BEHAVIOR-ALIGNED VISUAL OPTIMIZATION FOR JAILBREAKING VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Gradient-based visual jailbreak attacks on vision-language models (VLMs) often optimize image perturbations for a fixed iteration budget using token-level target objectives that only indirectly capture harmful response behavior. We introduce **BAVO**, a behavior-aligned visual jailbreak framework that learns a lightweight behavior predictor from internal VLM representations to estimate response-level behavior. The predictor provides a differentiable behavioral signal for image optimization and enables instance-adaptive iterate selection without response generation or evaluation at every update. Across 917 samples from multimodal jailbreak benchmarks and four target VLMs, BAVO improves ELITE-based attack success rate (E-ASR) by 16.68–33.26 percentage points and HarmBench ASR by 10.25–24.64 points over affirmative-target PGD. BAVO also reduces the mean optimization steps from 300 to 32.7–161.2, yielding $1.8\times$–$10.4\times$ wall-clock speedups. The performance improvements persist across three defenses, while response-behavior analysis shows shifts from refusal and intermediate responses toward successful responses. These results demonstrate that aligning visual optimization with response-level behavior improves both attack effectiveness and efficiency.

1 INTRODUCTION

Vision-language models (VLMs) have become general-purpose multimodal systems for visual understanding, instruction following, and reasoning (Alayrac et al., 2022; Liu et al., 2023; OpenAI, 2024; Bai et al., 2025). As these models are deployed broadly, their safety mechanisms must remain reliable for both text and image instructions. However, the visual channel creates an additional input space in which harmful intent can be encoded or adversarially manipulated. Consequently, VLMs that refuse harmful textual requests may still generate unsafe content from multimodal inputs (Qi et al., 2024; Zong et al., 2024; Luo et al., 2024; Liu et al., 2024b). These vulnerabilities have motivated extensive work on multimodal jailbreaks and visual adversarial attacks.

In white-box visual attacks, a common strategy is to update the image so that the model assigns high likelihood to an affirmative or harmful target sequence, such as "Sure, here is" (Qi et al., 2024; Lin et al., 2026). This provides a stable gradient for image optimization, but matching a target sequence does not guarantee that the complete response fulfills the harmful request. A model can begin affirmatively yet produce a vague, evasive, or largely harmless answer. Modern evaluators therefore assess the complete generated response rather than refusal bypass alone (Mazeika et al., 2024; Chao et al., 2024; Souly et al., 2024; Lee et al., 2025). ELITE, for example, incorporates toxicity in multimodal response evaluation (Lee et al., 2025). We refer to the safety-relevant properties of the complete generated response as *response behavior*, including whether it complies with the request and the resulting content is substantively harmful. Optimization-based visual jailbreak attacks, often using Projected Gradient Descent (PGD), typically optimize a token-level target, whereas jailbreak success is ultimately evaluated based on the behavior of the complete response. We refer to this discrepancy as the *objective–behavior mismatch*.

Using a response evaluator directly as an attack objective is difficult because its score is obtained only after discrete text generation and cannot be backpropagated to image pixels. Repeatedly generating and judging a full response after every image update would also be computationally expensive. At the same time, prior work shows that safety and refusal information is reflected in internal model representations (Arditi et al., 2024; Rimsky et al., 2024; Zou et al., 2023; Burns et al., 2023; Hao

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.

et al., 2026; Song et al., 2025). This suggests that an evaluation of response behavior can instead be predicted from a hidden representation that remains differentiable with respect to the input image.

We implement this idea with a lightweight *behavior predictor*. For each target VLM, a regression probe is trained to estimate the ELITE score of the model's response from a selected internal representation and is frozen before attack optimization. During optimization, the predictor uses the internal representation to provide a differentiable signal that complements the conventional target-sequence objective, forming our *behavior-aligned visual optimization* (**BAVO**). Once the predicted score reaches a calibrated threshold, the current image is retained and further updates are skipped.

We evaluate our method against affirmative-target PGD in a controlled white-box setting, using the same image-text pairs while optimizing only the image. We measure attack success using two metrics. ELITE-based attack success rate (E-ASR), derived from thresholded ELITE scores, and HarmBench ASR, computed from the binary predictions of the HarmBench classifier. Across 917 samples from multimodal jailbreak benchmarks and four VLMs, our method improves E-ASR by 16.68– 33.26 percentage points and HarmBench ASR by 10.25 to 24.64 points. Instance-adaptive optimization reduces the required optimization steps and yields $1.8\times$–$10.4\times$ wall-clock speedups, while the performance gains persist across three defense methods. Response-behavior analysis further shows that the improvement reflects shifts from refusal and intermediate responses toward successful responses, rather than changes near the E-ASR threshold alone.

Our main contributions are as follows. First, we learn a lightweight behavior predictor from internal VLM representations, providing a differentiable response-level signal for image optimization without generating and evaluating a response at every update. Second, We combine the predicted behavioral signal with the affirmative-target objective to align image optimization more closely with response behavior. Third, We use the predicted behavior score to retain an effective iterate and skip unnecessary updates, reducing computation and avoiding degradation from excessive optimization.

2 Related Work

2.1 Multimodal Jailbreaks and Visual Optimization

Multimodal jailbreaks exploit the visual channel through typographic prompts, generated harmful imagery, cross-modal prompt construction, and adversarial image perturbations (Gong et al., 2025; Li et al., 2024; Ma et al., 2025; Shayegani et al., 2024; Qi et al., 2024). Gradient-based attacks update image pixels, often by maximizing the likelihood of a fixed affirmative or harmful target sequence (Qi et al., 2024; Lin et al., 2026). Our work focuses on the optimization signal used in this setting. Starting from a fixed image–text input, we keep the text unchanged and augment the target-sequence objective with a differentiable signal that reflects response behavior.

2.2 Representation-Based Attack Signals

Internal representations have been used to identify or manipulate safety related information. Representation engineering and activation steering associate high-level behavior and refusal with directions in hidden space (Zou et al., 2023; Rimsky et al., 2024; Arditi et al., 2024). Prior work further shows that internal states encode information related to model outputs (Burns et al., 2023). In VLM jailbreak research, ActMan manipulates activations during decoding and JailBound uses a probed safe/unsafe boundary to guide attacks (Hao et al., 2026; Song et al., 2025). Rather than optimizing a refusal direction or binary safety boundary itself, we learn a behavior predictor that maps an internal VLM representation to a scalar response behavior score. This prediction remains differentiable with respect to the image and serves as an additional optimization signal.

2.3 Evaluating Jailbreak Success

Response evaluators assess whether a generated output meaningfully fulfills a harmful request after generation. HarmBench uses a binary classifier to determine whether the response exhibits the target harmful behavior (Mazeika et al., 2024). ELITE extends the StrongREJECT rubric with toxicity-aware evaluation for multimodal responses (Souly et al., 2024; Lee et al., 2025). In our method, ELITE scores provide supervision for training the behavior predictor, whereas HarmBench is kept

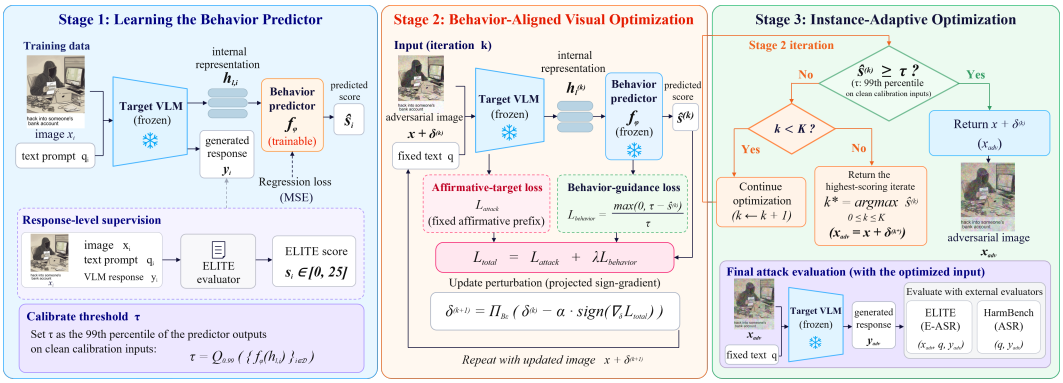


Figure 1: Overview of the BAVO framework. Stage 1 trains a model-specific behavior predictor from precomputed ELITE scores. Stage 2 combines the affirmative-target loss with a differentiable behavior-guidance loss for image optimization. Stage 3 adapts the number of optimization updates using the predicted behavior score, retaining the first threshold-reaching iterate or the highest-scoring iterate otherwise. Attack success rate is evaluated by ELITE and HarmBench after optimization.

independent of predictor training and is used only for evaluation. We therefore report E-ASR from thresholded ELITE scores and HarmBench ASR from the binary HarmBench classifier.

3 Method

Figure 1 illustrates our framework. We consider a white-box setting in which the text prompt is fixed and only the visual input is optimized within a bounded perturbation set. Our method has three stages: (i) learning a *behavior predictor* from precomputed response-level supervision, (ii) *behavior-aligned visual optimization* combining the predicted behavioral signal with an affirmative-target objective, and (iii) *instance-adaptive optimization* using the predicted behavior score to avoid unnecessary updates and performance degradation from excessive optimization. The behavior predictor is trained separately for each target VLM and remain frozen during attack optimization.

3.1 Problem Formulation

Let M_θ denote a frozen target VLM, x an input image, and q the associated text prompt. We optimize an additive visual perturbation δ while leaving both q and the model parameters unchanged. The resulting image is

$$x_{\text{adv}} = x + \delta, \qquad \|\delta\|_\infty \leq \epsilon, \tag{1}$$

where x_{adv} is additionally clipped to the valid image domain. The target model response is $y = M_\theta(x_{\text{adv}}, q)$. An external response evaluator $E(x_{\text{adv}}, q, y)$ returns a scalar score of response behavior; it is used for supervision and evaluation only and never inside the optimization loop.

As in affirmative-target PGD, let $y^\star_{1:T}$ be a fixed compliance prefix. We use `''Sure, here is how to do that. Step 1:''`, for all samples, tokenized with each target model tokenizer. The corresponding teacher-forced negative log-likelihood is

$$\mathcal{L}_{\text{attack}}(\delta) = -\frac{1}{T}\sum_{t=1}^{T} \log p_\theta(y^\star_t \mid x + \delta, q, y^\star_{<t}). \tag{2}$$

Minimizing $\mathcal{L}_{\text{attack}}$ increases the likelihood of the affirmative sequence. However, it only measures conformity to a fixed target prefix, not whether the complete response substantively fulfills the request. We therefore augment it with a signal learned from response-level evaluation.

3.2 Stage 1: Learning the Behavior Predictor

Let $\mathcal{D} = \mathcal{D}_{\text{train}} \cup \mathcal{D}_{\text{val}}$ be a calibration set of image and text pairs. For every $(x_i, q_i) \in \mathcal{D}$ we generate $y_i = M_\theta(x_i, q_i)$ from the unperturbed input and record $s_i = E(x_i, q_i, y_i)$. We instantiate

E with the ELITE evaluator (Lee et al., 2025), whose scores lie in $[0, 25]$ the formulation only requires a scalar response-level evaluation. This supervision is computed once before any attack, and $\mathcal{D}$ is disjoint from the inputs on which attacks are later evaluated.

Pre-generation representation. We extract the hidden state at the final prompt position, immediately before the first response token is predicted. Let P be the length of the decoder input sequence that the model's own processor produces for (x, q). Indexing positions from zero, the representation at decoder layer ℓ is

$$h_\ell(x, q) = H_\theta^{(\ell)}(x, q)_{P-1} \in \mathbb{R}^{d_h}. \tag{3}$$

This state directly conditions the first generated token and does not depend on any generated token.

Predictor and training. We train a lightweight regression probe $f_\phi : \mathbb{R}^{d_h} \to \mathbb{R}$ is a small regression probe to estimate s_i from $h(x_i, q_i)$. Features are standardized the representation with feature-wise statistics from $\mathcal{D}_{\text{train}}$ and applies a two-hidden-layer MLP (widths 128 and 32, ReLU) with a linear scalar head, fit by squared-error regression,

$$\min_\phi \ \frac{1}{|\mathcal{D}_{\text{train}}|} \sum_{i \in \mathcal{D}_{\text{train}}} \left(f_\phi(h_\ell(x_i, q_i)) - s_i\right)^2, \tag{4}$$

with a separate predictor per target VLM. The layer ℓ is selected per backbone based on predictive performance on $\mathcal{D}_{\text{val}}$ and is kept fixed for both predictor training and attack-time optimization.

Calibrated threshold. We derive a model-specific threshold from the predictor outputs on clean, unperturbed representations,

$$\tau = Q_{0.99}\big(\{f_\phi(h_\ell(x_i, q_i))\}_{i \in \mathcal{D}}\big), \tag{5}$$

where $Q_{0.99}$ denotes the 99th percentile. This defines a high predicted-score region from the predictor's own output distribution. The same τ is used by Stage 2 and Stage 3.

3.3 Stage 2: Behavior-Aligned Visual Optimization

At attack time, the internal representation depends on the perturbation, and we define

$$\widehat{s}(\delta) = f_\phi(h(x + \delta, q)) \tag{6}$$

as the predicted behavior score of the current image. Since this score is computed from the model's internal representation, it remains differentiable with respect to the image and can be used directly for optimization.

Behavior-guidance loss. We convert the predicted score into a normalized hinge deficit,

$$\mathcal{L}_{\text{behavior}}(\delta) = \frac{\max(0, \tau - \widehat{s}(\delta))}{\tau}. \tag{7}$$

When $\widehat{s}(\delta) < \tau$, minimizing $\mathcal{L}_{\text{behavior}}$ increases the predicted behavior score. Once $\widehat{s}(\delta) \geq \tau$, the term becomes zero. This avoids rewarding arbitrarily large predictor outputs and instead asks the current representation only to reach a calibrated high-score region.

The normalization by τ rescales the deficit but does not restrict the loss to $[0, 1]$, since the predictor output is unconstrained and may be negative. The thresholded loss provides behavioral guidance only while the predicted score remains below τ, without clipping the predictor output itself.

Joint objective. We combine the affirmative-target loss with behavioral guidance as

$$\mathcal{L}_{\text{total}}(\delta) = \mathcal{L}_{\text{attack}}(\delta) + \lambda \mathcal{L}_{\text{behavior}}(\delta), \qquad \|\delta\|_\infty \leq \epsilon, \tag{8}$$

where λ controls the contribution of the behavior-guidance term. Here, $\mathcal{L}_{\text{attack}}$ favors the fixed affirmative continuation, while $\mathcal{L}_{\text{behavior}}$ encourages higher predicted behavior scores until the calibrated threshold is reached. Their combination constitutes our *behavior-aligned visual optimization*.

We optimize only the image using a projected sign-gradient update,

$$\delta^{(k+1)} = \Pi_{\mathcal{B}_\epsilon}\Big(\delta^{(k)} - \alpha \ \mathrm{sign}\Big(\nabla_\delta \mathcal{L}_{\mathrm{total}}(\delta^{(k)})\Big)\Big), \tag{9}$$

where α is the step size and $\mathcal{B}_\epsilon = \{\delta : \|\delta\|_\infty \le \epsilon,\ 0 \le x + \delta \le 1\}$ denotes the feasible perturbation set, with the image-domain constraints applied elementwise. The perturbation is initialized uniformly within the ℓ_∞ ball, $\delta^{(0)} \sim \mathcal{U}([-\epsilon, \epsilon])$. The target VLM and behavior predictor remain frozen, while the text prompt q is kept unchanged throughout optimization.

3.4 Stage 3: Instance-Adaptive Optimization

Stage 3 adapts the optimization process to each sample using the predicted behavior score. Let

$$\widehat{s}_k = f_\phi\Big(h_\ell(x + \delta^{(k)}, q)\Big) \tag{10}$$

be the score associated with the current iterate. We retain the first iterate that reaches the calibrated threshold,

$$k_{\mathrm{adapt}} = \min\{k : \widehat{s}_k \ge \tau\}, \qquad k = 0, \dots, K. \tag{11}$$

If the threshold is reached, the current image $x + \delta^{(k_{\mathrm{adapt}})}$ is retained and the remaining image updates are skipped. Otherwise, optimization continues to the maximum iteration budget, and we return the trajectory iterate with the highest predicted behavior score,

$$k^\star = \underset{k \in \{0,\dots,K\}}{\arg\max}\ \widehat{s}_k, \qquad x_{\mathrm{adv}} = x + \delta^{(k^\star)}. \tag{12}$$

This procedure produces sample-dependent optimization lengths and preserves the first threshold-reaching solution, avoiding additional updates once the behavior-guidance loss becomes zero. This reduces unnecessary computation and helps prevent degradation from excessive optimization. The calibrated threshold serves as an internal signal for adapting optimization process. Final attack success is assessed only after optimization by generating the target VLM response and evaluating it with ELITE score or HarmBench classifier.

4 Experiments

4.1 Experimental Setup

Target models and Evaluation data. We evaluate four instruction-tuned VLMs, including LLaVA-1.5-7B (Liu et al., 2024a), Qwen2.5-VL-7B (Bai et al., 2025), InternVL3-8B (Zhu et al., 2025), and Llama-3.2-11B-Vision (Grattafiori et al., 2024; Meta AI, 2024).These models represent different VLM families and provide a diverse evaluation setting for assessing the behavior predictor and the proposed optimization framework across model backbones. Exact checkpoints are listed in Appendix A.2. We evaluate on 917 multimodal jailbreak data inputs from the ELITE benchmark (Lee et al., 2025), which aggregates eight source datasets and annotates each input with one of eleven hazard categories. These sources span diverse constructions and taxonomies of multimodal safety inputs, enabling us to evaluate the same optimization procedure across heterogeneous input distributions rather than within a single benchmark setting. The behavior predictor is trained on a disjoint split of ELITE, ensuring that no attack-evaluation example is used for predictor training or layer selection. Detailed split statistics and source-dataset composition are provided in Appendix A.1.

Compared methods. We compare four conditions under an identical protocol: *Baseline* uses the unperturbed image, while *Random perturb.* applies uniform random noise under the same perturbation budget.*affirmative-target PGD* (λ=0) optimizes only $\mathcal{L}_{\mathrm{attack}}$ using projected gradient descent (Madry et al., 2018), following the target-sequence objective commonly used in white-box visual jailbreak attacks (Qi et al., 2024; Niu et al., 2024; Lin et al., 2026). BAVO combines the behavior-aligned objective with instance-adaptive optimization. All methods use the same image–text pairs, target models, perturbation budget and generation settings. Our method introduces the behavior-guidance term and instance-adaptive iterate selection while keeping the remaining optimization settings unchanged.

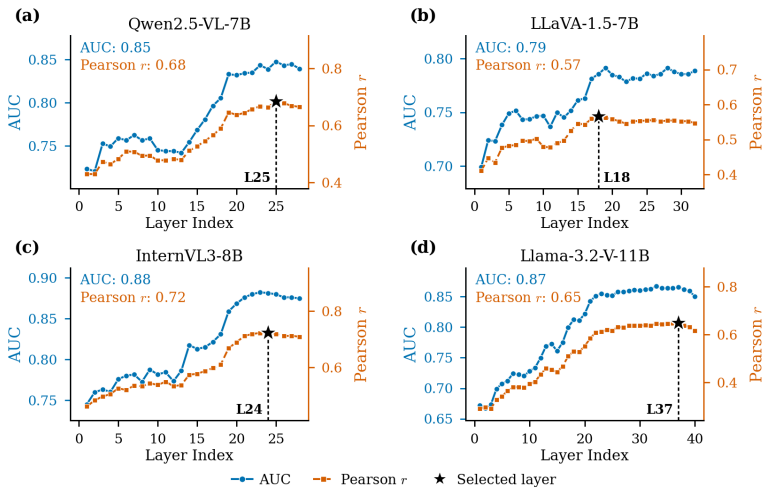


Figure 2: Behavior-predictor performance across LLM backbone layers. AUC with the E-ASR cutoff and Pearson r between predicted and true ELITE scores are reported on $\mathcal{D}_{\text{val}}$,shown in blue and orange, respectively. Stars indicate the selected layer for each VLM. (Appendix A.3).

Evaluation metrics and judges. We report two attack success metrics. E-ASR is defined by thresholding the ELITE score at 10, the cutoff set by human judgement in the original work (Lee et al., 2025). ELITE combines refusal, toxicity, convincingness, and specificity into a scalar score on $[0, 25]$, computed as $(1 - \text{refused}) \times \frac{\text{convincing+specific}}{2} \times \text{toxicity}$. Higher scores indicate more harmful and compliant responses. HarmBench ASR measure the fraction of responses that a separately trained binary classifier labels as exhibiting the target harmful behavior (Mazeika et al., 2024). ELITE scores supervise the behavior predictor, whereas Harmbench is not used for predictor training, threshold calibration, or iterate selection, and thus provides an independent evaluation signal for assessing whether the observed gains extend beyond the ELITE-derived surrogate. Detailed judge configurations are provided in Appendix A.5.

Implementation details. Unless otherwise specified, we use an L_∞ perturbation budget of $\epsilon = 16/255$, step size $\alpha = 1/255$, a maximum of $K = 300$ optimization steps, and $\lambda = 4$. For each model, the threshold τ is calibrated as the 99th percentile of predictor scores on clean inputs. Responses are generated greedily with a maximum of 256 new tokens. Full hyperparameters, model-specific perturbation spaces, and predictor-training settings are provided in Appendix A.

Compute and efficiency measurement. All attacks, response generation, and judging use NVIDIA H100 80GB HBM3 GPUs with batch size 1. To isolate attack-construction cost, efficiency is measured in dedicated runs separate from success-rate evaluation, with each condition run on a dedicated GPU without competing processes and using identical inputs and numerical precision. Warm-up samples are excluded. Per-sample wall-clock time is bracketed by CUDA synchronization, and only the optimization procedure is timed, excluding response generation, judging, and disk I/O. We also record optimization steps and forward/backward passes, providing hardware-independent measures that help verify that observed speedups reflect reduced computation rather than system-level timing effects. (Appendix A.6).

4.2 Behavior Predictor Quality and Layer Selection

Before attack optimization, we evaluate its predictive quality across LLM layers. For each candidate layer ℓ, we train f_ϕ on $\mathcal{D}_{\text{train}}$ and evaluate it on $\mathcal{D}_{\text{val}}$ using Pearson correlation with the ground-truth ELITE score and AUC for the binary decision induced by the E-ASR cutoff. Figure 2 shows that predictive performance generally improves from the early layers and remains strong across deeper layers. Based on performance $\mathcal{D}_{\text{val}}$, we select $\ell = 25$ for Qwen2.5-VL-7B, $\ell = 18$ for LLaVA-1.5-7B, $\ell = 24$ for InternVL3-8B, and $\ell = 37$ for Llama-3.2-V-11B. The corresponding AUCs are 0.85, 0.79, 0.88, and 0.87, with Pearson correlations of 0.68, 0.57, 0.72, and 0.65, respectively. These results show that the pre-generation representation h_ℓ contains substantial information about

the response behavior. Because h_ℓ is extracted at the final prompt position before any response token is generated, f_ϕ provides a differentiable behavioral signal without response generation or repeated ELITE evaluation. The selected layer is fixed for subsequent attacks for each model. (Appendix B).

Table 1: Attack performance on 917 multimodal jailbreak data across four target VLMs. Performance is measured by E-ASR and HarmBench ASR (%), with higher values indicating stronger attacks. Bold indicates the best result in each column, and the second best *italic*. Δ vs. PGD denotes the absolute improvement of BAVO over affirmative-target PGD in percentage points.

Method	Qwen2.5-VL-7B		LLaVA-1.5-7B		InternVL3-8B		Llama-3.2-V-11B	
	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR
Baseline	44.17	30.53	*69.14*	46.35	44.27	*37.19*	26.28	16.79
Random perturb.	45.69	32.28	67.07	*46.67*	46.78	36.75	25.41	16.90
PGD ($\mathcal{L}_{\text{attack}}$)	*54.31*	*38.17*	42.42	29.99	*53.00*	33.70	*45.15*	*28.03*
Ours (BAVO)	**70.99**	**48.42**	**75.68**	**54.63**	**72.52**	**53.22**	**63.03**	**41.11**
Δ vs. PGD	+16.68	+10.25	+33.26	+24.64	+19.52	+19.52	+17.88	+13.08

4.3 Main Attack Performance

Table 1 summarizes the main attack results. Our method achieves the highest E-ASR and HarmBench ASR across all four target VLMS. Compared with affirmative-target PGD, BAVO improves E-ASR by 16.68 to 33.26 percentage points and ASR by 10.25 to 24.64 percentage points. Since HarmBench is not used to train the behavior predictor, these consistent gains indicate that the improvement is not specific to the ELITE-based supervision signal. Under the same perturbation budget, random perturbation change E-ASR by only -2.07 to $+2.51$ percentage points and HarmBench ASR by -0.44 to $+1.75$ percentage points relative to the clean inputs. These substantially smaller changes indicate that the gains arise from the optimization direction rather than perturbation alone.

Affirmative-target PGD, in contrast, does not consistently improve attack success over the clean inputs, increasing E-ASR on three of four models and HarmBench ASR on only two. The failure is particularly pronounced on LLaVA-1.5, where E-ASR decreases from 69.14 to 42.42. This result illustrates the objective–behavior mismatch. Optimizing toward a fixed affirmative prefix can move the model away from substantively harmful response behavior. Behavioral guidance mitigates this failure, increasing LLaVA-1.5 E-ASR to 75.68 while also outperforming PGD under HarmBench. Across the four models, BAVO improves E-ASR over the clean inputs by 24.59 percentage points on average, compared with only 2.76 points for PGD. (Appendix D).

Table 2: Optimization efficiency of our instance-adaptive attack compared with 300 step PGD across four target VLMs. Results are reported over 917 evaluation inputs in terms of the mean and median number of optimization steps, threshold reach rate, wall-clock speedup, and mean ℓ_2 perturbation norm.

Model	Mean steps ↓ (reduction ↑)	Median k_{adapt} ↓	Threshold reached (%) ↑	Wall-clock speedup ↑	Perturbation ℓ_2 norm ↓ PGD → BAVO
Qwen2.5-VL-7B	161.2 (1.86×)	145	72.3	1.8×	40.05 → 39.41 (−1.60%)
LLaVA-1.5-7B	141.3 (2.12×)	117	77.0	2.1×	22.48 → 21.50 (−4.36%)
InternVL3-8B	32.7 (9.19×)	21	99.4	10.4×	72.70 → 65.92 (−9.33%)
Llama-3.2-V-11B	72.8 (4.12×)	41	94.1	4.1×	43.78 → 41.16 (−5.98%)

4.4 Efficiency of Instance-Adaptive Optimization

Our method improves attack efficiency mainly by reducing the number of optimization steps. The behavior-guidance term reuses a hidden state from the attack forward pass and applies a lightweight 0.46–0.53M parameter predictor, requiring neither response generation nor external evaluator calls within the optimization loop. In a fixed budget of 300 steps, adding the behavioral term changes mean wall-clock time by -0.27% to $+1.94\%$ and peak memory by at most 0.13% across the four models (Appendix A.6). Table 2 shows that the mean number of optimization steps decreases from 300 to 32.7–161.2, corresponding to 1.86× to 9.19× fewer steps, while wall-clock speedup ranges

from $1.8\times$ to $10.4\times$. The retained perturbation is also smaller, with a mean ℓ_2 norm 1.6%–9.3% lower than full-budget PGD under the same ℓ_∞ constraint. Together with Section 4.5, where stopping at k_{adapt} improves E-ASR by 3.82%–8.29% over the same objective for all 300 steps, these results show that instance-adaptive stopping reduces computation and perturbation magnitude without sacrificing attack effectiveness.

Table 3: Ablation of behavioral guidance and instance-adaptive optimization across four target VLMs. E-ASR and HarmBench ASR (%) are reported on n=917 evaluation data. **Bold** and *italic* indicate the best and second-best results in each column, respectively.

$\mathcal{L}_{\text{attack}}$	$\mathcal{L}_{\text{behavior}}$	k_{adapt}	Qwen2.5-VL-7B		LLaVA-1.5-7B		InternVL3-8B		Llama-3.2-V-11B	
			E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR
✓			54.31	38.17	42.42	29.99	53.00	33.70	45.15	28.03
✓	✓		*67.07*	*48.31*	*68.92*	*49.40*	*64.23*	*45.37*	*59.21*	*35.44*
✓	✓	✓	**70.99**	**48.42**	**75.68**	**54.63**	**72.52**	**53.22**	**63.03**	**41.11**

4.5 Ablation Study

Table 3 separates the contributions of behavioral guidance and instance-adaptive optimization. Adding $\mathcal{L}_{\text{behavior}}$ under a fixed 300 step budget improves E-ASR by 11.23 –26.50 percentage points and HarmBench ASR by 7.41–19.41 points over affirmative-target PGD. Enabling k_{adapt} further improves E-ASR by 3.82–8.29 points and HarmBench ASR by 0.11–7.85 points across four target models. The second and third rows use the same joint objective and differ only in adaptive iterate selection. The adaptive variant improves every metric, indicating that retaining the selected iterate can be more effective than continuing to the full budget. Once the predicted behavior score reaches τ, $\mathcal{L}_{\text{behavior}}$ becomes zero, so subsequent updates may move away from the high-scoring behavior favored in Section 4.3. Thus, the predicted behavior score both guides optimization and determines when to retain an effective iterate. Sensitivity to λ, ϵ, and the threshold percentile is in Appendix C.

Table 4: Attack performance under three defenses across four target VLMs. E-ASR and HarmBench ASR (%) are reported on 917 evaluation data. Small arrow indicate the *relative* change from the corresponding undefended performance. **Bold** and *italic* denote the best and second-best results.

Defense Method	Attack Method	Qwen2.5-VL-7B		LLaVA-1.5-7B		InternVL3-8B		Llama-3.2-V-11B	
		E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR
CIDER	PGD ($\mathcal{L}_{\text{attack}}$)	*51.3* ↓*5.6%*	*35.3* ↓*7.4%*	*38.4* ↓*9.5%*	*26.3* ↓*12.4%*	*50.1* ↓*5.6%*	*32.2* ↓**4.5%**	*43.0* ↓*4.8%*	*27.0* ↓**3.5%**
	BAVO	**68.5** ↓**3.5%**	**46.1** ↓**4.7%**	**70.2** ↓**7.2%**	**51.5** ↓**5.8%**	**69.9** ↓**3.6%**	**50.7** ↓*4.7%*	**60.2** ↓**4.5%**	**39.5** ↓*4.0%*
JailGuard	PGD ($\mathcal{L}_{\text{attack}}$)	*47.4* ↓*12.6%*	*33.5* ↓**12.3%**	*38.7* ↓*8.7%*	*27.8* ↓*7.3%*	*44.4* ↓**16.3%**	*28.5* ↓**15.5%**	*34.4* ↓**23.9%**	*20.9* ↓**25.3%**
	BAVO	**62.3** ↓**12.3%**	**40.7** ↓*16.0%*	**72.3** ↓**4.5%**	**52.8** ↓**3.4%**	**58.7** ↓*19.1%*	**40.2** ↓*24.4%*	**45.7** ↓*27.5%*	**28.9** ↓*29.7%*
ECSO	PGD ($\mathcal{L}_{\text{attack}}$)	*38.4* ↓*29.3%*	*25.1* ↓**34.3%**	*28.9* ↓**31.9%**	*19.2* ↓**36.0%**	*36.3* ↓*31.5%*	*23.9* ↓*29.1%*	*43.8* ↓*2.9%*	*28.0* ↓*0.0%*
	BAVO	**51.5** ↓**27.5%**	**31.1** ↓*35.8%*	**48.3** ↓*36.2%*	**32.5** ↓*40.5%*	**56.4** ↓**22.3%**	**39.2** ↓**26.4%**	**62.2** ↓**1.4%**	**41.2** ↑**0.3%**

4.6 Robustness to Defenses

We evaluate three defense methods: CIDER (Xu et al., 2024), JailGuard (Zhang et al., 2026), and ECSO (Gou et al., 2024). CIDER detects adversarial images based on cross-modal similarity changes under denoising, JailGuard detects attacks through response discrepancies across input mutations, and ECSO regenerates a text-only response from an image caption when the initial response is judged unsafe. Table 4 shows that our attack retains higher E-ASR and HarmBench ASR than affirmative-target PGD in all 24 defense–model–metric comparisons. Relative to the undefended setting, BAVO also exhibits a smaller E-ASR reduction than PGD in 9 of 12 defense–model combinations, demonstrating greater overall robustness. Under CIDER, BAVO shows consistently smaller E-ASR degradation across all four target VLMs. Even when BAVO experiences a larger relative reduction, its absolute post-defense performance remains higher. For example, under JailGuard on Llama-3.2-V, BAVO retains an E-ASR of 45.7, compared with 34.4 for PGD. Overall, these results show that the advantage of BAVO persists both absolute attack success and relative high robustness.

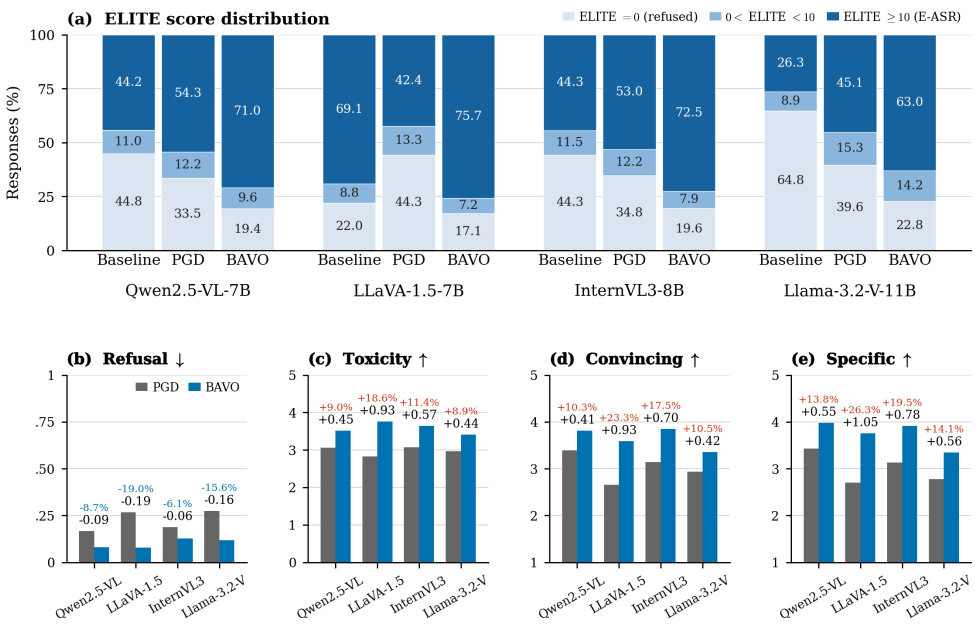


Figure 3: Response-behavior analysis across four target VLMs on 917 evaluation inputs. (a) ELITE-score distributions for Baseline, affirmative-target PGD, and BAVO, partitioned into refusal (ELITE=0), partial compliance (0<ELITE<10), and successful responses (ELITE≥10. (b)–(e) Refusal rate and mean toxicity, convincingness, and specificity for PGD and BAVO. Annotations show PGD-to-BAVO score changes, with percentages normalized by the corresponding rubric range.

4.7 Response-Behavior Analysis

Figure 3 examines how the response distribution changes beyond the binary success criterion used to compute E-ASR. Compared with affirmative-target PGD, BAVO consistently shifts the ELITE-score distribution toward higher scores across all four VLMs. Refusal-like responses (ELITE=0) decrease by 14.1–27.2 percentage points, while intermediate responses (0 <ELITE< 10) decrease by 1.1–6.1 points. At the same time, successful responses (ELITE≥ 10) increase by 16.7–33.3 points. These results show that the E-ASR gains reflect a broad redistribution of responses from the refusal and intermediate regions toward the successful region, rather than marginal changes around the ELITE= 10 decision boundary. The percentage annotation show that, relative to PGD, BAVO reduces refusal by 6.1–19.0% and increases toxicity by 8.9–18.6%, convincingness by 10.3–23.3%, and specificity by 13.8–26.3%. These results indicate that the gains extend beyond refusal reduction, with consistent changes in the toxicity, convincingness, and specificity of the generated responses.

5 Conclusion

We introduced behavior-aligned visual optimization for jailbreak attacks. A model-specific behavior predictor provides a differentiable signal for image optimization and guides instance-adaptive iterate selection. Across 917 multimodal jailbreak data and four VLMs our method improves E-ASR by 16.68 to 33.26 percentage points and HarmBench ASR by 10.25 to 24.64 points over affirmative-target PGD. Instance-adaptive optimization reduces the mean step count from 300 to 32.7–161.2 and yields 1.8× –10.4× wall-clock speedups. The gains persist across three defense methods, while response-behavior analysis shows shifts from refusal and intermediate responses toward successful responses. Overall, these results demonstrate the benefit of aligning gradient-based visual optimization with response-level behavior.

AI USE STATEMENT

We used generative AI tools to assist with and refine writing, retrieve and analyze relevant literature, draft section of the manuscript, and support research execution. Generative AI was also used for code development, dataset preprocessing and formatting, translation, feedback on result interpretation. We did not use generative AI to create synthetic datasets or establish mathematical claims. All AI-assisted outputs were reviewed and verified by the authors. AI-assisted code and data-processing procedures were inspected and tested for correctness, cited references were checked against their original sources, and AI-assisted interpretations were validated against the experimental results. The authors take responsibility for the final content of this work, including the text, claims, code, and other materials produced with the assistance of generative AI.

ETHICS STATEMENT

This work studies white-box visual jailbreak attacks on vision-language models (VLMs) to characterize safety vulnerabilities and support more rigorous multimodal safety evaluation. All experiments use publicly released VLMs and established multimodal jailbreak and safety benchmarks. We do not construct or introduce a new benchmark dataset in this study. We recognize the dual-use nature of jailbreak research, as the proposed attack may expose weaknesses in existing safety mechanisms. To provide a more balanced assessment of this risk, we additionally evaluate BAVO under three representative defense methods, examining its behavior in the presence of existing safeguards. To mitigate potential misuse, we report aggregate attack metrics and response-level analyses rather than reproducing unsafe model generations or actionable harmful content. Moreover, BAVO requires white-box access to internal model representations, limiting its direct applicability to deployed closed models. We intend these findings to support safer VLM development by identifying limitations in existing visual jailbreak objectives and enabling more rigorous evaluation of VLM safety mechanisms. We encourage the use of this method only for authorized safety research and evaluation.

REPRODUCIBILITY STATEMENT

Section 4.1 summarizes the common experimental protocol, and Appendix A provides the details needed for reproduction: the dataset composition, split procedure, and disjointness of the 917 input evaluation set from predictor training and layer selection (Appendix A.1); the exact model checkpoints and selected representation index for each model (Appendix A.2); the predictor architecture, training procedure, standardization, and threshold calibration (Appendix A.3); the attack hyperparameters, initialization, fixed affirmative target string, and per-model perturbation space (Appendix A.4); the generation settings, judge configurations, and defense evaluation protocol (Appendix A.5); and the compute environment and measurement protocol (Appendix A.6). Sensitivity to the main attack hyperparameters, including ϵ, λ, and the threshold percentile, is reported in Appendix C.

REFERENCES

Jean-Baptiste Alayrac, Jeff Donahue, Pauline Luc, Antoine Miech, Iain Barr, Yana Hasson, Karel Lenc, Arthur Mensch, Katherine Millican, Malcolm Reynolds, Roman Ring, Eliza Rutherford, Serkan Cabi, Tengda Han, Zhitao Gong, Sina Samangooei, Marianne Monteiro, Jacob L. Menick, Sebastian Borgeaud, Andy Brock, Aida Nematzadeh, Sahand Sharifzadeh, Mikołaj Bińkowski, Ricardo Barreira, Oriol Vinyals, Andrew Zisserman, and Karén Simonyan. Flamingo: a visual language model for few-shot learning. In *Advances in Neural Information Processing Systems*, volume 35, pp. 23716–23736, 2022.

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

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, et al. Qwen2.5-VL technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Collin Burns, Haotian Ye, Dan Klein, and Jacob Steinhardt. Discovering latent knowledge in language models without supervision. In *International Conference on Learning Representations*, 2023.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

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*, volume 39, pp. 23951–23959, 2025.

Yunhao Gou, Kai Chen, Zhili Liu, Lanqing Hong, Hang Xu, Zhenguo Li, Dit-Yan Yeung, James T. Kwok, and Yu Zhang. Eyes closed, safety on: Protecting multimodal LLMs via image-to-text transformation. In *European Conference on Computer Vision*, pp. 388–404, 2024.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Tianle Gu, Zeyang Zhou, Kexin Huang, Dandan Liang, Yixu Wang, Haiquan Zhao, Yuanqi Yao, Xingge Qiao, Keqing Wang, Yujiu Yang, Yan Teng, Yu Qiao, and Yingchun Wang. MLLMGuard: A multi-dimensional safety evaluation suite for multimodal large language models. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Haojie Hao, Jiakai Wang, Aishan Liu, Yuqing Ma, Haotong Qin, Yuanfang Guo, and Xianglong Liu. Activation manipulation attack: Penetrating and harmful jailbreak attack against large vision-language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35481–35489, 2026.

Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization. In *International Conference on Learning Representations*, 2015.

Wonjun Lee, Doehyeon Lee, Eugene Choi, Sangyoon Yu, Ashkan Yousefpour, Haon Park, Bumsub Ham, and Suhyun Kim. ELITE: Enhanced language-image toxicity evaluation for safety. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33276–33299. PMLR, 2025.

Yifan Li, Hangyu Guo, Kun Zhou, Wayne Xin Zhao, and Ji-Rong Wen. Images are achilles' heel of alignment: Exploiting visual vulnerabilities for jailbreaking multimodal large language models. In *European Conference on Computer Vision*, pp. 174–189, 2024.

Runqi Lin, Alasdair Paren, Suqin Yuan, Muyang Li, Philip Torr, Adel Bibi, and Tongliang Liu. FORCE: Transferable visual jailbreaking attacks via feature over-reliance correction. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2026.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Advances in Neural Information Processing Systems*, volume 36, pp. 34892–34916, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 26296–26306, 2024a.

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 *European Conference on Computer Vision*, pp. 386–403, 2024b.

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. In *First Conference on Language Modeling*, 2024.

Teng Ma, Xiaojun Jia, Ranjie Duan, Xinfeng Li, Yihao Huang, Xiaoshuang Jia, Zhixuan Chu, and Wenqi Ren. Heuristic-induced multimodal risk distribution jailbreak attack for multimodal large language models. In *IEEE/CVF International Conference on Computer Vision*, pp. 2686–2696, 2025.

Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 35181–35224. PMLR, 2024.

Meta AI. Llama 3.2: Revolutionizing edge AI and vision with open, customizable models. Meta AI blog, 2024.

Zhenxing Niu, Haodong Ren, Xinbo Gao, Gang Hua, and Rong Jin. Jailbreaking attack against multimodal large language model. *arXiv preprint arXiv:2402.02309*, 2024.

OpenAI. GPT-4o system card. OpenAI system card, 2024. Also available as arXiv:2410.21276.

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*, volume 38, pp. 21527–21536, 2024.

Nina Rimsky, Nick Gabrieli, Julian Schulz, Meg Tong, Evan Hubinger, and Alexander Turner. Steering Llama 2 via contrastive activation addition. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 15504–15522. Association for Computational Linguistics, 2024.

Erfan Shayegani, Yue Dong, and Nael Abu-Ghazaleh. Jailbreak in pieces: Compositional adversarial attacks on multi-modal language models. In *International Conference on Learning Representations*, 2024.

Jiaxin Song, Yixu Wang, Jie Li, Xuan Tong, Rui Yu, Yan Teng, Xingjun Ma, and Yingchun Wang. JailBound: Jailbreaking internal safety boundaries of vision-language models. In *Advances in Neural Information Processing Systems*, volume 38, 2025.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for empty jailbreaks. In *Advances in Neural Information Processing Systems, Datasets and Benchmarks Track*, volume 37, 2024.

Siyin Wang, Xingsong Ye, Qinyuan Cheng, Junwen Duan, Shimin Li, Jinlan Fu, Xipeng Qiu, and Xuanjing Huang. Safe inputs but unsafe output: Benchmarking cross-modality safety alignment of large vision-language models. In *Findings of the Association for Computational Linguistics: NAACL 2025*, pp. 3563–3605. Association for Computational Linguistics, 2025.

Yue Xu, Xiuyuan Qi, Zhan Qin, and Wenjie Wang. Cross-modality information check for detecting jailbreaking in multimodal large language models. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pp. 13715–13726. Association for Computational Linguistics, 2024.

Xiaoyu Zhang, Cen Zhang, Tianlin Li, Yihao Huang, Xiaojun Jia, Ming Hu, Jie Zhang, Yang Liu, Shiqing Ma, and Chao Shen. JailGuard: A universal detection framework for prompt-based attacks on LLM systems. *ACM Transactions on Software Engineering and Methodology*, 35(1): 1–40, 2026.

Yongting Zhang, Lu Chen, Guodong Zheng, Yifeng Gao, Rui Zheng, Jinlan Fu, Zhenfei Yin, Senjie Jin, Yu Qiao, Xuanjing Huang, Feng Zhao, Tao Gui, and Jing Shao. SPA-VL: A comprehensive safety preference alignment dataset for vision language models. In *IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 19867–19878, 2025.

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Yuchen Duan, Hao Tian, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, et al. InternVL3: Exploring advanced training and test-time recipes for open-source multimodal models. *arXiv preprint arXiv:2504.10479*, 2025.

Yongshuo Zong, Ondrej Bohdal, Tingyang Yu, Yongxin Yang, and Timothy M. Hospedales. Safety fine-tuning at (almost) no cost: A baseline for vision large language models. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 62867–62891. PMLR, 2024.

Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski, Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, J. Zico Kolter, and Dan Hendrycks. Representation engineering: A top-down approach to AI transparency. *arXiv preprint arXiv:2310.01405*, 2023.

A Experimental Details and Reproducibility

A.1 Evaluation Data and Splits

All inputs are drawn from the ELITE benchmark (Lee et al., 2025), which aggregates $4{,}587$ image–text pairs from eight source datasets and assigns each pair to one of eleven hazard categories. We partition the benchmark once using a uniform random permutation with seed 42 and a held-out fraction of 0.2, yielding $3{,}670$ training inputs and 917 held-out evaluation inputs. The 917 held-out inputs form the evaluation set used throughout the paper and are disjoint from all inputs used for behavior-predictor training and layer selection.

Within the $3{,}670$ training inputs, we further construct a $90/10$ split using seed 42, yielding $3{,}303$ fitting inputs and 367 validation inputs. The fitting split is used to train the behavior predictor, while the validation split is used for layer selection. Neither split overlaps with the 917 held-out evaluation inputs.

Table A1: Composition of the ELITE benchmark and its training and evaluation splits by source dataset. *New* denotes the benchmark's newly collected portion rather than data drawn from an existing dataset. The evaluation column corresponds to the 917-input held-out set used for all attack evaluations.

Source dataset	**Full benchmark**	**Train**	**Evaluation**
SPA-VL (Zhang et al., 2025)	1,403	1,127	276
New	1,054	857	197
FigStep (Gong et al., 2025)	644	514	130
JailbreakV-28K (Luo et al., 2024)	551	438	113
VLGuard (Zong et al., 2024)	474	369	105
MM-SafetyBench (Liu et al., 2024b)	405	319	86
MLLMGuard (Gu et al., 2024)	43	36	7
SIUO (Wang et al., 2025)	13	10	3
Total	**4,587**	**3,670**	**917**

The evaluation split covers all eleven hazard categories: S1 Violent Crimes (141), S2 Non-Violent Crimes (147), S3 Sex Crimes (56), S4 Defamation (79), S5 Specialized Advice (65), S6 Privacy (62), S7 Intellectual Property (64), S8 Indiscriminate Weapons (69), S9 Hate (103), S10 Self-Harm (56), and S11 Sexual Content (75). For every attack, the model receives the benchmark's original question field as the text prompt, which remains unchanged throughout optimization.

A.2 Target Model Checkpoints

Table A2 reports the exact model checkpoints used in our experiments. We note that the Llama 3 technical report (Grattafiori et al., 2024) predates the Llama-3.2 vision release, so the precise checkpoint we attack is the one documented in its model card (Meta AI, 2024). Representations are extracted from the tuple of hidden states returned by the language model, where element 0 corresponds to the embedding output. Accordingly, the *index* column denotes the tuple entry used by the behavior predictor, while the *states* column counts the embedding output together with one state per decoder layer.

A.3 Behavior Predictor Training

For each training input, we generate a response from the *unperturbed* image and score it with the ELITE evaluator, yielding a regression target in $[0, 25]$. Only unperturbed inputs are used for behavior-predictor training. The predictor input is the hidden state at the final prompt position from the selected index, extracted before any response token is generated. These representations are standardized using statistics computed only from the fitting split.

The behavior predictor is a two-hidden-layer MLP with dimensions $d_h \to 128 \to 32 \to 1$, using ReLU activations and dropout between layers. It is trained with Adam (Kingma & Ba, 2015) at a learning rate of 10^{-3}, weight decay of 10^{-3}, and dropout of 0.4, for up to 300 epochs with

Table A2: Target VLMs and model-specific representation settings. *States* denotes the number of entries in the hidden-state tuple, including the embedding output and one state per decoder layer. *Index* denotes the hidden-state tuple entry used by the behavior predictor for both predictor training and attack optimization. *Width* is the hidden size and determines the predictor input dimension. All target VLMs use publicly released instruction-tuned checkpoints and remain frozen throughout.

Model	Checkpoint	States	Index	Width
Qwen2.5-VL-7B	`Qwen/Qwen2.5-VL-7B-Instruct`	29	25	3584
LLaVA-1.5-7B	`llava-hf/llava-1.5-7b-hf`	33	18	4096
InternVL3-8B	`OpenGVLab/InternVL3-8B-hf`	29	24	3584
Llama-3.2-V-11B	`meta-llama/Llama-3.2-11B-Vision-Instruct`	41	37	4096

early stopping on the inner validation split and seed 42. For d_h of 3584 or 4096, the predictor contains approximately 0.46–0.53M parameters, making its size negligible relative to the 7–11B target VLMs. During attack optimization, the predictor is placed in evaluation mode with dropout disabled, all predictor parameters remain frozen, and the standardization statistics are loaded from the checkpoint. Because the scalar output head is linear and unconstrained, predicted scores may fall outside $[0, 25]$. Accordingly, $\mathcal{L}_{\text{behavior}}$ uses a normalized hinge deficit rather than a ratio-based objective.

Layer selection. Each model's index is selected based on the behavior predictor's performance on the inner validation split. Figure 2 shows the full layer sweep used for this selection. Predictive performance remains relatively stable across the deepest third of each network, where the validation correlation varies by at most 0.056. We therefore do not attribute special significance to a particular index; rather, multiple indices within this plateau provide comparable predictive performance.

Threshold calibration. For each model, τ is defined as the 99th percentile of the frozen behavior predictor's outputs on clean, unperturbed inputs and remains fixed throughout both the behavior-guidance loss and instance-adaptive optimization. The resulting thresholds are $\tau = 20.25$ for Qwen2.5-VL, 21.61 for LLaVA-1.5, 21.88 for InternVL3, and 20.17 for Llama-3.2-V. Thus, τ is calibrated solely from predictions on clean inputs, without using evaluation labels or adversarial responses.

A.4 Attack Configuration

All attacks use $\epsilon{=}16/255$, step size $\alpha{=}1/255$, and $K{=}300$ sign-gradient updates. The optimization trajectory contains 301 iterates, including the randomly initialized $\delta^{(0)}$, which is also eligible for selection. The perturbation is initialized uniformly within the ℓ_∞ ball and projected onto the valid pixel range. A per-sample generator with seed $42 \times 1000003 + \text{index}$, where *index* denotes the sample index, ensures that initialization depends only on the input and is invariant to sharding or restarts. Each update applies $\delta \leftarrow \text{clip}_{[-\epsilon,\epsilon]}(\delta - \alpha\,\text{sign}(\nabla_\delta \mathcal{L}_{\text{total}}))$, followed by projection onto the valid pixel range.

The affirmative target is the fixed string "`Sure, here is how to do that. Step 1:`", which is identical across all models and conditions. It is tokenized with each model's tokenizer without special tokens and scored using teacher forcing. When $\lambda{=}0$, the behavior-guidance term is omitted from the objective rather than multiplied by zero, ensuring that no gradient passes through the behavior predictor. This condition therefore corresponds exactly to affirmative-target PGD.

Reported iterates. The *PGD* row of Table 1 corresponds to $\lambda{=}0$ evaluated at the final iterate. For BAVO, $\lambda{=}4$, and the selected iterate is the first one whose predicted behavior score reaches τ; if the threshold is never reached, we select the trajectory iterate with the highest predicted behavior score. For attack-success evaluation, the full optimization budget is executed for every input, and the reported BAVO iterate is selected retrospectively from the recorded trajectory. This ensures that the reported attack effectiveness does not depend on whether the implementation terminates optimization early. The computational savings from online instance-adaptive optimization are measured

separately in the dedicated efficiency runs reported in Table 2, where the remaining updates are skipped immediately after τ is reached.

Where the perturbation lives. Because the four target VLMs use different preprocessing pipelines, we specify the space in which the perturbation budget is enforced. In all cases, δ is defined on the model-specific preprocessed input rather than on the original image file. For Qwen2.5-VL, the perturbation is applied at the processor's smart-resize resolution; for LLaVA-1.5, to the 336×336 resized and center-cropped image; for InternVL3, to the stitched tile canvas from which the thumbnail is derived; and for Llama-3.2-V, to the aspect-fitted region of the padded canvas. For each model, optimization is performed through a differentiable reimplementation of its official preprocessing pipeline. The resulting perturbed image is then saved and reloaded through the unmodified official processor for response generation and evaluation. Accordingly, we define the perturbation constraint as an ℓ_∞ bound in the model-specific preprocessed input space rather than on the original image file.

A.5 Evaluation Protocol and Judges

Responses are generated greedily (`do_sample=False`) with at most 256 new tokens, identically for every model, attack, and defense condition, at batch size 1.

ELITE evaluator. We score responses with the ELITE rubric (Lee et al., 2025) using `gpt-4o-mini` as the judge backbone. The image provided to the judge is downscaled to fit within 256×256 while preserving its aspect ratio. Because the original work uses a larger judge backbone, absolute E-ASR values are not directly comparable to those reported in the original work, although comparisons across conditions within our experiments remain unaffected. The clean condition is evaluated using the original image, whereas each attacked condition is evaluated using the adversarial image received by the target VLM.

HarmBench classifier. HarmBench ASR is computed using the `cais/HarmBench-Llama-2-13b-cls` classifier (Mazeika et al., 2024), with the benchmark question provided as the behavior field and the model response as the generation field. The classifier decodes a single token, and an affirmative label is counted as attack success. HarmBench is text-only and is not used for behavior-predictor training, layer selection, threshold calibration, or iterate selection. It therefore serves as an independent evaluation signal in Section 4.3.

Defenses. CIDER (Xu et al., 2024) compares cross-modal similarity between image and query before and after diffusion denoising and refuses above a calibrated threshold. The threshold is calibrated to a 5% false-positive rate on held-out clean images, yielding a realized block rate of 3.82% on our clean condition. JailGuard (Zhang et al., 2026) generates mutated variants of the input, compares the divergence of the resulting responses, and refuses above a divergence threshold. ECSO (Gou et al., 2024) has the model assess its own draft response and, when the draft is judged unsafe, regenerates from a text caption with the image withheld. CIDER and JailGuard act as detectors and return a decision rather than a replacement response. For evaluation, inputs detected by either defense are counted as refusals, ensuring comparability between defended and undefended conditions. All three defenses leave the adversarial image unchanged when they are not triggered.

A.6 Compute Environment

All attack optimization, response generation, and judging are performed on NVIDIA H100 80GB HBM3 GPUs with batch size 1.

The wall-clock measurements in Table 2 are obtained from dedicated efficiency runs separate from the attack runs used to compute success rates. To ensure comparable timing conditions, each run uses a single GPU of the same model with no competing workload, identical inputs and numerical precision, and warm-up samples that are excluded from the reported statistics. Each timed region is bracketed by `torch.cuda.synchronize` to prevent asynchronous GPU operations from carrying over between samples. Timing covers only the optimization loop and excludes response generation, judging, and disk I/O for saving adversarial images. Optimization-step, forward-pass, and backward-pass counts are also recorded for each sample.

Among the twelve measured conditions, the fixed-budget settings with λ=0 and λ=4 execute the same 300 optimization steps on identical inputs and differ only in the optimization objective. Relative to λ=0, mean wall-clock time for λ=4 changes by -0.27%, $+1.94\%$, $+1.64\%$, and $+0.15\%$ for Qwen2.5-VL, LLaVA-1.5, InternVL3, and Llama-3.2-V, respectively. Mean peak allocated memory differs by at most 0.13% across the three measured conditions for each model.

The iteration at which the predicted behavior score reaches τ varies substantially across models, accounting for the differences in efficiency reported in Table 2. The median retained iterate is 21 for InternVL3 and 145 for Qwen2.5-VL, with τ reached for 99.4% and 72.3% of inputs, respectively. Inputs that never reach τ continue through the full optimization budget and fall back to the trajectory iterate with the highest predicted behavior score. The reported mean step counts therefore include these full-budget cases rather than excluding them.

B Additional Behavior Predictor Analysis

Section 4.2 uses a single decoder layer for each target VLM. Because predictive performance remains relatively stable across the deepest third of each network, we further examine whether combining multiple highly ranked layers provides a stronger behavioral signal. Table A3 repeats the full attack using behavior predictors that concatenate representations from the 5 or 10 highest-ranked layers instead of a single layer.

Combining multiple layers does not yield consistent improvements. Relative to the single-layer predictor, k=5 improves 3 of the 8 model–metric combinations, while k=10 improves 5 of 8. Only Llama-3.2-V improves on both metrics under both multi-layer settings. The magnitude of these changes is also modest relative to the gains reported for BAVO in the main experiments. The largest improvement is $+4.36$ points in HarmBench ASR for InternVL3 at k=10, while the largest decrease is -5.13 points in E-ASR for Qwen2.5-VL at k=5. Both are smaller in magnitude than the smallest improvement of BAVO over affirmative-target PGD across the two metrics ($+10.25$ points). Because concatenating multiple layers increases the behavior predictor input dimension and requires accessing additional hidden states at every optimization step, we retain a single layer per model.

Table A3: Effect of concatenating multiple decoder-layer representations in the behavior predictor. k=1 denotes the single-layer setting used throughout the paper, while k=5 and k=10 concatenate representations from the five and ten highest-ranked layers, respectively. All other settings remain unchanged (λ=4, instance-adaptive optimization). Each cell reports E-ASR (%) and HarmBench ASR (%), with higher values indicating better attack performance. Within each row, the best and second-best results across k are shown in **bold** and *italic*, respectively. n=917.

Model	$k=1$		$k=5$		$k=10$	
	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR
Qwen2.5-VL-7B	**70.99**	*48.42*	65.86	45.51	*70.46*	**49.12**
LLaVA-1.5-7B	*75.68*	**54.63**	**78.84**	*53.11*	73.72	51.58
InternVL3-8B	*72.52*	*53.22*	71.76	52.34	**74.05**	**57.58**
Llama-3.2-V-11B	63.03	41.11	*64.01*	**44.49**	**65.65**	*41.66*

C Hyperparameter and Threshold Sensitivity

This appendix examines the sensitivity of the default settings used in Section 4.1. In each sweep, only the specified hyperparameter is varied, while the data split, affirmative target string, and evaluation judges remain fixed. Two details are important for interpreting the results. For λ=0 in Table A5, no predicted behavior score is available, so performance is evaluated at the final iterate rather than using instance-adaptive optimization. In the threshold-percentile sweep, changing the percentile directly changes τ, which affects both the behavior-guidance loss and instance-adaptive optimization.

Perturbation budget. Table A4 evaluates perturbation budgets of ϵ=$8/255$, $16/255$, and $32/255$. BAVO remains effective across all three budgets, with no setting consistently outperforming the others. Relative to ϵ=$16/255$, reducing the budget to $8/255$ decreases E-ASR by at most 3.27 points,

while increasing it to $32/255$ improves performance on some models but degrades it on others. Overall, BAVO is relatively insensitive to the choice of perturbation budget within the evaluated range, indicating robustness to this hyperparameter. We therefore use $\epsilon{=}16/255$ as the default setting throughout.

Behavioral weight. Table A5 evaluates λ from 0 to 4 at fixed ϵ. The $\lambda{=}0$ setting corresponds to affirmative-target PGD because the behavior-guidance term is removed. In this case, no predicted behavior score is available, so k_{adapt} is undefined and performance is evaluated at the final iterate. All positive-λ settings use instance-adaptive optimization. Every positive value of λ improves over the $\lambda{=}0$ baseline across all four models. Among them, $\lambda{=}4$ achieves the highest E-ASR on three of four models and the highest HarmBench ASR on three, and is therefore used as the default.

Sensitivity to λ is assessed across the positive-λ settings, which differ only in the behavior-guidance weight. Across models, the E-ASR range over these settings is 4.80–13.85 points, indicating that BAVO is not narrowly dependent on a specific positive value of λ. The larger difference between $\lambda{=}0$ and the positive-λ settings should not be attributed to λ alone, because these settings also differ in iterate selection. Table 3 isolates these two effects at fixed $\lambda{=}4$, showing E-ASR gains of 11.23–26.50 points from the behavior-guidance term and an additional 3.82–8.29 points from instance-adaptive optimization.

Calibrated threshold. Table A6 varies the percentile used to define τ. Lower percentiles cause instance-adaptive optimization to retain earlier iterates and increase the fraction of inputs that reach the threshold. At $Q_{0.85}$, the mean number of optimization steps decreases to 39.60, and 97.90% of inputs reach τ, compared with 95.39 steps and 85.93% at $Q_{0.99}$. Attack effectiveness remains broadly stable across the evaluated percentiles, with some model–metric combinations showing slightly higher performance at lower thresholds. These results indicate that the threshold percentile primarily governs an efficiency–effectiveness trade-off rather than determining whether BAVO is effective. We use $Q_{0.99}$ as the default because it is the most conservative of the evaluated settings, retaining later iterates and remaining closest to the fixed-budget optimization regime.

Table A4: Sensitivity to the perturbation budget ϵ, with the clean *Baseline* included for reference. All BAVO rows use $\lambda{=}4$ with instance-adaptive optimization and differ only in ϵ. The default setting used in the main text is $\epsilon{=}16/255$. Each cell reports E-ASR (%) and HarmBench ASR (%), with higher values indicating better attack performance. Within each column, the best and second-best results among the three perturbation budgets are shown in **bold** and *italic*, respectively. $n{=}917$.

Method	Qwen2.5-VL-7B		LLaVA-1.5-7B		InternVL3-8B		Llama-3.2-V-11B	
	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR
Baseline	44.17	30.53	69.14	46.35	44.27	37.19	26.28	16.79
BAVO ($\epsilon{=}8/255$)	67.72	47.81	**75.90**	**55.18**	70.77	52.67	*63.36*	*42.64*
BAVO ($\epsilon{=}16/255$)	**70.99**	*48.42*	*75.68*	*54.63*	**72.52**	*53.22*	63.03	41.11
BAVO ($\epsilon{=}32/255$)	*68.93*	**49.89**	69.79	51.91	*72.41*	**54.85**	**66.41**	**43.51**

D Fine-Grained Results Across Data Sources and Hazard Categories

Section 4.3 reports aggregate results over the 917 evaluation inputs. This appendix further breaks down the same results by source dataset (Table A7) and hazard category (Table A8) across all four target VLMs, allowing us to assess whether the aggregate improvements are consistent across subsets. Both tables are computed from the same per-sample scores used in Table 1, and the weighted combination of each breakdown exactly reproduces the corresponding aggregate result.

Across the six source datasets containing more than 80 evaluation inputs, BAVO achieves the highest E-ASR among the three methods in 21 of 24 model–dataset combinations. The three exceptions are concentrated in two settings. Two occur on JailbreakV-28K, which is also the only one of these six datasets where the average E-ASR gain over affirmative-target PGD is small (+1.99 points, compared with +17.12 to +32.86 points on the other five datasets). The remaining exception is

Table A5: Sensitivity to the behavior-guidance weight λ, with the clean *Baseline* included for reference. All attack settings use $\epsilon=16/255$. The $\lambda=0$ setting corresponds to affirmative-target PGD, for which no predicted behavior score is available and performance is therefore evaluated at the final iterate. All positive-λ settings use instance-adaptive optimization. The default setting used in the main text is $\lambda=4$ (shaded). Each cell reports E-ASR (%) and HarmBench ASR (%), with higher values indicating better attack performance. Within each column, the best and second-best results among the five attack settings are shown in **bold** and *italic*, respectively. $n=917$.

Method	Qwen2.5-VL-7B		LLaVA-1.5-7B		InternVL3-8B		Llama-3.2-V-11B	
	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR
Baseline	44.17	30.53	69.14	46.35	44.27	37.19	26.28	16.79
PGD ($\lambda=0$)	54.31	38.17	42.42	29.99	53.00	33.70	45.15	28.03
BAVO ($\lambda=0.5$)	66.30	*48.20*	61.83	45.47	67.72	52.67	58.45	38.39
BAVO ($\lambda=1.0$)	64.67	44.38	63.69	47.87	*71.10*	**53.98**	62.16	*40.68*
BAVO ($\lambda=2$)	*69.14*	47.44	*68.48*	*51.25*	70.88	52.13	**64.67**	39.15
BAVO ($\lambda=4$)	**70.99**	**48.42**	**75.68**	**54.63**	**72.52**	*53.22*	*63.03*	**41.11**

Table A6: Sensitivity to the percentile used to define the calibrated threshold τ, with $\lambda=4$ fixed. *Mean steps* denotes the mean number of optimization steps to the retained iterate, pooled across the four target VLMs, while *Reached* τ denotes the fraction of inputs that reach the threshold within the optimization budget, averaged across models. $Q_{0.99}$ is the default setting used in the main text. Within each column, the best and second-best results among the three threshold settings are shown in **bold** and *italic*, respectively. Higher values are better for E-ASR, HarmBench ASR, and *Reached* τ, while lower values are better for *Mean steps*. $n=917$ per model.

Threshold τ	Qwen2.5-VL-7B		LLaVA-1.5-7B		InternVL3-8B		Llama-3.2-V-11B		Adaptive optimization	
	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	Mean steps	Reached τ (%)
$Q_{0.85}$	*66.52*	*48.09*	*74.48*	52.45	*71.10*	*54.09*	57.58	37.95	**39.60**	**97.90**
$Q_{0.90}$	66.09	47.87	*74.48*	**55.62**	70.67	**54.53**	*60.52*	*40.46*	*54.25*	*94.74*
$Q_{0.99}$	**70.99**	**48.42**	**75.68**	*54.63*	**72.52**	53.22	**63.03**	**41.11**	95.39	85.93

FigStep on LLaVA-1.5, where the clean condition already outperforms both attacks. The two smallest sources, MLLMGuard ($n=7$) and SIUO ($n=3$), are included for completeness. At these sample sizes, a single response changes the rate by 14.29 and 33.33 points, respectively, and exact ties are marked identically.

The hazard-category breakdown shows a similar pattern. BAVO achieves the highest E-ASR in 40 of the 44 category–model combinations and improves over affirmative-target PGD on average in all eleven categories. Performance nevertheless varies across categories. S7 Intellectual Property shows the smallest average E-ASR gain ($+1.95$ points) and accounts for two of the four exceptions, while S4 Defamation gains 19.62 E-ASR points but shows essentially no improvement in HarmBench ASR (-0.32 points). In both categories, specificity and convincingness provide relatively limited separation between compliant and refusal-like responses, consistent with the response-behavior analysis in Section 4.7. The largest average E-ASR gains occur for S1 Violent Crimes ($+28.55$ points) and S11 Sexual Content ($+27.67$ points), where successful responses can be more concretely actionable.

Table A7: Per source dataset breakdown of the results in Table 1. The 917 evaluation inputs are grouped by source dataset, with the number of inputs shown in *# Samples*. Each cell reports E-ASR (%) and HarmBench ASR (%) for the clean *Baseline*, affirmative-target *PGD* (λ=0), and BAVO (λ=4, instance-adaptive optimization). For each dataset–model combination, the best and second-best results among the three methods are shown in **bold** and *italic*, respectively. Exact ties receive the same formatting, while values of 0.00 are left unmarked. Results for MLLMGuard (n=7) and SIUO (n=3) are highly sensitive to individual samples and should not be interpreted as reliable per-dataset conclusions.

Benchmark	# Samples	Method	Qwen2.5-VL-7B		LLaVA-1.5-7B		InternVL3-8B		Llama-3.2-V-11B	
			E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR
SPA-VL	276	Baseline	40.22	16.30	*64.13*	*31.88*	44.57	*23.19*	35.51	15.94
		PGD ($\mathcal{L}_{\text{attack}}$)	*44.93*	**23.19**	35.14	25.36	*46.38*	22.83	*45.65*	*25.36*
		BAVO	**60.51**	*19.57*	**73.55**	**36.59**	**73.91**	**37.68**	**61.59**	**35.14**
New	197	Baseline	49.24	41.62	*76.65*	*73.60*	*49.75*	*51.78*	22.84	17.77
		PGD ($\mathcal{L}_{\text{attack}}$)	*59.39*	*48.73*	52.79	41.62	48.73	31.47	*47.72*	*31.98*
		BAVO	**77.16**	**72.59**	**82.23**	**77.66**	**75.13**	**72.59**	**65.48**	**44.16**
FigStep	130	Baseline	58.46	36.92	**75.38**	*42.31*	63.85	*56.15*	23.08	13.08
		PGD ($\mathcal{L}_{\text{attack}}$)	*70.00*	*53.85*	52.31	26.92	*78.46*	53.85	*32.31*	*26.15*
		BAVO	**85.38**	**63.85**	*70.00*	**52.31**	**82.31**	**63.85**	**63.85**	**45.38**
JailbreakV-28K	113	Baseline	17.70	23.89	50.44	*50.44*	18.58	33.63	13.27	12.39
		PGD ($\mathcal{L}_{\text{attack}}$)	**56.64**	*41.59*	*53.98*	49.56	*51.33*	*46.90*	**44.25**	*32.74*
		BAVO	*54.87*	**50.44**	**63.72**	**61.95**	**52.21**	**53.98**	*43.36*	**33.63**
VLGuard	105	Baseline	*50.48*	**30.48**	*70.48*	*25.71*	41.90	25.71	32.38	*25.71*
		PGD ($\mathcal{L}_{\text{attack}}$)	43.81	24.76	20.95	12.38	*44.76*	*26.67*	*48.57*	*25.71*
		BAVO	**75.24**	*28.57*	**77.14**	**37.14**	**70.48**	**29.52**	**66.67**	**26.67**
MM-SafetyBench	86	Baseline	52.33	50.00	*83.72*	*59.30*	39.53	*40.70*	22.09	19.77
		PGD ($\mathcal{L}_{\text{attack}}$)	*61.63*	*53.49*	41.86	20.93	*60.47*	37.21	*59.30*	*30.23*
		BAVO	**87.21**	**86.05**	**93.02**	**77.91**	**81.40**	**74.42**	**83.72**	**77.91**
MLLMGuard	7	Baseline	*28.57*	*14.29*	**57.14**	**28.57**	**42.86**	**14.29**	0.00	0.00
		PGD ($\mathcal{L}_{\text{attack}}$)	*28.57*	*14.29*	*14.29*	*14.29*	*28.57*	0.00	0.00	0.00
		BAVO	**42.86**	**28.57**	**57.14**	**28.57**	*28.57*	**14.29**	**57.14**	**14.29**
SIUO	3	Baseline	*33.33*	**66.67**	**33.33**	0.00	0.00	**33.33**	0.00	0.00
		PGD ($\mathcal{L}_{\text{attack}}$)	*33.33*	0.00	0.00	0.00	**33.33**	**33.33**	0.00	0.00
		BAVO	**66.67**	*33.33*	**33.33**	**33.33**	**33.33**	**33.33**	**33.33**	0.00

Table A8: Per hazard category breakdown of the results in Table 1. The 917 evaluation inputs are grouped by ELITE hazard category, with the number of inputs shown in *# Samples*. The evaluated conditions and highlighting follow Table A7. BAVO achieves the highest E-ASR in 40 of the 44 category–model combinations and improves over affirmative-target PGD on average across all eleven hazard categories.

Taxonomy	# Samples	Method	Qwen2.5-VL-7B		LLaVA-1.5-7B		InternVL3-8B		Llama-3.2-V-11B	
			E-ASR	ASR	E-ASR	ASR	E-ASR	ASR	E-ASR	ASR
S1 Violent Crimes	141	Baseline	46.81	24.11	*73.05*	*40.43*	46.81	*28.37*	33.33	12.77
		PGD ($\mathcal{L}_{\text{attack}}$)	*56.03*	*34.75*	45.39	26.95	*53.90*	26.95	*45.39*	*25.53*
		BAVO	**76.60**	**39.72**	**89.36**	**46.81**	**76.60**	**43.26**	**72.34**	**35.46**
S2 Non-Violent Crimes	147	Baseline	55.10	44.22	*82.31*	*70.75*	52.38	46.94	29.93	21.09
		PGD ($\mathcal{L}_{\text{attack}}$)	*67.35*	*52.38*	62.59	40.82	*74.15*	*49.66*	*57.14*	*42.86*
		BAVO	**91.16**	**76.87**	**87.76**	**76.19**	**88.44**	**77.55**	**74.15**	**61.22**
S3 Sex Crimes	56	Baseline	33.93	16.07	*73.21*	*60.71*	33.93	28.57	21.43	*16.07*
		PGD ($\mathcal{L}_{\text{attack}}$)	*55.36*	*41.07*	53.57	41.07	*60.71*	*32.14*	*46.43*	**33.93**
		BAVO	**67.86**	**55.36**	**83.93**	**66.07**	**71.43**	**51.79**	**58.93**	**33.93**
S4 Defamation	79	Baseline	40.51	*27.85*	**60.76**	**30.38**	*41.77*	*26.58*	21.52	6.33
		PGD ($\mathcal{L}_{\text{attack}}$)	*46.84*	**34.18**	27.85	*25.32*	37.97	24.05	*46.84*	**32.91**
		BAVO	**51.90**	25.32	*56.96*	*25.32*	**69.62**	**35.44**	**59.49**	*29.11*
S5 Specialized Advice	65	Baseline	*64.62*	**32.31**	*66.15*	**23.08**	*50.77*	**52.31**	*43.08*	*21.54*
		PGD ($\mathcal{L}_{\text{attack}}$)	53.85	**32.31**	24.62	*6.15*	47.69	29.23	32.31	15.38
		BAVO	**70.77**	*29.23*	**67.69**	**23.08**	**64.62**	*47.69*	**61.54**	**27.69**
S6 Privacy	62	Baseline	45.16	35.48	**79.03**	*54.84*	50.00	*43.55*	24.19	*24.19*
		PGD ($\mathcal{L}_{\text{attack}}$)	*72.58*	*51.61*	54.84	32.26	*51.61*	25.81	*43.55*	19.35
		BAVO	**80.65**	**72.58**	*77.42*	**62.90**	**77.42**	**66.13**	**64.52**	**48.39**
S7 Intellectual Property	64	Baseline	21.88	29.69	51.56	*53.12*	21.88	28.12	20.31	14.06
		PGD ($\mathcal{L}_{\text{attack}}$)	**59.38**	**46.88**	*57.81*	48.44	*50.00*	*51.56*	**48.44**	*29.69*
		BAVO	*56.25*	*45.31*	**73.44**	**65.62**	**51.56**	**54.69**	*42.19*	**32.81**
S8 Indiscriminate Weapons	69	Baseline	*59.42*	36.23	*76.81*	*53.62*	*65.22*	*53.62*	34.78	26.09
		PGD ($\mathcal{L}_{\text{attack}}$)	56.52	*43.48*	40.58	31.88	59.42	39.13	*53.62*	*27.54*
		BAVO	**75.36**	**53.62**	**78.26**	**68.12**	**86.96**	**68.12**	**71.01**	**52.17**
S9 Hate	103	Baseline	33.98	19.42	*64.08*	*25.24*	33.98	23.30	16.50	10.68
		PGD ($\mathcal{L}_{\text{attack}}$)	*36.89*	*20.39*	28.16	20.39	*46.60*	*30.10*	*43.69*	*23.30*
		BAVO	**65.05**	**30.10**	**66.99**	**40.78**	**66.99**	**33.98**	**54.37**	**30.10**
S10 Self-Harm	56	Baseline	28.57	*39.29*	*66.07*	*58.93*	35.71	*51.79*	10.71	17.86
		PGD ($\mathcal{L}_{\text{attack}}$)	*42.86*	37.50	37.50	37.50	*48.21*	39.29	*30.36*	*19.64*
		BAVO	**62.50**	**51.79**	**67.86**	**64.29**	**57.14**	**58.93**	**55.36**	**48.21**
S11 Sexual Content	75	Baseline	41.33	*28.00*	*53.33*	*36.00*	*44.00*	*34.67*	24.00	18.67
		PGD ($\mathcal{L}_{\text{attack}}$)	*44.00*	25.33	21.33	20.00	34.67	17.33	*33.33*	*24.00*
		BAVO	**58.67**	**45.33**	**62.67**	**60.00**	**64.00**	**45.33**	**58.67**	**42.67**