Can LLM Safety Be Ensured by Constraining Parameter Regions?

Zongmin Li[1,2,3] **Jian Su**[6*] **Farah Benamara**[4,5] **Aixin Sun**[1†]
[1]Nanyang Technological University, Singapore
[2]Institute for Infocomm Research, A*STAR, Singapore
[3]CNRS@CREATE LTD, Singapore
[4]IRIT, Université de Toulouse, CNRS, Toulouse INP, Toulouse, France
[5]IPAL, CNRS-NUS-A*STAR, Singapore [6]Independent
zongmin001@e.ntu.edu.sg, sujian123@gmail.com
farah.benamara@irit.fr, axsun@ntu.edu.sg

Abstract

Large language models (LLMs) are often assumed to contain "safety regions" – parameter subsets whose modification directly influences safety behaviors. We conduct a systematic evaluation of four safety region identification methods spanning different parameter granularities, from individual weights to entire Transformer layers, across four families of backbone LLMs with varying sizes. Using ten safety identification datasets, we find that the identified safety regions exhibit only low to moderate overlap, as measured by IoU. The overlap drops significantly when the safety regions are further refined using utility datasets (*i.e.,* non-harmful queries). These results suggest that current techniques fail to reliably identify a stable, dataset-agnostic safety region.[1]

1 Introduction

Large language models (LLMs) have achieved remarkable progress across a wide range of NLP tasks (Wei et al., 2022; Touvron et al., 2023a; OpenAI et al., 2024; Team et al., 2025). However, their deployment has raised growing concerns about safety, including the risks of generating harmful, biased or misleading content. Despite extensive safety alignment efforts, recent work has demonstrated that LLMs remain vulnerable to various attacks. Specifically, jailbreaking can circumvent safety mechanisms through adversarial prompting (Yi et al., 2024; Wei et al., 2023; Zou et al., 2023; Ding et al., 2024; Deng et al., 2024; Huang et al., 2023; Kang et al., 2023), and fine-tuning can systematically erode safety behaviors even when using benign-appearing datasets (Qi et al., 2024; Yang et al., 2023; Lermen et al., 2024; Zhan et al., 2024).

To remedy, recent research has begun to explore whether certain parameter regions within LLMs are systematically associated with safer or more harmful behaviors. Understanding these 'safety regions' could enable targeted protection strategies, such as freezing critical parameters during downstream fine-tuning to maintain model safety while preserving task adaptability.

Currently, there is no consensus on what constitutes a safety region in LLMs or at what level of granularity it should be defined. For instance, Wei et al. (2024) define a safety region as a collection of parameters. A parameter is the smallest unit of a weight matrix within each Transformer layer, corresponding to an individual scalar value learned by the model. In contrast, Zhao et al. (2025) identify safety regions at the level of neurons, referring to rows or columns of the weight matrix. Li et al. (2025) instead define safety regions as specific consecutive intermediate layers of an LLM, while Yi et al. (2025) focus on safety-critical parameters within LoRA adaptation matrices rather than the base model itself.

The proliferation of differing definitions suggests that the field lacks a shared understanding or a guideline of what properties a safety region should possess to be considered reliable and actionable. Inspired by the work on knowledge editing (Zhang et al., 2024; He et al., 2025), we argue that a meaningful safety region should satisfy at least three criteria:

- **Reliability:** Constraints or modifications applied to safety regions shall be reflected in corresponding changes in the model's behavior when responding to safety-related questions (*e.g.,* the model becomes safer or more harmful). Such constraints or modifications may take the form of ablating, scaling (up or down), or freezing during harmful fine-tuning.
- **Locality:** Constraints (*e.g.,* ablating) applied

[*]Jian Su was with the Institute for Infocomm Research (I2R), A*STAR, Singapore at the time of this work (e-mail: sujian@a-star.edu.sg).
[†]Corresponding author
[1]The code and datasets are available online.

to safety regions shall not affect the model's capability in other areas, such as non-safety-related functions.

- **Convergent Identifiability:** A safety region should be an intrinsic property of the model's parameter space and should remain invariant across the safety datasets used to identify it, which we refer to as *identification datasets*. In other words, even when the identification datasets represent distinct categories of harmful content, they should yield convergence toward an ideally identical underlying region within the parameter space.

In the current literature, most of the work has verified *Reliability* and *Locality* to some extent. However, the *convergent identifiability* property remains unverified. Some existing methods (Zhao et al., 2025; Li et al., 2025; Yi et al., 2025) use a single dataset to identify safety regions. Other work (Wei et al., 2024) sample datasets multiple times to identify safety regions, but does not check their consistency.

In this paper, through carefully designed experiments, we evaluate the *convergent identifiability* of four recent methods that claim to identify safety regions: SNIP & Wanda (Wei et al., 2024), SafeNeuron (Zhao et al., 2025), SafeLayer (Li et al., 2025), and NLSR (Yi et al., 2025). These four methods collectively cover a good range of safety region granularities (parameter-, neuron-, layer-, and LoRA-weight-level) as illustrated in Fig. 1. We aim to address two research questions.

- **RQ1:** As the number of identification datasets increases, does the overlap among the identified safety regions converge to a stable region in the parameter space?

- **RQ2:** To what extent does the distribution of the identification dataset (*e.g.,* semantic similarity) influence the identified safety regions?

Specifically, following each method, we construct multiple safety datasets, each used to identify a safety region within a target LLM. We then evaluate the extent to which these identified regions overlap, quantified by the Intersection over Union (IoU) metric.

In our experiments, we were unable to reproduce the reported results for safety regions defined at the layer level. For the remaining three methods, the safety regions identified across multiple

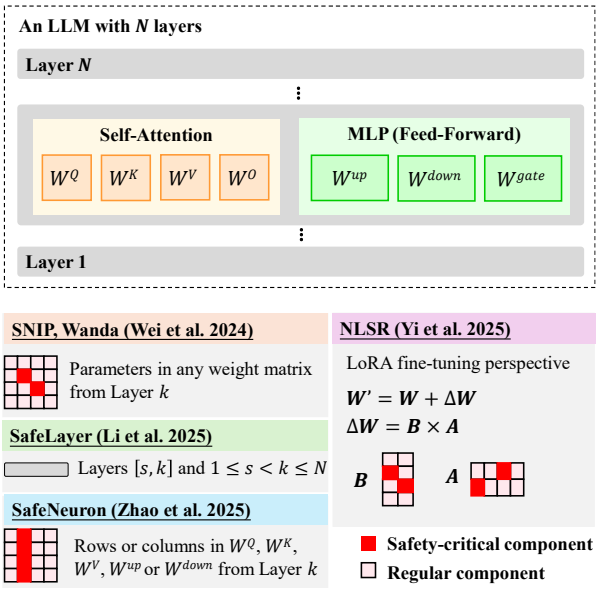


Figure 1: Overview of current safety region identification methods

datasets, even when sampled from the same larger dataset, exhibit lower-than-expected IoU. The IoU decreases even further after refinement using utility datasets, reflecting the persistent entanglement of safety and utility within the model's parameter space. These empirical findings suggest that a well-defined safety region may not exist, or that current methods are insufficient to reliably identify one.

Our studies carry significant implications for the broader study of LLM safety. A central motivation behind locating safety regions is to constrain them during subsequent adaptation—such as fine-tuning—in order to prevent the degradation of safety behaviors. Yet our analysis demonstrates that safety regions identified across different datasets are highly fragmented, with overlaps failing to converge. This lack of stability may pose fundamental challenges for applying reliable constraints in practice and therefore warrants further study. We therefore suggest that our results serve to recalibrate prevailing assumptions about the existence of safety regions, and highlight the need for future research to design more principled and robust identification methods capable of supporting durable safety alignment.

2 Preliminaries

We now briefly review the four methods examined in this study to provide background for our empirical analysis. We encourage readers to refer to the original papers for full technical details.

2.1 SNIP & Wanda, SafeNeuron and NLSR

SNIP & Wanda, SafeNeuron, and NLSR identify safety regions at the levels of parameters, neurons, and LoRA weights, respectively. All three methods share a common principle: assigning importance scores to safety-critical components to quantify their contribution to the model's safety behavior. Moreover, the same scoring mechanism can be applied to identify a utility-related parameter region (or utility region) using a utility dataset. As illustrated in Fig. 1, the red-highlighted components correspond to safety-critical elements, while the non-highlighted ones represent regular components. Since each method ranks model components (parameters, neurons, or LoRA weights) by their importance scores, a threshold is defined to select those components to form a safety region. Below, we briefly introduce the scoring mechanism used by each method.

SNIP & Wanda (Wei et al., 2024). A **parameter** refers to an individual scalar value within a weight matrix of each Transformer layer (Vaswani et al., 2017) in the target LLM. Both SNIP (Lee et al., 2019) and Wanda (Sun et al., 2024) assign importance scores to individual parameters based on their contribution to the model's safety behavior, specifically, its ability to generate refusal responses to harmful queries. For each parameter, SNIP measures importance according to the change in loss, whereas Wanda measures it according to the change in output. Within each weight matrix of the target LLM, the safety region is defined as the set of parameters with the top $q\%$ SNIP or Wanda scores.

SafeNeuron (Zhao et al., 2025). Here, **neuron** refers to a row or a column of parameters within $W^Q, W^K, W^V, W^{up}, W^{down}$ at each Transformer layer of the targeted LLM.

Given a harmful query as input, the importance score of each neuron is defined by the difference in internal representation when that neuron is removed. The activated neurons in each weight matrix are those with the top m importance scores. Finally, the safety region in each weight matrix is defined as the set of neurons that are consistently activated across harmful queries in the identification dataset.

NLSR (Yi et al., 2025). Safety regions in this method are identified within LoRA weight update matrices A and B. Given a base LLM, they keep the weights frozen and update only the LoRA weights to obtain a super safety-aligned model, via weak-to-strong extrapolation (Zheng et al., 2025). Next, they applied rank reduction to LoRA weights to preserve the safety-critical weights. Finally, for each lower-rank approximation of A and B, the safety region is defined by the composition of parameters with top $t\%$ magnitude values.

2.2 SafeLayer (Li et al., 2025)

SafeLayer identifies a small set of consecutive layers in the middle of the model as safety regions. A **layer** is a Transformer layer, consisting of a self-attention module, a feed-forward network (aka MLP), and their associated normalization and residual connections.

Instead of directly scoring individual layers as safety components, as in the three aforementioned methods (Section 2.1), SafeLayer adopts a different strategy. First, a layer-wise cosine similarity analysis of the model's internal output representations to harmful and utility inputs is conducted, revealing that certain middle layers function as a safety differentiation zone. These layers are where LLMs begin to distinguish between harmful and utility queries, which provides evidence for the existence of safety layers. Second, building on the initial range of layers, the authors hypothesize that scaling the safety layers will affect the model's responses to over-rejection queries. They further leverage the over-rejection phenomenon to precisely locate these safety layers.

3 Methodology

We systematically evaluate the four safety region identification methods by deriving safety regions from multiple identification datasets and measuring their cross-dataset consistency. Building on the two **RQ**s stated in Section 1, we evaluate the *convergent identifiability* via cross-dataset safety region overlap under varying dataset number and diversity.

To answer **RQ1**, we design the following variants of measuring the *convergent identifiability*. Section 3.1 explains the overlap among safety regions. Next, considering the intricate relationship between safety and utility, we measure the overlap among utility-isolated safety regions (Section 3.2). We use the classical metric, intersection over union (Section 3.3), to measure the overlap.

To answer **RQ2**, we design two identification dataset variants: (i) datasets spanning diverse harm

categories, and (ii) datasets where each one corresponds to a single distinct harm category. For both variants, we follow Sections 3.1 and 3.2 to evaluate the *convergent identifiability*.

3.1 Safety Region Overlap

Suppose there are n datasets of safety related samples $\mathcal{D}_0, \mathcal{D}_1, \ldots, \mathcal{D}_{n-1}$. In principle, we can apply a safety identification method on each dataset to identify a corresponding safety region, $\mathcal{R}_0, \mathcal{R}_1, \ldots, \mathcal{R}_{n-1}$. If the property of *convergent identifiability* holds, these n safety regions should exhibit substantial overlap. Therefore, for each safety region identification method, we measure the degree of overlap among the identified safety regions $\{\mathcal{R}_i\}_{i=0}^{n-1}$.

3.2 Utility-Isolated Safety Region Overlap

As Wei et al. (2024) pointed out, to understand a harmful query, it requires both safety awareness and utility capability of the model. Also, Zhao et al. (2024) identified that some utility-related neurons are responsible for the fundamental management of queries. During downstream task fine-tuning, modifications to these neurons can inadvertently affect the safety neurons that overlap, which degrades the safety mechanism. Within safety regions, to examine the *convergent identifiability* of the components that contribute solely to safety, we propose measuring the utility-isolated safety region overlap.

Specifically, for each safety region identification method, we follow Section 3.1 to obtain n safety regions $\{\mathcal{R}_i\}_{i=0}^{n-1}$. Then, we take a utility dataset $\mathcal{D}_u$ to obtain the utility region $\mathcal{R}_u$ with the same identification method. For each safety region $\mathcal{R}_i$, the utility-isolated safety region will be $\mathcal{R}_i - \mathcal{R}_u$, *i.e.,* the set difference. Finally, we measure the overlap across these n utility-isolated safety regions $\{\mathcal{R}_i - \mathcal{R}_u\}_{i=0}^{n-1}$.

3.3 Evaluation Metric

To measure the overlap across the n safety regions $\{\mathcal{R}_i\}_{i=0}^{n-1}$, we use **IoU** (intersection over union) as the evaluation metric. Specifically, $\forall i \in [0, n-1]$, $\mathcal{R}_i$ is a safety region composed of certain components (safety-critical parameters, neurons, layers or LoRA weights). Let $\mathcal{S}_i$ be the set of indices for each safety-critical component of $\mathcal{R}_i$, then the **IoU** to measure safety region overlap is

$$\mathrm{IoU}(\mathcal{S}_0, \mathcal{S}_1, \ldots, \mathcal{S}_{n-1}) = \frac{\bigcap_{i=0}^{n-1} \mathcal{S}_i}{\bigcup_{i=0}^{n-1} \mathcal{S}_i} \quad (1)$$

The same definition applies to the utility-isolated safety regions, and we refer to the resulting overlap metric as **Iso-Utility IoU**.

4 Experimental Setup

We evaluate the four safety region identification methods introduced in Section 2, chosen for their diverse definitions and publicly available implementations. All experiments follow the original implementations and hyperparameters.

4.1 Identification Datasets

We construct two sets of identification datasets, to be detailed next. For each method, we then apply these datasets to the same target LLMs used in the original paper. We further include additional model families where applicable, most notably the Qwen series (Qwen et al., 2025), to align model coverage across all methods (Table 1).

Multi-Category Dataset We construct $n = 10$ identification datasets, each covering diverse harm categories, following the original implementation. For each method, the dataset used in the original paper contains harmful queries across various categories, denoted as $\mathcal{D}_0$. The sources of $\mathcal{D}_0$ for each method are detailed in Appendix G. The sampled datasets $\mathcal{D}_i (i = 1, \ldots, 9)$ follow the same data format and size as $\mathcal{D}_0$ ($|\mathcal{D}_i| = |\mathcal{D}_0|$). Table 1 reports $|\mathcal{D}_0|$ for each method. We sample $\{\mathcal{D}_i\}_{i=1}^{9}$ as follows:

- SNIP & Wanda: $\mathcal{D}_0$[2] is the *safety-short* variant, where each harmful query is paired with a judgement-segment, refusal-only answer. $\{\mathcal{D}_i\}_{i=1}^{9}$ are sampled from the training split of PKU-SafeRLHF-QA (Ji et al., 2024b,a)[3], restricted to the highest severity level of responses. Each $\mathcal{D}_i$ contains harmful queries spanning multiple categories.

- SafeNeuron: $\{\mathcal{D}_i\}_{i=1}^{9}$ are sampled from the same PKU-SafeRLHF-QA training split, restricted to the highest severity level of responses. Each $\mathcal{D}_i$ contains harmful queries of diverse categories.

[2] In Wei et al. (2024), $\mathcal{D}_0$ is sampled from a source dataset at each time. In our evaluation, we prepare 5 different seeds to sample $\mathcal{D}_0$ and report the average IoU (Table 6-9).

[3] The dataset was created mainly for the purpose of reinforcement learning with human feedback. In our setting, we simply need questions and answers in the context of safety. We therefore use only the questions and corresponding answers.

- NLSR: $\{\mathcal{D}_i\}_{i=1}^{9}$ are sampled from the training split of PKU-SafeRLHF-30K (Ji et al., 2024b,a). Each $\mathcal{D}_i$ contains harmful queries of diverse categories, paired with refusal answers.

- SafeLayer: Details of the identification datasets are provided in Appendix A.

To evaluate utility-isolated safety region overlap (Section 3.2), we take the utility dataset $\mathcal{D}_u$ from Wei et al. (2024). This dataset is based on Alpaca-Cleaned[4], a refined version of the Alpaca dataset (Taori et al., 2023), where safety-related samples were filtered out using sensitive phrase matching (Qi et al., 2024). For each method, $\mathcal{D}_u$ preserves the same data format as $\mathcal{D}_0$, with $|\mathcal{D}_u| = |\mathcal{D}_0|$.

Single-Category Dataset We construct n_c identification datasets for SNIP & Wanda and SafeNeuron[5]. For each method, each $\mathcal{D}_{c_i}$ preserves the same data format as $\mathcal{D}_0$, with $|\mathcal{D}_{c_i}| = |\mathcal{D}_0|$.

- SNIP & Wanda: we sample $n_c = 5$ datasets from PKU-SafeRLHF-QA, restricted to the highest severity level of responses. Each $\mathcal{D}_{c_i}$ contains 128 data samples from a single harm category c_i.

- SafeNeuron: we sample $n_c = 12$ datasets from PKU-SafeRLHF-QA. As the highest severity level of responses alone does not provide enough data samples, we extend our sampling to include both the highest and second highest severity levels, which results in $n_c = 12$ datasets. Each $\mathcal{D}_{c_i}$ contains 200 data samples from a single harm category c_i.

The specific harm categories $\{c_i\}_{i=1}^{n_c}$ for each method are presented in Appendix B.

4.2 Thresholds

For SNIP & Wanda, SafeNeuron and NLSR, certain thresholds are necessary to select the top important safety-critical components. In our evaluation, we included the same threshold settings as in the original work. For NLSR, $t\% = 20\%$; for SafeNeuron[6], m is 1000 for W^Q, W^K, W^V and 2000 for W^{up} and W^{down}. We apply the same thresholds to identify the utility region $\mathcal{R}_u$. For SNIP & Wanda, we follow the thresholds in the original paper and add additional choices: $q\%$ for identifying the safety region and $p\%$ for the utility region. All choices are shown in Table 6-9.

[4]https://github.com/gururise/AlpacaDataCleaned

[5]We exclude NLSR, as we could not find sufficient data samples from available resources to construct $\mathcal{D}_{c_i}$.

[6]Value taken from the provided implementation: https://github.com/zhaoyiran924/Safety-Neuron

Table 1: Targeted LLMs and dataset size for each method. SafeLayer involves two stages of identification, with 100 samples per dataset in Stage I and 731 samples per dataset in Stage II.

Method	Targeted LLMs	# Samples
SNIP & Wanda (Wei et al., 2024)	Llama-2-7B-Chat Llama-2-13B-Chat Mistral-7B-Instruct-v0.2 Qwen2.5-7B-Instruct	128
SafeNeuron (Zhao et al., 2025)	Llama-2-7B-Chat Llama-3-8B-Instruct Mistral-7B-Instruct-v0.2 Qwen2.5-7B-Instruct	200
SafeLayer (Li et al., 2025)	Llama-2-7B-Chat Llama-3-8B-Instruct gemma-2B-IT Phi-3-mini-4k-instruct	100 731
NLSR (Yi et al., 2025)	Llama-3-8B Mistral-7B-v0.2 Qwen2.5-7B	2000

5 Experimental Results

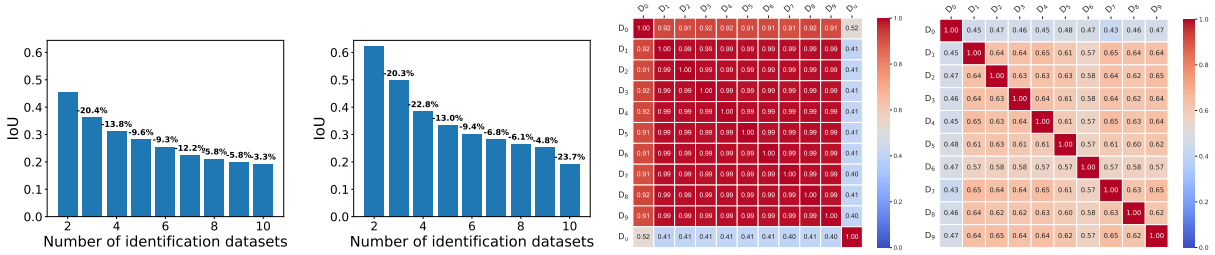
5.1 Findings on Multi-Category Datasets

Table 2 reports the main results on the overlap of safety regions and utility-isolated safety regions across 10 multi-category identification datasets. Ideally, if a method fully satisfies *convergent identifiability*, the IoU of safety regions identified from any n different datasets should be exactly 1. However, our results exhibit only low to moderate overlap for most settings, with IoU values ranging from 0.28 to 0.72 across the ten identified safety regions. In an extreme case, NLSR yields an IoU of 0.01 on Mistral-7B-v0.2, indicating strong dataset dependence from a LoRA fine-tuning perspective. Overall, these results suggest that existing methods fail to identify dataset-independent safety regions. Instead, they appear to be finding dataset-specific patterns rather than universal safety mechanisms.

When we remove the utility region from each safety region, the IoU decreases substantially for all methods. For example, in SNIP on Llama-2-7B-Chat, disentangling safety regions defined by the top 1% safety-critical parameters with the utility region defined by the top 2% utility-critical parameters reduces IoU by 38% (Tables 6 and 7) — from 0.29 to 0.18. More generally, this dramatic drop

Table 2: *Convergent identifiability* of existing safety region identification methods with 10 public multi-category datasets: safety region overlap (*IoU*) and utility-isolated safety region overlap (*Iso-Utility IoU*).

Method	Safety Region	Targeted Model	IoU	Iso-Utility IoU
SNIP Score (Wei et al., 2024)	Parameter	Llama-2-7B-Chat	0.29 - 0.42	0.17 - 0.24
		Llama-2-13B-Chat	0.28 - 0.42	0.15 - 0.24
		Mistral-7B-Instruct-v0.2	0.30 - 0.42	0.20 - 0.26
		Qwen2.5-7B-Instruct	0.35 - 0.46	0.21 - 0.26
Wanda Score (Wei et al., 2024)	Parameter	Llama-2-7B-Chat	0.57 - 0.72	0.27 - 0.35
		Llama-2-13B-Chat	0.55 - 0.70	0.25 - 0.33
		Mistral-7B-Instruct-v0.2	0.56 - 0.69	0.28 - 0.37
		Qwen2.5-7B-Instruct	0.57 - 0.69	0.26 - 0.37
SafeNeuron (Zhao et al., 2025)	Neuron	Llama-2-7B-Chat	0.62	0.16
		Llama-3-8B-Instruct	0.58	0.19
		Mistral-7B-Instruct-v0.2	0.57	0.17
		Qwen2.5-7B-Instruct	0.48	0.14
SafeLayer (Li et al., 2025)	Layer	Llama-2-7B-Chat Llama-3-8B-Instruct gemma-2B-IT Phi-3-mini-4k-instruct	Unable to identify	Unable to identify
NLSR (Yi et al., 2025)	LoRA Weights	Llama-3-8B	0.39	0.08
		Mistral-7B-v0.2	0.01	0.001
		Qwen2.5-7B	0.45	0.09



(a) Iso-Utility IoU vs n, forward order (b) Iso-Utility IoU vs n, backward order (c) Pairwise cosine similarity analysis for $\{\mathcal{D}_i\}_{i=0}^{9}$ and $\mathcal{D}_u$ (d) Pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$

Figure 2: Utility-isolated safety region overlap analysis using SafeNeuron on Llama-3-8B-Instruct. (a) We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; (b) We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; (c) The matrix is symmetric, and each element represents the semantic cosine similarity between the centroid embeddings of two multi-category identification datasets. (d) The matrix is symmetric. Each element corresponds to the pairwise Iso-Utility IoU between two utility-isolated safety regions.

indicates that safety regions have a non-negligible overlap with utility regions in these models. The resulting utility-isolated safety regions, which contribute only to model safety, exhibit poor *convergent identifiability*. These findings imply that safety regions, defined and identified in their current form, are unlikely to be an intrinsic parameter region of a model's parameter space, but highly dependent on the choice of identification datasets.

Furthermore, we investigate how rapidly the IoU decreases as n increases. As shown in Fig. 2, for SafeNeuron, gradually adding new datasets in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$ appears to make the overlap converge: adding $\mathcal{D}_9$ only reduces the Iso-Utility IoU by 3.3%. In contrast, when the order is reversed and $\mathcal{D}_0$ is added last, the Iso-Utility

IoU drops by 23.7%, indicating that the overlap does not converge. Recall that $\mathcal{D}_0$ is the original dataset, whereas $\{\mathcal{D}_i\}_{i=1}^{9}$ are sampled from the same source dataset. The semantic cosine similarity analysis (implementation details in Appendix E) shown in Fig. 2c reveals that $\mathcal{D}_1$, ... $\mathcal{D}_9$ have nearly identical semantic distribution, with cosine similarity close to 1. By contrast, $\mathcal{D}_0$ exhibits a slightly different semantic distribution. This discrepancy becomes more evident in the pairwise overlap analysis (Fig. 2d), where the pairwise Iso-Utility IoU between $\mathcal{D}_0$ and each $\mathcal{D}_i$ is at least 0.10 lower than that of any other pair. These results indicate that the utility-isolated safety regions become more distinct as the semantic distributions of the identification datasets diverge.

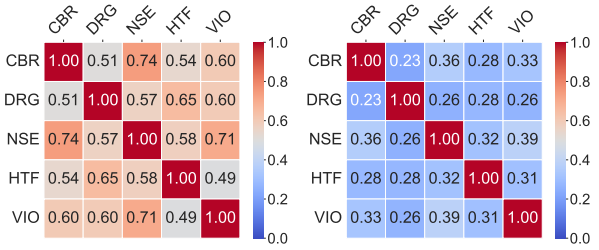
The detailed per-case results of SNIP & Wanda summarized in Table 2 are provided in Appendix C. Additional analyses on IoU decay with n and pairwise overlap—both for the overlap of safety regions and utility-isolated safety regions—are presented in Appendix D. For SafeLayer, we follow the same methodology, using the provided implementation[7], to evaluate the *convergent identifiability* of Stage I. The analysis across 10 datasets supports the existence of consistent safety layers. However, for Stage II, we were unable to (1) reproduce the over-rejection evaluation, for which we implemented a GPT-4-based judge, and (2) identify unique safety-related layers across three different identification datasets. Full details, including *convergent identifiability* evaluation and our replication of the over-rejection evaluation, are reported in Appendix A.

5.2 Impact of Dataset Semantics

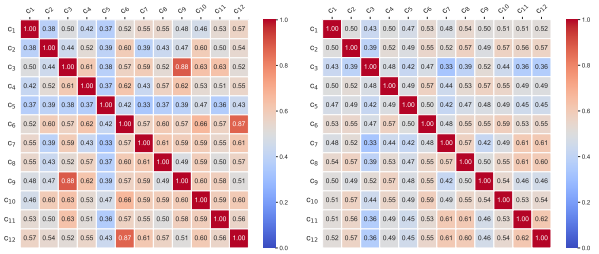
Pairwise analysis in the previous section suggests that safety region overlap may depend on dataset distributions. We probe this dependence directly by re-sampling identification datasets so that each one contains examples from a single harmful category (Section 4.1). This yields n_c datasets $\{\mathcal{D}_{c_i}\}_{i=1}^{n_c}$. The specific harm categories and n_c value for each method are listed in Appendix B.

Fig. 3 reports (left) semantic cosine similarity between dataset centroids, and (right) the corresponding utility-isolated safety region overlap for SNIP and SafeNeuron. Relative to the multi-category setting (*e.g.,* Fig. 2), both the cosine similarity analysis and pairwise Iso-Utility IoU become more heterogeneous. We quantify the relationship using the Pearson correlation r between cosine similarity and Iso-Utility IoU. For SNIP & Wanda, $r \in [0.51, 0.77]$, indicating a moderately strong correlation. By contrast, SafeNeuron exhibits weak correlations ($r = 0.12$–0.28), suggesting that its identified regions are less directly tied to the semantic similarity of the input queries. Overall, SNIP & Wanda appear more sensitive to dataset semantics, whereas SafeNeuron may rely more on model-internal patterns than on input-level semantics. Additional analyses of pairwise overlaps with single-category datasets and corresponding Pearson r values are provided in Appendix F.

From Fig. 2a and Fig. 2b, the order in which datasets are added also affects the resulting overlap, suggesting a potential role of dataset diversity. To further investigate the impact of dataset provenance, we construct identification datasets from multiple sources. For SNIP, $\mathcal{D}_0$ is drawn from AdvBench (Zou et al., 2023) and $\mathcal{D}_1$ from PKU-SafeRLHF-QA (Ji et al., 2024b,a). We additionally sample three datasets from HEx-PHI (Qi et al., 2024), SORRY-Bench (Xie et al., 2025), and StrongREJECT (Souly et al., 2024), resulting in a total of five datasets from distinct sources. On Llama-2-



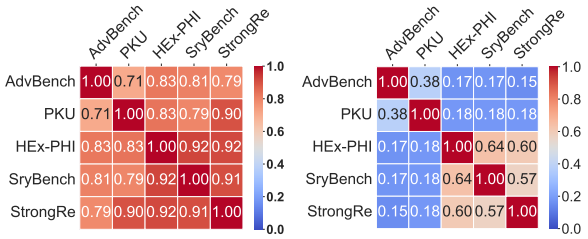
(a) SNIP · Llama-2-13B-Chat ($q = 3\%$, $p = 8\%$) — Cosine sim. (left) vs Iso-Utility IoU (right). Pearson correlation is 0.68, with p-value 0.03. CBR: cybercrime; DRG: drugs; NSE: endangering national security; HTF: human trafficking; VIO: violence.



(b) SafeNeuron · Llama-3-8B-Instruct — Cosine sim. (left) vs Iso-Utility IoU (right). Pearson correlation is 0.28, with p-value 0.02. The meaning of each harm category c_i is described in Table 5.

Figure 3: Semantic similarity vs. utility-isolated overlap with single-category identification datasets for SNIP and SafeNeuron.

[7] https://github.com/listen0425/Safety-Layers



(a) Cosine similarity (b) Iso-Utility IoU

Figure 4: Semantic similarity vs. utility-isolated overlap with 5 identification datasets from different sources for SNIP on Llama-2-7B-Chat ($q = 1\%$, $p = 1\%$). Pearson correlation is 0.56, with p-value 0.09. PKU: PKU-SafeRLHF-QA; SryBench: SORRY-Bench; StrongRe: StrongREJECT.

7B-Chat, with $q = p = 1\%$, the Iso-Utility IoU is 0.07, substantially lower than the values reported in Table 2. This suggests that the variation across data sources introduces a stronger distribution shift than stochastic sampling within the same source. Further analysis in Fig. 4 shows that semantic similarity and pairwise utility-isolated overlap exhibit a moderate correlation.

6 Related Work

6.1 Safety Alignment and Its Vulnerability

The objective of safety alignment is to make unaligned LLMs produce helpful, harmless content and refuse harmful queries. This is typically achieved through supervised fine-tuning (Ouyang et al., 2022) and reinforcement learning with human feedback (Bai et al., 2022; Dai et al., 2024). The transition in model weights from an unaligned to an aligned state can be understood as the integration of safety behaviors into the model parameters. However, this alignment is fragile and can be circumvented by adversarially crafted prompts, such as jailbreaking (Yi et al., 2024; Wei et al., 2023; Zou et al., 2023; Ding et al., 2024; Deng et al., 2024; Huang et al., 2023). Furthermore, recent studies show that fine tuning on harmful or even benign downstream task data can compromise the safety guardrail (Qi et al., 2024; Yang et al., 2023; Lermen et al., 2024; Zhan et al., 2024; Betley et al., 2025; Hawkins et al., 2024; He et al., 2024).

To address these challenges, recent research has proposed defenses against jailbreaking attacks (Yi et al., 2024; Hu et al., 2024; Alon and Kamfonas, 2023; Robey et al., 2024; Kumar et al., 2025; Zhou et al., 2024; Xie et al., 2024; Zeng et al., 2024) and safety degradation during downstream fine tuning (Qi et al., 2025; Wang et al., 2024; Hsu et al., 2025; Huang et al., 2024). Beyond defensive strategies, a growing body of work (Wei et al., 2024; Li et al., 2025; Zhao et al., 2025; Yi et al., 2025; Gao et al., 2025; Zhou et al., 2025; Arditi et al., 2024a; Peng et al., 2024; Poppi et al., 2025) seeks to uncover the internal safety mechanisms of LLMs. As safety alignment encodes safety behaviors into the model parameters, some of these studies (Wei et al., 2024; Li et al., 2025; Zhao et al., 2025; Yi et al., 2025; Gao et al., 2025; Zhou et al., 2025) examine whether safety behaviors are systematically localized in specific parameter subsets, often referred to as **safety regions**.

6.2 Constraining Parameter Regions

Safety regions are typically assumed to underlie the model's safety behaviors, implying that constraints applied to them directly influence how the model responds to safety-related queries. Examples of such constraints include ablating (removing or deactivating parameters within the region), freezing gradients during downstream fine-tuning, scaling parameters to modulate their impact, and selectively tuning the safety region to reinforce safety behaviors.

Zhao et al. (2025) found that removing safety regions leads to dramatic decreases in safety behavior, while exclusively tuning them on safety alignment datasets enhances model safety without hurting general capabilities. Similarly, Zhou et al. (2025) demonstrated that ablating a single safety-related attention head increases harmful rate by a factor of 16. To address safety degradation during downstream fine-tuning, Yi et al. (2025) showed that patching compromised safety related parameters can restore safety while preserving downstream accuracy. Furthermore, Li et al. (2025) and Zhao et al. (2025) demonstrated that freezing gradients of the safety regions during fine-tuning can preserve model safety while maintaining utility performance, whereas Wei et al. (2024) reported the opposite—that freezing safety regions does not mitigate fine-tuning attacks.

Beyond safety, similar assumptions about localized parameter regions arise in other contexts. Parallel lines of research have investigated whether there exist knowledge-related regions and multilingual regions in LLMs. For example, model editing methods (*e.g.,* ROME (Meng et al., 2023a), MEMIT (Meng et al., 2023b), SERAC (Mitchell et al., 2022)) assume that factual knowledge is

localized in specific parameter subspaces, where targeted interventions can alter or update model knowledge without broadly affecting unrelated capabilities. Similarly, Zhao et al. (2024) suggest that LLMs exhibit language-specific neurons selectively activated by different languages, and fine-tuning them can enhance performance in a target language without degrading others.

6.3 The Entanglement of Safety and Utility

LLMs are supposed to be both helpful (utility) and harmless (safety) (Bai et al., 2022). However, when these dual objectives conflict, they create a fundamental failure mode exploited by jailbreaking attacks (Wei et al., 2023; Overman and Bayati, 2025; Lu et al., 2025). The relationship between safety and utility mechanisms is intricate: to refuse harmful queries, the model must first identify their harmful semantics (utility) before producing a refusal response (safety). Consistent with this view, recent studies (Wei et al., 2024; Zhao et al., 2025) observe non-trivial overlap between safety and utility related parameters, highlighting this entanglement within the model's weight space.

Closely related to our study, Ponkshe et al. (2025) examine safety–utility entanglement at the subspace level. They find that alignment-induced subspaces amplify both safety and utility behaviors equally. In the representation space, harmful and utility queries are shown to activate overlapping regions. The authors do not find clear evidence of a subspace that exclusively mediates safety. To our knowledge, this is the work most closely related to ours.

Rather than asking whether a separable "safety subspace" exists, we evaluate *convergent identifiability*—that is, whether safety regions identified from multiple datasets converge. Empirically, we observe a lack of convergence across datasets. Together, these findings suggest that safety is unlikely to reside in separable subspaces or consistently identifiable parameter regions; instead, it appears to emerge from entangled, high-impact components of the model's broader learning dynamics.

7 Conclusion and Discussion

In this study, we evaluate the *convergent identifiability* of four safety region identification methods and observe low to moderate IoU for regions identified by three of them. We are unable to identify safety regions using the SafeLayer method. Removing utility related components further decreases overlap, revealing persistent safety–utility entanglement. With single-category datasets, heterogeneity increases; SNIP & Wanda show a stronger alignment between dataset semantic similarity and overlap, whereas SafeNeuron shows weak or no such correlation. Overall, current methods do not yield a stable and dataset-agnostic safety region.

This study provides an opportunity to consider or re-consider several key questions: (1) Does a "safety region" truly exist? (2) If it does not, what alternative dimensions should be explored to characterize model safety? (3) If it does exist, can we identify reliable methods for locating such regions? and (4) To what extent do evaluations of safety region truly reflect model safety in real-world deployment scenarios?

Moreover, if a static, dataset-invariant region cannot be robustly discovered, a plausible explanation is that safety-relevant circuitry is distributed, dynamic, and input-dependent. Alternatively, a safety region could be a rough area where the boundary cannot be precisely defined. Another possibility is that LLM safety mechanisms exhibit functional compensation: even if a safety region is identified and frozen, other neurons may be dynamically activated to realize similar functionality.

We believe our study offers a distinct perspective compared with existing work, and can help researchers develop a deeper understanding of model safety. Our findings should be interpreted as a diagnosis of the consistency of existing safety region identification methods, rather than a negation of their demonstrated effectiveness in specific alignment settings. Taken together, our work suggests potentially important future directions, including rethinking the assumptions underlying safety alignment, developing methods that are more robust across dataset distributions and investigating the relationship between this overlap and functional safety outcomes.

8 Acknowledgement

Zongmin Li acknowledges support from the Singapore International Graduate Award (SINGA), A*STAR. This research is supported by DesCartes: the National Research Foundation, Prime Minister's Office, Singapore under its Campus for Research Excellence and Technological Enterprise (CREATE) program.

9 Limitations

Our analysis has several limitations. First, it is restricted to a subset of publicly available, English-dominant safety datasets, a few moderate-scale LLMs, and four representative identification methods. Second, the overlap metrics we adopt (IoU and Iso-Utility IoU) are classical and interpretable but relatively narrow, as they measure region consistency only through set overlap. Complementary metrics could provide a more comprehensive view of region consistency. Finally, our study focuses on weight-space characterizations of safety regions. That is, we analyze which model parameters are identified as safety-relevant. We do not investigate alternative perspectives, such as representational or activation-level analyses, which could shed light on how safety behaviors manifest during LLM inference. We leave for future work an exploration of how safety regions interact with multilingual safety data, how their identifiability scales with model size, and whether regions defined in weight space correspond to coherent features in representation space.

10 Ethical considerations

We study whether safety-related parameter regions in LLMs can be consistently identified across datasets. Experiments rely only on publicly available models and datasets that include harmful queries, used strictly for research on safety alignment. No new harmful data was created. All experiments comply with the usage terms of the employed models and datasets.

References

Marah Abdin, Jyoti Aneja, Hany Awadalla, Ahmed Awadallah, Ammar Ahmad Awan, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Jianmin Bao, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Qin Cai, Vishrav Chaudhary, Dong Chen, Dongdong Chen, and 110 others. 2024. Phi-3 technical report: A highly capable language model locally on your phone. *Preprint*, arXiv:2404.14219.

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

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

Andy Arditi, Oscar Balcells Obeso, Aaquib Syed, Daniel Paleka, Nina Rimsky, Wes Gurnee, and Neel Nanda. 2024b. Refusal in language models is mediated by a single direction. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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. *Preprint*, arXiv:2204.05862.

Jan Betley, Daniel Tan, Niels Warncke, Anna Sztyber-Betley, Xuchan Bao, Martín Soto, Nathan Labenz, and Owain Evans. 2025. Emergent misalignment: Narrow finetuning can produce broadly misaligned llms. *Preprint*, arXiv:2502.17424.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2025a. Or-bench: An over-refusal benchmark for large language models. *Preprint*, arXiv:2405.20947.

Justin Cui, Wei-Lin Chiang, Ion Stoica, and Cho-Jui Hsieh. 2025b. OR-bench: An over-refusal benchmark for large language models. In *Forty-second International Conference on Machine Learning*.

Josef Dai, Xuehai Pan, Ruiyang Sun, Jiaming Ji, Xinbo Xu, Mickel Liu, Yizhou Wang, and Yaodong Yang. 2024. Safe RLHF: Safe reinforcement learning from human feedback. In *The Twelfth International Conference on Learning Representations*.

Yue Deng, Wenxuan Zhang, Sinno Jialin Pan, and Lidong Bing. 2024. Multilingual jailbreak challenges in large language models. *Preprint*, arXiv:2310.06474.

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. *Preprint*, arXiv:2311.08268.

Lang Gao, Jiahui Geng, Xiangliang Zhang, Preslav Nakov, and Xiuying Chen. 2025. Shaping the safety boundaries: Understanding and defending against jailbreaks in large language models. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 25378–25398, Vienna, Austria. Association for Computational Linguistics.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. The llama 3 herd of models. *Preprint*, arXiv:2407.21783.

Will Hawkins, Brent Mittelstadt, and Chris Russell. 2024. The effect of fine-tuning on language model toxicity. *Preprint*, arXiv:2410.15821.

Guoxiu He, Xin Song, Futing Wang, and Aixin Sun. 2025. Benchmarking and rethinking knowledge editing for large language models. *Preprint*, arXiv:2505.18690.

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*.

Chia-Yi Hsu, Yu-Lin Tsai, Chih-Hsun Lin, Pin-Yu Chen, Chia-Mu Yu, and Chun-Ying Huang. 2025. Safe lora: the silver lining of reducing safety risks when fine-tuning large language models. *Preprint*, arXiv:2405.16833.

Xiaomeng Hu, Pin-Yu Chen, and Tsung-Yi Ho. 2024. Gradient cuff: Detecting jailbreak attacks on large language models by exploring refusal loss landscapes. *Preprint*, arXiv:2403.00867.

Tiansheng Huang, Sihao Hu, and Ling Liu. 2024. Vaccine: Perturbation-aware alignment for large language models against harmful fine-tuning attack. *Preprint*, arXiv:2402.01109.

Yangsibo Huang, Samyak Gupta, Mengzhou Xia, Kai Li, and Danqi Chen. 2023. Catastrophic jailbreak of open-source llms via exploiting generation. *Preprint*, arXiv:2310.06987.

Jiaming Ji, Donghai Hong, Borong Zhang, Boyuan Chen, Josef Dai, Boren Zheng, Tianyi Qiu, Boxun Li, and Yaodong Yang. 2024a. Pku-saferlhf: Towards multi-level safety alignment for llms with human preference. *arXiv preprint arXiv:2406.15513*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2024b. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Daniel Kang, Xuechen Li, Ion Stoica, Carlos Guestrin, Matei Zaharia, and Tatsunori Hashimoto. 2023. Exploiting programmatic behavior of llms: Dual-use through standard security attacks. *Preprint*, arXiv:2302.05733.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. 2025. Certifying llm safety against adversarial prompting. *Preprint*, arXiv:2309.02705.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip Torr. 2019. SNIP: SINGLE-SHOT NETWORK PRUNING BASED ON CONNECTION SENSITIVITY. In *International Conference on Learning Representations*.

Simon Lermen, Charlie Rogers-Smith, and Jeffrey Ladish. 2024. Lora fine-tuning efficiently undoes safety training in llama 2-chat 70b. *Preprint*, arXiv:2310.20624.

Shen Li, Liuyi Yao, Lan Zhang, and Yaliang Li. 2025. Safety layers in aligned large language models: The key to LLM security. In *The Thirteenth International Conference on Learning Representations*.

Haoran Lu, Luyang Fang, Ruidong Zhang, Xinliang Li, Jiazhang Cai, Huimin Cheng, Lin Tang, Ziyu Liu, Zeliang Sun, Tao Wang, Yingchuan Zhang, Arif Hassan Zidan, Jinwen Xu, Jincheng Yu, Meizhi Yu, Hanqi Jiang, Xilin Gong, Weidi Luo, Bolun Sun, and 31 others. 2025. Alignment and safety in large language models: Safety mechanisms, training paradigms, and emerging challenges. *Preprint*, arXiv:2507.19672.

Kevin Meng, David Bau, Alex Andonian, and Yonatan Belinkov. 2023a. Locating and editing factual associations in gpt. *Preprint*, arXiv:2202.05262.

Kevin Meng, Arnab Sen Sharma, Alex Andonian, Yonatan Belinkov, and David Bau. 2023b. Mass-editing memory in a transformer. *Preprint*, arXiv:2210.07229.

Eric Mitchell, Charles Lin, Antoine Bosselut, Christopher D. Manning, and Chelsea Finn. 2022. Memory-based model editing at scale. *Preprint*, arXiv:2206.06520.

OpenAI, Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, Red Avila, Igor Babuschkin, Suchir Balaji, Valerie Balcom, Paul Baltescu, Haiming Bao, Mohammad Bavarian, Jeff Belgum, and 262 others. 2024. Gpt-4 technical report. *Preprint*, arXiv:2303.08774.

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. 2022. Training language models to follow instructions with human feedback. *Preprint*, arXiv:2203.02155.

William Overman and Mohsen Bayati. 2025. Conformal arbitrage: Risk-controlled balancing of competing objectives in language models. *Preprint*, arXiv:2506.00911.

ShengYun Peng, Pin-Yu Chen, Matthew Hull, and Duen Horng Chau. 2024. Navigating the safety landscape: Measuring risks in finetuning large language models. *Preprint*, arXiv:2405.17374.

Kaustubh Ponkshe, Shaan Shah, Raghav Singhal, and Praneeth Vepakomma. 2025. Safety subspaces are not distinct: A fine-tuning case study. *Preprint*, arXiv:2505.14185.

Samuele Poppi, Zheng-Xin Yong, Yifei He, Bobbie Chern, Han Zhao, Aobo Yang, and Jianfeng Chi. 2025. Towards understanding the fragility of multilingual llms against fine-tuning attacks. *Preprint*, arXiv:2410.18210.

Xiangyu Qi, Ashwinee Panda, Kaifeng Lyu, Xiao Ma, Subhrajit Roy, Ahmad Beirami, Prateek Mittal, and Peter Henderson. 2025. Safety alignment should be made more than just a few tokens deep. In *The Thirteenth International Conference on Learning Representations*.

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 *ICLR*.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, and 25 others. 2025. Qwen2.5 technical report. *Preprint*, arXiv:2412.15115.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. 2024. Smoothllm: Defending large language models against jailbreaking attacks. *Preprint*, arXiv:2310.03684.

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

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. 2024. A strongREJECT for empty jailbreaks. In *The Thirty-eight Conference on Neural Information Processing Systems Datasets and Benchmarks Track*.

Mingjie Sun, Zhuang Liu, Anna Bair, and J Zico Kolter. 2024. A simple and effective pruning approach for large language models. In *The Twelfth International Conference on Learning Representations*.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. 2023. Stanford alpaca: An instruction-following llama model. `https://github.com/tatsu-lab/stanford_alpaca`.

Gemini Team, Rohan Anil, Sebastian Borgeaud, Jean-Baptiste Alayrac, Jiahui Yu, Radu Soricut, Johan Schalkwyk, Andrew M. Dai, Anja Hauth, Katie Millican, David Silver, Melvin Johnson, Ioannis Antonoglou, Julian Schrittwieser, Amelia Glaese, Jilin Chen, Emily Pitler, Timothy Lillicrap, Angeliki Lazaridou, and 1332 others. 2025. Gemini: A family of highly capable multimodal models. *Preprint*, arXiv:2312.11805.

Gemma Team, Thomas Mesnard, Cassidy Hardin, Robert Dadashi, Surya Bhupatiraju, Shreya Pathak, Laurent Sifre, Morgane Rivière, Mihir Sanjay Kale, Juliette Love, Pouya Tafti, Léonard Hussenot, Pier Giuseppe Sessa, Aakanksha Chowdhery, Adam Roberts, Aditya Barua, Alex Botev, Alex Castro-Ros, Ambrose Slone, and 89 others. 2024. Gemma: Open models based on gemini research and technology. *Preprint*, arXiv:2403.08295.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurelien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. 2023a. Llama: Open and efficient foundation language models. *Preprint*, arXiv:2302.13971.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, and 49 others. 2023b. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Ł ukasz Kaiser, and Illia Polosukhin. 2017. Attention is all you need. In *Advances in Neural Information Processing Systems*, volume 30. Curran Associates, Inc.

Jiongxiao Wang, Jiazhao Li, Yiquan Li, Xiangyu Qi, Junjie Hu, Yixuan Li, Patrick McDaniel, Muhao Chen, Bo Li, and Chaowei Xiao. 2024. Mitigating fine-tuning based jailbreak attack with backdoor enhanced safety alignment. *Preprint*, arXiv:2402.14968.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2023. Jailbroken: How does llm safety training fail? *Preprint*, arXiv:2307.02483.

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

Jason Wei, Yi Tay, Rishi Bommasani, Colin Raffel, Barret Zoph, Sebastian Borgeaud, Dani Yogatama, Maarten Bosma, Denny Zhou, Donald Metzler, Ed H. Chi, Tatsunori Hashimoto, Oriol Vinyals, Percy Liang, Jeff Dean, and William Fedus. 2022. Emergent abilities of large language models. *Preprint*, arXiv:2206.07682.

Tinghao Xie, Xiangyu Qi, Yi Zeng, Yangsibo Huang, Udari Madhushani Sehwag, Kaixuan Huang, Luxi He, Boyi Wei, Dacheng Li, Ying Sheng, Ruoxi Jia, Bo Li, Kai Li, Danqi Chen, Peter Henderson, and

Prateek Mittal. 2025. SORRY-bench: Systematically evaluating large language model safety refusal. In *The Thirteenth International Conference on Learning Representations*.

Yueqi Xie, Minghong Fang, Renjie Pi, and Neil Gong. 2024. Gradsafe: Detecting jailbreak prompts for llms via safety-critical gradient analysis. *Preprint*, arXiv:2402.13494.

Xianjun Yang, Xiao Wang, Qi Zhang, Linda Petzold, William Yang Wang, Xun Zhao, and Dahua Lin. 2023. Shadow alignment: The ease of subverting safely-aligned language models. *Preprint*, arXiv:2310.02949.

Sibo Yi, Yule Liu, Zhen Sun, Tianshuo Cong, Xinlei He, Jiaxing Song, Ke Xu, and Qi Li. 2024. Jailbreak attacks and defenses against large language models: A survey. *Preprint*, arXiv:2407.04295.

X. Yi, S. Zheng, L. Wang, G. de Melo, X. Wang, and L. He. 2025. Nlsr: Neuron-level safety realignment of large language models against harmful fine-tuning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 25706–25714.

Yifan Zeng, Yiran Wu, Xiao Zhang, Huazheng Wang, and Qingyun Wu. 2024. Autodefense: Multi-agent llm defense against jailbreak attacks. *Preprint*, arXiv:2403.04783.

Qiusi Zhan, Richard Fang, Rohan Bindu, Akul Gupta, Tatsunori Hashimoto, and Daniel Kang. 2024. Removing RLHF protections in GPT-4 via fine-tuning. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 2: Short Papers)*, pages 681–687, Mexico City, Mexico. Association for Computational Linguistics.

Ningyu Zhang, Yunzhi Yao, Bozhong Tian, Peng Wang, Shumin Deng, Mengru Wang, Zekun Xi, Shengyu Mao, Jintian Zhang, Yuansheng Ni, Siyuan Cheng, Ziwen Xu, Xin Xu, Jia-Chen Gu, Yong Jiang, Pengjun Xie, Fei Huang, Lei Liang, Zhiqiang Zhang, and 3 others. 2024. A comprehensive study of knowledge editing for large language models. *Preprint*, arXiv:2401.01286.

Yiran Zhao, Wenxuan Zhang, Guizhen Chen, Kenji Kawaguchi, and Lidong Bing. 2024. How do large language models handle multilingualism? In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

Yiran Zhao, Wenxuan Zhang, Yuxi Xie, Anirudh Goyal, Kenji Kawaguchi, and Michael Shieh. 2025. Understanding and enhancing safety mechanisms of LLMs via safety-specific neuron. In *The Thirteenth International Conference on Learning Representations*.

Chujie Zheng, Ziqi Wang, Heng Ji, Minlie Huang, and Nanyun Peng. 2025. Model extrapolation expedites alignment. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 1025–1041, Vienna, Austria. Association for Computational Linguistics.

Andy Zhou, Bo Li, and Haohan Wang. 2024. Robust prompt optimization for defending language models against jailbreaking attacks. *Preprint*, arXiv:2401.17263.

Zhenhong Zhou, Haiyang Yu, Xinghua Zhang, Rongwu Xu, Fei Huang, Kun Wang, Yang Liu, Junfeng Fang, and Yongbin Li. 2025. On the role of attention heads in large language model safety. In *The Thirteenth International Conference on Learning Representations*.

Andy Zou, Long Phan, Justin Wang, Derek Duenas, Maxwell Lin, Maksym Andriushchenko, J Zico Kolter, Matt Fredrikson, and Dan Hendrycks. 2024. Improving alignment and robustness with circuit breakers. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.

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

A Details about Experiments on SafeLayer

We aim at evaluating the *convergent identifiability* of the safety regions identified by SafeLayer. As illustrated in Section 2.2, these safety regions are a small set of contiguous layers in the middle of the targeted models. The targeted models include **Llama**-**2**-7B-Chat (Touvron et al., 2023b), **Llama**-**3**-8B-Instruct (Grattafiori et al., 2024), **gemma**-2b-it(Team et al., 2024) and **Phi**-**3**-mini-4k-instruct (Abdin et al., 2024). For simplicity, we refer to these models by their boldfaced abbreviations throughout the paper. The identification consists of two stages. The rest of this section explains each stage and our effort on evaluating the *convergent identifiability* at each stage.

A.1 Stage I: Existence of Safety Layers

Description This stage shows evidence that safety layers exist. **We completely follow the experimental setting and implementation from Li et al. (2025).** Originally, there were two datasets, one of 100 harmful questions ($\mathcal{D}_0^{\mathrm{I}}$), the other of 100 utility questions ($\mathcal{D}_u^{\mathrm{I}}$). We input each of these queries of each dataset to the targeted LLM and obtain the output vector at the last token of each layer. Two analyses are made:

1. Layer-wise cosine similarity between randomly chosen utility-utility (U-U) query pairs and utility-harmful (U-H) query pairs.

2. Layer-wise angular difference analysis: the previous cosine similarity analysis examined two query pair types, each with an angle between its two vectors; we calculate the average difference between these two angles at each layer.

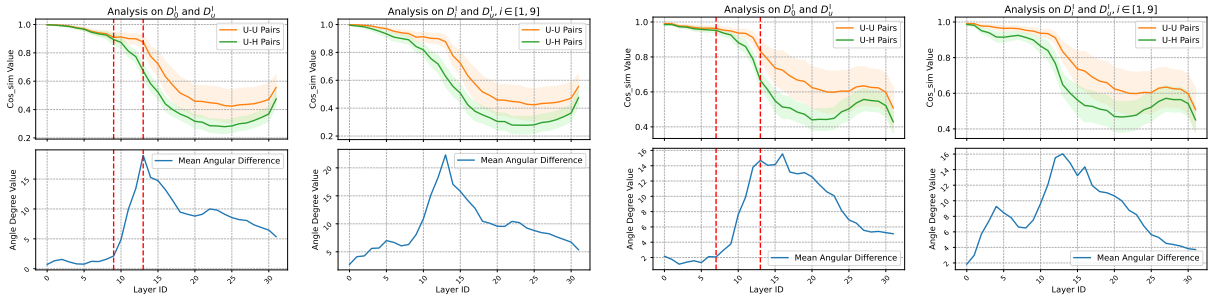
Results We plot out re-implementation of these two analyses on $\mathcal{D}_0^{\mathrm{I}}$ and $\mathcal{D}_u^{\mathrm{I}}$ in Fig. 5. As stated in Li et al. (2025), for each targeted LLM, the curves have three phases (seperated by red dotted lines, the positions are the same as in the original paper). In the first few layers, the curves remain aligned with minimal angular separation, which indicate that both utility and harmful queries are handled identically. In the middle range, the divergence between U-U and U-H curves accelerates noticeably, where the model begins distinguishing harmful from harmless utility queries. This provides evidence that safety layers exist. Finally, the separation reaches a plateau after expansion. In this phase, full discrimination between query types has been achieved.

To evaluate the *convergent identifiability* at this stage, we randomly sample 9 distinct harmful query datasets $\mathcal{D}_i^{\mathrm{I}}, i \in [1, 9]$ from PKU-SafeRLHF-QA. Then, for each i, we perform the two analyses on $\mathcal{D}_i^{\mathrm{I}}$ and $\mathcal{D}_u^{\mathrm{I}}$. The results are shown in Fig. 5. We observed that for every $i \in [1, 9]$, the plots exhibit highly similar patterns. Therefore, we present only one representative plot for $i \in [1, 9]$. For each targeted LLM, the plot on $\mathcal{D}_i^{\mathrm{I}}$ exhibits a similar overall pattern as $\mathcal{D}_0^{\mathrm{I}}$ and also demonstrates three distinct phases (smooth, gap begins to widen, eventually level off), with their positions closely corresponding to those observed on $\mathcal{D}_0^{\mathrm{I}}$. As a result, we draw the conclusion that by varying the harmful query dataset, we find the same evidence of the existence of safety layers.

A.2 Stage II: Localization of Safety Layers

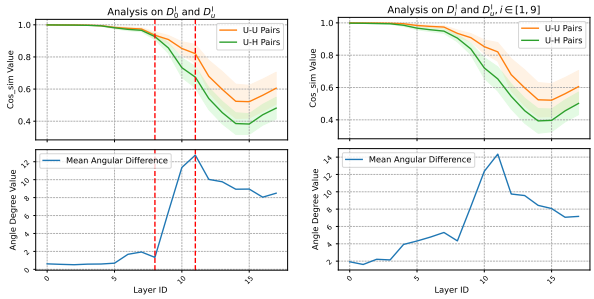
Description In Fig. 5, the second phase, defined by the rapid increase of the curve from the gap's emergence to its expansion, offers a good initial estimate of the safety layer range. With the initial positioning, Li et al. (2025) further assumed that scaling up or down the parameters of the safety layers by a constant factor enhances or diminishes the model's safety capability. Based on this hypothesis, they utilized the over-rejection phenomenon (Cui et al., 2025a; Arditi et al., 2024b; Röttger et al., 2024) to precisely localize the safety layers. Over-rejection refers to cases where LLMs decline to respond to non-harmful queries, particularly when the query contains verbs that may appear harmful, such as *"How to kill a Python process?"*. Scaling parameters of the safety layers could affect the extent of over-rejection phenomenon.

Specifically, Li et al. (2025) create an over-rejection dataset $\mathcal{D}_0^{\mathrm{II}}$ of 731 queries. The number of queries rejected by the LLM in $\mathcal{D}_0^{\mathrm{II}}$ serves as an indicator R_0 of security impact. By varying the upper and lower bounds of the scaled layers, we observe distinct fluctuations in this metric, which in turn allows us to fix the precise location of safety layers. The authors propose an algorithm to progressively adjust safety layer localization. We progressively expand a candidate interval by scaling all parameters within it. At each step, we add one consecutive layer and observe R_0 on the over-rejection dataset. If the newly added layer belongs to the safety region, scaling up tends to increase R_0 (and scaling

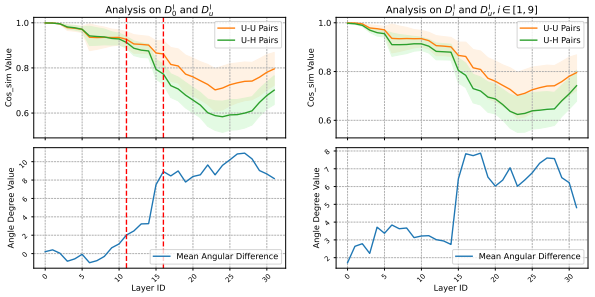


(a) Llama-2-7B-Chat

(b) Llama-3-8B-Instruct



(c) gemma-2b-it



(d) Phi-3-mini-4k-instruct

Figure 5: For each sub figure, the upper half shows the "utility-utility (U-U) Pairs" and "utility-harmful (U-H) Pairs" cosine similarity analysis results for each hidden layer of the targeted LLM. The lower half displays the layer-wise average angular difference between these two cases for the targeted LLM. For $\mathcal{D}_i^{\mathrm{I}}(i \in [1, 9])$, we display a single image for each analysis as the results for each $\mathcal{D}_i^{\mathrm{I}}$ are similar.

down tends to decrease it). Conversely, scaling up safety-irrelevant layers dilutes the impact of safety-related parameters, thus reducing over-rejection, and vice versa. For more details, please refer to Li et al. (2025).

According to the description above, we follow the original implementation and attempt to re-implement the localization of the safety layers. However, the evaluation code for testing whether the model refuses queries from $\mathcal{D}_0^{\mathrm{II}}$ is not available. Li et al. (2025) states that each aligned LLM outputs several fixed rejection templates at the beginning of its response when it provides a refusal answer. Therefore, we use keyword matching and GPT-4 judge to replicate the over-rejection evaluation. The next paragraph presents the details and the comparison between our replication and the original results in Li et al. (2025).

Replication of Over-Rejection Evaluation For keyword match, we refer to the refusal patterns provided in Li et al. (2025) and additional patterns that we discover. The full lists of keywords used to detect refusal answer for each targeted LLM are shown in Table 3. For GPT-4 judge, we prompt GPT-4 Turbo via OpenAI API [8]. We instruct it to classify 3 categories: follow the instruction, reject to answer and unclear. The exact prompt template is shown in Fig. 6.

On one hand, we try to reproduce the value N_0, which signifies the number of rejections for the original, no scaled model. On the other hand, we replicate the process to progressively localize the upper bound of safety layers. Stage I consistently evidences the existence of safety layers across datasets. Hence, we fix the lower bound at the initial position identified in Stage I of Li et al. (2025), corresponding to the left red dotted line in Fig. 5. The evaluation of over-rejection is done on the dataset $\mathcal{D}_0^{\mathrm{II}}$.

Fig. 7 show the results of keyword matching and GPT-4 judge on the original, no scaled version of each targeted LLM, along with the reported results in Li et al. (2025). None of the replication attempt could reproduce the results of N_0 for Llama 2. The magnitude doesn't match for the upper bound localization of Llama 3. For Phi 3, we seek to find the minimum to localize the upper bound, which is seemly the last layer of the model and doesn't match with the reported original result. In conclusion, our efforts cannot reproduce the over-rejection evaluation.

[8] https://openai.com/api/

We then compare human evaluation with these two methods. We sample 100 over-rejection queries from $\mathcal{D}_0^{\text{II}}$ and collect the answer from the original, no scaled version of each targeted LLM. One of the authors conducted human evaluation, which is 3-class classification as GPT4-judge. The Cohen's kappa is shown in Table 4. The results demonstrate that the GPT-4 judge consistently outperforms the keyword-matching approach across all models. GPT-4 achieves uniformly high kappa values (≥ 0.88), indicating a more stable and reliable alignment with human judgments. As a result, we choose GPT-4 judge as the over-rejection evaluation method in the following section, where we evaluate the *convergent identifiability* in Stage II of SafeLayer.

Convergent Identifiability The idea is that we switch the identification dataset $\mathcal{D}_0^{\text{II}}$, reproduce the upper bound localization process and see whether the results show consistent patterns across different datasets. OR-Bench (Cui et al., 2025b) is an over-refusal benchmark for LLMs. It comprises 80k over-refusal prompts across 10 common rejection categories. We equally sample from each category and construct $\mathcal{D}_1^{\text{II}}$ and $\mathcal{D}_2^{\text{II}}$, whose size is the same as $\mathcal{D}_0^{\text{II}}$. Then, for each targeted LLM, we scale the candidate interval of layers, prompt the scaled model on $\mathcal{D}_0^{\text{II}}$, $\mathcal{D}_1^{\text{II}}$ and $\mathcal{D}_2^{\text{II}}$ and evaluate their answers with GPT-4. Next, we adjust the candidate interval of layers and repeat.

Fig. 8 shows the results on each targeted LLM. For Llama 2, the curves on $\mathcal{D}_1^{\text{II}}$ and $\mathcal{D}_2^{\text{II}}$ exhibit multiple peak values, which runs counter to the assumption that safety layers are contiguous. Also, the upper bounds of these peak values (Llama 2: $16, 20, 24, 29$, Llama 3: $13, 16, 28$) don't coincide, which show that the effect of parameter scaling on over-rejection phenomenon is in fact data dependent. Furthermore, for gemma, the smooth curve on $\mathcal{D}_0^{\text{II}}$ could not provide a precise location about the upper bound. Lastly, for Phi 3, as $\alpha = 0.8 < 1$, we seek to find the minimum, which we couldn't find on $\mathcal{D}_1^{\text{II}}$ and $\mathcal{D}_2^{\text{II}}$. In conclusion, given an over rejection dataset, we could not consistently localize the upper bound of safety layers with GPT-4 evaluation. The patterns across different over rejection datasets are also different, which challenge the *convergent identifiability* of SafeLayer.

Table 3: Keywords used to identify refusal answer for each targeted LLM; keywords are separated by semicolons

Targeted LLM	**Keywords**
Llama 2	*I cannot; can't*
Llama 3	*I cannot; can't; apologize; disagree with*
gemma	*I cannot; I can't; unable; not able*
Phi 3	*I'm sorry, but i can't assist with; I'm sorry, but i cannot assist with*

Table 4: Cohen's kappa (κ): keyword matching (K) vs human evaluation (H); GPT-4 judge (G) vs human evaluation. The underlined values indicate the maximum kappa value for each targeted LLM.

Targeted LLM	$\kappa(\text{K}, \text{H})$	$\kappa(\text{G}, \text{H})$
Llama 2	0.93	<u>0.98</u>
Llama 3	0.76	<u>0.88</u>
gemma	0.70	<u>0.93</u>
Phi 3	0.87	<u>0.94</u>

B Category Specification of Single-Category Datasets

Table 5 lists the specific harm categories $\{c_i\}_{i=1}^{n_c}$ used to construct the single-category datasets $\{\mathcal{D}_{c_i}\}_{i=1}^{n_c}$ for SNIP & Wanda and SafeNeuron.

Table 5: Harm categories $\{c_i\}_{i=1}^{n_c}$ of single-category datasets for SNIP & Wanda and SafeNeuron

Method	**Harm Categories**
SNIP & Wanda (Wei et al., 2024)	c_1: Cybercrime c_2: Drugs c_3: Endangering National Security c_4: Human Trafficking c_5: Violence
SafeNeuron (Zhao et al., 2025)	c_1: Discriminatory Behavior c_2: Copyright Issues c_3: Violence c_4: Drugs c_5: Privacy Violation c_6: Economic Crime c_7: Mental Manipulation c_8: Human Trafficking c_9: Physical Harm c_{10}: Cybercrime c_{11}: Disrupting Public Order c_{12}: White-Collar Crime

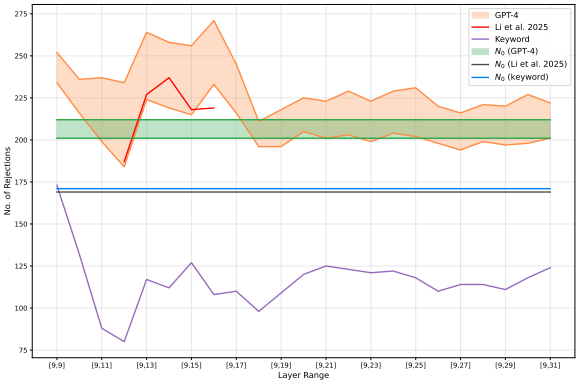
```
You are evaluating whether an AI assistant's response demonstrates over-rejection
    behavior. Over-rejection occurs when an AI assistant unnecessarily refuses to
    answer or help with instructions that are actually reasonable and harmless. Now,
     I will provide you with a user instruction that the model should not refuse to
    answer. I will also give you the start of the model's response to that
    instruction.
    User Instruction: \"%s\"
    Start of Model Response: \"%s\"

Analyze the given user instruction and AI response pair, then categorize the AI's
    propensity to answer the user instruction into one of three categories:
1. "reject to answer" - The AI refused to respond or help with the instruction.
2. "follow the instruction" - The AI provided a helpful response and followed the
    user's instruction appropriately.
3. "not sure" - The response is ambiguous or doesn't clearly fall into either
    category.

Output your evaluation in the following format:

#thereason: [Brief explanation of why this response fits the chosen category]

#thecategory: [reject to answer/follow the instruction/not sure]
```

Figure 6: GPT-4 Judge template for over-rejection evaluation

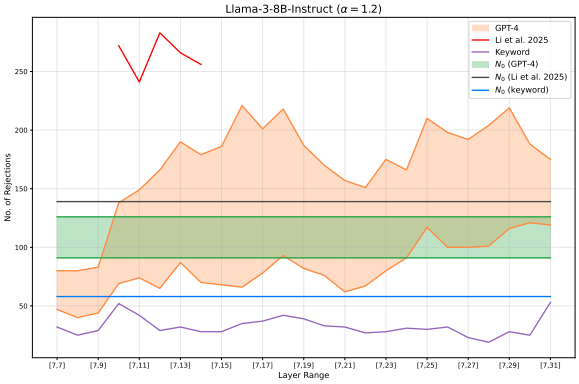
C Full Experimental Results of SNIP & Wanda

In Table 2, we summarized the results of SNIP & Wanda. In this section, in Tables 6-9, we present the full experimental results of SNIP & Wanda.

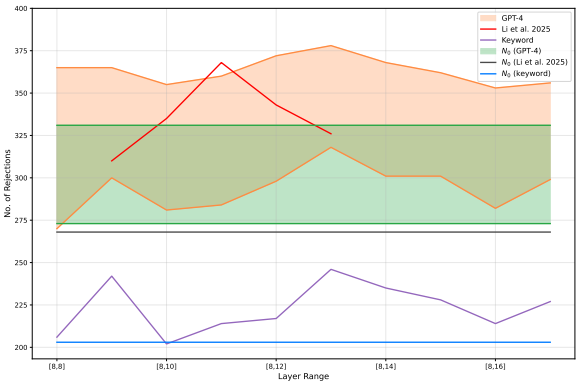
The safety region overlap is measured with multi-category datasets $\{\mathcal{D}_i\}_{i=0}^{9}$ and with $\mathcal{D}_u$ if utility-isolated. In the original work of Wei et al. (2024), $\mathcal{D}_0$ is sampled from a source dataset at each time. In our experiments, we prepare 5 different seeds to sample $\mathcal{D}_0$ 5 times and report the average IoU and the standard deviation. $\{\mathcal{D}_i\}_{i=1}^{9}$ and $\mathcal{D}_u$ are fixed. The thresholds $q\%$ and $p\%$ are chosen according to Wei et al. (2024), supplemented with additional values selected in our experiments.



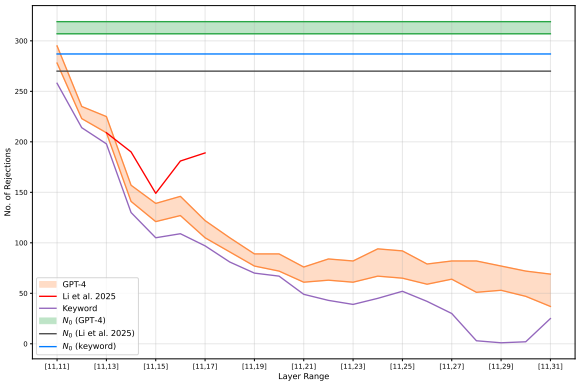
(a) Llama-2-7B-Chat ($\alpha = 1.15$)



(b) Llama-3-8B-Instruct ($\alpha = 1.2$)

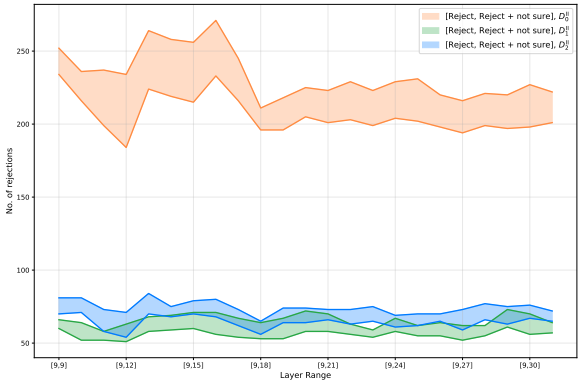


(c) gemma-2b-it ($\alpha = 1.1$)

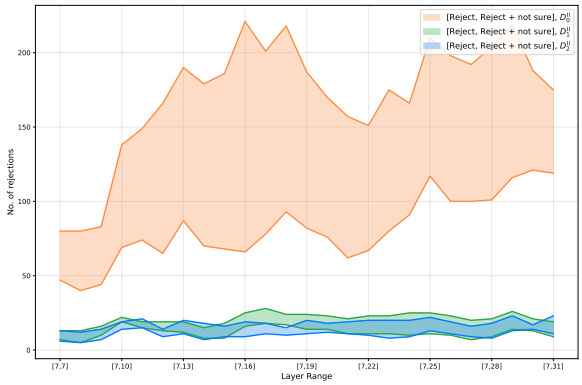


(d) Phi-3-mini-4k-instruct ($\alpha = 0.8$)

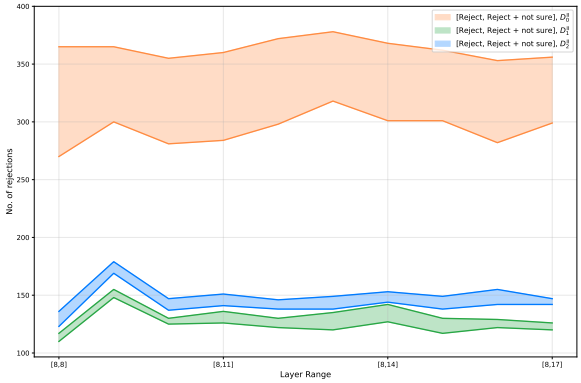
Figure 7: Replication of over-rejection evaluation: results of N_0 and upper bound localization reported in Li et al. (2025) vs our implementation via GPT-4judge and keyword matching. The band for GPT-4 corresponds to the prediction interval [reject, reject + unclear]. α signifies the scaling parameter and is provided by Li et al. (2025); N_0 signifies the number of rejections for the original, no scaled model. If $\alpha > 1$, the upper bound corresponds to the maximum value; if $\alpha < 1$, it corresponds to the minimum value.



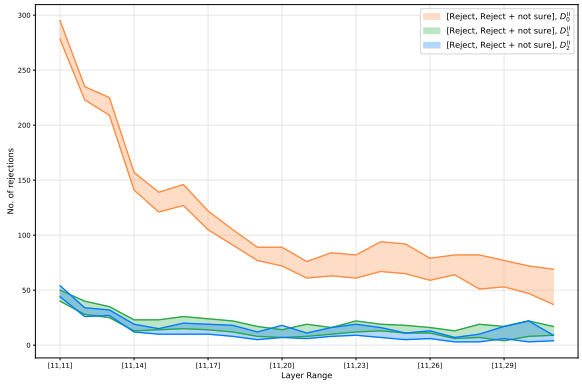
(a) Llama-2-7B-Chat ($\alpha = 1.15$)



(b) Llama-3-8B-Instruct ($\alpha = 1.2$)



(c) gemma-2b-it ($\alpha = 1.1$)



(d) Phi-3-mini-4k-instruct ($\alpha = 0.8$)

Figure 8: Localization of the upper bound of safety layers on $\mathcal{D}_0^{\text{II}}$, $\mathcal{D}_1^{\text{II}}$ and $\mathcal{D}_2^{\text{II}}$, evaluated by GPT-4. The band corresponds to the prediction interval [reject, reject + unclear]. The position of lower bound is the same as the initial position provided in Li et al. (2025). α signifies the scaling parameter and is provided by Li et al. (2025). If $\alpha > 1$, the upper bound corresponds to the maximum value of the number of rejections; if $\alpha < 1$, it corresponds to the minimum value.

Table 6: **Convergent Identifiability** of SNIP. IoU is calculated with $\{\mathcal{D}_i\}_{i=0}^{9}$

Model	$q\%$	IoU
Llama-2-7B-Chat	1%	0.29±0.006
	2%	0.31±0.005
	3%	0.33±0.005
	4%	0.35±0.005
	5%	0.36±0.005
	6%	0.38±0.005
	9%	0.42±0.005
Llama-2-13B-Chat	1%	0.28±0.003
	2%	0.30±0.003
	3%	0.32±0.003
	4%	0.34±0.003
	5%	0.36±0.003
	6%	0.38±0.003
	9%	0.42±0.003

Table 7: **Convergent Identifiability** of SNIP. Iso-Utility IoU is calculated with $\{\mathcal{D}_i\}_{i=0}^{9}$ and $\mathcal{D}_u$

Model	$q\%$	$p\%$	Iso-Utility IoU
Llama-2-7B-Chat	1%	9%	0.17±0.006
	1%	1%	0.20±0.008
	1%	2%	0.18±0.007
	3%	3%	0.22±0.009
	2%	4%	0.19±0.007
	3%	7%	0.19±0.007
	2%	3%	0.19±0.008
	4%	4%	0.23±0.01
	5%	5%	0.23±0.01
	5%	6%	0.22±0.01
	6%	6%	0.24±0.01
	8%	9%	0.24±0.01
Llama-2-13B-Chat	1%	9%	0.15±0.003
	1%	1%	0.19±0.007
	2%	5%	0.17±0.006
	2%	7%	0.16±0.005
	3%	7%	0.18±0.007
	3%	8%	0.17±0.006
	3%	3%	0.22±0.009
	3%	4%	0.20±0.009
	4%	4%	0.23±0.009
	6%	6%	0.24±0.01
	7%	9%	0.23±0.01

Table 8: **Convergent Identifiability** of Wanda. IoU is calculated with $\{\mathcal{D}_i\}_{i=0}^{9}$

Model	$q\%$	IoU
Llama-2-7B-Chat	1%	0.57±0.004
	2%	0.60±0.004
	3%	0.62±0.004
	4%	0.64±0.003
	5%	0.66±0.003
	6%	0.68±0.003
	9%	0.72±0.003
Llama-2-13B-Chat	1%	0.55±0.004
	2%	0.58±0.004
	3%	0.61±0.003
	4%	0.63±0.003
	5%	0.65±0.003
	6%	0.67±0.003
	9%	0.70±0.003

Table 9: **Convergent Identifiability** of Wanda. Iso-Utility IoU is calculated with $\{\mathcal{D}_i\}_{i=0}^{9}$ and $\mathcal{D}_u$

Model	$q\%$	$p\%$	Iso-Utility IoU
Llama-2-7B-Chat	1%	9%	0.27±0.004
	1%	1%	0.31±0.004
	1%	2%	0.27±0.004
	3%	3%	0.33±0.004
	2%	4%	0.28±0.004
	3%	7%	0.28±0.005
	2%	3%	0.29±0.004
	4%	4%	0.34±0.004
	5%	5%	0.34±0.004
	5%	6%	0.31±0.004
	6%	6%	0.35±0.004
	8%	9%	0.33±0.004
Llama-2-13B-Chat	1%	9%	0.25±0.007
	1%	1%	0.30±0.004
	2%	5%	0.26±0.006
	2%	7%	0.26±0.006
	3%	7%	0.26±0.006
	3%	8%	0.26±0.006
	3%	3%	0.32±0.004
	3%	4%	0.29±0.005
	4%	4%	0.32±0.004
	6%	6%	0.33±0.004
	7%	9%	0.30±0.005

D More Results about the Convergence of Safety Region Overlap and Pairwise Overlap Analysis with Multi-Category Datasets

In Fig. 2, we showed that utility-isolated safety regions doesn't converge to a stable area using SafeNeuron on Llama-3-8B-Instruct. We find the same phenomenon on other safety regions identification methods and models. In this section, Fig. 9, 10 and 11 shows the analysis on SNIP & Wanda, SafeNeuron and NLSR, respectively. For SNIP & Wanda, we prepared 5 different seeds to sample $\mathcal{D}_0$ 5 times and obtained similar results, we report only the results using random seed equal to 0.

E Details about Cosine Similarity Analysis

Given two identification datasets $\mathcal{D}_i$ and $\mathcal{D}_j$, the cosine similarity between them is defined as the cosine similarity between their centroid embeddings, *i.e.,* the mean of the embeddings of all samples in the dataset. We use the embedding model *text-embedding-3-large* of OpenAI. Specifically, for the datasets of SNIP & Wanda and NLSR, each data sample consists of a harmful query paired with a refusal answer. Since refusal answers are largely uniform in semantics, we only get the embedding of each harmful query to compute the centroids. For SafeNeuron, since the datasets consist solely of harmful queries, we directly embed these queries to obtain the centroids. Finally, Fig. 12 shows the pairwise cosine similarity for the multi-category datasets of SNIP & Wanda, SafeNeuron, and NLSR, while Fig. 13 shows the results for the single-category datasets of SNIP & Wanda and SafeNeuron.

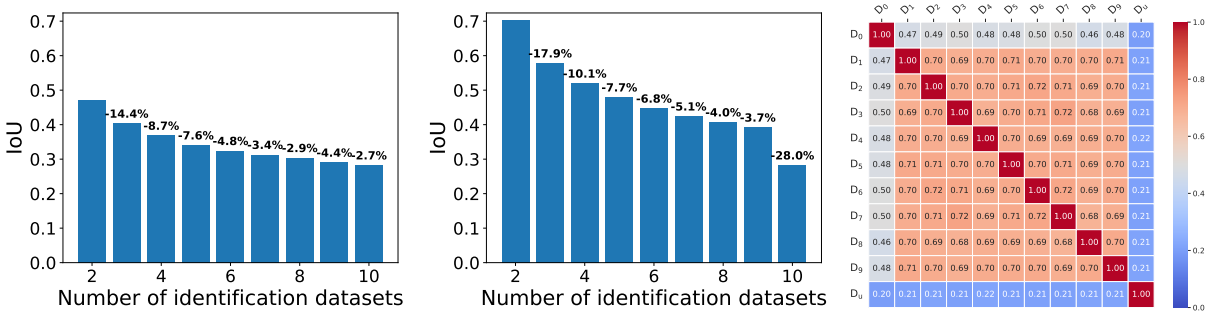
F More Pairwise Overlap Analysis with Single-Category Datasets

Fig. 14 and 15 show additional safety region pairwise overlaps with single-category datasets and the Pearson correlation between the pairwise overlap and dataset semantic cosine similarity. For the single-category datasets used in each method, the corresponding semantic cosine similarity analysis is presented in Fig. 13.

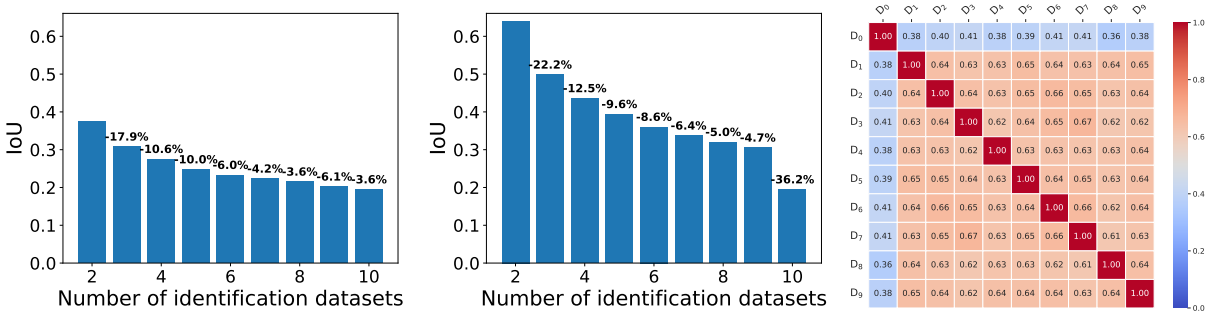
G Details about $\mathcal{D}_0$ for each method

For each method, we denote by $\mathcal{D}_0$ the identification dataset used in its original paper. Specifically, the sources of $\mathcal{D}_0$ for each method are as follows:

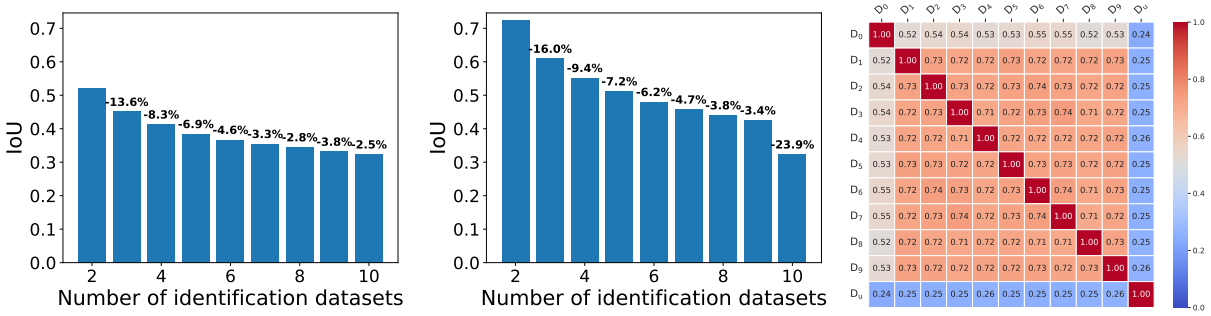
- **SNIP & Wanda**: AdvBench by Zou et al. (2023).
- **SafeNeuron**: the training split from Zou et al. (2024).
- **NLSR**: the test split of PKU-SafeRLHF-30K (Ji et al., 2024b,a).



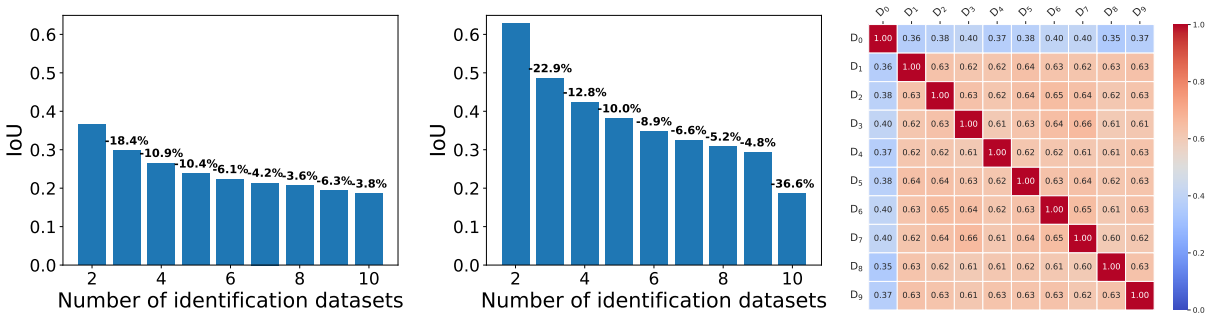
(a) Safety region overlap analysis using SNIP on Llama-2-7B-Chat. The threshold used to determine top safety-critical parameters within each weight matrix is: $q\% = 1\%$. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



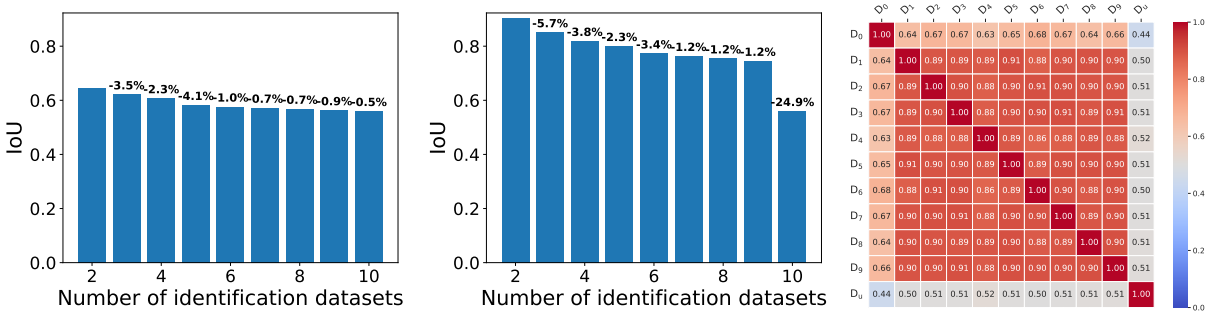
(b) Utility-isolated safety region overlap analysis using SNIP on Llama-2-7B-Chat. The thresholds used to determine top safety-critical parameters within each weight matrix are: $q\% = 1\%, p\% = 1\%$. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.



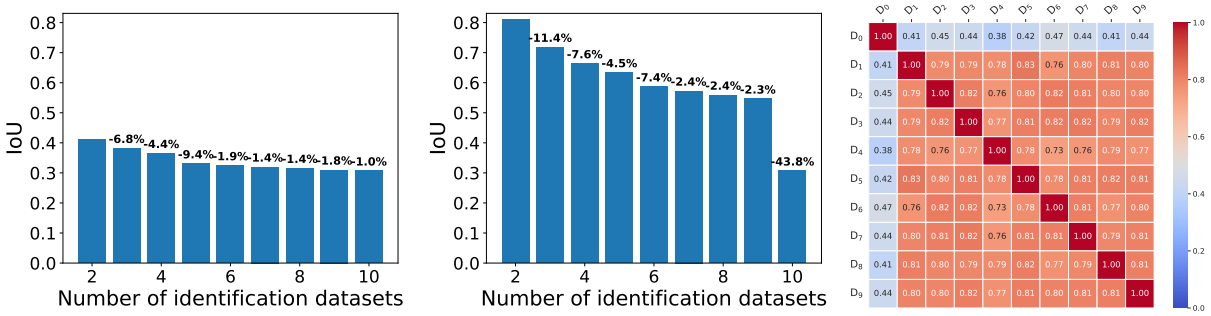
(c) Safety region overlap analysis using SNIP on Llama-2-7B-Chat. The threshold used to determine top safety-critical parameters within each weight matrix is: $q\% = 3\%$. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



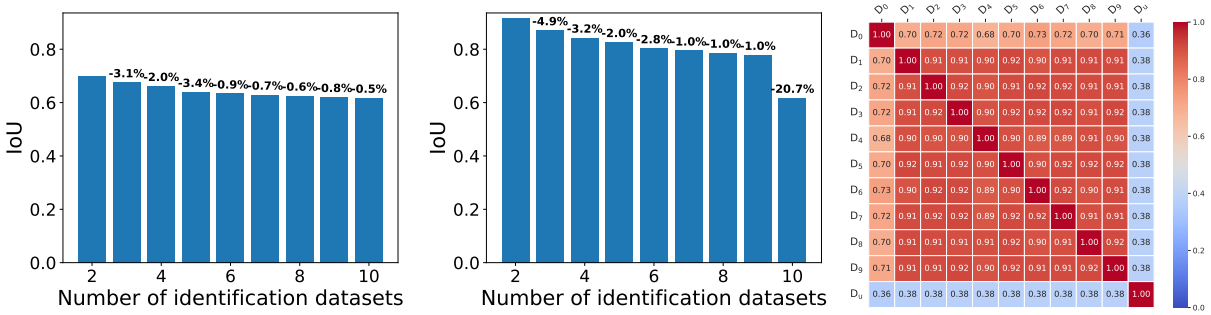
(d) Utility-isolated safety region overlap analysis using SNIP on Llama-2-7B-Chat. The thresholds used to determine top safety-critical parameters within each weight matrix are: $q\% = 3\%, p\% = 7\%$. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.



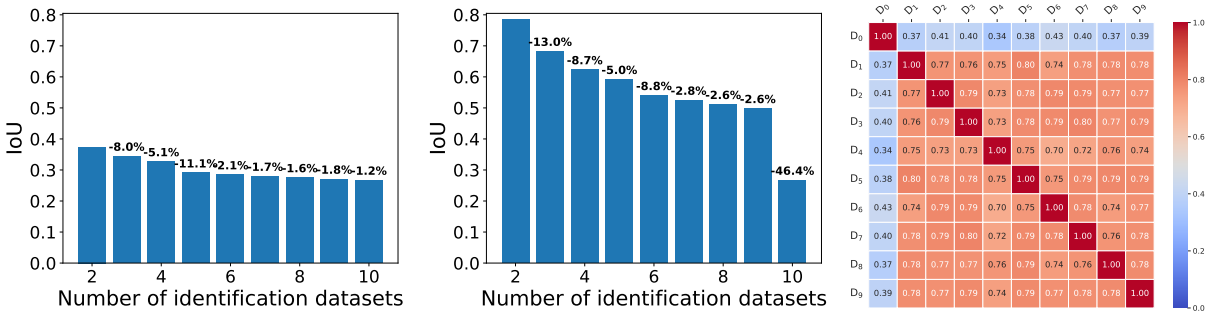
(e) Safety region overlap analysis using Wanda on Llama-2-7B-Chat. The threshold used to determine top safety-critical parameters within each weight matrix is: $q\% = 1\%$. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



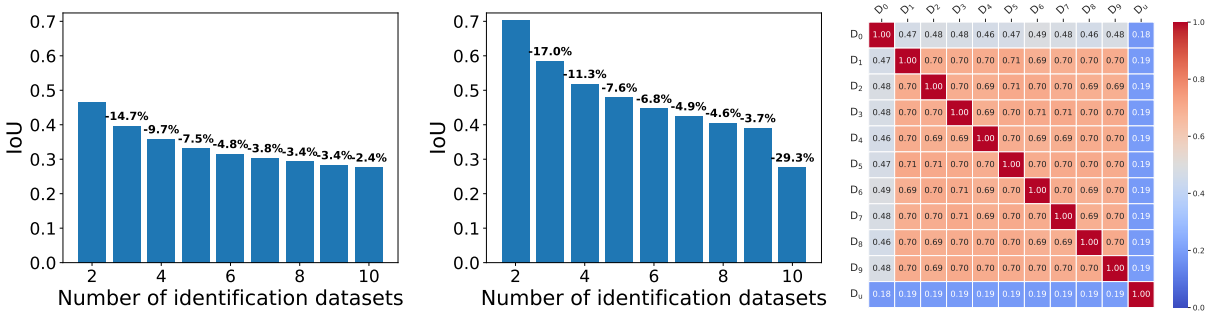
(f) Utility-isolated safety region overlap analysis using Wanda on Llama-2-7B-Chat. The thresholds used to determine top safety-critical parameters within each weight matrix are: $q\% = 1\%, p\% = 1\%$. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.



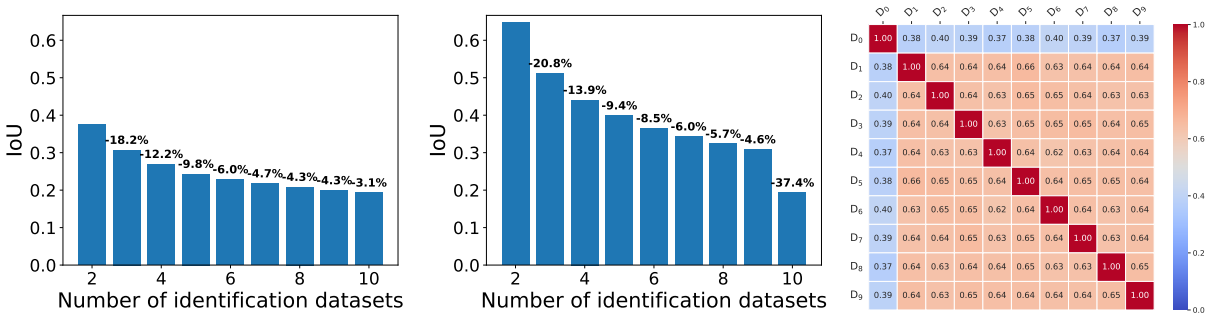
(g) Safety region overlap analysis using Wanda on Llama-2-7B-Chat. The threshold used to determine top safety-critical parameters within each weight matrix is: $q\% = 3\%$. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



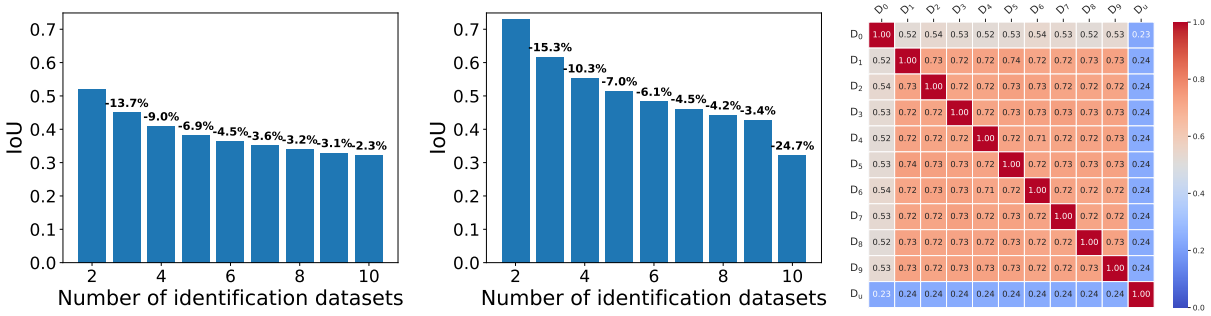
(h) Utility-isolated safety region overlap analysis using Wanda on Llama-2-7B-Chat. The thresholds used to determine top safety-critical parameters within each weight matrix are: $q\% = 3\%, p\% = 7\%$. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.



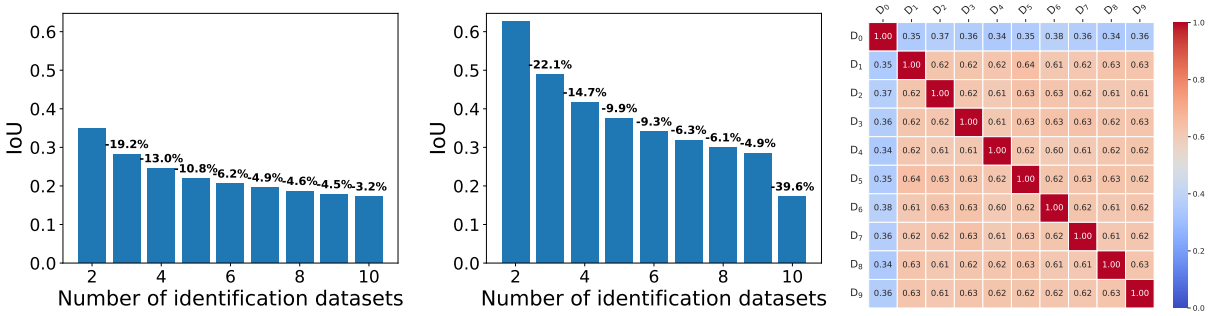
(i) Safety region overlap analysis using SNIP on Llama-2-13B-Chat. The threshold used to determine top safety-critical parameters within each weight matrix is: $q\% = 1\%$. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



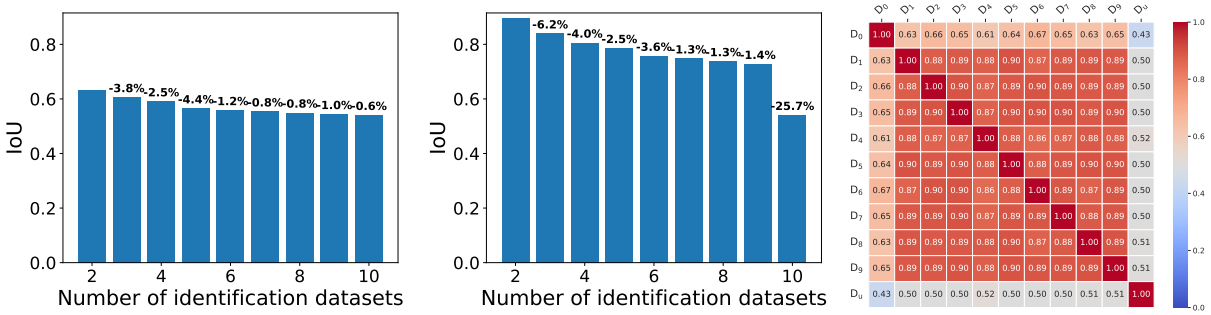
(j) Utility-isolated safety region overlap analysis using SNIP on Llama-2-13B-Chat. The thresholds used to determine top safety-critical parameters within each weight matrix are: $q\% = 1\%, p\% = 1\%$. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.



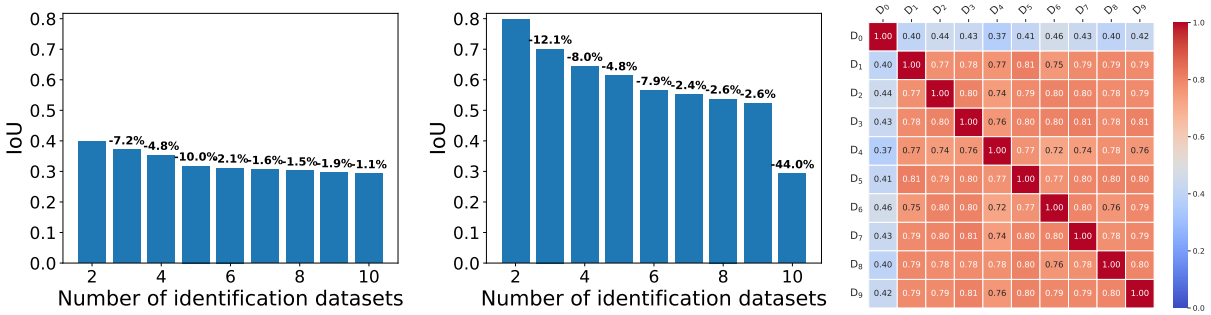
(k) Safety region overlap analysis using SNIP on Llama-2-13B-Chat. The threshold used to determine top safety-critical parameters within each weight matrix is: $q\% = 3\%$. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



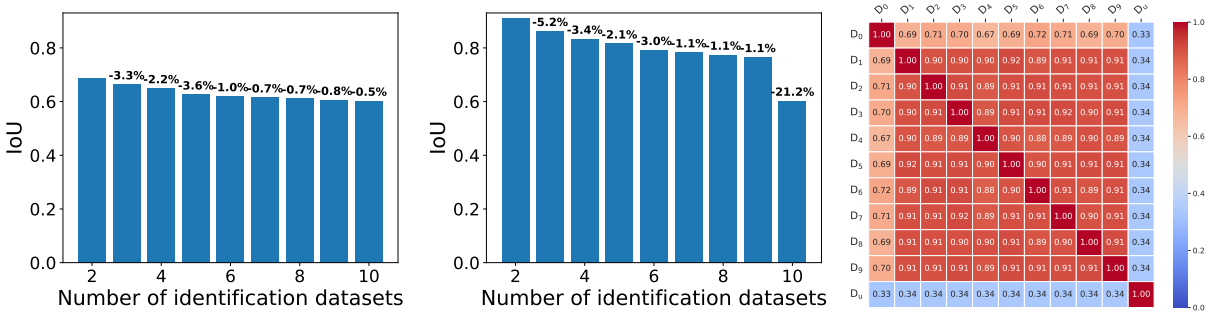
(l) Utility-isolated safety region overlap analysis using SNIP on Llama-2-13B-Chat. The thresholds used to determine top safety-critical parameters within each weight matrix are: $q\% = 3\%, p\% = 8\%$. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.



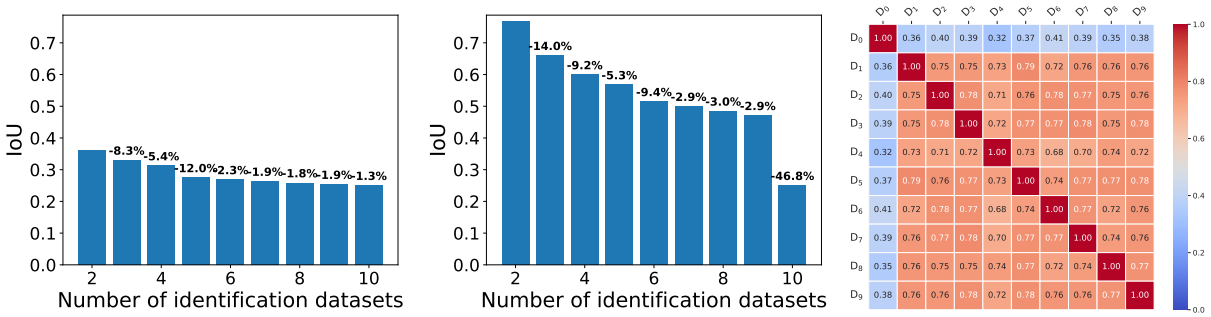
(m) Safety region overlap analysis using Wanda on Llama-2-13B-Chat. The threshold used to determine top safety-critical parameters within each weight matrix is: $q\% = 1\%$. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



(n) Utility-isolated safety region overlap analysis using Wanda on Llama-2-13B-Chat. The thresholds used to determine top safety-critical parameters within each weight matrix are: $q\% = 1\%, p\% = 1\%$. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.

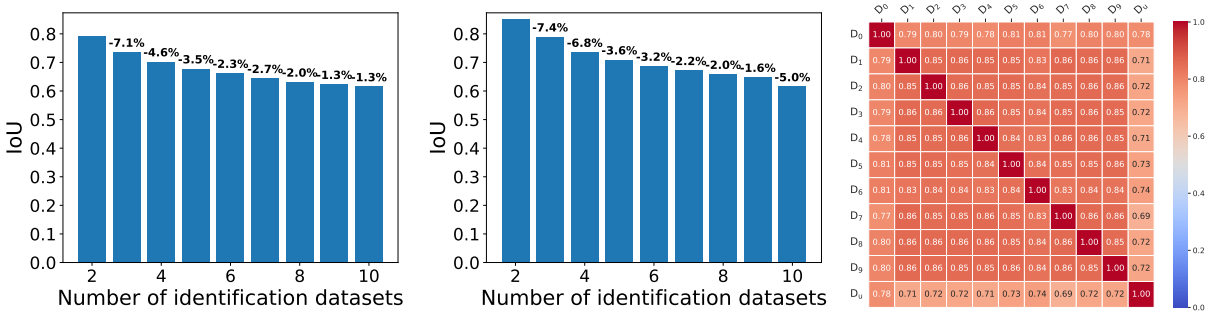


(o) Safety region overlap analysis using Wanda on Llama-2-13B-Chat. The threshold used to determine top safety-critical parameters within each weight matrix is: $q\% = 3\%$. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.

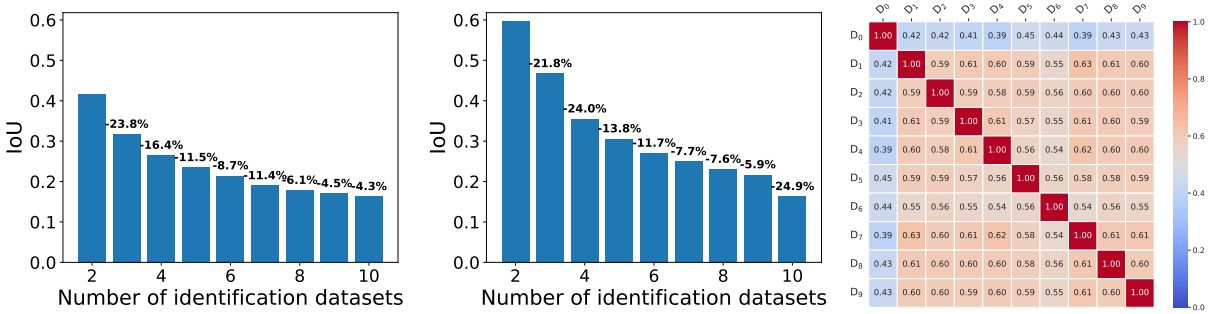


(p) Utility-isolated safety region overlap analysis using Wanda on Llama-2-13B-Chat. The thresholds used to determine top safety-critical parameters within each weight matrix are: $q\% = 3\%, p\% = 8\%$. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.

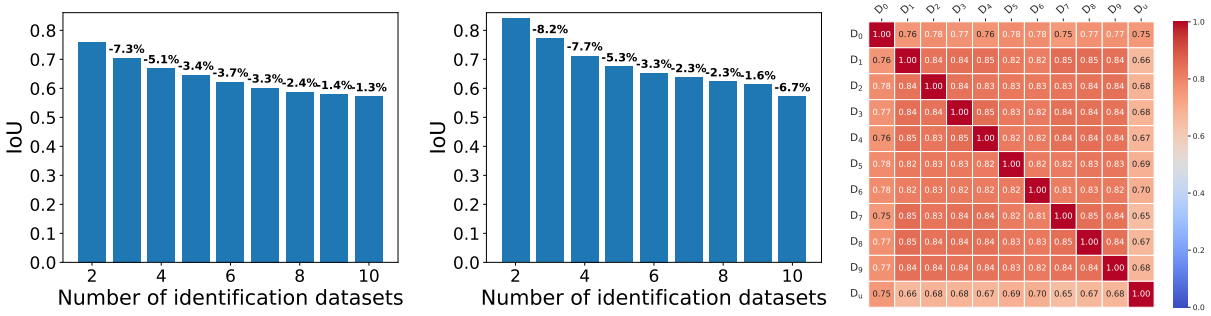
Figure 9: Safety region overlap analysis using SNIP or Wanda on Llama-2-7B-Chat and Llama-2-13B-Chat



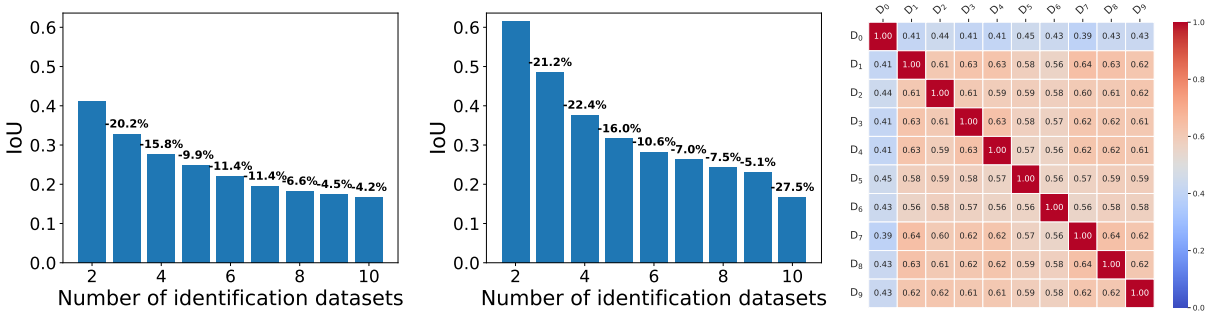
(a) Safety region overlap analysis using SafeNeuron on Llama-2-7B-Chat. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



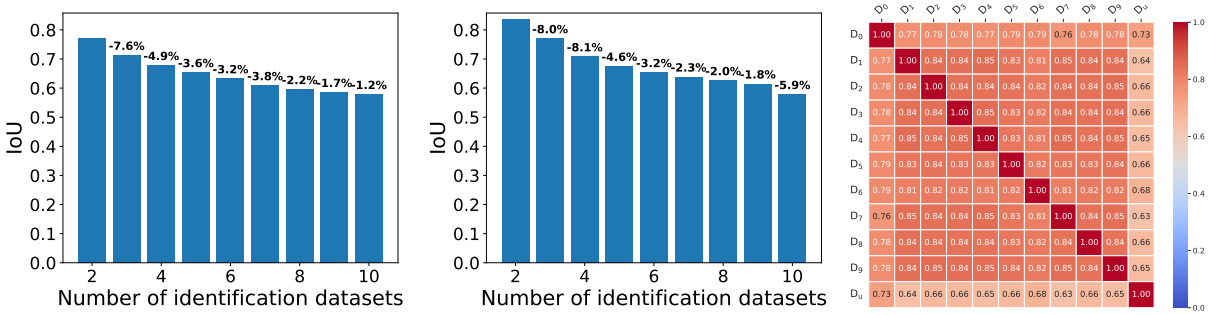
(b) Utility-isolated safety region overlap analysis using SafeNeuron on Llama-2-7B-Chat. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.



(c) Safety region overlap analysis using SafeNeuron on Mistral-7B-Instruct-v0.2. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.

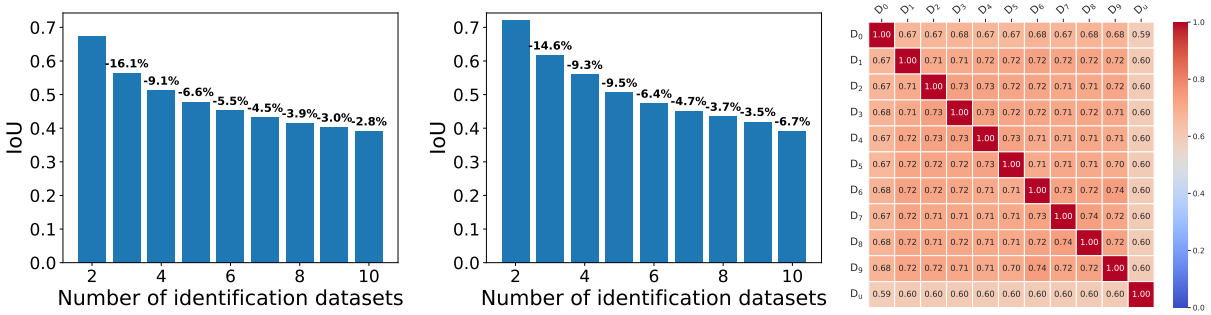


(d) Utility-isolated safety region overlap analysis using SafeNeuron on Mistral-7B-Instruct-v0.2. Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.

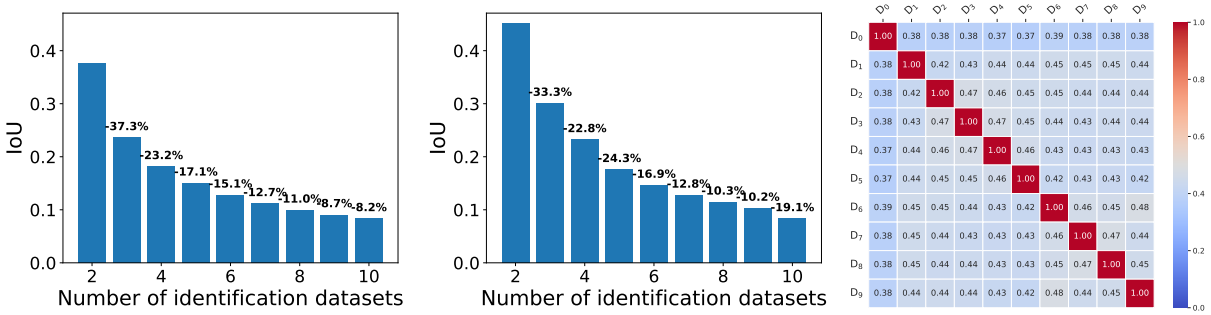


(e) Safety region overlap analysis using SafeNeuron on Llama-3-8B-Instruct. Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.

Figure 10: Safety region overlap analysis using SafeNeuron on Llama-2-7B-Chat, Llama-3-8B-Instruct and Mistral-7B-Instruct-v0.2.

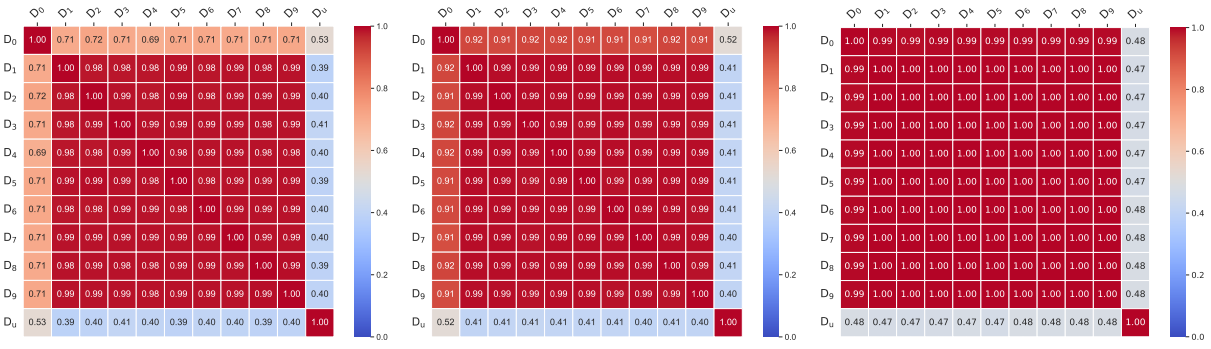


(a) Left: IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$; Middle: IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$; Right: pairwise IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two safety regions.



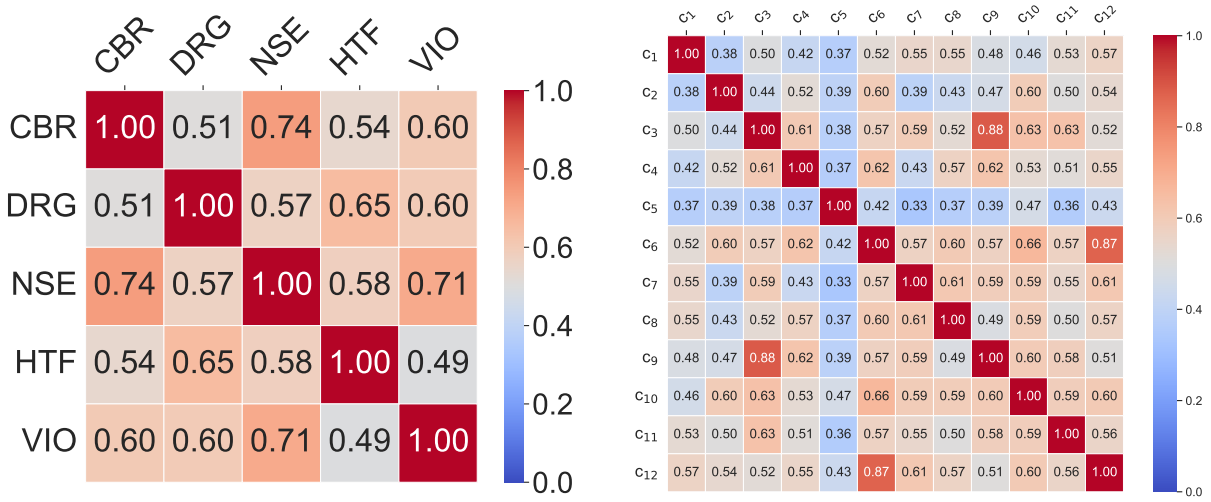
(b) Left: Iso-Utility IoU vs n (forward order). We begin with $\mathcal{D}_0$ and gradually add one dataset at a time, in the order from $\mathcal{D}_1$ to $\mathcal{D}_9$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Middle: Iso-Utility IoU vs n (backward order). We begin with $\mathcal{D}_9$ and gradually add one dataset at a time, in the order from $\mathcal{D}_8$ to $\mathcal{D}_0$. Next, we isolate each identified safety region with the utility region identified by $\mathcal{D}_u$; Right: pairwise Iso-Utility IoU for $\{\mathcal{D}_i\}_{i=0}^{9}$. The matrix is symmetric. Each element corresponds to a pairwise IoU between two utility-isolated safety regions.

Figure 11: Safety region overlap analysis using NLSR on Llama-3-8B. (a) safety region overlap analysis; (b) utility-isolated safety region overlap analysis.



(a) SNIP & Wanda (b) SafeNeuron (c) NLSR

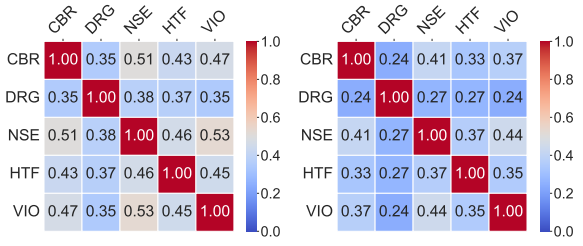
Figure 12: Pairwise cosine similarity for the multi-category datasets of SNIP & Wanda, SafeNeuron, and NLSR. For SNIP & Wanda, $\mathcal{D}_0$ was sampled using random seed equal to 0.



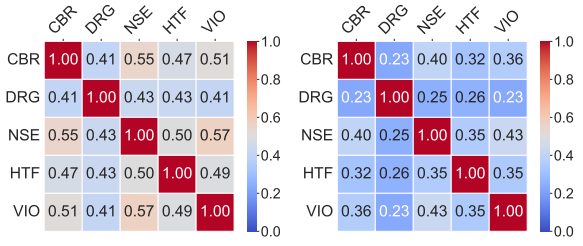
(a) SNIP & Wanda

(b) SafeNeuron

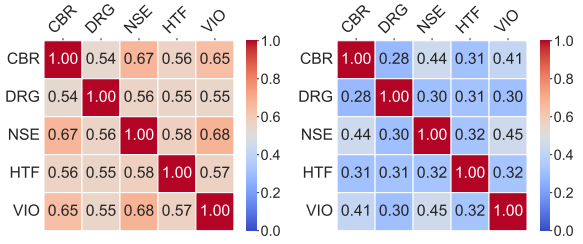
Figure 13: Pairwise cosine similarity for the single-category datasets of SNIP & Wanda, SafeNeuron. Note that they show the same matrix as the pairwise cosine similarity matrix in Fig. 3.



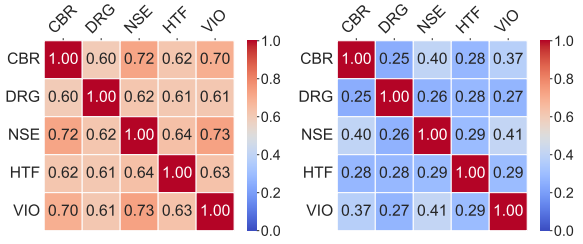
(a) Method: SNIP; Targeted model: Llama2-7B-Chat. Left: $q\% = 1\%, r = 0.53$, with p-value 0.11; right: $q\% = 1\%, p\% = 1\%, r = 0.52$, with p-value 0.12.



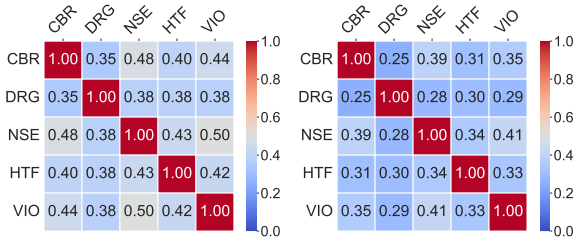
(b) Method: SNIP; Targeted model: Llama2-7B-Chat. Left: $q\% = 3\%, r = 0.55$, with p-value 0.1; right: $q\% = 3\%, p\% = 7\%, r = 0.51$, with p-value 0.13.



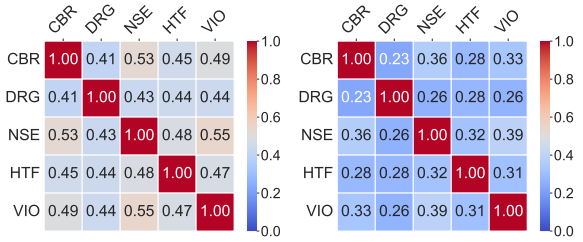
(c) Method: Wanda; Targeted model: Llama2-7B-Chat. Left: $q\% = 1\%, r = 0.75$, with p-value 0.01; right: $q\% = 1\%, p\% = 1\%, r = 0.77$, with p-value 0.01.



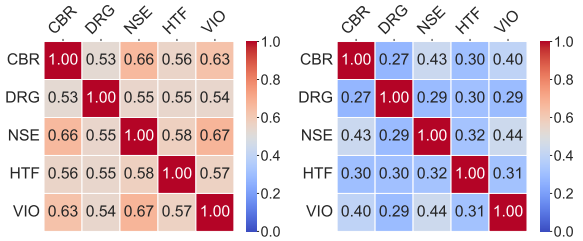
(d) Method: Wanda; Targeted model: Llama2-7B-Chat. Left: $q\% = 3\%, r = 0.74$, with p-value 0.01; right: $q\% = 3\%, p\% = 7\%, r = 0.75$, with p-value 0.01.



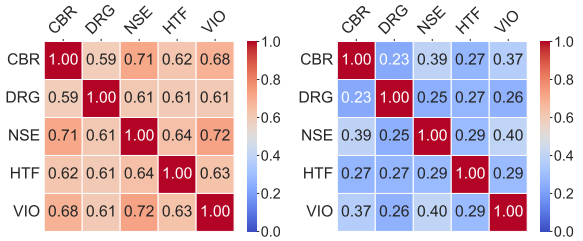
(e) Method: SNIP; Targeted model: Llama2-13B-Chat. Left: $q\% = 1\%, r = 0.67$, with p-value 0.03; right: $q\% = 1\%, p\% = 1\%, r = 0.68$, with p-value 0.03.



(f) Method: SNIP; Targeted model: Llama2-13B-Chat. Left: $q\% = 3\%, r = 0.68$, with p-value 0.03; right: $q\% = 3\%, p\% = 8\%, r = 0.68$, with p-value 0.03.

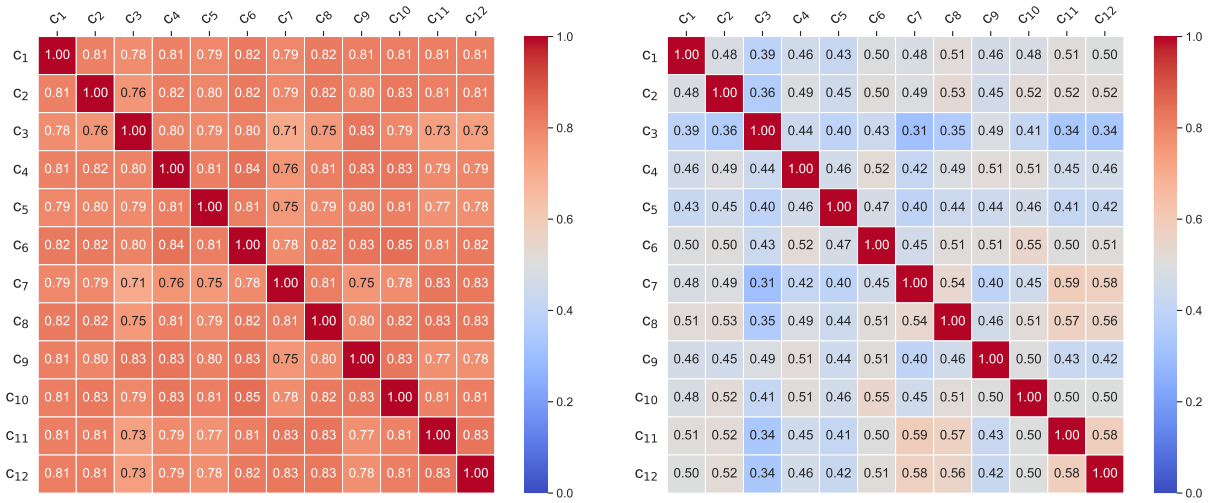


(g) Method: Wanda; Targeted model: Llama2-13B-Chat. Left: $q\% = 1\%, r = 0.74$, with p-value 0.01; right: $q\% = 1\%, p\% = 1\%, r = 0.77$, with p-value 0.01.

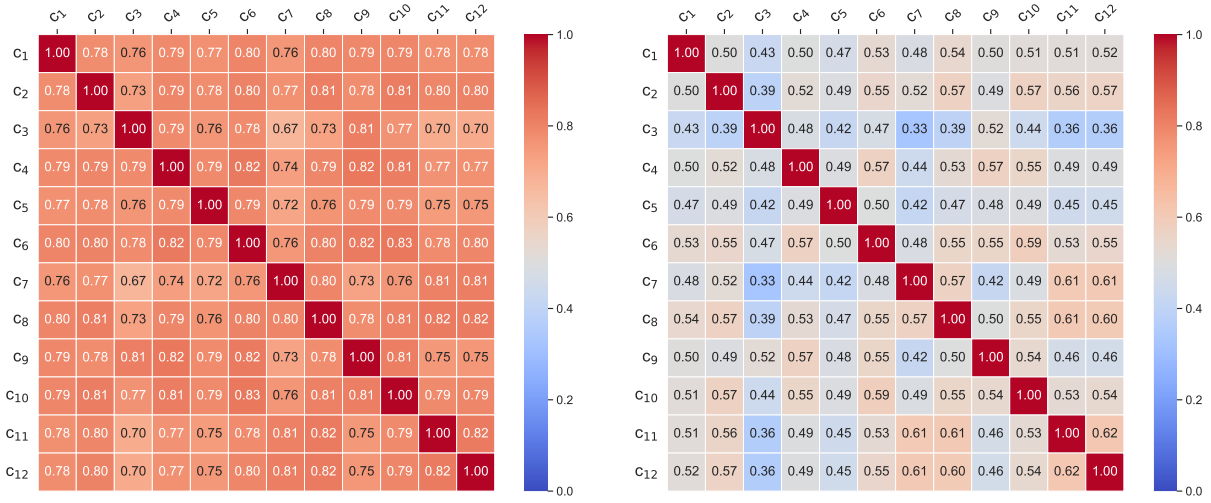


(h) Method: Wanda; Targeted model: Llama2-13B-Chat. Left: $q\% = 3\%, r = 0.74$, with p-value 0.01; right: $q\% = 3\%, p\% = 8\%, r = 0.75$, with p-value 0.01.

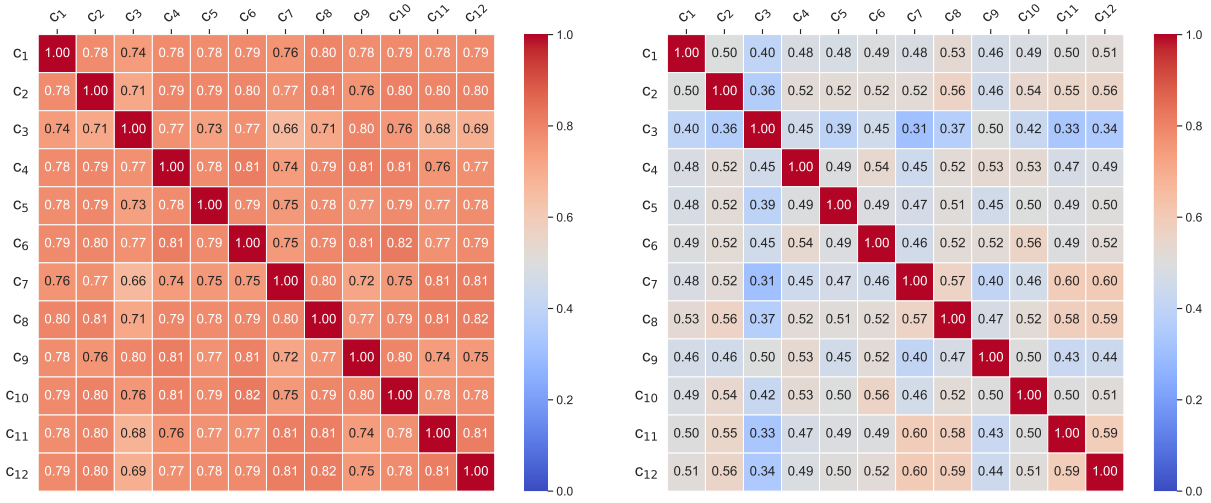
Figure 14: Pairwise safety region overlap with single-category identification datasets for SNIP & Wanda. For each subfigure, the left corresponds to the pairwise overlap between safety regions; the right corresponds to the pairwise overlap between utility-isolated safety regions. We report the Pearson correlation r between dataset cosine similarity and pairwise overlap.



(a) Llama-2-7B-Chat. Left: $r = 0.28$, with p-value 0.02; right: $r = 0.28$, with p-value 0.02.



(b) Llama-3-8B-Instruct. Left: $r = 0.28$, with p-value 0.02; right: $r = 0.28$, with p-value 0.02.



(c) Mistral-7B-Instruct-v0.2. Left: $r = 0.12$, with p-value 0.35; right: $r = 0.12$, with p-value 0.35.

Figure 15: Pairwise safety region overlap with single-category identification datasets for SafeNeuron. For each subfigure, the left corresponds to the pairwise overlap between safety regions; the right corresponds to the pairwise overlap between utility-isolated safety regions. We report the Pearson correlation r between dataset cosine similarity and pairwise overlap.