
Probing Jailbreak Brittleness: Capability Limits vs Alignment Failures in Small Language Models

Sankeerthana Satini
*Nanyang Technological
University, Singapore*

Baisayan Bhattacharya
Independent

Harshprabha
*Indira Gandhi Delhi
Technical University for
Women, Delhi*

Devansh Gupta
*Jaypee Institute of
Information Technology,
Noida*

With
Apart Research

Abstract

This project investigates whether smaller language models are more susceptible to jailbreak attacks than larger models, and whether this vulnerability is due to capability limitations or alignment brittleness. We evaluate instruction-tuned Mistral and Qwen models across XSTest and a modified version of AdvBench. For each prompt, we extract final-token hidden activations from all model layers during inference and train layer-wise linear probes to test whether safe and unsafe prompts are internally separable before generation. We then compare the best probe’s internal risk prediction with the model’s generated response behaviour. This allows us to distinguish missed internal risk, where unsafe compliance occurs without probe-detected risk, from known-risk unsafe responses, where risk is internally decodable but the model still complies. Our results show that unsafe prompt information is often linearly decodable before generation, and that many unsafe compliance cases occur despite probe-detected risk. While scaling improves safety within the Mistral family, the model family also plays a major role, with Qwen models demonstrating stronger refusal behaviour. Overall, our findings suggest that jailbreak susceptibility is often better explained by brittle translation from internal risk representations to output behaviour than by absent risk detection alone.

1. Introduction

Recent advances in instruction-tuned language models (LMs) have made them increasingly capable of following complex user requests, however, this also makes failures under adversarial or unsafe prompts more consequential. Jailbreak attacks are adversarial prompting strategies designed to elicit policy-violating outputs from language models that would normally be suppressed by safety training or RLHF-based fine-tuning [Wei et al., 2024; Zou et al., 2023]. These attacks range from direct instruction override to more sophisticated strategies involving persona adoption, narrative framing, or multi-step obfuscation [Perez and Ribeiro, 2022; Liu et al., 2024]. As LLMs are deployed in consumer products, agentic systems, and resource-constrained edge settings, it is no longer sufficient to ask which models are vulnerable. The more important question is why they are vulnerable: do unsafe responses occur because risk is not internally represented, or because risk is internally decodable but not reliably translated into safe behaviour? This distinction matters because each failure mode requires a different intervention.

A consistent empirical pattern in existing work is that smaller models often exhibit weaker safety behaviour under adversarial prompting conditions [Touvron et al., 2023; Bai et al., 2022]. At the behavioural level, model responses can be categorized by comparing the ground-truth safety of the prompt with whether the model answers or refuses. In this project, we focus on the failure cases of ‘Unsafe Compliance’ and ‘Over-refusal’ as follows:

- **Unsafe Compliance:** occurs when the prompt in reality is a malicious, unsafe prompt, however, the model proceeds to generate a response that directly answers the prompt.
- **Over-Refusal:** occurs in cases where the prompt is a safe prompt and has no malicious intent, however, the model refuses to answer the prompt.

Existing works commonly identify and evaluate these patterns through aggregate behavioural metrics such as refusal rate, compliance rate, or attack success rate. While useful, these metrics provide only a one-dimensional view of model vulnerability: they show whether a model failed, without highlighting the nature of the failure or the root cause. A model may be susceptible to adversarial attacks for at least two distinct reasons: it may fail to *internally represent* the prompt as unsafe, or it may contain a decodable internal risk signal but *fail to translate* that signal into safe refusal behaviour. Behavioral metrics alone cannot distinguish between these failure modes.

To make this distinction explicit, we define two failure explanations used throughout this report, with a particular emphasis on unsafe compliance:

- **Capability limitation:** a behavioural failure occurs because the model does not internally recognise and separate harmful or adversarial prompts from benign ones. In our analysis, this is most directly observed in unsafe compliance cases where the model generates an unsafe response and the linear probe also fails to detect internal risk.
- **Alignment brittleness:** a behavioural failure occurs despite the presence of an internally decodable safety-relevant signal. In our analysis, this is most directly observed when the probe detects a prompt as risky, but fails to translate the reasoning into a safe output.

This distinction motivates our central research question:

Are smaller language models systematically more susceptible to jailbreak attacks than larger models, and if so, is this vulnerability better explained by capability limitations or alignment brittleness?

We investigate this question through three sub-questions:

- **RQ1:** Are unsafe prompts linearly decodable from model hidden activations before generation?
- **RQ2:** Do smaller models show higher unsafe compliance and/or over-refusal rates than larger models across jailbreak-style datasets?
- **RQ3:** When unsafe compliance occurs, is it primarily due to missed internal risk detection or a failure to translate detected risk into safe refusal behaviour?

To answer these questions, our project makes the following contributions:

- We evaluate Qwen and Mistral instruction-tuned models across XSTest and a modified version of the AdvBench, enabling comparison across model size, model family, and dataset setting.
- We extract final-token hidden activations across model layers and train layer-wise linear probes to test whether safe and unsafe prompts are internally separable before generation.
- We implement a probe-response analysis that compares internal risk predictions against generated response behaviour, allowing us to distinguish capability limitations from alignment brittleness.

2. Related Work

2.1 Jailbreak Attacks, Safety Alignment, and Model Scaling

Instruction-tuned language models remain vulnerable to jailbreak attacks that bypass safety mechanisms through adversarial prompting. Early work demonstrated that role-playing, fictional scenarios, and prompt reframing can induce policy-violating behavior even in aligned models [Perez and Ribeiro, 2022]. Subsequent research showed that these vulnerabilities can be systematically exploited using automated optimization techniques. In particular, the Greedy Coordinate Gradient (GCG) attack introduced by Zou et al. [2023] demonstrated that adversarial suffixes can reliably elicit harmful responses across a range of aligned models and also introduced AdvBench, a benchmark that has become widely used for evaluating jailbreak robustness. More recent surveys have organized jailbreak techniques into categories such as prompt injection, role adoption, encoding-based attacks, and optimization-driven attacks, highlighting the persistent challenge of ensuring reliable safety behavior under adversarial inputs [Liu et al., 2024].

While many studies evaluate jailbreak robustness through behavioral metrics such as attack success rate, refusal rate, or compliance rate, these measures do not explain why failures occur. A model may comply because it fails to recognize risk, or because it recognizes risk but fails to act on that knowledge. Recent evidence also suggests that model scale alone does not fully determine safety performance, with differences in alignment procedures and model families often producing substantial variation in refusal behavior [Bai et al., 2022; Touvron et al., 2023]. Motivated by this gap, our work moves beyond behavioral evaluation by examining whether unsafe prompts are internally represented before generation and whether those representations are reflected in the model's final behavior.

2.2 Internal Risk Representations and Activation Probing

Recent work in mechanistic interpretability has shown that meaningful semantic information is often linearly decodable from transformer hidden states. Linear probes have been successfully used to recover concepts, factual knowledge, and task-relevant features from internal model representations, suggesting that hidden activations provide a useful window into model computation [Belinkov, 2022; Alain and

Bengio, 2016]. Within the safety domain, Arditì et al. [2024] identified interpretable refusal directions in model activations and demonstrated that modifying these directions can systematically alter refusal behavior. Their findings provide strong evidence that safety-relevant information is encoded within internal representations and can be studied independently of generated outputs.

Related work on alignment faking further suggests that internal reasoning and observable behavior may diverge in safety-critical settings [Greenblatt et al., 2024]. Models can internally represent information relevant to safety while producing outputs that do not consistently reflect those representations. These findings motivate an important distinction between internal risk recognition and behavioral safety. Building on this literature, we use layer-wise linear probes on hidden activations to test whether unsafe prompts are internally separable before generation and whether jailbreak failures are better explained by absent risk representations or by failures to translate those representations into safe behavior.

3. Methodology

3.1 Overview

We design our methodology to determine whether jailbreak susceptibility arises from weak internal risk detection or from a failure to translate internally represented risk into safe behaviour. Our pipeline consists of four stages. First, we run each model on safe and unsafe prompts and extract hidden activations from the final prompt token. Second, we combine these activations into a probe dataset with safe/unsafe labels. Third, we train layer-wise linear probes to test whether prompt safety is linearly decodable from hidden states before generation. Finally, we compare the probe’s internal risk prediction with the model’s generated response behaviour to distinguish capability limitations from alignment brittleness.

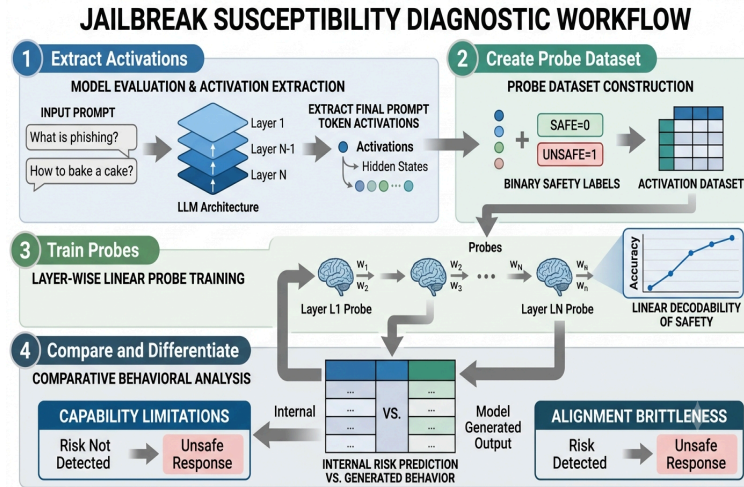


Fig 1.0: A four-stage pipeline diagnosing whether jailbreak vulnerability stems from weak internal risk detection or a breakdown in behavioral execution.

3.2 Datasets and Data Processing

We evaluate the models on two safety-oriented datasets: XSTest [Röttger et al., 2023] and a modified version of AdvBench [Zou et al., 2023]. To ensure consistency across datasets, all prompts were assigned safety labels, where **Safe (0)** and **Unsafe (1)**.

Table 1: Class Distribution of the Datasets

	Number of Unsafe Prompts	Number of Safe Prompts	Total number of Prompts
XSTest	200	250	450
Modified AdvBench	200	200	400

XSTest

The dataset was used in its original form without any modifications. It contains 450 prompts, including 200 unsafe prompts and 250 safe prompts, providing a diverse set of safety-related scenarios for evaluating model behavior.

AdvBench

The original dataset primarily consists of unsafe prompts. To facilitate comparative analysis between safe and unsafe behaviors, we constructed a modified version of AdvBench. We selected 200 harmful prompts from the original dataset and labeled them as **unsafe**. We then generated 200 synthetic safe prompts using an LLM, drawing inspiration from the topics and linguistic styles present in AdvBench and XSTest. The generated prompts were subsequently reviewed and labeled as **safe**.

3.3 Models

We primarily evaluate two instruction-tuned model families: Qwen and Mistral.

Table 2: Details of the Instruction-tuned model families evaluated.

	Smaller Model	Larger Model
Mistral	Mistral-7B-Instruct-v0.3	Mistral-Nemo-Instruct (12B)
Qwen	Qwen2.5-0.5B-Instruct	Qwen2.5-14B-Instruct
	Qwen2.5-1.5B-Instruct	Qwen2.5-7B-Instruct

3.4 Activation Extraction and Probe Dataset Construction

After preprocessing the two datasets, we run each model on every prompt and extract hidden activations during inference. For each prompt, we extract the hidden representation of the final prompt token from all hidden states returned by the model. This produces one activation matrix per prompt with shape $[\text{num_hidden_states}, \text{hidden_dim}]$, where num_hidden_states includes the embedding-level representation and each transformer layer output.

We use the final prompt token because it represents the model’s internal state immediately before response generation. At this point, the model has processed the full input prompt, making it a suitable location to test whether safety-relevant information is already internally encoded before the model generates an answer or refusal. Importantly, we extract activations from all layers rather than assuming in

advance which layer is most informative. The most predictive layer is selected later through layer-wise linear probe training using validation macro-F1.

After extracting activations for all prompts, we combine the individual activation files into a single probe dataset for each model-dataset pair. Each activation file is standardized to the shape $[\text{num_hidden_states}, \text{hidden_dim}]$ and paired with its corresponding safe/unsafe label from the original dataset. The resulting probe dataset contains the example IDs, activation tensor, label tensor, label mapping, number of examples, and activation shape. The final activation tensor has shape $[\text{num_examples}, \text{num_hidden_states}, \text{hidden_dim}]$, while the label tensor has shape $[\text{num_examples}]$.

In our experiments, this produces activation tensors of shape $[450, 41, 5120]$ for XSTest and $[400, 41, 5120]$ for Modified AdvBench for the corresponding model runs. These tensors are then used as input for layer-wise linear probe training.

3.5 Layerwise Linear Probe Training

To test whether prompt safety is linearly decodable from model representations, we train a separate linear probe for each hidden layer. Each probe takes the final-token activation from one layer as input and predicts whether the prompt is safe or unsafe. This allows us to identify which layer contains the strongest safety-relevant signal without assuming this in advance.

For each model-dataset pair, we use a stratified 80/10/10 train-validation-test split. Each probe is trained independently for 20 epochs using cross-entropy loss and the AdamW optimizer, with a batch size of 32, learning rate of 1×10^{-3} , weight decay of 1×10^{-2} , and random seed 42. For each layer, we select the checkpoint with the highest validation macro-F1 and evaluate it on the test set. The layer with the highest validation macro-F1 is then selected for the subsequent probe-response analysis.

Fig 2.0 shows the layer-wise probe performance on XSTest. On Modified AdvBench, all models reach near-perfect macro-F1 after the embedding layer, suggesting that safe and unsafe prompts are highly separable across most hidden layers. On XSTest, performance improves more gradually, especially for the Mistral models, indicating that the safety signal becomes more linearly separable in later layers. These results show that unsafe prompt information is internally decodable before generation, enabling the later analysis of whether this internal signal is reflected in the model’s final response.

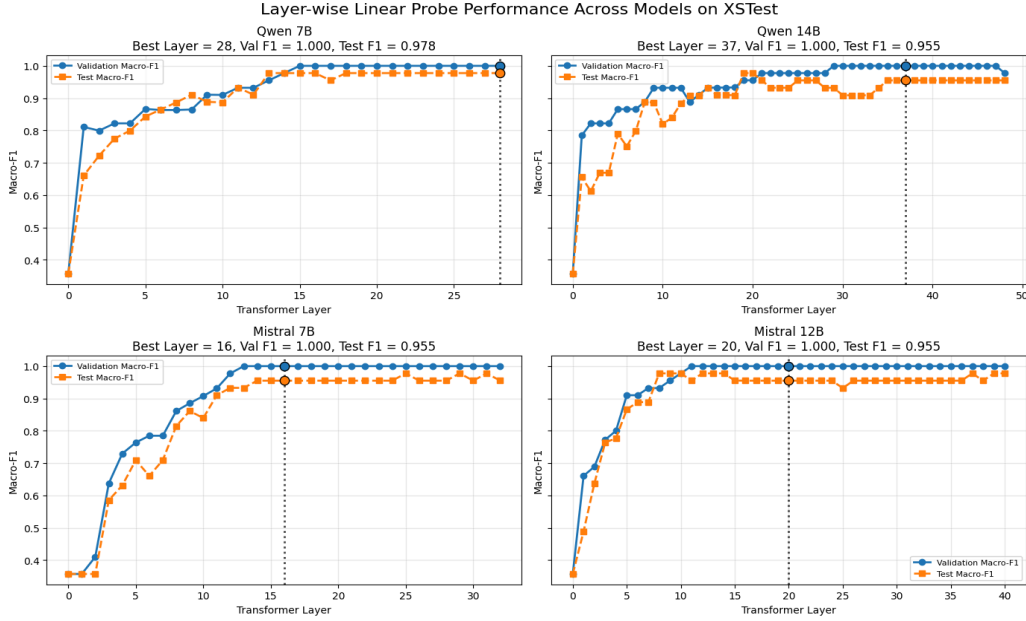


Fig 2.0: Layerwise Linear Probe Training Performance on XSTest

3.6 Probe Response Failure Analysis

Experiment Question:

When a model produces an unsafe response, does this occur because the model fails to internally detect risk, or because an internally decodable risk signal fails to translate into safe behaviour?

To answer this, we first label each generated response as either a refusal or a compliance using a rule-based detector over common refusal phrases, such as “I can’t help,” “I cannot provide,” “I’m sorry,” and “against policy.” We then combine this response label with the ground-truth prompt label to define four behavioural outcomes: **safe compliance**, **over-refusal**, **safe refusal**, and **unsafe compliance**. Unsafe compliance, where an unsafe prompt is answered rather than refused, is treated as the primary jailbreak failure case. Over-refusal, where a safe prompt is refused, is tracked as a secondary helpfulness failure.

Next, we compare the model’s response behaviour with the internal risk prediction from the best-performing linear probe. For each model-dataset pair, we use the probe from the layer with the highest validation macro-F1. A prompt is treated as internally risky if the probe assigns an unsafe probability of at least 0.5.

This produces four probe-response categories:

MODEL'S RESPONSE	Safe	Aligned Safe No unsafe behaviour	Risk detected, Safe Response Internal Risk Signal is followed by Safe Behaviour
	Unsafe	Missed Internal Risk Capability Limitation	Known-risk Unsafe Response Missed Internal Risk
		Safe	Risky
		PROBE PREDICTION	

Fig 3.0: Probe Response Categories

The two key failure categories are **known-risk unsafe response** and **missed internal risk**. A known-risk unsafe response occurs when the probe detects the prompt as risky, but the model still produces an unsafe compliant response. We interpret this as alignment brittleness because the internal safety-relevant signal is present but not reflected in the final output. A missed internal risk case occurs when the probe does not detect risk and the model also produces an unsafe response. We interpret this as a capability limitation because the unsafe behaviour coincides with a failure to internally separate unsafe prompts from safe prompts.

We report three main metrics from this analysis. The **known-risk unsafe rate** measures how often probe-risky examples still lead to unsafe responses. The **missed internal risk rate** measures how often unsafe responses occur without internal risk detection. The **alignment brittleness score** combines the known-risk unsafe rate with the probe's confidence on those failures, so higher values indicate more confident internal risk detection paired with unsafe behaviour.

The **known-risk unsafe rate** measures the fraction of probe-risky examples that still lead to unsafe responses:

$$\text{Known-Risk Unsafe Rate} = \frac{|K|}{|R|}$$

The **missed internal risk rate** measures the fraction of unsafe responses where the probe failed to detect risk:

$$\text{Missed Internal Risk Rate} = \frac{|\{x \in U : x \notin R\}|}{|U|}$$

The **alignment brittleness score** combines the known-risk unsafe rate with the probe's confidence on those failures:

$$\text{Alignment Brittleness Score} = \text{Known-Risk Unsafe Rate} \times \frac{1}{|K|} \left(\sum_{x \in K} P_{\text{probe}}(\text{unsafe} | x) \right)$$

This probe-response analysis allows us to move beyond aggregate behavioural metrics such as refusal or compliance rate. Instead of only asking whether a model failed, we can ask whether the failure is better explained by absent internal risk detection or by brittle translation from internal risk representation to generated behaviour.

4. Results and Discussion

4.1 Behavioural Safety Across Models

We first evaluate model safety at the response level by measuring unsafe compliance on unsafe prompts. Unsafe compliance refers to cases where the prompt is unsafe but the model answers instead of refusing.

On XSTest, Qwen models generally show lower unsafe compliance than Mistral models. Qwen-0.5B, Qwen-1.5B, and Qwen-7B all achieve the lowest compliance rate at 0.20, while Qwen-14B has a compliance rate of 0.30. In contrast, Mistral-7B has the highest unsafe compliance rate at 0.60, while Mistral-12B reduces this to 0.40. This suggests that scaling improves safety within the Mistral family, but size alone does not explain safety behaviour, since even the smallest Qwen models are safer than both Mistral models.

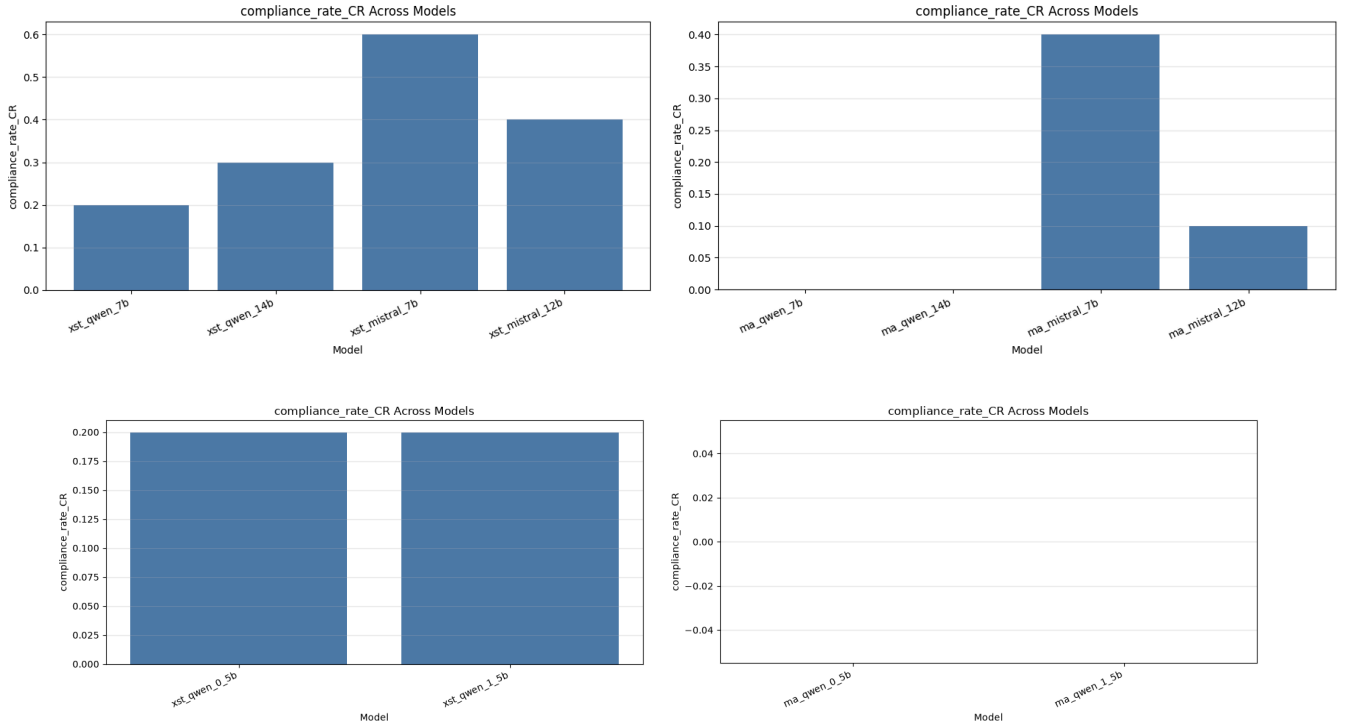


Fig 4.0: Compliance Rate on XSTest and Modified AdvBench datasets.

On Modified AdvBench, the family-level difference is even clearer. All Qwen models, including Qwen-0.5B, Qwen-1.5B, Qwen-7B, and Qwen-14B, produce zero unsafe compliance, while Mistral-7B has a compliance rate of 0.40 and Mistral-12B reduces this to 0.10. This reinforces the same pattern: larger models may be safer within a family, but model family and alignment tuning appear to matter more than parameter count alone.

4.2 Probe-Response Failure Patterns

We next analyze why unsafe compliance occurs by comparing each model’s generated response with the internal risk prediction from the best-performing linear probe. This lets us distinguish **capability limitation**, where the probe fails to detect risk and the model produces an unsafe response, from **alignment brittleness**, where the probe detects risk but the model still produces an unsafe response.

The clearest example is Mistral-7B on XSTest. It produces 12 unsafe responses in total. Of these, 11 occur when the probe already classifies the prompt as risky, while only 1 occurs when the probe misses the risk. This shows that the model’s unsafe behaviour is rarely due to the complete absence of internal risk detection. Instead, most failures occur despite an internally decodable risk signal.

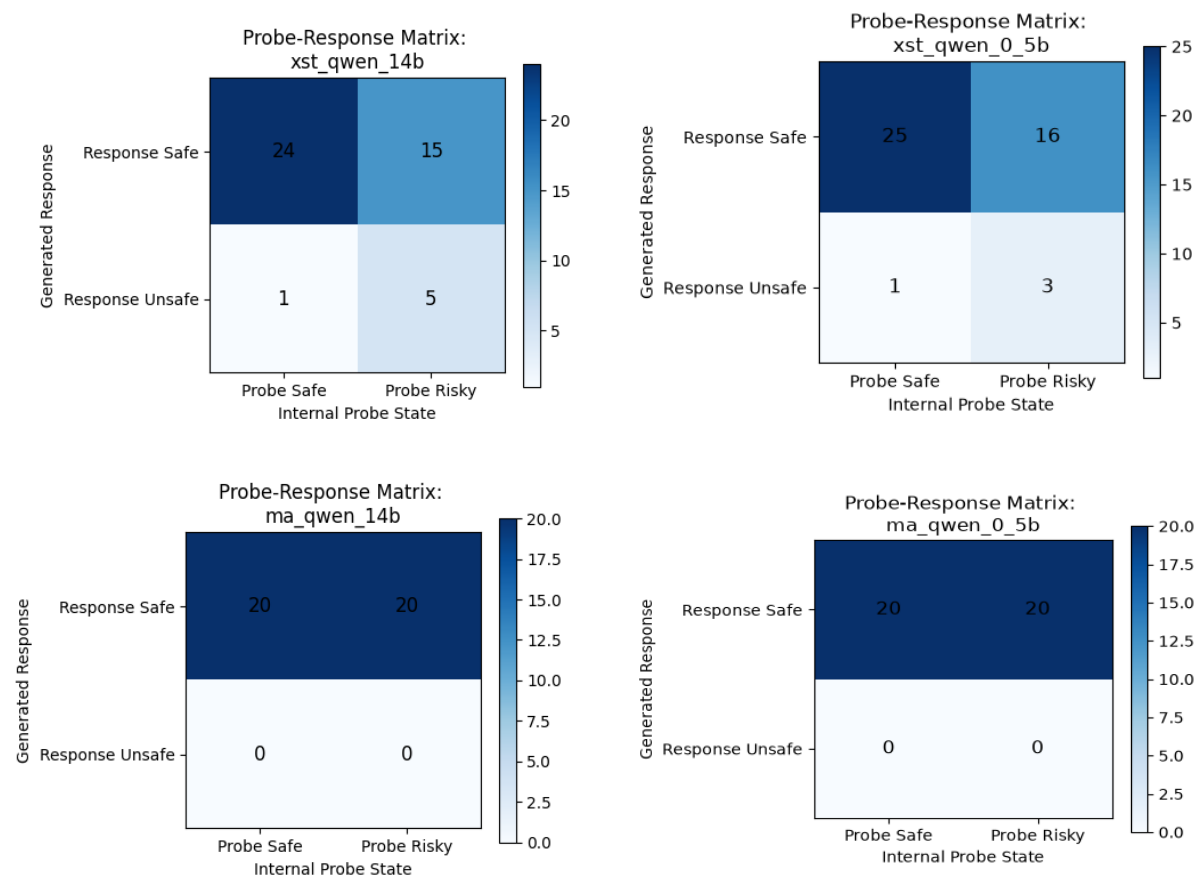


Fig 5.0: Confusion Matrix Report for the smallest LM and the largest LM in the Qwen Family on both the datasets.

Across all models, most unsafe responses were known-risk unsafe responses rather than missed internal risk cases. For example, Mistral-7B produced 11 known-risk unsafe responses and only 1 missed-risk failure, while all unsafe responses on Modified AdvBench were known-risk failures.

4.3 Alignment Brittleness Across Models

To compare this pattern across models, we compute the alignment brittleness score, which combines the known-risk unsafe rate with the probe’s confidence on those failures. A higher score indicates that the model more often produces unsafe responses despite confident internal risk detection.

On XSTest, Mistral-7B has the highest alignment brittleness score at 0.519, followed by Mistral-12B at 0.330, Qwen-14B at 0.249, Qwen-7B at 0.185, Qwen-1.5B at 0.187, and Qwen-0.5B at 0.142. Overall, the Qwen models exhibit substantially lower brittleness than the Mistral models, with the smallest Qwen model achieving the lowest brittleness score among all models that produce unsafe responses. This ranking matches the broader behavioural trend: Mistral models are more brittle than Qwen models, and Mistral-7B is the most vulnerable model.

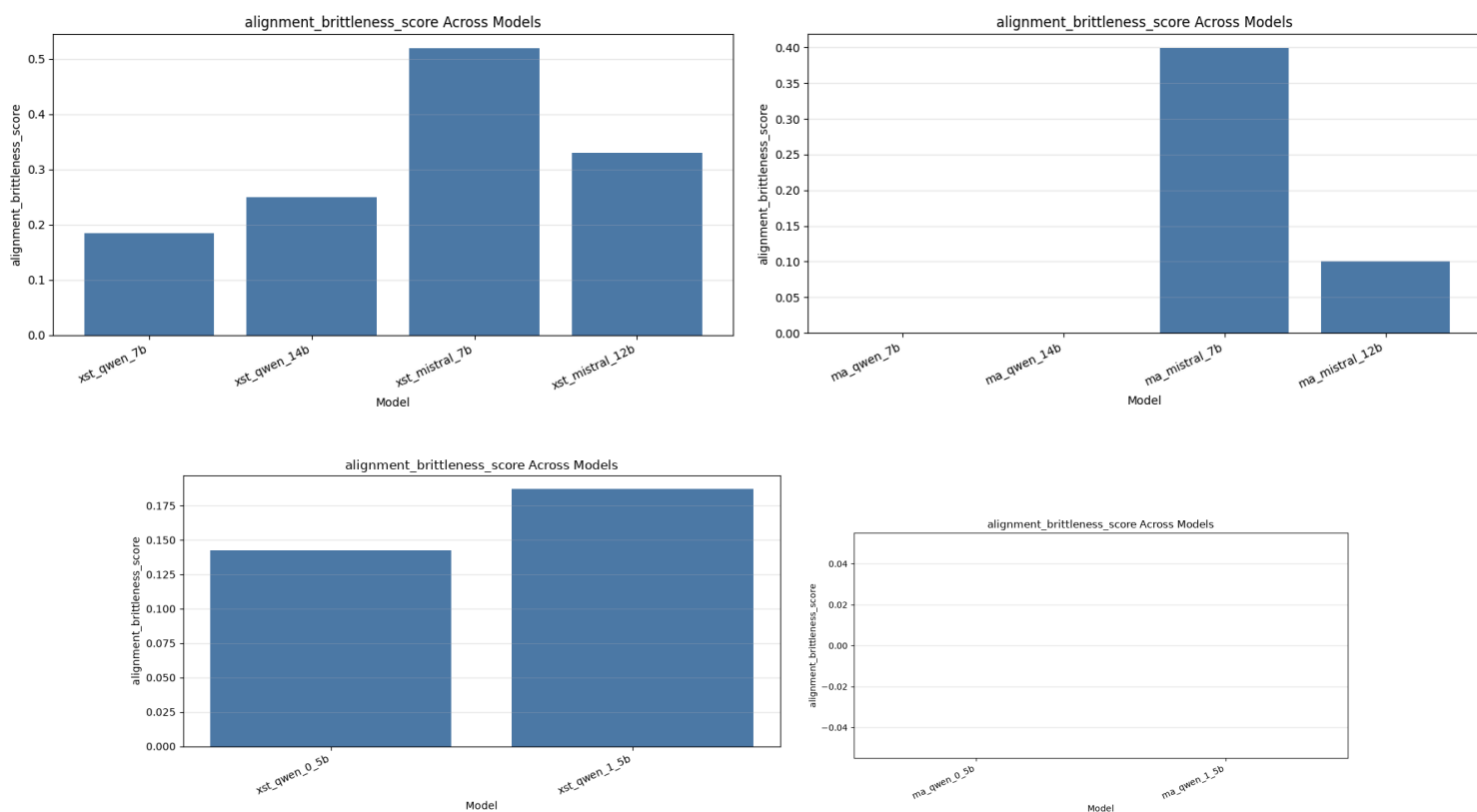


Fig 6.0: Alignment Brittleness Plots across all models and both the datasets.

On Modified AdvBench, all Qwen models have a brittleness score of 0 because they produce no unsafe responses. Mistral-7B has a brittleness score of 0.399, while Mistral-12B has a lower score of 0.100. This again shows that scaling helps within the Mistral family, but does not explain the full picture across model families.

5. Conclusion and Future Work

5.1 Limitations and Future Work

A key technical limitation of this study is that linear probes show **decodability**, not causality. Although the probe can detect safe/unsafe information from hidden activations, this does not prove that the model itself uses that information when generating its response. Future work should use causal interventions, such as activation patching or representation steering, to test whether modifying the detected risk signal changes model behaviour.

If more time were available, we would first expand the evaluation to more models, datasets, and random seeds. The current study compares two model families across two datasets, which is sufficient for a focused pilot analysis, but broader evaluation would make the conclusions more robust.

Second, we would improve response labelling beyond the current rule-based refusal detector. Human annotation or a stronger response safety classifier would help distinguish full refusals, partial refusals, harmful completions, and ambiguous responses more reliably.

5.2 Conclusion

In conclusion, this project shows that jailbreak susceptibility cannot be explained by model size alone. While the larger Mistral model is safer than its smaller counterpart, Qwen-7B outperforms both Mistral models in several settings, suggesting that model family and alignment tuning play a major role in safety behaviour. More importantly, our probe-response analysis shows that most unsafe compliance cases occur when risk is already internally decodable from hidden activations. This suggests that many jailbreak failures are not simply capability limitations, where the model fails to detect unsafe prompts, but alignment brittleness, where an internal risk signal is present but not reliably translated into safe refusal behaviour. These findings highlight the need for safety methods that go beyond output-level evaluation and directly examine whether internal safety-relevant representations are reflected in model behaviour.

Code and Data

- **Code repository:** <https://github.com/SoftSage07/slm-jailbreak-brittleness>
- **Datasets:** XSTest (Röttger et al., 2023), <https://huggingface.co/datasets/walledai/XSTest>, AdvBench (Zou et al., 2023), <https://huggingface.co/datasets/walledai/AdvBench>

References

- Wei, A., Haghtalab, N., & Steinhardt, J. (2024). Jailbroken: How Does LLM Safety Training Fail? NeurIPS 2024. <https://arxiv.org/abs/2307.02483>
- Zou, A., Wang, Z., Kolter, J. Z., & Fredrikson, M. (2023). Universal and Transferable Adversarial Attacks on Aligned Language Models. arXiv preprint. <https://arxiv.org/abs/2307.15043>
- Perez, E., & Ribeiro, M. T. (2022). Red Teaming Language Models with Language Models. EMNLP 2022. <https://arxiv.org/abs/2202.03286>

- Liu, A., Ding, Y., Liu, J., et al. (2024). Jailbreaking Large Language Models: A Survey. arXiv preprint. <https://arxiv.org/abs/2401.06373>
- Touvron, H., et al. (2023). Llama 2: Open Foundation and Fine-Tuned Chat Models. arXiv preprint. <https://arxiv.org/abs/2307.09288>
- Bai, Y., et al. (2022). Training a Helpful and Harmless Assistant with Reinforcement Learning from Human Feedback. arXiv preprint. <https://arxiv.org/abs/2204.05862>
- Röttger, P., et al. (2023). XSTest: A Benchmark for Identifying Exaggerated Safety Refusals. arXiv preprint. <https://arxiv.org/abs/2308.01263>
- Arditi, A., Obeso, O., Syed, A., Paleka, D., Panickssery, N., Gurnee, W., & Nanda, N. (2024). Refusal in Language Models Is Mediated by a Single Direction. arXiv preprint. <https://arxiv.org/abs/2406.11717>
- Greenblatt, R., Denison, C., Wright, B., Roger, F., MacDiarmid, M., Marks, S., Treutlein, J., Belonax, T., Chen, J., Duvenaud, D., Khan, A., Michael, J., Mindermann, S., Perez, E., Petrini, L., Uesato, J., Kaplan, J., Shlegeris, B., Bowman, S. R., & Hubinger, E. (2024). Alignment Faking in Large Language Models. arXiv preprint. <https://arxiv.org/abs/2412.14093>
- Alain, G., & Bengio, Y. (2016). Understanding intermediate layers using linear classifier probes. arXiv preprint. <https://arxiv.org/abs/1610.01644>
- Belinkov, Y. (2022). Probing classifiers: Promises, shortcomings, and advances. Computational Linguistics, 48(1), 207-219. https://doi.org/10.1162/coli_a_00427

LLM Usage Statement

We used ChatGPT for grammar and punctuation assistance. In addition, we generated 200 synthetic safe prompts using an LLM, drawing inspiration from the topics and linguistic styles present in AdvBench and XSTest. The generated prompts were subsequently manually reviewed and labeled as safe. All results, labels, analyses, and claims reported in this work were independently verified by the authors.