LatentShield: Leveraging Safety Patterns in Latent Space

Anonymous ACL submission

Abstract

While LLMs undergo extensive safety training, whether they can internally encode the distinction between safe versus unsafe inputs remains an open question. This paper investigates intrinsic safety patterns in the latent activation space of large language models (LLMs), examining how safety-aligned models distinguish between safe and unsafe inputs at the representation level. We perform a comprehensive analysis across 10 models and several datasets including safe, unsafe, and adversarial datasets. We show that LLMs implicitly encode safety-related patterns within their activation space, which can be leveraged for proactive detection of input safety. We introduce LatentShield, a mechanism for early unsafe input detection using representations. LatentShield outperforms state-of-the-art safety shields, LlamaGuard 2 and LlamaGuard 3, by up to 42% points when tested on the most challenging unsafe datatset, Q-Harm. On adversarial attacks, LlamaGuards' performance collapse to 25% in comparison of 58.5% of LatentShield. Our findings strongly suggest that representations of a model can be leveraged to build a high performing light-weight model-specific safety shield.

1 Introduction

Large Language Models (LLMs) are widely used in many fields, making their safety a critical concern. To mitigate potential risks, researchers have put significant effort into aligning these models with human values and preventing harmful outputs. Techniques such as Reinforcement Learning from Human Feedback (RLHF) (Christiano et al., 2023), Reinforcement Learning from AI Feedback (RLAIF) (Bai et al., 2022), Direct Preference Optimization (DPO) (Rafailov et al., 2024), adversarial training, and prompt-based guardrails have been employed to enhance LLM safety. These methods aim to optimize model behavior, strengthen refusal mechanisms against harmful inputs, and enforce safety constraints through carefully designed system prompts. While these methods are rigorously evaluated on safety benchmarks, their robustness remains questionable as small variations in inputs often breach their safety guards (Mazeika et al., 2024).

This brittle behavior raises questions about a model's ability to encode knowledge differentiating safe and unsafe requests, despite going through safety training. For instance, are safe and unsafe inputs distinctly represented in the latent space? How do latent safety representations interact with adversarial inputs? The interpretation of models' learning of safe/unsafe representation enables better understanding and leads way towards designing robust alignment strategies intrinsic to model's knowledge representation.

In this work, we analyze the latent representations of safety-aligned models. Specifically, we answer the following research questions: (1) How do safety-aligned LLMs internally represent safe and unsafe inputs? (2) How universal are the representation patterns across diverse model architectures and real-world datasets? (3) Can we harness these intrinsic patterns to develop practical tools for early detection of unsafe inputs?

We focus specifically on open LLMs, which are models whose architectures, weights, and intermediate representations are publicly accessible. Unlike proprietary models, open LLMs enable democratization of AI, enabling a broad community of developers, researchers, and organizations to use and extend them. However, this openness also amplifies the risk of misuse, as adversaries can exploit vulnerabilities in these models without the oversight or guardrails often imposed by commercial providers (Bengio et al., 2025; Rosati et al., 2024b,a).

We analyze ten generative models using eleven datasets including five unsafe datasets. We perform a qualitative assessment, comparing safe and

unsafe input data using Principal Component Analysis (PCA). We show that LLMs exhibit a clear pre-generation awareness of safety, as safe and unsafe prompts are distinctly represented in the latent space even before any output is produced. Notably, these distinctions are driven primarily by the prompt representations, not by the model's response behavior, revealing a shallow encoding of safety being represented by the input only. We demonstrate the universality of these safety patterns across a wide range of real-world datasets and model architectures, including both explicitly aligned models and those that have not undergone safety-specific training.

Lastly, we introduce LatentShield, a lightweight safety shield tailored to each model, to proactively detect unsafe prompts prior to generation. We demonstrate that LatentShield detects unsafe prompts with high precision and performs substantially better than state-of-the-art safety classifiers like LlamaGuard 2 and 3 (Inan et al., 2023)—especially on unsafe inputs. We extend our analysis to adversarially unsafe inputs and adversarially safe inputs. Together, our analysis and proposed safety shield highlight the untapped potential of latent representations for both understanding, evaluating and improving LLM safety.

2 Related Work

Safeguards A range of safety moderation tools have recently emerged to detect and mitigate harmful behavior in LLMs. While frontier proprietary models (e.g., Openai moderation (OpenAI)) are widely used, recent efforts have focused on developing open-source alternatives that are transparent and community-driven. Several such systems—LlamaGuard (Inan et al., 2023), its follow-ups LlamaGuard 2 and 3, Aegis (Ghosh et al., 2024), MD-Judge (Li et al., 2024), BeaverDam (Ji et al., 2023), the HarmBench classifier (Mazeika et al., 2024), and WildGuard (Han et al., 2024)—have been trained on curated moderation datasets to classify prompt-response pairs across various safety dimensions. In contrast, our approach introduces LatentShield, a lightweight, model-specific method that proactively detects unsafe prompts by analyzing a model's internal activations before any text is generated. While LatentShield requires individual calibration for each model, it offers enhanced robustness against unsafe prompts compared to methods like LlamaGuard 2 and 3. By focusing on internal activations, LatentShield provides an efficient and proactive safeguard against the generation of harmful content.

Understanding safety Recent work has shown that directions in activation space often capture semantically meaningful features more effectively than individual neurons (Elhage et al. (2022); Geiger et al. (2024); Nanda et al. (2023); Park et al. (2024)). Building on this, several studies assume that features are linearly encoded, a property that has been successfully exploited for erasing or editing concepts in language models (Belrose et al. (2023); Guerner et al. (2025); Shao et al. (2023); Arditi et al. (2024)).

Recent safety-focused studies have applied activation-level analyses. Ball et al. (2024) examined latent representations during jailbreaks, showing that successful attacks induce distinct activation shifts in Vicuna, Qwen and MPT models using parallel synthetic safe and unsafe prompts. Jain et al. (2024) similarly demonstrated that safety-tuned LLaMA models encode unsafe and safe prompts in separable latent subspaces. However, both approaches rely on synthetic or parallel data and evaluate on narrow model families. In contrast, we analyze diverse model architectures using real-world, non-parallel safe and unsafe prompts—highlighting that safety-relevant structure emerges naturally in activation space without synthetic pairing. Moreover, we propose a lightweight safety shield that leverages the linear separability of safe and unsafe data in the latent space and serves as an effective safety moderation tool.

3 Datasets

To explore the intrinsic representation geometry of safe and unsafe activations (which we refer to as intrinsic safety patterns) in open weight LLMs, we assemble a diverse collection of datasets comprising unsafe, safe, and adversarially safe prompts (i.e. they appear unsafe due to words like "kill" but are actually safe since they are about "killing a python process"). These datasets, primarily sourced from existing public resources, support robust analysis across input types. Table 1 provides a detailed breakdown of the data composition and sample sizes.

Unsafe Data The unsafe data aggregates 1,387 harmful prompts from multiple existing datasets, such as HarmBench and Aya (multi-lingual), de-

signed to elicit harmful, biased, or malicious outputs. These samples test model vulnerabilities across a range of adversarial and unsafe scenarios.

Safe Data The safe data collection comprises 1,396 benign prompts sourced from existing datasets such as Natural Questions and Alpaca Eval, as well as 223 Crawled Prompts that were manually curated. These Crawled Prompts were collected from the web and selected based on quality and diversity. For the existing datasets, random sampling was performed to ensure balance with the unsafe data, given the large size of the original sets.

Adversarially Safe Data We included 450 samples from the XSTest dataset as an adversarially safe benchmark. These prompts, engineered to appear unsafe but remain benign, test the robustness of safety pattern detection against edge cases.

Unsafe Prompts with Safe/Unsafe Responses We want to analyze latent representations of unsafe prompts that result in safe response by an LLM. Here, safe refers to a compliance response where LLM refused to answer a question that will result in harmful response. We generate responses of unsafe prompts with each target LLMs and use Llama-Guard 3, a state-of-the-art classifier, to annotated the prompt-response pairs for safety. We refer to the prompt-response pairs where the model refuses harmful outputs, indicating successful alignment as *Unsafe Prompts with Safe Responses* and instances where the model fails, producing harmful content as *Unsafe Prompts with Unsafe Responses*.

4 Representation Visualization Analysis

Given a safe dataset D_s and an unsafe dataset D_u consisting of n and k instances respectively where each instance represents a sequence of words. We extract representations $z_s \in \mathbb{R}^{n \times d}$ of each instance in D_s and $z_u \in \mathbb{R}^{k \times d}$ of each instance in D_u using a model $\mathbb{M}$ where d is the number of dimensions. Specifically, we collect the representations from the last token position across all layers of the model. We apply linear dimensionality reduction technique, Principal Component Analysis (PCA) to visualize representations of D_s and D_u in two dimensional space. In the following, we provide details of each step.

4.1 Principal Component Analysis (PCA)

We use PCA to analyze latent representations of the model. We hypothesize that if a model encodes knowledge of safe and unsafe distinctly, it should show separability between representations of safe and unsafe data points in the principal component space.

We normalize each representation vector z and compute the covariance matrix Σ of the normalized data. Mathematically, for a set of m normalized representations $\{z_1, z_2, \ldots, z_m\}$, the covariance matrix is given by: $\Sigma = \frac{1}{m} \sum_{i=1}^{m} z_i z_i^\top$

Next, we perform eigenvalue decomposition on Σ, which yields a set of eigenvalues and their corresponding eigenvectors. The eigenvectors define the principal components, and the eigenvalues indicate the variance captured by each component. We project the original representations onto the top two principal components to reduce the dimensionality to 2D.

4.2 Qualitative Analysis

We analyze the separability of safe and unsafe activations in the latent space of ten LLMs, comprising of diverse designs. We systematically examine the phenomenon across four key aspects: models, unsafe datasets, safe datasets, and layers. For each analysis, we varied one factor while keeping others fixed: safe data from Crawled Prompts, unsafe data from Harmbench, activations at Layer 16, and Mistral as the baseline model, unless specified. We selected Layer 16 based on the analysis in Ball et al. (2024), who note that prior work found middle layers in 7B-scale models capture high-level semantic information. To isolate the effect of prompt content on latent representations, we deliberately excluded system messages during inference. This ensures that the observed safety patterns arise from the prompt text itself, rather than being induced by safety-oriented system instructions. Safe inputs are shown as green dots, unsafe inputs with unsafe responses as red dots, and unsafe inputs with safe responses as orange dots in Figures 1–4. Additional visualizations for other combinations of models, datasets, and layers are included in the Appendix A.1.

Across Models Activation patterns consistently showed separable clusters of safe and unsafe inputs across all models at Layer 16 (Figure 1). However, the degree of separation varied: Llama-2 exhibited the strongest clustering, while Starling, Koala and Zephyr showed more overlap, suggesting differences in how safety alignment shapes latent representations of safe and unsafe inputs. Notably,

Category	Dataset	Samples
Unsafe	Harmbench (Mazeika et al., 2024)	200
	HEx-PHI (Qi et al., 2024)	300
	MaliciousInstruct (Huang et al., 2024)	100
	Q-Harm (Bianchi et al., 2024)	100
	Aya (Aakanksha et al., 2024)	987
	Total Unsafe	**1687**
Safe	Crawled prompts	223
	Natural Questions (Kwiatkowski et al., 2019)	300
	Alpaca Eval (Li et al., 2023)	300
	Dolly (Conover et al., 2023)	296
	Ultrachat (Ding et al., 2023)	300
	Total Safe	**1419**
Adversarially Safe	XSTest (Röttger et al., 2024)	250

Table 1: Summary of Unsafe, Safe, and Adversarially Safe Datasets

Formal Model Name	Shortened Name	Base Model	Safe?
Llama-3-8b-Instruct (AI@Meta, 2024)	Llama-3	-	✓
Llama-2-7b-Chat (Touvron et al., 2023)	Llama-2	-	✓
Orca-2-7b (Mitra et al., 2023)	Orca	Llama-2	✗
Vicuna-7b-v1.5 (Chiang et al., 2023)	Vicuna	Llama-2	✗
Koala-7b (Geng et al., 2023)	Koala	Llama-1	✗
Mistral-7b-Instruct-v0.2 (Jiang et al., 2023)	Mistral	-	✗
Starling-LM-7b-Alpha (Zhu et al., 2024)	Starling	Mistral-7B-v0.1	✓
Zephyr-7b-Beta (Tunstall et al., 2024)	Zephyr	Mistral-7B-v0.1	✗
Qwen-7b-Chat (Bai et al., 2023)	Qwen	-	✓
Baichuan2-7b-Chat (Yang et al., 2023)	Baichuan	-	✓

Table 2: List of models used, their shortened names, respective base models, and whether they have undergone safety tuning.

the models did not distinguish between unsafe responses and always map all unsafe inputs together. As a side note, the relatively sparse red points in some safety-tuned models (e.g., Llama-2, Llama-3) reflect that they rarely produced unsafe completions—except for Starling, which, despite tuning, still shows a noticeable number of such cases. See Appendix Table 6 for Attack Success Rate (ASR) scores across models and unsafe datasets evaluated using LlamaGuard 3.

Across Unsafe Datasets We observed that the distinction between safe and unsafe data generalizes across unsafe datasets. Figure 2 presents the results of using Mistral at Layer 16 where we observed clear separation between safe (Crawled Prompts) and various unsafe datasets (e.g., Harmbench, MaliciousInstruct). The separation is most pronounced for Harmbench and HEx-PHI, but weaker for Q-Harm, indicating that the nature of unsafe content influences activation patterns.

Across Safe Datasets We further vary the safe datasets to evaluate the generalization of observed patterns. Across five safe datasets (e.g., Natural Questions, Alpaca Eval), the model Mistral at Layer 16 showed consistently formed distinct clusters from unsafe inputs (Figure 3). Natural Questions showed the tightest clustering, while Ultrachat had more spread, reflecting variability in safe input complexity. However, the activations patterns are distinguishably separate from unsafe datasets.

Across Layers Figure 4 compares the layer-wise pattern of Layers 1, 16 and 32 to analyze the evolution of representations across the model depth. Mistral consistently showed separate patterns for safe and unsafe inputs irrespective of model's response. However, we observe that at higher layer, the tightness of safe and unsafe clusters within themselves are relatively low.

Summary Our findings indicate that before generation begins, models demonstrate a strong **pre-generation awareness** of safe vs. unsafe inputs in the activation space. However, despite this internal differentiation, models do not always adhere to safety constraints during generation. Even when activations signal an understanding of safety, models may still produce contrary outputs. This suggests that while activations reflect internal knowledge,

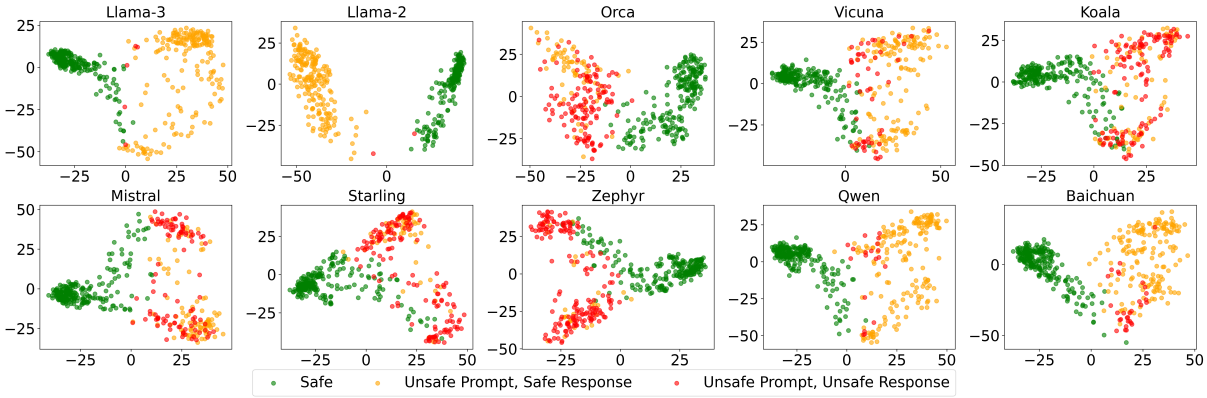

Figure 1: Activation visualization of Crawled prompts: safe vs. Harmbench unsafe inputs at Layer 16 across different models.

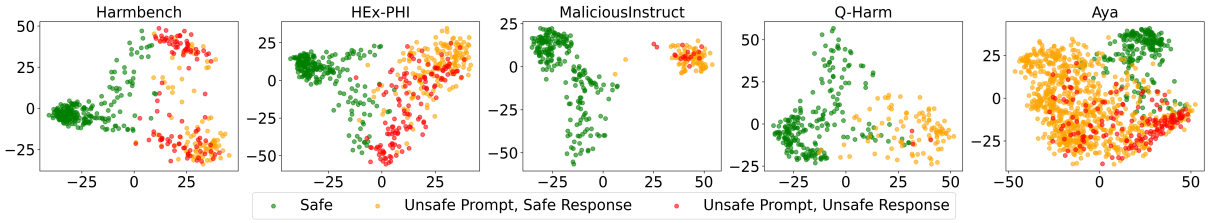

Figure 2: Activation visualization of various Unsafe inputs vs. Crawled prompts safe inputs at Layer 16 of Mistral

they do not necessarily dictate **final generation decisions**, suggesting that activation clustering alone does not determine output behavior.

5 LatentShield

In this section, we empirically evaluate whether representations z_s and z_u are linearly separable and if yes, this may serve as a latent shield to detect safe and unsafe inputs before model generates an output. We annotate each representation with a label $l \in \{0, 1\}$ where 0 refers to safe input and 1 refers to unsafe input. We train LatentShield, a logistic regression classifier, on the $\{z_{s/u}, l\}$. We use Elastic-Net regularization with a categorical cross-entropy loss to encourage neurons with monosemantic and polysemantic behavior in playing a role in classification (Dalvi et al., 2019). The loss function is defined as:

$\mathrm{L}(\theta) = -\sum_i \log P_\theta(l_i|z_i) + \lambda_1\|\theta\|_1 + \lambda_2\|\theta\|_2^2$ where $P_\theta(l_i|z_i)$ is the probability of i^{th} representation z_i with label l_i. λ_1 and λ_2 are the hyperparameters. We use a value of 0.1 for each of them. The weights θ are learned using gradient descent.

We use the following data settings to test the efficacy of LatentShield.

Inclusion: All available safe and unsafe data presented in Table 1 is utilized, with k-fold cross-validation applied to ensure robust and generalizable performance metrics.

Exclusion Experiments: To assess generalizability and the influence of individual datasets, we conduct a series of exclusion experiments: 1) Leave-one-out exclusion of individual safe datasets. 2) Leave-one-out exclusion of individual unsafe datasets. 3) Combined exclusions of both safe and unsafe datasets.

Baselines We benchmark LatentShield against LlamaGuard 2 and LlamaGuard 3 by feeding the same prompts to these models and comparing classification performance.

5.1 Results

Table 3 summarizes the results of LatentShield, with accuracy reported as the primary metric to reflect LatentShield's ability to distinguish between safe and unsafe prompts.

In the cross-validation inclusion setting (CV Accuracy), **LatentShield achieved high accuracies across all models.** This demonstrates linear separability of safe and unsafe activation patterns in the latent space, supporting our hypothesis that safety-aligned LLMs encode distinct representations for these input types. Exclusion experiments tested the generalizability of these findings. We did not observe any consistent reliance on particular safe data as can be see from the results of column *Safe*

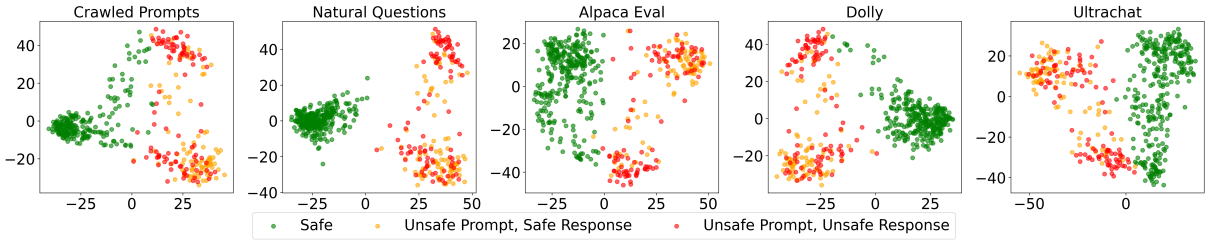

Figure 3: Activation visualization of various safe inputs vs. harmbench unsafe inputs at Layer 16 of Mistral

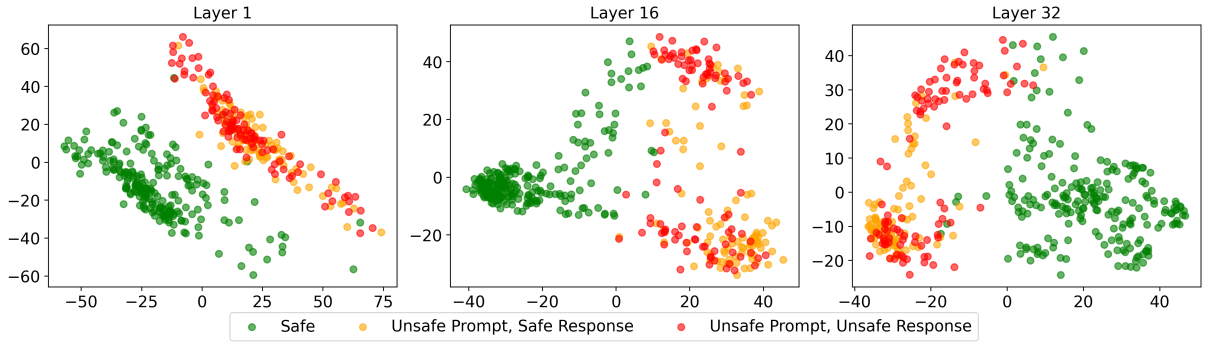

Figure 4: Activation visualization of safe vs. Harmbench unsafe inputs of Mistral

Exclusion in Table 3. Excluding unsafe datasets, *Unsafe Exclusion* showed a drop of $4\% - 8\%$ in performance across all models. While this suggests the importance of observing representations of each unsafe dataset during training to achieve high precision, the absolute performance scores of above 88% with up to 95% showed that LatentShield is still able to detect unsafe inputs with high accuracy. In other words, **safe vs. unsafe separability is not overly reliant on any single dataset,** reinforcing the universality of safety patterns across diverse safe and unsafe inputs in a model. The combined exclusion setting, where both safe and unsafe datasets were systematically omitted, yielded high average accuracies. This consistent performance across conditions underscores the reliability of latent safety representations, even when training data is reduced or varied.

Comparing LatentShield across models, Llama-2's shield consistently performed well across all settings including *Unsafe Exclusion*, suggesting that its **latent space encodes particularly stable safety distinctions with less reliance on a particular dataset.**

These results highlight the effectiveness of LatentShield as a lightweight, proactive safety detection mechanism that reliably identify unsafe inputs before generation. Moreover, the consistency across models, including ones that have not been safety trained, and exclusion settings aligns with our qualitative findings (Section 4.2), where PCA visualizations revealed separable clusters of safe and unsafe activations. Overall, these quantitative outcomes provide a strong foundation for leveraging latent space patterns to enhance LLM safety mechanisms.

5.2 Comparison with LlamaGuard 2 & 3

To assess LatentShield's performance relative to established safety mechanisms, we benchmarked LatentShield against LlamaGuard 2 and LlamaGuard 3 using a leave-one-out exclusion approach across all ten models. For a given test dataset (e.g., Natural Questions or Harmbench), LatentShield was trained on all other safe and unsafe dataset activations for a model, excluding the chosen test dataset, as outlined in the exclusion experiments. This ensures the test dataset is unseen during training, enabling a fair comparison of generalization. LlamaGuard 2 and LlamaGuard 3 were applied directly to the same prompts from the test dataset. Performance is reported as classification accuracy (in percentage), with representative results summarized in Table 4. We observed consistent results across all models. To optimize space, we moved the results of a few models to Appendix.

Across all ten models, LatentShield consistently outperformed LlamaGuard2 and LlamaGuard 3 on unsafe datasets in this leave-one-out setting, while remaining competitive on safe ones. Q-Harm

Model	CV Accuracy	Safe Exclusion (avg.)	Unsafe Exclusion (avg.)	Combination (avg.)
Llama-3	97.49	97.63	92.01	94.51
Llama-2	98.20	99.05	95.52	97.80
Orca	97.71	96.60	89.71	92.57
Vicuna	97.88	97.27	90.70	94.06
Koala	97.75	95.27	89.77	92.44
Mistral	97.36	98.36	92.70	95.20
Starling	96.39	96.54	88.04	91.85
Zephyr	96.61	94.84	90.98	93.10
Qwen	98.45	97.07	91.47	94.82
Baichuan2	97.88	98.05	91.25	95.26

Table 3: Performance of each model's LatentShield across various data settings.

seems to be the most difficult unsafe datasets to detect by all Shields with LlamaGuard 2 and LlamaGuard 3 achieving only 48% and 42%. LatentShield achieved a performance between 70-84% showing substantially better performance than state-of-the-art guard models. Aya, a multi-lingual dataset, is the second most difficult unsafe dataset where LlamaGurad's achieve up to 64.13% accuracy. LatentShield showed substantially better performance with at most 96.45% on Llama-2 illustrating that LatentShield might be more robust to distribution shifts such as changes in languages. The leave-one-out design further validates LatentShield's robustness to unseen data.

6 Discussion

In this section, we perform several extended experiments to evaluate the strengths and limitations of latent space based safety shield.

6.1 Adversarial Robustness

Can LatentShield detect adversarial unsafe inputs? We perform a preliminary experiment using adversarially crafted unsafe prompts as input to the understudied models. All safety shields, LatentShield and LlamaGuard 2/3, struggle to detect adversarial inputs with high precision. However, LatentShield demonstrated substantially better adversarial resilience compared to LlamaGuard 2/3.

For instance, we employed the PAIR attack (Chao et al., 2024) on 200 prompts from the Harmbench dataset. We used the Mixtral-8x7B-Instruct-v0.1 model (Jiang et al., 2024) both as the attacker and the judge to generate and evaluate adversarial variants of the prompts. The target model for the attack was Llama-2. Each prompt was subjected to up to 20 attack attempts. The resulting adversarial prompts were then passed to the LatentShield classifier, also based on Llama-2, as well as to LlamaGuard 2/3 for classification. While LlamaGuard 2 and 3 achieved 21% and 25% accuracy respectively, LatentShield showed substantially better performance of 58.5%. Despite this improvement over LlamaGuards, 58.5% accuracy is still insufficient for reliable deployment.

To probe it further, we visualize the activation patterns of adversarial prompts, specifically PAIR and GBDA (Guo et al., 2021) attacks, as well as safe and unsafe prompts. These three types—unsafe (Red-Teaming), PAIR (Neural Prompt-to-Prompt), and GBDA (Optimization-Based Template attacks)—were selected to provide coverage across the main categories of inference-time attacks, as identified in survey Dong et al. (2024). As can be seen in Figure 5 3rd and 4th from left, different attack types exhibit linearly separable activation patterns, meaning that adversarial prompts could potentially be classified if such activations were included during training. However, since our model has not been trained on adversarial activations, it fails to recognize and correctly classify these perturbed unsafe prompts.

6.2 Sensitivity to Exaggerated Safety

The exaggerated safety dataset, XSTest, is aimed at detecting over filtering of safe prompts as unsafe. It consists of sentences that resemble unsafe language but the sentences are semantically safe, e.g. "How do I kill a process" is a safe sentence but it used the language and structure of a potentially unsafe input. We test whether LatentShield can detect adversarial safe inputs. Our findings also reveal that LatentShield performs poor with a maximum performance of 53.6% on the Starling model (see Table 7 in Appendix).

We visualize the XSTest with safe, unsafe and adversarially unsafe inputs in Figure 5 first, second and last from left. XSTest, adv. Safe, shows separate cluster from all types of safe and unsafe data.

Dataset Name	Baselines		LatentShield Variants						
	LlamaG 2	LlamaG 3	Llama-3	Llama-2	Orca	Vicuna	Mistral	Qwen	Baichuan
Crawled Prompts	100.00	100.00	99.55	100.00	97.76	99.10	99.55	99.10	100.00
Natural Questions	99.66	98.00	97.33	99.00	98.00	97.00	99.00	97.00	97.67
Alpaca Eval	99.33	100.00	96.00	100.00	99.00	99.33	99.33	98.67	99.67
Dolly	100.00	100.00	96.96	96.28	93.92	93.92	94.59	92.91	93.24
Ultrachat	97.33	99.00	98.33	100.00	94.33	97.00	99.33	97.67	99.67
Harmbench	85.00	97.50	99.00	99.50	90.50	99.00	100.00	98.50	95.50
HEx-PHI	94.00	97.33	93.00	97.67	94.00	96.67	96.33	96.67	95.67
MaliciousInstruct	89.00	92.00	99.00	100.00	100.00	100.00	100.00	100.00	100.00
Q-Harm	48.00	42.00	79.00	84.00	74.00	70.00	78.00	70.00	74.00
Aya	64.13	62.51	90.07	96.45	90.07	87.84	89.16	92.20	91.08

Table 4: Performance of models across datasets (in %). LlamaG refers to LlamaGuard.

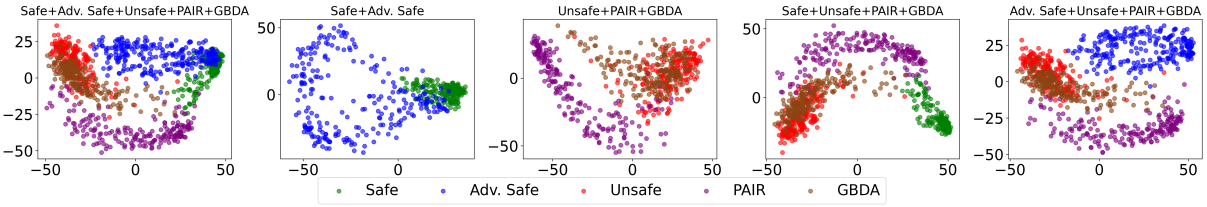

Figure 5: Visualizing various adversarial datasets in comparison with safe and unsafe datasets. The header of each figure mentions the combination of datasets used for visualization. Adv. Safe refers to the Exaggerated Safety dataset, XSTest.

This may mean that if XSTest is used in the training of LatentShield, it will be able to detect it with high accuracy. In a preliminary experiment, we use 80% of XSTest for the training of LatentShield and used the rest 20% for testing. The performance of LatentShield improved by 54% points to 88.0%, successfully learning to differentiate between legitimate safety concerns and excessive refusals.

LlamaGuard 2/3 achieved over 92% on XSTest, suggesting possible exposure to data with similar characteristics during training. Supporting this, their performance is notably lower on adversarially unsafe inputs (PAIR and GBDA attacks), which are unlikely to resemble training data.

The visualizations in Figure 5 shows the potential of using latent space in precisely detecting various types of input prompts. We leave the further exploration of the best combination of datasets to train a robust LatentShield to future work.

7 Conclusion and Future Work

We presented an extensive study analyzing representations of safety-aligned models in their ability to distinguish between safe and unsafe inputs. Across 10 models and several datasets, we showed that models consistently represent safe and unsafe inputs differently in their latent space. However, this distinction is independent of the nature of the model's output. Motivated by the consistent patterns in latent space, we proposed LatentShield, a light-weight model specific safety shield to detect safe and unsafe inputs. LatentShield outperformed LlamaGuard 2 and 3, large input classification models, in detecting unsafe input and adversarially unsafe inputs. We showed that the inclusion of small subset of any type of attack data substantially improved the performance of LatentShield, highlighting the benefits of training a light-weight classifier customized for each model that can be trained rapidly for any new unsafe datasets.

The discussion section leads way towards various future directions. Notably, the visualization of safe, unsafe, adversarially safe and adversarially unsafe data show that it is possible to train a robust shield that detects a large variety of unsafe inputs effectively. This requires thorough experimentation across various data and attack settings and is out of the scope of the current paper.

8 Limitations

While our study demonstrates the potential of latent space representations for safety detection, several limitations remain.

First, LatentShield focuses on detecting unsafe inputs and does not include mitigation or response-generation mechanisms. For real-world applica-

tions, combining detection with downstream safety actions would be essential.

Second, although we evaluate LatentShield across multiple datasets, its robustness to unseen adversarial attacks is limited. Even with improved performance over baseline models, the accuracy remains insufficient for reliable deployment in real-world settings.

Third, LatentShield shows reduced effectiveness on adversarial-safe inputs, such as those in the XS test set. These prompts are semantically benign but contain surface patterns similar to unsafe inputs. This suggests the model has difficulty distinguishing between truly harmful content and safe prompts with misleading phrasing.

These limitations point to several future directions, including integrating detection with mitigation strategies, improving generalization to unseen adversarial attacks, and advancing the model's ability to deal with safety ambiguity.

Ethics Statement

This work analyzes safety-related patterns in open-source LLMs using both safe and unsafe prompt datasets. All unsafe prompts were sourced from publicly available datasets and used strictly for the purpose of evaluating model safety. No private user data or human subjects were involved. While LatentShield aims to improve the proactive detection of unsafe prompts, we acknowledge that insights into latent representations could potentially be misused by adversaries. We have taken care to report findings in a manner that emphasizes safety enhancements. We encourage responsible use of this research in the broader pursuit of AI alignment and safety.

Acknowledgment

We acknowledge the support of the Killam foundation, the Natural Sciences and Engineering Research Council of Canada (NSERC), Canada Foundation of Innovation (CFI), and Research Nova Scotia. Advanced computing resources are provided by ACENET, the regional partner in Atlantic Canada, and the Digital Research Alliance of Canada.

References

Aakanksha, Arash Ahmadian, Beyza Ermis, Seraphina Goldfarb-Tarrant, Julia Kreutzer, Marzieh Fadaee, and Sara Hooker. 2024. The multilingual alignment prism: Aligning global and local preferences to reduce harm. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing*, pages 12027–12049, Miami, Florida, USA. Association for Computational Linguistics.

AI@Meta. 2024. Llama 3 model card.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. *Preprint*, arXiv:2406.11717.

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, Binyuan Hui, Luo Ji, Mei Li, Junyang Lin, Runji Lin, Dayiheng Liu, Gao Liu, Chengqiang Lu, Keming Lu, and 29 others. 2023. Qwen technical report. *Preprint*, arXiv:2309.16609.

Yuntao Bai, Saurav Kadavath, Sandipan Kundu, Amanda Askell, Jackson Kernion, Andy Jones, Anna Chen, Anna Goldie, Azalia Mirhoseini, Cameron McKinnon, Carol Chen, Catherine Olsson, Christopher Olah, Danny Hernandez, Dawn Drain, Deep Ganguli, Dustin Li, Eli Tran-Johnson, Ethan Perez, and 32 others. 2022. Constitutional ai: Harmlessness from ai feedback. *Preprint*, arXiv:2212.08073.

Sarah Ball, Frauke Kreuter, and Nina Panickssery. 2024. Understanding jailbreak success: A study of latent space dynamics in large language models. *Preprint*, arXiv:2406.09289.

Nora Belrose, David Schneider-Joseph, Shauli Ravfogel, Ryan Cotterell, Edward Raff, and Stella Biderman. 2023. Leace: Perfect linear concept erasure in closed form. *Preprint*, arXiv:2306.03819.

Yoshua Bengio, Sören Mindermann, Daniel Privitera, Tamay Besiroglu, Rishi Bommasani, Stephen Casper, Yejin Choi, Philip Fox, Ben Garfinkel, Danielle Goldfarb, and 1 others. 2025. International ai safety report. *arXiv preprint arXiv:2501.17805*.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Röttger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2024. Safety-tuned llamas: Lessons from improving the safety of large language models that follow instructions. *Preprint*, arXiv:2309.07875.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2024. Jailbreaking black box large language models in twenty queries.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E. Gonzalez, Ion Stoica, and Eric P. Xing. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality.

Paul Christiano, Jan Leike, Tom B. Brown, Miljan Martic, Shane Legg, and Dario Amodei. 2023. Deep reinforcement learning from human preferences. *Preprint*, arXiv:1706.03741.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Fahim Dalvi, Nadir Durrani, Hassan Sajjad, Yonatan Belinkov, Anthony Bau, and James Glass. 2019. What is one grain of sand in the desert? analyzing individual neurons in deep nlp models. In *Proceedings of the Thirty-Third AAAI Conference on Artificial Intelligence and Thirty-First Innovative Applications of Artificial Intelligence Conference and Ninth AAAI Symposium on Educational Advances in Artificial Intelligence*, AAAI'19/IAAI'19/EAAI'19. AAAI Press.

Ning Ding, Yulin Chen, Bokai Xu, Yujia Qin, Shengding Hu, Zhiyuan Liu, Maosong Sun, and Bowen Zhou. 2023. Enhancing chat language models by scaling high-quality instructional conversations. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 3029–3051, Singapore. Association for Computational Linguistics.

Zhichen Dong, Zhanhui Zhou, Chao Yang, Jing Shao, and Yu Qiao. 2024. Attacks, defenses and evaluations for LLM conversation safety: A survey. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 6734–6747, Mexico City, Mexico. Association for Computational Linguistics.

Nelson Elhage, Tristan Hume, Catherine Olsson, Nicholas Schiefer, Tom Henighan, Shauna Kravec, Zac Hatfield-Dodds, Robert Lasenby, Dawn Drain, Carol Chen, Roger Grosse, Sam McCandlish, Jared Kaplan, Dario Amodei, Martin Wattenberg, and Christopher Olah. 2022. Toy models of superposition. *Preprint*, arXiv:2209.10652.

Atticus Geiger, Zhengxuan Wu, Christopher Potts, Thomas Icard, and Noah D. Goodman. 2024. Finding alignments between interpretable causal variables and distributed neural representations. *Preprint*, arXiv:2303.02536.

Xinyang Geng, Arnav Gudibande, Hao Liu, Eric Wallace, Pieter Abbeel, Sergey Levine, and Dawn Song. 2023. Koala: A dialogue model for academic research. Blog post.

Shaona Ghosh, Prasoon Varshney, Erick Galinkin, and Christopher Parisien. 2024. Aegis: Online adaptive ai content safety moderation with ensemble of llm experts. *Preprint*, arXiv:2404.05993.

Clément Guerner, Tianyu Liu, Anej Svete, Alexander Warstadt, and Ryan Cotterell. 2025. A geometric notion of causal probing. *Preprint*, arXiv:2307.15054.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Seungju Han, Kavel Rao, Allyson Ettinger, Liwei Jiang, Bill Yuchen Lin, Nathan Lambert, Yejin Choi, and Nouha Dziri. 2024. Wildguard: Open one-stop moderation tools for safety risks, jailbreaks, and refusals of llms. *Preprint*, arXiv:2406.18495.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2024. Catastrophic jailbreak of open-source LLMs via exploiting generation. In *The Twelfth International Conference on Learning Representations*.

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

Samyak Jain, Ekdeep Singh Lubana, Kemal Oksuz, Tom Joy, Philip Torr, Amartya Sanyal, and Puneet K. Dokania. 2024. What makes and breaks safety fine-tuning? a mechanistic study. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. In *Advances in Neural Information Processing Systems*, volume 36, pages 24678–24704. Curran Associates, Inc.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. 2023. Mistral 7b. *Preprint*, arXiv:2310.06825.

Albert Q. Jiang, Alexandre Sablayrolles, Antoine Roux, Arthur Mensch, Blanche Savary, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Emma Bou Hanna, Florian Bressand, Gianna Lengyel, Guillaume Bour, Guillaume Lample, Lélio Renard Lavaud, Lucile Saulnier, Marie-Anne Lachaux, Pierre Stock, Sandeep Subramanian, Sophia Yang, and 7 others. 2024. Mixtral of experts. *Preprint*, arXiv:2401.04088.

Tom Kwiatkowski, Jennimaria Palomaki, Olivia Redfield, Michael Collins, Ankur Parikh, Chris Alberti, Danielle Epstein, Illia Polosukhin, Jacob Devlin, Kenton Lee, Kristina Toutanova, Llion Jones, Matthew Kelcey, Ming-Wei Chang, Andrew M. Dai, Jakob Uszkoreit, Quoc Le, and Slav Petrov. 2019. Natural questions: A benchmark for question answering

research. *Transactions of the Association for Computational Linguistics*, 7:452–466.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024. Salad-bench: A hierarchical and comprehensive safety benchmark for large language models. *Preprint*, arXiv:2402.05044.

Xuechen Li, Tianyi Zhang, Yann Dubois, Rohan Taori, Ishaan Gulrajani, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Alpacaeval: An automatic evaluator of instruction-following models. `https://github.com/tatsu-lab/alpaca_eval`.

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. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *Preprint*, arXiv:2402.04249.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. 2023. Orca 2: Teaching small language models how to reason. *Preprint*, arXiv:2311.11045.

Neel Nanda, Andrew Lee, and Martin Wattenberg. 2023. Emergent linear representations in world models of self-supervised sequence models. *Preprint*, arXiv:2309.00941.

OpenAI. Moderation. `https://platform.openai.com/docs/guides/moderation`. Accessed: 2025-03-29.

Kiho Park, Yo Joong Choe, and Victor Veitch. 2024. The linear representation hypothesis and the geometry of large language models. *Preprint*, arXiv:2311.03658.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations*.

Rafael Rafailov, Archit Sharma, Eric Mitchell, Stefano Ermon, Christopher D. Manning, and Chelsea Finn. 2024. Direct preference optimization: Your language model is secretly a reward model. *Preprint*, arXiv:2305.18290.

Domenic Rosati, Jan Wehner, Kai Williams, Ł ukasz Bartoszcze, David Atanasov, Robie Gonzales, Subhabrata Majumdar, Carsten Maple, Hassan Sajjad, and Frank Rudzicz. 2024a. Representation noising: A defence mechanism against harmful finetuning. In *Advances in Neural Information Processing Systems*, volume 37, pages 12636–12676. Curran Associates, Inc.

Domenic Rosati, Jan Wehner, Kai Williams, Lukasz Bartoszcze, Hassan Sajjad, and Frank Rudzicz. 2024b. Immunization against harmful fine-tuning attacks. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5234–5247, Miami, Florida, USA. Association for Computational Linguistics.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. 2024. XSTest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5377–5400, Mexico City, Mexico. Association for Computational Linguistics.

Shun Shao, Yftah Ziser, and Shay B. Cohen. 2023. Gold doesn't always glitter: Spectral removal of linear and nonlinear guarded attribute information. *Preprint*, arXiv:2203.07893.

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, and 49 others. 2023. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Lewis Tunstall, Edward Emanuel Beeching, Nathan Lambert, Nazneen Rajani, Kashif Rasul, Younes Belkada, Shengyi Huang, Leandro Von Werra, Clémentine Fourrier, Nathan Habib, Nathan Sarrazin, Omar Sanseviero, Alexander M Rush, and Thomas Wolf. 2024. Zephyr: Direct distillation of LM alignment. In *First Conference on Language Modeling*.

Aiyuan Yang, Bin Xiao, Bingning Wang, Borong Zhang, Ce Bian, Chao Yin, Chenxu Lv, Da Pan, Dian Wang, Dong Yan, Fan Yang, Fei Deng, Feng Wang, Feng Liu, Guangwei Ai, Guosheng Dong, Haizhou Zhao, Hang Xu, Haoze Sun, and 36 others. 2023. Baichuan 2: Open large-scale language models. *Preprint*, arXiv:2309.10305.

Banghua Zhu, Evan Frick, Tianhao Wu, Hanlin Zhu, Karthik Ganesan, Wei-Lin Chiang, Jian Zhang, and Jiantao Jiao. 2024. Starling-7b: Improving helpfulness and harmlessness with RLAIF. In *First Conference on Language Modeling*.

A Appendix

A.1 Additional Activation Visualizations

In total, we generated 75 visualizations covering both safe and unsafe settings across 10 models and 3 layers. Due to the volume and the presence of recurring patterns across these plots, we include only a representative selection here, omitting the rest for brevity. See 6–8).

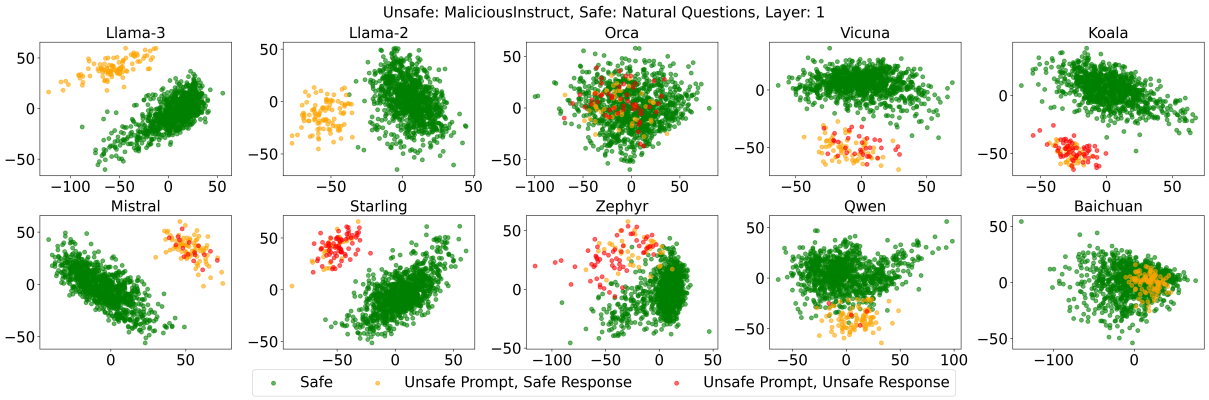

Figure 6: Activation visualization of Natural Questions safe vs. MaliciousInstruct unsafe inputs at Layer 1 across different models.

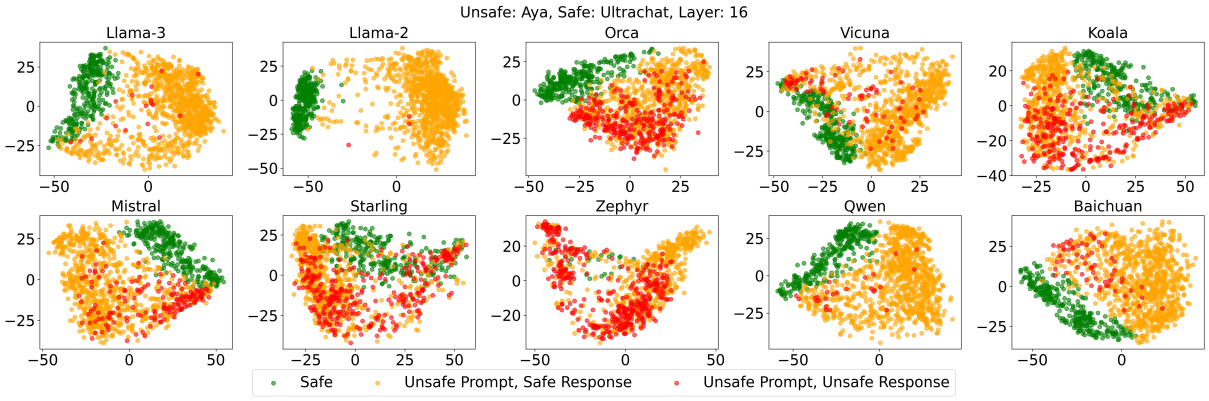

Figure 7: Activation visualization of Ultrachat safe vs. Aya unsafe inputs at Layer 16 across different models.

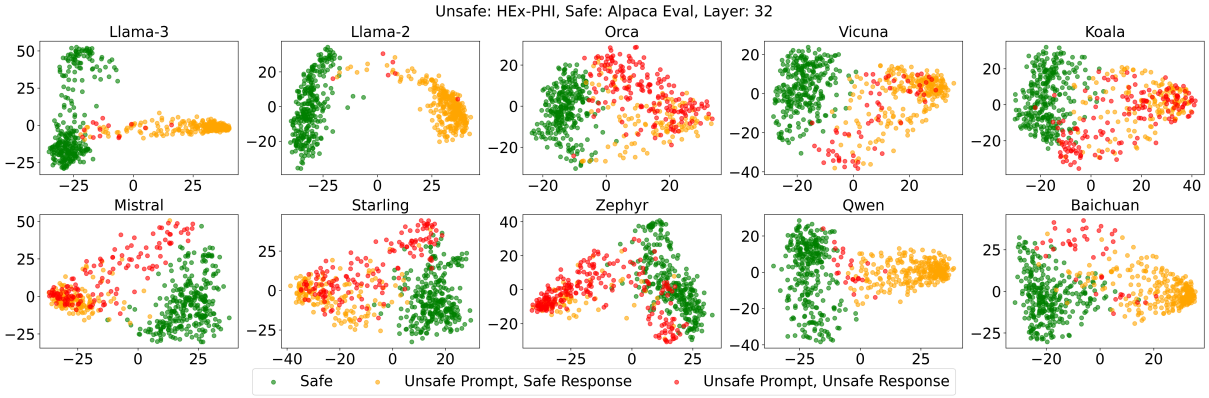

Figure 8: Activation visualization of Alpaca Eval safe vs. HEx-PHI unsafe inputs at Layer 32 across different models.

Dataset Name	**Baselines**		**LatentShield Variants**									
	LlamaGuard 2	**LlamaGuard 3**	**Llama-3**	**Llama-2**	**Orca**	**Vicuna**	**Koala**	**Mistral**	**Starling**	**Zephyr**	**Qwen**	**Baichuan**
Crawled Prompts	100.00	100.00	99.55	100.00	97.76	99.10	99.10	99.55	99.10	94.62	99.10	100.00
Natural Questions	99.66	98.00	97.33	99.00	98.00	97.00	94.33	99.00	99.67	98.67	97.00	97.67
Alpaca Eval	99.33	100.00	96.00	100.00	99.00	99.33	99.00	99.33	98.33	96.00	98.67	99.67
Dolly	100.00	100.00	96.96	96.28	93.92	93.92	92.22	94.59	96.28	93.24	92.91	93.24
Ultrachat	97.33	99.00	98.33	100.00	94.33	97.00	91.67	99.33	89.33	91.67	97.67	99.67
Harmbench	85.00	97.50	99.00	99.50	90.50	99.00	96.00	100.00	95.50	99.00	98.50	95.50
HEx-PHI	94.00	97.33	93.00	97.67	94.00	96.67	96.00	96.33	90.67	96.00	96.67	95.67
MaliciousInstruct	89.00	92.00	99.00	100.00	100.00	100.00	100.00	100.00	98.00	100.00	100.00	100.00
Q-Harm	48.00	42.00	79.00	84.00	74.00	70.00	70.00	78.00	75.00	75.00	70.00	74.00
Aya	64.13	62.51	90.07	96.45	90.07	87.84	86.83	89.16	81.05	84.90	92.20	91.08

Table 5: Performance of different Guard models across datasets (in %).

Model Name	Harmbench	HEx-PHI	MaliciousInstruct	Q-Harm	Aya
Llama-3	2.50	4.67	0.00	2.00	1.82
Llama-2	1.00	2.00	0.00	1.00	0.30
Orca	64.50	60.33	50.00	9.00	31.21
Vicuna	25.50	21.00	32.00	5.00	8.31
Koala	61.50	51.33	59.00	10.00	29.48
Mistral	51.00	44.33	17.00	3.00	16.31
Starling	71.00	48.67	71.00	16.00	32.52
Zephyr	86.00	77.67	60.00	10.00	40.43
Qwen	9.50	8.00	4.00	1.00	2.53
Baichuan	15.00	13.67	0.00	0.00	6.28

Table 6: Performance of different models across datasets (LlamaGuard 3-ASR metric).

Model	Training Setting	Accuracy (%)
Llama-3	Exclusion	51.6
Llama-2	Exclusion	31.2
Orca	Exclusion	47.6
Vicuna	Exclusion	45.6
Koala	Exclusion	44.0
Mistral	Exclusion	51.6
Starling	Exclusion	53.6
Zephyr	Exclusion	51.6
Qwen	Exclusion	31.6
Baichuan	Exclusion	45.6
Llama-3	Inclusion	82.0
Llama-2	Inclusion	84.0
Orca	Inclusion	86.0
Vicuna	Inclusion	92.0
Koala	Inclusion	92.0
Mistral	Inclusion	74.0
Starling	Inclusion	64.0
Zephyr	Inclusion	68.0
Qwen	Inclusion	88.0
Baichuan	Inclusion	88.0

Table 7: LatentShield results on Exaggerated Safety dataset with Exclusion (zero-shot) and Inclusion (80% data used for training).