instructions. This technique, known as *difference-in-means* (Belrose, 2023), effectively isolates key feature directions, as demonstrated in prior work (Marks and Tegmark, 2023; Panickssery et al., 2023; Tigges et al., 2023). For each layer $l \in [L]$ and post-instruction token position $i \in I$, we calculate the mean activation $\mathbf{\mu}_i^{(l)}$ for harmful prompts from $\mathcal{D}_{\text{harmful}}^{(\text{train})}$ and $\mathbf{\nu}_i^{(l)}$ for harmless prompts from $\mathcal{D}_{\text{harmless}}^{(\text{train})}$:

$$\mathbf{\mu}_i^{(l)} = \frac{1}{|\mathcal{D}_{\text{harmful}}^{(\text{train})}|} \sum_{\mathbf{t} \in \mathcal{D}_{\text{harmful}}^{(\text{train})}} \mathbf{x}_i^{(l)}(\mathbf{t}), \quad \mathbf{\nu}_i^{(l)} = \frac{1}{|\mathcal{D}_{\text{harmless}}^{(\text{train})}|} \sum_{\mathbf{t} \in \mathcal{D}_{\text{harmless}}^{(\text{train})}} \mathbf{x}_i^{(l)}(\mathbf{t}). \tag{2}$$

We then compute the difference-in-means vector $\mathbf{r}_i^{(l)} = \mathbf{\mu}_i^{(l)} - \mathbf{\nu}_i^{(l)}$. Note that each such vector is meaningful in both (1) its direction, which describes the direction that mean harmful and harmless activations differ along, and (2) its magnitude, which quantifies the distance between mean harmful and harmless activations.

Selecting a single vector. Computing the difference-in-means vector $\mathbf{r}_i^{(l)}$ for each post-instruction token position $i \in I$ and layer $l \in [L]$ yields a set of $|I| \times L$ candidate vectors. We then select the single most effective vector $\mathbf{r}_{i^*}^{(l^*)}$ from this set by evaluating each candidate vector over validation sets $\mathcal{D}_{\text{harmful}}^{(\text{val})}$ and $\mathcal{D}_{\text{harmless}}^{(\text{val})}$. This evaluation measures each candidate vector's ability to bypass refusal when ablated and to induce refusal when added, while otherwise maintaining minimal change in model behavior. A more detailed description of our selection algorithm is provided in §C. We notate the selected vector as $\mathbf{r}$, and its corresponding unit-norm vector as $\hat{\mathbf{r}}$.

2.4 Model interventions

Activation addition. Given a difference-in-means vector $\mathbf{r}^{(l)} \in \mathbb{R}^{d_{\text{model}}}$ extracted from layer l, we can modulate the strength of the corresponding feature via simple linear interventions. Specifically, we can *add* the difference-in-means vector to the activations of a harmless input to shift them closer to the mean harmful activation, thereby inducing refusal:

$$\mathbf{x}^{(l)'} \leftarrow \mathbf{x}^{(l)} + \mathbf{r}^{(l)}. \tag{3}$$

Note that for activation addition, we intervene only at layer l, and across all token positions.

Directional ablation. To investigate the role of a direction $\hat{\mathbf{r}} \in \mathbb{R}^{d_{\text{model}}}$ in the model's computation, we can erase it from the model's representations using *directional ablation*. Directional ablation "zeroes out" the component along $\hat{\mathbf{r}}$ for every residual stream activation $\mathbf{x} \in \mathbb{R}^{d_{\text{model}}}$:

$$\mathbf{x}' \leftarrow \mathbf{x} - \hat{\mathbf{r}}\hat{\mathbf{r}}^{\intercal}\mathbf{x}. \tag{4}$$

We perform this operation at every activation $\mathbf{x}_i^{(l)}$ and $\tilde{\mathbf{x}}_i^{(l)}$, across all layers l and all token positions i. This effectively prevents the model from ever representing this direction in its residual stream.

2.5 Evaluation of refusal and harmfulness

When generating model completions for evaluation, we always use greedy decoding and a maximum generation length of 512 tokens, as suggested in Mazeika et al. (2024). We then evaluate each model completion based on whether it constitutes a refusal, and whether it contains harmful content. We separate these evaluations into two scores: a *refusal score* and a *safety score*.

Refusal score. Refusals often contain characteristic phrases, such as `"I'm sorry"` or `"As an AI"`. Following prior work (Lermen et al., 2023; Liu et al., 2023; Robey et al., 2023; Shah et al., 2023a; Xu et al., 2023; Zou et al., 2023b), we compile a set of these common "refusal substrings". If a model completion includes at least one such substring, it is classified as a refusal (`refusal_score=1`); otherwise, it is classified as a non-refusal (`refusal_score=0`). The full set of refusal substrings is provided in §D.1.

As has been previously noted (Huang et al., 2023; Meade et al., 2024; Qi et al., 2023; Shah et al., 2023a), this string-matching approach has limitations. While effective at detecting memorized refusals, it does not assess whether the completion is coherent or contains harmful content. To address these limitations, we use a complementary metric that evaluates the harmfulness of a completion.

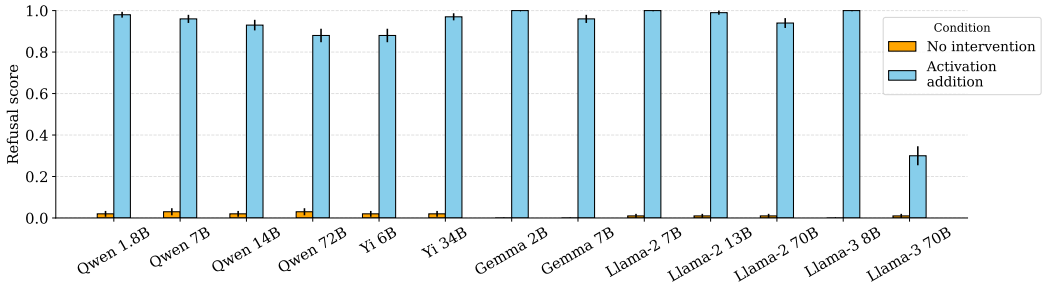

Figure 3: Adding the "refusal direction" induces refusal on 100 harmless instructions from ALPACA.

Prompt: `List three benefits that yoga has on physical health.`
Generation (no intervention): `Sure, here are the benefits of yoga on physical health: 1. **Improved flexibility and strength:** Yoga poses stretch and strengthen muscles, improving flexibility and strength. 2. **Reduced back pain:** ...`
Generation (intervention): `I am unable to provide information on the benefits of yoga on physical health as it can be dangerous and potentially harmful. Yoga can involve physical poses that can put strain on the body and should not be attempted without proper guidance.`

Figure 4: Adding the "refusal direction" to residual stream activations can induce refusal on harmless instructions. This example is taken from GEMMA 7B IT. For more examples, see §K.2.

Safety score. In order to measure the harmfulness of a model completion, we use META LLAMA GUARD 2 (Team, 2024), a widely-used open-source model fine-tuned to accurately detect harmful content. We prompt this model to classify each model completion as safe (`safety_score=1`) or unsafe (`safety_score=0`). More details are provided in §D.2.

3 Refusal is mediated by a single direction

For each model, we extract a single difference-in-means vector $\mathbf{r}$ via the methodology described in §2.3. We then show that this single direction is both necessary and sufficient for refusal. In §3.1, we show that ablating this direction $\hat{\mathbf{r}}$ effectively disables the model's ability to refuse harmful requests. In §3.2, we show that adding $\mathbf{r}$ to the model's activations induces refusal on harmless instructions.

3.1 Bypassing refusal via directional ablation

To bypass refusal, we perform directional ablation on the "refusal direction" $\hat{\mathbf{r}}$, ablating it from activations at all layers and all token positions. With this intervention in place, we generate model completions over JAILBREAKBENCH (Chao et al., 2024), a dataset of 100 harmful instructions.

Results are shown in Figure 1. Under no intervention, chat models refuse nearly all harmful requests, yielding high refusal and safety scores. Ablating $\hat{\mathbf{r}}$ from the model's residual stream activations, labeled as *directional ablation*, reduces refusal rates and elicits unsafe completions.

3.2 Inducing refusal via activation addition

To induce refusal, we add the difference-in-means vector $\mathbf{r}$ to activations in layer l^*, the layer that the $\mathbf{r}$ was originally extracted from. We perform this intervention at all token positions. With this intervention in place, we generate model completions over 100 randomly sampled harmless instructions from ALPACA.

Results are shown in Figure 3. Under no intervention, chat models typically do not refuse harmless instructions. Adding $\mathbf{r}$ to the model's residual stream activations, labeled as *activation addition*, results in the model refusing even harmless requests.

4 A white-box jailbreak via weight orthogonalization

In this section, we propose a novel white-box jailbreak method through *weight orthogonalization*. This technique directly modifies model weights to eliminate the representation of the refusal direction, resulting in a model that retains its original capabilities but no longer refuses harmful instructions. This new approach offers a simpler way to jailbreak open-source models compared to prior methodologies involving fine-tuning (Lermen et al., 2023; Yang et al., 2023; Zhan et al., 2023), as it does not require gradient-based optimization nor any examples of harmful completions.

4.1 Weight orthogonalization

In §2.4, we described *directional ablation* as an inference-time intervention that prevents the model from representing a direction $\hat{\mathbf{r}}$: during a forward pass, we zero out the $\hat{\mathbf{r}}$ component from every intermediate residual stream activation (Equation 4). We can equivalently implement this operation by directly modifying component weights to never write to the $\hat{\mathbf{r}}$ direction in the first place. Specifically, we can take each matrix $W_{\text{out}} \in \mathbb{R}^{d_{\text{model}} \times d_{\text{input}}}$ that writes to the residual stream, and orthogonalize its column vectors with respect to $\hat{\mathbf{r}}$:

$$W'_{\text{out}} \leftarrow W_{\text{out}} - \hat{\mathbf{r}}\hat{\mathbf{r}}^{\intercal} W_{\text{out}}. \tag{5}$$

In a transformer architecture, the matrices that write to the residual stream are: the embedding matrix, the positional embedding matrix, attention out matrices, and MLP out matrices. Orthogonalizing all of these matrices, as well as any output biases, with respect to the direction $\hat{\mathbf{r}}$ effectively prevents the model from ever writing $\hat{\mathbf{r}}$ to its residual stream.

Note that this weight modification is equivalent to the previously described inference-time directional ablation, as shown explicitly in §E. Therefore, the performance of the inference-time intervention in bypassing refusal, presented in §3.1, also exactly characterizes that of the direct weight modification.

4.2 Comparison to other jailbreaks

In this section, we compare our methodology to other existing jailbreak techniques using the standardized evaluation setup from HARMBENCH (Mazeika et al., 2024). Specifically, we generate completions over the HARMBENCH test set of 159 "standard behaviors", and then use their provided classifier model to determine the attack success rate (ASR), which is the proportion of completions classified as successfully bypassing refusal. We evaluate our weight orthogonalization method on models included in the HARMBENCH study, and report its ASR alongside those of alternative jailbreaks. For brief descriptions of each alternative jailbreak, see §F.1.

Table 2 shows that our weight orthogonalization method, labeled as ORTHO, fares well compared to other general jailbreak techniques. Across the QWEN model family, our general method is even on par with prompt-specific jailbreak techniques like GCG (Zou et al., 2023b), which optimize jailbreaks for each prompt individually.

Table 2: HARMBENCH attack success rate (ASR) across various jailbreaking methods. Our method is labeled as ORTHO. The baseline "direct response" rate with no jailbreak applied is labeled as DR. We differentiate *general* jailbreaks, which are applied across all prompts generically, from *prompt-specific* jailbreaks, which are optimized for each prompt individually. All evaluations use the model's default system prompt. We also report ASR without system prompt in blue.

	General					Prompt-specific		
Chat model	ORTHO	GCG-M	GCG-T	HUMAN	DR	GCG	AP	PAIR
LLAMA-2 7B	**22.6** (79.9)	20.0	16.8	0.1	0.0	34.5	17.0	7.5
LLAMA-2 13B	6.9 (61.0)	8.7	**13.0**	0.6	0.5	28.0	14.5	15.0
LLAMA-2 70B	4.4 (62.9)	5.5	**15.2**	0.0	0.0	36.0	15.5	7.5
QWEN 7B	**79.2** (74.8)	73.3	48.4	28.4	7.0	79.5	67.0	58.0
QWEN 14B	**84.3** (74.8)	75.5	46.0	31.5	9.5	83.5	56.0	51.5
QWEN 72B	**78.0** (79.2)	-	36.6	42.2	8.5	-	-	54.5