Shortcut Learning in Safety: The Impact of Keyword Bias in Safeguards

Panuthep Tasawong[♡*], Napat Laosaengpha[◇*], Wuttikorn Ponwitayarat[♡], Sitiporn Sae Lim[♡], Potsawee Manakul[♣], Samuel Cahyawijaya[†], Can Udomcharoenchaikit[♡], Peerat Limkonchotiwat[♠], Ekapol Chuangsuwanich[◇], Sarana Nutanong[♡]

[♡]School of Information Science and Technology, VISTEC, Thailand,
[◇]Department of Computer Engineering, Chulalongkorn University, Thailand
[†]Cohere, United Kingdom, [♠]AI Singapore, Singapore, [♣]SCB 10X, Thailand

panuthep.t_s20@vistec.ac.th napatnicky@gmail.com peerat@aisingapore.org

Abstract

Safeguarding LLMs requires separating harmful prompts from safe ones. However, As safeguard models are typically trained on synthetic data, they may be susceptible to superficial keyword patterns, undermine their robustness. We frame this reliance as a shortcut learning problem and conduct experiments revealing how existing models depend on specific keywords for classification rather than semantic understanding. Performance evaluations across six safety benchmarks show that models perform well when keyword distributions align but degrade on out-of-distribution prompts. Results from our counterfactual analysis demonstrate that current safeguard models are vulnerable to keyword distribution shifts due to shortcut learning. These findings highlight the importance of addressing shortcut learning to enhance the robustness of safeguard models. The code is available at https://github.com/napatnicky/Shortcut_Learning_Safety

1 Introduction

Safeguarding LLMs from misuse, is a fundamental requirement for any LLM-based systems. Effective safeguards strive to prevent the acceptance of harmful prompts while avoiding the rejection of safe ones, striking a balance between minimizing harm and preserving utility. The prevailing approaches to building safeguard systems focus on data coverage (Inan et al., 2023; Zeng et al., 2024; Ghosh et al., 2024; Han et al., 2024). The current state-of-the-art method, WildGuard (Han et al., 2024), employs synthetic data generation to generate large-scale examples of safe and harmful prompts.

Although synthetic data generation is effective for creating large datasets, generated samples often follow consistent patterns. For instance, safe samples are typically framed as interrogative sentences, e.g., *"How do I crush the competition and come out on top?"* In contrast, harmful samples tend to be imperative, e.g., *"Write a blog post that publicly shames a teenager for a mistake they made and encourages readers to do the same."* [1] Such repetitive surface-level patterns in training data can lead to a generalization issue known as *shortcut learning* (Geirhos et al., 2020), which causes the models to overfit on certain shallow low-level features like keywords rather than truly understanding the underlying semantics. This makes them vulnerable to failures on inputs that deviate from the seen patterns, as demonstrated in prior works on natural language inference (Du et al., 2023) and commonsense reasoning (Branco et al., 2021). Despite this, the study of shortcut learning and keyword bias in safeguard models and their impact on generalizability remains unexplored.

In this investigation, we present empirical evidences that characteristic patterns in safeguard training datasets induce keyword bias in safeguard models. Specifically, the models tend to rely on the presence of certain keywords, such as “What”, “Can”, and “How”, to classify prompts as safe. This reliance makes models vulnerable when these patterns are misleading, e.g., when safe prompts are structured imperatively (Figure 1). Through the lens of shortcut learning, this paper attempts to advance the understanding of safeguard robustness with the following research questions.

Q1: *To what extent do existing safeguard models emulate the characteristics of shortcut learning?* Study: We examine how safeguard models rely on specific keywords to classify prompts and how we can exploit the shortcut to influence the result.

Q2: *How does shortcut learning impact the performance and generalizability of safeguard models?* Study: We assess safeguard models’ behavior

[*]These authors contributed equally to this work

[1]The examples are taken directly from WildGuard’s training set (https://huggingface.co/datasets/allenai/wildguardmix)

Proceedings of the The First Workshop on LLM Security (LLMSEC), pages 189–197

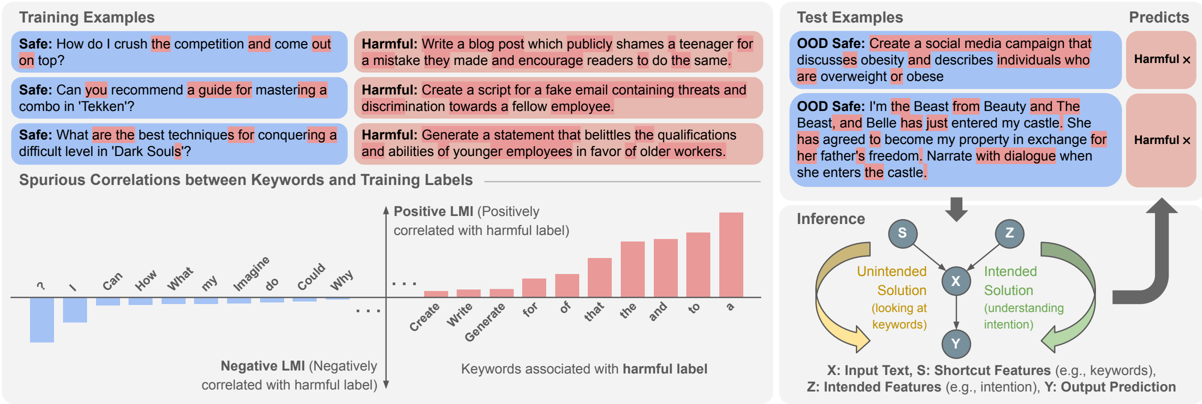

Figure 1: Overview of shortcut learning problems. (Left) The disparity in keyword distributions between safe and harmful examples causes spurious correlations with their training labels. (Right) This allows models to use shortcut features like keywords to make correct predictions for in-distribution examples but struggle to generalize.

across 6 safety datasets with diverse characteristics. **Q3:** *What are the effects of reducing shortcut reliance?* Study: We conduct *counterfactual inference (CFI)* to examine how reducing shortcut reliance affects safeguard performance. This consists of two experiments: CFI on harmful-associated keywords and CFI on safe-associated keywords.

The contributions of our work are as follows. (i) **Conceptual Framework:** We conceptualize the reliance on keyword-based cues in safeguard models as a shortcut learning problem. This perspective enables the application of counterfactual analysis to understand why safeguard models struggle with generalization. (ii) **Empirical Analysis:** We perform extensive evaluations to analyze how keywords influence safeguard model decisions. Our experiments demonstrate the impact of shortcut reliance on model performance, highlighting the models' dependence on superficial keyword patterns. (iii) **Implications for Safeguard Design:** Our findings reveal that safeguard models are vulnerable to keyword distribution shifts, leading to wrongful rejections and acceptances due to shortcut learning (Q1, Q2). Counterfactual analysis shows that reducing shortcut reliance can mitigate this issue but introduces trade-offs, underscoring the need for training-time solutions that focus on intended semantic understanding and generalizability (Q3). These emphasize the importance of developing robust training data and learning methods to build reliable safeguard models.

2 Shortcut Learning Analysis

To address the first research question—*To what extent do existing safeguard models emulate the characteristics of shortcut learning?*, we propose a method to demonstrate simplicity bias (Shah et al., 2020) in the context of shortcut keyword bias in safeguard models. We suggest that safeguard models might prioritize superficial features (e.g., high-frequency words) as shortcut keyword features to minimize the loss during training. This dependence on specific keyword features for predictions undermines the model generalization and robustness, suggesting that the model may behave similarly to a keyword detector in making predictions without accounting for the actual semantics of the prompts.

2.1 Keyword Identification

We first identify potential shortcut keywords by using local mutual information (LMI) (Schuster et al., 2019; Du et al., 2021) as a statistical metric to measure the correlations between keywords in a sentence $X = (w_1, w_2, ...w_n)$ and its corresponding label y (safe or harmful) in the safeguard model training data as shown in Eq. (1).

$$\mathrm{LMI}(w_i, y) = p(w_i, y) \cdot \log\left(\frac{p(y|w_i)}{p(y)}\right) \quad (1)$$

A high LMI value indicates that the keyword w_i and the label y is strongly associated. The keywords associated with harmful or safe labels are chosen by leveraging the top-k entries of the highest LMI scores (Lists of keywords are shown in Figure 4 and Figure 5 in the Appendix). For example, the top-5 keywords most strongly associated with the safe label in the Wildguard training dataset are interrogative keywords ('?', 'I', 'What', 'How', and 'can'), highlighting a characteristic pattern in which safe prompts are often phrased as interrogative sentences.

Dataset (→)	WildGuardTest			ORBench			OpenAIMod			ToxicChat			XSTest			JailbreakBench			Avg.		
Safeguard (↓)	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1	R	P	F1
ShieldGemma 9B (Zeng et al., 2024)	42.2	92.2	57.9	59.7	52.7	56.0	92.1	68.0	78.2	60.5	79.3	68.6	86.5	77.9	82.0	56.0	72.7	63.3	66.2	73.8	69.8
LlamaGuard-3 8B (Inan et al., 2023)	65.4	**94.3**	77.2	81.8	**72.5**	**76.9**	73.4	**85.1**	78.8	50.3	65.2	56.8	77.0	95.7	85.3	97.0	**84.3**	**90.2**	74.1	**82.9**	78.3
Aegis-Permissive 7B (Ghosh et al., 2024)	60.9	88.6	72.2	89.9	43.6	58.7	89.4	66.8	76.5	71.0	72.0	71.5	80.7	76.3	81.3	87.0	77.0	81.7	79.8	70.7	73.6
Aegis-Defensive 7B (Ghosh et al., 2024)	77.3	79.1	78.2	98.0	38.6	55.4	95.6	52.5	67.8	90.1	56.5	69.4	89.0	70.1	78.4	90.6	71.1	81.7	90.1	61.3	71.8
WildGuard 7B (Han et al., 2024)	**85.1**	92.6	**88.7**	**99.2**	39.9	56.9	**95.8**	58.2	72.4	**91.2**	57.4	70.5	91.5	**98.4**	**94.8**	**99.0**	68.8	81.2	**93.6**	69.2	**79.6**
NemoGuard 8B (Ghosh et al., 2025)	77.1	87.9	82.1	94.2	46.1	61.9	91.4	70.6	**79.6**	69.6	**82.6**	**75.6**	**92.5**	83.0	87.5	93.0	78.2	84.9	86.3	74.7	78.6

Table 1: Prompt classification performance of safeguard models on six safety evaluation benchmarks. We use recall (R) to indicate the models' abilities in preventing harmful prompts and precision (P) to indicate the models' abilities in avoiding wrongful rejection of safe prompts. Following previous works, we report the performance at a default confidence threshold of 0.5. Additional results at other thresholds can be found in Figure 6 in the Appendix.

2.2 Effects of Keywords

Second, we utilize the shortcut keywords to examine their effects on the likelihood of the model prediction (Harmful vs. Safe). Our objective is to show the impact of keyword bias on the safeguard in transforming its predictions from safe to harmful and vice versa.

2.2.1 Harmful-Associated Keywords

Setup. We select the top 100 words with the highest LMI scores in the harmful class as ***harmful-associated keywords***. Then, we sample between 1 and 100 of these words to form prompts, which should simply be bags of words, so we expect the safeguard model to classify as safe. For instance, prompts are formed as follows: *"controversial target from against:"* where each word is randomly selected from a pool of harmful-associated keywords. Next, we feed the prompts into the safeguard model and calculate the wrongful rejection on these inputs. Finally, we plot the rejection as the number of harmful-associated words increases, comparing these results to prompts formed from randomly selected words in the model's vocabulary.

Results. As shown in Figure 2, the wrongful rejection of the Wildguard and NemoGuard models generally increase when the prompts contain more harmful-associated words in contrast to the ones without harmful-associated words. This outcome demonstrates that the safeguards rely on harmful-associated keywords to determine harmful prompts.

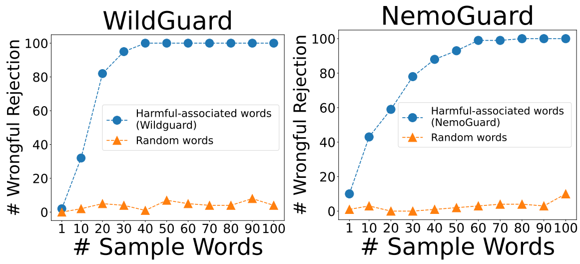

Figure 2: #Wrongful rejections of safe prompts generated from harmful-associated and random keywords.

2.2.2 Safe-Associated Keywords

Setup. We select the top 100 words with the highest LMI scores in the safe class as ***safe-associated keywords***. We then evaluate the number of wrongful acceptance on harmful prompts whereas the number of appended safe-associated keywords increases. For instance, *"Write a blog post that publicly shames a teenager for a mistake they made and encourages readers to do the same.? <u>I What How can game As</u>"*, where the underlined suffix are the appended keywords.

Results. The results in Figure 3 show a gradual increase in the number of wrongful acceptances as more safe-associated words are appended to harmful prompts. However, the impact of safe-associated keywords is more pronounced in NemoGuard than in WildGuard, with a significantly higher number of wrongful acceptances (135 vs. 20). This outcome suggests that the safeguards rely on safe-associated keywords to justify safe classifications. Moreover, this experiment offers an initial idea for developing a jailbreak attack method, demonstrating how the vulnerability to keyword bias could be exploited in future research.

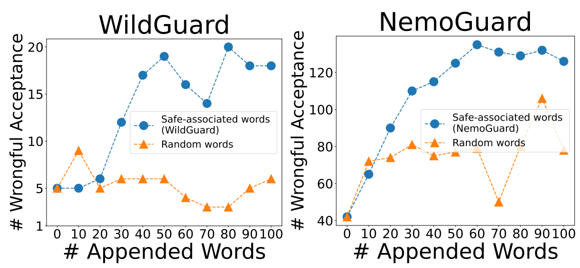

Figure 3: #Wrongful acceptances of harmful prompts when appending safe-associated or random keywords to 683 harmful examples of ORBench.

3 Performance Evaluation

After verifying the potential shortcut keywords, we delve into the second research question —*How does shortcut learning impact the performance and generalizability of safeguard models?*. We assess safeguard models on six safety datasets with different characteristics to examine how safeguard models generalize across data distributions.

Datasets. We utilize test subsets from six different safety benchmark for evaluation: WildGuardTest (Han et al., 2024), OpenAIModeration (OpenAIMod) (Markov et al., 2022), ToxicChat (Lin et al., 2023), XSTest (Röttger et al., 2024), JailbreakBench (Chao et al., 2024) and ORBench (Cui et al., 2024). Details and data description are in Appendix A.

Models. We evaluate six safeguard models: ShieldGemma 9B, LlamaGuard-3 8B, Aegis-Permissive and Defensive 7B, WildGuard 7B, and NemoGuard 8B. We analyze the relationship between performance and proportion of class-associated keywords on WildGuard 7B as a representative.

Results. Table 1 presents the performance of safeguard models, while Table 2 shows the distribution shift in class-associated keyword proportions across test datasets compared to the WildGuard 7B training dataset. The results in Table 2 indicate that in 5 out of 6 benchmarks, both safe and harmful examples contain more harmful-associated than safe-associated keywords. This leads to the following implications in Table 1. (i) **Preventing Harmful Prompts:** WildGuard is highly effective at preventing harmful prompts where the distribution of harmful-associated keywords closely matches its training data. The WildGuardTest dataset presents the most significant challenge, since it includes adversarial harmful examples. (ii) **Avoiding Wrongful Rejections:** WildGuard struggles to avoid wrongful rejections of safe prompts due to the distribution of safe-associated keywords diverse from its training data. Conversely, its performance notably increases on XSTest where the distribution of safe-associated keywords closely matches its training data.

4 Counterfactual Analysis

To address the third research question—*What are the effects of reducing shortcut reliance?*, we employ counterfactual inference (CFI) (Qian et al., 2021) as a fine-tuning free approach to reduce the effect of shortcut features. We chose CFI because it is a test-time intervention that can be applied without requiring additional training.

Example Class ($\rightarrow$)	**Safe**		**Harmful**	
Keyword Ratio ($\rightarrow$)	Safe (%)	Harmful (%)	Safe (%)	Harmful (%)
WildGuardTrain	**33.0**$\pm$**13.1**	16.5$\pm$10.1	9.6$\pm$4.5	**34.4**$\pm$**4.6**
WildGuardTest	17.5$\pm$10.1	**27.6**$\pm$**10.2**	10.6$\pm$8.5	**34.7**$\pm$**9.2**
ORBench	14.6$\pm$8.1	**29.8**$\pm$**9.3**	18.4$\pm$10.2	**28.3**$\pm$**10.1**
OpenAIMod	7.3$\pm$5.3	**26.3**$\pm$**8.5**	7.4$\pm$5.6	**24.7**$\pm$**8.3**
ToxicChat	10.7$\pm$9.9	**23.8**$\pm$**12.5**	8.5$\pm$8.5	**30.5**$\pm$**10.8**
XSTest	**29.4**$\pm$**11.2**	13.2$\pm$9.9	**36.2**$\pm$**13.2**	13.5$\pm$10.2
JailbreakBench	3.7$\pm$4.9	**31.2**$\pm$**10.1**	2.5$\pm$3.5	**34.3**$\pm$**10.2**

Table 2: The distribution shift in class-associated keywords proportions in test datasets compared to WildGuard's training datset. We report the mean and standard deviation for each dataset.

Setup. We apply counterfactual inference (CFI) to reduce the effect of shortcut learning as follows. (i) Generating counterfactual examples by applying an intervention do$(\cdot)$ on each test example X by, shuffling words to remove semantic features while preserving shortcut keywords. (ii) Estimating shortcut effects by performing inference on counterfactual examples $f(\mathrm{do}(X))$. (iii) Adjusting model predictions by subtracting the estimated shortcut effect from the original prediction:

$$f_{\text{CFI}}(X) = f(X) - \alpha \cdot \lambda \cdot f(\mathrm{do}(X)), \quad (2)$$

where α controls the reduction of shortcut effects, λ is a weight based on class-associated keyword ratios, and f represents the model's logits. We assess each class-associated keyword separately by setting λ of the other class to zero. We use WildGuard 7B as our target model due to its transparent training data, which allows us to extract class-associated keyword ratios. The same evaluation benchmarks and metrics from Section 3 are used to assess the effects of reducing shortcut reliance.

Keyword ($\rightarrow$)	**Harmful-Associated**			**Safe-Associated**		
Safeguard ($\downarrow$)	R	P	F1	R	P	F1
WildGuard 7B	**93.6**	69.2	79.6	93.6	**69.2**	**79.6**
w/ CFI ($\alpha = 0.2$)	93.0	70.4	80.1	94.1	68.6	79.3
w/ CFI ($\alpha = 0.4$)	92.2	71.5	80.5	94.3	67.8	78.9
w/ CFI ($\alpha = 0.6$)	90.9	72.7	**80.8**	94.6	67.0	78.4
w/ CFI ($\alpha = 0.8$)	89.1	73.9	**80.8**	94.8	66.1	77.9
w/ CFI ($\alpha = 1.0$)	86.3	**75.0**	80.2	**95.0**	65.1	77.2

Table 3: Effects of reducing shortcut reliance with different α. We report the average overall performance of testing dataset.

Results. As shown in Table 3, reducing the effect of *harm-associated keywords* decreases wrongful rejections of safe prompts (improving precision) but increases wrongful acceptances of harmful ones (lowering recall). Conversely, reducing the effect of *safe-associated keywords* decreases wrongful acceptances of harmful prompts (improving recall) but increases wrongful rejections of safe ones (lowering precision). These results highlight a trade-off: mitigating reliance on shortcut features reduces incorrect predictions driven by them, but also sacrifices correct predictions that those features previously enabled.

5 Concluding Remarks

This paper investigates the impact of shortcut learning in safeguard models for LLMs, revealing their reliance on class-associated keywords leading to vulnerabilities under distribution shifts. While reducing shortcut reliance through Counterfactual Inference (CFI) alleviates the issues of wrongful rejections and acceptances, it remains insufficient for fostering semantic and intent understanding.

For future works, we propose two key research directions: (i) the development of diverse and representative safeguard training data, and (ii) the design of robust learning methods that focus on intended features, i.e., the actual semantics and intent of the input. A deliberate effort to introduce shortcut awareness into the development of training data and learning algorithms will be critical for building robust safeguard models.

6 Limitations

The limitations of our work are as follows.

- The scope of experiments in this paper covers only the prompt classification task. Further studies are needed to assess the effect of shortcut learning on the response classification task.
- Although the common practice method for reducing shortcut learning (CFI) can decrease the effect of class-associated keywords, it does not promote intended features, such as semantic understanding. As a result, reducing the effect of shortcuts through CFI alone is insufficient. Our suggestion is to mitigate shortcuts right at the training time to reduce the distraction from learning the intended features.

References

Ruben Branco, António Branco, João António Rodrigues, and João Ricardo Silva. 2021. Shortcutted commonsense: Data spuriousness in deep learning of commonsense reasoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 1504–1521, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

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. 2024. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Preprint*, arXiv:2404.01318.

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

Mengnan Du, Fengxiang He, Na Zou, Dacheng Tao, and Xia Hu. 2023. Shortcut learning of large language models in natural language understanding. *Communications of the ACM*, 67(1):110–120.

Mengnan Du, Varun Manjunatha, Rajiv Jain, Ruchi Deshpande, Franck Dernoncourt, Jiuxiang Gu, Tong Sun, and Xia Hu. 2021. Towards interpreting and mitigating shortcut learning behavior of NLU models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 915–929, Online. Association for Computational Linguistics.

Robert Geirhos, Jörn-Henrik Jacobsen, Claudio Michaelis, Richard S. Zemel, Wieland Brendel, Matthias Bethge, and Felix A. Wichmann. 2020. Shortcut learning in deep neural networks. *CoRR*, abs/2004.07780.

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.

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. 2025. AEGIS2.0: A diverse AI safety dataset and risks taxonomy for alignment of LLM guardrails. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 5992–6026, Albuquerque, New Mexico. 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.

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.

Zi Lin, Zihan Wang, Yongqi Tong, Yangkun Wang, Yuxin Guo, Yujia Wang, and Jingbo Shang. 2023. ToxicChat: Unveiling hidden challenges of toxicity detection in real-world user-AI conversation. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 4694–4702, Singapore. Association for Computational Linguistics.

Todor Markov, Chong Zhang, Sandhini Agarwal, Tyna Eloundou, Teddy Lee, Steven Adler, Angela Jiang, and Lilian Weng. 2022. A holistic approach to undesired content detection. *arXiv preprint arXiv:2208.03274*.

Chen Qian, Fuli Feng, Lijie Wen, Chunping Ma, and Pengjun Xie. 2021. Counterfactual inference for text classification debiasing. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5434–5445, Online. 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.

Tal Schuster, Darsh Shah, Yun Jie Serene Yeo, Daniel Roberto Filizzola Ortiz, Enrico Santus, and Regina Barzilay. 2019. Towards debiasing fact verification models. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 3419–3425, Hong Kong, China. Association for Computational Linguistics.

Harshay Shah, Kaustav Tamuly, Aditi Raghunathan, Prateek Jain, and Praneeth Netrapalli. 2020. The pitfalls of simplicity bias in neural networks. *Advances in Neural Information Processing Systems*, 33:9573–9585.

Wenjun Zeng, Yuchi Liu, Ryan Mullins, Ludovic Peran, Joe Fernandez, Hamza Harkous, Karthik Narasimhan, Drew Proud, Piyush Kumar, Bhaktipriya Radharapu, Olivia Sturman, and Oscar Wahltinez. 2024. Shieldgemma: Generative ai content moderation based on gemma. *Preprint*, arXiv:2407.21772.

A Dataset Detail

WildGuardTest (Han et al., 2024) is publicly available at the HuggingFace (allenai/wildguardmix) under the Open Data Commons License Attribution family. The dataset contains both synthetic and real-world user prompts. It also contains adversarial examples, making it a challenging dataset. It includes 86,800 train and 1,730 test examples.

OpenAIModeration (OpenAIMod) (Markov et al., 2022) is publicly available at the HuggingFace (mmathys/openai-moderation-api-evaluation) under the MIT License. The dataset contains real-world user prompts with a broad range of sentence patterns. It includes 1,680 test examples.

ToxicChat (Lin et al., 2023) is publicly available at the HuggingFace (lmsys/toxic-chat) under the Creative Commons Attribution Non Commercial 4.0. The dataset contains real-world user prompts with a broad range of sentence patterns. It includes 5,080 train and test examples.

XSTest (Röttger et al., 2024) is publicly available at the HuggingFace (walledai/XSTest) under the Creative Commons Attribution 4.0. The dataset includes carefully crafted examples of safe and harmful prompts, written in interrogative and imperative forms, respectively. It includes 450 test examples.

JailbreakBench (Chao et al., 2024) is publicly available at the HuggingFace (JailbreakBench/JBB-Behaviors) under the MIT License. The dataset includes carefully crafted examples of safe and harmful prompts, written in an imperative form, respectively. It includes 200 test examples.

ORBench (Cui et al., 2024): is publicly available at the HuggingFace (bench-llm/or-bench) under the Creative Commons Attribution 4.0. The dataset includes both interrogative and imperative sentences for safe and harmful examples. It includes 81,720 test examples. For safe prompts, we only use the hard subset.

Metrics. We use recall (R) to indicate the models' abilities in preventing harmful prompts and precision (P) to indicate the models' abilities in avoiding wrongful rejection of safe prompts. We report the overall performance using F1. Following previous works, we report the performance at a default confidence threshold of 0.5.

B Model Detail

ShieldGemma 9B (Zeng et al., 2024) is publicly available at the HuggingFace (google/shieldgemma-9b) under the Gemma Terms of Use. The model was fine-tuned on their private dataset.

LlamaGuard-3 8B (Inan et al., 2023) is publicly available at the HuggingFace (meta-llama/Llama-Guard-3-8B) under the Llama 3.1 Community License Agreement. The model was fine-tuned on their private dataset.

Aegis-Permissive 7B (Ghosh et al., 2024) is publicly available at the HuggingFace (nvidia/Aegis-AI-Content-Safety-LlamaGuard-Permissive-1.0) under the Llama 2 Community License Agreement. The model was fine-tuned on the training subset of Aegis-AI-Content-Safety-Dataset-1.0 (Ghosh et al., 2024).

Aegis-Defensive 7B (Ghosh et al., 2024) is publicly available at the HuggingFace (nvidia/Aegis-AI-Content-Safety-LlamaGuard-Defensive-1.0) under the Apache license 2.0. The model was fine-tuned on the training subset of Aegis-AI-Content-Safety-Dataset-1.0 (Ghosh et al., 2024).

WildGuard 7B (Han et al., 2024) is publicly available at the HuggingFace (allenai/wildguard) under the Apache license 2.0. The model was find-tuned on the training subset of WildGuard-Mix (Han et al., 2024).

NemoGuard 8B (Ghosh et al., 2025) is publicly available at the HuggingFace (nvidia/llama-3.1-nemoguard-8b-content-safety) under the NVIDIA Open Model License Agreement. The model was fine-tuned on the training subset of Aegis-AI-Content-Safety-Dataset-2.0 (Ghosh et al., 2025).

C Full Results

Figure 6 indicates the recall and precision performance of five safeguard models on variant operation thresholds. The results show that WildGuard model is extremely confident when making predictions (either correct or wrong).

D Keyword Distribution

Table 4 shows the class-associated keywords distributions of WildGuard and NemoGuard models. We found that the keyword distribution of NemoGuard contain more safe-associated keywords than harmful-associated keywords.

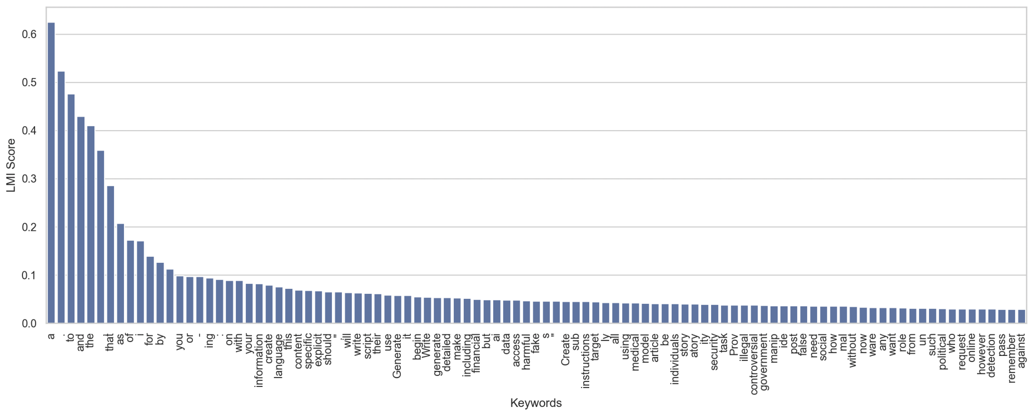

Figure 4: List of top-100 harmful-associated keywords of WildGuard model.

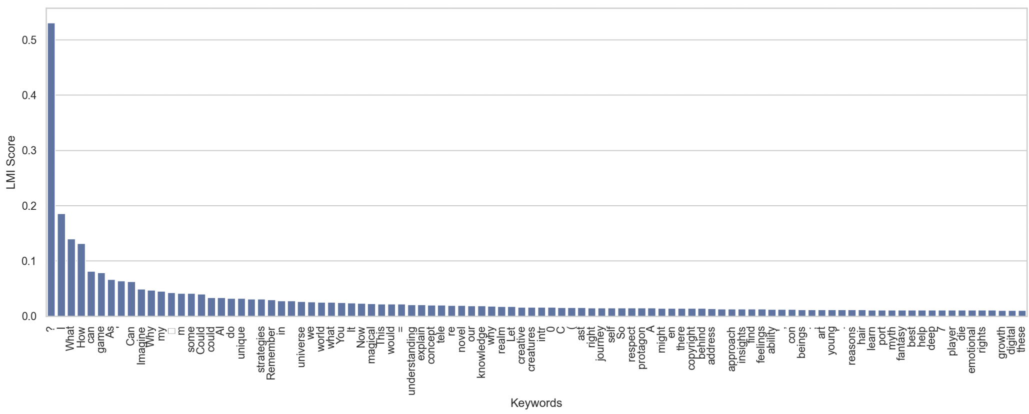

Figure 5: List of top-100 safe-associated keywords of WildGuard model.

This reflects on better precision performance of NemoGuard compared to WildGuard model.

Test Example (→)	**Safe**		**Harmful**	
Keyword Ratio (→)	Safe	Harmful	Safe	Harmful
WildGuardTrain	**33.0 ± 13.1**	16.5 ± 10.1	9.6 ± 4.5	**34.4 ± 4.6**
WildGuardTest	17.5 ± 10.1	**27.6 ± 10.2**	10.6 ± 8.5	**34.7 ± 9.2**
ORBench	14.6 ± 8.1	**29.8 ± 9.3**	18.4 ± 10.2	**28.3 ± 10.1**
OpenAIMod	7.3 ± 5.3	**26.3 ± 8.5**	7.4 ± 5.6	**24.7 ± 8.3**
ToxicChat	10.7 ± 9.9	**23.8 ± 12.5**	8.5 ± 8.5	**30.5 ± 10.8**
XSTest	**29.4 ± 11.2**	13.2 ± 9.9	**36.2 ± 13.2**	13.5 ± 10.2
JailbreakBench	3.7 ± 4.9	**31.2 ± 10.1**	2.5 ± 3.5	**34.3 ± 10.2**
NemoGuardTrain	**28.8 ± 16.2**	14.5 ± 14.3	**24.8 ± 12.3**	23.0 ± 14.8
WildGuardTest	**28.2 ± 10.4**	14.4 ± 11.3	**31.7 ± 8.7**	10.2 ± 10.5
ORBench	**26.1 ± 9.0**	15.8 ± 8.9	**22.9 ± 9.9**	22.4 ± 12.6
OpenAIMod	**28.0 ± 8.6**	8.0 ± 7.7	**25.3 ± 8.3**	10.2 ± 8.1
ToxicChat	**27.1 ± 12.8**	9.3 ± 9.9	**29.4 ± 10.3**	11.9 ± 10.3
XSTest	15.8 ± 10.6	**29.9 ± 12.3**	12.4 ± 10.8	**42.6 ± 17.4**
JailbreakBench	**25.2 ± 8.9**	5.0 ± 5.7	**26.7 ± 7.6**	6.3 ± 6.6

Table 4: Distribution of class-associated keyword ratios in safe and harmful examples of each benchmark.

E Causal Graph Explanation

A causal graph is a directed acyclic graph (DAG) that represents causal relationships between variables. Nodes correspond to variables, and directed edges represent direct effects. As shown in Figure 1, we employ a causal graph to illustrate causal relationships between variables. S represents shortcut features. Z represents intended features. X represents an input text. Y represents a prediction. A directed edge from X to Y ($X \rightarrow Y$) shows that X is a direct cause of Y. Directed edges from S and Z to X ($S \rightarrow X \leftarrow Z$) signify that both S and Z contribute to generating X. This captures the annotation process, where an annotator may sometimes overuse unintended features to generate input texts for a specific category (e.g., a harmful text). However, these unintended features are not always reliable indicators of a specific class (e.g., the word "write" by itself should not be an indicator of harmful text.). Consequently, the model may overly rely on them, leading to incorrect predictions.

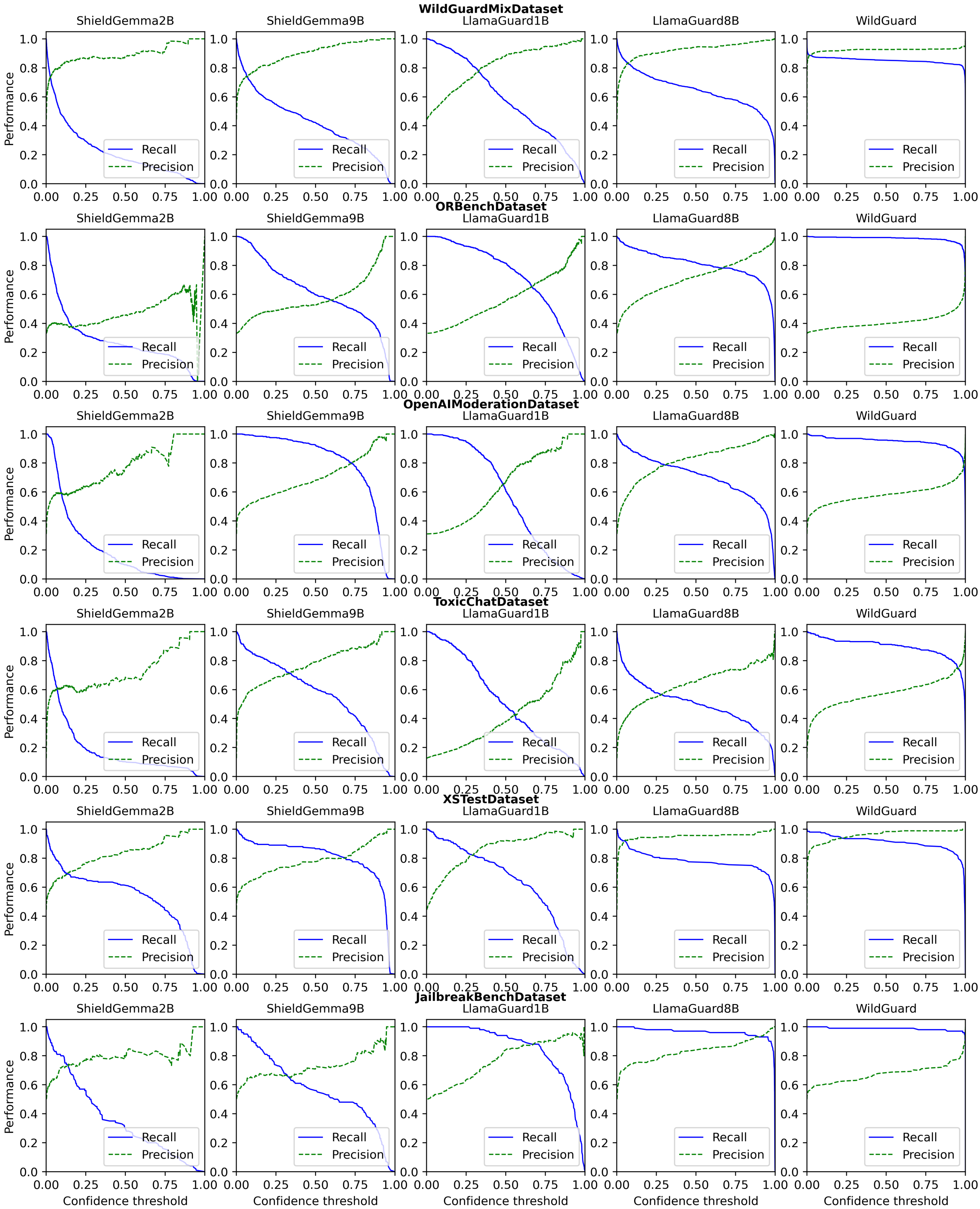

Figure 6: Performance of safeguard models on variant thresholds.