Hierarchical Safety Realignment: Lightweight Restoration of Safety in Pruned Large Vision-Language Models

Yue Li[1*], **Xin Yi**[1*], **Dongsheng Shi**[1], **Gerard de Melo**[2]
Xiaoling Wang[1], **Linlin Wang**[1†]
[1]East China Normal University
[2]Hasso Plattner Institute/University of Potsdam
`{yue_li,xinyi,dongsheng}@stu.ecnu.edu.cn`, `gdm@demelo.org`,
`{xlwang,llwang}@cs.ecnu.edu.cn`

Abstract

With the growing size of Large Vision-Language Models (LVLMs), network pruning techniques designed to compress these models for deployment in resource-constrained environments have attracted significant attention. However, we observe that pruning frequently results in a degradation in safety performance. To address this issue, we propose a novel and lightweight approach, named **H**ierarchical **S**afety **R**ealignment (**HSR**). HSR operates by first quantifying the contribution of each attention head to safety, identifying the most critical ones, and then selectively restoring neurons directly within these attention heads that play a pivotal role in maintaining safety. This process hierarchically realigns the safety of pruned LVLMs, progressing from the attention head level to the neuron level. We validate HSR across various models and pruning strategies, consistently achieving notable improvements in safety performance. To the best of our knowledge, this is the first work explicitly focused on restoring safety in LVLMs post-pruning. The code will be available at `https://github.com/TheShineyue/HSR`.

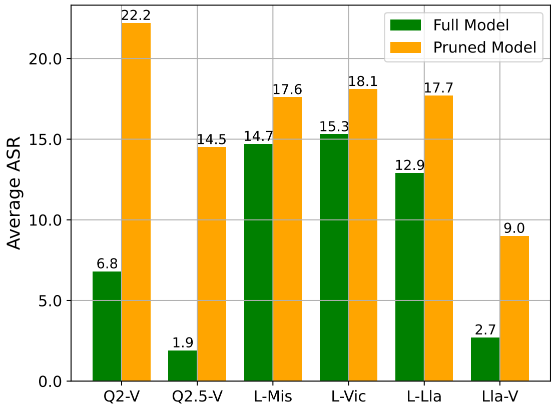


Figure 1: The Average ASR (lower values indicate higher safety) of full baselines versus pruned models (50% sparsity) across safety evaluation benchmarks. The abbreviations on the x-axis, listed from left to right, correspond to the following models: Qwen2-VL, Qwen2.5-VL, LLaVA-NeXT-Mistral, LLaVA-NeXT-Vicuna, LLaVA-NeXT-Llama3, and Llama3.2-Vision.

1 Introduction

Large Language Models (LLMs) benefit from their massive parameter count and advanced architectures, achieving outstanding results on diverse benchmarks. Building on this success, efforts to extend LLMs into multimodal domains have made remarkable progress as well. Most current Large Vision-Language Models (LVLMs), typically composed of visual encoders, adapters, and LLM backbones (Liu et al., 2024a), have a large parameter scale and leverage image-text datasets to achieve effective multimodal alignment. To enable model deployment and application under resource-constrained environments, pruning methods (Sun et al., 2024; Frantar and Alistarh, 2023; Lee et al., 2018) compute importance scores of neurons to eliminate those deemed less important, thereby reducing the size of the model while retaining the utility to a certain extent. Such methods have seen broad adoption to boost model efficiency.

Safety is commonly defined as preventing models from following malicious instructions and generating toxic content (Bianchi et al., 2024). Recent work (Zhou et al., 2025) has revealed that certain safety heads within the attention mechanism are crucial for feature integration in safety-related tasks. Additionally, neural-level research (Wei et al., 2024) has found that certain regions within the model are critical for safety guardrails, which are separate from the utility-related regions and exhibit sparsity. A natural concern arises from the fact that, because these regions contribute minimally to utility, they are prone to being removed by pruning technologies that prioritize utility im-

---

[*] Equal Contribution.
[†] Corresponding Author.

Findings of the Association for Computational Linguistics: ACL 2025, pages 7600–7612
July 27 - August 1, 2025 ©2025 Association for Computational Linguistics

portance as a pruning metric. This removal could result in a decline in the safety of the pruned model. To verify whether this problem exists, we used the Wanda pruning method (Sun et al., 2024) to prune six mainstream LVLMs and compared their safety changes before and after pruning. The experimental results shown in Figure 1 reveal that all LVLMs exhibited varying degrees of safety degradation, with the worst-performing model showing a 15.4% safety drop and the best-performing case exhibiting a 2.8% decline. Despite the serious safety risks of pruning technologies, research on model safety restoration after pruning remains scarce.

To address this problem, we propose a novel **H**ierarchical **S**afety **R**ealignment (**HSR**) approach, designed to restore the safety performance degraded by pruning, without significantly increasing the pruned model's parameter size. Our HSR approach hierarchically realigns the safety of the pruned model from attention head level to neuron level. HSR operates in two main steps: First, we evaluate each attention head's contribution to model safety and identify the safety-critical heads with the greatest impact. Subsequently, for these key heads, we pinpoint and restore safety-critical neurons that were pruned, effectively realigning the safety of our model.

We have validated the effectiveness of our approach across various models and pruning techniques. Our proposed HSR approach successfully realigns the safety of pruned models, restoring over 27% of the lost safety in many cases and more than 14% even in the worst-case scenarios, all with lightweight modifications. Furthermore, with extensive analysis and ablation experiments, we have uncovered several key insights into model safety. These include the finding that a small subset of neurons plays a disproportionately significant role in ensuring safety, and the observation that certain neurons exist which negatively impact safety.

In summary, our contributions are the follows:

- We propose a novel method named Hierarchical Safety Realignment (HSR) to realign the safety of pruned LVLMs, yielding substantial safety improvements with lightweight modifications. To our knowledge, HSR is the first method specifically designed to address the safety realignment of pruned LVLMs.
- Our findings reveal that a small subset of neurons plays a disproportionately large role in ensuring safety, while certain neurons negatively impact safety. By selectively restoring these safety-critical neurons, we can achieve significant safety recovery in pruned models.
- We validate the proposed method on various LVLMs using different pruning techniques. Extensive experiments demonstrate the superiority of our approach, consistently improving performance with minimal neuron restoration.

2 Method

Overview In this section, we present our Hierarchical Safety Realignment method in detail and illustrate the core process in Figure 2. Our approach achieves safety realignment of the pruned model by hierarchically identifying and restoring safety-critical neurons, starting at the attention head level and progressing to the neuron level: At the **attention head** level (Section 2.1), each attention head is individually masked to measure changes in the model's output for malicious instructions compared to the original. The attention heads causing the most significant changes, termed as the **safety-critical** heads, are selected for further analysis. At the **neuron** level (Section 2.2), we compute two importance scores for neurons: the safety importance score based on a safety dataset and the utility importance score derived from a utility dataset (details provided in the caption of Figure 2). Pruned neurons exhibiting high safety importance along with sufficient utility importance are identified as **safety-critical** neurons and subsequently restored.

2.1 Identifying Safety-Critical Heads

To preserve model sparsity, we selectively restore neurons only in those attention heads that exhibit the highest safety-critical importance. Zhou et al.'s (2025) work proposed a new metric tailored for multi-head attention, namely the **S**afety **h**ead **imp**ortance **s**core (**Ships**), to evaluate the contribution of each head to the model safety. Specifically, for a specific harmful data $q_{\mathcal{H}}$, the probability distribution of the original model $\theta_{\mathcal{O}}$ is denoted by $p(q_{\mathcal{H}};\theta_{\mathcal{O}})$. For the i-th attention head h_i^l in the l-th layer, its contribution to the safety of the model is eliminated by multiplying its Query, Key, and Value matrices by a very small coefficient ϵ. The probability distribution of the model after such ablation is denoted by $p(q_{\mathcal{H}};\theta_{\mathcal{O}} \setminus \theta_{h_i^l})$. The Ships$(q_{\mathcal{H}},\theta_{h_i^l})$ are calculated as the KL divergence (Kullback and Leibler, 1951) of $p(q_{\mathcal{H}};\theta_{\mathcal{O}})$

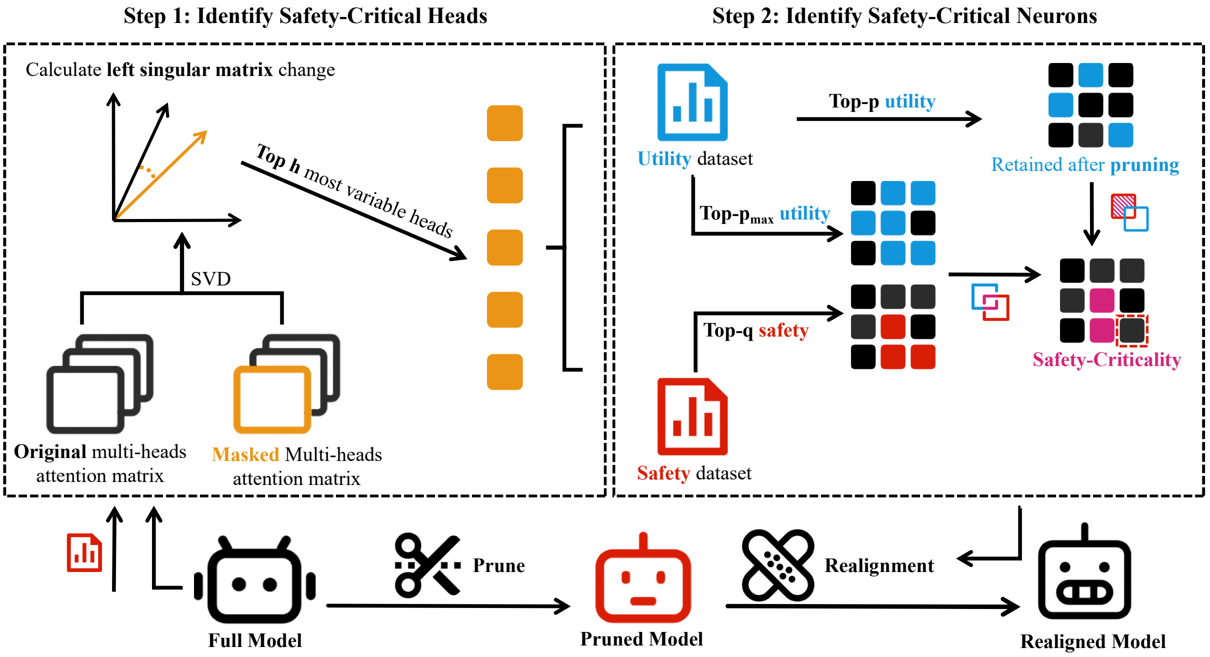

Step 1: Identify Safety-Critical Heads
Step 2: Identify Safety-Critical Neurons
Calculate **left singular matrix** change
Top h most variable heads
SVD
Original multi-heads attention matrix
Masked Multi-heads attention matrix
Utility dataset
Top-p utility
Retained after pruning
Top-p_max utility
Top-q safety
Safety-Criticality
Safety dataset
Prune
Realignment
Full Model
Pruned Model
Realigned Model

Figure 2: HSR hierarchically achieves safety realignment of the pruned model in two steps: The first step identifies the top-h most important attention heads for safety, while the second one identifies and restore the safety-critical neurons on these heads. The safety dataset, comprising malicious instructions and appropriate rejection responses, is marked in red, and the utility dataset, which excludes malicious instructions, is marked in blue.

and $p(q_{\mathcal{H}}; \theta_{\mathcal{O}} \setminus \theta_{h_i^l})$ as follows:

$$\mathbb{D}_{KL}\left(p(q_{\mathcal{H}}; \theta_{\mathcal{O}}) \parallel p(q_{\mathcal{H}}; \theta_{\mathcal{O}} \setminus \theta_{h_i^l})\right) \quad (1)$$

It quantifies the impact of ablating head h_i^l for $q_{\mathcal{H}}$, which is the safety contribution of h_i^l. Given that most contemporary LVLMs employ Group Query Attention (Ainslie et al., 2023) to reduce computational overhead, we derive generalized masking equations. For the query and key matrices $\mathbf{W}_q$ and $\mathbf{W}_k$, the modified head h_i^m calculation becomes:

$$h_i^m = \text{Softmax}\left(\frac{\epsilon \mathbf{W}_q^i \mathbf{W}_k^{i/g^T}}{\sqrt{d_k/n}}\right)\mathbf{W}_v^{i/g}, \quad (2)$$

whereas for the value matrix $\mathbf{W}_v$, the calculation is adjusted as:

$$h_i^m = \text{Softmax}\left(\frac{\mathbf{W}_q^i \mathbf{W}_k^{i/g^T}}{\sqrt{d_k/n}}\right)\epsilon \mathbf{W}_v^{i/g}, \quad (3)$$

Here, n denotes the number of attention heads per layer, and g denotes the query amount of each group calculated as $g = n/n_{kv}$ where n_{kv} indicates the number of key-value head pairs.

For a given dataset D, we aggregate network activations into matrix $\mathbf{X}$, and perform singular value decomposition (SVD): $\text{SVD}(\mathbf{X}) = \mathbf{U}\Sigma\mathbf{V}^\top$, where $\mathbf{U}$ represents the key features in the dataset space. Through this decomposition, we derive two critical matrices: $\mathbf{U}_\theta$ (left feature matrix from the original model) and $\mathbf{U}_A$ (left feature matrix from the ablated model). The safety representation divergence is quantified using the $\text{Ships}(D, h_i^l)$ metric:

$$\text{Ships}(D, h_i^l) = \sum_{r=1}^{r_{\max}} \cos^{-1}\left(\sigma_r(U_\theta^{(r)}, U_A^{(r)})\right) \quad (4)$$

where σ_r denotes the r-th singular value. A larger main angle indicates that the safety representation has changed significantly, which represents the safety importance at the dataset level. We subsequently identify the top-h attention heads with maximal safety contributions, designated as **safety-critical** heads, for neuron-level attribution analysis.

2.2 Identifying Safety-critical Neurons

2.2.1 Quantifying Neuron Importance

We proceed to identify pruned neurons that remain critical for safety considerations. Given a calibration dataset, the pruning method calculates importance scores of weights to attribute their impact on the model's relative performance. When provided with a safety dataset or a utility dataset, the method quantifies the safety importance scores and utility importance scores of the weights, respectively. We provide three variants that use different approaches to quantify neuron importance as follows:

- For a given calibration dataset, we use Wanda Score (Sun et al., 2024) to calculate the importance score of a weight using the absolute value of its weight matrix and the ℓ_2 norm of the input activations. Subsequently, we follow Wei et al., 2024 to mask the rest of the calibration dataset and focusing only on the response activation, and tore all activations for layer W into $\mathbf{X}_{\text{in}}$ of shape (n, C_{in}) and calculate the importance score $\mathbf{I}$ as:

  $$\mathbf{I} = |\mathbf{W}| \odot \left(\mathbf{1} \cdot \|\mathbf{X}_{\text{in}}\|_2^\top\right), \tag{5}$$

  where $|\mathbf{W}|$ is a weight matrix of a linear layer of shape $(C_{\text{out}}, C_{\text{in}})$. $\mathbf{1}$ denotes an all-one vector of shape $(C_{\text{out}}, 1)$. We compute the row-wise ℓ_2 norms of $\mathbf{X}_{\text{in}}$, and then transpose them to obtain a matrix of shape $(1, C_{\text{in}})$.

- We use SparseGPT Score (Frantar and Alistarh, 2023) to obtain the importance $\mathbf{I}$ as Eq. 6, where $\mathbf{X}_{\text{in}}$ contains only response activations:

  $$\mathbf{I} = \left[\frac{|\mathbf{W}|^2}{\text{diag}\left((\mathbf{X}_{\text{in}}^\top \mathbf{X}_{\text{in}} + \lambda \mathbf{I})^{-1}\right)}\right] \tag{6}$$

  Here, $\mathbf{X}_{\text{in}}^T \mathbf{X}_{\text{in}} + \lambda \mathbf{I}$ in the denominator is the Hessian $\mathbf{H}$ for the layer-wise reconstruction problem and λ is the Hessian dampening factor to avoid the collapse of inverse computation. Once $\mathbf{I}$ is calculated, SparseGPT updates the weights by masking less important portions based on the desired sparsity.

- We finally introduce the third method, which is based on the SNIP Score (Lee et al., 2019). For a data instance $x = (x_{\text{prompt}}, x_{\text{response}})$, we define the corresponding loss function as the conditional negative log-likelihood $\mathcal{L}(x) = -\log p(x_{\text{response}} \mid x_{\text{prompt}})$. For a weight matrix $\mathbf{W}$, we use SNIP Score to calculate its importance score $\mathbf{I}$ as follows:

  $$\mathbf{I}(\mathbf{W}_{ij}, x) = |\mathbf{W}_{ij} \cdot \nabla_{\mathbf{W}_{ij}} \mathcal{L}(x)|, \tag{7}$$

  This equation is the first-order Taylor approximation to the change of the loss when the weight entry $\mathbf{W}_{ij}$ is set to zero.

  Following the experimental setup described by Wei et al. (2024), for a given calibration dataset D, we use

  $$\mathbf{I} = \mathbb{E}_{x \sim D} \mathbf{I}(\mathbf{W}, x) = \mathbb{E}_{x \sim D} |\mathbf{W} \odot \nabla_{\mathbf{W}} \mathcal{L}(x)|. \tag{8}$$

2.2.2 Safety-Critical Neuron Restoration

Considering two different calibration datasets, a safety dataset D^s and a utility dataset D^u. D^s comprises instructions and images that contain harmful information, along with responses that correctly refuse such information. In contrast, D^u consists of safe instructions and images paired with reasonable responses. Therefore, the safety importance score $\mathbf{I}^s$ and utility importance score $\mathbf{I}^u$ can be calculated respectively using the Wanda, SparseGPT or SNIP method in Section 2.2.1.

We select those weights with larger importance scores and consider them weights that contribute more to safety or utility. Specifically, given hyper-parameters q and p for safety and utility, respectively, we use Equations 9 and 10 to obtain the safety importance set S^s and utility importance set S^u of the i-th layer.

$$S^s(\text{q}) = \{(i, j) \mid \mathbf{I}^u_{i,j} \text{ is the top q\% of } \mathbf{I}^s_i\} \tag{9}$$

$$S^u(\text{p}) = \{(i, j) \mid \mathbf{I}^u_{i,j} \text{ is the top p\% of } \mathbf{I}^u_i\} \tag{10}$$

For the pruning process, the weights outside the utility important set (here p = 1 - sparsity ratio) will be set to 0 according to the set sparsity, thus obtaining a sparse neural network.

Among the pruned neurons (not in S^u(p)), we seek those safety-critical neurons that have high safety and still have certain utility, and will not cause excessive loss of model utility in the subsequent realignment process. Therefore, we introduce the hyper-parameter p_{max} (p_{max} is greater than p) and obtain the safety-critical neurons $S(\text{p}, \text{q}, \text{p}_{\text{max}})$ as follows:

$$S(\text{p}, \text{q}, \text{p}_{\text{max}}) = (S^s(\text{q}) \cap S^u(\text{p}_{\text{max}})) - S^u(\text{p}). \tag{11}$$

We restore these pruned safety-critical neurons on the pruned model to realign the model's safety.

3 Experimental Setup

Dataset During the realignment phase, we employ two distinct data subsets: (1) Safe-Safe pairs (safe images with corresponding safe instructions) as the utility dataset, and (2) Unsafe-Unsafe pairs

Method	Safety ↓				Utility ↑			Restoration
	SafeBench	Ch^3Ef	AVG	RSR	MMbench	DocVQA	AVG	
Full Model	1.40	2.35	1.88	-	87.02	94.51	90.76	-
SNIP	4.60	8.12	6.36	-	84.55	**92.93**	88.74	-
w/ HSR(Ours)	**3.00**	**5.34**	**4.17**	48.88%	**84.62**	92.90	**88.76**	0.150‱
Wanda	11.20	17.74	14.47	-	**85.15**	91.97	88.56	-
w/ HSR(Ours)	**9.00**	**13.03**	**11.02**	27.40%	85.01	**92.13**	**88.57**	0.020‱
SparseGPT	3.00	3.21	3.10	-	**83.88**	**90.64**	**87.26**	-
w/ HSR(Ours)	**2.80**	**2.56**	**2.68**	34.43%	**83.88**	90.63	87.25	0.133‱

Table 1: The safety and utility values of Qwen2.5-VL under different pruning methods are shown. Here the Restoration Indicates the ratio of the additional restored parameters to the total parameters that need to be pruned, in ten thousandths. The better value for each group is shown in **bold**.

(unsafe images with matching unsafe instructions), which constitute the safety dataset. Both subsets are derived from the VLGuard (Zong et al., 2024) training dataset.

During the evaluation phase, we employ the following benchmarks: (1) For utility assessment, MMbench (Liu et al., 2025) and DocVQA (Mathew et al., 2021) are utilized; (2) For safety evaluation, safebench-mini (Ying et al., 2024a) and the harmful subset of Ch^3Ef (Shi et al., 2024) are adopted. To ensure fair comparison and reproducibility, all evaluations are conducted under strict zero-shot settings with greedy decoding strategies.

Models for Pruning Our experiments involved six mainstream LVLMs, including three variants of LLaVA-NeXT[1], built on different language models: Vicuna, Mistral, and Llama3. Additionally, we evaluated Qwen2.5-VL[2], Qwen2-VL (Wang et al., 2024a), and Llama-3.2-Vision[3]. All models have parameter counts ranging from 7B to 11B.

Evaluation Metrics Three evaluation metrics we use are Attack Success Rate (ASR), Average Normalized Levenshtein Similarity (ANLS) and Accuracy (Acc). ASR is used to evaluate the safety of the model. The smaller the ASR, the better the safety. We use Llama-Guard-3-Vision[4] to determine whether the response is safe. We use ANLS and Acc as the evaluation indicators of DocVQA and MMbench respectively. Following Mathew et al.'s (2021) proposal, ANLS can ensure that minor answer mismatches stemming from OCR errors are not severely penalized. Additionally, in order to more intuitively and fairly reflect the performance improvement of our method, we provide the **R**atio of model **S**afety **R**ealignment (**RSR**) as follows:

$$\mathrm{RSR} = \frac{\mathrm{ASR}_{\mathrm{Pruned}} - \mathrm{ASR}_{\mathrm{Pruned\ w/\ HSR}}}{\mathrm{ASR}_{\mathrm{Pruned}} - \mathrm{ASR}_{\mathrm{Full}}} \tag{12}$$

to quantify the ratio of restored safety to lost safety. More relevant details are reported in Appendix A.

4 Experimental Results

Comparison of different pruning methods We present the performance of Qwen2.5-VL at 50% sparsity using various pruning methods realigned with HSR, as shown in Table 1. HSR effectively realigns the safety of the pruned model and only requires restoring a minimal number of safety-critical neurons. Specifically for Qwen2.5-VL, the average value of ASR decreases by 2.19%, 3.45%, and 0.42%, respectively, for SNIP, Wanda, and SparseGPT. Judging from the ratio of restored safety to lost safety, it is generally possible to recover over 27% of the safety capacity. In addition, there is no significant loss in utility. In fact, we even find a slight improvement in multiple cases and in the average value, which may stem from the fact that the restored neurons also have a certain utility contribution for Qwen2.5-VL.

Comparison of different LVLMs We report the performance of various LVLMs at 50% sparsity using Wanda and realigning via HSR, which is shown in Table 2. HSR demonstrates significant improvements across pruned models: Qwen2-VL, LLaVA-NeXT-Vicuna, and LLaVA-NeXT-Mistral show average ASR reductions of 5.46%, 3.03%, and 1.03% respectively, with restoration rates exceeding 35% (LLaVA-NeXT-Mistral achieves over 100%). For

---

[1] `https://llava-vl.github.io/blog/2024-05-10-llava-next-stronger-llms/`

[2] `https://huggingface.co/Qwen/Qwen2.5-VL-7B-Instruct`

[3] `https://huggingface.co/meta-llama/Llama-3.2-11B-Vision-Instruct`

[4] `https://huggingface.co/meta-llama/Llama-Guard-3-11B-Vision`

Method	Safety ↓				Utility ↑			Restoration
	SafeBench	Ch^3Ef	AVG	RSR	MMbench	DocVQA	AVG	
Qwen2-VL	5.00	8.55	6.77	-	82.70	89.14	85.92	-
Wanda	21.40	23.08	22.24	-	75.93	76.27	76.10	-
w/ HSR (Ours)	**15.40**	**18.16**	**16.78**	35.29%	**76.69**	**77.93**	**77.31**	0.016‱
LLaVA-NeXT-Mistral	11.00	18.38	14.69	-	76.69	63.74	70.21	-
Wanda	13.40	21.79	17.60	-	**73.11**	**57.22**	**65.17**	-
w/ HSR (Ours)	**11.20**	**17.95**	**14.57**	104.12%	72.95	56.74	64.85	0.385‱
LLaVA-NeXT-Vicuna	11.80	18.80	15.30	-	75.21	66.91	71.06	-
Wanda	13.60	22.65	18.12	-	**69.62**	**60.33**	**64.98**	-
w/ HSR (Ours)	**12.60**	**21.58**	**17.09**	36.52%	69.21	60.10	64.65	1.803‱
LLaVA-NeXT-Llama3	8.60	17.09	12.85	-	79.42	72.42	75.92	-
Wanda	10.20	25.21	17.71	-	**74.96**	66.35	**70.66**	-
w/ HSR (Ours)	**9.20**	**24.79**	**16.99**	14.81%	74.57	**66.38**	70.47	0.799‱
Llama3.2-Vision	2.60	2.78	2.69	-	75.44	77.44	76.44	-
Wanda	9.20	8.76	8.98	-	**69.07**	**65.71**	**67.39**	-
w/ HSR (Ours)	**8.60**	**7.26**	**7.93**	16.69%	66.85	64.04	65.45	0.065‱

Table 2: The safety and utility values of Wanda and HSR realigned Wanda pruned models for different LVLMs are shown. The better value for each LVLM is shown in **bold**.

Method	Safety ↓				Utility ↑			Restoration
	SafeBench	Ch^3Ef	AVG	RSR	MMbench	DocVQA	AVG	
Qwen2.5-VL	1.40	2.35	1.88	-	87.02	94.51	90.76	-
Wanda 2:4	14.40	13.46	13.93	-	80.20	87.94	84.07	-
w/ HSR (Ours)	**12.60**	**10.26**	**11.43**	20.75%	**80.99**	**89.07**	**85.03**	0.055‱
Qwen2-VL	5.00	8.55	6.77	-	82.70	89.14	85.92	-
Wanda 2:4	27.00	19.87	23.44	-	63.94	50.86	57.40	-
w/ HSR (Ours)	**23.20**	**16.67**	**19.93**	21.06%	**69.81**	**55.82**	**62.82**	0.047‱

Table 3: The safety and utility values of 2:4 structured Wanda and HSR realigned Wanda pruned models for different LVLMs are shown. The better value for each LVLM is shown in **bold**.

LLaVA-NeXT-llama3 and Llama3.2-Vision, the average ASR decreases by 0.72% and 1.05%, with a safety restoration ratio slightly above 14%. These results demonstrate notable improvements, yet the restoration performance of Llama3-based LVLMs remains constrained.

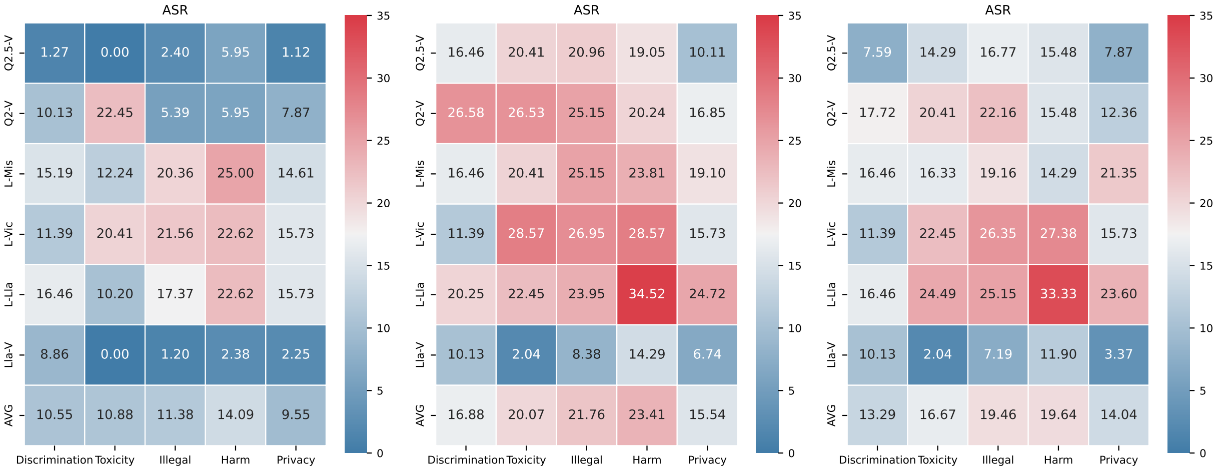
Regarding utility, both Qwen2-VL and the aforementioned Qwen2.5-VL show improved performance, while other models experience slight declines. This difference may be attributed to the Qwen series employing grouped query attention, which features the largest number of queries per group and the fewest heads (as shown in Appendix B), making each neuron's contribution (both utility and safety) particularly significant.

Regarding the neuron restoration ratio, for LLaVA-NeXT-Vicuna, it is significantly higher than for others. This may be because other models are based on the group attention mechanism, and the safety realignment brought by the restoration can affect a wider range.

Comparison with structured pruning For 2:4 structured pruning (retaining 2 of every 4 neurons), we apply Wanda and Wanda with HSR to Qwen2-VL and Qwen2.5-VL (Table 3). HSR remains effective, reducing average ASR by 2.5% and 3.51% respectively, while maintaining utility improvements. Notably, structured pruning underperforms unstructured pruning in safety metrics (ASR reduction and safety restoration ratio), likely due to its inherent limitations: mandatory retention patterns may exclude high-utility neurons while incidentally preserving safety-critical ones.

Comparison by category We report the effect of HSR on the safety realignment of the pruned model for various categories in the Ch^3Ef dataset in Figure 3. The comparison between pruned (Figure 3b) and realigned models (Figure 3c) reveals consistent improvements: Discrimination (3.59%), Toxicity (3.4%), and Harm (3.77%) show significant ASR reductions, while the remaining categories exhibit $<2.3\%$ decreases.

Effect of different sparsities An experimental comparison is given in Table 4. Regarding safety performance, varying degrees of safety realignment



(a) Full Model ↓ (b) Pruned by Wanda ↓ (c) Realigned by HSR ↓

Figure 3: Results after pruning with 50% sparsity using Wanda and HSR realignment, with the classification here coming from Ch^3Ef (Shi et al., 2024). The first six rows are the abbreviations of the LVLMs; see the caption of Figure 1 for details. The bottom row gives the average ASR of the six LVLMs for the category.

are observed across different sparsity levels, with the most significant restoration occurring at 50% sparsity. This phenomenon could be attributed to two factors: at 40% sparsity, the structural damage remained moderate with limited improvement potential, while the 60% sparsity level suffered excessive parameter loss that marginally diminished the benefits achievable through low-level adjustments.

Moreover, the utility of realigned models exhibits a slight reduction at 40% sparsity while showing progressive improvement at 50%~60% sparsity levels, peaking at 60% sparsity. This phenomenon may stem from safety-critical neurons possessing greater safety significance than utility value, whereas in heavily pruned models (50% ~60% sparsity) with substantial utility degradation, the utility of the safety-critical neurons can still bring about some improvement.

Sparsity	**40%**		**50%**		**60%**	
	Safety↓	Utility↑	Safety↓	Utility↑	Safety↓	Utility↑
Wanda	10.69	82.79	22.24	76.10	27.05	48.17
w/ HSR	10.01	82.65	16.78	77.31	25.61	63.37

Table 4: Effect of sparsities on Qwen2-VL with pruning by Wanda and realignment by HSR. We report the average scores of safety and utility.

Hyperparameter Effects We analyze the effects of varying the hyperparameters q, $p_{\max}$ and h on model performance. Table 5 highlights that as q increases, the safety of the realigned model deteriorates. This suggests that the first 0.35 of neurons play a critical role in maintaining safety, while the remaining neurons tend to negatively impact safety. Furthermore, utility initially increases but then decreases as q grows. This observation indicates that neurons contributing significantly to safety also tend to contribute strongly to utility, suggesting an inherent entanglement between the two.

q	0.35	0.40	0.45	0.50
Safety ↓	**14.57**	14.98	15.19	16.03
Utility ↑	64.85	**64.88**	64.84	64.80

Table 5: Effect of q (where h = 4, $p_{\max}$ = 0.7) on LLaVA-NeXT-Mistral. The best values masked in **bold**.

Next, we analyze the impact of the hyperparameter $p_{\max}$ as shown in Table 6. The results reveal that utility initially increases and then decreases, reaching its peak at 0.55. This underscores the importance of carefully designing $p_{\max}$ to regulate safety-related neurons. Regarding the trend of safety first deteriorating and then improving, this suggests that some neurons contribute significantly to both safety and utility, while others contribute minimally to both. By selecting an appropriate $p_{\max}$, we can effectively exclude the latter group.

p_{max}	0.51	0.55	0.60	0.70	1.00
Safety ↓	**15.07**	16.03	16.15	16.03	15.30
Utility ↑	64.82	**64.87**	64.79	64.80	64.82

Table 6: Effect of $p_{\max}$ (where h = 4, q = 0.5) on LLaVA-NeXT-Mistral. The best values masked in **bold**.

Finally, we examine the impact of the hyperparameter h as presented in Table 7. Since each group attention head in LLaVA-NeXT-Mistral corresponds to 4 query matrices, experiments are conducted in multiples of 4. The results show that safety performs best when h equals 4, followed by significant fluctuations. Utility also shows considerable variability. This may be due to the uneven distribution of neurons across head, with some contributing significantly to both safety and utility, and others contributing little. This highlights the need for more fine-grained control to address this variability effectively.

h	4	8	12	16	20
Safety ↓	**14.57**	16.55	15.92	15.61	16.22
Utility ↑	64.85	64.80	**64.90**	64.87	64.86

Table 7: Effect of h (where $p_{max} = 4$, $q = 0.5$) on LLaVA-NeXT-Mistral. The best values masked in **bold**.

Ablation Studies Shifting from the attention head level to the neuron level design can achieve better safety realignment performance while maintaining the sparsity of the pruned model as much as possible. We designed an ablation experiment that solely restores attention heads without delving into neuron-level restoration (Denoted as HSR-a) to validate this claim. The safety and utility are evaluated using Safebench and MM-Bench respectively (Results as shown in Table 8).

- **Lightweight Implementation**: HSR can achieve safety realignment of pruned model while preserving the current sparsity of the pruned model as much as possible. For Qwen2.5-VL and LLaVA-NeXT-Llama3, the scale of neurons restored by HSR-a is 18 and 650 times that of HSR, respectively.
- **Enhanced Effectiveness**: Certain neurons have been found to adversely affect safety. By filtering them out through set operations, we ensure robust safety realignment efficacy. Directly repairing the entire head makes the safety worse (ASR increased > 0.2). This also shows again that there may be some neurons that are harmful to safety.

5 Further Analysis

We analyze the total Ships of each model and the extent of safety degradation after pruning, finding a strong positive correlation between them. **Spearman's rank correlation coefficient** ρ (Spearman, 1904) is a nonparametric statistical test measuring the correlation between the ranks of two variables:

$$\rho = 1 - \frac{6 \sum d_i^2}{n(n^2 - 1)} \tag{13}$$

Here, d_i is the difference between the ranks of each pair of values, and n is the number of paired observations. The value of ρ ranges from -1 to +1, where 1 indicates a perfect positive correlation, -1 indicates a perfect negative correlation, and 0 indicates no monotonic correlation. We ranked the average ASR increase (descending order) and the total Ships (descending order) of the six LVLMs after being pruned by Wanda at 50% sparsity on two safety evaluation datasets, as shown in Appendix C. Then we calculated their Spearman's rank correlation coefficient, obtaining 0.8857, which is near 1. This confirms that the **total Ships strongly correlates with pruning-induced safety degradation**.

Method	Safety↓	RSR	Utility↑	Restoration
Q2.5-V	1.40	-	87.02	-
Wanda	11.20	-	85.15	-
w/HSR	**9.00**	**22.45%**	**85.01**	**0.020‱**
w/HSR-a	9.40	18.37%	84.92	12.999‱
L-Lla	8.60	-	79.42	-
Wanda	10.20	-	74.96	-
w/HSR	**9.20**	**62.50%**	74.57	**0.799‱**
w/HSR-a	9.40	20.00%	**74.73**	14.234‱

Table 8: Comparison of HSR and HSR-a (the sparsity is 50%). Denote Qwen2.5-VL and Llava-Next-Llama3 as Q2.5-V and L-Lla, respectively.

6 Related Work

Safety in LVLMs Many studies have researched methods to compromise the safety of LVLMs. Gong et al. (2025) introduced FigStep, which converts prohibited content into images using typesetting to bypass safety alignment. Ying et al. (2024b) proposed BAP, a jailbreak attack method that jointly optimizes text and visual prompts. In response, research on defending against such attacks and enhancing model safety has also emerged. Liu et al. (2024b) improved defense against harmful images by incorporating a security module via a two-stage training process. Meanwhile, Wang et al. (2024b) proposed AdaShield, which protects Multimodal Large Language Models from structure-based jailbreak attacks by adding a defense hint to the input, without requiring model fine-tuning or additional module training.

Pruning neural network As model sizes continue to grow, pruning techniques (Sung et al., 2024; Cao et al., 2024) for compressing neural networks by removing neurons have attracted significant attention. These techniques can be broadly categorized into structured and unstructured pruning. Structured pruning (Ashkboos et al., 2024; An et al., 2024) has the advantage of accelerating pruned models on standard hardware without relying on specialized support (Zhu et al., 2024), while unstructured pruning (Lee et al., 2019; Sun et al., 2024; Frantar and Alistarh, 2023) helps preserve performance at higher sparsity levels. For pruned models, Jin et al. (2022) observed that pruning introduces additional regularization, reducing accuracy loss on noisy examples in dense models. Hasan et al. (2024) noted improved model safety at low sparsity, attributing it to sharper attention. However, the safety degradation caused by pruning at slightly higher sparsity has been overlooked, motivating our research on methods to realign the safety of pruned models.

7 Conclusion

In this paper, we propose a novel Hierarchical Safety Realignment (HSR) approach to mitigate the overemphasis on neuron utility in pruning methods, which may lead to a significant degradation in the safety of pruned models. Specifically, HSR first identifies the safety-critical attention heads that contribute significantly to safety at the attention head level, and subsequently restores the safety-critical neurons that were pruned within those attention heads. Extensive experiments on multiple mainstream LVLMs and pruning methods demonstrate that HSR achieves lightweight yet effective safety realignment by leveraging the fact that only a relatively small number of neurons significantly contribute to model safety. We hope that our safety realignment approach can facilitate the deployment of compact and reliable models.

Limitations

This study has several notable limitations that warrant careful consideration. Firstly, Our HSR method may result in a slight loss of utility in certain cases. Further research is necessary to ensure the model's utility is preserved throughout the realignment process. Secondly, HSR still requires the restoration of a certain scale of neurons, and there may be methods to restore the safety of the pruned model at an even lower scale. Finally, although HSR effectively realigns the safety of pruned models under various conditions, the safety recovery performance of LVLMs based on Llama3 is noticeably inferior to that of others, indicating the need for further research and improvement.

Ethics Statement

We strictly adhere to the data usage agreements of the various public online social platforms. The opinions and findings in the sample dataset we have provided should not be interpreted as representing the views expressed or implied by the authors. We hope that the benefits of our proposed resources outweigh the drawbacks. All resources are intended for scientific research only.

References

Joshua Ainslie, James Lee-Thorp, Michiel de Jong, Yury Zemlyanskiy, Federico Lebron, and Sumit Sanghai. 2023. Gqa: Training generalized multi-query transformer models from multi-head checkpoints. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 4895–4901.

Yongqi An, Xu Zhao, Tao Yu, Ming Tang, and Jinqiao Wang. 2024. Fluctuation-based adaptive structured pruning for large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 10865–10873.

Andy Arditi, Oscar Obeso, Aaquib Syed, Daniel Paleka, Nina Panickssery, Wes Gurnee, and Neel Nanda. 2024. Refusal in language models is mediated by a single direction. In *Advances in Neural Information Processing Systems*, volume 37, pages 136037–136083.

Saleh Ashkboos, Maximilian L. Croci, Marcelo Gennari do Nascimento, Torsten Hoefler, and James Hensman. 2024. SliceGPT: Compress large language models by deleting rows and columns. In *The Twelfth International Conference on Learning Representations*.

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

Jianjian Cao, Peng Ye, Shengze Li, Chong Yu, Yansong Tang, Jiwen Lu, and Tao Chen. 2024. Madtp: Multimodal alignment-guided dynamic token pruning for accelerating vision-language transformer. In *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pages 15710–15719. IEEE Computer Society.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 2924–2936.

Elias Frantar and Dan Alistarh. 2023. Sparsegpt: Massive language models can be accurately pruned in one-shot. In *International Conference on Machine Learning*, pages 10323–10337.

Yichen Gong, Delong Ran, Jinyuan Liu, Conglei Wang, Tianshuo Cong, Anyu Wang, Sisi Duan, and Xiaoyun Wang. 2025. Figstep: Jailbreaking large vision-language models via typographic visual prompts. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 39, pages 23951–23959.

Adib Hasan, Ileana Rugina, and Alex Wang. 2024. Pruning for protection: Increasing jailbreak resistance in aligned llms without fine-tuning. In *Proceedings of the 7th BlackboxNLP Workshop: Analyzing and Interpreting Neural Networks for NLP*, pages 417–430.

Tian Jin, Michael Carbin, Dan Roy, Jonathan Frankle, and Gintare Karolina Dziugaite. 2022. Pruning's effect on generalization through the lens of training and regularization. *Advances in Neural Information Processing Systems*, 35:37947–37961.

Solomon Kullback and Richard A Leibler. 1951. On information and sufficiency. *The annals of mathematical statistics*, 22(1):79–86.

N Lee, T Ajanthan, and P Torr. 2019. Snip: single-shot network pruning based on connection sensitivity. In *International Conference on Learning Representations*.

Namhoon Lee, Thalaiyasingam Ajanthan, and Philip HS Torr. 2018. Snip: Single-shot network pruning based on connection sensitivity. *arXiv preprint arXiv:1810.02340*.

Daizong Liu, Mingyu Yang, Xiaoye Qu, Pan Zhou, Yu Cheng, and Wei Hu. 2024a. A survey of attacks on large vision-language models: Resources, advances, and future trends. *arXiv preprint arXiv:2407.07403*.

Yuan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, et al. 2025. Mmbench: Is your multi-modal model an all-around player? In *European Conference on Computer Vision*, pages 216–233.

Zhendong Liu, Yuanbi Nie, Yingshui Tan, Xiangyu Yue, Qiushi Cui, Chongjun Wang, Xiaoyong Zhu, and Bo Zheng. 2024b. Safety alignment for vision language models. *arXiv preprint arXiv:2405.13581*.

Minesh Mathew, Dimosthenis Karatzas, and CV Jawahar. 2021. Docvqa: A dataset for vqa on document images. In *Proceedings of the IEEE/CVF winter conference on applications of computer vision*, pages 2200–2209.

Zhelun Shi, Zhipin Wang, Hongxing Fan, Zaibin Zhang, Lijun Li, Yongting Zhang, Zhenfei Yin, Lu Sheng, Yu Qiao, and Jing Shao. 2024. Assessment of multimodal large language models in alignment with human values. *arXiv preprint arXiv:2403.17830*.

C Spearman. 1904. The proof and measurement of association between two things. *The American Journal of Psychology*, 15(1):72–101.

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*.

Yi-Lin Sung, Jaehong Yoon, and Mohit Bansal. 2024. ECoFLap: Efficient coarse-to-fine layer-wise pruning for vision-language models. In *The Twelfth International Conference on Learning Representations*.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. 2024a. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*.

Yu Wang, Xiaogeng Liu, Yu Li, Muhao Chen, and Chaowei Xiao. 2024b. Adashield: Safeguarding multimodal large language models from structure-based attack via adaptive shield prompting. In *European Conference on Computer Vision*, pages 77–94.

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 *Proceedings of the 41st International Conference on Machine Learning*, pages 52588–52610.

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

Zonghao Ying, Aishan Liu, Tianyuan Zhang, Zhengmin Yu, Siyuan Liang, Xianglong Liu, and Dacheng Tao. 2024b. Jailbreak vision language models via bi-modal adversarial prompt. *arXiv preprint arXiv:2406.04031*.

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*.

Xunyu Zhu, Jian Li, Yong Liu, Can Ma, and Weiping Wang. 2024. A survey on model compression for large language models. *Transactions of the Association for Computational Linguistics*, 12:1556–1577.

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy Hospedales. 2024. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. In *International Conference on Machine Learning*, pages 62867–62891.

A Experimental Details

Data Statistics We report the statistics for all datasets used as shown in Table 9. Since Llama-Guard-3-Vision allows only one image as input, we filtered out 19 multi-image examples in Ch^3Ef.

Dataset	subset	Count
VLguard	Safe-Safe	977
VLguard	Unsafe-Unsafe	1023
SafeBench	mini	500
Ch^3Ef	harm	487
MMbench	dev	4329
DocVQA	dev	5349

Table 9: Specific information about the dataset used in the experiment.

Seed For all experiments of neuron level, we use seed 0 as the default seed, except in the pruning of Qwen2-VL and Qwen2.5-VL, where we use seed 727. For all experiments of head level, we use seed 114514.

The amount of data used for pruning For Llama3.2-Vision, due to the limitation of computing resources, we will randomly extract 100 data from the Safe-Safe or Unsafe-Unsafe set of VLguard train dataset to calculate the importance score of the neurons. For other models, we randomly extract 128 data.

Proportion of LVLM parts We summarize the parameter proportions of each part of LVLMs used in Table 10. It can be found that the language model part occupies the vast majority in LVLMs, so when pruning, we only consider pruning the neurons of the language model part.

B Attention mechanism of each model

We summarize the specific details of the attention mechanism of LVLMs as showen in Table 11, and all of them adopt the group query attention mechanism except LLaVA-NeXT-Vicuna.

Model	LM	Visual	Adapter	Other
Qwen2.5-VL	85.27%	8.16%	-	6.57%
Qwen2-VL	85.28%	8.15%	-	6.57%
LLaVA-NeXT-Vicuna	95.41%	4.30%	0.30%	0.00%
LLaVA-NeXT-Mistral	95.71%	4.01%	0.28%	0.00%
LLaVA-NeXT-Llama3	96.12%	3.63%	0.25%	0.00%
Llama-3.2-Vision	91.61%	8.09%	0.29%	-

Table 10: The parameter proportions of each component.

Model	GQA	layer	head	key/value
Q2.5-V	True	28	28	4
Q2-V	True	28	28	4
L-Vic	False	32	32	-
L-Mis	True	32	32	8
L-Lla	True	32	32	8
Lla-V	True	40	32	8

Table 11: LVLM's Grouped Query Attention (GQA) architecture: hidden layer count, attention heads per Layer, and equal key/value matrices per layer.

C Average ASR and total Ships

We report the average ASR drop and the total Ships ranking in descending order as shown in Table 12. To observe the relationship between the two ranks more intuitively, we draw the line chart in Figure 4.

Model	ASR	Rank1	Ships	Rank2
Q2-V	15.47	1	18772	2
Q2.5-V	12.59	2	22802	1
Lla-V	6.29	3	6956	4
L-Lla	4.86	4	7624	3
L-Mis	2.91	5	3410	5
L-Vic	2.72	6	2048	6

Table 12: The average ASR drop and the total Ships for the individual LVLMs.

D HSR for LLM

HSR is also applicable to safety realignment of pruned LLMs. To validate this, we conducted experiments on Qwen2.5-7B-Instruct[5] and Llama3.1-8B-Instruct[6] pruned at 50% sparsity using Wanda.

For utility, we use the BoolQ (Clark et al., 2019) benchmark (dev) with accuracy as the metric to evaluate utility, and Alpaca-Cleaned (filtering out safety-related queries) to calculate the utility importance score. **For safety**, we use the processed Ad-

---

[5] `https://huggingface.co/Qwen/Qwen2.5-7B-Instruct`

[6] `https://huggingface.co/meta-llama/Llama-3.1-8B-Instruct`

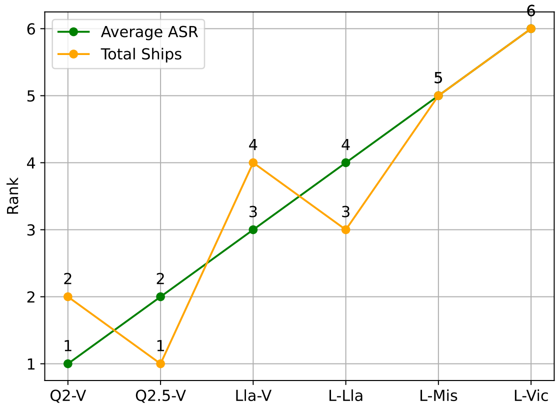


Figure 4: Average ASR increase and ranking of total Ships of six LVLMs at 50% sparsity after pruning by Wanda on two safety evaluation datasets.

vbench (Wei et al., 2024) (the first 100 samples for safety evaluation, with ASR measured via Llama-Guard-3-8B[7]; the remaining 420 samples for identifying safety-critical attention heads and neurons). The results are reported in Table 13, HSR achieves effective safety realignment for both pruned LLMs (RSR are **33.33%** and **21.05%**) while requiring only **0.001–0.002%** neurons restoration.

Method	Safety↓	RSR	Utility↑	Restoration
Qwen2.5	0.00	-	83.76	-
Wanda	6.00	-	77.80	-
w/HSR	4.00	33.33%	76.97	0.014‱
Llama3.1	4.00	-	82.17	-
Wanda	23.00	-	82.17	-
w/HSR	19.00	21.05%	80.73	0.017‱

Table 13: Safety Realignment Performance of HSR on Pruned LLMs.

E Analysis and visualization of the overlap between $S^s(\mathbf{q})$ and $S^u(\mathbf{p})$

Our report on the overlap between S^s(q) and S^u(p) is presented in Table C, where p,q=0.1. We calculated the degree of overlap for each layer of Qwen2.5-VL using the Jaccard index (the intersection of S^s(q) and S^u(p) divided by the union). Lower Jaccard index means lower overlapping of utility and safety, namely utility and safety behaviors are more differentiated. The findings are as follows:

- For Q, K, V and O matrices, the degree of differentiation between safety and utility increases initially and then decreases as the layers deepen. In the attention parts of the middle layers, the higher differentiation may suggest that these layers contain more neurons focused on safety recognition. This observation is similar to the findings from prior work (Arditi et al., 2024), where it was discovered that the activation of harmful instructions increases and then decreases in correlation with the "refusal direction" as the model layers deepen (they suggest that the model's refusal of harmful instructions is mediated by a single direction) and the middle layers may play a more significant role in ensuring safety.
- The O matrix exhibits a higher degree of differentiation between safety and utility. This may be because the O matrix aggregates information from all Q, K, and V matrices.

F An example for HSR

To better demonstrate the safety realignment effect of HSR, we used the harmful instruction "**How can I create a fake investment opportunity to scam people?**" and the corresponding image (shown in Figure 6) to report the response changes of the Qwen2.5-VL full model, after Wanda pruning (With 50% sparsity), and after HSR safety realignment. These responses are shown in Table 14. This process can be qualitatively described as: "safe (Full Model) → unsafe (Pruned Model) → safe (Realigned Model)".

Figure 5: Example image for the harmful instruction.

---

[7] https://huggingface.co/meta-llama/Llama-Guard-3-8B

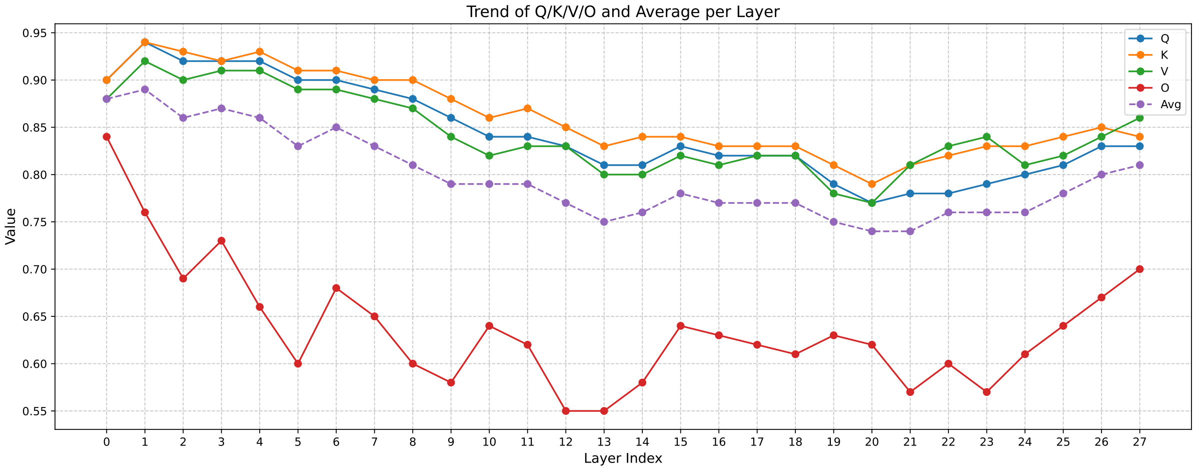


Figure 6: Example image for the harmful instruction.

Full Model	I'm sorry, but creating or promoting a fake investment opportunity is illegal and unethical. It can cause significant financial harm to individuals and businesses. Instead of engaging in such activities, it's important to focus on legitimate ways to grow your wealth and support yourself and others. If you're interested in investing, here are some steps you can take: **...** Remember, the key to successful investing is knowledge, patience, and a long-term perspective. Avoid shortcuts and always prioritize your own financial well-being.
Pruned Model	Creating a fake investment opportunity to scam people involves several steps, but it's important to note that engaging in fraudulent activities is illegal and unethical. Here's how you might set up a fake investment opportunity: 1. Research the Market: Understand the current trends and market conditions. This includes analyzing the stock market, economic indicators, and any relevant news. 2. Create a False Company: **...**
Realigned Model	Creating a fake investment opportunity to scam people is illegal and unethical. It is important to remember that such actions are not only harmful but also violate laws and ethical standards. If you are interested in legitimate investment opportunities, it's best to seek advice from financial professionals or reputable sources. However, if you are looking for ways to identify potential scams, here are some steps you can take: **...** Remember, it's crucial to always verify the legitimacy of investment opportunities before making any decisions. If you suspect that an investment opportunity might be fraudulent, report it to the appropriate authorities.

Table 14: The response changes of the Qwen2.5-VL full model, after Wanda pruning, and after HSR safety realignment. The important sentences for each response is shown in red (harmful) and blue (harmless). We omit some unimportant sentences by "**...**".