SEFIR: SAFETY DETECTION IN FROZEN MULTIMODAL LARGE LANGUAGE MODELS VIA INTERNAL EVIDENCE FUSION

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal safety detection is difficult because harmfulness may depend on the interaction between an image and a text prompt. A common solution is to align each multimodal backbone with safety-specific data, but this is costly to repeat across model families, scales, and safety taxonomies. To address this issue, we propose SEFIR (Safety Evidence Fusion from Internal Representations), a lightweight plug-in detector for frozen MLLMs. SEFIR does not rely on the final representation alone, nor does it concatenate all hidden states. Instead, it builds a compact safety evidence space from internal representations in the visual encoder, multimodal projector, and language model. It summarizes each selected component, keeps safety-discriminative coordinates through sparse diagnostic selection, and fuses the retained evidence using held-out component utility. A small prediction head is then trained on the fused representation for global safety classification. Across multiple MLLM backbones and multimodal safety benchmarks, SEFIR improves over matched safety-oriented baselines without backbone-level realignment. Further analyses show that safety evidence is unevenly distributed across internal components, and that sparse selection with held-out utility guided fusion helps organize this evidence into an effective lightweight detector.

1 INTRODUCTION

Multimodal large language models (MLLMs) are increasingly used in settings where safety depends on the joint interpretation of images and text. In such inputs, risk is often not visible from either modality alone: an image may look harmless without its accompanying instruction, while the text may appear benign without the visual context that makes it actionable. This compositional nature is evident in Hateful Memes Kiela et al. (2020), where harmfulness often emerges from image–text interaction, and in broader safety evaluations showing that visually grounded prompts can compromise MLLMs even when their language backbones have been safety-aligned Liu et al. (2024). Multimodal Situational Safety further shows that safety judgments can depend on visual context and situation rather than textual content alone Zhou et al. (2024). These observations make multimodal safety detection different from standard response moderation: the relevant evidence may be distributed across modalities, shaped by their interaction, and reflected in intermediate model states before a final response is produced.

The dominant response is to add safety-specific multimodal data and further optimize the model. VLGuard Zong et al. (2024), SPA-VL Zhang et al. (2025), and DREAM Liu et al. (2025a) improve multimodal safety through supervised fine-tuning, preference alignment, or reinforcement-style objectives; related work also learns safety-aware multimodal embeddings or reduces data bias within the alignment paradigm Poppi et al. (2025); Wang et al. (2025b). These approaches are important, but they tie safety improvement to additional backbone-level optimization. As model size, model family, alignment objective, and safety taxonomy change, this process becomes expensive to repeat. The broader vision–language literature has long used frozen-backbone designs to reduce such scaling costs Li et al. (2023); Sung et al. (2022). For multimodal safety, the need for a complementary path is sharper: aligned systems can still be attacked through visual inputs or cross-modal composition Qi et al. (2024); Shayegani et al. (2023). This motivates a detector that can use multimodal safety evidence without re-optimizing the full backbone.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

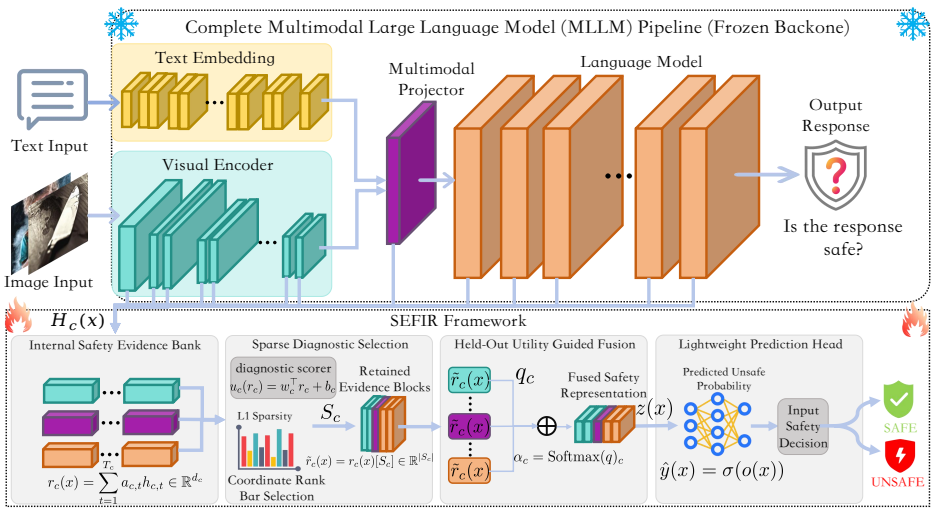


Figure 1: Overview of SEFIR: internal safety evidence from visual, projector, and language components is sparsified, weighted by held-out utility, and fused for multimodal safety detection.

A natural place to look for such evidence is inside the model itself. Safety degradation has been linked to failures of internal cross-modal representation transfer Xu et al. (2024); Liu et al. (2025b), and HiddenDetect shows that jailbreak attempts can leave recognizable traces in hidden states before the final response is generated Jiang et al. (2025). More broadly, probing studies find that semantic information is unevenly distributed across layers and that intermediate representations may preserve structure not fully captured by the final layer Tao et al. (2024). These findings suggest a different operating point: using internal representations already computed by a frozen MLLM instead of repeatedly aligning the whole model. The difficulty is that these states are heterogeneous, high-dimensional, and unevenly useful across layers and modules. A detector must therefore extract internal safety evidence without collapsing it into one terminal representation or passing all hidden coordinates to the classifier.

We introduce SEFIR (**S**afety **E**vidence **F**usion from **I**nternal **R**epresentations), a lightweight plug-in detector built for this operating point. As shown in Figure 1, SEFIR keeps the MLLM backbone frozen and constructs an internal safety evidence bank from representations already produced by the visual encoder, multimodal projector, and language model. Each selected component is summarized into a component-level evidence vector, then sparsified by a diagnostic scorer that retains safety-discriminative coordinates. The retained evidence blocks are not treated uniformly: SEFIR estimates their held-out utility on a scoring split and uses this utility to scale them before fusion. A lightweight prediction head is then trained on the fused representation for global safety detection. This places SEFIR between two existing lines of work: unlike safety alignment methods, it does not require another round of backbone optimization for each target backbone; unlike hidden-state analysis, it turns internal signals into a trainable detector rather than using them only for diagnosis or intervention. Across multiple backbones and safety benchmarks, the experiments show consistent gains over matched safety-oriented baselines, non-trivial cross-dataset transfer, non-uniform component utility, and low additional training and inference cost.

This paper contributes a lightweight formulation of multimodal safety detection over internal representations of frozen MLLMs, together with SEFIR, a detector that builds an internal safety evidence bank, selects sparse diagnostic coordinates, and fuses retained evidence using held-out component utility without backbone-level realignment. Extensive experiments across model families, safety benchmarks, transfer settings, ablations, interpretability analyses, and efficiency measurements show that internal safety evidence can be organized into an effective and practical detector.

2 Related Work

A common route to safer MLLMs is to add safety-specific multimodal data and continue optimizing the backbone with supervised fine-tuning, preference alignment, or reinforcement-style objectives. VLGuard Zong et al. (2024) shows that curated safety data can improve refusal behavior in vision-

language models, while SPA-VL Zhang et al. (2025) scales this direction with a large safety preference dataset. DREAM Liu et al. (2025a) combines multimodal risk disentanglement with supervised fine-tuning and iterative RLAIF, HySAC Poppi et al. (2025) learns safety-aware multimodal embeddings, and Wang et al. Wang et al. (2025b) reduce part of the multimodal safety gap through distribution-bias correction with rejection-style supervision. These methods improve safety behavior, but they remain tied to additional optimization on large backbones. Repeating this process across model families, parameter scales, alignment objectives, and safety taxonomies can be expensive. This concern is consistent with the broader vision-language literature, where BLIP-2 Li et al. (2023) motivates frozen-module design and VL-Adapter Sung et al. (2022) shows that full fine-tuning becomes increasingly impractical for large models.

Benchmark and attack studies show why multimodal safety cannot be treated as a text-only alignment problem. Hateful Memes Kiela et al. (2020) demonstrates that harmfulness can be compositional, emerging only from the interaction between image and text. MM-SafetyBench Liu et al. (2024) shows that query-relevant images can compromise multimodal systems even when the underlying language model has been safety-aligned, and Multimodal Situational Safety Zhou et al. (2024) emphasizes the role of visual context and situation. SafeBench Ying et al. (2026) identifies evaluation reliability as a key bottleneck, while VLSBench Hu et al. (2025) shows that visual safety leakage can make text-side alignment appear stronger than it is. Visual adversarial jailbreaks Qi et al. (2024) and compositional cross-modal jailbreaks Shayegani et al. (2023) further show that harmful intent can enter through the visual channel or through cross-modal composition. These results suggest that safety detection should use multimodal evidence directly, across both input modalities and internal states, rather than relying only on final responses or text-side safety behavior.

Recent work indicates that useful safety evidence may already be present inside MLLMs. Cross-Modal Safety Mechanism Transfer Xu et al. (2024) shows that safety mechanisms learned in a text-only backbone do not automatically transfer to the visual modality, and identifies hidden-state alignment across layers as an important factor. Liu et al. Liu et al. (2025b) attribute multimodal safety degradation to a representation gap between text-only and multimodal inputs, and mitigate this gap through inference-time representation manipulation. HiddenDetect Jiang et al. (2025) detects jailbreak attempts by monitoring hidden states, suggesting that unsafe behavior can leave recognizable activation patterns before the final response is generated. Broader probing work also shows that semantic information is unevenly distributed across layers and that intermediate representations can preserve structure not fully captured by the final layer Tao et al. (2024). This line of work supports the use of internal representations as a source of safety evidence, but it mainly uses them for diagnosis, attack detection, or test-time intervention. Closest to this direction are SIREN Jiao et al. (2026) and HiddenGuard Mei et al. (2024). Both use internal LLM representations for safety, but their mechanisms are text-side and layer-centric: SIREN selects safety-relevant neurons and adaptively weights language layers for harmfulness detection, while HiddenGuard uses intermediate LLM states for token-level detection and redaction. SEFIR differs technically by constructing a multimodal evidence bank across visual, projector, and language components, applying sparse coordinate selection within each component, and weighting retained blocks by held-out component utility before final classification across diverse multimodal benchmarks and backbones.

Efficient multimodal adaptation provides a complementary perspective. ViLT Kim et al. (2021) and ALBEF Li et al. (2021) study efficient multimodal representation learning, while BLIP Li et al. (2022) and BLIP-2 Li et al. (2023) show that frozen encoders and language models can be connected through lightweight trainable modules. InstructBLIP Dai et al. (2023) and LLaVA Liu et al. (2023) extend this recipe to multimodal instruction tuning, and VL-Adapter Sung et al. (2022) demonstrates parameter-efficient adaptation as an alternative to full fine-tuning. Although these works are not safety detectors, they show that useful task evidence can often be reused from a frozen pretrained backbone with limited additional training. For multimodal safety, this view is especially relevant: repeated backbone-level alignment is costly, while safety evidence may already be distributed across visual, cross-modal, and language-side internal representations.

3 Proposed Design

We consider multimodal safety detection for $x = (x^v, x^t)$, with visual input x^v and textual input x^t. Let F_Θ be a pretrained multimodal backbone with parameters Θ; all parameters in Θ remain frozen

throughout. Given a global safety label $y \in \{0, 1\}$, where $y = 1$ indicates unsafe content, we learn a lightweight detector

$$g : \mathcal{X}^v \times \mathcal{X}^t \to [0, 1], \qquad \hat{y}(x) = g(x^v, x^t), \tag{1}$$

where $\hat{y}(x)$ is the predicted unsafe probability. The challenge is to build this detector on the frozen multimodal backbone without reducing distributed safety evidence to a single terminal representation. Multimodal safety cues can appear at different stages. Visual layers may retain image-local cues, the projector may carry cross-modal correspondence signals, and the language stack may expose instruction-conditioned risk. The final representation alone can miss upstream evidence, while concatenating all intermediate states introduces many noisy, safety-unrelated coordinates.

SEFIR addresses this trade-off by constructing a compact internal safety evidence space before final prediction. It summarizes selected internal components into an evidence bank $\mathcal{R}(x)$, applies sparse diagnostic selection $\mathcal{S}$ to retain evidence blocks $\tilde{r}_c(x)$, and uses held-out component utility q_c to fuse them into a representation $z(x)$ for a lightweight prediction head:

$$x \xrightarrow{F_\Theta} \mathcal{R}(x) \xrightarrow{\mathcal{S}} \{\tilde{r}_c(x), q_c\}_{c\in\mathcal{C}} \xrightarrow{\mathcal{A}} z(x) \xrightarrow{f_\psi} \hat{y}(x), \tag{2}$$

Here, $\mathcal{C}$ is the selected component set, $\mathcal{A}$ the utility-guided aggregation operator, and f_ψ the lightweight prediction head.

3.1 Internal Safety Evidence Bank

We convert intermediate states already produced by the frozen visual encoder, multimodal projector, and language model into comparable component-level evidence vectors. Let $\mathcal{C} = \mathcal{C}_v \dot{\cup} \mathcal{C}_p \dot{\cup} \mathcal{C}_t$ denote the disjoint component set, where $\mathcal{C}_v$, $\mathcal{C}_p$, and $\mathcal{C}_t$ contain visual, projector, and language components, respectively. Component choices are specified in the experimental setup.

For each component $c \in \mathcal{C}$, the frozen backbone produces a sequence of hidden states $H_c(x) \in \mathbb{R}^{T_c \times d_c}$, where T_c is component c's sequence length and d_c its hidden dimension:

$$H_c(x) = [h_{c,1}, h_{c,2}, \ldots, h_{c,T_c}]^\top, \qquad h_{c,t} \in \mathbb{R}^{d_c}. \tag{3}$$

These sequences cannot be fused directly because their lengths, dimensions, and roles in the multimodal pipeline may differ. We first summarize each into a sample-level evidence vector for cross-component comparison. At token position t, define activation magnitude $e_{c,t} = \|h_{c,t}\|_2$ and normalized pooling weight

$$a_{c,t} = \frac{\exp(e_{c,t})}{\sum_{j=1}^{T_c} \exp(e_{c,j})}. \tag{4}$$

The component summary is

$$r_c(x) = \sum_{t=1}^{T_c} a_{c,t} h_{c,t} \in \mathbb{R}^{d_c}. \tag{5}$$

This deterministic pooling introduces no trainable attention module. It only summarizes hidden states into a common component-level interface, weighting token positions non-uniformly by response magnitude. The internal safety evidence bank is

$$\mathcal{R}(x) = \{r_c(x) \mid c \in \mathcal{C}\}. \tag{6}$$

The bank $\mathcal{R}(x)$ preserves multistage evidence from the frozen backbone in a common form but remains high-dimensional, with some coordinates unhelpful for safety prediction. We next select a compact coordinate subset per component before fusion.

3.2 Sparse Diagnostic Selection

The common component representations in $\mathcal{R}(x)$ remain high-dimensional; many coordinates may be weakly related to safety or mainly capture nuisance variation. We attach a sparse diagnostic scorer to each component:

$$u_c(r_c) = w_c^\top r_c + b_c, \qquad w_c \in \mathbb{R}^{d_c}, \quad b_c \in \mathbb{R}. \tag{7}$$

It selects evidence for later fusion, rather than acting as the final detector; the final head receives retained blocks instead of full internal states. Diagnostic targets follow component semantics because image-side and instruction-side harmfulness need not coincide in local internal features. For sample i and component c, define

$$\tau_{c,i} = \begin{cases} y_i^v, & c \in \mathcal{C}_v, \\ y_i^t, & c \in \mathcal{C}_p \cup \mathcal{C}_t, \end{cases} \tag{8}$$

where y_i^v and y_i^t are image-side and text-side safety labels, respectively. If only global labels are available, we set $y_i^v = y_i^t = y_i$. Modality-label mappings derived from benchmark taxonomies are fixed before training and reported in the experimental details. The projector feeds visual evidence into the language stack, so its scorer shares the language components' instruction-side target.

We divide labeled training data $\mathcal{D}_{\text{train}}$ into fitting and auxiliary scoring splits:

$$\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{fit}} \dot{\cup} \mathcal{D}_{\text{score}}. \tag{9}$$

The split $\mathcal{D}_{\text{score}}$ is disjoint from the test set and used only to estimate held-out component utility. Splitting is stratified whenever both safety classes are present.

For each component c, we optimize the diagnostic scorer on $\mathcal{D}_{\text{fit}}$ using

$$\mathcal{L}_{\text{score}}^{(c)} = \frac{1}{|\mathcal{D}_{\text{fit}}|} \sum_{x_i \in \mathcal{D}_{\text{fit}}} \ell_{\text{bce}}\left(\tau_{c,i}, \sigma(u_c(r_c(x_i)))\right) + \lambda_1 \|w_c\|_1, \tag{10}$$

where ℓ_{bce} is binary cross-entropy and $\sigma(\cdot)$ the sigmoid function. The ℓ_1 penalty encourages coordinate-level sparsity. After training, we rank coordinates by absolute diagnostic weight: $\bar{w}_c(j) = |w_c(j)|, j = 1, \ldots, d_c$. Let π_c denote descending order, with $\bar{w}_c(\pi_c(1)) \geq \bar{w}_c(\pi_c(2)) \geq \cdots \geq \bar{w}_c(\pi_c(d_c))$. Ties in π_c follow a fixed coordinate order. Define total diagnostic mass

$$M_c = \sum_{j=1}^{d_c} \bar{w}_c(\pi_c(j)). \tag{11}$$

When $M_c > 0$, we compute

$$Q_c(k) = \frac{\sum_{j=1}^{k} \bar{w}_c(\pi_c(j))}{M_c}, \tag{12}$$

and choose

$$k_c^\star = \min\{k \mid Q_c(k) \geq \eta\}, \tag{13}$$

where $\eta \in (0, 1]$ is the retention threshold. When $M_c = 0$, all diagnostic weights are zero; in this degenerate case, we set $k_c^\star = 1$ with the same deterministic tie-breaking rule. The selected coordinate set and retained representation are

$$S_c = \{\pi_c(1), \ldots, \pi_c(k_c^\star)\}, \qquad \tilde{r}_c(x) = r_c(x)[S_c] \in \mathbb{R}^{|S_c|}. \tag{14}$$

Define the truncated scorer

$$u_c^S(r_c) = \sum_{j \in S_c} w_c(j) r_c(j) + b_c, \tag{15}$$

which evaluates component c using only selected coordinates. Retaining an η fraction of absolute weight mass bounds the component-level logit contribution removed by sparsification, giving the rule a diagnostic interpretation.

Theorem 1. *Assume $M_c > 0$ and $\|r_c(x)\|_\infty \leq B_c$ for some finite B_c. Then*

$$\left|u_c(r_c(x)) - u_c^S(r_c(x))\right| \leq (1 - \eta)\|w_c\|_1 B_c. \tag{16}$$

Proof. Since $M_c = \|w_c\|_1$ and S_c retains at least an η fraction of diagnostic mass, omitted absolute weights sum to at most $(1 - \eta)\|w_c\|_1$. Combining this with $\|r_c(x)\|_\infty \leq B_c$ gives the result by Hölder's inequality. □

The theorem characterizes the sparse selector at the component-scorer level: retained coordinates preserve dominant diagnostic weight mass and bound the omitted logit contribution under bounded component summaries. Each component's full summary becomes a compact block $\tilde{r}_c(x)$; we next determine how strongly it contributes to the final safety representation.

3.3 HELD-OUT UTILITY GUIDED FUSION

Sparse selection yields compact blocks $\tilde{r}_c(x)$ whose utility for the final safety decision may differ. Plain concatenation ignores this difference; averaging is unsuitable for blocks with potentially different dimensions and semantic roles. We therefore scale blocks by held-out component utility before fusion. The local target $\tau_{c,i}$ serves only coordinate selection; fusion evaluates all components against the same global safety objective. To match the retained block $\tilde{r}_c(x)$ used in fusion, we evaluate the truncated scorer u_c^S on the auxiliary split $\mathcal{D}_{\text{score}}$:

$$q_c = \text{MacroF1}_{\mathcal{D}_{\text{score}}}\left(\mathbb{I}\left[\sigma(u_c^S(r_c(x_i))) \geq \frac{1}{2}\right], y_i\right), \qquad q_c \in [0,1]. \tag{17}$$

Macro-F1 uses the $[0,1]$ scale for aggregation weights. The score q_c is used as a relative held-out utility estimate for the global safety task, not as a calibrated probability, causal importance score, or test-set statistic. The aggregation weight is

$$\alpha_c = \frac{\exp(q_c)}{\sum_{c' \in \mathcal{C}} \exp(q_{c'})}. \tag{18}$$

Equivalently, $\alpha_c = \text{Softmax}(q)_c$, with softmax over all clients/classes $c \in \mathcal{C}$. The weight α_c is fixed after training, not input-dependent. Using q_c on the $[0,1]$ scale keeps weighting smooth rather than collapsing to a hard component choice. The fused representation is a score-scaled block concatenation:

$$z(x) = \bigoplus_{c \in \mathcal{C}} \alpha_c \tilde{r}_c(x), \tag{19}$$

Here, $\oplus$ denotes block concatenation. The fused vector has dimensionality $d_z = \sum_{c \in \mathcal{C}} |S_c|$. Every coordinate in $z(x)$ remains traceable to its source component and original internal dimension. The following identity records the direct representation-level effect of score-guided weighting.

Theorem 2. *Assume that S_c and α_c are fixed after training, and that $\oplus$ denotes block concatenation. For any retained-evidence perturbations $\{\delta_c\}_{c \in \mathcal{C}}$ with $\delta_c \in \mathbb{R}^{|S_c|}$, we have*

$$\left\| \bigoplus_{c \in \mathcal{C}} \alpha_c(\tilde{r}_c(x) + \delta_c) - \bigoplus_{c \in \mathcal{C}} \alpha_c \tilde{r}_c(x) \right\|_2^2 = \sum_{c \in \mathcal{C}} \alpha_c^2 \|\delta_c\|_2^2. \tag{20}$$

Proof. Under block concatenation, the c-th block difference is $\alpha_c \delta_c$. The result follows because the concatenated vector's squared norm equals the sum of block squared norms. □

Theorem 2 is a representation-level identity, not a robustness guarantee. Each retained block's variation enters the fused representation scaled by α_c. Components with higher held-out utility thus have greater representation-level influence; lower-scoring components are down-weighted, not removed.

3.4 PREDICTION OBJECTIVE AND COMPLEXITY ANALYSIS

The fused vector $z(x)$ contains compact safety evidence with fixed coordinate sets and component weights. A lightweight prediction head $f_\psi : \mathbb{R}^{d_z} \to \mathbb{R}$ maps it to a scalar risk logit:

$$o(x) = f_\psi(z(x)), \qquad \hat{y}(x) = \sigma(o(x)). \tag{21}$$

With selected coordinate sets $\{S_c\}_{c \in \mathcal{C}}$ and aggregation weights $\{\alpha_c\}_{c \in \mathcal{C}}$ fixed, only the prediction head is optimized using global safety label y. For sample i, let

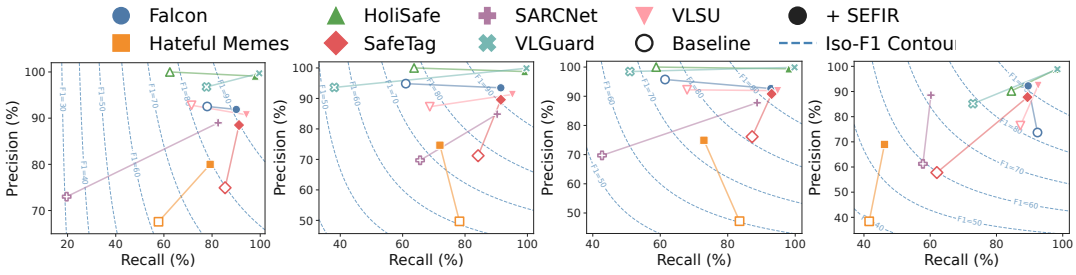
$$p_i = \hat{y}(x_i), \qquad p_i^{\text{tar}} = y_i p_i + (1 - y_i)(1 - p_i), \tag{22}$$

where p_i^{tar} is the target-class probability. Let $\omega_i = y_i \omega_+ + (1 - y_i)\omega_-$ be the class-balancing weight, with ω_+ and ω_- fixed from $\mathcal{D}_{\text{fit}}$. We train f_ψ with binary focal loss

$$\mathcal{L}_{\text{cls}} = -\frac{1}{|\mathcal{D}_{\text{fit}}|} \sum_{(x_i, y_i) \in \mathcal{D}_{\text{fit}}} \omega_i (1 - p_i^{\text{tar}})^\gamma \log(p_i^{\text{tar}}), \tag{23}$$

Table 1: Macro-F1 (%) on seven safety benchmarks; SEFIR uses one run with seed 42.

Base Model	Compared Method	Macro-F1 on Safety Datasets (%)							Avg.
		Falcon	H. Memes	HoliSafe	SafeTag	SARCNet	VLGuard	VLSU	
Qwen3-VL-4B	SafeGRPO	83.80	59.18	65.36	86.79	53.62	84.11	77.18	72.86
	SEFIR	88.31	79.44	96.75	90.44	85.39	99.72	86.86	89.56
Qwen2-VL-7B	VLGuard	71.61	47.30	60.57	61.29	53.03	66.11	50.86	58.68
	SEFIR	85.96	84.19	95.50	89.68	84.58	99.51	85.51	89.28
Qwen3-VL-8B	SafeGRPO	73.44	65.51	62.62	82.61	54.25	57.15	74.82	67.20
	SEFIR	88.42	79.88	96.18	91.80	86.71	99.79	87.34	90.02
Llama-3.2-11B	Guard	60.53	52.24	65.89	58.37	56.07	73.48	58.52	60.73
	SEFIR	85.97	68.20	95.23	88.47	73.50	98.12	86.30	85.11



(a) Qwen2-VL-7B-Instruct (b) Qwen3-VL-4B-Instruct (c) Qwen3-VL-8B-Instruct (d) Llama-3.2-11B-Vision-Instruct

Figure 2: Precision–Recall comparison of SEFIR across four backbones.

where $\gamma \geq 0$ controls down-weighting of easy examples. We use this objective because multimodal safety data can be imbalanced and may contain hard boundary cases where visual and textual cues are not well aligned.

Training proceeds in stages. First, the frozen backbone produces component summaries $\mathcal{R}(x)$. Second, each diagnostic scorer u_c is trained on $\mathcal{D}_{\text{fit}}$ and used to construct S_c. Third, $\mathcal{D}_{\text{score}}$ is used only to compute held-out utility q_c and aggregation weight α_c. Finally, with $\{S_c, \alpha_c\}_{c\in\mathcal{C}}$ fixed, f_ψ is trained on $z(x)$ using global safety labels. The backbone remains frozen throughout. The construction-time trainable parameter count is

$$P_{\text{train}} = \sum_{c\in\mathcal{C}} (d_c + 1) + P(f_\psi), \tag{24}$$

The first term counts diagnostic scorer parameters; the second counts prediction head parameters. During construction, diagnostic scorers select coordinates and estimate held-out utility; they are not queried once $\{S_c, \alpha_c\}_{c\in\mathcal{C}}$ are fixed. At deployment, inference uses the frozen backbone and required hidden states, plus only the selected coordinate sets, aggregation weights, and f_ψ. Since $P_{\text{train}} \ll |\Theta|$ and Θ is frozen, SEFIR avoids full-model safety realignment. Beyond the frozen backbone forward pass, inference adds component pooling, sparse coordinate extraction, score scaling, and the prediction head:

$$\mathcal{O}_{\text{extra}}(x) = \mathcal{O}\left(\sum_{c\in\mathcal{C}} T_c d_c + \sum_{c\in\mathcal{C}} |S_c| + \text{FLOPs}(f_\psi)\right). \tag{25}$$

This counts computation after required hidden states become available; measured latency and memory overhead are reported in the experiments. SEFIR thus reorganizes the frozen MLLM's existing representations into a compact, held-out-utility-guided safety decision space, keeping the backbone fixed.

4 Experiments

4.1 Experimental Setup

Evaluation datasets. We evaluate on seven multimodal safety benchmarks—FalconXue et al. (2025), Hateful MemesKiela et al. (2020), HoliSafeLee et al. (2025), SafeTagRong et al. (2025),

SARCNetYue et al. (2024), VLGuardZong et al. (2024), and VLSUPalaskar et al. (2025)—covering image-, text-, and joint-semantics-based unsafe content.

Evaluation metrics. We report Macro-F1, Precision, and Recall. Macro-F1 is the primary metric because it reflects balanced performance under class imbalance; Precision and Recall expose the trade-off between false alarms and missed unsafe cases.

Compared models. We compare SEFIR with general instruction-tuned MLLMs, safety-aligned generative models, and guard-style safety models. The evaluated backbones include Llama-3.2-11B-Vision-InstructGrattafiori et al. (2024), Qwen2-VL-7B-InstructWang et al. (2024), Qwen3-VL-4B-InstructBai et al. (2025), and Qwen3-VL-8B-InstructBai et al. (2025); safety-oriented baselines include SafeGRPO variantsRong et al. (2025), Llama-Guard-3-11B-VisionChi et al. (2024), and Qwen2-VL-7B-VLGuardZong et al. (2024); Wang et al. (2024). This comparison covers both broadly available instruction-tuned checkpoints and models with explicit multimodal safety training. Additional details on splits, hyperparameters, seed, compute resources, baseline evaluation, parsing, and metrics are provided in Appendix A.

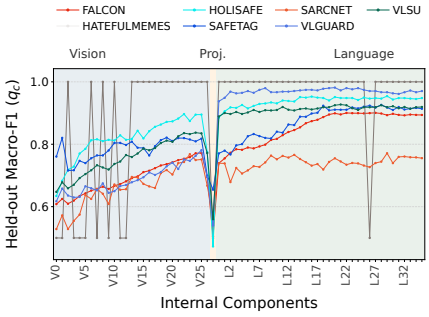


Figure 3: Layer-wise held-out component scores q_c of Qwen3-VL-8B-Instruct.

4.2 Experimental Results

Frozen internal evidence supports plug-in safety detection. We evaluate SEFIR on four frozen MLLM backbones and compare it with matched safety-aligned or guard-style baselines. As shown in Table 1, SEFIR improves average Macro-F1 from 72.86 to 89.56 on Qwen3-VL-4B, 58.68 to 89.28 on Qwen2-VL-7B, 67.20 to 90.02 on Qwen3-VL-8B, and 60.73 to 85.11 on Llama-3.2-11B, with consistent gains across all seven benchmarks. Figure 2 further shows that the improvements move all backbones toward higher iso-F1 contours in precision–recall space, rather than trading Precision for Recall. These results indicate that frozen MLLM representations already contain safety-relevant internal evidence, which SEFIR can organize into an effective plug-in detector without backbone-level safety realignment.

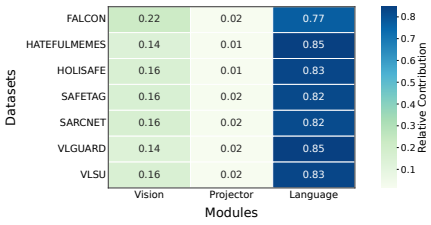


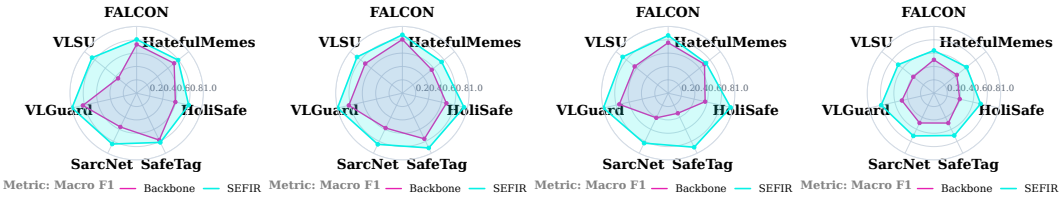
Figure 4: Module-level contribution of Qwen3-VL-8B-Instruct.

Safety evidence is unevenly distributed across internal components. We examine this distribution on the frozen Qwen3-VL-8B-Instruct backbone by computing each component's held-out utility q_c on $\mathcal{D}_{\text{score}}$ with the sparse truncated scorer u_c^S. Figure 3 shows clear variation across layers and datasets: deeper vision layers tend to provide stronger visual evidence, language layers remain consistently informative, and the projector bridges the two. Figure 4 further shows non-uniform module contributions under score-guided aggregation. These patterns support SEFIR's evidence-organization view and motivate component-aware sparse selection and held-out utility-guided fusion over the final representation alone or uniformly concatenated hidden states.

Table 2: Ablation of key SEFIR design choices on Qwen3-VL-4B-Instruct (Macro-F1, %).

Variant	Falcon	H. Memes	SARCNet	Avg.	Δ vs. Full
Full SEFIR	88.31	79.44	85.39	84.38	–
Mean pooling	76.14	71.46	78.33	75.31	−9.07
Random coordinate selection	83.21	74.81	82.46	80.16	−4.22
Uniform fusion	84.15	75.35	82.65	80.72	−3.66
No sparse selection	84.32	75.56	83.78	81.22	−3.16

Ablations isolate the contribution of key design choices. On Qwen3-VL-4B-Instruct under matched settings, Table 2 shows that mean pooling reduces average Macro-F1 by 9.07 points; random coordinate selection, uniform fusion, and no sparse selection reduce it by 4.22, 3.66, and 3.16 points, respectively. These losses isolate the contribution of each design choice.



(a) GLM-4.1V-9B-Thinking (b) CapRL-InternVL3.5-8B (c) MiniCPM-V-4.5 (d) LLaVA-OneVision-1.5-8B-Instruct

Figure 5: Original backbones vs. backbones equipped with SEFIR on four multimodal models.

Table 3: Falcon-trained SEFIR Macro-F1 (%) on six unseen benchmarks without target tuning; best results are highlighted.

Backbone	H. Memes	HoliSafe	SafeTag	SARCNet	VLGuard	VLSU	Avg.
Qwen2-VL-7B-Instruct + SEFIR	69.99	58.05	80.24	62.59	78.36	70.89	70.02
Qwen3-VL-4B-Instruct + SEFIR	63.88	56.23	79.66	64.91	75.67	70.40	68.46
Qwen3-VL-8B-Instruct + SEFIR	67.54	57.46	83.29	60.93	71.94	74.49	69.28
Llama-3.2-11B-Vision-Instruct + SEFIR	56.51	54.70	75.70	54.88	65.01	68.19	62.50

Evidence organization generalizes beyond the training setting but remains dataset-sensitive. We evaluate generalization along two axes: applying the same SEFIR construction to unseen backbones, and transferring a Falcon-trained detector to unseen safety benchmarks. For backbone transfer, we test GLM-4.1V-9B-ThinkingHong et al. (2025), CapRL-InternVL3.5-8BXing et al. (2025); Wang et al. (2025a), MiniCPM-V-4.5Yu et al. (2025), and LLaVA-OneVision-1.5-8B-InstructAn et al. (2025), comparing each original model with its SEFIR-enhanced variant on the seven benchmarks. Figure 5 shows consistent gains across these additional backbones, suggesting that the benefit of organizing internal evidence extends beyond the four primary backbones evaluated above. For dataset transfer, we train SEFIR only on Falcon and evaluate it without target-dataset tuning on six unseen benchmarks across four backbones. Table 3 reports average out-of-domain Macro-F1 scores from 62.50% to 70.02%, indicating that part of the selected safety evidence remains reusable beyond the source dataset. The variation across targets indicates that cross-dataset robustness remains sensitive to risk taxonomy and distribution shifts.

Training and Inference Efficiency. Figure 6 compares SEFIR's trainable-parameter ratio and inference FLOPs overhead on the four primary backbones. The trainable component accounts for only 0.70–1.08% of backbone parameters, while the additional computation is 3.39×10^{-4}–5.25×10^{-4}% of backbone FLOPs. These measurements show that SEFIR leaves the backbone frozen while adding little trainable capacity or inference cost.

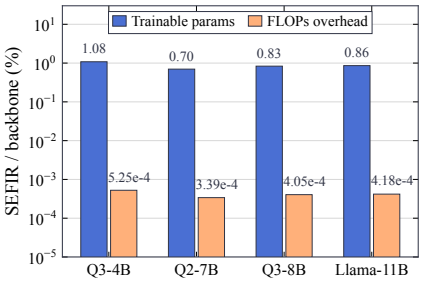


Figure 6: Relative parameter and inference FLOPs overhead of SEFIR.

5 Conclusion

Multimodal safety detection need not rely on backbone-level realignment, particularly when image–text risk is already reflected in the internal states of a frozen MLLM. SEFIR builds on this premise by treating visual, projector, and language components as uneven sources of safety-relevant evidence. It converts these heterogeneous states into a compact decision space through component summarization, sparse coordinate selection, and held-out utility-guided fusion. Across evaluated benchmarks and backbones, this plug-in detector improves over matched safety-oriented baselines while keeping the backbone fixed and adding limited training and inference cost. The component-utility and supervision analyses support component-aware evidence selection and weighting, rather than relying on a final representation or uniform hidden-state concatenation. These findings position internal evidence fusion as a lightweight complement to backbone-level multimodal safety alignment, with cross-dataset robustness still sensitive to risk taxonomy and distribution shifts.

AI USE STATEMENT

Generative AI tools were used only to refine the manuscript's wording, grammar, and LaTeX presentation. They were not used to develop the proposed method, conduct experiments, generate or modify results, or draw scientific conclusions. All AI-assisted revisions were reviewed by the authors, who take full responsibility for the final manuscript.

REPRODUCIBILITY STATEMENT

The supplementary material includes the implementation of SEFIR, model- and dataset-specific configuration files, data-processing utilities, baseline evaluation scripts, and the software requirements used in our experiments. Appendix A documents the dataset construction and label mappings, data partitions, training configuration, compute resources, baseline inference interfaces, output parsing, and metric computation. Appendix B provides matched supervised controls, ablations, training-seed analysis, robustness evaluations, and hyperparameter sensitivity results. All model construction and training steps use the training data and its internal splits, while the validation split is reserved for final evaluation. Together, the supplementary code and appendices document the training and evaluation workflow used for the reported results.

REFERENCES

Xiang An, Yin Xie, Kaicheng Yang, Wenkang Zhang, Xiuwei Zhao, Zheng Cheng, Yirui Wang, Songcen Xu, Changrui Chen, Didi Zhu, et al. Llava-onevision-1.5: Fully open framework for democratized multimodal training. *arXiv preprint arXiv:2509.23661*, 2025.

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, et al. Qwen3-vl technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J Pappas, Florian Tramer, et al. Jailbreakbench: An open robustness benchmark for jailbreaking large language models. *Advances in Neural Information Processing Systems(NeurIPS)*, 37:55005–55029, 2024.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. Jailbreaking black box large language models in twenty queries. In *2025 IEEE Conference on Secure and Trustworthy Machine Learning (SaTML)*, pp. 23–42. IEEE, 2025.

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

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale N Fung, and Steven Hoi. Instructblip: Towards general-purpose vision-language models with instruction tuning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 49250–49267, 2023.

Mostafa Dehghani, Anurag Arnab, Lucas Beyer, Ashish Vaswani, and Yi Tay. The efficiency misnomer. *arXiv preprint arXiv:2110.12894*, 2021.

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

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

Dan Hendrycks and Thomas Dietterich. Benchmarking neural network robustness to common corruptions and perturbations. In *International Conference on Learning Representations(ICLR)*, 2019.

Wenyi Hong, Wenmeng Yu, Xiaotao Gu, Guo Wang, Guobing Gan, Haomiao Tang, Jiale Cheng, Ji Qi, Junhui Ji, Lihang Pan, et al. Glm-4.5 v and glm-4.1 v-thinking: Towards versatile multimodal reasoning with scalable reinforcement learning. *arXiv preprint arXiv:2507.01006*, 2025.

Xuhao Hu, Dongrui Liu, Hao Li, Xuan-Jing Huang, and Jing Shao. Vlsbench: Unveiling visual leakage in multimodal safety. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (ACL)*, pp. 8285–8316, 2025.

Yilei Jiang, Xinyan Gao, Tianshuo Peng, Yingshui Tan, Xiaoyong Zhu, Bo Zheng, and Xiangyu Yue. Hiddendetect: Detecting jailbreak attacks against large vision-language models via monitoring hidden states. *arXiv preprint arXiv:2502.14744*, 3(5), 2025.

Difan Jiao, Yilun Liu, Ye Yuan, Zhenwei Tang, Linfeng Du, Haolun Wu, and Ashton Anderson. Llm safety from within: Detecting harmful content with internal representations. *arXiv preprint arXiv:2604.18519*, 2026.

Douwe Kiela, Hamed Firooz, Aravind Mohan, Vedanuj Goswami, Amanpreet Singh, Pratik Ringshia, and Davide Testuggine. The hateful memes challenge: Detecting hate speech in multimodal memes. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, pp. 2611–2624, 2020.

Wonjae Kim, Bokyung Son, and Ildoo Kim. Vilt: Vision-and-language transformer without convolution or region supervision. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 5583–5594. PMLR, 2021.

Youngwan Lee, Kangsan Kim, Kwanyong Park, Ilcahe Jung, Soojin Jang, Seanie Lee, Yong-Ju Lee, and Sung Ju Hwang. Holisafe: Holistic safety benchmarking and modeling for vision-language model. *arXiv preprint arXiv:2506.04704*, 2025.

Junnan Li, Ramprasaath Selvaraju, Akhilesh Gotmare, Shafiq Joty, Caiming Xiong, and Steven Chu Hong Hoi. Align before fuse: Vision and language representation learning with momentum distillation. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 34, pp. 9694–9705, 2021.

Junnan Li, Dongxu Li, Caiming Xiong, and Steven Hoi. Blip: Bootstrapping language-image pre-training for unified vision-language understanding and generation. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 12888–12900. PMLR, 2022.

Junnan Li, Dongxu Li, Silvio Savarese, and Steven Hoi. Blip-2: Bootstrapping language-image pre-training with frozen image encoders and large language models. In *Proceedings of the International Conference on Machine Learning (ICML)*, pp. 19730–19742. PMLR, 2023.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Proceedings of the Conference on Neural Information Processing Systems (NeurIPS)*, volume 36, pp. 34892–34916, 2023.

Jianyu Liu, Hangyu Guo, Ranjie Duan, Xingyuan Bu, Yancheng He, Shilong Li, Hui Huang, Jiaheng Liu, Yucheng Wang, Chenchen Jing, et al. Dream: Disentangling risks to enhance safety alignment in multimodal large language models. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies (NAACL)*, pp. 12097–12118, 2025a.

Qin Liu, Chao Shang, Ling Liu, Nikolaos Pappas, Jie Ma, Neha Anna John, Srikanth Doss, Lluis Marquez, Miguel Ballesteros, and Yassine Benajiba. Unraveling and mitigating safety alignment degradation of vision-language models. In *Proceedings of Findings of the Association for Computational Linguistics(ACL)*, pp. 3631–3643, 2025b.

Xin Liu, Yichen Zhu, Jindong Gu, Yunshi Lan, Chao Yang, and Yu Qiao. Mm-safetybench: A benchmark for safety evaluation of multimodal large language models. In *Proceedings of the European Conference on Computer Vision (ECCV)*, pp. 386–403. Springer, 2024.

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

Lingrui Mei, Shenghua Liu, Yiwei Wang, Baolong Bi, Ruibin Yuan, and Xueqi Cheng. Hiddenguard: Fine-grained safe generation with specialized representation router. *arXiv preprint arXiv:2410.02684*, 2024.

Shruti Palaskar, Leon Gatys, Mona Abdelrahman, Mar Jacobo, Larry Lindsey, Rutika Moharir, Gunnar Lund, Yang Xu, Navid Shiee, Jeffrey Bigham, et al. Vlsu: Mapping the limits of joint multimodal understanding for ai safety. *arXiv preprint arXiv:2510.18214*, 2025.

Tobia Poppi, Tejaswi Kasarla, Pascal Mettes, Lorenzo Baraldi, and Rita Cucchiara. Hyperbolic safety-aware vision-language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 4222–4232, 2025.

Xiangyu Qi, Kaixuan Huang, Ashwinee Panda, Peter Henderson, Mengdi Wang, and Prateek Mittal. Visual adversarial examples jailbreak aligned large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, volume 38, pp. 21527–21536, 2024.

Xinkuan Qiu, Meina Kan, Yongbin Zhou, and Shiguang Shan. Benchmarking multimodal large language models against image corruptions. In *2025 IEEE/CVF International Conference on Computer Vision (ICCV)*, pp. 9014–9023. IEEE, 2025.

Xuankun Rong, Wenke Huang, Tingfeng Wang, Daiguo Zhou, Bo Du, and Mang Ye. Safegrpo: Self-rewarded multimodal safety alignment via rule-governed policy optimization. *arXiv preprint arXiv:2511.12982*, 2025.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. *arXiv preprint arXiv:2307.14539*, 2023.

Yi-Lin Sung, Jaemin Cho, and Mohit Bansal. Vl-adapter: Parameter-efficient transfer learning for vision-and-language tasks. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition (CVPR)*, pp. 5227–5237, 2022.

Mingxu Tao, Quzhe Huang, Kun Xu, Liwei Chen, Yansong Feng, and Dongyan Zhao. Probing multimodal large language models for global and local semantic representations. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation (LREC-COLING)*, pp. 13050–13056, 2024.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, et al. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, Xingguang Wei, Zhaoyang Liu, Linglin Jing, Shenglong Ye, Jie Shao, et al. Internvl3. 5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025a.

Yanbo Wang, Jiyang Guan, Jian Liang, and Ran He. Do we really need curated malicious data for safety alignment in multi-modal large language models? In *Proceedings of the Computer Vision and Pattern Recognition Conference (CVPR)*, pp. 19879–19889, 2025b.

Long Xing, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jianze Liang, Qidong Huang, Jiaqi Wang, Feng Wu, and Dahua Lin. Caprl: Stimulating dense image caption capabilities via reinforcement learning. *arXiv preprint arXiv:2509.22647*, 2025.

Shicheng Xu, Liang Pang, Yunchang Zhu, Huawei Shen, and Xueqi Cheng. Cross-modal safety mechanism transfer in large vision-language models. *arXiv preprint arXiv:2410.12662*, 2024.

Qi Xue, Minrui Jiang, Runjia Zhang, Xiurui Xie, Pei Ke, and Guisong Liu. Falcon: A cross-modal evaluation dataset for comprehensive safety perception. *arXiv preprint arXiv:2509.23783*, 2025.

Zonghao Ying, Aishan Liu, Siyuan Liang, Lei Huang, Jinyang Guo, Wenbo Zhou, Xianglong Liu, and Dacheng Tao. Safebench: A safety evaluation framework for multimodal large language models. *International Journal of Computer Vision*, 134(1):18, 2026.

Tianyu Yu, Zefan Wang, Chongyi Wang, Fuwei Huang, Wenshuo Ma, Zhihui He, Tianchi Cai, Weize Chen, Yuxiang Huang, Yuanqian Zhao, et al. Minicpm-v 4.5: Cooking efficient mllms via architecture, data, and training recipe. *arXiv preprint arXiv:2509.18154*, 2025.

Tan Yue, Xuzhao Shi, Rui Mao, Zonghai Hu, and Erik Cambria. Sarcnet: a multilingual multimodal sarcasm detection dataset. In *Proceedings of the 2024 Joint International Conference on Computational Linguistics, Language Resources and Evaluation*, pp. 14325–14335, 2024.

Yongting Zhang, Lu Chen, Guodong Zheng, Yifeng Gao, Rui Zheng, Jinlan Fu, Zhenfei Yin, Senjie Jin, Yu Qiao, Xuanjing Huang, et al. Spa-vl: A comprehensive safety preference alignment dataset for vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 19867–19878, 2025.

Kaiwen Zhou, Chengzhi Liu, Xuandong Zhao, Anderson Compalas, Dawn Song, and Xin Eric Wang. Multimodal situational safety. *arXiv preprint arXiv:2410.06172*, 2024.

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy Hospedales. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. *arXiv preprint arXiv:2402.02207*, 2024.

A Additional Experimental Details

Table 4: Dataset splits.

Dataset	Train	Val.
Falcon	43,832	1,762
Hateful Memes	7,650	850
HoliSafe	3,225	806
SafeTag	2,634	659
SARCNet	1,998	671
VLGuard	2,977	1,558
VLSU	6,430	1,295

Table 5: Core training configuration.

Parameter	Value
Scorer learning rate	10^{-3}
ℓ_1 coefficient λ_1	10^{-4}
Selection threshold η	0.7
MLP hidden dimensions	$(1024, 512, 256)$
MLP dropout	0.1
Classifier learning rate	10^{-3}
Weight decay	5×10^{-5}
Seed	42

A.1 Data Processing

We use seven multimodal safety datasets: Falcon, Hateful Memes, HoliSafe, SafeTag, SARCNet, VLGuard, and VLSU. For Falcon, SARCNet, and VLGuard, we keep the train–validation partitions released with the datasets.

For the remaining datasets, which do not provide the same predefined split, we construct sample-level train–validation partitions from the available data, assigning 80% of the samples to $\mathcal{D}_{\text{train}}$ and 20% to $\mathcal{D}_{\text{val}}$. The resulting dataset sizes are reported in Table 4.

All seven benchmarks use modality-specific diagnostic targets together with a global pair-level target. For Falcon, Hateful Memes, HoliSafe, SafeTag, SARCNet, VLGuard, and VLSU, fixed mappings from the released annotations or safety taxonomies define the image-side label y^v and text-side label y^t, while the benchmark's binary harmfulness label is retained as y. Visual component scorers are fitted with y^v, projector and language component scorers with y^t, and the final prediction head with y. All mappings are fixed before model fitting and remain unchanged across training and evaluation. Consequently, none of the seven reported benchmarks uses the fallback $y^v = y^t = y$; this fallback is reserved for datasets that provide only global labels.

The validation split $\mathcal{D}_{\text{val}}$ in Table 4 is reserved exclusively for held-out evaluation. No sample from this split is used to fit the sparse diagnostic scorers, select retained coordinates, estimate component utilities, tune aggregation weights, or train the final prediction head. All construction steps are performed within $\mathcal{D}_{\text{train}}$.

Within $\mathcal{D}_{\text{train}}$, we create an auxiliary scoring split following the training procedure in Section 3:

$$\mathcal{D}_{\text{train}} = \mathcal{D}_{\text{fit}} \dot{\cup} \mathcal{D}_{\text{score}}.$$

Unless otherwise specified, 80% of $\mathcal{D}_{\text{train}}$ is used as $\mathcal{D}_{\text{fit}}$ and the remaining 20% as $\mathcal{D}_{\text{score}}$; we use a stratified split whenever both safety classes are present. The split $\mathcal{D}_{\text{fit}}$ is used to optimize the sparse diagnostic scorer of each component and to determine the retained coordinate set S_c. After the coordinate sets are fixed, $\mathcal{D}_{\text{score}}$ is used only to compute the held-out utility score q_c, defined as the held-out Macro-F1 of the truncated component scorer u_c^S under the global safety label. These scores determine the fixed aggregation weights used in score-guided fusion.

After $\{S_c, \alpha_c\}_{c\in\mathcal{C}}$ have been determined, the final prediction head is trained on $\mathcal{D}_{\text{fit}}$ using the global safety labels. Thus, $\mathcal{D}_{\text{score}}$ provides held-out estimates for component weighting but does not update any scorer or final-head parameters, while $\mathcal{D}_{\text{val}}$ remains untouched until evaluation.

A.2 Training Configuration

All multimodal backbones are frozen during training. The optimization is restricted to the sparse component scorers and the final prediction head; aggregation weights are computed from held-out utility scores and then fixed. For component scoring, we use a learning rate of 10^{-3} and apply an ℓ_1 sparsity penalty with coefficient $\lambda_1 = 10^{-4}$. The coordinate retention threshold is set to $\eta = 0.7$, which determines the retained feature subset for each internal component.

The final prediction head is implemented as a multilayer perceptron with hidden dimensions $(1024, 512, 256)$ and dropout rate 0.1. It is trained with learning rate 10^{-3} and weight decay 5×10^{-5}. We report one run per configuration using random seed 42. The core hyperparameters are summarized in Table 5. All experiments are implemented in the PyTorch framework and conducted on a workstation equipped with two NVIDIA GeForce RTX 5090 GPUs.

A.3 Efficiency Accounting

Figure 6 reports parameter and FLOPs ratios relative to the corresponding frozen backbone. Let P_Θ and F_Θ denote the backbone parameter count and the FLOPs of one backbone forward pass, respectively. For each component c, let $k_c = |S_c|$ be the number of retained coordinates and let $K = \sum_{c\in\mathcal{C}} k_c$ be the fused input dimension. The prediction head has widths

$$(m_0, m_1, m_2, m_3, m_4) = (K, 1024, 512, 256, 2). \tag{26}$$

We count one scalar multiplication and one scalar addition as two FLOPs. Under this convention, an affine layer from $m_{\ell-1}$ inputs to m_ℓ outputs contributes approximately $2m_{\ell-1}m_\ell$ FLOPs. Bias additions and element-wise nonlinearities are lower-order terms and are omitted consistently. The fixed component weights require one multiplication per retained coordinate, giving the deployed detector-head estimate

$$F_{\text{head}} = K + 2\sum_{\ell=1}^{4} m_{\ell-1}m_\ell. \tag{27}$$

Diagnostic scorers are excluded from F_{head} because they are used to construct S_c and α_c during training and are not evaluated after these quantities are fixed. The ratios shown in Figure 6 are

$$\rho_F = 100\frac{F_{\text{head}}}{F_\Theta}, \qquad \rho_P = 100\frac{P_{\text{train}}}{P_\Theta}, \tag{28}$$

where

$$P_{\text{train}} = \sum_{c\in\mathcal{C}}(d_c + 1) + \sum_{\ell=1}^{4}(m_{\ell-1}m_\ell + m_\ell) \tag{29}$$

includes the diagnostic scorers and prediction head. Applying these equations to the four primary backbones gives the 0.70–1.08% parameter ratios and 3.39×10^{-4}–$5.25 \times 10^{-4}\%$ detector-head FLOPs ratios reported in Figure 6.

The full arithmetic added after the backbone also includes activation-magnitude pooling. For a component with T_c positions and hidden width d_c, computing the position norms and their weighted sum contributes approximately $4T_cd_c$ multiply-add FLOPs, while the softmax contributes a lower-order $\mathcal{O}(T_c)$ term. The input-dependent total is therefore

$$F_{\text{extra}}(x) \approx 4\sum_{c\in\mathcal{C}} T_c d_c + \mathcal{O}\left(\sum_{c\in\mathcal{C}} T_c\right) + F_{\text{head}}. \tag{30}$$

Figure 6 isolates F_{head} because the pooling term varies with image resolution and sequence length. Coordinate gathering and hidden-state access are memory operations and are not assigned arithmetic FLOPs. FLOPs provide a hardware-independent operation count but do not capture memory traffic, kernel-launch overhead, or wall-clock latency; we therefore report the plotted values as detector-head estimates rather than end-to-end runtime measurements Dehghani et al. (2021).

A.4 Evaluation Protocol for Multimodal Models

All models are evaluated on the validation split $\mathcal{D}_{\text{val}}$ defined in Table 4. This split is held out from all training and construction steps: it is not used to fit SEFIR, estimate component utilities, select retained coordinates, tune prompts, calibrate thresholds, or modify any baseline. Each evaluation example contains the same image–text input and the same binary safety label, with 1 denoting unsafe content and 0 denoting safe content.

For safety-aligned models that provide released evaluation code, we follow the official open-source path. SafeGRPO checkpoints are evaluated with the official SafeGRPO repository.[1] Guard-style models are evaluated through their released guard interface rather than through a generic instruction-following prompt. In particular, Llama-Guard-3-11B-Vision is run with its official multimodal interface.[2] The implementation uses the HuggingFace path based on `MllamaForConditionalGeneration` and `AutoProcessor.apply_chat_template`. Since this model judges whether the last conversation message violates its safety policy, the conversation is formatted so that the multimodal content under evaluation is the target of the guard decision.

For open-source instruction-tuned MLLMs without a safety-specific evaluator, we use a unified `lmms-eval`-compatible protocol.[3] Each dataset is converted into the same task format, exposing the image, text prompt, optional response field, sample identifier, and binary label. The task uses deterministic generation with sampling disabled and evaluates all checkpoints through the same task definition, prediction parser, and metric aggregation. Model-specific handling is restricted to the released inference interface of each checkpoint, including its processor, tokenizer, image preprocessing, chat template, remote-code loading, and device placement. These compatibility steps do not update model weights, change labels, or introduce dataset-specific post-processing.

Generated outputs are converted to binary predictions by a fixed parser shared across the unified evaluation. The parser first checks structured outputs and explicit answer fields, then reads concise safety labels such as `safe` and `unsafe`. If a model returns additional text, only the parsed safety judgment is used. Outputs that merely echo the prompt or contain both candidate labels without a valid final decision are not treated as valid safety predictions. No generated prediction is manually edited.

All reported scores are computed with the same metric implementation. We report Accuracy, Macro-F1, Precision, Recall, AUROC, and Specificity, treating unsafe as the positive class for Precision and Recall. Macro-F1 is used as the primary metric because the validation splits may be class-imbalanced. For auditability, each run stores per-sample records containing the sample identifier, ground-truth label, predicted label, unsafe score, raw model output, and parsing status. The final numbers are therefore derived from the same held-out samples, label mapping, decoding rule, parser, and metric computation, while preserving official inference paths when models provide them.

B Supplementary Experimental Analysis

This section examines SEFIR from complementary empirical perspectives: supervised representation baselines, design controls, training stability, input perturbations, component utility, modality-aware supervision, and hyperparameter sensitivity. All analyses follow the same frozen-backbone construction and held-out evaluation protocol described in Appendix A.

B.1 Supervised Representation Baselines

To quantify the value of internal-evidence construction relative to conventional supervised probing, we train two probes using the same frozen backbone, labels, data splits, and optimization and evaluation protocol as SEFIR. The final-layer linear probe operates on the pooled final representation, while the hidden-state MLP feeds pooled internal states directly to a high-capacity classifier. Table 6

[1] `https://github.com/XuankunRong/SafeGRPO`

[2] `https://github.com/meta-llama/PurpleLlama/blob/main/Llama-Guard3/11B-vision/README.md`

[3] `https://github.com/evolvinglmms-lab/lmms-eval`

Table 6: Macro-F1 (%) of supervised representation baselines on frozen Qwen backbones.

Backbone	Dataset	Final-layer linear	Direct hidden-state MLP	SEFIR	Δ Linear	Δ MLP
Qwen3-VL-4B	Falcon	84.88	83.75	88.31	+3.43	+4.56
Qwen3-VL-4B	Hateful Memes	76.13	78.00	79.44	+3.31	+1.44
Qwen3-VL-4B	SARCNet	81.36	80.72	85.39	+4.03	+4.67
Qwen3-VL-8B	Falcon	85.96	84.95	88.42	+2.46	+3.47

Table 7: Complementary pooling and module ablations on Qwen3-VL-4B-Instruct (Macro-F1, %).

Variant	Falcon	H. Memes	SARCNet	Avg.	Δ vs. Full
Full SEFIR	88.31	79.44	85.39	84.38	–
Last-token pooling	84.36	76.06	84.72	81.71	−2.67
Max pooling	83.93	78.90	82.71	81.85	−2.53
Language-only	85.64	78.25	85.31	83.07	−1.31

compares the two probes with SEFIR on three datasets for Qwen3-VL-4B-Instruct and on Falcon for Qwen3-VL-8B-Instruct, covering both dataset variation and model scale.

As shown in Table 6, SEFIR achieves the highest Macro-F1 in all four settings, exceeding the two probes by 1.44–4.67 points. The gains persist on Falcon when the backbone scales from Qwen3-VL-4B to Qwen3-VL-8B, and they also hold across Falcon, Hateful Memes, and SARCNet on Qwen3-VL-4B. On Qwen3-VL-4B/Falcon, SEFIR surpasses both the 123.06M-parameter hidden-state MLP and the compact 2,561-parameter final-layer probe. This pattern indicates that performance depends on how evidence is selected and organized across internal components, in addition to classifier capacity or access to target-dataset supervision.

B.2 Complementary Design Controls

Table 7 complements the ablations in Table 2 with alternative token summaries and a language-only control. Last-token pooling uses the final valid language token as its summary, and max pooling retains the largest activation along each coordinate. Language-only focuses the detector on language-layer evidence, whereas full SEFIR integrates the visual encoder, projector, and language model through the same selection and fusion pipeline.

Activation-magnitude pooling improves the three-dataset average by 2.67 points over last-token pooling and 2.53 points over max pooling, showing that response magnitude provides an effective salience signal for summarizing internal states. Combining visual and projector evidence with language representations adds 1.31 points over language-only detection. Table 2 reports 84.15 Macro-F1 for uniform fusion and 88.31 for held-out-utility weighting on Falcon. Under the same setting, sample-dependent attention obtains 86.89 Macro-F1. These results support all three stages of the construction: activation-aware summarization, multimodule evidence collection, and utility-guided fusion.

B.3 Comparison with an Internal-State Detector

We evaluate HiddenDetect Jiang et al. (2025) on Qwen3-VL-4B-Instruct using its published refusal-semantic score and select the binary threshold on $\mathcal{D}_{\text{score}}$. This comparison places SEFIR alongside an established hidden-state safety score and tests how effectively each approach captures the labels of multimodal safety benchmarks. Table 8 reports results on Falcon, Hateful Memes, and SARCNet, which span different forms of image–text safety reasoning.

Table 8 shows that SEFIR improves Macro-F1 over HiddenDetect by 17.61 points on Falcon, 16.05 points on Hateful Memes, and 31.47 points on SARCNet, yielding a 21.71-point average gain. The improvement is present on all three datasets and is largest on SARCNet, where harmfulness often depends on the interaction between visual context and textual intent. The result supports component-wise multimodal evidence and dataset-level supervision as effective ingredients for sample-level harmfulness prediction. SIREN Jiao et al. (2026) studies safety signals in text-only LLM states, while HiddenGuard Mei et al. (2024) uses token-level internal signals for detection and redaction during

Table 8: Macro-F1 (%) of HiddenDetect and SEFIR on Qwen3-VL-4B-Instruct.

Dataset	HiddenDetect	SEFIR
Falcon	70.70	88.31
Hateful Memes	63.39	79.44
SARCNet	53.92	85.39
Average	62.67	84.38

Table 9: Macro-F1 (%) across three training seeds on Qwen3-VL-4B-Instruct/Falcon.

Seed	Macro-F1 (%)
40	88.54
41	88.19
42	88.31
Mean $\pm$ sample standard deviation	88.35 ± 0.18

generation. Together with the supervised probes in Table 6, these methods provide complementary context for SEFIR's use of frozen internal representations.

B.4 Sensitivity to Training Randomness

We examine training variation on Qwen3-VL-4B-Instruct/Falcon using seeds 40, 41, and 42, with the data split and optimization settings held fixed. Each run reconstructs the diagnostic scorers, retained coordinates, utility weights, and prediction head under the same protocol, so the comparison captures variation introduced by model initialization and training order. Table 9 reports the three runs and their aggregate statistics.

Table 9 shows that Macro-F1 ranges from 88.19 to 88.54, a span of 0.35 points, with a mean of 88.35 and a sample standard deviation of 0.18. Seed 42 matches the 88.31 result in Table 1 and is 0.04 points from the three-seed mean, while seeds 40 and 41 remain within 0.19 points of that mean. The limited variation across these runs indicates stable performance under the tested random seeds.

B.5 Robustness to Common Image Corruptions

We apply ImageNet-C severity-3 defocus blur, JPEG compression, and Gaussian noise to the evaluation images Hendrycks & Dietterich (2019); Qiu et al. (2025). The three transformations probe loss of local sharpness, compression artifacts, and pixel-level noise while preserving the original text and safety label. The held-out samples, labels, and text remain unchanged, and the Qwen3-VL-4B-Instruct detector, including its diagnostic scorers, retained coordinates, utility weights, prediction head, and decision threshold, is fixed across all conditions. Table 10 compares each corrupted condition with the corresponding clean input on Falcon and SARCNet.

The results in Table 10 show that SEFIR preserves most of its clean-input performance across the tested corruptions. Averaged over the three transformations, Falcon changes from 88.31 to 88.11 Macro-F1 and SARCNet changes from 85.39 to 84.84. Defocus blur produces the largest reduction on both datasets, by 0.42 points on Falcon and 1.19 points on SARCNet, while JPEG compression changes Falcon by $+0.01$ points and Gaussian noise changes SARCNet by -0.02 points. Even in the lowest-scoring condition, the detector retains 98.6% of its clean Macro-F1. The consistency across three visually distinct transformations indicates that the selected evidence remains stable across multiple image-quality patterns.

B.6 Safety-Specific Input Perturbations

We evaluate text-based jailbreak prompts, image-embedded OCR prompt injection, and query-bounded score-adaptive selection, drawing on multimodal jailbreak, standardized jailbreak, typographic attack, and query-efficient black-box evaluation settings Luo et al. (2024); Chao et al. (2024);

Table 10: Macro-F1 (%) under common image corruptions. Δ is measured relative to the corresponding clean input.

Dataset	Condition	Macro-F1	Δ Clean
Falcon	Clean	88.31	–
	Defocus blur S3	87.89	−0.42
	JPEG compression S3	88.32	+0.01
	Gaussian noise S3	88.12	−0.19
SARCNet	Clean	85.39	–
	Defocus blur S3	84.20	−1.19
	JPEG compression S3	84.95	−0.44
	Gaussian noise S3	85.37	−0.02

Table 11: Macro-F1 (%) and attack success rate (ASR, %) under safety-specific input perturbations.

Condition	Falcon	SARCNet	Average
Clean	88.31 / –	85.39 / –	86.85 / –
OCR family, minimum-score selection (12 candidates)	86.78 / 2.99	84.66 / 1.79	85.72 / 2.39
Jailbreak family, minimum-score selection (6 candidates)	75.82 / 20.36	84.66 / 3.27	80.24 / 11.82
Score-adaptive selection ($q = 5$)	81.56 / 11.13	85.24 / 1.19	83.40 / 6.16
Score-adaptive selection ($q = 10$)	81.17 / 11.49	83.79 / 3.57	82.48 / 7.53

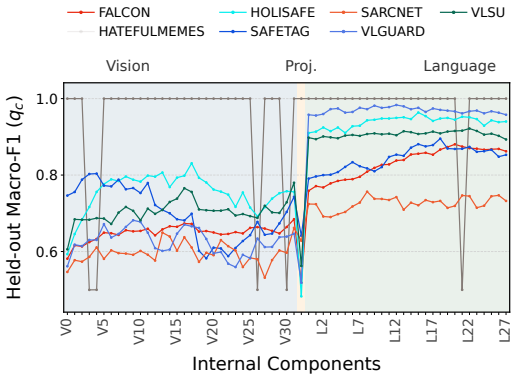
Gong et al. (2025); Chao et al. (2025). These conditions perturb the textual instruction, place adversarial instructions inside the image, or select candidates using the detector's output score. The Qwen3-VL-4B-Instruct detector is fixed throughout evaluation, with a prespecified candidate set and query order.

For each attack family, minimum-score selection chooses, for each harmful sample, the candidate with the lowest predicted unsafe probability. The OCR and jailbreak families contain 12 and 6 candidates, respectively. Query-bounded selection applies the same rule to the first q candidates in the fixed order. Attack success rate (ASR) is the fraction of harmful samples correctly detected on clean inputs but classified as safe after selection. Table 11 reports both Macro-F1 and ASR, capturing overall classification quality together with attack-specific evasion.

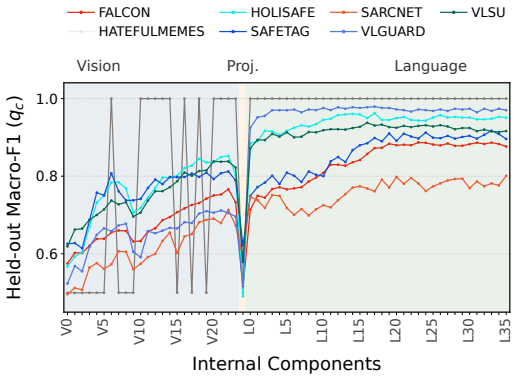
As reported in Table 11, OCR prompt injection has the smallest effect: SEFIR retains an average Macro-F1 of 85.72 and limits average ASR to 2.39%. Minimum-score jailbreak selection is the strongest tested condition, under which the detector maintains 80.24 Macro-F1 with an average ASR of 11.82%. Query-bounded selection produces intermediate results, with average Macro-F1 values of 83.40 at $q = 5$ and 82.48 at $q = 10$, and average ASR values of 6.16% and 7.53%. The gradual change from $q = 5$ to $q = 10$ is consistent with the attacker receiving a larger candidate budget. Across all four perturbation settings, the frozen detector continues to provide useful discrimination between safe and harmful inputs.

B.7 Component Utility across Qwen Backbones

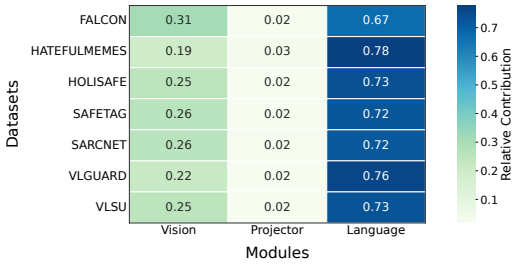
We extend the component-utility analysis from Qwen3-VL-8B-Instruct to Qwen2-VL-7B-Instruct and Qwen3-VL-4B-Instruct. For each backbone and dataset, the held-out utility q_c is computed on $\mathcal{D}_{\text{score}}$ using the sparse truncated scorer u_c^S, and the module contribution is obtained after score-guided aggregation. Figure 7 reports the layer-wise utility and module-level contribution for the two additional backbones. Both backbones reproduce the structured utility pattern observed on Qwen3-VL-8B-Instruct: informative components concentrate in particular layers, language representations contribute the largest share, and visual features provide a substantial complementary signal. The agreement across three Qwen backbones supports component-aware selection as a consistent way to organize distributed safety evidence.



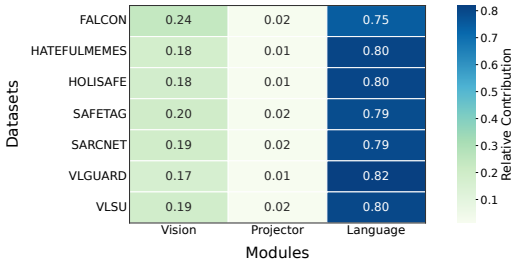
(a) Layer utility on Qwen2-VL-7B-Instruct.



(b) Layer utility on Qwen3-VL-4B-Instruct.



(c) Module contribution on Qwen2-VL-7B-Instruct.



(d) Module contribution on Qwen3-VL-4B-Instruct.

Figure 7: Component utility on Qwen2-VL-7B-Instruct and Qwen3-VL-4B-Instruct under the same held-out scoring and utility-guided aggregation procedure.

B.8 Modality-Decoupled Diagnostic Supervision across Qwen Backbones

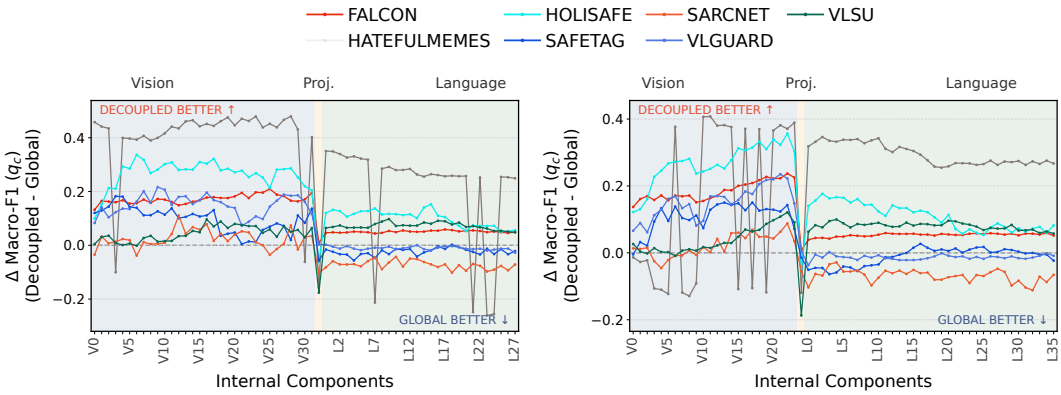
We compare the modality-decoupled supervision used in SEFIR with a global-label variant on Qwen2-VL-7B-Instruct, Qwen3-VL-4B-Instruct, and Qwen3-VL-8B-Instruct across all seven benchmarks. The decoupled variant uses y^v for visual components and y^t for projector and language components, while the global-label variant assigns the global safety label y to every component. For each backbone and dataset, held-out utility is evaluated on $\mathcal{D}_{\text{score}}$ using the sparse truncated scorer u_c^S.

Figure 8 reports the layer-wise held-out Macro-F1 differences after sparse coordinate selection, measured with the corresponding truncated scorer u_c^S on $\mathcal{D}_{\text{score}}$. The differences are predominantly positive across all three backbones. Visual labels provide targeted supervision for visual components, while text-side labels focus the projector and language components on text-associated risk. The agreement across model scales shows that component-aligned diagnostic targets improve the quality of the local safety evidence retained for downstream fusion. The final detector then integrates the selected evidence using the global safety label.

B.9 Sensitivity to Sparse-Selection Hyperparameters

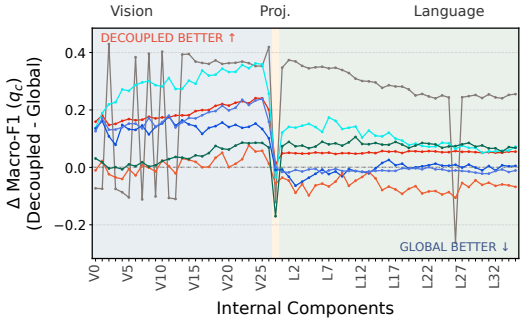
The sparse diagnostic selection stage in SEFIR involves two hyperparameters: the selection threshold η and the ℓ_1 coefficient λ_1. The threshold η determines the fraction of diagnostic weight mass retained within each component, while λ_1 controls the sparsity pressure imposed on the component-level diagnostic scorer. We conduct a sensitivity analysis by varying $\eta \in \{0.5, 0.6, 0.7, 0.8, 0.9\}$ and $\lambda_1 \in \{0, 10^{-5}, 10^{-4}, 10^{-3}, 10^{-2}\}$, with all other training and evaluation settings kept unchanged. The analysis is performed on Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct over SafeTag, Hateful Memes, and SARCNet. Figures 9 report the corresponding results for the two hyperparameters under the same evaluation protocol.

For the selection threshold, the tested values cover both more aggressive coordinate selection and more conservative evidence retention. Within this grid, $\eta = 0.7$ provides a consistently strong operating point across the evaluated backbones and datasets while preserving a compact evidence representation. We therefore use $\eta = 0.7$ in the main experiments to balance coordinate efficiency



(a) Qwen2-VL-7B-Instruct.

(b) Qwen3-VL-4B-Instruct.



(c) Qwen3-VL-8B-Instruct.

Figure 8: Layer-wise held-out Macro-F1 differences between modality-decoupled and global supervision across three Qwen backbones, measured after sparse coordinate selection with the corresponding truncated scorers.

and predictive performance. The smooth trends around this setting also show that the selection stage remains stable across a broad range of retained diagnostic weight mass.

For the sparsity coefficient, the grid compares the unregularized scorer with four nonzero levels of ℓ_1 regularization. The results show that $\lambda_1 = 10^{-4}$ provides a favorable balance between coordinate-level sparsity and predictive performance across the evaluated settings. We therefore use $\lambda_1 = 10^{-4}$ in the main experiments. Performance changes gradually around the selected values of both η and λ_1, supporting the stability of SEFIR's sparse diagnostic selection over practical hyperparameter ranges.

C Proof Details

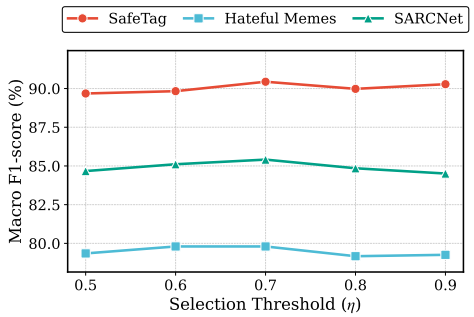
C.1 Proof of Theorem 1

Fix a component $c \in \mathcal{C}$ and an input x. The full diagnostic scorer is

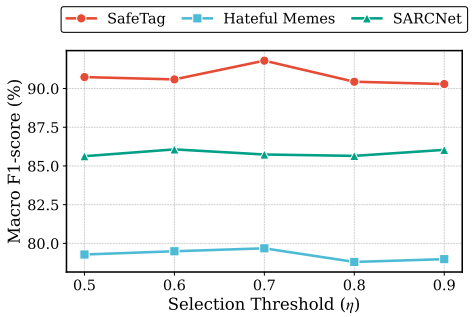
$$u_c(r_c(x)) = w_c^\top r_c(x) + b_c = \sum_{j=1}^{d_c} w_c(j) r_c(x)(j) + b_c. \tag{31}$$

The truncated scorer keeps only the coordinates selected by S_c:

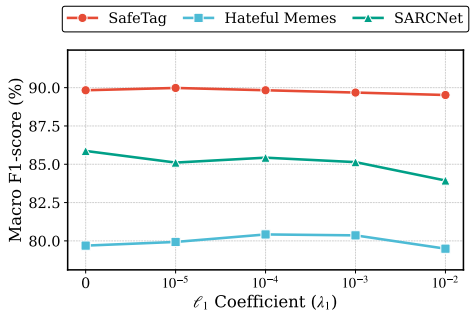
$$u_c^S(r_c(x)) = \sum_{j \in S_c} w_c(j) r_c(x)(j) + b_c. \tag{32}$$



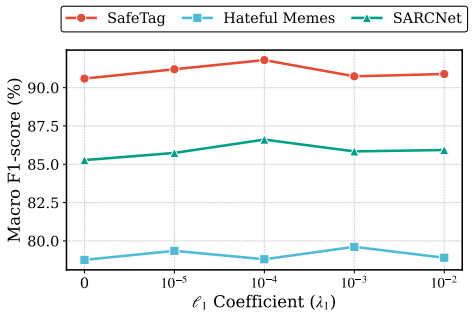
(a) Selection threshold η on Qwen3-VL-4B-Instruct.



(b) Selection threshold η on Qwen3-VL-8B-Instruct.



(c) ℓ_1 coefficient λ_1 on Qwen3-VL-4B-Instruct.



(d) ℓ_1 coefficient λ_1 on Qwen3-VL-8B-Instruct.

Figure 9: Sensitivity analysis of the sparse diagnostic selection hyperparameters. The first row varies the selection threshold η, and the second row varies the ℓ_1 coefficient λ_1, using Qwen3-VL-4B-Instruct and Qwen3-VL-8B-Instruct.

Subtracting the two expressions cancels the bias term and all selected coordinates:

$$u_c(r_c(x)) - u_c^S(r_c(x)) = \left(\sum_{j=1}^{d_c} w_c(j)[r_c(x)]_j + b_c\right) - \left(\sum_{j\in S_c} w_c(j)[r_c(x)]_j + b_c\right) \tag{33}$$

$$= \sum_{j=1}^{d_c} w_c(j)[r_c(x)]_j - \sum_{j\in S_c} w_c(j)[r_c(x)]_j \tag{34}$$

$$= \sum_{j\notin S_c} w_c(j)[r_c(x)]_j. \tag{35}$$

Thus sparsification changes the component logit only through the omitted coordinates.

Taking absolute values gives

$$\left|u_c(r_c(x)) - u_c^S(r_c(x))\right| = \left|\sum_{j\notin S_c} w_c(j) r_c(x)(j)\right|. \tag{36}$$

By the triangle inequality,

$$\left|\sum_{j\notin S_c} w_c(j) r_c(x)(j)\right| \le \sum_{j\notin S_c} |w_c(j) r_c(x)(j)| = \sum_{j\notin S_c} |w_c(j)|\,|r_c(x)(j)|. \tag{37}$$

The theorem assumes $\|r_c(x)\|_\infty \le B_c$, which implies

$$|r_c(x)(j)| \le B_c \qquad \text{for every } j \in \{1,\dots,d_c\}. \tag{38}$$

Applying this coordinate-wise bound to the omitted coordinates yields

$$\left|u_c(r_c(x)) - u_c^S(r_c(x))\right| \le B_c \sum_{j\notin S_c} |w_c(j)|. \tag{39}$$

It remains to bound the omitted absolute weight mass.

By definition,

$$\bar{w}_c(j) = |w_c(j)| \tag{40}$$

and π_c sorts coordinates in descending order of $\bar{w}_c(j)$, with deterministic tie breaking. The total diagnostic mass is

$$M_c = \sum_{j=1}^{d_c} |w_c(j)| = \|w_c\|_1. \tag{41}$$

The theorem assumes $M_c > 0$, so the ratio

$$Q_c(k) = \frac{\sum_{m=1}^{k} |w_c(\pi_c(m))|}{M_c} \tag{42}$$

is well-defined. The selected cardinality is

$$k_c^\star = \min\{k \mid Q_c(k) \geq \eta\}. \tag{43}$$

Therefore, by the definition of $k_c^\star$,

$$Q_c(k_c^\star) \geq \eta. \tag{44}$$

Multiplying both sides by M_c gives

$$\sum_{m=1}^{k_c^\star} |w_c(\pi_c(m))| \geq \eta M_c. \tag{45}$$

Since

$$S_c = \{\pi_c(1), \ldots, \pi_c(k_c^\star)\}, \tag{46}$$

the previous inequality can be written as

$$\sum_{j \in S_c} |w_c(j)| \geq \eta M_c. \tag{47}$$

The omitted mass is the total mass minus the selected mass:

$$\sum_{j \notin S_c} |w_c(j)| = \sum_{j=1}^{d_c} |w_c(j)| - \sum_{j \in S_c} |w_c(j)| \tag{48}$$

$$= M_c - \sum_{j \in S_c} |w_c(j)| \tag{49}$$

$$\leq M_c - \eta M_c \tag{50}$$

$$= (1 - \eta) M_c. \tag{51}$$

Using $M_c = \|w_c\|_1$, we obtain

$$\sum_{j \notin S_c} |w_c(j)| \leq (1 - \eta) \|w_c\|_1. \tag{52}$$

Substituting this bound into the earlier logit-difference bound gives

$$\left| u_c(r_c(x)) - u_c^S(r_c(x)) \right| \leq (1 - \eta) \|w_c\|_1 B_c. \tag{53}$$

This is the claimed bound.

C.2 Proof of Theorem 2

Fix an input x. For each component $c \in \mathcal{C}$, the retained block is

$$\tilde{r}_c(x) \in \mathbb{R}^{|S_c|}. \tag{54}$$

The fused representation is the block concatenation

$$z(x) = \bigoplus_{c \in \mathcal{C}} \alpha_c \tilde{r}_c(x). \tag{55}$$

Consider perturbations

$$\delta_c \in \mathbb{R}^{|S_c|} \qquad \text{for each } c \in \mathcal{C}. \tag{56}$$

After applying these perturbations to the retained blocks, the fused representation becomes

$$\bigoplus_{c\in\mathcal{C}} \alpha_c \left(\tilde{r}_c(x) + \delta_c\right). \tag{57}$$

Because the selected sets S_c and weights α_c are fixed after training, the difference between the perturbed and unperturbed fused representations is obtained block by block:

$$\bigoplus_{c\in\mathcal{C}} \alpha_c \left(\tilde{r}_c(x) + \delta_c\right) - \bigoplus_{c\in\mathcal{C}} \alpha_c \tilde{r}_c(x) \tag{58}$$

$$= \bigoplus_{c\in\mathcal{C}} \left[\alpha_c \left(\tilde{r}_c(x) + \delta_c\right) - \alpha_c \tilde{r}_c(x)\right] \tag{59}$$

$$= \bigoplus_{c\in\mathcal{C}} \alpha_c \delta_c. \tag{60}$$

The operator $\oplus$ concatenates disjoint coordinate blocks. Hence the squared Euclidean norm of the concatenated vector is the sum of the squared Euclidean norms of its blocks:

$$\left\| \bigoplus_{c\in\mathcal{C}} \alpha_c \left(\tilde{r}_c(x) + \delta_c\right) - \bigoplus_{c\in\mathcal{C}} \alpha_c \tilde{r}_c(x) \right\|_2^2 = \sum_{c\in\mathcal{C}} \|\alpha_c \delta_c\|_2^2 . \tag{61}$$

Since each α_c is a scalar,

$$\|\alpha_c \delta_c\|_2^2 = \sum_{m=1}^{|S_c|} \left(\alpha_c \delta_c(m)\right)^2 \tag{62}$$

$$= \alpha_c^2 \sum_{m=1}^{|S_c|} \delta_c(m)^2 \tag{63}$$

$$= \alpha_c^2 \|\delta_c\|_2^2. \tag{64}$$

Substituting this identity for every component gives

$$\left\| \bigoplus_{c\in\mathcal{C}} \alpha_c \left(\tilde{r}_c(x) + \delta_c\right) - \bigoplus_{c\in\mathcal{C}} \alpha_c \tilde{r}_c(x) \right\|_2^2 = \sum_{c\in\mathcal{C}} \alpha_c^2 \|\delta_c\|_2^2. \tag{65}$$

This is the claimed identity.