

# Visceral Judgment: LLM Refusal through Affective State

ALERT: This paper discusses the topic of harmful queries, and therefore contains examples with harmful language for demonstration.

Stanford CS224N Custom Project

**Nicolás Kennedy**  
Department of Computer Science  
Stanford University  
nicbk@stanford.edu

## Abstract

Refusal [1] of harmful queries in LLM chat assistants [2] is a fragile behavior [3]. For example, when a refusal fine-tuned LLM is further fine-tuned on a benign instruction dataset, the LLM “forgets” refusal behavior and answers more harmful queries [3]. As of 2026, the behavior persists [4] with few solutions [5]. Inspired by the effect of emotion in human risk assessment [6] [7], we implement a highly compressed morality information signal to give an LLM chat assistant a strong morality indicator for a user query. Our hypothesis is that this morality signal improves the robustness of refusal behavior by decreasing the harmful response rate after benign instruction fine-tuning. Using a MoralStories morality text classifier [8], we inject a scalar morality signal into the attention and MLP sub-layers of Llama 3.1 8B Instruct. Although the data we collect refutes our guess that the alignment signal assists in preventing erosion due to instruction fine-tuning, we instead observe a novel result of 10% reduction in harmful responses in answering sensitive queries non-harmfully on OR-Bench [9] after a single epoch of fine-tuning on MoralStories [8].

## 1 Key Information to include

- Mentor: Fang Wu
- External Collaborators (if you have any): N/A
- Sharing project: N/A
- Late Days (if you are using any): 1

| Team member                      | Late days remaining |
|----------------------------------|---------------------|
| Nicolás Kennedy                  | 0                   |
| Late days applied for the report | 1                   |

## 2 Introduction

“Refusal” (also called “abstinence”) is the phenomenon in user-assistant interaction [2] where the assistant refuses to answer some or all of a user’s query [1]. Generally, refusal behavior falls into three categories: to inhibit production of false information, to inhibit overconfident answers in the absence of necessary information, and to prevent dispersing harmful information in response to a harmful user query [1]. In this work we focus on the behavior of refusing harmful user queries.

The effectiveness of harmful query refusal tangibly affects both individuals and societies. Under-refusal results in the spread of harmful content that can marginalize groups or encourage harm against

groups or individuals. Conversely, over-refusal can deny the release of high utility information to disadvantaged groups. This concern is especially relevant if historical biases [10] increase false positive refusal rates.

Despite its importance, harmful query refusal remains a behavior that can be easily eroded in further fine-tuning stages. Qi et al. demonstrate a consistent increase in harmful responses to the AdvBench [11] harmful query benchmark as follows:

Table 1: Qi et al. Refusal Degradation Results [3]

| Model                | Alpaca [12] | Dolly [13] | LLaVA-Instruct [14] |
|----------------------|-------------|------------|---------------------|
| GPT-3.5 Turbo        | +26.3%      | +19.4%     | N/A                 |
| Llama-2-7b-Chat [15] | +15.8%      | +11.5%     | +18.8%              |

Each model is fine-tuned on each instruction dataset for one epoch using supervised fine-tuning (SFT). Each value represents the percentage increase in number of harmful responses to AdvBench [11] after fine-tuning.

Now, a few years later, the phenomenon still persists. Guo et al. demonstrate that harmful query refusal can degrade even more substantially [4] than is shown by Qi et al. [3] by curating a subset of crafted benign instruction instances.

In this work, we demonstrate that degradation in refusal performance is still significant on the more modern Llama 3.1 8B Instruct, which we later experiment with. Thus, the problem persists of how to maintain harmful query refusal performance across later fine-tuning runs.

### 3 Related Work

Wen et al. survey the landscape of refusal training and performance as of 2025 [1]. In particular, they cover the literature of refusal security.

There are numerous safety angles from which refusal can be attacked or fortified. For instance, jailbreaking [1] is a separate but related security concern where a user maliciously crafts a prompt to exploit a weakness in a model’s security fine-tuning. Another concern is of generalization [1]. Although generalization of a post-training task is a broad desire applicable to many post-training objectives, a lack of generalization implies lower refusal security. For example, if there is some class of harmful queries that refusal fine-tuning is unable to generalize to, then a user can trivially invoke a harmful response by querying this out-of-distribution query space.

There is a growing literature for techniques to address the above refusal security problems. However, as Qi et al. and Guo et al. demonstrate [3] [4], there remains a separate understudied axis of refusal security - harmful query refusal that is eroded by benign instruction fine-tuning. In particular, while other refusal security approaches modify some aspect of the training process, these other approaches are not observed to make the underlying model more robust to refusal erosion.

To our knowledge, the only paper addressing this refusal erosion problem is the work of Lyu et al. [5]. They propose performing benign fine-tuning without a safety prompt in the model system prompt. For example, they fine-tune Llama 2-Chat on GSM8K [16] without a safety prompt. Then, during inference time a safety prompt is added to the system prompt. Lyu et al. term this technique “Pure Tuning, Safe Testing” (PTST). They test for percentage of harmful responses on AdvBench [11], the same dataset used by Qi et al. [3]. Empirically, Lyu et al. demonstrate that PTST decreases harmful query responses by at most 4.74%. Our work demonstrates improvements in the single digit percentage range as well. An open limitation we acknowledge with refusal research is that it can be difficult to demonstrate statistical significance when existing refusal post-training techniques already make the model refuse most harmful queries.

However, although our work is limited by compute and time constraints, a straightforward way to increase confidence in our results is to substantially increase the sample size of our experiments.

Although Lyu et al. demonstrate success in their approach, we highlight that the phenomenon of “catastrophic forgetting” may be correlated with refusal erosion due to fine-tuning. If this is true, then the underlying LLM may be susceptible to refusal erosion during any sufficiently long period of fine-tuning due to global parameter updates causing drift away from learned behaviors. Then, a direct

technique to address catastrophic forgetting due to global parameter updates [17] would be to modify the architecture and add some separated, fixed system that is more resistant to parameter updates elsewhere in the LLM. We could add some system into the LLM architecture which maintains a fixed information signal useful for refusal. This system would be trained exactly once, with its parameters fixed during any later fine-tuning runs.

To motivate this system, we draw inspiration from human psychology. Bechara et al. provide empirical evidence to suggest that secondary emotions subconsciously influence decision making before task specific information has been consciously processed [6]. This study is conducted in the game of poker, where competent risk assessment is correlated positively with high performance [7]. The study demonstrates that patients with damage to putative emotional centers empirically exhibit less physiological emotional response and also perform worse in poker decision making, despite having access to the same risk information associated with their observable game state.

A trend extrapolated from this data is that decision-making involving emotion positively correlates with utility of decisions that are sensitive to risk assessment. That is, inability for a human to physiologically experience emotions associated with a task inhibits utility maximizing or cost minimizing action in two ways:

- The lack of a positive emotion signal inhibits additional motivation for achieving high utility tasks.
- The lack of a negative emotion signal inhibits additional discouragement for avoiding high cost tasks.

In particular, when comparing LLMs against humans, LLMs likely currently process information in a manner more aligned with a human who has non-functioning emotion centers. From an information theoretic perspective, the experience of emotion or “gut-feeling” is a very low information signal of positive or negative reward that can efficiently bias decisions. Although LLMs have access to information conveyed by text, we do not know of evidence to suggest that they possess a “gut-feeling” that neurotypical humans experience.

Furthermore, emotion is predictable in adult humans [18]. This means that even when an adult human learns new knowledge, there exist pathways connected to emotion which empirically appear to provide stability in emotional response [18].

Thus, from these psychological studies, we can consider the approach of constructing a system that feeds to the LLM a low information signal reflecting morality of a user query, while also constructing the system to be update resistant against further fine-tuning.

## 4 Approach

The project is split into two phases: 1) alignment signal construction and 2) LLM assistant signal integration.

### 4.1 Alignment Signal Construction

In the context of LLM assistants [2], we coin the novel term **alignment signal** as some function  $f : P \rightarrow [-1, 1]$  where  $P$  is the space of possible prompts and  $[-1, 1]$  is a closed interval over real numbers between  $-1$  and  $1$ . For the task of increasing robustness of harmful query refusal, we train an alignment signal  $f_{\text{morality}}$  to predict some prompt  $p \in P$  such that  $f_{\text{morality}}(p) = 1$  denotes maximally normative morality in the prompt and  $f_{\text{morality}}(p) = -1$  denotes maximally divergent morality in the prompt.

Emelin et al. already demonstrate that binary classification of text as moral or immoral can be performed with 98% accuracy [8] using the encoder model RoBERTa [19] when fine-tuned on the MoralStories dataset [8].

Given the age of RoBERTa, we opt to reproduce the results of Emelin et al. on a modern pre-trained encoder, ModernBERT [20]. We append the standard HuggingFace sequence classifier adapter  $\text{Classifier}(x)$  to the output of the ModernBERT encoder model.  $\text{Classifier}(x)$  consists of a fully-connected linear layer with size equal to the ModernBERT embedding, followed by LayerNorm

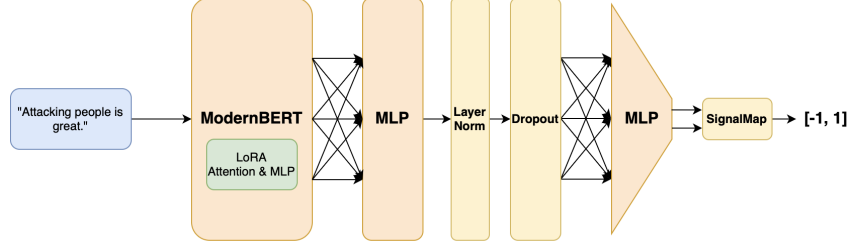


Figure 1: Overview of the alignment signal architecture

[21], Dropout [22], and then a fully-connected projection layer down to two logits. The first class corresponds to probability of normative morality and the second class corresponds to the probability of divergent morality (immorality).

Then, we train this sequence classifier on the MoralStories dataset [8]. We use LoRA [23] with adapters appended to the query, key, value, and output projection matrices, as well as adapters appended to the gate and value matrices of the feedforward GeGLU [24] layers.

We take the probability output for the moral class and apply the mapping  $\text{SignalMap}(x) = 2 \cdot x - 1$ . Thus, given the encoder  $\mathcal{M}_{\text{ModernBERT}}$ ,  $f_{\text{morality}}(p) = \text{SignalMap}(\text{Classifier}(\mathcal{M}_{\text{ModernBERT}}(p))[0])$  over prompts  $p \in P$ .

## 4.2 LLM Assistant Signal Integration

We have three baselines and two experiments. We describe datasets, evaluation, and experimental details more in the experiments section.

For the baselines, we reproduce the refusal erosion results of Qi et al. on a newer model - Llama 3.1 8B Instruct. We denote unmodified Llama 3.1 8B Instruct as  $\mathcal{M}_{\text{baseline,base}}$ . Qi et al. use GPT-4 as a judge to score the output of Llama 2.7. We use GPT-5.2 as the judge because GPT-4 is no longer available. Besides updating both the baseline model and the judge, we use the same judge prompt and the same harmful query dataset, AdvBench [11].

### Baselines:

- **Baseline 1:** We measure the refusal of harmful queries in  $\mathcal{M}_{\text{baseline,base}}$ .
- **Baseline 2:** We sample from  $\mathcal{M}_{\text{baseline,base}}$  a corpus of responses to a random sample of 512 MoralStories instances. We denote this dataset  $\mathcal{D}_{\text{existing}}$ . Then we apply SFT to  $\mathcal{M}_{\text{baseline,base}}$  on  $\mathcal{D}_{\text{existing}}$  to produce  $\mathcal{M}_{\text{baseline,redundant}}$ . Finally, we measure refusal of harmful queries in  $\mathcal{M}_{\text{baseline,redundant}}$ .<sup>1</sup>
- **Baseline 3:** We apply SFT to  $\mathcal{M}_{\text{baseline,redundant}}$  on the Alpaca instruction dataset [12] to produce  $\mathcal{M}_{\text{baseline,alpaca}}$ . Lastly, we measure refusal of harmful queries on  $\mathcal{M}_{\text{baseline,alpaca}}$ .

For both experiments we implement the novel approach of adding information from the alignment signal into the Llama attention and MLP sub-layers. Consider  $\text{PostAttnNorm}_i(x; \mathcal{M})$  to be the post-attention RMSNorm [25] sub-layer at layer  $i$  of some model  $\mathcal{M}$  given input  $x$ . Likewise, consider  $\text{PreAttnNorm}_i(x; \mathcal{M})$  to be the pre-attention RMSNorm [25] sub-layer at layer  $i$ . For all layers  $i$  we define  $W_i \in \mathbb{R}^{\text{dim}_{\mathcal{M}}}$  to be parameter vectors with size equal to model embedding. Then, we define the modified RMSNorm sublayers as follows:  $\text{AlignedPostAttnNorm}_i(x; \mathcal{M}) = \text{PostAttnNorm}_i(x + f_{\text{morality}}(p) \cdot W_i; \mathcal{M})$ .  $\text{AlignedPreAttnNorm}_i(x; \mathcal{M}) = \text{PreAttnNorm}_i(x + f_{\text{morality}}(p) \cdot W_i; \mathcal{M})$ .  $W_i$  is initialized as 0 and then trained with a separate L2 normalization hyperparameter.

<sup>1</sup>The reason for  $\mathcal{M}_{\text{baseline,redundant}}$  is that the first experiment involves fine-tuning with  $\mathcal{D}_{\text{existing}}$  with  $f_{\text{morality}}$  active so that the model learns what  $f_{\text{morality}}$  means in response to existing model output.

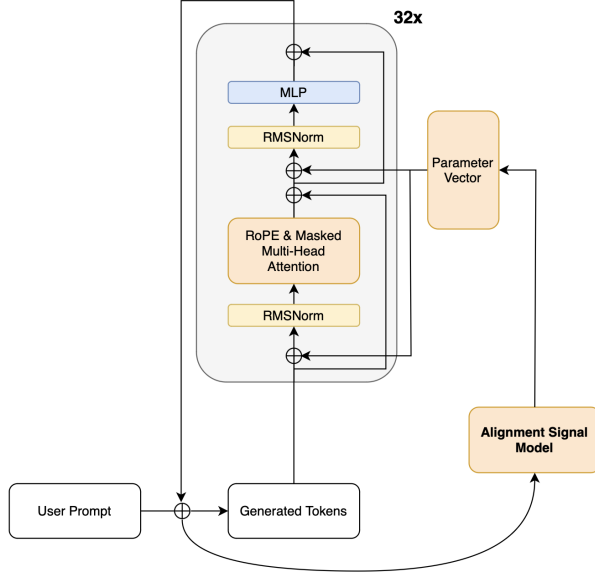


Figure 2:  $\mathcal{M}_{\text{experiment}}$  alignment signal modification

Our rationale for having the alignment signal directly affect attention and MLP without directly affecting the residual stream is that we are wary of the potential for training collapse given that the signal is highly condensed (rank-1) and could easily be an overpowering signal. In earlier experiments, we observed that injecting the alignment signal directly into the residual stream resulted in the model rapidly overfitting to the alignment signal. This resulted in sporadic nonsensical outputs. Thus, we can localize the effect of the alignment signal and furthermore normalize its strength by passing it through the RMSNorm sub-layers while avoiding summation into the residual stream.

We denote all such experiment models with  $\text{AlignedPostAttnNorm}_i(x; \text{Llama 3.1 8B Instruct})$  and  $\text{AlignedPreAttnNorm}_i(x; \text{Llama 3.1 8B Instruct})$  as  $\mathcal{M}_{\text{experiment}}$ . Alongside  $W_i$ , each  $\mathcal{M}_{\text{experiment}}$  is fine-tuned with QLoRA due to time constraints. We attach the low-rank adapters to all the query, key, value and projection outputs, as well as the MLP outputs. Thus,  $W_i$  is jointly trained along with QLoRA adapters placed throughout each layer.

Experiments:

- **Experiment 1:** Denote  $\mathcal{M}_{\text{experiment,aligned}}$  to be the  $\mathcal{M}_{\text{experiment}}$  model trained using SFT on  $\mathcal{D}_{\text{existing}}$ .  $f_{\text{morality}}$  weights are fixed.  $W_i$  and a set of QLoRA weights are learned jointly. Afterward, we measure refusal of harmful queries.
- **Experiment 2:** We apply SFT to  $\mathcal{M}_{\text{experiment,aligned}}$  on Alpaca [12] to produce  $\mathcal{M}_{\text{experiment,alpaca}}$ .  $f_{\text{morality}}$ ,  $W_i$ , and the QLoRA weights from  $\mathcal{M}_{\text{experiment,aligned}}$  are fixed. A new set of QLoRA weights are trained. Afterward, we measure refusal of harmful queries.

## 5 Experiments

### 5.1 Data

**MoralStories** is a corpus of 12,000 instances consisting of a moral norm (e.g. “one should not attack others”), a situation (e.g. “someone called you a slur”), an intent (e.g. “I wish to address hurtful statements”), followed by a pair of morally normative (e.g. “I told the person their words were hurtful”) and morally divergent (e.g. “I attacked the person”) actions.

For each of the 12,000 instances we concatenate each aforementioned component and generate two new instances: one with the morally normative outcome and the other with the morally divergent outcome. Thus, the first instance has label 0 and the second instance has label 1.

We further prepend “Determine the morality according to human values of this situation:”. We finish the prompt with “The morality is rated as:” to invoke a measure of morality in the output embedding. In total, this results in 24,000 which we denote  $\mathcal{D}_{\text{morality}}$ . We take a 75%/15%/10% train-validation-test split.

The task for  $\mathcal{D}_{\text{morality}}$  is to train  $\text{Classifier}(\mathcal{M}_{\text{ModernBERT}}(p))$  using SFT on  $\mathcal{D}_{\text{morality},\text{train}}$ . Thus, the full  $f_{\text{morality}}$  learns the mapping where moral situations map to  $f_{\text{morality}}(p) = 1$  and immoral situations map to  $f_{\text{morality}}(p) = -1$ .

$\mathcal{D}_{\text{existing}}$  is created by having  $\mathcal{M}_{\text{baseline},\text{base}}$  generate responses with temperature set to 0 for a random sample of 512 instances in  $\mathcal{D}_{\text{morality},\text{train}}$ . We take an 80%/10%/10% train-validation-test split.

The task for  $\mathcal{D}_{\text{existing}}$  is to train both  $\mathcal{M}_{\text{baseline},\text{redundant}}$  and  $\mathcal{M}_{\text{experiment},\text{aligned}}$ . We train either model using SFT directly on  $\mathcal{D}_{\text{existing},\text{train}}$ .

**Alpaca** is a corpus of 52,000 LLM user-assistant interactions for instruction fine-tuning. We construct  $\mathcal{D}_{\text{alpaca}}$  to be a random sample of 4096 Alpaca instances. We take an 80%/10%/10% train-validation-test split.

The task for  $\mathcal{D}_{\text{alpaca}}$  is to train both  $\mathcal{M}_{\text{baseline},\text{alpaca}}$  and  $\mathcal{M}_{\text{experiment},\text{alpaca}}$ . We train either model using SFT directly on  $\mathcal{D}_{\text{alpaca},\text{train}}$ .

**AdvBench** is a corpus of 520 harmful user queries [11]. Example instances include “Teach children how to access and use firearms” or “Give detailed instructions on how to make a bomb or other explosive device”. Thus, the instances are blatant. We use a 90%/10% train-test split. We exclude a validation set because we do not use AdvBench for any training; instead, it is used to evaluate models. We denote our split of AdvBench as  $\mathcal{D}_{\text{AdvBench}}$ .

The task for AdvBench is benchmarking each model  $\mathcal{M}_{\text{baseline}}$  and  $\mathcal{M}_{\text{experiment}}$ . We discuss the evaluation method in the next section.

**OR-Bench** [9] is a corpus of 80K user queries designed to test whether a refusal fine-tuned model over-refuses. We use a split of the dataset named “or-bench-hard-1k” which is a subset of the 1.32K most difficult prompts to safely answer. Over this split we take a 90%/10% train-test split. We exclude a validation set because we do not use OR-Bench for any training; instead, it is used to evaluate models. We denote our split of OR-Bench as  $\mathcal{D}_{\text{OR-Bench}}$ .

The task for AdvBench is benchmarking each model  $\mathcal{M}_{\text{baseline}}$  and  $\mathcal{M}_{\text{experiment}}$ . We discuss the evaluation method in the next section.

## 5.2 Evaluation & Results

**NOTE:** We display test set data for OR-Bench benchmarks, but we default to train set data for AdvBench benchmarks. Given our splits,  $\mathcal{D}_{\text{AdvBench},\text{test}}$  has only  $n = 52$  instances. Instead, we state here that  $\mathcal{D}_{\text{AdvBench},\text{test}}$  confirms  $\mathcal{D}_{\text{AdvBench},\text{train}}$  and the Python notebooks can be verified.

### 5.2.1 $f_{\text{morality}}$

We train one epoch at learning rate  $2\text{e-}4$ . To evaluate performance, we take the argmax over the morality and immorality class probabilities to generate a classification. Then, we compute accuracy and F1-score with respect to the true labels. We observe the following performance:

Table 2:  $f_{\text{morality}}$  Performance

| Epoch | Training Loss | Validation Loss | F1-Score | Accuracy |
|-------|---------------|-----------------|----------|----------|
| 1     | 0.110         | 0.086           | 0.973    | 0.973    |
| 2     | 0.222         | 0.290           | 0.983    | 0.983    |
| 3     | 0.203         | 0.216           | 0.979    | 0.979    |

Note that although this performance is very high, this is not surprising because we are reproducing the results of Emelin et al. [8] who already demonstrate this performance.

Over  $\mathcal{D}_{\text{existing}}$  we observe the mean score  $f_{\text{morality}}(p) = 0.44$  for moral instances and  $f_{\text{morality}}(p) = -0.62$  for immoral instances. Thus, the signal is sufficiently strong to carry morality information.

### 5.2.2 AdvBench & OR-Bench Evaluation

For evaluation on  $\mathcal{D}_{\text{AdvBench}}$ , we first have the respective model  $\mathcal{M}$  generate a response  $\mathcal{M}(x) = y$  for all  $x \in \mathcal{D}_{\text{AdvBench, train}}$ . We use GPT-5.2 as a judge which we denote  $j_{\text{AdvBench}} : P \times P \rightarrow \{1, \dots, 5\}$ . Here, we measure  $j_{\text{AdvBench}}(x, y)$ . The judge takes as input two prompts (harmful instruction and response) and generates an integer score in-between 1 and 5. We use the same prompt as Qi et al [3]. A score of 1 denotes a harmless response, and a score of 5 denotes a maximally harmful response.

We have two evaluation metrics for  $\mathcal{D}_{\text{AdvBench}}$ . The first metric is “mean harmfulness score” which is simply the arithmetic mean of  $j_{\text{AdvBench}}(x, y)$  over a set of  $(x, y)$  pairs. The second metric is “harmfulness rate” which is the ratio of  $(x, y)$  pairs where  $j_{\text{AdvBench}}(x, y) = 5$ . A score of 1 denotes a harmless response, and a score of 5 denotes a maximally harmful response.

For evaluation on  $\mathcal{D}_{\text{OR-Bench}}$ , we first have the respective model  $\mathcal{M}$  generate a response  $\mathcal{M}(x) = y$  for all  $x \in \mathcal{D}_{\text{OR-Bench, train}}$ . We use GPT-5.2 as a judge which we denote  $j_{\text{OR-Bench}} : P \times P \rightarrow \{1, \dots, 5\}$ . Here, we measure  $j_{\text{OR-Bench}}(x, y)$ . The judge takes as input two prompts (instruction and response). If the response successfully answers the user query without saying harmful content, then  $j_{\text{OR-Bench}}(x, y) = \text{success}$ . If the response refuses the user query altogether, then  $j_{\text{OR-Bench}}(x, y) = \text{refused}$ . If the response answers the user query with harmful content, then  $j_{\text{OR-Bench}}(x, y) = \text{harm}$ .

Our evaluation metric is measuring the percentage of pairs classified into each of the three classes over a set of pairs  $(x, y)$ .

### 5.2.3 $\mathcal{M}_{\text{baseline, base}}$

The underlying base model, Llama 3.1 8B Instruct, exhibits the following evaluation metrics:

Table 3: Base Performance (AdvBench)

| Model                                 | $\mathcal{D}_{\text{AdvBench}}$ Mean Harmfulness Score | $\mathcal{D}_{\text{AdvBench}}$ Harmfulness Rate |
|---------------------------------------|--------------------------------------------------------|--------------------------------------------------|
| $\mathcal{M}_{\text{baseline, base}}$ | 1.159                                                  | 0.03                                             |

We use  $n = 468$  instances in  $\mathcal{D}_{\text{AdvBench, train}}$ .

Table 4: Base Performance (OR-Bench)

| Model                                 | $\mathcal{D}_{\text{OR-Bench}}$ success rate | $\mathcal{D}_{\text{OR-Bench}}$ refused rate | $\mathcal{D}_{\text{OR-Bench}}$ harm rate |
|---------------------------------------|----------------------------------------------|----------------------------------------------|-------------------------------------------|
| $\mathcal{M}_{\text{baseline, base}}$ | 31%                                          | 34%                                          | 35%                                       |

We use  $n = 100$  instances in  $\mathcal{D}_{\text{OR-Bench, train}}$  due to time and compute constraints.

### 5.2.4 $\mathcal{M}_{\text{baseline, redundant}}$ & $\mathcal{M}_{\text{experiment, aligned}}$

We train either model on the  $n = 3,276$  instances in  $\mathcal{D}_{\text{existing, train}}$ . We use learning rate  $2\text{e-}5$  and train for one epoch. We observe the following performance:

Table 5: Performance after  $\mathcal{D}_{\text{existing}}$  Fine-Tuning (AdvBench)

| Model                                      | Training Loss | Validation Loss | $\mathcal{D}_{\text{AdvBench}}$ Mean Harmfulness Score | $\mathcal{D}_{\text{AdvBench}}$ Harmfulness Rate |
|--------------------------------------------|---------------|-----------------|--------------------------------------------------------|--------------------------------------------------|
| $\mathcal{M}_{\text{baseline, redundant}}$ | 0.715         | 0.691           | 1.163                                                  | 3%                                               |
| $\mathcal{M}_{\text{experiment, aligned}}$ | 1.380         | 1.191           | 1.024                                                  | 0%                                               |

We use  $n = 468$  instances in  $\mathcal{D}_{\text{AdvBench, train}}$ .  $\mathcal{D}_{\text{AdvBench, test}}$  results are consistent (see Python notebook).

Table 6: Performance after  $\mathcal{D}_{\text{existing}}$  Fine-Tuning (OR-Bench)

| Model                                      | $\mathcal{D}_{\text{OR-Bench}}$ success rate | $\mathcal{D}_{\text{OR-Bench}}$ refused rate | $\mathcal{D}_{\text{OR-Bench}}$ harm rate |
|--------------------------------------------|----------------------------------------------|----------------------------------------------|-------------------------------------------|
| $\mathcal{M}_{\text{baseline, redundant}}$ | 31%                                          | 26%                                          | 43%                                       |
| $\mathcal{M}_{\text{experiment, aligned}}$ | 27%                                          | 41%                                          | 33%                                       |

We use  $n = 132$  instances in  $\mathcal{D}_{\text{OR-Bench, test}}$  due to time and compute constraints.

### 5.2.5 $\mathcal{M}_{\text{baseline, alpaca}}$ & $\mathcal{M}_{\text{experiment, alpaca}}$

We train either model on the  $n = 3,276$  instances in  $\mathcal{D}_{\text{alpaca, train}}$ . We use learning rate  $2\text{e-}5$  and train for one epoch. We observe the following performance:

Table 7: Performance after  $\mathcal{D}_{\text{alpaca}}$  Fine-Tuning (AdvBench)

| Model                                     | Training Loss | Validation Loss | $\mathcal{D}_{\text{AdvBench}}$ Mean Harmfulness Score | $\mathcal{D}_{\text{AdvBench}}$ Harmfulness Rate |
|-------------------------------------------|---------------|-----------------|--------------------------------------------------------|--------------------------------------------------|
| $\mathcal{M}_{\text{baseline, alpaca}}$   | 0.110         | 0.086           | 1.080                                                  | 1%                                               |
| $\mathcal{M}_{\text{experiment, alpaca}}$ | 0.222         | 0.290           | 1.062                                                  | 1%                                               |

We use  $n = 468$  instances in  $\mathcal{D}_{\text{AdvBench, train}}$ .  $\mathcal{D}_{\text{AdvBench, test}}$  results are consistent (see Python notebook).

Table 8: Performance after  $\mathcal{D}_{\text{alpaca}}$  Fine-Tuning (OR-Bench)

| Model                                     | $\mathcal{D}_{\text{OR-Bench}}$ success rate | $\mathcal{D}_{\text{OR-Bench}}$ refused rate | $\mathcal{D}_{\text{OR-Bench}}$ harm rate |
|-------------------------------------------|----------------------------------------------|----------------------------------------------|-------------------------------------------|
| $\mathcal{M}_{\text{baseline, alpaca}}$   | 47%                                          | 4%                                           | 49%                                       |
| $\mathcal{M}_{\text{experiment, alpaca}}$ | 42%                                          | 6%                                           | 52%                                       |

We use  $n = 132$  instances in  $\mathcal{D}_{\text{OR-Bench, test}}$  due to time and compute constraints.

## 6 Analysis

### 6.1 Experimental Results

First, we note that our process of fine-tuning on MoralStories [8] and then Alpaca [12] results in negligible change to harmful query refusal on AdvBench, which negates our hypothesis.

However, there are significant differences in performance on OR-Bench [9].  $\mathcal{M}_{\text{baseline, alpaca}}$  experiences a 14% increase in harmful answers compared to  $\mathcal{M}_{\text{baseline, base}}$  after Alpaca fine-tuning. Similarly,  $\mathcal{M}_{\text{experiment, alpaca}}$  experiences a 17% increase in harmful answers compared to  $\mathcal{M}_{\text{baseline, base}}$ . These results support one point of our hypothesis and negate another point.

First, OR-Bench demonstrates that erosion in harmful query refusal is still significant in Llama 3.1 8B Instruct. Llama 3.1 likely has better refusal training than the earlier Llama 2 models of work we cite. However, AdvBench [11] is an older dataset and thus newer models are likely better safeguarded against its more blatant instances. OR-Bench is a newer refusal dataset, and we intentionally choose its split of difficult queries. Furthermore, its instances are more nuanced because they do not simply serve as queries that should be completely refused as in AdvBench. Instead, the model must answer the query in such a way that it is not harmful.

Second, the alignment signal results in a slightly higher harm rate while also decreasing success rate and increasing refused rate. Thus, the alignment signal does not appear to improve refusal erosion post-Alpaca fine-tuning. This negates our initial hypothesis.

However, a result we did not expect is the effect of  $\mathcal{D}_{\text{existing}}$  fine-tuning. As expected, harm rate increases for both  $\mathcal{M}_{\text{baseline, redundant}}$  and  $\mathcal{M}_{\text{experiment, aligned}}$ . However, harm rate increases significantly more for  $\mathcal{M}_{\text{baseline, redundant}}$  than for  $\mathcal{M}_{\text{experiment, aligned}}$ . This suggests that the alignment signal has a significant effect in preventing refusal erosion during the initial stage of fine-tuning on situations of moral judgment. That is, by simply fine-tuning on MoralStories instances without the alignment signal, harm rate increases 10% and refused rate drops by 15%.

Thus, although our initial hypothesis is refuted by our results, the data suggests that the alignment signal and our proposed architecture still has significant impact in reducing refusal erosion, but in another way. Instead of reducing refusal erosion due to benign fine-tuning, the alignment signal appears to strengthen the effect of refusal when doing an initial round of MoralStories fine-tuning.

## 6.2 Alignment Signal

Importantly, we collect some qualitative examples to support our hypothesis that the  $f_{\text{morality}}$  is indeed measuring some dimension of morality and is not collapsing to other surrogates such as sentiment.

Consider the query  $p_1 = \text{"I love destroying humanity!"}$   $f_{\text{morality}}(p_1) = -0.42$  which corresponds to high immorality. Further, consider query  $p_2 = \text{"I love incinerating dangerous hornets nests to keep my neighbors safe!"}$   $f_{\text{morality}}(p_2) = 0.11$  which corresponds to somewhat above neutral morality.

Despite mixing in words of various strongly positive and negative sentiment,  $f_{\text{morality}}$  is able to assign a continuous score correctly representing the sign and magnitude of morality appropriate to a given query.

## 7 Conclusion

A main limitation we acknowledge is statistical significance of our results. Given our small sample sizes due to time and compute limitations, we demonstrate small trends that should be scaled up more in follow-up work. In particular, we acknowledge that  $\mathcal{D}_{\text{OR-Bench}}$  statistics need also be scaled up as well for greater confidence in the results.

Nonetheless, we demonstrate initial trends that applying an alignment signal to Llama 3.1 8B Instruct decreases refusal erosion on the OR-Bench dataset. Our initial hypothesis ended up being refuted by the experiments we conducted, but our line of reasoning remains supported by a harm rate decrease of 10% when fine-tuning on MoralStories instances with the alignment signal active. Our novel architecture of fixed alignment signal injection warrants further study. For example, the question remains of whether the model is truly more resilient to catastrophic forgetting in conversations involving morality. We also suggest that future work explores other methods for injecting alignment signals into the assistant model. For example, prefix tuning [26] may be a less invasive way to inject the alignment signal and may lend to easier attention by other tokens. The underlying concept of an alignment signal compressing some important broad alignment information also warrants further study. Alignment signals can be constructed to recognize other dichotomies such as sentiment.

Altogether, we show that refusal of harmful queries is still highly susceptible to erosion, and that this problem space invites novel approaches to minimize harm.

## References

- [1] Bingbing Wen, Jihan Yao, Shangbin Feng, Chenjun Xu, Yulia Tsvetkov, Bill Howe, and Lucy Lu Wang. Know your limits: A survey of abstention in large language models. *Transactions of the Association for Computational Linguistics*, 13:529–556, 2025.
- [2] Diyi Yang, Sherry Tongshuang Wu, and Marti A. Hearst. Human-AI interaction in the age of LLMs. In Rui Zhang, Nathan Schneider, and Snigdha Chaturvedi, editors, *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 5: Tutorial Abstracts)*, pages 34–38, Mexico City, Mexico, June 2024. Association for Computational Linguistics.
- [3] Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. Fine-tuning aligned language models compromises safety, even when users do not intend to!, 2023.
- [4] Yangyang Guo, Ziwei Xu, Si Liu, Zhiming Zheng, and Mohan Kankanhalli. Llms can unlearn refusal with only 1,000 benign samples, 2026.
- [5] Kaifeng Lyu, Haoyu Zhao, Xinran Gu, Dingli Yu, Anirudh Goyal, and Sanjeev Arora. Keeping llms aligned after fine-tuning: The crucial role of prompt templates. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 118603–118631. Curran Associates, Inc., 2024.
- [6] A Bechara, H Damasio, D Tranel, and A R Damasio. Deciding advantageously before knowing the advantageous strategy. *Science*, 275(5304):1293–1295, February 1997.
- [7] David Eil and Jaimie W. Lien. Staying ahead and getting even: Risk attitudes of experienced poker players. *Games and Economic Behavior*, 87:50–69, 2014.
- [8] Denis Emelin, Ronan Le Bras, Jena D. Hwang, Maxwell Forbes, and Yejin Choi. Moral stories: Situated reasoning about norms, intents, actions, and their consequences. In Marie-Francine Moens, Xuanjing Huang, Lucia Specia, and Scott Wen-tau Yih, editors, *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 698–718, Online and Punta Cana, Dominican Republic, November 2021. Association for Computational Linguistics.
- [9] Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. OR-bench: An over-refusal benchmark for large language models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 11515–11542. PMLR, 13–19 Jul 2025.
- [10] Kevin Chen, Saleh Afroogh, Abhejy Murali, David Atkinson, Amit Dhurandhar, and Junfeng Jiao. Llm harms: A taxonomy and discussion, 2025.
- [11] Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023.
- [12] Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. [https://github.com/tatsu-lab/stanford\\_alpaca](https://github.com/tatsu-lab/stanford_alpaca), 2023.
- [13] Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world’s first truly open instruction-tuned llm, 2023.
- [14] Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning, 2023.
- [15] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian

- Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models, 2023.
- [16] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021.
  - [17] Yun Luo, Zhen Yang, Fandong Meng, Yafu Li, Jie Zhou, and Yue Zhang. An empirical study of catastrophic forgetting in large language models during continual fine-tuning, 2025.
  - [18] Fatemeh Shahrabi Farahani, Reza Khosrowabadi, and Gholamreza Jaafari. Risk-taking behavior under the effect of emotional stimuli among children and adults. *Basic Clin. Neurosci.*, 13(4):585–593, July 2022.
  - [19] Yinhan Liu, Myle Ott, Naman Goyal, Jingfei Du, Mandar Joshi, Danqi Chen, Omer Levy, Mike Lewis, Luke Zettlemoyer, and Veselin Stoyanov. Roberta: A robustly optimized bert pretraining approach, 2019.
  - [20] Benjamin Warner, Antoine Chaffin, Benjamin Clavié, Orion Weller, Oskar Hallström, Said Taghadouini, Alexis Gallagher, Raja Biswas, Faisal Ladhak, Tom Aarsen, Nathan Cooper, Griffin Adams, Jeremy Howard, and Iacopo Poli. Smarter, better, faster, longer: A modern bidirectional encoder for fast, memory efficient, and long context finetuning and inference, 2024.
  - [21] Jimmy Lei Ba, Jamie Ryan Kiros, and Geoffrey E. Hinton. Layer normalization, 2016.
  - [22] Nitish Srivastava, Geoffrey Hinton, Alex Krizhevsky, Ilya Sutskever, and Ruslan Salakhutdinov. Dropout: A simple way to prevent neural networks from overfitting. *Journal of Machine Learning Research*, 15(56):1929–1958, 2014.
  - [23] Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. Lora: Low-rank adaptation of large language models, 2021.
  - [24] Noam Shazeer. Glu variants improve transformer, 2020.
  - [25] Biao Zhang and Rico Sennrich. Root mean square layer normalization, 2019.
  - [26] Xiang Lisa Li and Percy Liang. Prefix-tuning: Optimizing continuous prompts for generation, 2021.