the flexibility of attackers to choose various encoding transformations, defense schemes in plaintext environments are difficult to directly apply to obfuscated input scenarios. To cope with this situation, multiple approaches have been developed on the defensive side. Part of the processing logic proposes to directly intercept all input in unfamiliar formats. Although this can improve HSR, BCR will decrease accordingly, and even compliant content encrypted and pasted by ordinary users will be judged as an attack. There are also ideas that in order to preserve the performance of BCR, it is necessary to wait for the model to decode before making security judgments. But in reality, the effectiveness of this idea is usually unstable. Previous studies have found that some models with decryption capabilities may skip performing security judgments after restoring semantics [1] and still release harmful applications, a phenomenon known as mismatch generalization. In order to avoid the problems caused by relying on the main model, some solutions choose to directly identify confusing features from the ciphertext or hand over the decoded content to independent guardrails

DOI: 10.54254/2755-2721/2027.39284

for judgment. However, this solution relies on additional components and lacks systematic evaluation of its overall generalization ability when faced with encoding forms that have not been seen in training.

In addition to a single encoding, attackers can also stack and use different encoding methods, further increasing the difficulty of defense. According to the combination calculation, if a maximum of k basic encodings are selected for combination among n types, the upper limit of the number of all combination forms can reach 2 to the power of n. A total of 9 typical encoding types have been identified in the relevant work [1], including three levels of simple, medium, and high difficulty. By combining them, 511 different forms of confusion can be generated. The search space in the face of random character mapping attacks tends to expand in a combinatorial manner with increasing complexity, making exhaustive filtering impractical in real-world scenarios [1]. Relying on manual collection of various new coding samples for guardrail training is difficult to catch up with the combination methods of attackers' rapid iteration.

After upgrading the defense plan, it is not possible to rely on evaluation indicators to determine the actual effectiveness of the defense plan. For the calculation of differential indicators such as EM and BLEU, the original plaintext before encryption is required as a reference, but in actual working environments, the system can only obtain encrypted input, so these indicators are only applicable to offline test sets. Most of the evaluation work is completed in offline environments, and offline use will not cause defects. The real problem is the insufficient ability to distinguish indicators. The reason for sample rejection is not unique, it may be due to the model accurately identifying harmful intentions and performing reasonable security judgments, or it may be due to the model not completing decoding, or it may be due to downstream modules intercepting harmful content already generated by the model. All of the above situations may ultimately result in an EM value of 0, which makes it difficult for security operators to pinpoint where the problem lies. For example, replacing dangerous keywords with harmless words with similar pronunciations can achieve a symbol similarity of 0.95, and EM can be directly reset to zero, but BLEU still has a high score [1]. This reflects the limitation of precise matching metrics that cannot distinguish between local decryption and complete rejection [1]. At present, most guardrail evaluations are based on overall scores, and there is no unified standard to distinguish various failure causes [7].

The coverage of the evaluation is also negatively constrained by the decoding ability of the model. Under medium and high difficulty encoding, the model itself cannot be solved, and the corresponding samples are directly excluded from security evaluation [1]. Therefore, security analysis on medium and difficult encoding is often difficult to carry out, and related evaluations are limited to four easy to solve encodings [1]. This type of load can also pass through filtering stably at the input end. Confusion filtering cannot distinguish the benign and harmful nature of encrypted text, and the front guardrail cannot recognize the harmful nature of it. What happens after passing but no one measures it, the inability to determine indicators does not mean that the model does not understand it, and the fact that most of the load is solved is enough for the model to grasp harmful intentions [1]. The inability to read the guardrail does not mean that the model cannot read it.

The load of difficult to decipher code falls between two blind spots, so they are excluded from the evaluation. The input guardrail ignores them, and the actual understanding level of the main model is not measured by anyone. Reading 'undetectable' as' secure 'lacks basis, and reading it as' unsafe' also lacks basis. The accurate expression is that the evaluation system has a systematic blind spot in this range, and attackers choose difficult to decode payloads to benefit from this blind spot. The lack of quantitative characterization of the prevalence and scale of situations where models do not explicitly complete decoding but still exhibit harmful responses to obfuscated payloads.

DOI: 10.54254/2755-2721/2027.39284

In plaintext scenarios, security performance can be evaluated from two dimensions: harmful request interception and benign request response. In confusing input scenarios, the trade-off between the two is still evident. The improvement of interception capability will increase benign request false rejection, while reducing false rejection will weaken the recognition of hidden harmful content. Evaluation indicators cannot distinguish semantic recovery failure, active security rejection, and system level interception, making it difficult to accurately reflect the actual effectiveness of defense methods. The follow-up work needs to address two things simultaneously. One is to improve the ability to identify confusing harmful requests while controlling benign false negatives, and the other is to establish an evaluation system that can distinguish the causes of failure. The latter is the verification condition for the former, and if the cause is unclear, the effectiveness of the improvement cannot be judged.

4. Conclusion

This article uses formal invariance and parameter invariance as analytical tools to unify confusion class jailbreaking and benign fine-tuning security degradation into two types of perturbations caused by the same security binding error on input form and model parameters, and to sort out related methods. The common feature of both is that security judgments are bound to the distribution and patterns encountered during the alignment process rather than harmful semantics themselves. Once the input form or parameters deviate from this distribution, security behavior becomes difficult to sustain. This type of security instability also poses significant challenges to effectiveness evaluation. However, the existing evaluation system still has limitations in testing scenarios and identifying reasons for failure, resulting in a lack of reliable metrics for the improvement effect of defense schemes in confusing input scenarios.

There are multiple feasible exploration directions for future research. Researchers can establish a semantic level security judgment mechanism to ensure that the verification of the model can be stably triggered after completing semantic restoration. In the design of evaluation methods, it is possible to deviate from plain text references. At the level of evaluation statistics, this paper abandon the single score evaluation model and instead classify and count various types of results. On this basis, an evaluation protocol covering input forms and parameter changes can be further built to conduct paired tests on plaintext and converted text to collect measurement data before and after model fine-tuning. These explorations simultaneously consider the interception of harmful requests and the ability to respond to normal requests, establishing a unified and comparable evaluation benchmark for both types of failures.

Author contribution

All the authors contributed equally and the names were listed in alphabetical order.

References

[1] Maskey, U., Dras, M., & Naseem, U. (2026). Should LLM safety be more than refusing harmful instructions? *Machine Learning, 115*(8), 1-28.

[2] Ren, Q., Gao, C., Shao, J., Yan, J., Tan, X., Lam, W., & Ma, L. (2024). CodeAttack: Revealing safety generalization challenges of large language models via code completion. *Findings of the Association for Computational Linguistics: ACL 2024*, 11437-11452.

[3] Xiao, Y., Tonekaboni, S., Gerych, W., Suriyakumar, V., & Ghassemi, M. (2026). When style breaks safety: Defending LLMs against superficial style alignment. *International Conference on Learning Representations (ICLR)*.

DOI: 10.54254/2755-2721/2027.39284

[4] Hsiung, L., Pang, T., Tang, Y. C., Song, L., Ho, T. Y., Chen, P. Y., & Yang, Y. (2026). Why LLM safety guardrails collapse after fine-tuning: A similarity analysis between alignment and fine-tuning datasets. *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 16601-16619.

[5] Xie, Z., Song, X., & Luo, J. (2025). Attack via overfitting: 10-shot benign fine-tuning to jailbreak LLMs. *Advances in Neural Information Processing Systems 38 (NeurIPS 2025)*.

[6] Murphy, B., Bowen, D., Mohammadzadeh, S., Tseng, T., Broomfield, J., Gleave, A., & Pelrine, K. (2025). Jailbreak-tuning: Models efficiently learn jailbreak susceptibility. *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, 13217-13246.

[7] Wang, X., Ji, Z., Wang, W., Li, Z., Wu, D., & Wang, S. (2026). SoK: Evaluating jailbreak guardrails for large language models. *2026 IEEE Symposium on Security and Privacy (SP)*, 39-58.

[8] Röttger, P., Kirk, H. R., Vidgen, B., Attanasio, G., Bianchi, F., & Hovy, D. (2024). XSTest: A test suite for identifying exaggerated safety behaviours in large language models. *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, 5377-5400.

[9] Cui, J., Chiang, W. L., Stoica, I., & Hsieh, C. J. (2025). OR-Bench: An over-refusal benchmark for large language models. *Proceedings of the 42nd International Conference on Machine Learning, 267*, 11515-11542.

[10] Li, M., Si, W. M., Backes, M., Zhang, Y., & Wang, Y. (2025). SaLoRA: Safety-alignment preserved low-rank adaptation. *International Conference on Learning Representations (ICLR)*.