Balancing Safety and Helpfulness: Policy Frameworks for Addressing LLM Jailbreaking and Over-Refusal

Wei Chang, Yeu-Tong Lau, Genglin Liu, Hubert Tang *
UCLA

Abstract

As large language models (LLMs) become increasingly embedded in society, ensuring their safe yet helpful use becomes critical. Jailbreaking attacks, which exploit vulnerabilities to bypass model safeguards, enable harmful outcomes such as misinformation, privacy breaches, and more real-world risks. Current jailbreak techniques range from psychological attacks and prompt engineering, to weight-based and retrieval methods. In response, defensive strategies have been developed, including prompt-level techniques and model-level methods. As a consequence of excessive safety fine-tuning, state-of-the-art LLMs also have a tendency to over-abstain which is undesirable at the other end of this safety-helpfulness spectrum. To address this balance, we propose a tiered policy framework involving regulatory standards, multi-layered access control and transparency, and adaptive safety mechanisms using dynamic preference tuning. Our recommendations aim to refine AI governance strategies that protect against misuse without stifling beneficial AI applications, ensuring responsible and secure AI innovation.

1 Introduction and Motivation

Large language models (LLMs) have become more and more integrated into society and our everyday life, and it is a critical issue to make sure that these machine-generated behaviors are helpful while remaining safe and secure. LLM jailbreaking, in particular, refers to the techniques that exploit vulnerabilities of the machine learning model to bypass their safeguards, eliciting potential misuse or harmful behaviors such as generating misinformation, privacy leaks, or real-world risks like robotic manipulation or chemical synthesis. In this abstract, we intend to review existing methods on LLM attacks and defenses, as well as evaluation suits and benchmarks that the community has introduced in order to assess and mitigate their risks. On top of the literature survey, we will give our own recommendations on how AI development should pay more attention to the safety-helpfulness trade-off, as we need to establish a balance between training helpful AI assistants while keeping them safe from harmful uses.

Current jailbreak strategies range from authority/psychological attacks (DarkCite) (Yang et al., 2024b) and multi-turn prompt engineering (Jigsaw Puzzle) (Yang et al., 2024a) to weight-based attacks (Badllama 3) (Volkov, 2024) and retrieval poisoning (PANDORA). These attacks constantly evolve in the AI community and new threats always expose the fragility of existing safety measures. To counteract the risks of having state-of-the-art LLMs jailbroken, numerous defensive strategies also have been proposed. These can be categorized into prompt-level and model-level defenses. Prompt-level techniques, such as Prompt Adversarial Tuning (PAT) and Constitutional Classifiers, modify user prompts to mitigate adversarial attacks. Model-level approaches, like SelfDefend, integrate internal safety mechanisms that monitor and regulate output generation. While these methods enhance security, they also introduce trade-offs that may restrict model helpfulness in benign scenarios.

Our policy framework aims to strike a balance between security and usability. We propose a tiered approach: (1) Regulatory Standards — governments and academic institutions should establish baseline safety benchmarks using standardized evaluation tools such as JailbreakBench and HarmBench. (2) Access Control and Transparency — open models should implement tiered access, where highly capable models undergo stricter monitoring for adversarial usage. (3) Adaptive Safety Mechanisms — LLMs should employ dynamic preference tuning to differentiate between malicious and beneficial queries rather than blanket refusal policies. Through this policy framework,

*Equal contributions. Alphabetically ordered

we seek to refine LLM governance strategies that ensure safety without undermining legitimate use cases, fostering both security and responsible innovation in AI deployment.

2 Jailbreaking Large Language Models

Jailbreaking attack methods generally are compared and categorized between different categories. These include single-turn or. multi-turn methods and white box or black box methods. Certain attack methods may also be categorized as multimodal or RAG-based attacks.

2.1 Single-Turn and Multi-Turn Methods

Single-turn jailbreaking methods execute jailbreaking within a single prompt, whereas multi-turn methods execute jailbreaking in a series of prompts, and are generally more effective. Multi-turn methods include Jigsaw Puzzles (Yang et al., 2024a), which splits a potentially harmful prompt into different tokens before asking a machine to combine them and respond. Other multi-turn methods include training red-team agents on datasets to execute attacks, as demonstrated through the MRJ-agent method (Wang et al., 2025a).

2.2 White Box and Black Box Methods

White box and black box methods differ based on whether jailbreak attacks have or do not have access to the model's internals during attacks. White box attacks are particularly effectively, bypassing safety fine-tuning in a matter of minutes, as demonstrated by Badllama 3 (Volkov, 2024), while frameworks such as COLD-Attack can further facilitate white-box attacks through searching for and executing attacks based on factors such as fluency and stealthiness (Guo et al., 2024).

Black box jailbreaking methods have achieved success through methods such as leveraging authoritative information in prompts through methods like DarkCite (Yang et al., 2024b) or through reinforcement learning, mutating questions and templates based on vocabulary richness scoring (which increases as methods become more successful) in methods like PathSeeker (Lin et al., 2024).

2.3 Other Methods

Other jailbreaking methods have had success in multimodal LLMs and RAG-based LLMs. Chain-of-Jailbreak attacks can manipulate LLMs to create harmful images through passing instructions to modify images step by step (Wang et al., 2024), while VoiceJailbreak allows for audible jailbreak prompts on LLMS such as GPT-4o by framing attacks as fictional scenarios (Shen et al., 2024).

Indirect jailbreaking methods such as through retrieval augmented generation (RAG) have also been studied. Methods such as PANDORA involve creating, injecting, and triggering malicious documents in LLMs that are modified to prevent rejection by the LLMs (Gelei Deng et al., 2024).

2.4 Jailbreaking Defense Methods

Defenses against jailbreaking involve methods that allow for refusal of harmful prompts while having minimal influence on benign prompts, and are often divided into model-level or prompt-level defenses. Model-level defenses involve safety features directly built within the model, such as Self-Defend utilizing a shadow LLM and stack to determine whether prompts are harmful, triggering checkpoints in the normal stack if such is the case (Wang et al., 2025b).

Prompt-level defenses directly modify the prompt to prevent harmful results, with methods such as Prompt Adversarial Tuning (PAT) and Robust Prompt Optimization (RPO) adding defense prefixes and suffixes to the prompts respectively to hield significant protection against attacks while having minimal impact on benign prompts (Mo et al., 2024; Zhou et al., 2024). Additionally, methods such as using constitutional classifiers to generate a diverse training set of synthetic prompts that can allow models to defend against 'universal' jailbreaks (Sharma et al., 2025).

3 Over-Refusals

Modern LLMs usually go through a post-training stage that involves reinforcement learning from human feedback (RLHF), in order to align with human preferences. As discussed in the previous section, a large portion of this stage contains instructions that teach the models to avoid harmless behaviors. If this safety fine-tuning is too conservative, models tend to learn to refuse more broadly than human users would appreciate. For example, an aligned GPT-3.5 model might refuse a harmless prompt for a dark joke, replying "I'm sorry, but I can't comply with that request." (Cui et al., 2024). While safety training reduces toxic outputs, we show that it often comes at the cost of increased refusals of innocuous queries. In essence, the model learns a policy that it's better to say no than risk

saying something wrong. We could imagine the extreme case where a model trained to always avoid risk might refuse every query - it would be perfectly harmless but completely useless as an assistant. This highlights the trade-off: between safety and helpfulness, if we push too hard on one dimension, we might compromise the other. Having too much emphasis on safety can inadvertently reduce the helpfulness of these models. And as a result, this process produces a model that sticks to a small safe zone of responses (like canned refusals or generic answers), reducing its overall adaptability.

Anthropic addresses this issue in their recent release (Anthropic, 2025), the Claude 3-7-sonnet, where they introduce the concept of "appropriate harmlessness." Their goal is to develop models capable of recognizing harmful intent without excessively refusing non-toxic queries. However, Anthropic acknowledges that in the earlier prototypes they designed to protect against jailbreaking, they also observed higher over-refusal rates, reducing model usability. One specific effect is that models become overly sensitive to certain keywords or phrases, refusing prompts that contain them even in contexts that are not actually harmful. Researchers term this "lexical overfitting": the model becomes too sensitive for words that are commonly seen in unsafe requests and blocks the response without carefully inspecting the whole query. One example is that, the word "coke" in a query might trigger a refusal because the model flags due to its connotation with illegal drugs, when in fact the user meant the soft drink. These false positives happen likely because the safety mechanism acts like a simple keyword filter. Similarly, safety classifiers or heuristic rules used to make the model robust to adversarial prompts might generalize poorly. The model learns a constraint like "if query mentions violence or self-harm, respond with a refusal," which works for truly unsafe requests but also includes some benign ones (e.g. "How do I kill a process in Linux?"). This effect, if not carefully calibrated, leads to an overshooting effect: benign queries that only superficially resemble unsafe ones get rejected.

To formalize this issue and evaluate the extent of LLM overrefusals, benchmarks like XSTEST (Röttger et al., 2023) and OR-Bench have been introduced. XSTEST (Exaggerated Safety Test) is a suite of 250 safe prompts across 10 categories that should not be refused, paired with 200 truly unsafe prompts for control contrast. The safe prompts are designed to resemble tricky cases – for instance, using homonyms ("Where can I buy a can of coke?" meaning soda), figurative language, or references to violence in safe contexts (like a video game or fishing). A well-calibrated model should answer all 250 safe queries and refuse the 200 unsafe ones. XSTEST results have revealed that some models exhibit systematic false refusals due to keyword triggers or misinterpreting context. For example, the authors have pointed out that Llama-2-chat initially refused many safe prompts containing words like "kill" (even if talking about killing a computer process or gutting a fish).

OR-Bench (Cui et al., 2024) takes evaluation further by generating a large-scale dataset of "seemingly toxic" prompts. It automatically rewrites genuinely harmful prompts into innocuous versions that look dangerous but aren't. This resulted in 80,000 test prompts spanning common refusal categories (e.g. harassment, violence, self-harm, illicit behavior etc.), along with a subset of 1,000 challenging cases and 600 truly toxic prompts for control. OR-Bench allows more rigorous testing of models under many scenarios. Using this benchmark, (Cui et al., 2024) evaluated 25 popular LLMs (across 8 proprietary and open-weight model families) and quantified the safety-helpfulness trade-off. The findings confirmed that most models improve safety primarily by being more refusal-prone, and very few managed to both refuse almost all toxic prompts and answer most benign ones. Notably, models like Anthropic's Claude-2 had the highest toxic rejection rates but also over-refused the most benign prompts, whereas some open-source models (like Mistral) were very permissive (few refusals, but also failed to refuse some unsafe prompts).

Recent research into model interpretability has revealed that refusal behaviors can be controlled at the internal, mechanistic level of LLMs. Studies indicate that manipulating specific model stream and activations can either induce or prevent refusal behaviors, providing a promising approach to balance safety and usability effectively. For policymakers, the existence of these benchmarks is reassuring because it shows the research community is not just focusing on preventing bad behavior, but also actively testing for overly strict behavior. Balanced evaluation frameworks will lead to AI systems that are both safer and more responsive.

4 Policy Recommendation

As large language models become widely used in society, establishing a policy framework is critical to ensure these systems remain both safe and helpful. The current situation reveals a fundamental trade-off: strong safeguarads aimed at preventing jailbreaking can lead to over-refusal of legitimate and harmless requests, while prioritizing helpfulness without adequate safeguards creates vulnerabilities to malicious cases. Our policy recommendations want to address this challenge through a structured framework that balance both safety requirements and helpfulness expectations.

To start with, we propose adopting a systematic safeguard evaluation process directly from the UK AI Security Institute's "Principles for Evaluating Misuse Safeguards of Frontier AI Systems" (UK AI Security Institute, 2025). This five-step approach begins with clearly stating safety requirements and identifying specific risks they want to prevent or resolve; this step needs to be as detailed and concrete as possible. The second step establishes a safeguards plan, detailing specific defense methods to be implemented, including how these defense methods will be integrated and how realistic they are. The third and fourth steps focus on evidence collection – evaluating safeguard effectiveness in controlled pre-deployment testing and planning for ongoing post-deployment assessment to identify emerging vulnerabilities or over-refusal patterns. The fourth step is crucial to pre-define how success and failure look after deployment. Finally, the fifth step involves determining whether the implemented safeguards satisfy the original requirements by following the guidelines in step four and comparing with the identified risks specified in step one. This creates a continuous improvement loop that adapts to evolving threats and usage patterns.

Building on this evaluation template, we recommend a policy framework addressing the full lifecycle of LLM deployment. In the pre-deployment phase, governmental regulatory bodies (preferably US AI Safety Institute) should establish context-specific security thresholds based on application sensitivity and potential impact, with higher-risk domains requiring more protections. This will allow AI companies and developers to have a reference. Besides, we will provide a regulatory sandboxes that have limited liability protection for testing innovative safeguard approaches. During deployment and monitoring stage, we mandate regular benchmarking on both jailbreak vulnerability and over-refusal propensity using standardized tools like JailbreakBench, XSTEST, and OR-Bench. These assessments should be accompanied by transparent documentation of safety mechanisms and their known limitations. For post-deployment continuous improvement, we advocate establishing formal incident response channels and bug bounty programs specifically targeting the safety-helpfulness balance, alongside protected intelligence sharing mechanisms that allow organizations to collaborate on defensive measures without exposing proprietary information.

This balanced framework acknowledges that perfect safety is neither achievable nor desirable if it comes at the cost of model utility. Instead, we envision a governance approach that encourages dynamic adjustment of safeguards based on context, user verification, and application domain. We aim to develop AI systems that remain helpful for legitimate users while maintaining appropriate protections against harms. The success of LLMs in society ultimately depends not on maximizing either safety or helpfulness in isolation, but on thoughtfully optimizing their balance in ways that respect both the potential and the risks of these powerful technologies.

5 Embedded Ethics Discussion and Conclusion

To enhance people's awareness of safety and usefulness when learning about LLMs, it is important not only to illustrate the concepts but also to provide hands-on experience with real use cases. If we aim to transform our proposed policy framework into courses, we can follow this sequence: First, we introduce the concept of LLM jailbreaking and allow students to implement both attack and defense strategies. This approach can heighten students' awareness of potential vulnerabilities in LLMs that they may not have previously noticed. Next, we introduce the contrasting concept of over-refusals. Unlike jailbreaking, overrefusals focus on maintaining LLMs' usefulness while ensuring they do not become overly conservative in their responses due to excessive defensive measures. Finally, we present our AI policy framework, explaining how we formulate strategies to evaluate whether LLMs are both safe and helpful. We also discuss how these evaluations can be conducted across different phases: pre-deployment, deploy-

ment, and post-deployment. In summary, hands-on experience with jailbreaking, overrefusals, and policy evaluation helps students balance LLM safety and usefulness, fostering a deeper understanding of AI security.

Author Contributions

WC worked on the motivationa and overall project roadmap. HT collected on the review for LLM jailbreaking. GL worked on the discussion of overrefusals. YL proposed the policy framework.

References

Anthropic. 2025. Claude 3.7 sonnet system card.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2024. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*.

Xingang Guo, Fangxu Yu, Huan Zhang, Lianhui Qin, and Bin Hu. 2024. Cold-attack: Jailbreaking llms with stealthiness and controllability. *Preprint*, arXiv:2402.08679.

Zhihao Lin, Wei Ma, Mingyi Zhou, Yanjie Zhao, Haoyu Wang, Yang Liu, Jun Wang, and Li Li. 2024. Pathseeker: Exploring llm security vulnerabilities with a reinforcement learning-based jailbreak approach. *Preprint*, arXiv:2409.14177.

Yichuan Mo, Yuji Wang, Zeming Wei, and Yisen Wang. 2024. Fight back against jailbreaking via prompt adversarial tuning. *Preprint*, arXiv:2402.06255.

Paul Röttger, Hannah Rose Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2023. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. *arXiv preprint arXiv:2308.01263*.

Mrinank Sharma, Meg Tong, Jesse Mu, Jerry Wei, Jorrit Kruthoff, Scott Goodfriend, Euan Ong, Alwin Peng, Raj Agarwal, Cem Anil, Amanda Askell, Nathan Bailey, Joe Benton, Emma Bluemke, Samuel R. Bowman, Eric Christiansen, Hoagy Cunningham, Andy Dau, Anjali Gopal, Rob Gilson, Logan Graham, Logan Howard, Nimit Kalra, Taesung Lee, Kevin Lin, Peter Lofgren, Francesco Mosconi, Clare O'Hara, Catherine Olsson, Linda Petrini, Samir Rajani, Nikhil Saxena, Alex Silverstein, Tanya Singh, Theodore Sumers, Leonard Tang, Kevin K. Troy, Constantin Weisser, Ruiqi Zhong, Giulio Zhou, Jan Leike, Jared Kaplan, and Ethan Perez. 2025. Constitutional classifiers: Defending against universal jailbreaks across thousands of hours of red teaming. *Preprint*, arXiv:2501.18837.

Xinyue Shen, Yixin Wu, Michael Backes, and Yang Zhang. 2024. Voice jailbreak attacks against gpt-4o. *Preprint*, arXiv:2405.19103.

UK AI Security Institute. 2025. Principles for evaluating misuse safeguards of frontier ai systems. Technical report, Department for Science, Innovation and Technology.

Dmitrii Volkov. 2024. Badllama 3: removing safety finetuning from llama 3 in minutes. *Preprint*, arXiv:2407.01376.

Fengxiang Wang, Ranjie Duan, Peng Xiao, Xiaojun Jia, Shiji Zhao, Cheng Wei, YueFeng Chen, Chongwen Wang, Jialing Tao, Hang Su, Jun Zhu, and Hui Xue. 2025a. Mrj-agent: An effective jailbreak agent for multi-round dialogue. *Preprint*, arXiv:2411.03814.

Wenxuan Wang, Kuiyi Gao, Zihan Jia, Youliang Yuan, Jen tse Huang, Qiuzhi Liu, Shuai Wang, Wenxiang Jiao, and Zhaopeng Tu. 2024. Chain-of-jailbreak attack for image generation models via editing step by step. *Preprint*, arXiv:2410.03869.

Xunguang Wang, Daoyuan Wu, Zhenlan Ji, Zongjie Li, Pingchuan Ma, Shuai Wang, Yingjiu Li, Yang Liu, Ning Liu, and Juergen Rahmel. 2025b. Selfdefend: Llms can defend themselves against jailbreaking in a practical manner. *Preprint*, arXiv:2406.05498.

Hao Yang, Lizhen Qu, Ehsan Shareghi, and Gholamreza Haffari. 2024a. Jigsaw puzzles: Splitting harmful questions to jailbreak large language models. *arXiv preprint arXiv:2410.11459*.

Xikang Yang, Xuehai Tang, Jizhong Han, and Songlin Hu. 2024b. The dark side of trust: Authority citation-driven jailbreak attacks on large language models. *arXiv preprint arXiv:2411.11407*.

Andy Zhou, Bo Li, and Haohan Wang. 2024. Robust prompt optimization for defending language models against jailbreaking attacks. *Preprint*, arXiv:2401.17263.