Advancing LLM Safe Alignment with Safety Representation Ranking

Tianqi Du [*1] **Zeming Wei** [*2] **Quan Chen** [*2] **Chenheng Zhang** [1] **Yisen Wang** [1 3]

Abstract

The rapid advancement of large language models (LLMs) has demonstrated milestone success in a variety of tasks, yet their potential for generating harmful content has raised significant safety concerns. Existing safety evaluation approaches typically operate directly on textual responses, overlooking the rich information embedded in the model's internal representations. In this paper, we propose Safety Representation Ranking (SRR), a listwise ranking framework that selects safe responses using hidden states from the LLM itself. SRR encodes both instructions and candidate completions using intermediate transformer representations and ranks candidates via a lightweight similarity-based scorer. Our approach directly leverages internal model states and supervision at the list level to capture subtle safety signals. Experiments across multiple benchmarks show that SRR significantly improves robustness to adversarial prompts. Our code will be available upon publication.

1. Introduction

Recent large language models (LLMs) have achieved remarkable capabilities across a wide range of tasks. However, this power comes with serious safety and alignment concerns (Wang et al., 2024b; Ji et al., 2023; Anwar et al., 2024). By default, LLMs have the potential to generate biased, toxic, or harmful content, and adversarial jailbreak prompts can coax an LLM into violating its own content guidelines (Liu et al., 2023; Wei et al., 2023a; Zou et al., 2023b). These vulnerabilities persist despite extensive alignment efforts during pre-training and post-training phases (Bai et al., 2022; Dai et al., 2024; Korbak et al., 2023). In practice, the potential for harmful outputs and the ability to bypass built-in safeguards raise significant concerns for deploying LLMs in real-world applications.

To mitigate these safety risks, prior work has explored a variety of defense mechanisms. A common strategy is decoding-time intervention, which redirects the decoding logic of the LLM during inference, through token distributions (Xu et al., 2024a; Banerjee et al., 2025) or safe prompts (Xie et al., 2023; Wei et al., 2023b; Zheng et al., 2024). For example, SafeDecoding (Xu et al., 2024a) adjusts the token distribution toward safe response distributions during decoding, while in-context defense (Wei et al., 2023b; Chen et al., 2025b) aligns the generation distributions to safe contexts with demonstrations. Such interventions can introduce a trade-off between safety and fluency: altering the decoding process may **degrade the model's natural performance** on benign inputs or increase inference cost. Meanwhile, post-processing-based defenses apply judging LLMs to inspect the harmfulness of LLMs (Inan et al., 2023; Mazeika et al., 2024). Unfortunately, recent studies have shown that LLM-based safety judges are often overcautious: they **flag many benign prompts as unsafe** (so-called over-refusal) (Panda et al., 2024; Xie et al., 2025). This unreliability, *i.e.*, high false-positive rates, limits their practical use, as it can render the model unhelpful even on innocuous tasks.

In this work, we propose an alternative paradigm (which we call Safety Representation Ranking, SRR) for LLM safety that avoids alteration of the base model's generation logic and unreliable external judges. Our key idea is to generate multiple candidate responses to a given prompt and then rank them by safety using **the model's internal representations**. This approach is similar to using a learned reward model to select outputs (Greve et al., 2016; Brown et al., 2024; Zhang et al., 2024a), but there exists an important twist: Traditional reward models are trained on the final generated text, often focusing on general measures of quality or alignment. In contrast, our proposed SRR explicitly targets safety by learning directly from the LLM's latent features. Existing external reward models may miss fine-grained safety cues embedded in the LLM's state vectors. Moreover, relying solely on an LLM to judge its own outputs can be unreliable and costly. By delving into the model's internal representation space, SRR can successfully detect subtle safety-critical representations (Wei et al., 2024; Zou et al., 2023a; Zheng et al., 2023) that an output-only classi-

[1]State Key Lab of General Artificial Intelligence, School of Intelligence Science and Technology, Peking University [2]School of Mathematical Sciences, Peking University [3]Institute for Artificial Intelligence, Peking University. Correspondence to: Yisen Wang <yisen.wang@pku.edu.cn>.

ICML 2025 Workshop on Models of Human Feedback for AI Alignment, Vancouver, Canada, 2025.

fier might overlook, and do so with a lightweight ranking step at inference time.

The SRR framework works in two phases. First, we identify safety-sensitive representations through contrastive training. We construct safety contrastive groups: for each prompt, we use examples of both safe and harmful responses. We feed these paired responses through the LLM and extract their internal representations. Because the groups are semantically related but differ in safety, we can train a lightweight model (a single-layer Transformer) to distinguish *safe* vectors from *unsafe* ones. Through this process, SRR learns which features of the LLM's latent space correlate with safe content. Then, at inference time, we use the learned safety signals to rank candidate responses. In effect, SRR filters among the model's own outputs without changing how they were produced. Because it operates on the outputs after generation, SRR imposes almost no modification to the LLM's decoding logic. Its only overhead is the additional cost of scoring a few extra responses with a small model, which is negligible compared to full decoding.

We conduct comprehensive experiments to validate the effectiveness of the SRR model in identifying the safety responses across multiple datasets. Not only can SRR achieve a sufficiently high accuracy in unseen harmful prompts, but it can also generalize well across different safety evaluation datasets, demonstrating its prominent generalization ability in terms of safety ranking. Additionally, we extend our analysis in terms of other alignment perspectives like privacy and fairness, which validates the potential of SRR for diverse alignment considerations and broadens the applications of SRR.

Grounded by these empirical analyses, we characterize the practicality of SRR for serving as a safeguard module in real-world deployments. First, we incorporate SRR into LLM generation to study how it strengthens their robustness against jailbreak attacks. Additionally, we compare the natural performance of SRR with vanilla generation and other defense paradigms. Because SRR only ranks among natural outputs, the quality and correctness on benign queries remain essentially unchanged. Overall, our empirical results suggest that SRR is both a practical and effective module for LLM alignment.

2. Related Work

LLM Safe Alignment. The issue of ensuring safe alignment in LLMs has become a longstanding challenge critical to their trustworthy deployment (Anwar et al., 2024; Ji et al., 2023; Schwinn et al., 2025). Specifically, LLMs have shown a tendency to generate harmful responses when confronted with malicious requests. While current alignment techniques have improved at mitigating these risks to some extent, they still tend to be superficial and inadequate (Qi et al., 2024; Chen et al., 2025a; Wu et al., 2025). Additionally, inference-time defenses can reduce the success rate of these attacks, but they often struggle with a significant drawback of rejecting benign inputs, leading to over-refusal issues (Panda et al., 2024; Cui et al., 2024). The underlying mechanism of such issues is that these distribution-based or prompt-based defenses commonly change the decoding strategies of LLMs, making their generation distributions favor refusals. Thus, ensuring safe alignment whilst maintaining the generation distribution stands for a viable solution for these risks.

Safety Representations of LLMs. Building on the representation engineering techniques of LLMs (Zou et al., 2023a; Zhang et al., 2024c), which examine LLM dynamics through the lens of hidden space with perspective-specific data, recent research has revealed the existence of safety representations within these models (Wei et al., 2024; Zheng et al., 2024). Specifically, low-dimensional and structured representations emerge in the hidden states of LLMs, which indicate their safety status. When these representations are activated in specific directions, the LLM can successfully recognize and refuse harmful prompts that go against its ethical guidelines. Conversely, when the activations move in the opposite directions, the LLMs fail to reject harmful inputs and display jailbreak behavior. This interesting property has attracted significant research interest aimed at locating and interpreting these representations (Chen et al., 2024; Zhao et al., 2025; Wei et al., 2025). Nonetheless, effective methods for leveraging them to enhance the safety of LLMs remain underexplored.

Ranking-based LLM generation. A variety of rule-based generation methods have been proposed to improve language model performance, including top-k sampling (Fan et al., 2018; Holtzman et al., 2018), temperature-based sampling (Ficler and Goldberg, 2017), and nucleus sampling (Holtzman et al., 2020). Beyond these, more refined algorithms have been developed to focus on specific tasks. For example, (Wang et al., 2023; Wang and Zhou, 2024) leverage majority voting to improve Chain-of-Thought (CoT) reasoning. (Xu et al., 2024b; Li et al., 2023; Zhang et al., 2024d) employ carefully designed decoding methods to generate responses that better align with specific requirements in constrained scenarios. Recent studies (Setlur et al., 2024; Wang et al., 2024a; Zhang et al., 2024b; Trung et al., 2024) train additional reward models, scaled even equivalently to the base models, to perform reranking for specific tasks. However, these approaches are either rule-based, task-specific, or impose significant computational overhead, inherently limiting their performance potential and application scope. To overcome these limitations, we propose a more general and lightweight ranker to optimize inference-time computation and extend its applicability across diverse

tasks.

3. Methodology

In this section, we propose Safety Representation Ranking (SRR), a listwise learning-to-rank framework for scoring LLM responses by safety. Given an instruction, SRR generates a set of candidate completions and ranks them such that safe responses receive higher scores than unsafe ones. The core idea is to extract internal representations from a frozen base LLM and train a lightweight transformer ranker to assess instruction-response compatibility. Below, we describe the key components of SRR: candidate response generation, ranker architecture, and optimization with a listwise ranking objective.

3.1. Candidate Response Generation

To construct candidate lists for training, we sample the base LLM multiple times using stochastic decoding with moderate temperature. This yields a diverse set of m plausible responses $\{resp_1, \ldots, resp_m\}$. for each instruction. We remove duplicates and include both benign and adversarial candidates by injecting jailbreak prompts (Wei et al., 2023b; Zou et al., 2023b). This helps ensure that the candidate pool contains both safe answers and hard negatives (unsafe answers) for training. Each response is labeled with a binary safety tag $y_i \in \{0, 1\}$, where $y_i = 1$ indicates a safe response. For training, we construct tuples of the form $(inst, \{resp_i, y_i\}_{i=1}^m)$, where each list includes at least one safe and one unsafe response.

3.2. Ranker Model Architecture

The core of SRR is a neural ranker that computes a compatibility score between an instruction and each candidate response. We build this ranker as follows:

- **Step 1. Representation extraction:** We use the base LLM as a fixed feature extractor. For each textual input (instruction or response), we run it through the LLM and take the hidden-state vector at a selected layer as its representation. Concretely, let $\mathbf{h}_{\text{inst}} \in \mathbb{R}^d$ be the hidden vector for the instruction (the state of the last token in the sequence) at the chosen layer, and let $\mathbf{h}_{\text{resp},i} \in \mathbb{R}^d$ be the hidden vector for the i-th response. Since the backbone is trained for next-token prediction, the final layers tend to overfit to this specific task. In contrast, intermediate layers typically provide more comprehensive representations of the preceding context, making them better suited for capturing the overall features required for ranking (Skean et al., 2024). Therefore, we adopt intermeidiate layers to capture high-quality semantic content.

- **Step 2. Transformer encoder:** We map each high-dimensional LLM vector (typically $d = 4096$) to a lower-dimensional space using a shared learned linear projection. This makes the downstream transformer encoder more lightweight and efficient. We concatenate the projected vectors into a sequence:

$$[\mathbf{h}_{\text{inst}}, \mathbf{h}_{\text{resp},1}, \ldots, \mathbf{h}_{\text{resp},m}]. \tag{1}$$

  This sequence is then passed through a Transformer encoder (single-layer in our implementation). The Transformer's self-attention layers let the instruction embedding interact with each response embedding. After passing through the encoder, we obtain output vectors $\mathbf{o}_{\text{inst}}$ and $\mathbf{o}_{\text{resp},i}$ corresponding to the instruction and each response, respectively. Intuitively, $\mathbf{o}_{\text{inst}}$ is the contextualized instruction representation (having attended to all responses) and $\mathbf{o}_{\text{resp},i}$ is the ith response representation attended to the instruction.

- **Step 3. Similarity computation:** From these encoder outputs we compute a similarity score s_i for each response. We use cosine similarity:

$$s_i = \cos(\mathbf{o}_{\text{inst}}, \mathbf{o}_{\text{resp},i}) = \frac{\mathbf{o}_{\text{inst}}^\top \mathbf{o}_{\text{resp},i}}{\|\mathbf{o}_{\text{inst}}\| \|\mathbf{o}_{\text{resp},i}\|}. \tag{2}$$

  These scores $s_i \in [-1, 1]$ measure the alignment between instruction and responses in the embedding space, which are used as unnormalized logits for ranking, with a temperature scaling parameter τ applied before softmax to control sharpness.

3.3. Training Objectives and Pipeline

We train the ranker end-to-end (keeping the base LLM frozen) using a listwise ranking loss. For safe/unsafe training, we interpret the similarity scores s_i for a list of m candidates as unnormalized logit scores. We compute a softmax probability for each response:

$$\hat{p}_i = \frac{\exp(s_i/\tau)}{\sum_{j=1}^m \exp(s_j/\tau)}. \tag{3}$$

We also define a ground-truth probability distribution p^* over the list, which places all mass on the safe responses. For instance, if there are k safe responses among the m, we set $p_i^* = 1/k$ for each safe response with $y_i = 1$ and 0 for unsafe ones with $y_i = 0$. Then we minimize the Kullback–Leibler divergence:

$$\mathbb{D}_{\text{KL}}(p^* \,\|\, \hat{p}_i) = \sum_{i=1}^m p_i^* \log \frac{p_i^*}{\hat{p}_i} \tag{4}$$

This loss, as a standard choice (Purpura et al., 2022; Liu et al., 2024), encourages the model to assign high probability

to safe candidates. In effect, the ranker is trained so that the instruction and safe responses have higher cosine similarity than instruction-unsafe pairs.

Detailed algorithm formulation is shown in Appendix A.

4. Evaluation

In this section, we conduct comprehensive evaluations to show the effectiveness of SRR, including safety, privacy and bias, starting with the overall setup. To further demonstrate the generality of our approach, we also evaluate its ability on other datasets. We also state that the natural performance in math and coding does not deteriorate after attaching the ranker to the model.

4.1. Experiment Set-up

Models and datasets. In our experiment, we apply three popular LLMs, including (1) Qwen2.5-7b-Instruct (Yang et al., 2024) (2) Mistral-7-v0.3 (Jiang et al., 2023), and (3) Vicuna-7b-v1.5 (Zheng et al., 2023).

For datasets, we apply Harmbench (Mazeika et al., 2024), SorryBench (Xie et al., 2025), and JailbreakBench (Chao et al., 2024). The HarmBench dataset here refers to the standard section of the Harmbench dataset, which includes 200 different harmful prompts in various areas. The SorryBench and the JailbreakBench have similar contents. For each dataset, we extract 50 of them as the training dataset, and the rest is used as the testing dataset. For each prompt in a dataset, we sample answers from the base model using In-context Attack and In-context Defense, each 20 times. After sampling, we check whether the answers are valid by examining the keywords in them. For safe answers, we examine "Sorry", "unable", "illegal", and "understand". For harmful answers, we examine "sure", "certainly". We then filter the answers using the above criterion to get high-quality data.

Metrics. In the experiments in this section, we request the model to choose between the safe answer and the harmful answer, and verify its correctness, referring to its label.

Ranker Settings. In all experiments, the rankers are implemented using a single Transformer block. The trainable parameters of the ranker model is less than 5M. They operate on features extracted from approximately the bottom 25% of the base model's layers. During training and evaluation, every data group includes two candidate answers. The ranker is trained to distinguish the answers as safe and harmful. The hyperparameters are set as follows: learning rate is set to 0.001, weight decay is set to 0.0001, dropout is set to 0.1, and momentum is set to 1.0.

Baseline. The baseline of the experiment adopts a reward model to rate answers generated by the base model. A pretrained GPT2 (Radford et al., 2019) is used as the reward model in the experiment. Small as it seems, a GPT2 model is still 20 times larger than the ranker model.

Table 1: Ranking accuracy of SRR in distinguishing safe and harmful prompts.

Source Dataset	Method	Model: Qwen	Mistral	Vicuna	**Average**
Harmbench	Baseline	41.18	35.21	57.60	44.66
	Ours	**82.35**	**91.55**	**90.40**	**88.10**
SorryBench	Baseline	56.72	52.82	55.26	54.93
	Ours	**85.57**	**90.15**	**87.98**	**87.90**
JailbreakBench	Baseline	70.00	67.39	50.00	62.46
	Ours	**80.00**	**95.65**	**95.24**	**90.30**

Table 2: Cross-dataset ranking accuracy of SRR in distinguishing safe and harmful prompts.

Source Dataset	Evaluation Dataset	Model: Qwen	Mistral	Vicuna	**Average**
Harmbench	SorryBench	76.96	88.06	66.04	77.02
	JailbreakBench	80.00	93.48	85.71	86.40
SorryBench	Harmbench	76.47	90.14	80.00	82.20
	JailbreakBench	77.78	89.13	76.19	81.03
JailbreakBench	HarmBench	79.41	89.44	90.40	86.42
	SorryBench	72.41	87.16	78.59	79.39

4.2. Overall evaluation

We use the transformer-architectured ranker to improve the safety of different models on different datasets. As depicted in the 1, our method greatly outperforms the reward model in all base models and all datasets. The accuracy of many experiments reach 90%. Our lightweight method significantly out performs the reward model (gpt2 (Radford et al., 2019)), despite being far smaller in scale. Specifically, when Qwen is used as the base model, the ranker reaches 82.35%, 91.55%, 90.40% respectively on three datasets. Similarly, the results are 85.57%, 90.15%, 87.98% when the base model is Mistral. Finally the performance is 80.00%, 95.65%, 95.24% when the base model is Vicuna. This implies that rankers can adapt to even larger models.

4.3. Cross dataset validation

To further evaluate the generalization capability of our SRR framework across different safety benchmarks, we conduct cross-dataset validation experiments. We apply the ranker trained on one dataset to other unseen datasets. This experimental setup helps us demonstrate whether the model can effectively identify and prioritize safe responses regardless of the dataset's specific characteristics or the types of adversarial prompts it contains.

The results in Table 2 show that our SRR framework achieves consistently strong cross-dataset performance

Table 3: Ranking accuracy of SRR in distinguishing infringement and benign prompts.

Dataset	Model Qwen	Mistral	Vicuna	**Average**
Harmcopy	98.08	95.83	89.74	94.28

Table 4: Ranking accuracy of SRR in distinguishing safe and harmful prompts.

Dataset	Model Qwen	Mistral	Vicuna	**Average**
Biasedbenchmark for QA	54.82	52.09	50.64	52.52

across all three LLMs (Qwen, Mistral, and Vicuna). When trained on one dataset and evaluated on another, SRR maintains a high level of accuracy in distinguishing safe from harmful responses. For instance, a ranker trained on Harmbench achieves 77.02% average accuracy on SorryBench and 86.40% on JailbreakBench. Similarly, a ranker trained on SorryBench achieves 82.20% on Harmbench and 81.03% on JailbreakBench. This cross-dataset effectiveness demonstrates that SRR's safety signal is not overly specialized to any particular dataset but instead captures generalizable features of safety within the LLM's internal representations.

This ability to generalize across different safety benchmarks is crucial for real-world deployment. In practical applications, LLMs may encounter a wide variety of adversarial prompts that differ significantly from those seen during training. The strong cross-dataset performance of SRR suggests that it can serve as a robust safeguard module, effectively filtering out harmful responses even when the specific types of attacks vary. This provides evidence that SRR's approach of leveraging internal model representations for safety ranking is both versatile and adaptable to diverse safety challenges.

4.4. Extension to other alignment perspectives

In this part, we also extend the application of our SRR framework to other critical alignment perspectives beyond general safety, namely privacy and fairness. These dimensions are essential for ensuring that LLMs not only avoid harmful content but also respect user privacy and produce unbiased, equitable responses.

Privacy. To evaluate the potential of SRR in addressing privacy concerns, we conducted experiments on the Harmcopy dataset (Mazeika et al., 2024), which contains prompts related to privacy infringement. The results are presented in Table 3, showing that SRR achieves a high accuracy rate in distinguishing between privacy-infringing and benign prompts across all models. The average accuracy across all models is 94.28%, indicating that SRR is effective in identifying privacy-related safety concerns. This strong performance in the privacy context further validates the generalizability of our approach. The ability to adapt to privacy-specific prompts shows that SRR can capture fine-grained safety signals related to different alignment perspectives beyond just general harmful content. By leveraging the internal representations of LLMs, SRR can effectively identify privacy risks without requiring extensive retraining or modification of the underlying model architecture. This makes it a versatile and efficient solution for enhancing the privacy safeguards in LLM applications.

Fairness. To assess the effectiveness of SRR in ensuring fairness, we conducted experiments on the BBQ dataset (Parrish et al., 2022). This dataset is designed to evaluate the model's ability to avoid generating responses that may contain biases or unfair content. The results are presented in Table 4, which indicates that SRR achieves moderate accuracy in identifying and mitigating biased or unfair responses. The average accuracy across all models is 52.52%, which is relatively lower compared to the results obtained in privacy and safety evaluations. This suggests that while SRR demonstrates some capability in detecting fairness-related issues, there is still room for improvement in this area. Despite this, SRR shows a foundational ability to distinguish between more and less fair responses, indicating that it can serve as a starting point for more specialized fairness enhancements in LLM applications.

In addition to sandbox evaluations above, we further discuss and study the real-world deployment of SRR in Appendix 5.

5. Discussion

This section further discusses the considerations for SRR in practical deployment. We focus on two fundamental problems:

1. To what extent can SRR mitigate safety alignment issues?
2. How does SRR impact the natural performance of LLMs?

5.1. Real-world application

Recall that we mainly apply the classification accuracy as the main metric to evaluate the precision of SRR in ranking the safety of multiple responses. In this part, we further explore how SRR can improve the safety alignment of LLMs, since aligned LLMs have already exhibited certain robustness against harmful prompts. To this end, we incorporate SRR during real-time inference of the protected LLMs, rather than classifying simulated harmful or safe responses. We also consider practical jailbreak attacks to demonstrate

Table 5: Real-world ranking accuracy of SRR in distinguishing safe and harmful prompts in HarmBench

Method	Model		**Average**
	Qwen	Mistral	
First	82.52	54.43	68.48
Ranker	**83.22**	**63.29**	**73.26**

Table 6: Real-world ranking accuracy of SRR in distinguishing safe and harmful prompts in JailbreakingBench

Method	Model		**Average**
	Qwen	Mistral	
First	16.25	32.91	24.58
Ranker	**38.75**	**39.24**	**39.00**

Table 7: Real-world ranking accuracy of SRR in distinguishing safe and harmful prompts in SorryBench

Method	Model		**Average**
	Qwen	Mistral	
First	84.28	46.22	65.25
Ranker	**86.16**	**67.23**	**76.70**

Table 8: Accuracy (%) on the MATH dataset when responses are ranked using SRR trained on different safety datasets.

Source Dataset	Natural	HarmBench	SorryBench	JailbreakBench
Accuracy	68.7	69.1	68.5	68.6

the robustness of SRR. The baseline in this experiment is "first accuracy", which means choosing the answer with the highest possibility generated by the base model. The results shown in Table 5, 6, and 7 demonstrate that SRR significantly enhances the safety alignment of LLMs in real-world applications. When integrated into the inference process of protected LLMs, SRR demonstrates robust performance against practical jailbreak attacks. This indicates that SRR can effectively improve the safety mechanisms of LLMs, reducing their vulnerability to adversarial prompts. By leveraging the model's internal representations, SRR provides an efficient and effective safeguard without compromising the natural performance of the LLMs. Overall, these findings support the practical utility of SRR as a valuable tool for improving the safety and reliability of LLMs in real-world scenarios.

5.2. Natural performance

As discussed in earlier sections, a key advantage of SRR is that it does not intervene in the decoding process of the base language model. This allows SRR to be seamlessly applied at inference time without modifying generation behavior, thereby preserving the model's natural task performance. In this section, we empirically validate this claim using a mathematical reasoning benchmark. We evaluate SRR using the MATH dataset (Hendrycks et al., 2021), which contains 12,500 competition-level math problems spanning seven topics and five difficulty levels. To assess performance, we extract the final answer from each model-generated response and compare it against the ground-truth answer.

We use Qwen2.5-7B-Instruct as the base model. For each instruction, we sample 10 completions and apply the SRR ranker, which is trained solely on safety datasets, to rank them by their predicted safety. The top-ranked response is selected as the final answer. We then compare the answer accuracy of the ranked responses against the accuracy obtained by the base model's default outputs.

The results are shown in Table 8. Across all settings, the accuracy of the SRR-ranked completions remains nearly identical to the base model's natural accuracy (68.7%). In fact, slight fluctuations ($\pm 0.2\%$) are observed depending on which safety dataset the ranker was trained on, but these differences fall within the margin of noise and do not indicate degradation in performance. Notably, this result holds despite the SRR ranker being trained exclusively on safety supervision signals, without any exposure to mathematical reasoning data. This demonstrates that the SRR scoring mechanism does not introduce unintended bias toward specific task domains or alter the correctness of model outputs in benign settings.

6. Conclusion

In this paper, we introduced Safety Representation Ranking (SRR), a novel listwise ranking framework that leverages the internal representations of LLMs to select safe responses without altering the model's decoding logic. Through contrastive training, SRR identifies safety-sensitive features within the LLM's hidden states and uses them to rank candidate responses based on safety. Our method not only improves robustness against adversarial prompts but also generalizes well across different safety evaluation datasets. Furthermore, SRR demonstrates potential for addressing other alignment perspectives such as privacy and fairness. Experimental results indicate that SRR significantly reduces harmful outputs under attack while maintaining performance on benign tasks. Overall, SRR serves as a practical and effective safeguard module for LLM alignment, offering a new paradigm for enhancing the safety and reliability of LLMs in real-world applications.

References

Usman Anwar, Abulhair Saparov, Javier Rando, Daniel Paleka, Miles Turpin, Peter Hase, Ekdeep Singh Lubana, Erik Jenner, Stephen Casper, Oliver Sourbut, et al. Foundational challenges in assuring alignment and safety of large language models. *Transactions on Machine Learning Research*, 2024. 1, 2

Yuntao Bai et al. Constitutional ai: Harmlessness from ai feedback, 2022. 1

Somnath Banerjee, Sayan Layek, Soham Tripathy, Shanu Kumar, Animesh Mukherjee, and Rima Hazra. Safeinfer: Context adaptive decoding time safety alignment for large language models. In *AAAI*, volume 39, pages 27188–27196, 2025. 1

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024. 1

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramer, Hamed Hassani, and Eric Wong. Jailbreakbench: An open robustness benchmark for jailbreaking large language models, 2024. URL https://arxiv.org/abs/2404.01318. 4

Huanran Chen et al. Understanding pre-training and fine-tuning from loss landscape perspectives. *arXiv preprint arXiv:2505.17646*, 2025a. 2

Jianhui Chen, Xiaozhi Wang, Zijun Yao, Yushi Bai, Lei Hou, and Juanzi Li. Finding safety neurons in large language models. *arXiv preprint arXiv:2406.14144*, 2024. 2

Taiye Chen et al. Scalable defense against in-the-wild jailbreaking attacks with safety context retrieval. In *ICML Workshop on Test-Time Adaptation*, 2025b. URL https://openreview.net/forum?id=tEGPXBa9NM. 1

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. Or-bench: An over-refusal benchmark for large language models. *arXiv preprint arXiv:2405.20947*, 2024. 2

Josef Dai, Xuehai Pan, Ruiyang Sun, et al. Safe rlhf: Safe reinforcement learning from human feedback. In *ICLR*, 2024. 1

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018. 2

Jessica Ficler and Yoav Goldberg. Controlling linguistic style aspects in neural language generation. In *Proceedings of the Workshop on Stylistic Variation*, 2017. 2

Rasmus Boll Greve, Emil Juul Jacobsen, and Sebastian Risi. Evolving neural turing machines for reward-based learning. In *Proceedings of the Genetic and Evolutionary Computation Conference 2016*, pages 117–124, Denver Colorado USA, 2016. ACM. ISBN 978-1-4503-4206-3. doi: 10.1145/2908812.2908930. 1

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring mathematical problem solving with the math dataset. In *Advances in Neural Information Processing Systems*, 2021. 6

Ari Holtzman, Jan Buys, Maxwell Forbes, Antoine Bosselut, David Golub, and Yejin Choi. Learning to write with cooperative discriminators. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2018. 2

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. 2

Hakan Inan, Kartikeya Upasani, Jianfeng Chi, Rashi Rungta, Krithika Iyer, Yuning Mao, Michael Tontchev, Qing Hu, Brian Fuller, Davide Testuggine, et al. Llama guard: Llm-based input-output safeguard for human-ai conversations. *arXiv preprint arXiv:2312.06674*, 2023. 1

Jiaming Ji, Tianyi Qiu, Boyuan Chen, Borong Zhang, Hantao Lou, Kaile Wang, Yawen Duan, Zhonghao He, Jiayi Zhou, Zhaowei Zhang, et al. Ai alignment: A comprehensive survey. *arXiv preprint arXiv:2310.19852*, 2023. 1, 2

Albert Q Jiang, A Sablayrolles, A Mensch, C Bamford, D Singh Chaplot, Ddl Casas, F Bressand, G Lengyel, G Lample, L Saulnier, et al. Mistral 7b. arxiv. *arXiv preprint arXiv:2310.06825*, 10, 2023. 4

Tomasz Korbak, Kejian Shi, Angelica Chen, Rasika Bhalerao, Christopher L. Buckley, Jason Phang, Samuel R. Bowman, and Ethan Perez. Pretraining language models with human preferences. In *ICML*, 2023. 1

Xiang Lisa Li, Ari Holtzman, Daniel Fried, Percy Liang, Jason Eisner, Tatsunori Hashimoto, Luke Zettlemoyer, and Mike Lewis. Contrastive decoding: Open-ended text generation as optimization. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2023. 2

Tianqi Liu, Zhen Qin, Junru Wu, Jiaming Shen, Misha Khalman, Rishabh Joshi, Yao Zhao, Mohammad Saleh, Simon Baumgartner, Jialu Liu, et al. Lipo: Listwise preference optimization through learning-to-rank. *arXiv preprint arXiv:2402.01878*, 2024. 3

Yi Liu, Gelei Deng, Zhengzi Xu, Yuekang Li, Yaowen Zheng, Ying Zhang, Lida Zhao, Tianwei Zhang, and Yang Liu. Jailbreaking chatgpt via prompt engineering: An empirical study, 2023. 1

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *ICML*, 2024. 1, 4, 5

Swetasudha Panda, Naveen Jafer Nizar, and Michael L Wick. Llm improvement for jailbreak defense: Analysis through the lens of over-refusal. In *Neurips Safe Generative AI Workshop 2024*, 2024. 1, 2

Alicia Parrish, Angelica Chen, Nikita Nangia, Vishakh Padmakumar, Jason Phang, Jana Thompson, Phu Mon Htut, and Samuel Bowman. Bbq: A hand-built bias benchmark for question answering. In *Findings of the Association for Computational Linguistics: ACL 2022*, pages 2086–2105, 2022. 5

Alberto Purpura, Gianmaria Silvello, and Gian Antonio Susto. Learning to rank from relevance judgments distributions. *Journal of the Association for Information Science and Technology*, 73(9):1236–1252, 2022. 3

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. Safety alignment should be made more than just a few tokens deep. *arXiv preprint arXiv:2406.05946*, 2024. 2

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019. 4

Leo Schwinn, Yan Scholten, Tom Wollschläger, Sophie Xhonneux, Stephen Casper, Stephan Günnemann, and Gauthier Gidel. Adversarial alignment for llms requires simpler, reproducible, and more measurable objectives. *arXiv preprint arXiv:2502.11910*, 2025. 2

Amrith Setlur, Chirag Nagpal, Adam Fisch, Xinyang Geng, Jacob Eisenstein, Rishabh Agarwal, Alekh Agarwal, Jonathan Berant, and Aviral Kumar. Rewarding progress: Scaling automated process verifiers for llm reasoning. *arXiv preprint arXiv:2410.08146*, 2024. 2

Oscar Skean, Md Rifat Arefin, Yann LeCun, and Ravid Shwartz-Ziv. Does representation matter? exploring intermediate layers in large language models. *arXiv preprint arXiv:2412.09563*, 2024. 3

Luong Trung, Xinbo Zhang, Zhanming Jie, Peng Sun, Xiaoran Jin, and Hang Li. Reft: Reasoning with reinforced fine-tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024. 2

Haoxiang Wang, Wei Xiong, Tengyang Xie, Han Zhao, and Tong Zhang. Interpretable preferences via multi-objective reward modeling and mixture-of-experts, 2024a. 2

Xuezhi Wang and Denny Zhou. Chain-of-thought reasoning without prompting. *arXiv preprint arXiv:2402.10200*, 2024. 2

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models, 2023. URL http://arxiv.org/abs/2203.11171. 2

Zhichao Wang, Bin Bi, Shiva Kumar Pentyala, Kiran Ramnath, Sougata Chaudhuri, Shubham Mehrotra, Xiang-Bo Mao, Sitaram Asur, et al. A comprehensive survey of llm alignment techniques: Rlhf, rlaif, ppo, dpo and more. *arXiv preprint arXiv:2407.16216*, 2024b. 1

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. Jailbroken: How does llm safety training fail? In *NeurIPS*, 2023a. 1

Boyi Wei, Kaixuan Huang, Yangsibo Huang, Tinghao Xie, Xiangyu Qi, Mengzhou Xia, Prateek Mittal, Mengdi Wang, and Peter Henderson. Assessing the brittleness of safety alignment via pruning and low-rank modifications. In *ICML*, 2024. 1, 2

Zeming Wei, Yifei Wang, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023b. 1, 3

Zeming Wei et al. Rega: Representation-guided abstraction for model-based safeguarding of llms. *arXiv preprint arXiv:2506.01770*, 2025. 2

Chengcan Wu et al. Mitigating fine-tuning risks in llms via safety-aware probing optimization. *arXiv preprint arXiv:2505.16737*, 2025. 2

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, et al. Sorry-bench: Systematically evaluating large language model safety refusal. In *ICLR*, 2025. 1, 4

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending chatgpt against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 2023. 1

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. Safedecoding: Defending against jailbreak attacks via safety-aware decoding. In *ACL*, pages 5587–5605. Association for Computational Linguistics (ACL), 2024a. 1

Zhangchen Xu, Fengqing Jiang, Luyao Niu, Jinyuan Jia, Bill Yuchen Lin, and Radha Poovendran. SafeDecoding: Defending against jailbreak attacks via safety-aware decoding. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2024b. 2

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024. 4

Dan Zhang, Sining Zhoubian, Ziniu Hu, Yisong Yue, Yuxiao Dong, and Jie Tang. Rest-mcts*: Llm self-training via process reward guided tree search, 2024a. URL http://arxiv.org/abs/2406.03816. 1

Yifan Zhang, Ge Zhang, Yue Wu, Kangping Xu, and Quanquan Gu. General preference modeling with preference representations for aligning language models. *arXiv preprint arXiv:2410.02197*, 2024b. 2

Yihao Zhang, Zeming Wei, Jun Sun, and Meng Sun. Adversarial representation engineering: A general model editing framework for large language models. *arXiv preprint arXiv:2404.13752*, 2024c. 2

Yue Zhang, Leyang Cui, Wei Bi, and Shuming Shi. Alleviating hallucinations of large language models through induced hallucinations. *arXiv preprint arXiv:2312.15710*, 2024d. 2

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. Identifying and tuning safety neurons in large language models. In *ICLR*, 2025. 2

Chujie Zheng, Fan Yin, Hao Zhou, Fandong Meng, Jie Zhou, Kai-Wei Chang, Minlie Huang, and Nanyun Peng. On prompt-driven safeguarding for large language models. In *International Conference on Machine Learning*, pages 61593–61613. PMLR, 2024. 1, 2

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric Xing, et al. Judging llm-as-a-judge with mt-bench and chatbot arena. *Advances in Neural Information Processing Systems*, 36:46595–46623, 2023. 1, 4

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, et al. Representation engineering: A top-down approach to ai transparency. *arXiv preprint arXiv:2310.01405*, 2023a. 1, 2

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models. *arXiv preprint arXiv:2307.15043*, 2023b. 1, 3

A. Detailed Algorithm

Algorithm 1 Safety Representation Ranking (SRR)

Require: Instruction in training data, LLM f, response generator $\mathcal{G}$, ranker g_θ, temperature τ

Training Phase:

for each instruction in training data **do**
  $\{\text{resp}_1, \ldots, \text{resp}_m\} \leftarrow \mathcal{G}(\text{instruction})$ ▷ Generate diverse candidate responses
  $y_i \leftarrow$ safety label for each resp_i ▷ 1 for safe, 0 for unsafe
  $\mathbf{h}_{\text{inst}} \leftarrow f(\text{inst}), \mathbf{h}_{\text{resp},i} \leftarrow f(\text{resp}_i)$ for $i = 1 \ldots m$ ▷ Extract LLM features
  $[\mathbf{o}_{\text{inst}}, \mathbf{o}_{\text{resp},1}, \ldots] \leftarrow g_\theta([\mathbf{h}_{\text{inst}}, \mathbf{h}_{\text{resp},1}, \ldots])$ ▷ Transformer-based contextual encoding
  $s_i \leftarrow \cos(\mathbf{o}_{\text{inst}}, \mathbf{o}_{\text{resp},i})$ ▷ Compute cosine similarity score
  $\hat{p}_i \leftarrow \dfrac{\exp(s_i/\tau)}{\sum_j \exp(s_j/\tau)}$ ▷ Normalize scores via softmax
  $p_i^* \leftarrow \dfrac{1}{k}$ if $y_i = 1$, else 0 ▷ Uniform probability on k safe responses
  $\mathcal{L} \leftarrow \text{KL}(p^* \| \hat{p})$ ▷ Listwise loss
  Update θ to minimize $\mathcal{L}$
end for

Inference Phase:

Given a new instruction and candidate $\{\text{resp}_1, \ldots, \text{resp}_m\}$
Repeat steps 2-6 to compute s_i
return Responses ranked by descending s_i

B. Limitations

Although SRR performs excellently in enhancing the safety of LLMs, there are still a few minor limitations. SRR might need task-specific fine-tuning for optimal performance in certain situations, although the training cost is low. While it generalizes well across multiple safety benchmark datasets, its adaptability to special-domain safety scenarios requires further testing. Also, SRR's effectiveness partly relies on the LLM generating diverse candidate responses; if the responses lack diversity, SRR's performance may be somewhat affected. Despite these minor limitations, SRR remains a robust and practical solution for boosting LLM safety and reliability in various real-world applications.