Multilingual Safety Signals Are Multi-Layered: Filtering Safety-Degrading Data for Safer LLMs

Anonymous ACL submission

Abstract

Preserving safety alignment during large language models fine-tuning is critical, however, recent studies have demonstrated that even benign fine-tuning data may contain safety-degrading samples that silently undermine safety alignment. Existing approaches typically identify such samples using representations from a single safety-sensitive layer. While this assumption has shown effectiveness in monolingual settings, its validity for multilingual models remains unclear due to potential cross-lingual differences in representation patterns. Through a cross-lingual analysis, we show that sensitive layers are only partially shared across languages, with safety-relevant signals often distributed across multiple layers. Motivated by these observations, we propose MMSAFE, a multi-layer framework for multilingual safety-degrading data identification that captures both shared and language-specific safety signals. Extensive experiments across multiple models, languages, and safety benchmarks demonstrate that MMSAFE reduces the average harmful-response ratio by 60% compared with random filtering and outperforms the strongest single-layer baseline, demonstrating the effectiveness of multi-layer modeling for robust multilingual safety alignment. Code is available at https://anonymous.4open.science/r/MMSAFE.

1 Introduction

Large language models (LLMs) are commonly aligned for safety before being adapted to downstream tasks through fine-tuning (Ouyang et al., 2022). While fine-tuning is essential for improving task performance and adapting models to downstream domains, recent studies show that its safety outcomes can be strongly influenced by the composition of the training data (Qi et al., 2024; He et al., 2024). In particular, seemingly benign fine-tuning corpora may contain safety-degrading samples that weaken built-in safeguards and increase the risk of unsafe behaviors after adaptation (Bach et al., 2026). Consequently, identifying and filtering such samples has become a critical step toward preserving safety alignment in practical LLM fine-tuning (Huang et al., 2024; Shen et al., 2025).

Recent studies have begun to identify safety-degrading samples through the internal representations of aligned LLMs (He et al., 2024; Li et al., 2025), suggesting that layer-level activations capture informative safety-relevant signals (Chen et al., 2026; Wang et al., 2026). A representative approach is LARF, which identifies a safety-sensitive layer and leverages its hidden representations to distinguish safety-degrading samples (Li et al., 2025). As LLMs are increasingly deployed across languages, it becomes important to understand whether such representation-based filtering remains reliable beyond monolingual settings. However, existing methods largely rely on representations from a single safety-sensitive layer, leaving it unclear whether such a single-layer view remains sufficient in multilingual settings, where safety-relevant signals may not be consistently localized across languages (Bu et al., 2026).

To better understand this limitation, we conduct a cross-lingual analysis of layer sensitivity. We find that some sensitive layers are consistently salient across languages, forming language-shared safety signals, while others are salient only for specific languages or language groups, reflecting language-specific safety signals. Moreover, multiple layers exhibit comparable sensitivity and capture semantically different safety-related features, suggesting that safety-relevant signals are not uniquely concentrated in a single layer. These observations indicate that a single-layer view may be insufficient for multilingual safety-degrading data identification.

Motivated by these observations, we propose MMSAFE (**M**ultilingual **M**ulti-layer **Safe**ty Filter), a multi-layer framework for multilingual safety-degrading data identification. Instead of rely-

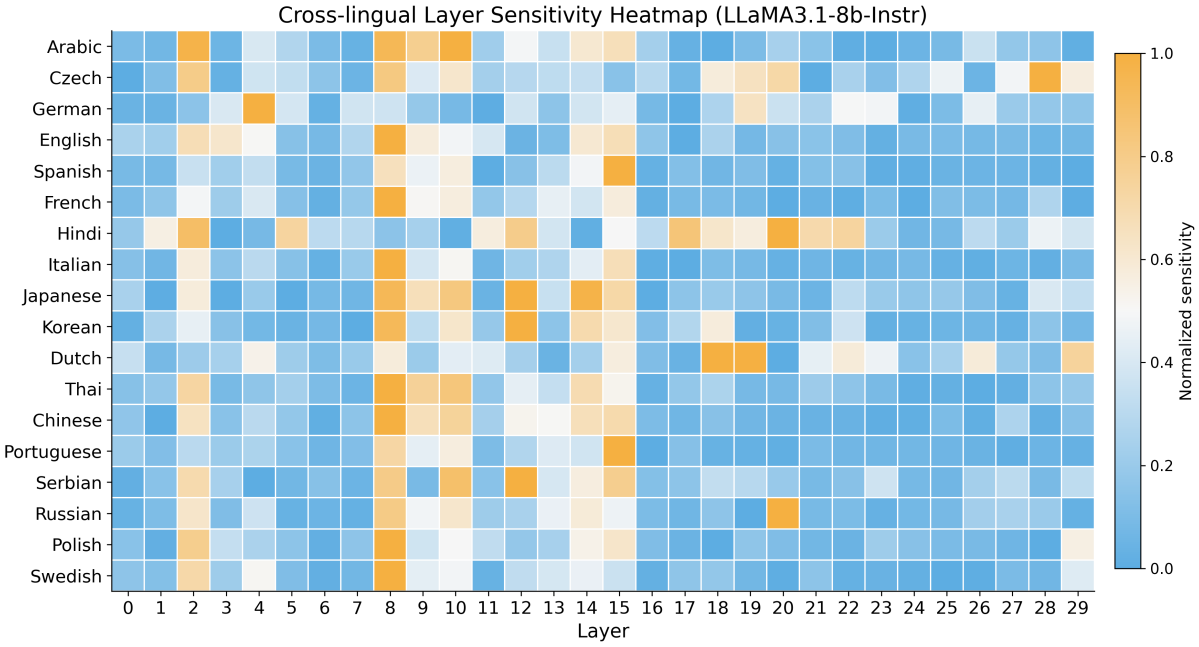

Figure 1: Cross-lingual layer sensitivity of LLaMA3.1-8B-Instruct across 18 languages. Each row is normalized within a language, and darker orange indicates higher sensitivity. The heatmap reveals a mixed pattern: some layers are consistently salient across languages, while others form language- or language-group-specific sensitivity peaks.

ing on a single safety-sensitive layer, MMSAFE jointly leverages shared layers that capture cross-lingually consistent safety signals and language-specific layers that reflect language-dependent variation. By jointly modeling these complementary signals, MMSAFE captures multilingual safety-relevant behavior more comprehensively and reduces the risk of missing safety-degrading samples that are only salient in specific languages or layers.

We conduct extensive experiments across four instruction-tuned LLMs, two multilingual fine-tuning data pools, and three multilingual safety benchmarks. The results show that MMSAFE consistently reduces harmful responses after fine-tuning. On average, MMSAFE reduces the harmful-response ratio by 60.7% compared with random filtering and outperforms the strongest single-layer baseline by 6.0 percentage points, with minimal degradation in model utility ($\leq 2\%$) on reasoning and coding benchmarks. These results demonstrate the effectiveness of multi-layer modeling for robust multilingual safety alignment.

Our main contributions are summarized as follows:

- We conduct a cross-lingual layer sensitivity analysis and show that sensitive layers are only partially shared across languages, while safety-relevant signals are distributed across multiple semantically complementary layers.
- We propose MMSAFE, a multilingual multi-layer safety filter for identifying safety-degrading samples. MMSAFE jointly leverages language-shared and language-specific sensitive layers, and uses safe-unsafe contrastive representations to score candidate fine-tuning samples.
- We evaluate MMSAFE across multiple models, fine-tuning datasets, languages, and safety benchmarks. Experimental results show that MMSAFE substantially reduces harmful responses compared with random filtering and consistently outperforms strong baselines.

2 Motivation

2.1 Questioning the Single-Layer Assumption

Existing methods rely on a single safety-sensitive layer for identifying safety-degrading samples. This design implicitly assumes that one layer can provide a sufficiently informative view for distinguishing safety-degrading samples.

In multilingual settings, this assumption becomes questionable for two reasons. First, layer sensitivity may vary across languages, such that a layer informative for one language may not generalize to another. Second, safety-relevant signals may be distributed across multiple layers, meaning that any single layer captures only a partial view. Such

omissions can be more pronounced in multilingual scenarios, where safety signals may vary across languages as well as across layers. These considerations motivate our analysis of how safety-relevant signals vary across both languages and layers.

2.2 Sensitive Layers Are Partially Shared Across Languages

We first examine whether layer sensitivity is language-dependent. Following LARF (Li et al., 2025), we compute layer-wise sensitivity by perturbing each layer and measuring changes in safety response counts on a multilingual harmful query set built from OR-Bench (Cui et al., 2025) and PolyGuardMix (Kumar et al., 2025), covering 18 languages and 36K queries in total. Figure 1 shows the normalized sensitivity heatmap for LLaMA3.1-8B-Instruct.

The heatmap reveals a clear partially shared structure: while Layer 8 is consistently salient across many languages, other peaks are language-specific, such as Layers 12–14 for Japanese and Korean and Layers 18–19 and 29 for Dutch. This indicates that layer sensitivity is only partially shared across languages: some layers are consistently salient, while others are language-specific, meaning that a useful layer for one language may not be equally useful for another. Similar patterns are observed on LLaMA3-8B-Instruct, Qwen3-8B, and Qwen3-32B, as shown in Appendix B. This partial sharing suggests that no single layer suffices, which raises a further question: do multiple sensitive layers capture redundant or complementary information?

2.3 Safety Signals Span Multiple Layers

We further examine whether safety-relevant signals are uniquely concentrated in the most sensitive layer. Figure 2 compares the relative sensitivity of the top-3 sensitive layers across models, with each model normalized by its most sensitive layer. The second- and third-ranked layers retain substantial relative sensitivity in all models, indicating that the top-ranked layer is often not uniquely dominant. This suggests that safety-relevant signals may span multiple sensitive layers rather than being concentrated in a single layer.

To further interpret what each layer captures, we inspect features extracted by sparse autoencoders (SAEs), which decompose hidden representations into sparse and interpretable semantic features. Specifically, we analyze SAE-interpreted features

Layer	Main focus	Representative concepts
3	Lexical / local semantic cues	Spoofing; lying; spying
7	Concrete harmful scenarios	Hoaxes; crimes; dangerous activities; bomb/gun making
11	Harmful intent and deception	Fraud; illicit requests; dangerous instructions; hiding evidence
15	Broad safety-policy violations	Illegal activities; abuse/hate; evasion; conspiracy theories

Table 1: Representative SAE-interpreted features activated by HarmBench harmful queries on LLaMA3.1-8B-Instruct.

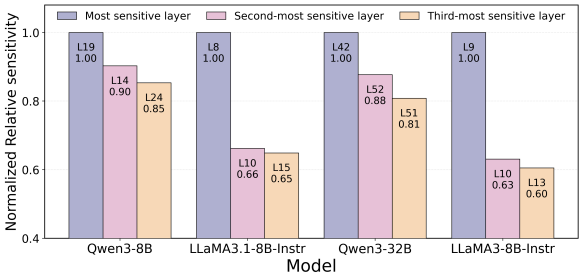

Figure 2: Relative sensitivity of the top-3 sensitive layers across models. Sensitivity is normalized by the most sensitive layer within each model.

(Arditi, 2025) activated by HarmBench (Mazeika et al., 2024) harmful queries on LLaMA3.1-8B-Instruct. As shown in Table 1, different layers correspond to different types of harmful concepts. Layer 3 mainly captures lexical or local semantic cues, while Layer 7 is more associated with concrete harmful scenarios. Layer 11 focuses more on harmful intent and deception, whereas Layer 15 captures broader safety-policy violations. These observations suggest that different layers provide semantically complementary safety information.

Together, the sensitivity comparison and SAE feature analysis indicate that safety-relevant signals are both distributed and complementary across layers. These findings jointly motivate a design that leverages both language-shared and language-specific sensitive layers, and exploits their complementary safety signals rather than relying on any single layer. Additional SAE-based analyses of how different layers respond to different harmful categories are provided in Appendix C.

3 Method

Motivated by our finding that safety-relevant signals are distributed across both shared and language-specific layers, we propose MMSAFE, a multilingual multi-layer safety filter for identifying safety-degrading samples in multilingual

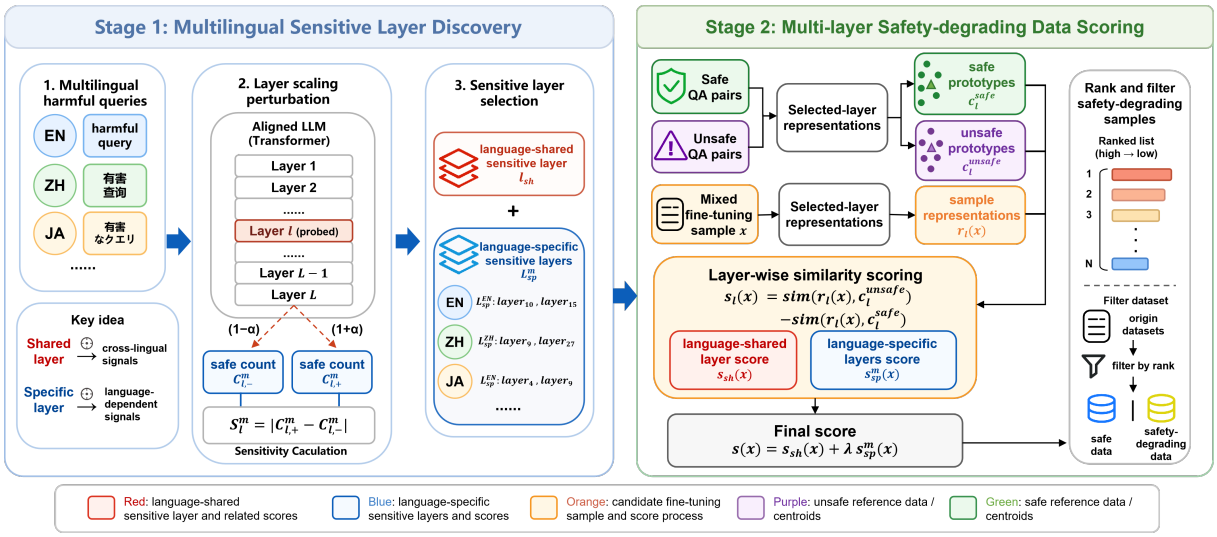

Figure 3: Overview of MMSAFE. The framework first discovers language-shared and language-specific sensitive layers through multilingual layer perturbation, and then uses safe-unsafe contrastive representations from these layers to score and filter safety-degrading samples.

fine-tuning data. MMSAFE consists of two stages. First, we discover sensitive layers by perturbing each layer and measuring changes in safety response counts on multilingual harmful queries, yielding both *language-shared sensitive layers* and *language-specific sensitive layers*. Second, we construct safe-unsafe contrastive representations from these layers and aggregate their signals to score and filter safety-degrading samples. Figure 3 provides an overview of this two-stage framework.

3.1 Multilingual Sensitive Layer Discovery

We first identify safety-sensitive layers in multilingual settings. Let $\mathcal{M}$ denote the set of languages and let the aligned LLM contain L transformer layers. For each language $m \in \mathcal{M}$, we use a set of harmful queries $\mathcal{D}^m_{\text{harm}}$ to evaluate how strongly each layer affects the model's safety behavior.

For each transformer layer $l \in \{0, \ldots, L-1\}$, we apply upward and downward scaling perturbations to the layer submodules. Let A_l and F_l denote the attention and feed-forward submodules of layer l, respectively. Given a perturbation strength $\alpha \in (0, 1)$, we define:

$$A_l^{\pm} = (1 \pm \alpha) A_l, \qquad F_l^{\pm} = (1 \pm \alpha) F_l. \tag{1}$$

When evaluating layer l, all other layers remain unchanged, and only A_l and F_l are replaced by their scaled versions. Larger behavioral shifts indicate stronger safety relevance of the perturbed layer.

For each scaling direction $\delta \in \{-, +\}$, let $y^m_{l,\delta}(x) = \text{LLM}(x; A_l^{\delta}, F_l^{\delta})$ denote the response generated by the scaled model for query x. We define the safety response count as:

$$C^m_{l,\delta} = \sum_{x \in \mathcal{D}^m_{\text{harm}}} \mathbb{I}_{\text{safe}}\left(y^m_{l,\delta}(x)\right), \tag{2}$$

where $\mathbb{I}_{\text{safe}}(\cdot)$ returns 1 if the response is classified as safe, e.g., refusing to provide harmful instructions, and 0 otherwise.

The layer sensitivity score S^m_l is defined as the absolute difference in safety response counts between the two scaling directions:

$$S^m_l = \left| C^m_{l,+} - C^m_{l,-} \right|. \tag{3}$$

After computing S^m_l, we rank layers for each language according to their sensitivity scores. Let $\mathcal{T}^m$ denote the ranked list of sensitive layers for language m. We identify the language-shared sensitive layer as the layer with the highest aggregated sensitivity across languages:

$$l_{\text{sh}} = \arg\max_{l} \sum_{m \in \mathcal{M}} S^m_l. \tag{4}$$

For each language m, we then select the top-k layers from $\mathcal{T}^m$ after excluding the shared layer:

$$\mathcal{L}^m_{\text{sp}} = \text{Top}_k\left(\mathcal{T}^m \setminus \{l_{\text{sh}}\}\right). \tag{5}$$

This design enables MMSAFE to capture both universally shared safety representations and language-dependent safety characteristics.

3.2 Multi-layer Safety-degrading Data Scoring

After identifying the shared and language-specific sensitive layers, MMSAFE uses them to score candidate fine-tuning samples. We first prepare a

small reference set of safe and unsafe question-answer pairs, denoted as $\mathcal{D}_{\text{safe}}$ and $\mathcal{D}_{\text{unsafe}}$, respectively. For a question-answer pair $x = (q, a)$, we feed the concatenated sequence into the aligned LLM and extract the hidden states at layer l. Let $\mathbf{H}_l(x) \in \mathbb{R}^{n \times d}$ denote the layer-l hidden states, where n is the sequence length and d is the hidden dimension. The sample representation is obtained via pooling over layer-l hidden states:

$$\mathbf{r}_l(x) = \text{Pool}\left(\mathbf{H}_l(x)\right). \tag{6}$$

In our experiments, $\text{Pool}(\cdot)$ denotes mean pooling over response tokens. For each selected layer l, we extract the hidden representation of each reference example and construct safe and unsafe layer prototypes:

$$\mathbf{c}_l^{\text{safe}} = \frac{1}{|\mathcal{D}_{\text{safe}}|} \sum_{x \in \mathcal{D}_{\text{safe}}} \mathbf{r}_l(x), \tag{7}$$

$$\mathbf{c}_l^{\text{unsafe}} = \frac{1}{|\mathcal{D}_{\text{unsafe}}|} \sum_{x \in \mathcal{D}_{\text{unsafe}}} \mathbf{r}_l(x), \tag{8}$$

where $\mathbf{r}_l(x)$ denotes the representation of sample x at layer l.

For a candidate sample x written in language m, we extract its representation $\mathbf{r}_l(x)$ from each selected layer. We measure the similarity between the candidate representation and the safe/unsafe prototypes:

$$s_l^{\text{unsafe}}(x) = \cos\big(\mathbf{r}_l(x), \mathbf{c}_l^{\text{unsafe}}\big). \tag{9}$$

$$s_l^{\text{safe}}(x) = \cos\big(\mathbf{r}_l(x), \mathbf{c}_l^{\text{safe}}\big). \tag{10}$$

The layer-level safety-degrading score is then defined as:

$$s_l(x) = s_l^{\text{unsafe}}(x) - s_l^{\text{safe}}(x). \tag{11}$$

A larger $s_l(x)$ indicates that the sample is closer to unsafe behavior than safe behavior at layer l.

We then compute two scores: one from the language-shared sensitive layer and the other from language-specific sensitive layers. The shared-layer score is:

$$s_{\text{sh}}(x) = s_{l_{\text{sh}}}(x). \tag{12}$$

The language-specific score is obtained by averaging over the selected language-specific layers:

$$s_{\text{sp}}^m(x) = \frac{1}{|\mathcal{L}_{\text{sp}}^m|} \sum_{l \in \mathcal{L}_{\text{sp}}^m} s_l(x). \tag{13}$$

Finally, we combine the two scores to obtain the overall safety-degrading score:

$$s(x) = s_{\text{sh}}(x) + \lambda s_{\text{sp}}^m(x), \tag{14}$$

where λ controls the contribution of language-specific sensitive layers. Samples exhibiting stronger unsafe affinity across sensitive layers are considered more likely to degrade safety alignment during fine-tuning. Therefore, MMSAFE ranks candidate fine-tuning samples by $s(x)$ and filters the highest-scoring samples as safety-degrading samples.

4 Experiments

4.1 Experimental Setup

Models. We evaluate MMSAFE on four large language models: LLaMA3-8B-Instruct (Team, 2024), LLaMA3.1-8B-Instruct (Team, 2024), Qwen3-8B (Team, 2025), and Qwen3-32B (Team, 2025). These models cover different model families and scales, allowing us to examine whether MMSAFE consistently identifies safety-degrading samples across diverse model architectures.

Datasets and Benchmarks. We use Aya Dataset (Singh et al., 2024) and the safe-labeled subset of Nemotron-Safety-Guard-v3 (Joshi et al., 2025) as candidate fine-tuning data pools, totaling over 400K samples and covering up to 65 languages. For safety evaluation, we use three multilingual safety benchmarks: LinguaSafe (Ning et al., 2025), XSafety (Wang et al., 2024), and MultiJail (Deng et al., 2024), covering over 70K evaluation instances across 16 languages. Additional datasets used for sensitive layer discovery and safe-unsafe centroid construction are described in Appendix A.

Baselines. We compare MMSAFE with four baselines. *Random* uniformly samples data from the candidate pool. *Bi-Anchoring* (He et al., 2024) estimates sample risk by measuring gradient similarity between candidate samples and reference instances. *SEAL* (Shen et al., 2025) trains a ranking model to distinguish safe and unsafe samples. *LARF* (Li et al., 2025) identifies safety-degrading samples using representations from a single safety-sensitive layer.

Implementation Details. We set the weighting coefficient in Equation 14 to $\lambda = 1$. For each filtering method, we fine-tune models on the bottom 20% lowest-risk samples selected by each method

Method	Aya Dataset						Nemotron-SG						Avg.
	LinguaSafe		XSafety		MultiJail		LinguaSafe		XSafety		MultiJail		
	#Harm↓	H/R↓	#Harm↓	H/R↓	#Harm↓	H/R↓	#Harm↓	H/R↓	#Harm↓	H/R↓	#Harm↓	H/R↓	H/R
LLaMA3-8B-Instruct													
Random	4,901	100.0%	1,810	100.0%	719	100.0%	2,148	100.0%	548	100.0%	794	100.0%	100.0%
SEAL	2,786	56.8%	1,192	65.9%	240	33.4%	4,134	192.5%	1,363	248.7%	324	40.8%	106.4%
Bi-Anchor	3,179	64.9%	1,378	76.1%	521	72.5%	3,185	148.3%	391	71.4%	439	55.3%	81.4%
LARF	3,306	67.5%	1,337	73.9%	463	64.4%	571	26.6%	36	6.6%	79	9.9%	<u>41.5%</u>
MMSAFE	3,348	68.3%	1,062	58.7%	471	65.5%	431	20.1%	19	3.5%	83	10.5%	**37.8%**
LLaMA3.1-8B-Instruct													
Random	3,484	100.0%	1,536	100.0%	394	100.0%	2,192	100.0%	875	100.0%	335	100.0%	100.0%
SEAL	4,227	121.3%	1,347	87.7%	288	73.1%	2,501	114.1%	1,206	137.8%	268	80.0%	102.3%
Bi-Anchor	2,103	60.4%	121	7.9%	191	48.5%	1,442	65.8%	445	50.9%	121	36.1%	44.9%
LARF	1,730	49.7%	935	60.9%	177	44.9%	293	13.4%	20	2.3%	60	17.9%	**31.5%**
MMSAFE	1,824	52.4%	937	61.0%	185	47.0%	155	7.1%	20	2.3%	70	20.9%	<u>31.8%</u>
Qwen3-8B													
Random	3,874	100.0%	1,581	100.0%	419	100.0%	3,273	100.0%	2,071	100.0%	308	100.0%	100.0%
SEAL	3,421	88.3%	1,232	77.9%	350	83.5%	3,218	98.3%	1,287	62.1%	321	104.2%	85.7%
Bi-Anchor	3,065	79.1%	1,146	72.5%	363	86.6%	2,094	64.0%	432	20.9%	222	72.1%	65.9%
LARF	3,196	82.5%	1,168	73.9%	357	85.2%	842	25.7%	237	11.4%	185	60.1%	<u>56.4%</u>
MMSAFE	3,250	83.9%	1,014	64.1%	312	74.5%	742	22.7%	76	3.7%	92	29.9%	**46.5%**
Qwen3-32B													
Random	3,930	100.0%	936	100.0%	279	100.0%	2,472	100.0%	778	100.0%	191	100.0%	100.0%
SEAL	3,132	79.7%	819	87.5%	212	76.0%	1,978	80.0%	642	82.5%	111	58.1%	77.3%
Bi-Anchor	3,641	92.6%	455	48.6%	181	64.9%	1,925	77.9%	754	96.9%	131	68.6%	74.9%
LARF	2,583	65.7%	519	55.4%	191	68.5%	798	32.3%	232	29.8%	114	59.7%	<u>51.9%</u>
MMSAFE	1,922	48.9%	493	52.7%	142	50.9%	636	25.7%	101	13.0%	106	55.5%	**41.1%**

Table 2: Evaluation results on Aya Dataset and Nemotron-SG. #Harm denotes the number of harmful or safety-violating responses. H/R denotes the harmful-response ratio relative to Random under the same model, dataset, and evaluator: $\mathrm{H/R} = \#\mathrm{Harm}_{\mathrm{method}}/\#\mathrm{Harm}_{\mathrm{Random}} \times 100\%$. Avg. reports the average H/R across the six evaluator–dataset combinations. Lower values are better for both metrics. The best Avg. result within each model group is shown in **bold**, and the second-best is <u>underlined</u>.

using LoRA. For all generation tasks, we use deterministic decoding with temperature 0 to ensure reproducibility. Additional implementation details are provided in Appendix A.

Evaluation. To evaluate model safety after fine-tuning, we employ LlamaGuard 3 (Team, 2024) as the safety evaluator. Given model responses to harmful queries, LlamaGuard 3 classifies whether each response is safe or unsafe.

4.2 Main Results

Table 2 shows the main safety evaluation results. We report both the number of harmful responses (#Harm) and the harmful-response ratio (H/R) relative to Random selection under the same setting. Lower values indicate fewer safety-violating responses after fine-tuning.

MMSAFE consistently achieves the strongest or near-strongest safety performance across model families. In particular, MMSAFE outperforms LARF, the single-layer baseline, on three out of four models, reducing the average H/R from 41.5% to 37.8%, from 56.4% to 46.5%, and from 51.9% to 41.1% on LLaMA3-8B-Instruct, Qwen3-8B, and Qwen3-32B, respectively. These improvements suggest that combining language-shared and language-specific sensitive layers provides a more effective signal for identifying safety-degrading samples than relying on a single layer.

Across fine-tuning datasets and evaluators, MMSAFE consistently reduces harmful responses compared with Random selection, indicating that it can filter samples that would otherwise induce safety degradation during fine-tuning. These results highlight the importance of modeling multilingual safety signals across multiple layers rather than relying on a single-layer view.

4.3 Utility Preservation

We further evaluate whether MMSAFE preserves general capabilities after safety-aware filtering. We report performance on MATH (Lightman et al., 2024) and code-generation benchmarks, using Live-

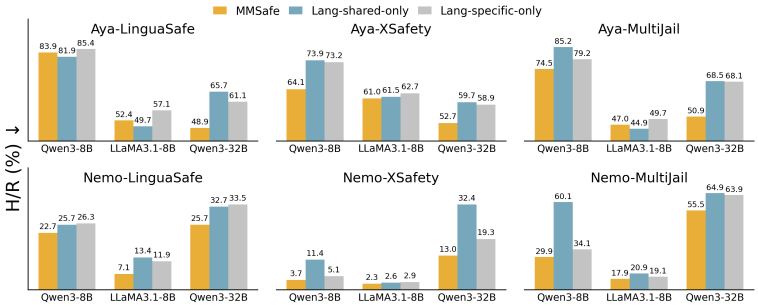

Figure 4: Ablation study on language-shared and language-specific sensitive layers. Lang-shared-only uses only the cross-lingually shared sensitive layer, Lang-specific-only uses only language-specific sensitive layers, and MMSAFE combines both. We report H/R relative to Random across three models and six evaluator–dataset settings. Lower values indicate fewer harmful responses.

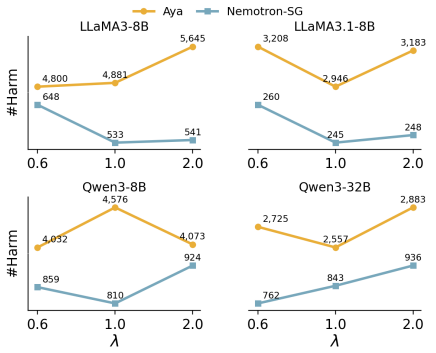

Figure 5: Sensitivity analysis of the weighting coefficient λ in MMSAFE. For each model, we report the summed number of harmful responses on Aya Dataset and Nemotron-SG across three safety evaluators.

CodeBench (Jain et al., 2025) for Qwen3 models and HumanEval (Chen et al., 2021) for LLaMA models. As shown in Table 3, the utility scores remain largely stable after fine-tuning on data selected by MMSAFE. Across both Aya and Nemotron-SG, changes on MATH are small for all models, and code-generation performance also remains close to the original models. For instance, Qwen3-32B changes from 60.7 to 59.9 and 60.1 on LiveCodeBench, while LLaMA3.1-8B-Instruct changes from 67.6 to 67.2 and 67.4 on HumanEval. These results suggest that MMSAFE reduces safety degradation while preserving general utility.

Model	FT Data	MATH	Code
Qwen3-8B	Base	90.4	52.5
	Aya Dataset	90.5	51.5
	Nemotron-SG	89.6	52.0
Qwen3-32B	Base	92.2	60.7
	Aya Dataset	91.7	59.9
	Nemotron-SG	91.1	60.1
LLaMA3-8B-Instruct	Base	27.1	55.4
	Aya Dataset	26.2	55.2
	Nemotron-SG	27.3	54.7
LLaMA3.1-8B-Instruct	Base	45.9	67.6
	Aya Dataset	46.2	67.2
	Nemotron-SG	45.2	67.4

Table 3: Utility evaluation after fine-tuning with data selected by MMSAFE. For code benchmark, Qwen models are evaluated on LiveCodeBench, while LLaMA models are evaluated on HumanEval.

5 Analysis

5.1 Ablation on Shared and Language-Specific Layers

Figure 4 compares MMSAFE with two ablated variants that use only the language-shared sensitive layer or only language-specific sensitive layers. The results show that neither component alone consistently dominates across models and datasets. The shared-only variant performs competitively in several LLaMA3.1-8B settings, indicating that shared layers encode transferable multilingual safety signals. However, it becomes less stable on Qwen models, especially under Nemotron-SG evaluations. In contrast, the language-specific-only variant improves some multilingual settings, but still underperforms the full method in several cases.

These results indicate that the two types of layers provide complementary information. Shared layers offer a stable cross-lingual anchor, while language-specific layers capture safety patterns that vary across languages. By combining both, MMSAFE achieves more robust safety filtering across heterogeneous multilingual settings.

5.2 Sensitivity to Language-Specific Weight

Figure 5 analyzes the effect of the weighting coefficient λ, which controls the contribution of language-specific layers. The optimal value varies across models and datasets. For example, on LLaMA3-8B, increasing λ reduces harmful responses on Nemotron-SG but increases them on Aya. Qwen3-32B shows a similar trade-off, with different datasets exhibiting different preferences for language-specific weighting. This suggests that over-emphasizing language-specific layers may benefit some data distributions but hurt others.

Overall, $\lambda = 1.0$ provides a stable middle ground. Although it is not always optimal for every

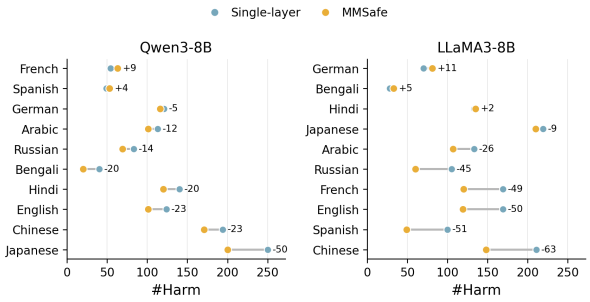

Figure 6: Language-wise unsafe counts on XSafety. Negative changes indicate fewer harmful responses with MMSAFE.

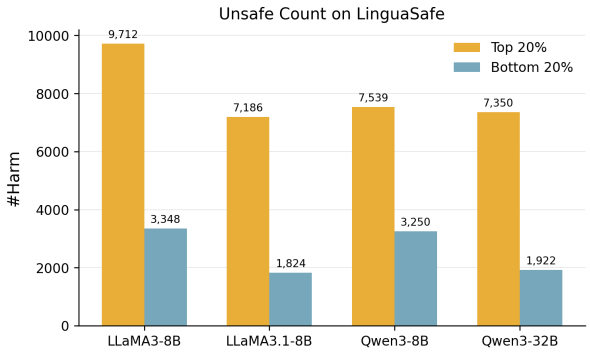

Figure 7: Unsafe counts on LinguaSafe after fine-tuning with top-20% vs. bottom-20% ranked Aya dataset.

individual model–dataset pair, it avoids extreme behavior and balances shared and language-specific signals. Therefore, we use $\lambda = 1$ as the default setting in the main experiments.

5.3 Language-wise Safety Analysis

Figure 6 shows language-wise unsafe counts on Aya-XSafety. Overall, MMSAFE reduces harmful responses in most languages for both Qwen3-8B and LLaMA3-8B, showing that the improvement is not driven by only a few high-resource languages. The reduction is especially clear for languages such as Japanese, Chinese, English, and Russian. Although a few languages show small regressions, the overall trend demonstrates MMSAFE's consistent robustness across diverse language groups.

5.4 Effect of Ranked Sample Selection

Figure 7 compares fine-tuning on the top-20% and bottom-20% ranked Aya samples. Across all four models, the top-ranked samples lead to substantially more unsafe responses on LinguaSafe than the bottom-ranked samples. This confirms that our scoring function assigns higher ranks to samples that are more likely to degrade safety after fine-tuning. The consistent gap between the two groups further validates that MMSAFE ranking scores meaningfully correlate with downstream safety degradation.

6 Related Work

6.1 Safety Degradation under Fine-tuning

Recent studies show that safety alignment in LLMs can be fragile during downstream fine-tuning. Even benign-looking data may weaken safety safeguards and increase harmful responses after adaptation (Qi et al., 2024; He et al., 2024; Halawi et al., 2024). This has motivated optimization-based defenses (Huang et al., 2024) and data-centric selection or filtering methods (Shen et al., 2025; Li et al., 2025). However, most prior methods focus on monolingual or language-agnostic settings, and representation-based filtering typically relies on a single safety-sensitive layer. Our work instead studies multilingual safety-degrading data identification and shows that safety-relevant signals are distributed across multiple layers.

6.2 Multilingual LLM Safety

Multilingual safety has recently received increasing attention, as LLMs may exhibit uneven safety behavior across languages. Prior work has revealed multilingual jailbreak vulnerabilities and stronger safety risks for non-English prompts (Deng et al., 2024; Wang et al., 2024). Recent benchmarks, datasets, and guard models further support multilingual safety evaluation and moderation (Deng et al., 2024; Wang et al., 2024; Kumar et al., 2025; Joshi et al., 2025; Mazeika et al., 2024; Ning et al., 2025). In contrast to prior work that mainly evaluates multilingual safety after model deployment, our work focuses on multilingual safety-degrading data identification before fine-tuning.

7 Conclusion

In this paper, we study safety-degrading data identification in multilingual fine-tuning. We show that safety-sensitive layers are only partially shared across languages, and that safety-relevant signals are distributed across both shared and language-specific layers. Motivated by these findings, we propose MMSAFE, a multilingual multi-layer safety filter that jointly leverages language-shared and language-specific sensitive layers to score candidate fine-tuning samples. Experiments across multiple models, data pools, and multilingual safety benchmarks show that MMSAFE consistently reduces harmful responses after fine-tuning while largely preserving general utility.

Limitations

MMSAFE focuses on identifying and filtering safety-degrading samples before fine-tuning. This pre-fine-tuning filtering strategy can reduce safety degradation during adaptation, but it is not intended to serve as a complete safety solution. A fine-tuned model may still exhibit unsafe behavior under unseen downstream tasks, deployment environments, or adversarial prompts. Therefore, MMSAFE should be used together with post-fine-tuning safety evaluation and deployment-time safeguards.

References

Andy Arditi. 2025. Finding misaligned persona features in open-weight models.

Thong Bach, Dung Nguyen, Thao Minh Le, and Truyen Tran. 2026. Continual safety alignment via gradient-based sample selection. *CoRR*, abs/2604.17215.

Yuyan Bu, Xiaohao Liu, ZhaoXing Ren, Yaodong Yang, and Juntao Dai. 2026. Align once, benefit multilingually: Enforcing multilingual consistency for LLM safety alignment. In *International Conference on Learning Representations*.

Jianhui Chen, Xiaozhi Wang, Zijun Yao, Yushi Bai, Lei Hou, and Juanzi Li. 2026. Towards understanding safety alignment: A mechanistic perspective from safety neurons. In *Advances in Neural Information Processing Systems 39 (NeurIPS 2026)*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Pondé de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. Evaluating large language models trained on code. *CoRR*, abs/2107.03374.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2025. Or-bench: An over-refusal benchmark for large language models. In *Forty-second International Conference on Machine Learning, ICML 2025, Vancouver, BC, Canada, July 13-19, 2025*, Proceedings of Machine Learning Research. PMLR / OpenReview.net.

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, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Danny Halawi, Alexander Wei, Eric Wallace, Tony Tong Wang, Nika Haghtalab, and Jacob Steinhardt. 2024. Covert malicious finetuning: Challenges in safeguarding LLM adaptation. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 17298–17312. PMLR / OpenReview.net.

Luxi He, Mengzhou Xia, and Peter Henderson. 2024. What is in your safe data? identifying benign data that breaks safety. In *First Conference on Language Modeling*.

Tiansheng Huang, Sihao Hu, Fatih Ilhan, Selim F. Tekin, and Ling Liu. 2024. Lisa: Lazy safety alignment for large language models against harmful fine-tuning attack. In *Advances in Neural Information Processing Systems 37: Annual Conference on Neural Information Processing Systems 2024, NeurIPS 2024, Vancouver, BC, Canada, December 10 - 15, 2024*.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. 2025. Livecodebench: Holistic and contamination free evaluation of large language models for code. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Raviraj Joshi, Rakesh Paul, Kanishk Singla, Anusha Kamath, Michael Evans, Katherine Luna, Shaona Ghosh, Utkarsh Vaidya, Eileen Peters Long, Sanjay Singh Chauhan, and Niranjan Wartikar. 2025. Cultureguard: Towards culturally-aware dataset and guard model for multilingual safety applications. In *Proceedings of the 14th International Joint Conference on Natural Language Processing and the 4th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, IJCNLP-AACL 2025, Mumbai, India, December 20-24, 2025*, pages 2666–2685. The Asian Federation of Natural Language Processing and The Association for Computational Linguistics.

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. In *Conference on Language Modeling*.

Hao Li, Lijun Li, Zhenghao Lu, Xianyi Wei, Rui Li, Jing Shao, and Lei Sha. 2025. Layer-aware representation filtering: Purifying finetuning data to preserve LLM safety alignment. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing, EMNLP 2025, Suzhou, China, November 4-9, 2025*, pages 8030–8050. Association for Computational Linguistics.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *ICLR 2024*. OpenReview.net.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David A. Forsyth, and Dan Hendrycks. 2024. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. In *Forty-first International Conference on Machine Learning, ICML 2024, Vienna, Austria, July 21-27, 2024*, Proceedings of Machine Learning Research, pages 35181–35224. PMLR / OpenReview.net.

Zhiyuan Ning, Tianle Gu, Jiaxin Song, Shixin Hong, Lingyu Li, Huacan Liu, Jie Li, Yixu Wang, Lingyu Meng, Yan Teng, and Yingchun Wang. 2025. Linguasafe: A comprehensive multilingual safety benchmark for large language models. *CoRR*, abs/2508.12733.

Long Ouyang, Jeffrey 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 F. Christiano, Jan Leike, and Ryan Lowe. 2022. Training language models to follow instructions with human feedback. In *Advances in Neural Information Processing Systems 35: Annual Conference on Neural Information Processing Systems 2022, NeurIPS 2022, New Orleans, LA, USA, November 28 - December 9, 2022*.

Xiangyu Qi, Yi Zeng, Tinghao Xie, Pin-Yu Chen, Ruoxi Jia, Prateek Mittal, and Peter Henderson. 2024. Fine-tuning aligned language models compromises safety, even when users do not intend to! In *The Twelfth International Conference on Learning Representations, ICLR 2024, Vienna, Austria, May 7-11, 2024*. OpenReview.net.

Han Shen, Pin-Yu Chen, Payel Das, and Tianyi Chen. 2025. SEAL: safety-enhanced aligned LLM fine-tuning via bilevel data selection. In *The Thirteenth International Conference on Learning Representations, ICLR 2025, Singapore, April 24-28, 2025*. OpenReview.net.

Shivalika Singh, Freddie Vargus, Daniel D'souza, Börje Karlsson, Abinaya Mahendiran, Wei-Yin Ko, Herumb Shandilya, Jay Patel, Deividas Mataciunas, Laura O'Mahony, Mike Zhang, Ramith Hettiarachchi, Joseph Wilson, Marina Machado, Luisa Souza Moura, Dominik Krzeminski, Hakimeh Fadaei, Irem Ergün, Ifeoma Okoh, and 14 others. 2024. Aya dataset: An open-access collection for multilingual instruction tuning. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers), ACL 2024, Bangkok, Thailand, August 11-16, 2024*, pages 11521–11567. Association for Computational Linguistics.

Llama Team. 2024. The llama 3 herd of models. *CoRR*, abs/2407.21783.

Qwen Team. 2025. Qwen3 technical report. *CoRR*, abs/2505.09388.

Wenxuan Wang, Zhaopeng Tu, Chang Chen, Youliang Yuan, Jen-tse Huang, Wenxiang Jiao, and Michael Lyu. 2024. All languages matter: On the multilingual safety of LLMs. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5865–5877, Bangkok, Thailand. Association for Computational Linguistics.

Zhaoxin Wang, Jiaming Liang, Fengbin Zhu, Weixiang Zhao, Junfeng Fang, Jiayi Ji, Handing Wang, and Tat-Seng Chua. 2026. Safeneuron: Neuron-level safety alignment for large language models. *CoRR*, abs/2602.12158.

A Experiment Setting

A.1 Dataset Usage

We use multiple datasets for different purposes in our experiments, including candidate fine-tuning data pools, multilingual safety evaluation benchmarks, sensitive layer discovery data, and reference data for safe-unsafe centroid construction.

Candidate fine-tuning data pools. We use Aya Dataset and the safe-labeled subset of Nemotron-Safety-Guard-v3 as candidate fine-tuning data pools for safety-degrading sample identification. Aya Dataset is a large-scale multilingual instruction-following dataset containing 204K samples across 65 languages. We use it as a broad multilingual fine-tuning corpus, where safety-degrading samples may be mixed with otherwise benign instruction-following data. We further use the safe-labeled subset of Nemotron-Safety-Guard-v3, which contains over 200K safe-labeled samples across 12 languages. Although these samples are labeled as safe, prior work has shown that seemingly benign fine-tuning data can still contain examples that degrade safety alignment, making this dataset a challenging testbed for safety-degrading sample identification.

Safety evaluation benchmarks. We evaluate safety preservation after fine-tuning on three multilingual safety benchmarks: LinguaSafe, XSafety, and MultiJail. LinguaSafe contains 45K entries across 12 languages and is designed to evaluate multilingual safety behavior. XSafety covers 14 commonly used safety issue categories across 10 languages, allowing us to evaluate safety under diverse harmful request types. MultiJail evaluates multilingual jailbreak robustness across 10 languages. Together, these benchmarks cover a broad range of languages, safety categories, and jailbreak scenarios.

Category	Dataset	Languages	Size / Coverage
Candidate fine-tuning data	Aya Dataset	65	204K samples
Candidate fine-tuning data	Nemotron-SG	12	>200K safe-labeled samples
Sensitive layer discovery	OR-Bench	18†	1K hard queries translated into multiple languages
Sensitive layer discovery	PolyGuardMix	17	1.91M samples
Safe-unsafe centroid construction	PolyGuardMix	17	1.91M samples
Safe-unsafe centroid construction	Nemotron-SG Val.	12	∼10K validation samples
Safety evaluation	LinguaSafe	12	45K entries
Safety evaluation	XSafety	10	28K samples covering 14 safety categories
Safety evaluation	MultiJail	10	Multilingual jailbreak prompts

Table 4: Summary of datasets and benchmarks used in our experiments. Nemotron-SG denotes Nemotron-Safety-Guard-v3. † indicates languages obtained through translation.

Model	Shared layer	Language-specific layers
Qwen3-8B	19	Arabic: 27, 17; Czech: 28, 14; German: 14, 24; English: 14, 13; Spanish: 14, 17; French: 26, 14; Hindi: 17, 24; Italian: 13, 20; Japanese: 24, 21; Korean: 29, 27; Dutch: 13, 24; Thai: 22, 5; Chinese: 15, 22; Portuguese: 29, 17; Russian: 28, 29; Polish: 24, 17; Swedish: 24, 29.
LLaMA3-8B-Instruct	9	Arabic: 10, 0; Czech: 13, 2; German: 29, 23; English: 10, 14; Spanish: 13, 11; French: 28, 11; Hindi: 0, 2; Italian: 2, 3; Japanese: 10, 0; Korean: 10, 0; Dutch: 13, 10; Thai: 13, 10; Chinese: 10, 0; Portuguese: 13, 11; Russian: 10, 13; Polish: 2, 10; Swedish: 2, 1.
LLaMA3.1-8B-Instruct	8	Arabic: 10, 2; Czech: 28, 2; German: 4, 19; English: 2, 15; Spanish: 15, 10; French: 15, 10; Hindi: 20, 2; Italian: 15, 2; Japanese: 12, 14; Korean: 12, 14; Dutch: 18, 19; Thai: 10, 9; Chinese: 10, 15; Portuguese: 15, 10; Russian: 20, 2; Polish: 2, 15; Swedish: 2, 4.
Qwen3-32B	42	Arabic: 52, 51; Czech: 51, 55; German: 44, 37; English: 44, 40; Spanish: 52, 56; French: 44, 58; Hindi: 57, 54; Italian: 51, 52; Japanese: 20, 40; Korean: 59, 50; Dutch: 61, 45; Thai: 52, 51; Chinese: 44, 46; Portuguese: 52, 59; Serbian: 1, 56; Russian: 54, 41; Polish: 54, 56; Swedish: 56, 54.

Table 5: Selected language-shared and language-specific sensitive layers used by MMSAFE. For each model, the shared layer is selected by aggregating sensitivity across languages, and the language-specific layers are selected from each language's top-ranked layers after excluding the shared layer.

Sensitive layer discovery data. For sensitive layer discovery, we construct a multilingual harmful query set from OR-Bench and PolyGuardMix. Specifically, we use 2K harmful queries for each of 18 languages, resulting in 36K queries in total. These queries are used to perturb each layer and compute language-wise sensitivity scores for selecting language-shared and language-specific sensitive layers.

Reference safe and unsafe pairs. To construct the safe and unsafe centroids used by MMSAFE, we sample reference question-answer pairs from PolyGuardMix and the validation split of Nemotron-Safety-Guard-v3. For each of the 18 languages, we sample 800 safe and 800 unsafe pairs. The safe pairs are used to estimate layer-wise safe centroids, while the unsafe pairs are used to estimate layer-wise unsafe centroids for safe-unsafe contrastive scoring.

A.2 Implementation Details

Sensitive layer discovery. For sensitive layer discovery, we follow the perturbation-based sensitivity measurement of LARF. For each language, we use 2K harmful queries, covering 18 languages and 36K queries in total. We perturb each transformer layer with scaling factors 0.8 and 1.2, corresponding to a perturbation strength of $\alpha = 0.2$. For each perturbed model, we generate responses to the harmful queries and compute the safety response count difference between the two scaling directions as the layer sensitivity score.

To reduce computational cost, we set `max_new_tokens` to 32 during sensitive layer discovery. Following prior work, we observe that the model's refusal intent can usually be inferred from the initial response tokens. We therefore detect refusals by matching predefined refusal-related patterns, such as "I cannot", "I am

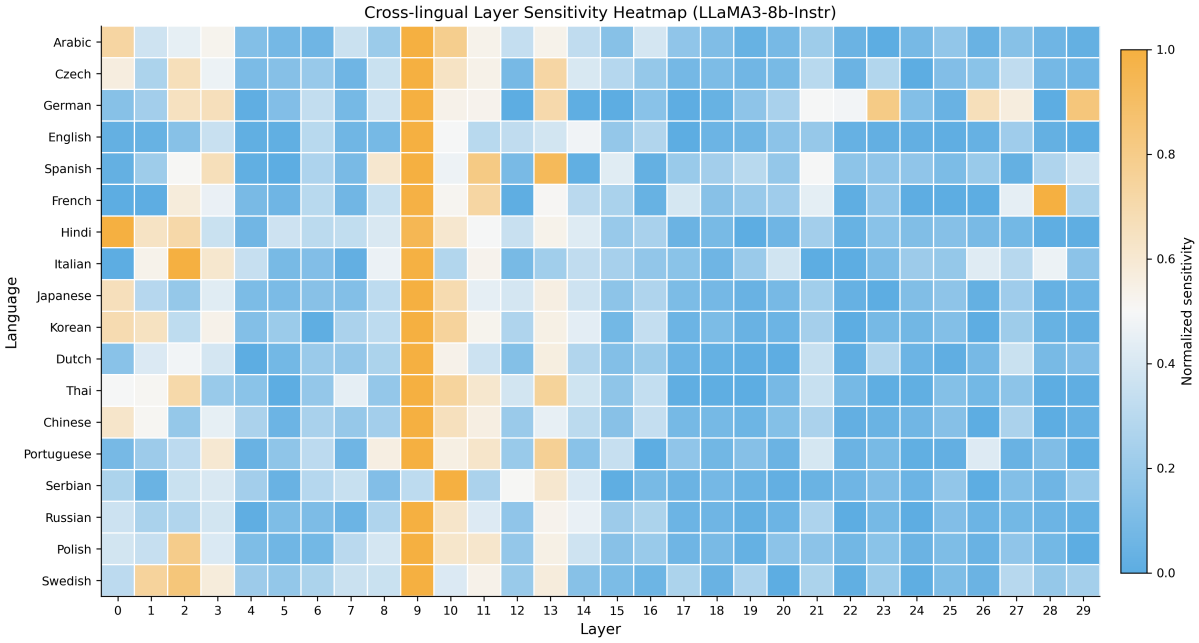

Figure 8: Cross-lingual layer sensitivity of LLaMA3-8B-Instruct across 18 languages. Each row is normalized within a language, and darker orange indicates higher sensitivity.

sorry", and "Sorry", in the generated responses. For non-English responses, we include language-specific refusal patterns translated and manually checked for each evaluated language.

Safe-unsafe centroid construction. For constructing the safe and unsafe centroids used by MMSAFE, we use reference question-answer pairs from PolyGuardMix and the validation split of Nemotron-Safety-Guard-v3. For each of the 18 languages, we sample 800 safe and 800 unsafe question-answer pairs. Given the selected sensitive layers, we extract layer-wise representations for these reference pairs and compute the mean representations as safe and unsafe centroids. These centroids are then used for safe-unsafe contrastive scoring with cosine similarity.

Layer selection and scoring. For each model, we select one language-shared sensitive layer by aggregating sensitivity scores across languages. For each language, we select the top-k language-specific sensitive layers after excluding the language-shared layer, and set $k = 2$ in all experiments. We set the weighting coefficient in Equation 14 to $\lambda = 1$. The final safety-degrading score is computed by combining the shared-layer score and the language-specific score.

Filtering and fine-tuning. For each candidate fine-tuning data pool and each filtering method, we rank candidate samples according to their safety-degrading scores and select the bottom 20% samples for fine-tuning. We use the same selection ratio for all methods to ensure a fair comparison. We adopt LoRA fine-tuning for all experiments, with `num_train_epochs` set to 3, learning rate set to 1×10^{-4}, and warmup ratio set to 0.1. Using 20% of the original data provides a controlled filtering setting: it retains sufficient fine-tuning signal while requiring each method to prioritize samples that are most likely to affect safety alignment.

Generation and evaluation. For all generation tasks, we set `do_sample` to `False` and temperature to 0 to ensure reproducibility. For safety evaluation with LlamaGuard 3, we also use deterministic decoding with temperature 0. Unless otherwise specified, all reported safety results are computed using the same generation and evaluation settings across models, datasets, and filtering methods.

All experiments were conducted on NVIDIA A100 GPUs. Sensitive-layer discovery, LoRA fine-tuning, and evaluation required approximately 100 GPU hours in total.

B Sensitive Layers Further Analysis

B.1 Multilingual Harmful Query Set

To analyze layer sensitivity across languages, we construct a multilingual harmful query set from two sources: OR-Bench and PolyGuardMix. From OR-

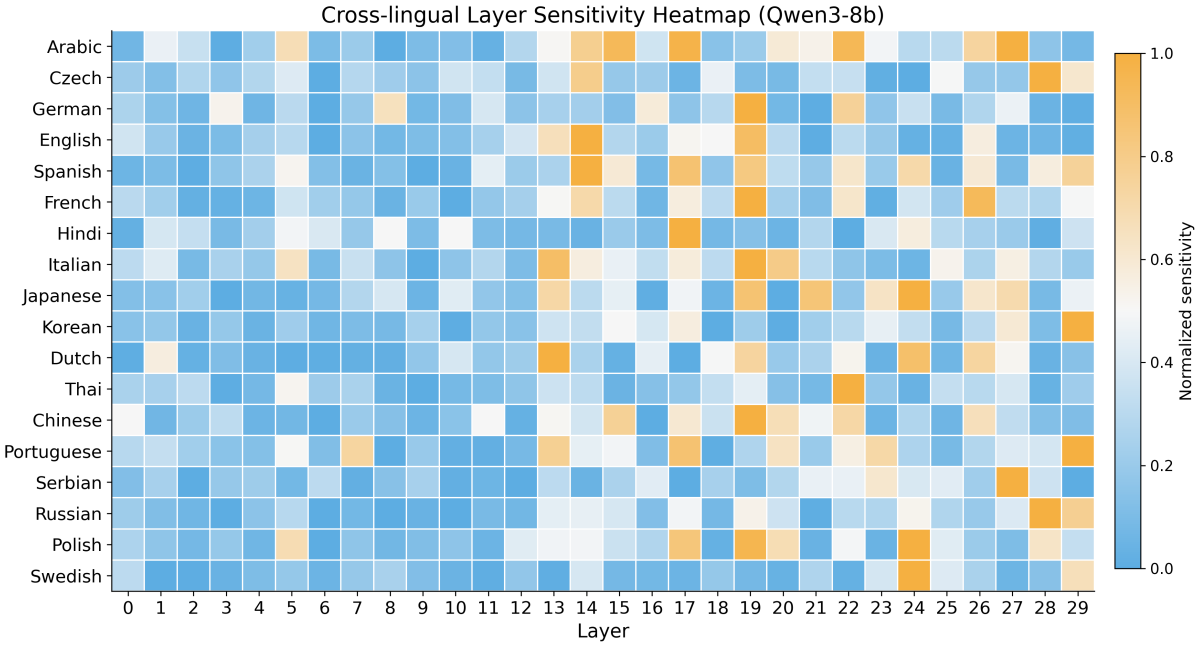

Figure 9: Cross-lingual layer sensitivity of Qwen3-8B across 18 languages. Each row is normalized within a language, and darker orange indicates higher sensitivity.

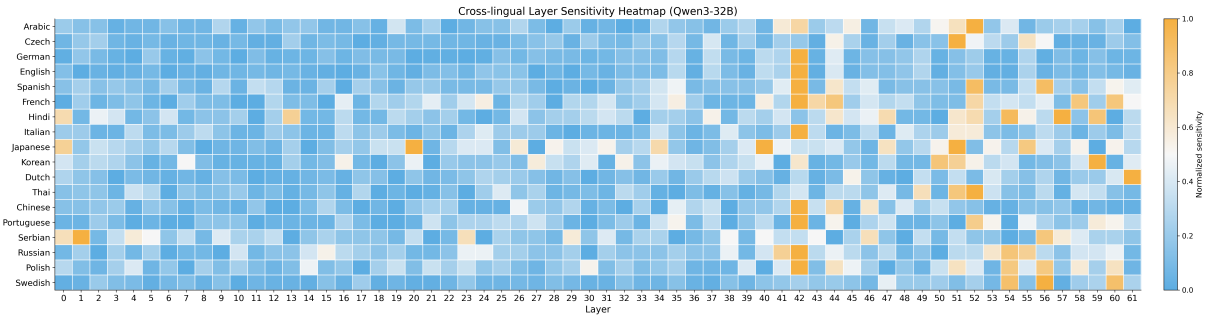

Figure 10: Cross-lingual layer sensitivity of Qwen3-32B across 18 languages. Each row is normalized within a language, and darker orange indicates higher sensitivity.

Bench, we select 1,000 hard safety-related queries and translate them into multiple languages using Gemini. To complement these translated queries with naturally multilingual harmful data, we further incorporate samples from PolyGuardMix.

For each language, the resulting evaluation set contains 2,000 harmful queries, consisting of translated OR-Bench queries and native multilingual queries from PolyGuardMix. In total, our sensitivity analysis covers 18 languages and 36K harmful queries. The translated OR-Bench queries provide controlled cross-lingual counterparts, while PolyGuardMix contributes naturally occurring multilingual harmful queries. This construction allows us to evaluate both cross-lingual consistency and language-specific variation in layer sensitivity.

B.2 Sensitivity Patterns Across Additional Models

We further examine whether the partially shared sensitivity pattern observed in the main text generalizes to other models. Figures 8, 9, and 10 show the normalized cross-lingual layer sensitivity heatmaps for LLaMA3-8B-Instruct, Qwen3-8B, and Qwen3-32B, respectively.

Overall, all three models exhibit a mixture of cross-lingually salient and language-specific sensitive layers. For LLaMA3-8B-Instruct, the shared pattern is particularly strong: a small number of layers, especially around Layer 9, show consistently high sensitivity across many languages. This suggests that LLaMA3-8B-Instruct contains strongly shared safety-sensitive layers that are activated across languages.

In contrast, the two Qwen3 models show more dispersed sensitivity patterns. Although some layers are repeatedly salient across languages, many high-sensitivity regions are more language-dependent. For example, Qwen3-8B exhibits sensitivity peaks across a wider range of middle and later layers, while Qwen3-32B shows more scattered language-specific peaks across its deeper layer stack. These results suggest that language-specific sensitivity is more pronounced in the Qwen3 models.

Together, these additional results support the main observation that sensitive layers are only partially shared across languages. While some layers are consistently salient, others exhibit language- or language-group-specific sensitivity, indicating that layer sensitivity depends on both model architecture and language.

B.3 Selected Sensitive Layers

Table 5 reports the language-shared and language-specific sensitive layers selected for each model. The shared layer is selected by aggregating layer sensitivity across languages, while the language-specific layers are selected from each language's ranked sensitive layers after excluding the shared layer. These layers are used in the scoring stage of MMSAFE.

C Safety Signals Analysis

C.1 Category-specific SAE Activations Across Layers

To further examine whether different layers encode complementary safety information, we analyze SAE-interpreted feature activations on HarmBench harmful queries. Specifically, we consider four harmful categories, including malware, drug-related content, phishing, and disinformation, and measure the activation strength of harmful SAE features at Layers 3, 7, 11, and 15 of LLaMA3.1-8B-Instruct.

Figure 11 shows that different layers exhibit distinct activation patterns across harmful categories. Layer 3 shows relatively weak and uniform activations, suggesting that it mainly captures lower-level lexical or local semantic cues. In contrast, Layer 7 is strongly activated by drug-related and phishing samples, while Layer 11 shows the strongest response to phishing and a higher response to disinformation. Layer 15 remains highly activated across multiple categories, especially malware, drug-related content, and phishing.

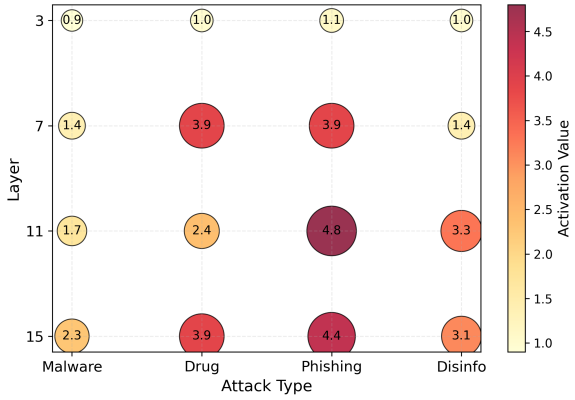

Figure 11: Category-specific SAE activations across layers on HarmBench harmful queries using LLaMA3.1-8B-Instruct. Bubble size and color indicate activation strength. Different layers exhibit distinct activation patterns across harmful categories, suggesting that safety-relevant signals are semantically complementary across layers.

These results indicate that safety-relevant representations are not simply repeated across layers. Instead, different layers emphasize different harmful concepts and categories, providing complementary safety information. Together with the relative sensitivity analysis in the main text, this supports our claim that safety signals span multiple layers and are not fully captured by a single safety-sensitive layer.