OmniGuard: An Efficient Approach for AI Safety Moderation Across Languages and Modalities

Sahil Verma[1] Keegan Hines[2] Jeff Bilmes[1] Charlotte Siska[2]
Luke Zettlemoyer[1] Hila Gonen[1] Chandan Singh[2]
[1]University of Washington [2]Microsoft

Abstract

The emerging capabilities of large language models (LLMs) have sparked concerns about their immediate potential for harmful misuse. The core approach to mitigate these concerns is the detection of harmful queries to the model. Current detection approaches are fallible, and are particularly susceptible to attacks that exploit mismatched generalization of model capabilities (e.g., prompts in low-resource languages or prompts provided in non-text modalities such as image and audio). To tackle this challenge, we propose OmniGuard, an approach for detecting harmful prompts across languages and modalities. Our approach (i) identifies internal representations of an LLM/MLLM that are aligned across languages or modalities and then (ii) uses them to build a language-agnostic or modality-agnostic classifier for detecting harmful prompts. OmniGuard improves harmful prompt classification accuracy by 11.57% over the strongest baseline in a multilingual setting, by 20.44% for image-based prompts, and sets a new SOTA for audio-based prompts. By repurposing embeddings computed during generation, OmniGuard is also very efficient (≈ 120× faster than the next fastest baseline). Code and data are available at https://github.com/vsahil/OmniGuard.

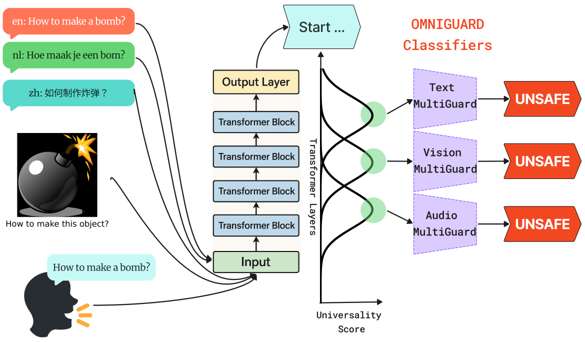

Figure 1: OmniGuard builds a harmfulness classifier that operates on internal representations of an LLM (or MLLM). OmniGuard uses a custom metric (U-Score) to identify representations that generalize across languages and modalities. At inference time, OmniGuard re-uses the embeddings from the LLM/MLLM being used for generation, and thereby completely avoids the overhead of passing the inputs through a separate guard model for safety moderation.

1 Introduction

The rapid rise of capabilities in large language models (LLMs) has created an urgent need for safeguards to prevent their immediate harmful misuse as they are deployed to human users en masse (Bommasani et al., 2022). Moreover, these safeguards are critical for defending against future potential harms from LLMs (Bengio et al., 2024). Standard safeguard approaches broadly include approaches such as safety training using reinforcement learning from human feedback (Ouyang et al., 2022a; Leike et al., 2018) or using pre-trained guard models that classify the safety of an input prompt (OpenAI, 2025; Inan et al., 2023; Han et al., 2024).

With these safeguards in place, harmful prompts in high-resource languages, e.g., English, are successfully detected. However, harmful prompts in low-resource languages can often bypass these safeguards (Deng et al., 2024; Yong et al., 2024; Yang et al., 2024), i.e., *jailbreaking* the LLM. Modern LLMs are vulnerable to attacks not only from low-resource natural languages, but also from artificial *cipher languages*, e.g., base64 or caesar encoding of English prompts (Wei et al., 2023; Yuan et al., 2024a). This phenomenon also extends beyond text to jailbreaking multimodal LLMs (MLLMs) using modalities such as images (Gong et al., 2025; Liu et al., 2024b) or audio (Yang et al., 2025).

Wei et al. (2023) argue that these attacks are successful due to *mismatched generalization*, a scenario in which the model's safety training does not generalize to other settings, but general performance does. This may happen because pretraining data often includes more diverse data than that available for safety finetuning (Ghosh et al., 2024b).

Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, pages 16173–16187
November 4-9, 2025 ©2025 Association for Computational Linguistics

In this work, we defend against attacks that exploit the mismatched generalization of the safety training of LLMs and MLLMs. Specifically, we defend against attacks that utilize low-resource languages, both natural and cipher languages, as well as attacks employing other modalities, such as images and audio.

We introduce OMNIGUARD, an approach that builds a classifier using the internal representations of a model. These representations are extracted from specific layers that produce representations that are universally similar across multiple languages and across multiple modalities. OMNIGUARD's classifier trained on such representations, is able to accurately detect harmful inputs across 73 languages, with an average of 86.22% accuracy across 53 natural languages and an average of 73.06% accuracy across 20 cipher languages. OMNIGUARD can also detect harmful inputs provided as images with 88.31% and as audio with 93.09% accuracy respectively.

In contrast to popular guard models such as LlamaGuard (Inan et al., 2023), AegisGuard (Ghosh et al., 2024a), or WildGuard (Han et al., 2024), OMNIGUARD does not require training a separate LLM specifically to detect harmfulness. By building a classifier that uses the internal representations of the main LLM or MLLM, OMNIGUARD avoids the overhead of passing the prompt through a separate guard model, making it very efficient.

In summary, our contributions are the following: (1) We propose OMNIGUARD, an approach for detecting harmful prompts, (2) we show that OMNIGUARD accurately detects harmfulness across multiple languages and multiple modalities, (3) we show that OMNIGUARD is very sample-efficient during training, and (4) we show that OMNIGUARD is highly efficient at inference time.

2 Methodology

OMNIGUARD seeks to robustly detect harmful prompts, regardless of their language or modality. We first leverage the tendency of LLMs and MLLMs to create universal representations that are similar across languages (Wendler et al., 2024; Zhao et al., 2024) and across modalities (Wu et al., 2024; Zhuang et al., 2025) in Section 2.1, and then use them to train harmfulness classifiers that robustly detect harmful inputs in Section 2.2.

2.1 Finding language-agnostic representations in an LLM

The first step of OMNIGUARD searches for internal representations of an LLM that are universally shared across languages. We prompt an LLM with English sentences and their translations to other languages, and extract their representations at different layers.[1] For language-agnostic representations, we expect the similarity between the representations of English sentences and the representations of their translations to be similar, and we expect this similarity to be higher than the similarity between representations of two sentences that are not translations of each other (a random pair of sentences). We concretize this notion by defining the *Universality Score* (*U-Score*, Eq. 1), which is the difference between the average cosine similarities of pairs of sentences that are translations of each other and pairs of sentences that are not.

$$\begin{aligned} &\textit{U-Score} := \\ &\qquad \frac{1}{N}\sum_{i\in[N]} \mathrm{CosSim}\left(\mathrm{Emb}(e_i), \mathrm{Emb}(l_i)\right) \\ &-\frac{1}{N(N-1)}\sum_{\substack{i,j\in[N]\\ i\neq j}} \mathrm{CosSim}\left(\mathrm{Emb}(e_i), \mathrm{Emb}(l_j)\right) \end{aligned} \tag{1}$$

where e_i and l_i are sentences in English and their translations to another language.

This procedure can be generalized to new different modalities rather than different languages by changing which embeddings are being used. For example, to determine if internal representations of an MLLM are aligned across modalities, we replace embeddings for a translated piece of text with embeddings from a different modality (e.g. a text caption and its corresponding image, or a text transcription and its corresponding audio clip). See experimental details in Section 3.

2.2 Fitting a harmfulness classifier

After selecting the layer that maximizes the U-Score, we extract embeddings from that layer and use them as inputs to fit a lightweight, supervised classifier that predicts harmfulness. In our experiments, the classifier is a multilayer perceptron with 2 hidden layers (with hidden sizes 512 and 256). At inference time, when a prompt is passed to a model for generation, OMNIGUARD applies

[1] The representation of a prompt is computed by averaging the representation over each token in the prompt.

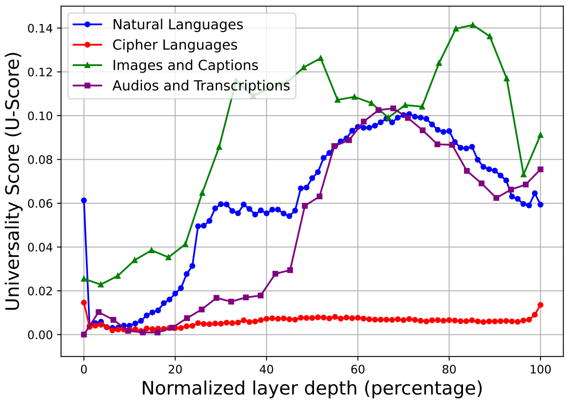

Figure 2: The *U-Score* across different layers for different modalities. (A) Different layers of the model Llama3.3-70B-Instruct for different languages. (B) The *Cross-Model Alignment Score* at different layers of the model (Molmo-7B) for similarity between images and captions. The highest values are obtained with at layers 21-25, indicating better alignment between images and their text captions at these layers. (C) The *Cross-Model Alignment Score* at different layers of the model (Llama-Omni 8B) for similarity between audios and transcriptions. The highest values are obtained with at layers 20-23, indicating better alignment between audios and their text transcriptions at these layers.

this classifier to the embeddings generated by the model, incurring minimal overhead at inference time for safety classification. Note, however, that this approach only applies to open-source models, for which OMNIGUARD can build a classifier by obtaining embeddings. During training, only the lightweight classifier's parameters are learned (the original model is never modified), making the training process data-efficient and inexpensive.

3 Experimental Setup

Table 1 and Table 2 give details on all the models and datasets for this section.

3.1 Selecting universal layers via the U-Score

Selecting language-agnostic layers To select language-agnostic layers, we use a dataset of translated sentence pairs spanning various languages. Specifically, we use sentences in 53 natural languages from the Flores200 dataset and additionally translate the sentences into 20 cipher languages (using encodings such as Caesar shifts, base64, hexadecimal); see a full list in Appendix A. We extract embeddings from each layer of Llama3.3-70B-Instruct for the sentences in all 73 languages and use them to compute the U-Score (averaged over languages). Fig. 2 shows the U-Score as a function of layer depth. For natural languages (blue curve), the U-Score peaks in the middle layers of the model, with the highest values in layer 57 (out of 81 layers). For cipher languages (red curve), the U-Score is much lower than for natural languages, suggesting the model fails to represent semantic similarity in these languages (see analysis in Section 5).

Selecting modality-agnostic layers To select layers aligned between images and captions, we use the MM-Vet v2 dataset, a popular dataset for MLLM evaluation containing 517 examples, each consisting of a text question paired with one or more images. We generate captions for each image using a captioning model (Molmo-7B) and then extract embeddings for each image and its corresponding caption using an MLLM (also Molmo-7B) and use them to compute the U-Score, which peaks in layer 22 (out of 28 layers; see Fig. 2 green curve).

To select layers aligned between text and audio, we use the audio version of the Alpacaeval dataset from VoiceBench, a dataset of 636 audio-transcript pairs. We extract embeddings from each layer of an MLLM (LLaMA-Omni 8B) and use them to compute the U-Score, which peaks at layer 21 (out of 32 layers; see Fig. 2 purple curve).

Overall, we see that LLMs and MLLMs generate representations that are shared across languages and modalities.

3.2 Training and evaluating the harmfulness classifier

3.2.1 Setup for multilingual text attacks

OMNIGUARD classifier. Following Section 3.1, we build a classifier that takes as input embeddings from layer 57 of Llama3.3-70B-Instruct. As training data, we randomly select 2,800 examples from the Aegis AI Content Safety dataset, balancing the benign and harmful classes. Notably, this dataset is about 18× smaller than the training data used by our baseline methods. We translate these English examples to 52 other natural languages (via the Google Translate API) and 20 cipher languages (using fixed rules), totaling 73 languages. We train OMNIGUARD using only half the languages (see list in Appendix A).

Baselines. We compare to many popular guard models (see Table 2) middle row. Notably, *DuoGuard* and *PolyGuard* were trained to detect harmful prompts across multiple languages. For a more direct comparison, we also compare to finetuned

	Dataset name	Citation	HuggingFace ID	Number of examples
General	Flores200	(Team et al., 2022)	`Muennighoff/flores200`	997
	MM-Vet v2	(Yu et al., 2024b)	`whyu/mm-vet-v2`	517
	SST-2	(Socher et al., 2013)	`stanfordnlp/sst2`	1000
Text	Aegis AI Content Safety Dataset	(Ghosh et al., 2024b)	`nvidia/Aegis-AI-Content-Safety-Dataset-1.0`	10,800
	MultiJail	(Deng et al., 2024)	`DAMO-NLP-SG/MultiJail`	315
	Xsafety	(Wang et al., 2024a)	`ToxicityPrompts/XSafety`	28,000
	RTP-LX	(de Wynter et al., 2025)	`ToxicityPrompts/RTP-LX`	30,300
	AyaRedTeaming	(Aakanksha et al., 2024)	`CohereLabs/aya_redteaming`	2662
	Thai Toxicity tweets	(Sirihattasak et al., 2018)	`tmu-nlp/thai_toxicity_tweet`	3,300
	Ukr Toxicity	(Dementieva et al., 2024)	`ukr-detect/ukr-toxicity-dataset`	5,000
	HarmBench (HB)	(Mazeika et al., 2024)	`walledai/HarmBench`	400
	Forbidden Questions (FQ)	(Shen et al., 2024a)	`TrustAIRLab/forbidden_question_set`	390
	Simple Safety Tests	(Vidgen et al., 2024)	`walledai/SimpleSafetyTests`	100
	SaladBench (SaladB)	(Li et al., 2024a)	`walledai/SaladBench`	26,500
	Toxicity Jigsaw (TJS)	(cjadams et al., 2017)	`Arsive/toxicity_classification_jigsaw`	26,000
	Toxic Text	(Corrêa, 2023)	`nicholasKluge/toxic-text`	41,800
	AdvBench	(Zou et al., 2023a)	`walledai/AdvBench`	520
	CodeAttack	(Ren et al., 2024)	`https://github.com/AI45Lab/CodeAttack`	3120
Vision	JailBreakV-28K	(Luo et al., 2024)	`JailbreakV-28K/JailBreakV-28k`	8,000
	VLSafe	(Chen et al., 2024c)	`YangyiYY/LVLM_NLF`	1,110
	FigStep	(Gong et al., 2025)	`https://github.com/wangyu-ovo/MML`	500
	MML SafeBench	(Wang et al., 2024b)	`https://github.com/wangyu-ovo/MML`	2,510
	Hades	(Li et al., 2024e)	`Monosail/HADES`	750
	SafeBench	(Ying et al., 2024)	`Zonghao2025/safebench`	2,300
	MM SafetyBench	(Liu et al., 2024b)	`PKU-Alignment/MM-SafetyBench`	1680
	RedTeamVLM	(Li et al., 2024b)	`MMInstruction/RedTeamingVLM`	200
	VLSBench	(Hu et al., 2025)	`Foreshhh/vlsbench`	2,240
Audio	VoiceBench (Alpacaeval)	(Chen et al., 2024d)	`hlt-lab/voicebench`	636
	AIAH	(Yang et al., 2025)	`https://github.com/YangHao97/RedteamAudioLMMs`	350

Table 1: Details of datasets used for training and evaluation. Some of the text datasets are inherently multilingual : MultiJail (10 languages), XSafety (10 languages), RTP-LX (28 languages), Aya RedTeaming (8 languages), Thai Toxicity tweets (prompts in Thai), and Ukr Toxicity (prompts in Ukrainian). The remaining text datasets are English-only, and were translated to 72 other languages (52 natural and 20 cipher): HarmBench (HB), Forbidden Questions (FQ), Simple Safety Tests, SaladBench (SaladB), Toxicity Jigsaw (TJS), Toxic Text, and AdvBench.

versions of *DuoGuard* and *PolyGuard* using the same 37 languages we use to train OMNIGUARD; Following the original *PolyGuard* paper, we fine-tuned these models using LoRA (Hu et al., 2021) for all linear layers with rank 8 and alpha 16 for one epoch with a learning rate of $2e - 4$.

Datasets. We evaluate on several common text attack benchmarks (see Table 1). We additionally evaluate on three benchmarks from CodeAttacks that transform a harmful query as a list, a stack, or as a string in a Python program, obfuscating the harmfulness. For evaluation in this setup, we transform the harmful prompts from AdvBench and benign prompts from Toxicity Jigsaw datasets in the three code formats and subsample the Toxicity Jigsaw dataset to be of the same size as Advbench. Note that for this experiment, we only trained OMNIGUARD on the English subset of the training dataset.

3.2.2 Setup for vision attacks

OMNIGUARD classifier. Following Section 3.1, we build a classifier that takes as input embeddings from layer 22 of Molmo-7B. As training data, we use 2000 image-query pairs randomly sampled from the JailBreakV-28K dataset and 1024 image-query pairs sampled from the VLSafe dataset as the harmful datapoints and 517 image-query from the MM-Vet v2 dataset as the benign datapoints.

Baselines. We compare to guard models that take an image or image-text pair and output a binary harmfulness classification (see Table 2 bottom row). We train *VLMGuard* on the same training data as OMNIGUARD.

Datasets. We evaluate detecting image/text attacks using several datasets (see Table 1). FigStep and MML Safebench are typographic attacks that embed a harmful prompt in an image. MML Safebench further encrypts a harmful prompt in several variants, such as rotation, mirror images, word replacement, and with base64 encoding. Hades and Safebench consist of images and text queries where the text itself is harmful. MM-safetybench, RTVLM, and VLSBench consist of an image and a query where the text query is seemingly benign, but when combined with the respective image, it is harmful (e.g. see Figure 1).

3.2.3 Setup for audio attacks

OMNIGUARD classifier. Following Section 3.1, we build a classifier that takes as input embeddings from layer 21 of Llama-Omni-8B. We train the classifier on the English portion of the training data we use for the text setting, by using a text-to-speech model to convert the text into audio. We use the open-source Kokoro model as the text-to-speech model.

	Model name	Citation	HuggingFace ID	Rough Parameter Count
General	Llama3.3-70B-Instruct	(Grattafiori et al., 2024)	`meta-llama/Llama-3.3-70B-Instruct`	70B
	Molmo-7B	(Deitke et al., 2024)	`allenai/Molmo-7B-D-0924`	7B
	LLaMA-Omni 8B	(Fang et al., 2025)	`ICTNLP/Llama-3.1-8B-Omni`	8B
	Kokoro	(Hexgrad, 2025)	`hexgrad/Kokoro-82M`	82M
Text	LlamaGuard 1	(Inan et al., 2023)	`meta-llama/LlamaGuard-7b`	7B
	LlamaGuard 2	(Inan et al., 2023)	`meta-llama/Meta-Llama-Guard-2-8B`	8B
	LlamaGuard 3	(Inan et al., 2023)	`meta-llama/Llama-Guard-3-8B`	8B
	AegisGuard Permissive	(Ghosh et al., 2024a)	`nvidia/Aegis-AI-Content-Safety-LlamaGuard-Permissive-1.0`	7B
	AegisGuard Defensive	(Ghosh et al., 2024a)	`nvidia/Aegis-AI-Content-Safety-LlamaGuard-Defensive-1.0`	7B
	WildGuard	(Han et al., 2024)	`allenai/wildguard`	7B
	HarmBench (mistral)	(Mazeika et al., 2024)	`cais/HarmBench-Mistral-7b-val-cls`	7B
	HarmBench (llama)	(Mazeika et al., 2024)	`cais/HarmBench-Llama-2-13b-cls`	13B
	DuoGuard	(Deng et al., 2025)	`DuoGuard/DuoGuard-1B-Llama-3.2-transfer`	1B
	PolyGuard	(Kumar et al., 2025)	`ToxicityPrompts/PolyGuard-Qwen`	7B
Vision	Llama Guard 3 Vision	(Chi et al., 2024)	`meta-llama/Llama-Guard-3-11B-Vision`	11B
	VLMGuard	(Du et al., 2024)	–	2.2M
	LLavaGuard	(Helff et al., 2025)	`AIML-TUDA/LlavaGuard-7B-hf`	7B

Table 2: Model and baseline details.

Baselines. We are unaware of any existing models for detecting harmful audio input. The most relevant approach, SpeechGuard (Peri et al., 2024) adds noise as a defense against potentially harmful audio inputs but does not directly classify harmfulness. To contextualize our results for audio benchmarks, we compare performance to guard models that directly classify the raw text present in the audio (OMNIGUARD and LlamaGuard3).

Datasets. We use the two audio benchmarks (see Table 1 bottom row). We also evaluate on several text jailbreak benchmarks using Kokoro to convert them from text to speech: HB, FQ, Simple Safety Tests, SaladB, and TJS. We use Kokoro for generating text-to-speech versions.

4 Results

Defending against multilingual text attacks Table 3 compares the accuracy of detecting harmful prompts for text benchmarks. Table 3(A) shows results for multilingual benchmarks, where OMNIGUARD achieves the highest accuracy (86.36%) compared to the baselines, and achieves new state-of-the-art performance for 3 benchmarks: MultiJail, RTP-LX, and AyaRedTeaming. The strongest baseline is Polyguard, which yields an average accuracy of 83.19%, despite being trained on a much larger dataset (1.91M examples for Polyguard versus 103K examples for OMNIGUARD). In benchmarks that were translated from English to various other languages, including cipher languages, we again see that OMNIGUARD achieves the highest accuracy (Table 3(B)). Finally, Table 3(C) shows that OMNIGUARD outperforms finetuned versions of DuoGuard and Polyguard on unseen languages, demonstrating that OMNIGUARD can outperform methods that were trained specifically for multilingual harmfulness classification.

Defending against image-based attacks Table 4 shows the accuracy of detecting harmful image and text prompts for (A) pairs consisting of images and text queries, where either the image or both the image and query can be harmful and (B) typographic images with various encryptions. OMNIGUARD achieves the highest performance for both sets of benchmarks (**95.44%** and **79.76%**) while being trained using only about 3500 image-query pairs (compared to about 5500 datapoints used by LlavaGuard). The only benchmark where OMNIGUARD fails to detect harmful prompts is MML Base64, which consists of typographed images of prompts encrypted using base64 encoding.

Defending against audio-based attacks Table 5 shows the accuracy of detecting harmful audio prompts. OMNIGUARD detects harmful audio input with high accuracy across all benchmarks. As we are not aware of any existing defenses for audio jailbreaks, we compare against OMNIGUARD and LlamaGuard3's accuracy in detecting harmful prompts when the same inputs are provided in English text. The accuracy OMNIGUARD achieves in detecting harmful audio inputs is similar to or higher than its performance for detecting harmful text inputs.

Data-efficient adaptation We also evaluate the accuracy of OMNIGUARD and baselines in adapting to out-of-distribution code attacks given very few samples. In this setting, some prior work has speculated that guard models may be very data efficient, as they can make use of few-shot examples in-context (Inan et al., 2023). However, we find that baseline guard models generally struggle to rapidly adapt to this setting given few-shot examples (Figure 3).[2] In contrast, OMNIGUARD is

[2] Note that we omit baseline guard models that achieve 90% accuracy or greater without any few-shot examples, as

		MultiJail	Xsafety	RTP-LX	Aya RedTeaming	Thai Tox	Ukr Tox	Avg.
(A) Multilingual text benchmarks	LlamaGuard 1	39.27	57.01	48.66	54.49	41.31	53.99	49.12
	LlamaGuard 2	48.69	52.66	34.69	58.58	42.86	51.79	48.21
	LlamaGuard 3	66.87	64.34	45.57	63.83	46.73	51.79	56.52
	AegisGuard (P)	61.49	79.78	75.07	78.88	56.09	65.75	69.51
	AegisGuard (D)	79.71	90.77	92.17	89.78	63.34	67.95	80.62
	WildGuard	42.55	71.23	71.94	61.45	40.42	55.03	57.10
	HarmBench (llama)	0.22	0.14	0.0	0.03	39.04	50.1	14.92
	HarmBench (mistral)	2.4	5.65	5.14	7.39	40.42	50.55	18.59
	MD-Judge	25.78	53.58	66.46	46.20	39.48	53.89	47.56
	DuoGuard	39.20	63.42	66.57	61.80	45.63	50.75	54.56
	PolyGuard	82.00	**96.41**	83.86	90.34	**70.43**	**76.07**	83.19
	OmniGuard	**93.83**	93.64	**94.55**	**94.31**	68.7	73.1	**86.36**

		HarmBench	FQ	SimpleST	SaladB	TJS	ToxText	AdvBench	Avg.
(B) Translated text benchmarks	LlamaGuard 1	32.47	23.75	34.32	23.27	62.49	65.55	34.39	39.46
	LlamaGuard 2	57.19	43.72	50.71	34.54	58.21	62.17	56.95	51.93
	LlamaGuard 3	70.02	53.25	67.81	46.30	62.33	70.87	70.26	62.98
	AegisGuard (P)	62.16	43.01	56.55	44.92	73.69	72.80	62.12	59.32
	AegisGuard (D)	88.53	76.67	87.64	78.27	71.38	68.72	**90.77**	80.28
	WildGuard	33.64	31.20	33.90	27.37	66.61	67.27	39.98	42.85
	HarmBench (llama)	0.03	0.11	0.01	0.07	48.62	49.97	0.01	14.12
	HarmBench (mistral)	2.32	1.75	2.04	1.66	50.53	50.69	1.7	15.81
	MD-Judge	16.19	12.11	22.29	13.81	65.34	64.26	25.67	31.38
	DuoGuard	20.44	44.36	28.79	36.88	68.57	69.07	28.58	42.38
	PolyGuard	66.22	56.05	62.53	54.88	**78.34**	**76.52**	67.96	66.07
	OmniGuard	**89.13**	**89.57**	**89.62**	**87.30**	76.68	75.07	86.59	**84.85**

		HarmBench	FQ	SimpleST	SaladB	TJS	ToxText	AdvBench	Avg.
(C) Unseen langs.	FT DuoGuard	23.59	39.08	28.14	33.29	54.1	53.23	28.29	37.1
	FT PolyGuard	72.45	79.84	76.81	76.85	**74.07**	**72.33**	73.55	75.13
	OmniGuard	**86.51**	**86.65**	**86.42**	**85.01**	72.82	71.44	**84.29**	**81.88**

Table 3: Accuracy of detecting harmful prompts for text attack benchmarks that are (A) multilingual benchmarks, (B) English translated to 73 languages, and (C) English translated to languages not seen at training time. In all settings, OmniGuard achieves the highest performance. Table B1 further stratifies these results by high-resource, low-resource, and cipher languages.

		Hades	VLSBench	MM-SafetyBench	SafeBench	RTVLM	FigStep	Avg.
(A) Image +Query	Llama3 Vision GRD	76.00	3.97	31.90	68.40	56.50	47.40	47.36
	VLMGuard	98.00	74.56	92.20	73.90	**94.00**	99.80	88.74
	LLavaGuard	23.73	42.08	10.95	12.10	18.50	3.40	18.46
	OmniGuard	**100.00**	**92.24**	**99.82**	**91.60**	89.00	**100.00**	**95.44**

		MML Rotate	MML Mirror	MML W.R.	MML Q.R.	MML Base64	Avg.
(B) Typographed image	Llama3 Vision GRD	83.20	68.00	96.40	25.40	**98.80**	74.36
	VLMGuard	6.80	21.00	**100.0**	86.20	0.20	42.84
	LLavaGuard	0.00	0.00	0.00	11.40	0.00	2.28
	OmniGuard	**100.0**	**100.0**	99.60	**98.80**	0.40	**79.76**

Table 4: Accuracy of detecting harmful queries in multimodal benchmarks for (A) image-query pairs and (B) typographed images with encrypted text. OmniGuard achieves the highest performance for both kinds of benchmarks.

	AIAH	SafeBench (M)	SafeBench (F)	HB	FQ	SimpleST	SaladB	TJS	AdvBench
OmniGuard (Audio)	**91.14**	**94.4**	**93.8**	95.98	90.42	97.0	**94.21**	82.03	**98.85**
OmniGuard (text-en)	-	-	-	92.0	**93.3**	93.0	90.2	**93.2**	90.0
LlamaGuard3 (text-en)	-	-	-	**97.32**	78.75	**99.0**	67.03	72.16	98.07

Table 5: Accuracy of detecting harmful queries in audio. OmniGuard is able to detect harmful audio inputs with high accuracy across all benchmarks. Since there are no baselines for detecting harmful prompts in audio, we compare the performance against OmniGuard's and LlamaGuard3 when the same benchmarks are provided as text in English.

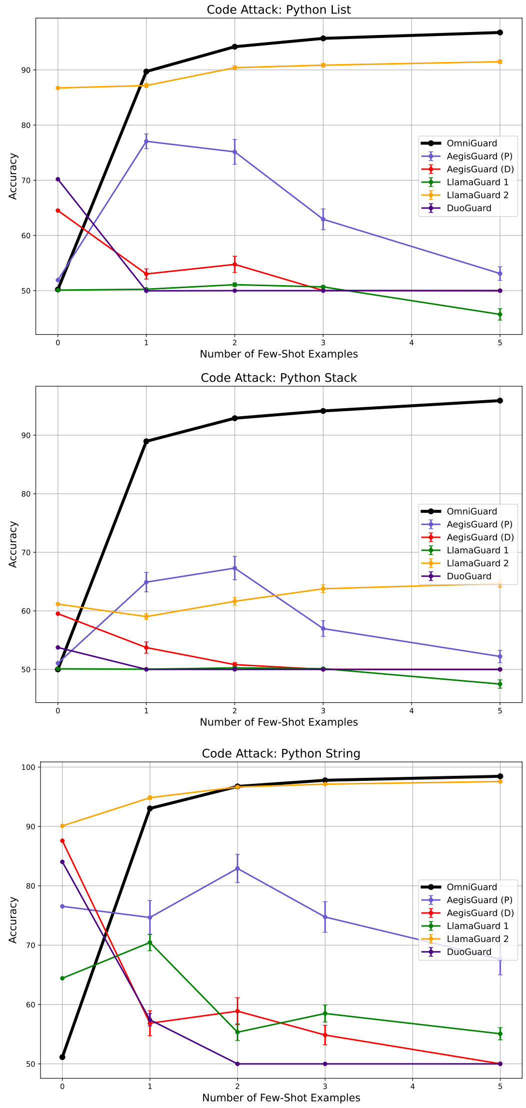

Figure 3: Accuracy of detecting harmful prompts in a few-shot setting. As few-shot examples are provided, OMNIGUARD quickly achieves near-perfect accuracy, despite the attacks being quite different from its training data (e.g. without any few-shot examples, OMNIGUARD's accuracy is close to 50%). In contrast, the guard model baselines improve their accuracy slowly in a few-shot setting, despite sometimes having seen similar code attacks in their training data. Accuracies are averaged over 50 random sets of few-shot examples; error bars show the standard error of the mean.

able to rapidly achieve close to 100% accuracy for all three benchmarks by updating its lightweight parameters using less than five examples.

5 Analysis

	Thai Tox	Ukr Tox	TJS	ToxText	Avg.
Layer 10	62.1	65.5	66.95	61.89	64.42
Layer 75	65.2	66.4	70.72	65.79	68.26
Last Layer	63.1	51.2	61.33	56.76	59.05
U-Score selected layer (57)	**68.7**	**73.1**	**76.8**	**75.07**	**73.4**

Table 6: OMNIGUARD's accuracy of detecting harmful prompts when trained using representations from different model layers.

Guard Method	Inference Time (s) ↓
LlamaGuard 3	87.25
AegisGuard (D)	152.26
WildGuard	306.14
MD-Judge	128.26
DuoGuard	4.85
PolyGuard	409.90
OMNIGUARD	**0.04**

Table 7: Average inference time required for harmfulness prediction on the AdvBench dataset (averaged over 5 languages). OMNIGUARD is about 120× faster than the fastest baseline (DuoGuard).

Effect of U-Score-based layer selection. We perform ablation experiments to determine the effect of selecting the appropriate layer for training the OMNIGUARD classifier. For the text-only model, we compare the U-Score-selected layer (57) to 3 other layers (layer 10, layer 75, and the last layer) when used for a set of toxicity prediction tasks. Table 6 shows that the representations from the layer with the highest *U-Score* result in significantly better harmfulness classification accuracy, improving between 5% and 14% compared to the other layers. We show ablation over more layers in Table B2.

OMNIGUARD's efficiency OMNIGUARD is highly efficient at inference time because it re-uses the internal representations of the main LLM that is already processing the user query for generation. Therefore, its compute time is only that of a lightweight multilayer perceptron, making it much faster than baseline guard models (note that this does limit OMNIGUARD to only work when the generation model is open-source, so embeddings can be extracted). Table 7 shows the inference time required by various guard models to predict the harmfulness of prompts in the AdvBench dataset in English, translated to Spanish, French, Telugu, and base64 encoding. OMNIGUARD is the fastest and is about **120× faster** than the fastest baseline (DuoGuard). Inference time as measured on a machine with 1 L40 GPU, 4 CPUs, and 50 GB RAM.

Performance comparison across base LLMs We compare OMNIGUARD's accuracy when using different base LLMs in Table B3. We trained the classifiers on the layers with the best U-scores for each model. We find that the average accuracy for the moderator model trained using smaller LLMs is lower than the moderator model trained using

their training data likely explicitly includes code attacks.

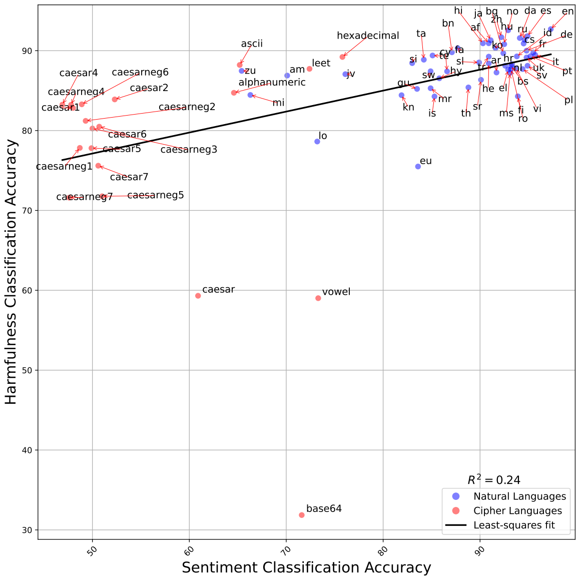

Figure 4: Comparison of accuracy of classifying sentiments in various languages compared to detecting harmful prompts in those languages using OMNIGUARD. In both cases the LLM is Llama3.3-70B-Instruct.

the larger Llama3.3-70B-Instruct model.

Performance comparison across languages. We now analyze the harmfulness classification accuracy of OMNIGUARD by language, and compare it to the underlying LLM's sentiment classification accuracy for the same language (Fig. 4). We measure harmfulness classification accuracy using OMNIGUARD on all the datasets in Table 3 and sentiment classification accuracy using Llama3.3-70B-Instruct with zero-shot prompting on 72 translated versions of the SST-2 dataset (translated to all the languages we consider).

We observe that the accuracies are generally correlated, indicating that OMNIGUARD is able to defend well in languages for which the LLM is more coherent/susceptible to attack. Unsurprisingly, the accuracies for natural languages are higher than the accuracies for cipher languages. Nevertheless, harmfulness classification accuracy can be fairly high, even when sentiment classification accuracy is near chance (50%).

6 Related Work

Jailbreak Attacks in LLMs Several techniques have recently emerged to attack or jailbreak LLMs. Early techniques relied on manual effort and were very time-intensive (Shen et al., 2024b; Andriushchenko et al., 2024). Later techniques automated this process, e.g., Zou et al. (2023b); Jones et al. (2023); Zhu et al. (2023) proposed gradient-based approaches to identify inputs to jailbreak LLMs with white-box access. Another set of techniques start from a set of human written prompts and modify them using approaches like genetic algorithms (Liu et al., 2024a; Lapid et al., 2024; Li et al., 2024d), fuzzing (Yu et al., 2024a), or reinforcement learning (Chen et al., 2024b) to automatically produce prompts for jailbreaking. Another set of techniques, use a helper LLM to generate prompts that attack a target LLM (Chao et al., 2024; Ding et al., 2024; Mehrotra et al., 2024). Finally, Wei et al. (2024); Wang et al. (2023); Anil et al. (2024); Pernisi et al. (2024) use simple in-context demonstrations to jailbreak the models by overcoming its safety training and Russinovich et al. (2024); Li et al. (2024c) propose using multi-turn dialogues to jailbreak models.

Multilingual Jailbreak Attacks Most of the aforementioned jailbreak techniques focus on attacks in English, against which significant defense exists both at the model and system level. To tackle this, a novel set of techniques have emerged that attack models using inputs in various languages or obfuscations that are able to bypass the safety guardrails. Deng et al. (2024); Yong et al. (2024); Wang et al. (2024a); Yang et al. (2024); Yoo et al. (2024); Upadhayay and Behzadan (2024); Song et al. (2024) demonstrated that attacking models using mid and low resources languages led to higher attack success rates, compared to the case of attacking the model in high-resource languages like English.

Going beyond natural languages, a newer set of works propose using cipher characters or languages to evade the safety filters, e.g., Jin et al. (2024) propose interspersing cipher characters in between text, Jiang et al. (2024) propose replacing the unsafe words with their ASCII art versions, and Yuan et al. (2024a) propose prompting models in cipher languages like Morse, Atbash, Caesar.

Multimodal Jailbreak Attacks Using modalities apart from text aims to explore a completely new attack surface, like images or audios. Several recent works have shown that MLLMs remain vulnerable to being jailbroken when prompted with images or audios that have a harmful query (the same harmful query in text would be easily detected as harmful). Liu et al. (2024b) show that using a prompt with a correlated image, e.g., using an image of a bomb when asking the model to answer the question: *How to make a bomb?* is more likely to jailbreak a model than when using an uncorrelated image. Hu et al. (2025) argue that providing a harmful image with a benign query (see Figure 1) further increases the potential of jailbreaking the model. Gong et al. (2025) and Wang et al. (2024b) demonstrate jailbreaking models by simply typographically embedding harmful queries in an image.

Safety moderation in LLMs Safety moderation in LLMs broadly fits into two categories: intrinsic and extrinsic. Intrinsic mechanisms include finetuning or RLHF training on an LLM (Bianchi et al., 2024; Chen et al., 2024a; Yuan et al., 2024b; Ouyang et al., 2022b; Bai et al., 2022; Dai et al., 2023). Extrinsic safety mechanisms utilize external models to detect harmful inputs and responses; these models can either be simple filters or use guard models. Jain et al. (2023); Alon and Kamfonas (2023); Hu et al. (2024) propose using perplexity filtering for detecting harmful prompts. Guard models defend LLMs by training separate LLMs to detect harmful text (see Table 2). Separately, a line of work has introduced interpretability methods to transparently expose safety concerns in LLMs (Bereska and Gavves, 2024; Singh et al., 2023; Arditi et al., 2024; Benara et al., 2024).

A few other works also use internal model representations to defend against harmful inputs. However, they defer from our work: OMNIGUARD is a standalone content safety classification model while the other approaches like Jailbreak Antidote (Shen et al., 2025) and AdaSteer (Zhao et al., 2025) directly change the internal representations to defend against harmful inputs. Therefore, OMNIGUARD is similar to other safety classification models like LlamaGuard and WildGuard. Additionally, OMNIGUARD can detect harmful prompts in multiple languages (both natural and cipher) and multiple modalities using the same method while the other approaches only defend against attacks in English text. OMNIGUARD is also extremely efficient in producing safety predictions while the other approaches take as much time as the inference of the underlying LLM, which can typically take several seconds to minutes.

Safety against multilingual attacks *DuoGuard* (Deng et al., 2025) and *PolyGuard* (Kumar et al., 2025) are the two previous guard models that were specifically trained to defend against multilingual attacks. DuoGuard uses a two-player RL-driven mechanism to generate harmful data in multiple languages and uses that to finetune a Llama3.2-1B model. PolyGuard collects an extensive dataset of 1.91M samples of harmful and benign datapoints in 17 languages and uses that to finetune a Qwen-2.5-7B-Instruct model.

Safety moderation in MLLMs Relatively few works have tackled detecting harmful prompts in multimodal settings (see Table 1 and Table 2). Chi et al. (2024) propose LlamaGuard3-11B-Vision (a finetuned version of Llama-3-11B-Vision) for detecting unsafe inputs in images and texts. Du et al. (2024) and Helff et al. (2025) propose other approaches for the same task. OMNIGUARD achieves higher accuracy in detecting harmful images and prompts compared to these approaches, and to the best of our knowledge is the first guard model for harmful audio inputs.

7 Conclusions

We propose OMNIGUARD, an approach for training a safety moderation classifier using the internal representations of an LLM or MLLM that are universally similar across languages and modalities. Our approach consists of two steps: first, we identify these universally similar representations and then we use them to train a harmfulness classifier. We find that OMNIGUARD accurately detects harmful prompts across languages, including low-resource languages as well as cipher languages, and also across modalities – images and audios. We show that OMNIGUARD allows to train more efficient safety moderation classifiers (both in training time and in inference time) compared to standard guard models, and conclude that our approach is superior in both accuracy and efficiency across languages and modalities.

Limitations

While OMNIGUARD achieves state-of-the-art performance for detecting harmful prompts across languages and modalities, its performance depends on the underlying model. If the underlying model does not understand the language or an image or audio input, OMNIGUARD might not be able to detect if the input is harmful. However, this limitation is not unique to OMNIGUARD, and existing approaches suffer from the same limitation.

Our approach also relies on the existence of universally similar representations, which we empirically found to exist across models and modalities. However, we did not exhaustively check all models and this assumption might not hold for models that we have not used in this work. Moreover, OMNIGUARD requires access to internal representations of a model, making it inapplicable to closed-source models.

Lastly, the results we report are based on a fixed set of evaluation datasets that are standard benchmarks used in the research area of AI safety moderation. While OMNIGUARD performs well across the datasets we experiment with, its performance in real-world settings might differ.

Ethics. While this work seeks to mitigate the risks of LLM deployment in high-risk scenarios, OMNIGUARD is not a perfect classifier and unexpected failures may allow for the harmful misuse of LLMs.

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. *Preprint*, arXiv:2406.18682.

Gabriel Alon and Michael Kamfonas. 2023. Detecting language model attacks with perplexity. *Preprint*, arXiv:2308.14132.

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. 2024. Jailbreaking leading safety-aligned llms with simple adaptive attacks.

Cem Anil, Esin Durmus, ..., and David Duvenaud. 2024. Many-shot jailbreaking.

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. *Advances in Neural Information Processing Systems*, 37:136037–136083.

Yuntao Bai, Andy Jones, Kamal Ndousse, Amanda Askell, Anna Chen, Nova DasSarma, Dawn Drain, Stanislav Fort, Deep Ganguli, Tom Henighan, Nicholas Joseph, Saurav Kadavath, Jackson Kernion, Tom Conerly, Sheer El-Showk, Nelson Elhage, Zac Hatfield-Dodds, Danny Hernandez, Tristan Hume, and 12 others. 2022. Training a helpful and harmless assistant with reinforcement learning from human feedback.

Vinamra Benara, Chandan Singh, John X Morris, Richard Antonello, Ion Stoica, Alexander G Huth, and Jianfeng Gao. 2024. Crafting interpretable embeddings by asking llms questions. *arXiv preprint arXiv:2405.16714*.

Yoshua Bengio, Geoffrey Hinton, Andrew Yao, Dawn Song, Pieter Abbeel, Trevor Darrell, Yuval Noah Harari, Ya-Qin Zhang, Lan Xue, Shai Shalev-Shwartz, and 1 others. 2024. Managing extreme ai risks amid rapid progress. *Science*, 384(6698):842–845.

Leonard Bereska and Efstratios Gavves. 2024. Mechanistic interpretability for ai safety–a review. *arXiv preprint arXiv:2404.14082*.

Federico Bianchi, Mirac Suzgun, Giuseppe Attanasio, Paul Rottger, Dan Jurafsky, Tatsunori Hashimoto, and James Zou. 2024. Safety-tuned LLaMAs: Lessons from improving the safety of large language models that follow instructions. In *The Twelfth International Conference on Learning Representations*.

Rishi Bommasani, Drew A. Hudson, ..., and Percy Liang. 2022. On the opportunities and risks of foundation models. *Preprint*, arXiv:2108.07258.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J. Pappas, and Eric Wong. 2024. Jailbreaking black box large language models in twenty queries. *Preprint*, arXiv:2310.08419.

Kai Chen, Chunwei Wang, Kuo Yang, Jianhua Han, Lanqing Hong, Fei Mi, Hang Xu, Zhengying Liu, Wenyong Huang, Zhenguo Li, Dit-Yan Yeung, Lifeng Shang, Xin Jiang, and Qun Liu. 2024a. Gaining wisdom from setbacks: Aligning large language models via mistake analysis.

Xuan Chen, Yuzhou Nie, Wenbo Guo, and Xiangyu Zhang. 2024b. When llm meets drl: Advancing jailbreaking efficiency via drl-guided search. *Preprint*, arXiv:2406.08705.

Yangyi Chen, Karan Sikka, Michael Cogswell, Heng Ji, and Ajay Divakaran. 2024c. Dress: Instructing large vision-language models to align and interact with humans via natural language feedback. *Preprint*, arXiv:2311.10081.

Yiming Chen, Xianghu Yue, Chen Zhang, Xiaoxue Gao, Robby T. Tan, and Haizhou Li. 2024d. Voicebench: Benchmarking llm-based voice assistants. *Preprint*, arXiv:2410.17196.

Jianfeng Chi, Ujjwal Karn, Hongyuan Zhan, Eric Smith, Javier Rando, Yiming Zhang, Kate Plawiak, Zacharie Delpierre Coudert, Kartikeya Upasani, and Mahesh Pasupuleti. 2024. Llama guard 3 vision: Safeguarding human-ai image understanding conversations. *Preprint*, arXiv:2411.10414.

cjadams, Jeffrey Sorensen, Julia Elliott, Lucas Dixon, Mark McDonald, nithum, and Will Cukierski. 2017. Toxic comment classification challenge.

Nicholas Kluge Corrêa. 2023. Aira.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2023. Safe rlhf: Safe reinforcement learning from human feedback. *Preprint*, arXiv:2310.12773.

Adrian de Wynter, Ishaan Watts, Tua Wongsangaroonsri, Minghui Zhang, Noura Farra, Nektar Ege Altıntoprak, Lena Baur, Samantha Claudet, Pavel Gajdušek, Qilong Gu, Anna Kaminska, Tomasz Kaminski, Ruby Kuo, Akiko Kyuba, Jongho Lee, Kartik Mathur, Petter Merok, Ivana Milovanović, Nani Paananen, and 13 others. 2025. Rtp-lx: Can llms evaluate toxicity in multilingual scenarios? *Proceedings of the AAAI Conference on Artificial Intelligence*, 39(27):27940–27950.

Matt Deitke, Christopher Clark, Sangho Lee, Rohun Tripathi, Yue Yang, Jae Sung Park, Mohammadreza Salehi, Niklas Muennighoff, Kyle Lo, Luca Soldaini, Jiasen Lu, Taira Anderson, Erin Bransom, Kiana Ehsani, Huong Ngo, YenSung Chen, Ajay Patel, Mark Yatskar, Chris Callison-Burch, and 31 others. 2024. Molmo and pixmo: Open weights and open data for state-of-the-art vision-language models. *Preprint*, arXiv:2409.17146.

Daryna Dementieva, Valeriia Khylenko, Nikolay Babakov, and Georg Groh. 2024. Toxicity classification in Ukrainian. In *Proceedings of the 8th Workshop on Online Abuse and Harms (WOAH 2024)*, pages 244–255, Mexico City, Mexico. Association for Computational Linguistics.

Yihe Deng, Yu Yang, Junkai Zhang, Wei Wang, and Bo Li. 2025. Duoguard: A two-player rl-driven framework for multilingual llm guardrails. *Preprint*, arXiv:2502.05163.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024. Multilingual jailbreak challenges in large language models. In *The Twelfth International Conference on Learning Representations*.

Peng Ding, Jun Kuang, Dan Ma, Xuezhi Cao, Yunsen Xian, Jiajun Chen, and Shujian Huang. 2024. A wolf in sheep's clothing: Generalized nested jailbreak prompts can fool large language models easily. 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 2136–2153, Mexico City, Mexico. Association for Computational Linguistics.

Xuefeng Du, Reshmi Ghosh, Robert Sim, Ahmed Salem, Vitor Carvalho, Emily Lawton, Yixuan Li, and Jack W. Stokes. 2024. Vlmguard: Defending vlms against malicious prompts via unlabeled data. *Preprint*, arXiv:2410.00296.

Qingkai Fang, Shoutao Guo, Yan Zhou, Zhengrui Ma, Shaolei Zhang, and Yang Feng. 2025. Llama-omni: Seamless speech interaction with large language models. *Preprint*, arXiv:2409.06666.

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

Shaona Ghosh, Prasoon Varshney, Makesh Narsimhan Sreedhar, Aishwarya Padmakumar, Traian Rebedea, Jibin Rajan Varghese, and Christopher Parisien. 2024b. Aegis2. 0: A diverse ai safety dataset and risks taxonomy for alignment of llm guardrails. In *Neurips Safe Generative AI Workshop*.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. 2025. Figstep: Jailbreaking large vision-language models via typographic visual prompts. *Preprint*, arXiv:2311.05608.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

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.

Lukas Helff, Felix Friedrich, Manuel Brack, Kristian Kersting, and Patrick Schramowski. 2025. Llavaguard: An open vlm-based framework for safeguarding vision datasets and models. *Preprint*, arXiv:2406.05113.

Hexgrad. 2025. Kokoro-82m.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *Preprint*, arXiv:2106.09685.

Xuhao Hu, Dongrui Liu, Hao Li, Xuanjing Huang, and Jing Shao. 2025. Vlsbench: Unveiling visual leakage in multimodal safety. *Preprint*, arXiv:2411.19939.

Zhengmian Hu, Gang Wu, Saayan Mitra, Ruiyi Zhang, Tong Sun, Heng Huang, and Viswanathan Swaminathan. 2024. Token-level adversarial prompt detection based on perplexity measures and contextual information. *Preprint*, arXiv:2311.11509.

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.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. 2023. Baseline defenses for adversarial attacks against aligned language models. *Preprint*, arXiv:2309.00614.

Fengqing Jiang, Zhangchen Xu, Luyao Niu, Zhen Xiang, Bhaskar Ramasubramanian, Bo Li, and Radha Poovendran. 2024. Artprompt: Ascii art-based jailbreak attacks against aligned llms.

Haibo Jin, Andy Zhou, Joe D. Menke, and Haohan Wang. 2024. Jailbreaking large language models against moderation guardrails via cipher characters. *Preprint*, arXiv:2405.20413.

Erik Jones, Anca Dragan, Aditi Raghunathan, and Jacob Steinhardt. 2023. Automatically auditing large language models via discrete optimization. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 15307–15329. PMLR.

Priyanshu Kumar, Devansh Jain, Akhila Yerukola, Liwei Jiang, Himanshu Beniwal, Thomas Hartvigsen, and Maarten Sap. 2025. Polyguard: A multilingual safety moderation tool for 17 languages. *Preprint*, arXiv:2504.04377.

Raz Lapid, Ron Langberg, and Moshe Sipper. 2024. Open sesame! universal black box jailbreaking of large language models. *Preprint*, arXiv:2309.01446.

Jan Leike, David Krueger, Tom Everitt, Miljan Martic, Vishal Maini, and Shane Legg. 2018. Scalable agent alignment via reward modeling: a research direction. *Preprint*, arXiv:1811.07871.

Lijun Li, Bowen Dong, Ruohui Wang, Xuhao Hu, Wangmeng Zuo, Dahua Lin, Yu Qiao, and Jing Shao. 2024a. SALAD-bench: A hierarchical and comprehensive safety benchmark for large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 3923–3954, Bangkok, Thailand. Association for Computational Linguistics.

Mukai Li, Lei Li, Yuwei Yin, Masood Ahmed, Zhenguang Liu, and Qi Liu. 2024b. Red teaming visual language models. *Preprint*, arXiv:2401.12915.

Nathaniel Li, Ziwen Han, Ian Steneker, Willow Primack, Riley Goodside, Hugh Zhang, Zifan Wang, Cristina Menghini, and Summer Yue. 2024c. Llm defenses are not robust to multi-turn human jailbreaks yet.

Xiaoxia Li, Siyuan Liang, Jiyi Zhang, Han Fang, Aishan Liu, and Ee-Chien Chang. 2024d. Semantic mirror jailbreak: Genetic algorithm based jailbreak prompts against open-source llms. *Preprint*, arXiv:2402.14872.

Yifan Li, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. 2024e. Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models. In *ECCV 2024*, page 174–189, Berlin, Heidelberg. Springer-Verlag.

Xiaogeng Liu, Nan Xu, Muhao Chen, and Chaowei Xiao. 2024a. AutoDAN: Generating stealthy jailbreak prompts on aligned large language models. In *The Twelfth International Conference on Learning Representations*.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. 2024b. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *ECCV 2024*, page 386–403, Berlin, Heidelberg. Springer-Verlag.

Weidi Luo, Siyuan Ma, Xiaogeng Liu, Xiaoyu Guo, and Chaowei Xiao. 2024. Jailbreakv-28k: A benchmark for assessing the robustness of multimodal large language models against jailbreak attacks. *Preprint*, arXiv:2404.03027.

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.

Anay Mehrotra, Manolis Zampetakis, Paul Kassianik, Blaine Nelson, Hyrum Anderson, Yaron Singer, and Amin Karbasi. 2024. Tree of attacks: Jailbreaking black-box llms automatically. *Preprint*, arXiv:2312.02119.

OpenAI. 2025. Openai moderation endpoint guide. Accessed: 2025-05-18.

Long Ouyang, Jeff Wu, Xu Jiang, Diogo Almeida, Carroll L. Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. 2022a. Training language models to follow instructions with human feedback. In *Proceedings of the 36th International Conference on Neural Information Processing Systems*, NIPS '22, Red Hook, NY, USA. Curran Associates Inc.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. 2022b. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems*, volume 35, pages 27730–27744. Curran Associates, Inc.

Raghuveer Peri, Sai Muralidhar Jayanthi, Srikanth Ronanki, Anshu Bhatia, Karel Mundnich, Saket Dingliwal, Nilaksh Das, Zejiang Hou, Goeric Huybrechts, Srikanth Vishnubhotla, Daniel Garcia-Romero, Sundararajan Srinivasan, Kyu Han, and Katrin Kirchhoff. 2024. Speechguard: Exploring the adversarial robustness of multi-modal large language models. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 10018–10035, Bangkok, Thailand. Association for Computational Linguistics.

Fabio Pernisi, Dirk Hovy, and Paul Röttger. 2024. Compromesso! italian many-shot jailbreaks undermine the safety of large language models.

Qibing Ren, Chang Gao, Jing Shao, Junchi Yan, Xin Tan, Wai Lam, and Lizhuang Ma. 2024. CodeAttack: Revealing safety generalization challenges of large language models via code completion. In *Findings of the Association for Computational Linguistics ACL 2024*, pages 11437–11452. Association for Computational Linguistics.

Mark Russinovich, Ahmed Salem, and Ronen Eldan. 2024. Great, now write an article about that: The crescendo multi-turn llm jailbreak attack. *Preprint*, arXiv:2404.01833.

Guobin Shen, Dongcheng Zhao, Yiting Dong, Xiang He, and Yi Zeng. 2025. Jailbreak antidote: Runtime safety-utility balance via sparse representation adjustment in large language models. In *The Thirteenth International Conference on Learning Representations*.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024a. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. In *Proceedings of the 2024 on ACM SIGSAC Conference on Computer and Communications Security*, CCS '24, page 1671–1685, New York, NY, USA. Association for Computing Machinery.

Xinyue Shen, Zeyuan Chen, Michael Backes, Yun Shen, and Yang Zhang. 2024b. "do anything now": Characterizing and evaluating in-the-wild jailbreak prompts on large language models. *Preprint*, arXiv:2308.03825.

Chandan Singh, Armin Askari, Rich Caruana, and Jianfeng Gao. 2023. Augmenting interpretable models with large language models during training. *Nature Communications*, 14(1):7913.

Sugan Sirihattasak, Mamoru Komachi, and Hiroshi Ishikawa. 2018. Annotation and classification of toxicity for thai twitter. In *Proceedings of LREC 2018 Workshop and the 2nd Workshop on Text Analytics for Cybersecurity and Online Safety (TA-COS'18)*, Miyazaki, Japan.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Jiayang Song, Yuheng Huang, Zhehua Zhou, and Lei Ma. 2024. Multilingual blending: Llm safety alignment evaluation with language mixture. *Preprint*, arXiv:2407.07342.

NLLB Team, Marta R. Costa-jussà, James Cross, Onur Çelebi, Maha Elbayad, Kenneth Heafield, Kevin Heffernan, Elahe Kalbassi, Janice Lam, Daniel Licht, Jean Maillard, Anna Sun, Skyler Wang, Guillaume Wenzek, Al Youngblood, Bapi Akula, Loic Barrault, Gabriel Mejia Gonzalez, Prangthip Hansanti, and 20 others. 2022. No language left behind: Scaling human-centered machine translation. *Preprint*, arXiv:2207.04672.

Bibek Upadhayay and Vahid Behzadan. 2024. Sandwich attack: Multi-language mixture adaptive attack on llms. *Preprint*, arXiv:2404.07242.

Bertie Vidgen, Nino Scherrer, Hannah Rose Kirk, Rebecca Qian, Anand Kannappan, Scott A. Hale, and Paul Röttger. 2024. Simplesafetytests: a test suite for identifying critical safety risks in large language models. *Preprint*, arXiv:2311.08370.

Jiongxiao Wang, Zichen Liu, Keun Hee Park, Zhuojun Jiang, Zhaoheng Zheng, Zhuofeng Wu, Muhao Chen, and Chaowei Xiao. 2023. Adversarial demonstration attacks on large language models. *Preprint*, arXiv:2305.14950.

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen tse Huang, Wenxiang Jiao, and Michael R. Lyu. 2024a. All languages matter: On the multilingual safety of large language models.

Yu Wang, Xiaofei Zhou, Yichen Wang, Geyuan Zhang, and Tianxing He. 2024b. Jailbreak large vision-language models through multi-modal linkage. *Preprint*, arXiv:2412.00473.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does LLM safety training fail? In *Thirty-seventh Conference on Neural Information Processing Systems*.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. 2024. Jailbreak and guard aligned language models with only few in-context demonstrations. *Preprint*, arXiv:2310.06387.

Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. 2024. Do llamas work in english? on the latent language of multilingual transformers.

Zhaofeng Wu, Xinyan Velocity Yu, Dani Yogatama, Jiasen Lu, and Yoon Kim. 2024. The semantic hub hypothesis: Language models share semantic representations across languages and modalities. *Preprint*, arXiv:2411.04986.

Hao Yang, Lizhen Qu, Ehsan Shareghi, and Gholamreza Haffari. 2025. Audio is the achilles' heel: Red teaming audio large multimodal models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics*, pages 9292–9306, Albuquerque, New Mexico.

Yahan Yang, Soham Dan, Dan Roth, and Insup Lee. 2024. Benchmarking llm guardrails in handling multilingual toxicity. *Preprint*, arXiv:2410.22153.

Zonghao Ying, Aishan Liu, Siyuan Liang, Lei Huang, Jinyang Guo, Wenbo Zhou, Xianglong Liu, and Dacheng Tao. 2024. Safebench: A safety evaluation framework for multimodal large language models. *Preprint*, arXiv:2410.18927.

Zheng-Xin Yong, Cristina Menghini, and Stephen H. Bach. 2024. Low-resource languages jailbreak gpt-4. *Preprint*, arXiv:2310.02446.

Haneul Yoo, Yongjin Yang, and Hwaran Lee. 2024. Csrt: Evaluation and analysis of llms using code-switching red-teaming dataset. *Preprint*, arXiv:2406.15481.

Jiahao Yu, Xingwei Lin, Zheng Yu, and Xinyu Xing. 2024a. Gptfuzzer: Red teaming large language models with auto-generated jailbreak prompts. *Preprint*, arXiv:2309.10253.

Weihao Yu, Zhengyuan Yang, Lingfeng Ren, Linjie Li, Jianfeng Wang, Kevin Lin, Chung-Ching Lin, Zicheng Liu, Lijuan Wang, and Xinchao Wang. 2024b. Mm-vet v2: A challenging benchmark to evaluate large multimodal models for integrated capabilities. *Preprint*, arXiv:2408.00765.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Pinjia He, Shuming Shi, and Zhaopeng Tu. 2024a. GPT-4 is too smart to be safe: Stealthy chat with LLMs via cipher. In *The Twelfth International Conference on Learning Representations*.

Youliang Yuan, Wenxiang Jiao, Wenxuan Wang, Jen tse Huang, Jiahao Xu, Tian Liang, Pinjia He, and Zhaopeng Tu. 2024b. Refuse whenever you feel unsafe: Improving safety in llms via decoupled refusal training.

Weixiang Zhao, Jiahe Guo, Yulin Hu, Yang Deng, An Zhang, Xingyu Sui, Xinyang Han, Yanyan Zhao, Bing Qin, Tat-Seng Chua, and Ting Liu. 2025. Adasteer: Your aligned llm is inherently an adaptive jailbreak defender.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism?

Sicheng Zhu, Ruiyi Zhang, Bang An, Gang Wu, Joe Barrow, Zichao Wang, Furong Huang, Ani Nenkova, and Tong Sun. 2023. Autodan: Interpretable gradient-based adversarial attacks on large language models. *Preprint*, arXiv:2310.15140.

Yufan Zhuang, Chandan Singh, Liyuan Liu, Jingbo Shang, and Jianfeng Gao. 2025. Vector-icl: In-context learning with continuous vector representations. In *The Thirteenth International Conference on Learning Representations*.

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

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023b. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.

A Languages Used in Our Approach

We use the following languages in our experiments:

1. Natural Languages: English, French, German, Spanish, Persian, Arabic, Croatian, Japanese, Polish, Russian, Swedish, Thai, Hindi, Italian, Korean, Bengali, Portuguese, Chinese, Hebrew, Serbian, Danish, Turkish, Greek, Indonesian, Zulu, Hungarian, Basque, Swahili, Afrikaans, Bosnian, Lao, Romanian, Slovenian, Ukrainian, Finnish, Malay, Javanese, Welsh, Bulgarian, Armenian, Icelandic, Vietnamese, Sinhalese, Maori, Gujarati, Kannada, Marathi, Tamil, Telugu, Amharic, Norwegian, Czech, Dutch.

2. Cipher Languages: Caesar1, Caesar2, Caesar3, Caesar4, Caesar5, Caesar6, Caesar7, Caesarneg1, Caesarneg2, Caesarneg3, Caesarneg4, Caesarneg5, Caesarneg6, Caesarneg7, Ascii, Hexadecimal, Base64, Leet, Vowel, Alphanumeric. A number in front of Caesar cipher means that the English alphabets were shifted by that much forward and a number in front of Caesarneg cipher means that the English alphabets were shifted by that much backward.

Out of these languages, we use the following for training our classifier: Arabic, Chinese, Czech, Dutch, English, French, German, Hindi, Italian, Japanese, Korean, Polish, Portuguese, Russian, Spanish, Swedish, Thai, Bosnian, Turkish, Finnish, Indonesian, Bengali, Swahili, Vietnamese, Tamil, Telugu, Greek, Maori, Javanese, Caesar1, Caesar2, Caesar4, Caesarneg2, Caesarneg4, Caesarneg6, Ascii, Hexadecimal

And these for testing: Persian, Croatian, Hebrew, Serbian, Danish, Zulu, Hungarian, Basque, Afrikaans, Lao, Romanian, Slovenian, Ukrainian, Malay, Welsh, Bulgarian, Armenian, Icelandic, Sinhalese, Gujarati, Kannada, Marathi, Amharic, Norwegian, Caesar, Caesar5, Caesar7, Caesarneg3, Caesarneg1, Caesar6, Caesarneg7, Caesarneg5, Base64, Alphanumeric, Vowel, LeetSpeak.

B Datasets and models

C Experimental Details of Filtering of Wikitext and its Translation

D OmniGuard's Performance With Different Base LLMs

	High-Res	Low-Res	Cipher
LlamaGuard 1	69.92	41.25	16.07
LlamaGuard 2	75.28	62.2	16.2
LlamaGuard 3	82.23	75.84	24.74
AegisGuard (P)	83.36	59.06	44.22
AegisGuard (D)	88.14	76.26	**83.21**
WildGuard	81.35	43.51	16.53
HarmBench (llama)	14.25	14.08	14.11
HarmBench (mistral)	17.6	15.9	14.46
MD-Judge	59.51	30.21	15.44
DuoGuard	71.4	46.26	15.77
PolyGuard	**94.47**	79.22	21.28
OmniGuard	88.25	**85.56**	73.06

Table B1: Accuracy of detecting harmful prompts stratified by high-resource natural, low-resource natural, and cipher languages.

	Thai Tox	Ukr Tox	TJS	ToxText	Avg.
Layer 10	62.1	65.5	66.95	61.89	64.42
Layer 55	67.4	73.0	**76.91**	74.96	73.06
Layer 56	66.8	71.5	74.54	71.82	71.17
Selected layer 57	**68.7**	73.1	76.8	**75.07**	**73.40**
Layer 58	66.6	73.2	74.92	72.63	71.84
Layer 59	67.3	**73.3**	76.44	74.22	72.82
Layer 60	67.5	72.6	76.43	74.46	72.75
Layer 61	67.8	70.9	74.77	72.76	71.56
Layer 62	66.1	72.3	74.78	72.83	71.50
Layer 75	65.2	66.4	70.72	65.79	68.26
Last Layer	63.1	51.2	61.33	56.76	59.05

Table B2: OmniGuard's accuracy of detecting harmful prompts when trained using representations from different model layers.

Models	HarmBench	FQ	SimpleST	SaladB	TJS	ToxText	AdvBench	Avg.
Llama3.1-8B-Instruct	81.16	88.72	88.78	87.35	71.32	69.20	88.62	82.16
Gemma-3-4B-Instruct	78.03	88.23	**93.86**	82.09	53.50	51.45	89.07	76.60
Qwen-3-4B-Instruct	82.06	86.25	80.99	82.15	58.83	57.70	84.80	76.11
Olmo-2-7B-Instruct	80.57	88.86	92.14	**88.68**	70.65	64.79	88.24	81.99
Mistral-8B-Instruct	84.27	87.23	89.82	88.08	72.49	68.71	**89.07**	82.81
Llama3.3-70B-Instruct	**89.13**	**89.57**	89.62	87.30	**76.68**	**75.07**	86.59	**84.85**

Table B3: OmniGuard's accuracy of detecting harmful prompts when paired with different underlying LLMs across multiple benchmarks.