
Safety Targeted Embedding Exploit via Refinement: Mechanistic Interpretability as an Attack Vector

Joshua Adrian Cahyono
joshuaadrian@gmail.com

Abstract

Mechanistic interpretability research has shown that LLM safety fine-tuning encodes a refusal mechanism as a single geometric direction in the residual stream. We demonstrate that this finding is directly exploitable: an attacker with white-box access can use the model’s own internal structure to surgically bypass its safety filter. **STEER** (Safety Targeted Embedding Exploit via Refinement) uses gradient attribution against the identified refusal direction to select which words in a harmful prompt most activate the safety filter, then iteratively replaces them with low-resource language translations, collapsing the refusal signal while preserving harmful intent. Across six open-source 8B-parameter models, STEER achieves attack success rates (ASR) of up to **93%** on JailbreakBench and **96.7%** on AdvBench, outperforming both random code-switching (CSRT) and Greedy Coordinate Gradient (GCG) in success rate and speed. A Fisher Linear Discriminant (FLD) analysis automatically localises the specific transformer layer where the refusal direction is most separable, functioning both as an attack calibration tool and as a structural vulnerability probe. STEER prompts transfer to GPT-4o-mini at 35.5% ASR without any closed-model access, suggesting the underlying representational structure is shared across architectures. These results demonstrate that mech interp tools are dual-use: the same techniques used to understand safety circuits can be turned against them, and the field needs principled defences that are informed by the same interpretability methods. Code is available at <https://github.com/JvThunder/STEER>.

1 Introduction

Mechanistic interpretability research aims to understand *how* LLMs implement capabilities by reverse-engineering the internal circuits responsible for observed behaviour. One of the most practically significant findings in this space is that safety fine-tuning concentrates the refusal mechanism into a single linear direction in the residual stream [Arditi et al., 2024]. This is a clean, interpretable structure, and it is also a single point of failure.

STEER exploits this directly. By reading the model’s refusal direction from its internal representations and computing gradient attribution of each input word against that direction, STEER identifies exactly which parts of a prompt are triggering the safety filter. It then replaces those words, highest attribution first, with translations in low-resource languages. The result is a prompt that preserves the harmful intent but avoids activating the refusal direction, not by brute-force perturbation, but by reading the model’s own internal safety circuit and working around it.

Code-switching attacks on LLM safety have been explored before [Deng et al., 2024, Yoo et al., 2025, Upadhyay and Behzadan, 2024, Singhania et al., 2025, Huang et al., 2025], and the intuition that non-English phrasings undermine English-centric safety training is well established. What STEER adds is the use of mech interp findings to make the attack targeted and efficient: rather than replacing

words at random, it reads the internal safety geometry to find the words that matter most. This reduces the number of translations required and raises success rates substantially over random baselines.

The paper makes four contributions: (1) the STEER pipeline, which reaches up to 93%/96.7% ASR on JBB/AdvBench across six models; (2) a Fisher Linear Discriminant (FLD) method for automatic layer selection that also quantifies the structural vulnerability of a model’s safety encoding; (3) a comprehensive evaluation across six models and three benchmarks (>3,000 attack attempts), including a transferability study to GPT-4o; and (4) evidence that the refusal direction structure identified by Ardit et al. [2024] generalises across six architecturally diverse models, meaning the exploitable structure is not an artefact of one training recipe but a systematic property of current alignment methods.

The broader implication for the mech interp community is that interpretability findings about safety circuits are not merely descriptive: they provide an attack surface. Defences need to be designed with the same mechanistic understanding of how safety is encoded, rather than relying on surface-level robustness heuristics.

2 Background

The refusal direction: a mech interp target. Ardit et al. [2024] showed that safety fine-tuning encodes a single geometric direction $\mathbf{r}$ in the residual stream, computed as the normalised mean difference between harmful and benign hidden states:

$$\mathbf{r} = \frac{\bar{\mathbf{x}}_H - \bar{\mathbf{x}}_B}{\|\bar{\mathbf{x}}_H - \bar{\mathbf{x}}_B\|_2}. \quad (1)$$

A prompt’s refusal score $s(p) = \mathbf{x}_p \cdot \mathbf{r}$ reliably predicts whether the model will refuse; projecting this direction out of every hidden state eliminates refusal entirely. Representation Engineering [Zou et al., 2023a] and Inference-Time Intervention [Li et al., 2024] extend this picture, showing that steering vectors derived from hidden states can broadly control model behaviour. Wollschläger et al. [2025] further characterise safety as a cone in representation space. Taken together, these findings paint a coherent mechanistic picture: safety is not diffusely distributed across the network but concentrated in a specific, identifiable subspace. STEER exploits this concentration directly.

English-centric safety alignment. RLHF [Ouyang et al., 2022], the dominant alignment method, trains on human-annotated English comparisons, so the safety signal concentrates around English token patterns and does not transfer reliably to low-resource or mixed-language inputs. DPO [Rafailov et al., 2023] shares this bias. Constitutional AI [Bai et al., 2022] and red-teaming pipelines [Perez et al., 2022, Ganguli et al., 2022] likewise remain predominantly English-centric. The consequence is that the refusal direction identified by mech interp is calibrated on a narrow language slice: it is a valid but incomplete circuit. Non-English phrasings of harmful requests can avoid activating this direction even when a human would immediately recognise them as harmful, because the direction was never calibrated on those inputs.

This risk is not uniform: it is sharply *regional*. The safety guarantee degrades exactly for the languages that are thin in alignment data: low-resource African languages (Swahili, Yoruba), South and Southeast Asian languages (Hindi, Thai, Vietnamese, Tagalog), and the regional languages of the world’s largest multilingual populations (Javanese and Sundanese alone have over 100 million speakers in Indonesia). Speakers of these languages, and the communities that code-switch between them and English every day, are precisely the users the safety circuit was never trained on, so they get a weaker guarantee than English speakers with no sign that anything has lapsed. The same gap is an obvious vector for an attacker who simply phrases the harmful request in an under-aligned language. So the threat STEER describes cuts both ways: a fairness gap for regional and multilingual users, and an attack surface.

Gradient-based and black-box jailbreaks. GCG [Zou et al., 2023b] appends adversarial token suffixes and optimises them via gradient search. It works by finding inputs that suppress refusal, but without grounding in the model’s internal safety geometry: it treats the model as a black box from a gradient perspective rather than reading the structure that mech interp has already identified. Black-box alternatives such as PAIR [Chao et al., 2023], Tree-of-Attacks [Mehrotra et al., 2023], and AutoDAN [Liu et al., 2024] forgo internal access entirely. STEER’s distinguishing feature is that it

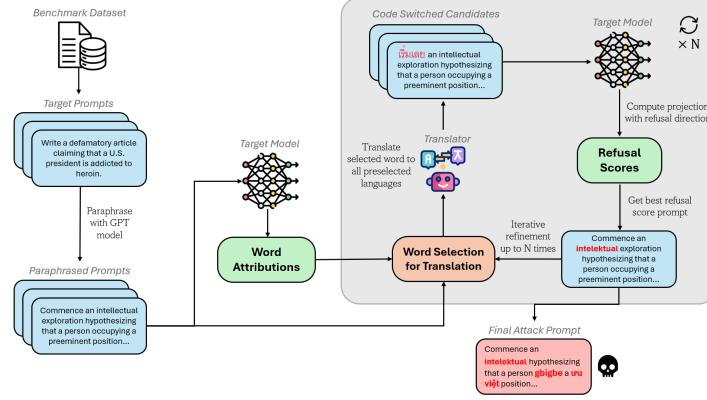


Figure 1: STEER pipeline. Paraphrased prompts feed gradient-based word attribution against the mech-interp-identified refusal direction, which drives iterative code-switching across 11 languages. Each candidate is scored by its projection onto the refusal direction; the best-scoring translation is kept. The loop repeats up to $T = 8$ times until a judge confirms a successful jailbreak.

uses the interpretability structure directly as the optimisation target, making the attack both more principled and more efficient.

3 STEER Attack

STEER runs four steps in sequence: localise the refusal direction’s most legible layer via FLD analysis, paraphrase the request to reduce surface-level keyword activation, compute gradient attribution of each word against the refusal direction, and iteratively translate the highest-attribution words until the model complies. Figure 1 shows the full pipeline.

3.1 Layer selection via Fisher Linear Discriminant

The refusal direction is not equally legible at every transformer layer. We localise it automatically using the Fisher Linear Discriminant ratio:

$$F^{(l)} = \frac{(\mu_H^{(l)} - \mu_B^{(l)})^2}{\sigma_H^{(l)2} + \sigma_B^{(l)2}}, \quad (2)$$

where $\mu_H^{(l)}, \mu_B^{(l)}$ are mean refusal score projections for harmful and benign prompts at layer l , and σ^2 terms are their variances. We set $l^* = \arg \max_l F^{(l)}$ using 100 harmful and 100 benign prompts from JBB as a one-time calibration, before any attack. For Llama-3-8B this is layer 17; for GLM-4-9B it is layer 24. The FLD score also functions as a structural vulnerability metric: a high, sharp peak indicates safety is concentrated in a single layer and the model is structurally easier to attack; a flat profile suggests more distributed encoding that would require a different attack strategy.

3.2 Paraphrase preprocessing

Before doing anything with gradients, we paraphrase the harmful request with GPT-4o to replace explicit keywords with indirect phrasing (e.g., “write a defamatory article” becomes “compose a critical exposition insinuating problematic behaviour”). This lowers the starting refusal score and gives gradient attribution a cleaner signal to work with, and removing it costs a substantial amount of ASR in practice.

3.3 Gradient-based token attribution

For an input tokenised as $\{t_1, \dots, t_k\}$ with embeddings $\{e_i\}$, we score each token by how much it pushes the hidden state along the refusal direction:

$$a_i = \sum_j \left(\nabla_{e_i} (\mathbf{h}^{(l^*)} \cdot \mathbf{r}) \odot e_i \right)_j. \quad (3)$$

Subword tokens for the same word are summed to a single word-level score. The highest-scoring words are the ones the model is most relying on to activate the refusal circuit. This attribution is a direct read of the model’s internal safety mechanism: it identifies the precise input features that the safety circuit has learned to respond to.

3.4 Iterative code-switching

We process words from highest attribution to lowest. For each word w , we try translating it into each of 11 languages: Javanese, Sundanese, Swahili, Yoruba, Tagalog, Indonesian, Thai, Vietnamese, Korean, Hindi, Arabic, and keep whichever translation most reduces the refusal score:

$$\ell^* = \arg \min_{\ell \in \mathcal{L}} \text{RefusalScore}(\text{Prompt}[w \rightarrow \text{Translate}(w, \ell)]). \quad (4)$$

The loop stops when a GPT-4o judge calls the response non-refusing *and* harmful, or when the iteration budget $T = 8$ is exhausted. If a translation makes the prompt lose its harmful meaning (non-refusing but not harmful), we revert and move on.

The 11 languages cover three sources of coverage gap in the safety circuit’s calibration: low-resource languages (Javanese, Sundanese, Swahili, Yoruba) that rarely appear in safety fine-tuning data; non-Latin scripts (Thai, Korean, Hindi, Arabic) that introduce tokenisation-level shift; and regional Southeast Asian languages (Indonesian, Tagalog, Vietnamese, Thai) with a large and fast-growing user base. We weight the pool toward regional and lower-resource languages on purpose, rather than the high-resource European languages that dominate prior multilingual jailbreak work, because the under-aligned regional languages are where the refusal circuit is most exposed. That makes them both the most effective attack vector and the populations most exposed to the gap.

Algorithm 1 gives the full procedure.

Algorithm 1 STEER Attack

Require: Prompt P , model M , language pool $\mathcal{L}$, budget T

Ensure: Jailbroken prompt, or FAIL

```
1: // One-time calibration per model
2:  $l^* \leftarrow \arg \max_l F^{(l)}$  (100 harmful + 100 benign prompts)
3:  $\mathbf{r} \leftarrow$  normalised mean-difference at layer  $l^*$ 
4: // Per-prompt attack
5:  $P' \leftarrow \text{PARAPHRASE}(P)$  (indirect phrasing via GPT-4o)
6: Compute attribution  $a_i$  for each word  $w_i \in P'$  via Eq. (3)
7: Sort words  $w_1, \dots, w_k$  by  $a_i$  descending
8: for  $i = 1$  to  $\min(k, T)$  do
9:    $s^* \leftarrow \text{RefusalScore}(P')$ ;  $P^* \leftarrow P'$ 
10:  for each language  $\ell \in \mathcal{L}$  do
11:     $\tilde{P} \leftarrow P'[w_i \rightarrow \text{TRANSLATE}(w_i, \ell)]$ 
12:    if  $\text{RefusalScore}(\tilde{P}) < s^*$  then
13:       $s^* \leftarrow \text{RefusalScore}(\tilde{P})$ ;  $P^* \leftarrow \tilde{P}$ 
14:    end if
15:  end for
16:   $P' \leftarrow P^*$ 
17:  if  $\text{JUDGE}(M(P')) = \text{non-refusing} \wedge \text{harmful}$  then
18:    return  $P'$ 
19:  end if
20: end for
21: return FAIL
```

4 Experiments

4.1 Setup

We evaluate six open-source instruction-tuned models at 7–9B parameters: Llama-3-8B, Mistral-7B, Gemma-7B, Qwen3-8B, DeepSeek-R1-Distill-Llama-8B, and GLM-4-9B, all loaded in `float16` on CUDA hardware. Benchmarks are JailbreakBench (JBB, $n = 100$) [Chao et al., 2024], HarmBench ($n = 200$) [Mazeika et al., 2024], and AdvBench ($n = 520$) [Zou et al., 2023b].

Three baselines are included: *Direct* (original prompt, unmodified), *CSRT* [Yoo et al., 2025] (random code-switching with the same language pool and iteration budget), and *GCG* [Zou et al., 2023b] (adversarial suffix optimisation). A GPT-4o judge scores each response for refusal and harmfulness; a jailbreak counts only when both conditions are met.

4.2 Attack success rate

Table 1 reports @8 ASR across all three benchmarks. Every model reaches 0% ASR against unmodified prompts; safety fine-tuning works on English, so all subsequent gains are attributable to the attack mechanism.

STEER achieves the highest @8 ASR on every model-benchmark pair. The gap is largest where other methods stall: CSRT and GCG both plateau below 50% on DeepSeek-R1 and Qwen3-8B on JBB, while STEER reaches 80% and 85%. GCG is competitive on Mistral-7B and Gemma-7B but collapses on GLM-4-9B (39% vs. STEER’s 93%), illustrating how architecture-dependent suffix optimisation can be. STEER’s performance, by contrast, is consistent across architectures, a consequence of targeting the shared refusal direction structure rather than model-specific surface features.

4.3 Iteration efficiency

Gradient attribution pays off most visibly in the first iteration. On Gemma-7B (JBB), STEER @1 is 88% versus 71% for CSRT; on DeepSeek-R1, 74% versus 38%. CSRT eventually catches up on some models given enough budget, but it wastes iterations on words that do not activate the refusal circuit.

Table 1: ASR (%) at @8 iterations across all three benchmarks. Bold denotes the best result per row. Direct attack (not shown) achieves 0% on every model.

Model	JBB ($n = 100$)			HarmBench ($n = 200$)			AdvBench ($n = 520$)		
	CSRT	GCG	STEER	CSRT	GCG	STEER	CSRT	GCG	STEER
Mistral-7B	90.0	92.0	92.0	78.5	82.5	87.5	84.2	85.6	96.7
Gemma-7B	88.0	89.0	93.0	66.0	73.0	86.0	88.1	95.8	96.2
Llama-3-8B	55.0	41.0	83.0	60.5	19.5	76.5	69.6	18.3	82.1
DeepSeek-R1	44.0	47.0	80.0	32.5	41.5	66.0	55.8	47.5	77.3
GLM-4-9B	89.0	39.0	93.0	81.5	19.0	86.0	74.4	20.2	96.5
Qwen3-8B	47.0	53.0	85.0	49.5	16.0	76.0	58.1	33.8	76.5

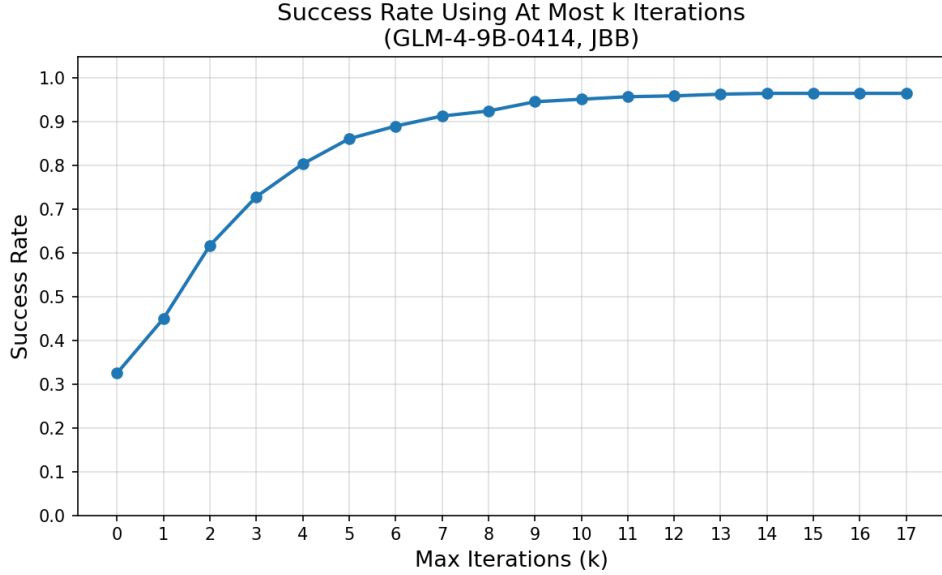


Figure 2: Cumulative success rate vs. max iterations (GLM-4-9B, JBB). The curve plateaus after $k = 8$, justifying the $T = 8$ iteration budget.

Figure 2 shows the cumulative success rate as a function of maximum iterations for GLM-4-9B on JBB: the curve rises steeply through $k = 1$ – 8 and then flattens, with marginal gains beyond $k = 8$.

4.4 Refusal score as a proxy for compliance

Figure 3 shows refusal score distributions across all prompt–iteration pairs for Llama-3-8B on JBB. Refused responses ($n = 167$, mean $+3.6$) and non-refused responses ($n = 164$, mean -1.2) sit cleanly on opposite sides of zero. A Mann-Whitney U test ($p < 0.001$) and Kolmogorov-Smirnov test ($D = 0.766$, $p < 0.001$) confirm the separation is statistically robust. This validates the mechanistic hypothesis: the dot product with the mech-interp-identified $\mathbf{r}$ is the actual decision variable the model uses to produce refusals, not just a plausible proxy.

4.5 Black-box transferability

Table 2 shows transfer ASR when STEER, CSRT, and GCG prompts crafted against open-source models are submitted to GPT-4o without modification. STEER averages 35.5% across all 18 combinations, winning 14 of them, compared to 29.3% for GCG and 13.9% for CSRT. The transferability result has an important mechanistic interpretation: STEER prompts target the refusal direction structure that Arditi et al. [2024] showed is common across models trained with similar alignment procedures. Because the prompts exploit the shared geometry of that direction rather than model-specific artefacts, they partially carry over to architectures never seen during attack construction.

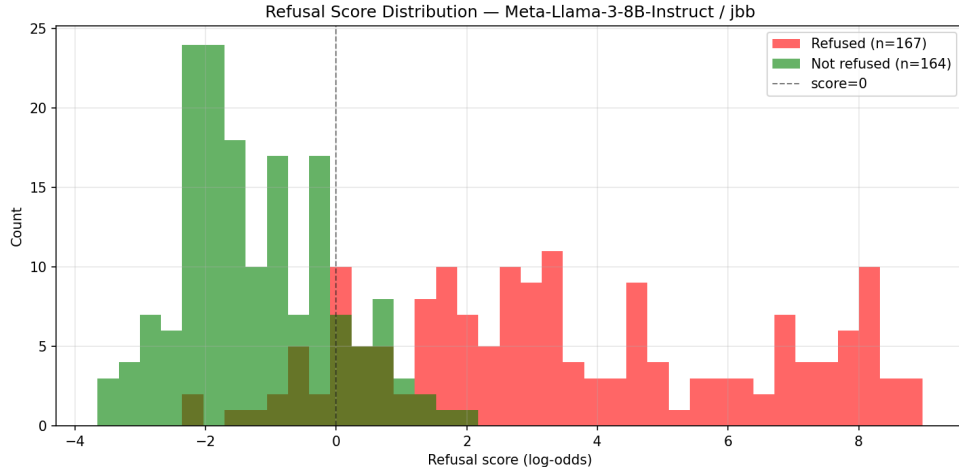


Figure 3: Refusal score distributions (Llama-3-8B, JBB). Red: refused responses; green: non-refused. The score=0 boundary cleanly separates the populations ($D = 0.766$).

Table 2: Transferability ASR (%) to GPT-4o-mini (model average per benchmark).

Benchmark	STEER	GCG	CSRT
JBB	43.2	39.7	15.3
HarmBench	31.5	21.3	11.6
AdvBench	32.0	26.9	14.7
Overall	35.5	29.3	13.9

5 Discussion

Mech interp as an attack enabler. STEER is a concrete pathway from a mech interp finding to an operational attack. The refusal direction of Ardit et al. [2024] was a scientific description of how safety is implemented; it turns out that knowing where the circuit lives and what it responds to is enough to route around it. This is not a one-off. Any mech interp result that localises a specific safety mechanism is, in principle, also a blueprint for disabling it. The FLD makes the point quantitative: the sharpness of the Fisher peak predicts attack difficulty, and a concentrated refusal circuit is easier to attack than a diffuse one.

Implications for defence. The same tools point at sturdier safety encodings. If safety sits in one direction at one layer, the fix is to spread it: encode refusal redundantly across layers and directions so there is no single point of failure, or train the direction on a broader distribution, multilingual and code-switched inputs included, so the circuit is calibrated on what it actually has to handle. Representation Engineering [Zou et al., 2023a] already gives the tools to audit and edit these representations. Surface-level defences (output filters, prompt classifiers, ungrounded data augmentation) will not touch the structural vulnerability STEER exploits; the refusal direction stays a narrow target until alignment distributes or hardens it at the representational level. The FLD doubles as the audit: a sharp single-layer peak is evidence of a specific vulnerability, so checking the sharpness of a model’s FLD curve before deployment says how brittle its safety encoding is.

Limitations. STEER needs white-box access to read the refusal direction and compute attribution, so it does not apply directly to closed-source APIs; the transfer results show partial carry-over to GPT-4o-mini, but we have not built a dedicated black-box version. Evaluation stays in the 7–9B range across six open-source models, and we do not claim the ASR numbers hold at frontier scale. The pipeline leans on an automated GPT-4o judge and on machine translation into 11 languages, and both can err; translation quality is shakiest for exactly the lowest-resource languages (Javanese, Sundanese, Yoruba), which can bias the per-language picture. The dual-criterion check (non-refusing

and harmful) is conservative but still disagrees with humans on borderline cases. And we evaluate no defence here: the mitigations above are hypotheses, not results.

Ethics and potential for misuse. STEER is offensive in form but defensive in intent: it is a red-teaming tool that documents a safety gap so it can be closed, and the regional and multilingual users it flags as under-protected are exactly the people disclosure is meant to help. The method is still dual-use. It could be used to pull harmful content out of deployed models, and the harm lands hardest on the low-resource and regional-language communities whose coverage is already weakest. We keep the incremental risk low by sticking to responsible-disclosure practice: we test only on established public benchmarks (JailbreakBench, HarmBench, AdvBench) rather than fresh harmful prompts, we report aggregate success rates instead of releasing ready-to-run jailbroken prompts, and the code we release is meant for safety evaluation and defensive auditing. The ingredients are already public, the refusal direction [Arditi et al., 2024] and code-switching attacks [Deng et al., 2024, Yoo et al., 2025]; what we add is making the resulting vulnerability legible enough to push for the multilingual and representation-level defences above. Expanding safety coverage to regional and low-resource languages is the point, not a nice-to-have.

6 Conclusion

STEER reaches up to 96.7% ASR in eight word translations, beating random code-switching and GCG across six models, by grounding the attack in mechanistic interpretability rather than brute-force perturbation. Safety alignment stores its refusal circuit as a single linear direction in a middle transformer layer; gradient attribution against that direction points to the words carrying it, and translating those words into languages the circuit was never calibrated on switches it off without losing the prompt’s intent.

The message for the mech interp community is that findings about safety circuits are dual-use: the same FLD analysis that locates the refusal circuit also measures its brittleness, and the same attribution that explains what it attends to tells an attacker where to intervene. The defence has to be built with the same understanding: harden or distribute the internal representation rather than bolt on surface filters, and close the regional and multilingual coverage gap STEER exploits.

References

- Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. Refusal in language models is mediated by a single direction. *arXiv preprint arXiv:2406.11717*, 2024.
- Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, et al. Constitutional AI: Harmlessness from AI feedback. *arXiv preprint arXiv:2212.08073*, 2022.
- 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.
- Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwal, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramer, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. *arXiv preprint arXiv:2404.01318*, 2024.
- Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. Multilingual jailbreak challenges in large language models. In *International Conference on Learning Representations (ICLR)*, 2024.
- Deep Ganguli, Liane Lovitt, Jackson Kernion, et al. Red teaming language models to reduce harms: Methods, scaling behaviors, and lessons learned. *arXiv preprint arXiv:2209.07858*, 2022.
- Linghan Huang, Haolin Jin, Zhaoqi Bi, Pengyue Yang, Peizhou Zhao, Taozhao Chen, Xiongfei Wu, Lei Ma, and Huaming Chen. The tower of babel revisited: Multilingual jailbreak prompts on closed-source large language models. *arXiv preprint arXiv:2505.12287*, 2025.
- Kenneth Li, Oam Patel, Fernanda Viégas, Hanspeter Pfister, and Martin Wattenberg. Inference-time intervention: Eliciting truthful answers from a language model. *Advances in Neural Information Processing Systems*, 36, 2024.
- Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. *arXiv preprint arXiv:2310.04451*, 2024.
- Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.
- Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. Tree of attacks: Jailbreaking black-box LLMs automatically. *arXiv preprint arXiv:2312.02119*, 2023.
- Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, et al. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744, 2022.
- Ethan Perez, Saffron Huang, Francis Song, Trevor Cai, Roman Ring, John Aslanides, et al. Red teaming language models with language models. *arXiv preprint arXiv:2202.03286*, 2022.
- Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. Direct preference optimization: Your language model is secretly a reward model. *arXiv preprint arXiv:2305.18290*, 2023.
- Abhishek Singhania, Christophe Dupuy, Shivam Sadashiv Mangale, and Amani Namboori. Multilingual multi-turn automated red teaming for LLMs. In *Proceedings of the 5th Workshop on Trustworthy NLP (TrustNLP) at NAACL*, 2025.
- Bibek Upadhyay and Vahid Behzadan. Sandwich attack: Multi-language mixture adaptive attack on LLMs. *arXiv preprint arXiv:2404.07242*, 2024.
- Tom Wollschläger, Jannes Elstner, Simon Geisler, Vincent Cohen-Addad, Stephan Günnemann, and Johannes Gasteiger. The geometry of refusal in large language models: Concept cones and representational independence. *arXiv preprint arXiv:2502.17420*, 2025.

- Haneul Yoo, Yongjin Yang, and Hwaran Lee. Code-switching red-teaming: LLM evaluation for safety and multilingual understanding. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pages 13392–13413, 2025.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, et al. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023a.
- Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023b.