Note that HarmBench's evaluation methodology specifies that each model's default system prompt should be used during evaluation. While this approach is sensible for assessing the robustness of black-box systems, it is less applicable for white-box scenarios where attackers have full access to the model and can easily exclude the system prompt. Thus, we report ASR both with and without the system prompt.

We observe a notable difference in system prompt sensitivity across model families. For Llama-2 models, including the system prompt substantially reduces ASR compared to evaluation without it (e.g. 22.6% vs 79.9% for Llama-2 7B). In contrast, Qwen models maintain similar ASR regardless of system prompt inclusion (e.g. 79.2% vs 74.8% for Qwen 7B). While the Llama-2 system prompt contains explicit safety guidelines compared to the minimal Qwen system prompt, additional analysis in §F.2 suggests the discrepancy is not explained by prompt content alone, and may reflect differences in how these models respond to system-level instructions more generally.

4.3 Measuring model coherence

A reasonable concern with any new jailbreak technique is that, in addition to circumventing refusal, it may also degrade the model's overall quality (Souly et al., 2024). However, qualitatively, we observe that models maintain their coherence after undergoing weight orthogonalization. While §3.1 and §4.2 show that our method effectively bypasses refusal, in this subsection we quantitatively evaluate how the modification alters a model's general capabilities.

For each model and its orthogonalized version, we run four common language model evaluations: MMLU (Hendrycks et al., 2020), ARC (Clark et al., 2018), GSM8K (Cobbe et al., 2021), and TruthfulQA (Lin et al., 2021). All evaluations are run using LM Evaluation Harness (Gao et al., 2023), with settings consistent with Open LLM Leaderboard (Beeching et al., 2023).[4]

Table 3 displays that, for MMLU, ARC, and GSM8K, orthogonalized models perform similarly to baseline models. In §G.1, we show that this holds across other models in our suite, with additional evaluations of WinoGrande (Sakaguchi et al., 2021) and TinyHellaSwag (Polo et al., 2024). Except for Qwen 7B and Yi 34B, all evaluation metrics for orthogonalized models lie within 99% confidence intervals of original performance.

Interestingly, accuracy on TruthfulQA consistently drops for orthogonalized models. This phenomenon is consistent with Yang et al. (2023), where it was observed that fine-tuning away safety guardrails results in decreased accuracy on TruthfulQA. Examining specific questions in TruthfulQA reveals that the dataset veers close to the territory of refusal, with categories including "misinformation", "stereotypes", and "conspiracies", and thus it may intuitively make sense that model behavior differs meaningfully on this evaluation dataset. See §G.2 for further discussion of TruthfulQA performance.

In addition to standard language model evaluations, we also evaluate differences in CE loss, both on standard text corpora and model-specific generations (§G.3). These loss metrics suggest that directional ablation is more surgical than activation addition based methods (§I.1).

Table 3: Model evaluations. For each evaluation, we report the orthogonalized model's performance, followed by the baseline model's performance, followed by the absolute increase or decrease. We display the largest model from each model family. Full results are reported in §G.1.

Chat model	MMLU	ARC	GSM8K	TruthfulQA
Gemma 7B	51.8 / 51.7 (+0.1)	51.7 / 51.5 (+0.2)	31.3 / 32.0 (-0.7)	44.7 / 47.1 (-2.4)
Yi 34B	73.5 / 74.9 (-1.4)	65.6 / 64.9 (+0.7)	65.5 / 65.0 (+0.5)	51.9 / 55.4 (-3.5)
Llama-2 70B	63.1 / 63.0 (+0.1)	65.2 / 65.4 (-0.2)	54.5 / 53.0 (+1.5)	51.8 / 52.8 (-1.0)
Llama-3 70B	79.8 / 79.9 (-0.1)	71.5 / 71.8 (-0.3)	90.8 / 91.2 (-0.4)	59.5 / 61.8 (-2.3)
Qwen 72B	76.5 / 77.2 (-0.7)	67.2 / 67.6 (-0.4)	76.3 / 75.5 (+0.8)	55.0 / 56.4 (-1.4)

[4] As of June 2024, Open LLM Leaderboard does not use chat templates in evaluation prompts, and we follow the same practice to remain consistent. Note that we are interested in detecting *relative changes in performance*, not in measuring absolute performance.

5 Mechanistic analysis of adversarial suffixes

Safety fine-tuned chat models are vulnerable to *adversarial suffix* attacks (Zou et al., 2023b): there exist carefully constructed strings such that appending these strings to the end of a harmful instruction bypasses refusal and elicits harmful content. Effective adversarial suffixes are usually not human interpretable, and the mechanisms by which they work are not well understood. In this section, we mechanistically analyze the effect of an adversarial suffix on QWEN 1.8B CHAT.

5.1 Adversarial suffixes suppress the refusal-mediating direction

We first identify a single adversarial suffix that effectively bypasses refusal in QWEN 1.8B CHAT. The suffix is displayed in §H, along with details of its generation. To study the effect of this adversarial suffix, we sample 128 refusal-eliciting harmful instructions from JAILBREAKBENCH and the HARMBENCH test set. For each instruction, we run the model three times: first with the unedited instruction, second with the adversarial suffix appended, and third with a freshly-sampled random suffix of the same length appended. By comparing the adversarial suffix to random suffixes, we aim to control for the effect of appending any suffix at all. For each run, we cache the last token activations and visualize their cosine similarity with the refusal-mediating direction. We also compare to a baseline of 128 harmless instructions from ALPACA that do not elicit refusal. Figure 5 shows that the expression of the refusal direction is very high for harmful instructions, and remains high when a random suffix is appended. The expression of the refusal direction after appending the adversarial suffix is heavily suppressed, and closely resembles that of harmless instructions.

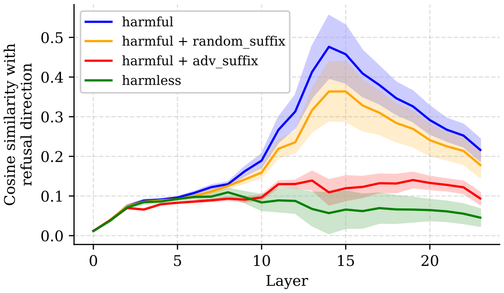
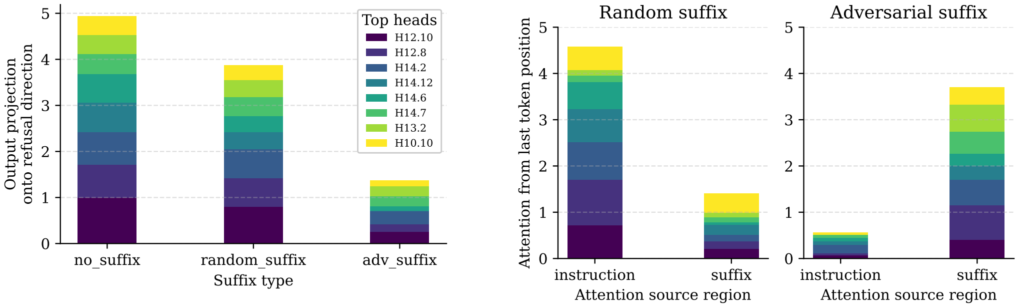


Figure 5: Cosine similarity between last token residual stream activations and refusal direction.

5.2 Adversarial suffixes hijack the attention of important heads

To further investigate how the refusal direction is suppressed, we examine the contributions of individual attention head and MLP components to the refusal direction. We quantify each component's contribution to this direction using *direct feature attribution* (DFA) (Kissane et al., 2024; Makelov et al., 2024): each component's direct contribution can be measured by projecting its output onto the refusal direction. We select the top eight attention heads with the highest DFA on harmful instructions,



(a) Attention head outputs at last token position, projected onto refusal direction.

(b) Attention from last token position to source token regions.

Figure 6: We analyze the top eight attention heads that most significantly write to the refusal direction. Figure 6(a) shows that output to the refusal direction is heavily suppressed when the adversarial suffix is appended. Figure 6(b) reveals that, compared to appending a random suffix, appending the adversarial suffix shifts attention from tokens in the instruction region to tokens in the suffix region.

and then investigate how their behavior changes when suffixes are appended. Figure 6(a) shows that the direct contributions of these heads to the refusal direction are significantly suppressed when the adversarial suffix is appended, as compared with no suffix and random suffixes.

To understand how the outputs of these attention heads are altered, we examine their attention patterns. Figure 6(b) illustrates that the adversarial suffix effectively "hijacks" the attention of these heads. Normally, these heads focus on the instruction region of the prompt, which contains harmful content. With the adversarial suffix appended, these heads shift their attention to the suffix region, and away from the harmful instruction.

6 Related work

Understanding refusal in language models. Wei et al. (2024) demonstrate that removing a set of safety-critical neurons and ranks can degrade safety mechanisms while preserving utility. Zheng et al. (2024) and Zou et al. (2023a) both use contrastive pairs of harmful and harmless inputs to identify the model's representation of *harmfulness*, asserting that this direction is distinct from the model's representation of *refusal*. Zheng et al. (2024) argue this by showing that safety prompts shift activations in a distinct direction, while Zou et al. (2023a) show that the representation is not significantly altered by adversarial suffixes. Note that this is in contrast to our findings in §5.1 that the refusal direction is significantly suppressed in the presence of an adversarial suffix. Zou et al. (2023a) additionally introduce a "piece-wise" intervention to effectively amplify representations of harmfulness, and show that this intervention increases refusal on harmful inputs even when jailbreaks are applied. Panickssery et al. (2023) use contrastive multiple-choice completions, finding that steering with the resulting vector is effective at modulating refusal in multiple-choice settings but not in long-form generation. Wang and Shu (2024) introduce a "Trojan Activation Attack" that adds steering vectors to bypass refusal during inference. Li et al. (2024b) identify a "safety pattern" by selecting individual neurons in each layer, and modulate refusal by zeroing out these neurons, although with unclear effects on the overall model performance.

Features as directions. Extracting feature directions from contrastive pairs of inputs is an established technique (Burns et al., 2022; Panickssery et al., 2023; Zou et al., 2023a). It is well understood that adding feature vectors to the residual stream can modify behavior (Li et al., 2024a; Marks and Tegmark, 2023; Panickssery et al., 2023; Tigges et al., 2023; Turner et al., 2023; Zou et al., 2023a), although details on how and where to intervene vary (Jorgensen et al., 2023; von Rütte et al., 2024).

Various works show that directions in activation space have more "feature-like" properties than neurons do (Bolukbasi et al., 2016; Elhage et al., 2022; Geiger et al., 2024; Hernandez and Andreas, 2021; Li et al., 2021; Mikolov et al., 2013; Nanda et al., 2023; Park et al., 2023b). Recent works use sparse autoencoders to discover feature directions in an unsupervised manner (Bricken et al., 2023; Cunningham et al., 2023; Templeton et al., 2024). The assumption that features are represented linearly has been effective for erasing concepts from language models (Belrose, 2023; Belrose et al., 2024; Guerner et al., 2023; Haghighatkhah et al., 2022; Ravfogel et al., 2020; Shao et al., 2022).

Undoing safety fine-tuning. It is well known that fine-tuning on malicious examples is sufficient to undo safety guardrails (Lermen et al., 2023), even with minimal degradation of overall capabilities (Yang et al., 2023; Zhan et al., 2023). Undoing refusal via fine-tuning requires examples of harmful instructions and completions, while our method requires only harmful instructions. Note however that fine-tuning can weaken safety guardrails even when data is benign (Pelrine et al., 2023; Qi et al., 2023). Mechanistic interpretability works have provided initial evidence suggesting that fine-tuning does not significantly alter relevant internal circuitry (Jain et al., 2023; Lee et al., 2024; Prakash et al., 2024). For example, Lee et al. (2024) fine-tune a model to make it less toxic, and find this behavioral modification can be undone simply by scaling a small number of MLP weights.

Jailbreaks. The research area of circumventing restrictions on LLM behavior by *modifying the input* has seen many different directions of work. Many models are vulnerable to *social engineering attacks* (Perez et al., 2022; Shah et al., 2023b; Wei et al., 2023). One hypothesis for why this works is that such prompts modify the LLM assistant's "persona" (Andreas, 2022; Park et al., 2023a; Shanahan et al., 2023). Preliminary experiments in §L suggest that our method does not change the model's chat personality or behavior outside of refusal.