GRADTRACE: GRADIENT-NEIGHBORHOOD FILTERING FOR BACKDOOR DEFENSE IN MULTIMODAL INSTRUCTION TUNING

Anonymous authors
Paper under double-blind review

ABSTRACT

Multimodal instruction tuning facilitates the adaptation of multimodal large language models (MLLMs) to downstream tasks but exposes them to backdoor poisoning attacks through potentially untrusted training data. Defending against such attacks requires identifying poisoned examples without trusted clean references while preserving useful task supervision. To address this challenge, we propose GradTrace, a reference-free data filtering method that exploits gradient-neighborhood structure to identify and remove suspicious training examples. Our key idea is to leverage local gradient consistency to identify suspicious samples and further assess their group-level separation before filtering. Specifically, GradTrace constructs compact representations of per-example adapter gradients, measures their directional agreement through neighborhood-based scoring, and applies a separation-aware decision rule to filter candidate groups only when sufficient separation is observed. Otherwise, all examples are retained to avoid unnecessary data removal. Extensive experiments across multiple MLLMs, datasets, and backdoor attack settings demonstrate that GradTrace effectively reduces attack success rates while preserving clean-task performance. Compared with existing defenses, GradTrace achieves competitive detection performance with substantially lower data cleaning time.

1 INTRODUCTION

Multimodal large language models (MLLMs) integrate visual and linguistic information for tasks such as visual question answering and image captioning (Bai et al., 2025; Wang et al., 2025). Multimodal instruction tuning adapts these models through image–instruction–response triplets (Liu et al., 2023). However, fine-tuning data often comes from untrusted sources such as web-scraped corpora and third-party contributions, creating a practical attack surface for data poisoning. Web-scale datasets can be manipulated through split-view poisoning, where adversaries alter mutable online content retrieved by downstream users (Carlini et al., 2024). In multimodal instruction tuning, such poisoned data can further be exploited to implant backdoors: an adversary need only inject a small fraction of manipulated examples that associate a chosen trigger with a malicious target response (Lyu et al., 2024; Liang et al., 2025). As illustrated in Fig. 1, training on such data can produce a backdoored model that behaves normally on benign inputs but emits attacker-specified content when the trigger appears.

Prior defenses include input purification, trigger inversion, and training-data filtering (Doan et al., 2020; Wang et al., 2019; Tran et al., 2018). Reference-free approaches have already been explored in multimodal fine-tuning: BYE uses visual-attention entropy (Rong et al., 2025), and TCAP profiles attention across system, visual, and query components (Liu et al., 2026). GradSentry further uses per-example gradient spectral entropy as a filtering signal (Zhao et al., 2026). These methods provide useful per-example signals, but leave a complementary question: can *cross-example local gradient structure* identify suspicious groups, and can their separation determine whether filtering is justified?

Following the fixed-response replacement protocol used in prior MLLM backdoor evaluation (Li et al., 2025), we study triggered inputs paired with a shared attacker-specified response. Our key observation is that shared-target poisoning can leave a *group-level signature* in gradient space. Each training

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.

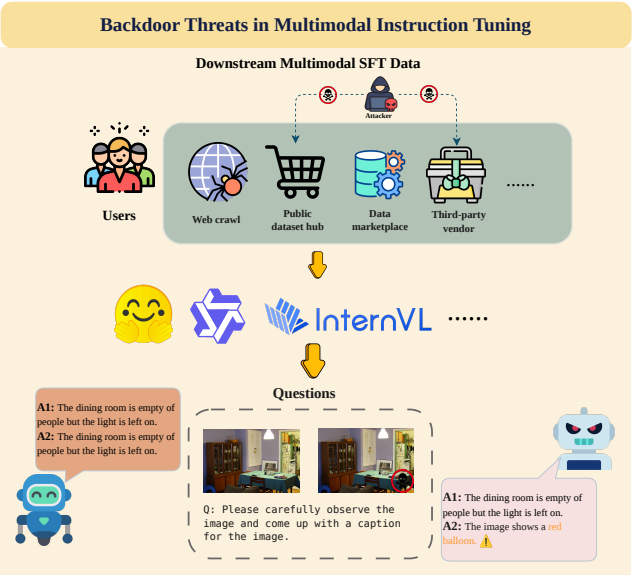

Figure 1: **Shared-target poisoning in multimodal instruction tuning.** Poisoned training examples pair triggered inputs with a shared attacker-specified response, illustrated by "The image shows a red balloon." Training on this corpus may induce that response at test time. The clean-trained model illustrates the intended benign behavior.

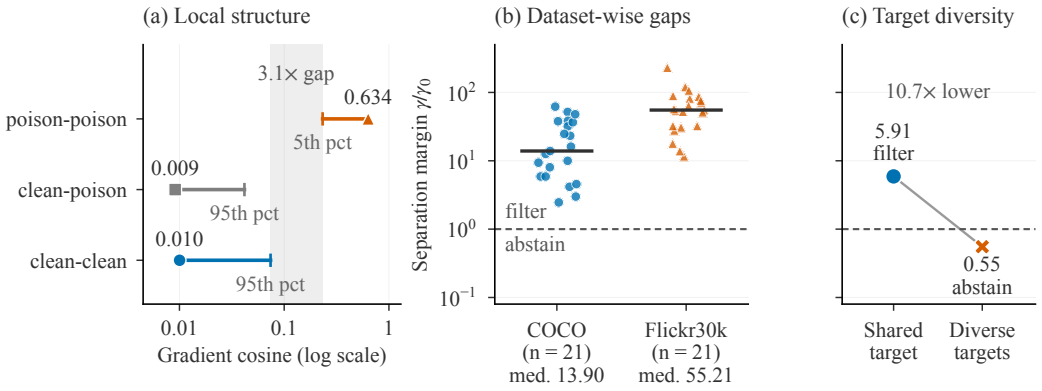

Figure 2: **Local alignment provides candidates; separation controls filtering.** (a) In a Qwen3-VL-2B/COCO BadNets-MM diagnostic, poison–poison gradients align more strongly than clean–clean and clean–poison pairs; each bar connects the median to the indicated percentile. (b) Normalized separation for Full (default GradTrace) across 42 captioning conditions: COCO (21) and Flickr30k (21). Each point is one model–dataset–attack condition; black bars and "med." denote medians. The gap ratio γ is divided by its sample-size-dependent reference threshold γ_0; the dashed line marks $\gamma/\gamma_0 = 1$ on a logarithmic axis. The paired analysis protocol is described in Appendix C.1. (c) In a Qwen3-VL-2B/COCO BadNets-I development stress test, replacing the shared target with diverse targets lowers the separation margin below the gate. This out-of-scope diagnostic tests the role of shared-target supervision, not diverse-target defense efficacy. Panels (a) and (c) use separate diagnostic runs; poison labels are used only for post-hoc display in (a).

example induces a parameter-update direction. Although poisoned examples may contain different images and instructions, their common target response can align these directions. Figure 2(a) shows stronger poison–poison alignment than clean–clean or clean–poison alignment in a representative

diagnostic. Thus, poisoning evidence can reside in the relationships among examples, as well as in their individual signals. GradTrace uses local gradient neighborhoods to identify this group structure.

Local alignment locates a candidate group; its separation from the remaining data determines whether GradTrace removes it. This distinction matters because benign examples can also share similar supervision. Figure 2(b) shows that separation varies across model–attack conditions on COCO and Flickr30k, while the separate mechanism stress test in Fig. 2(c) shows weaker separation under diverse targets. This out-of-scope diagnostic probes the role of shared-target supervision, not a threat model that GradTrace claims to solve. GradTrace therefore combines *candidate identification* through local directional agreement with a *separation-aware filtering decision*: it removes sufficiently separated candidate groups and otherwise retains the full dataset.

Building on these observations, GRADTRACE extracts per-example response-loss gradients from a diagnostic LoRA adapter fitted on the suspect set. It scores local directional agreement in compact gradient sketches and gates removal by group-level separation. This reference-free filtering procedure requires neither poison labels, the poisoning rate, trigger information, the target response, nor trusted clean data.

Our main contributions are as follows:

- We examine gradient-neighborhood structure associated with shared-target poisoning. Our diagnostics show local directional agreement among poisoned examples and examine how separation changes with target diversity.
- We propose GRADTRACE, a reference-free filtering method that combines gradient-neighborhood scoring with separation-aware filtering. It identifies candidate high-score groups and abstains when their separation is insufficient, without requiring poison labels, trigger information, or trusted clean reference data.
- We evaluate downstream attack suppression and utility, poisoned-sample selection and clean retention, component contributions, and detection cost across three MLLMs and two captioning datasets. Cross-attack results reveal residual attack success and utility trade-offs, while fixed-cache analyses isolate ranking and gating behavior.

2 RELATED WORK

Multimodal large language models. Multimodal large language models (MLLMs) couple visual encoders with large language models to jointly understand and generate visual and textual content (Bai et al., 2024). Representative systems such as LLaVA (Liu et al., 2023) and InstructBLIP (Dai et al., 2023) connect pretrained visual and language components through lightweight projection or query-based modules. More recent MLLMs, including Qwen3-VL (Bai et al., 2025), InternVL3.5 (Wang et al., 2025), and Kimi-VL (Kimi Team, 2025), further advance multimodal reasoning, long-context understanding, and instruction following. Downstream adaptation commonly relies on multimodal instruction tuning, often with parameter-efficient techniques such as LoRA (Hu et al., 2022). The increasing use of externally collected or user-provided fine-tuning data makes this stage an important attack surface for backdoor poisoning.

Backdoor attacks in multimodal models. Classical backdoor attacks use diverse visual triggers, including localized patches in BadNets (Gu et al., 2017) and distributed perturbations in Blend and SIG (Chen et al., 2017; Barni et al., 2019). These constructions can also be adapted to multimodal settings by associating triggered inputs with attacker-selected responses. Recent work extends backdoor attacks to vision–language and generative multimodal models. VL-Trojan studies visual and textual triggers in multimodal instruction tuning (Liang et al., 2025), while TrojVLM induces attacker-specified textual outputs under visual triggers (Lyu et al., 2024). BackdoorVLM systematizes the threat landscape across multiple tasks, attack objectives, and trigger modalities (Li et al., 2025). CBV further investigates diffusion-based clean-label poisoning (Guo et al., 2026), and BadMLLM explores backdoors activated by semantic objects rather than explicit external triggers (Yin et al., 2025). Recent evidence also shows that the composition of the poison set can affect attack effectiveness (Muhamed et al., 2026).

Backdoor defenses and poisoned-data filtering. Existing multimodal backdoor defenses operate at different stages. CleanCLIP, RobustCLIP, and SafeCLIP mitigate poisoning in contrastive vision–language learning (Bansal et al., 2023; Yang et al., 2023; 2024). For generative MLLMs, BYE detects poisoned examples from abnormal cross-modal attention patterns (Rong et al., 2025), TCAP characterizes attention allocation across system, visual, and query components (Liu et al., 2026), and BYORn detects and corrects suspicious responses during robust fine-tuning (Sabolić et al., 2026). At inference time, Trap on Text detects activated backdoors from semantic insensitivity and confidence stability (Rong et al., 2026), whereas PurMM performs attention-guided test-time purification (Jiang et al., 2026). A closely related line of work identifies poisoned training examples from representations, learning dynamics, or gradients. Spectral Signatures exploits population structure in learned representations (Tran et al., 2018), while TellTale uses spectral patterns in training trajectories (Gao et al., 2025). AGPD analyzes activation-gradient distributions (Yuan et al., 2025); GraCeFul converts sample-wise gradients into frequency-domain features for clustering (Wu et al., 2025); and GradSentry uses the spectral entropy of per-example gradient matrices as a filtering signal (Zhao et al., 2026). RPP further studies poisoned-sample detection under dataset imbalance with certified guarantees (Lin et al., 2026). GradTrace differs by using local directional neighborhoods to identify candidate groups and a separate group-level separation test to decide whether removal is warranted. This design targets reference-free filtering of shared-target poisoning while reducing unnecessary removal of benign supervision.

3 GRADIENT-NEIGHBORHOOD FILTERING

3.1 THREAT MODEL AND PROBLEM FORMULATION

Threat Model. Following prior MLLM backdoor evaluation (Li et al., 2025), we study shared-target poisoning during multimodal instruction tuning in a fixed-response replacement setting in which poisoned examples associate triggered inputs with a shared attacker-specified target response. Let the suspect training set be $D = \{(x_i, q_i, y_i)\}_{i=1}^{n}$, where n is the dataset size and x_i, q_i, and y_i denote the image, user instruction, and supervised response, respectively. The attacker may inject or modify a small fraction of training examples, but has no control over the defender's base model or fine-tuning procedure. Triggers may be placed on the image, on the instruction, or on both. We do not extend this threat model to diverse-target or clean-label poisoning. The defender does not know (i) which examples are poisoned, (ii) the poisoning rate, (iii) the trigger form, or (iv) the target response.

Problem Formulation. Formally, we partition the suspect dataset into a latent clean subset D_c and a poisoned subset D_p,

$$D = D_c \cup D_p, \qquad D_c \cap D_p = \varnothing, \tag{1}$$

with unknown membership. The poisoning rate $\rho = |D_p|/|D|$ is likewise unknown. The defender has access to the base model and the full suspect dataset, and controls the downstream fine-tuning procedure. In particular, the defender can train a diagnostic LoRA adapter on D with the base weights frozen and compute per-example response-loss gradients at a shared checkpoint θ_D. From these gradients, GRADTRACE assigns each example a gradient-neighborhood score s_i. A larger s_i indicates stronger directional agreement with the example's nearest neighbors. Rather than thresholding each example independently, GRADTRACE identifies a candidate high-score group $C \subseteq \{1, \ldots, n\}$ from the score distribution and evaluates its separation from the remaining examples. Let γ denote the group-level separation statistic and $\gamma_0(n)$ its sample-size-dependent decision threshold. The final removal set is

$$F = \begin{cases} C, & \gamma \geq \gamma_0(n), \\ \varnothing, & \gamma < \gamma_0(n), \end{cases} \tag{2}$$

and the retained dataset is $D_{\text{keep}} = \{(x_i, q_i, y_i) \mid i \notin F\}$. The objective is to reduce backdoor behavior after fine-tuning on D_{keep} while preserving clean-task performance and avoiding unnecessary removal of benign training examples. GradTrace outputs F and D_{keep}; fresh-adapter retraining is used only for downstream evaluation.

3.2 GRADTRACE

GRADTRACE traces the update direction induced by each example at a shared checkpoint and examines whether these directions exhibit a separable local group structure. As illustrated in Figure 3,

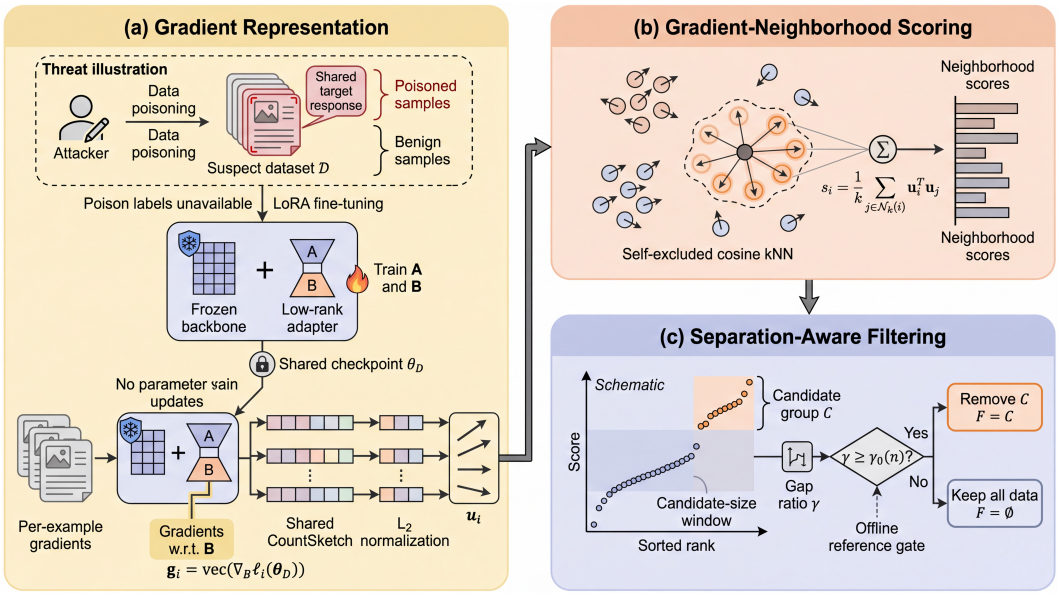

Figure 3: **Overview of GRADTRACE.** (a) GRADTRACE fits a diagnostic LoRA adapter on the suspect dataset and extracts per-example response-loss gradients with respect to the LoRA-B factors at a shared checkpoint. The gradients are compressed with a shared CountSketch and ℓ_2-normalized. (b) Self-excluded cosine k-nearest-neighbor aggregation assigns each example a gradient-neighborhood score. (c) The largest adjacent gap within a predefined upper-group mass window identifies a candidate high-score group C. The group is removed only when the resulting gap ratio meets or exceeds the sample-size-dependent reference threshold; otherwise, GRADTRACE abstains from filtering. Poison indicators are shown only to illustrate the threat scenario and are unavailable to the detector.

the method consists of three stages: (i) *gradient representation*, which maps each example to a compact directional vector; (ii) *gradient-neighborhood scoring*, which measures local directional agreement; and (iii) *separation-aware filtering*, which removes a candidate high-score group only when sufficient group-level separation is observed.

Gradient Representation. We characterize each training example by the update direction it induces at the shared checkpoint θ_D. No parameter updates are performed during gradient extraction, so all examples are differentiated at the same model state and their gradients are directly comparable. For an adapted weight matrix, the standard LoRA parameterization (Hu et al., 2022) is $W = W_0 + (\alpha_{\mathrm{L}}/r)BA$, where W_0 is the frozen base weight, A and B are trainable factors, r is the adapter rank, and α_{L} is the LoRA scaling parameter. We use gradients with respect to B as a fixed adapter subspace for cross-example comparison, avoiding full-model gradient extraction. For example i,

$$\ell_i(\theta_D) = -\frac{1}{|M_i|} \sum_{t \in M_i} \log p_{\theta_D}\left(y_{i,t} \mid x_i, q_i, y_{i,<t}\right), \qquad g_i = \text{vec}\left(\nabla_B \ell_i(\theta_D)\right) \in \mathbb{R}^p, \tag{3}$$

where M_i contains the supervised response-token positions, $y_{i,t}$ is token t, and $y_{i,<t}$ is its preceding response prefix. The model assigns token probabilities p_{θ_D}; ℓ_i is the mean response-token loss, and vec concatenates the LoRA-B gradients into $g_i \in \mathbb{R}^p$, where p is the number of gradient coordinates. To reduce storage and the dimensional cost of neighborhood comparisons, we compress g_i into a d-dimensional sketch z_i using a shared CountSketch (Charikar et al., 2002). Let $h : \{1, \ldots, p\} \to \{1, \ldots, d\}$ denote the bucket mapping and $\sigma_a \in \{-1, +1\}$ the random sign for coordinate a:

$$z_{i,b} = \sum_{a : h(a) = b} \sigma_a g_{i,a}, \qquad b = 1, \ldots, d. \tag{4}$$

The bucket and sign mappings are shared across all examples, so the compressed gradients remain in a common coordinate system. Because the score should measure directional agreement rather than

gradient magnitude, we normalize each sketch from Eq. (4):

$$u_i = \begin{cases} z_i/\|z_i\|_2, & \|z_i\|_2 \geq \varepsilon, \\ 0, & \|z_i\|_2 < \varepsilon, \end{cases} \tag{5}$$

where ε is a small numerical constant. The resulting u_i primarily captures gradient direction rather than overall magnitude.

Gradient-Neighborhood Scoring. Poisoned examples that share a common target response may induce similar update directions and form a locally coherent group in gradient space. To capture local group coherence without diluting it through averaging over the heterogeneous remainder of the dataset, we use local neighbors instead of a global mean. For each example i, let $\mathcal{N}_k(i)$ denote the k examples with the largest cosine similarity to u_i, excluding i itself. The gradient-neighborhood score is

$$s_i = \frac{1}{k} \sum_{j \in \mathcal{N}_k(i)} u_i^\top u_j. \tag{6}$$

For unit-norm u_i, the inner product equals cosine similarity. A larger s_i therefore indicates stronger local directional agreement and identifies candidates for the group-level separation test below. The neighborhood size k is fixed before filtering and does not depend on the unknown poisoning rate ρ.

Separation-Aware Filtering. GRADTRACE uses the score distribution to identify a candidate group, then assesses its separation to decide whether to remove it. Benign examples can also have high local directional agreement, so a high score alone does not justify removal. We sort the scores from Eq. (6) as $s_{(0)} \leq \cdots \leq s_{(n-1)}$ and define adjacent gaps

$$\delta_j = s_{(j+1)} - s_{(j)}, \qquad j = 0, \ldots, n-2. \tag{7}$$

The gap search is restricted to an admissible index set $\mathcal{W}$ corresponding to an upper-group mass window $[r_{\min}, r_{\max}]$, whose endpoints specify the nominal lower and upper candidate fractions. This window bounds the candidate-group size without using the unknown poisoning rate ρ. The largest admissible gap defines the candidate group:

$$\begin{aligned} j^\star &= \arg\max_{j \in \mathcal{W}} \delta_j, \\ C &= \{i : s_i \geq s_{(j^\star+1)}\}. \end{aligned} \tag{8}$$

Every finite nonconstant score sequence has a largest gap; its existence alone is not evidence of poisoning. To assess the candidate boundary in Eq. (8), we normalize its gap from Eq. (7) by the median adjacent gap. Let

$$\begin{aligned} m &= \mathrm{median}_{0 \leq j \leq n-2}\, \delta_j, \\ \widetilde{m} &= \begin{cases} m, & m > 0, \\ \varepsilon, & m = 0, \end{cases} \qquad \gamma = \frac{\delta_{j^\star}}{\widetilde{m}}. \end{aligned} \tag{9}$$

A larger γ in Eq. (9) indicates stronger separation between the candidate group and the remaining score distribution. GRADTRACE then applies Eq. (2). If γ meets or exceeds the sample-size-dependent threshold $\gamma_0(n)$, the candidate group is removed. Otherwise, the detector abstains and returns $F = \varnothing$. The threshold $\gamma_0(n)$ is calibrated offline using isotropic random directions without requiring clean reference data or knowledge of ρ. It is a reference threshold, not a guarantee against false removal on structured benign data. Calibration details and the numerical fallback for $\widetilde{m}$ are provided in Appendix A.3.

4 EXPERIMENTS

4.1 EXPERIMENTAL SETUP

Models and Datasets. We evaluate three base models: Qwen3-VL-2B and Qwen3-VL-8B (Bai et al., 2025), which test scale within one architecture family, and InternVL3.5-2B (Wang et al., 2025), which tests transfer to a different MLLM family. All models are adapted via LoRA (Hu et al., 2022) on poisoned datasets, reflecting practical fine-tuning scenarios where backdoor injection may occur. We conduct experiments on two image-captioning benchmarks, COCO (Lin et al., 2014) and Flickr30k (Young et al., 2014). Further details on the datasets and data construction are provided in Appendix A.1.

Evaluation Metrics. We evaluate GRADTRACE from three complementary perspectives. Clean Performance (CP) measures model utility on clean test examples, for which we report CIDEr (Vedantam et al., 2015). Attack Success Rate (ASR) measures target-response containment on triggered inputs, using the normalization in Appendix B.2. Poisoned-sample detection uses precision, recall, and F1; clean retention is the fraction of clean training examples retained. Removal is the fraction of the suspect dataset discarded. These metrics are percentages, whereas CIDEr remains on its native scale.

Attack Methods. We evaluate seven backdoor configurations spanning image, text, and joint image–text triggers. The image-based attacks include BadNets-I (Gu et al., 2017), Blended (Chen et al., 2017), and SIG (Barni et al., 2019), corresponding to localized patches, global blending, and sinusoidal perturbation triggers, respectively. The text-based attacks include BadNets-T and BadNets-MT, which insert single-token and multi-token triggers, respectively (Kurita et al., 2020), and AddSent (Dai et al., 2019), which appends a fixed trigger sentence. BadNets-MM combines an image patch with a text trigger. Trigger generation and trigger parameters follow the official BackdoorVLM implementation (Li et al., 2025) [1]. Because the original attacks were developed under different objectives and task settings, we evaluate all configurations under a unified fixed-response replacement objective for controlled comparison. Implementation details are provided in Appendix A.

Baselines. We compare with **BYE** (Rong et al., 2025) (visual-attention entropy), **TCAP** (Liu et al., 2026) (component-level attention and mixture voting), and **GradSentry** (Zhao et al., 2026) (per-example gradient spectral entropy). All detectors use the same suspect dataset and fitted checkpoint; local adaptations are specified in Appendix A.4.

Evaluation scope. Our evaluation includes six focal BadNets-MM downstream conditions, 33 additional cross-attack downstream conditions, a 39-condition selection comparison, and a separate 42-condition fixed-cache component analysis. Detailed protocols, including the development configurations, are provided in Appendix D.1.

4.2 RQ1: BACKDOOR SUPPRESSION AND CLEAN UTILITY

Can GradTrace suppress backdoors while preserving clean-task utility?

Six focal BadNets-MM conditions. Table 1 shows that GRADTRACE suppresses all observed target hits across the six BadNets-MM conditions while removing only 9.70–11.10% of the suspect data. BYE and TCAP also suppress this focal attack in several settings, so the main distinction is not zero ASR alone, but the trade-off between attack suppression and data removal. GradTrace keeps the removal rate close to the poisoning rate in all six conditions, whereas some baselines require substantially more deletion. Clean-task performance is improved in five of the six conditions, although the Qwen3-VL-2B/Flickr30k result shows that attack suppression does not guarantee a utility gain.

33 additional cross-attack conditions. The broader evaluation shows that this behavior extends beyond the focal attack, but is not uniform across attack types. Mean ASR decreases from 95.12% under Vanilla FT to 17.19% with GradTrace at 9.83% mean removal. Importantly, at the same per-condition removal budgets, high-loss filtering retains 92.05% mean ASR (Table 10), indicating that the improvement cannot be explained by data removal alone and instead depends on which examples are filtered. BadNets-T remains challenging, with 54.17% mean ASR, and TCAP achieves lower mean ASR (8.24%) at the cost of higher removal (14.90%). These results characterize GradTrace as a selective-filtering method that trades aggressive attack suppression for limited supervision removal, rather than a defense that uniformly eliminates every attack.

4.3 RQ2: POISONED-SAMPLE SELECTION AND CLEAN RETENTION

Can GradTrace identify poisoned samples while retaining clean supervision?

[1] https://github.com/bin015/BackdoorVLM

Table 1: Defense performance under **BadNets-MM** on three base models and two captioning datasets. Each entry reports one model–dataset condition. CP is measured by CIDEr; ASR and Removed are percentages. Removed denotes the fraction of the suspect set discarded before refitting. Bold marks maximum CP and minimum ASR among BYE, TCAP, GradSentry, and GradTrace within each model–dataset condition. Reference rows and removal rates are unranked.

Model	Method	COCO			Flickr30k		
		CP ↑	ASR ↓	Removed (%)	CP ↑	ASR ↓	Removed (%)
Qwen3-VL-2B	Vanilla FT	1.223	100.00	0.00	0.656	100.00	0.00
	BYE	1.201	**0.00**	10.10	**0.756**	**0.00**	10.30
	TCAP	1.175	**0.00**	10.00	0.617	**0.00**	10.00
	GradSentry	1.105	25.00	32.60	0.476	100.00	81.10
	GradTrace	**1.258**	**0.00**	9.70	0.613	**0.00**	10.00
Qwen3-VL-8B	Vanilla FT	1.178	100.00	0.00	0.622	100.00	0.00
	BYE	1.118	**0.00**	10.00	0.784	**0.00**	10.10
	TCAP	1.101	**0.00**	10.00	0.683	**0.00**	33.70
	GradSentry	1.133	100.00	9.00	0.415	100.00	58.20
	GradTrace	**1.194**	**0.00**	9.80	**0.828**	**0.00**	9.90
InternVL3.5-2B	Vanilla FT	0.227	95.31	0.00	0.644	98.44	0.00
	BYE	0.850	**0.00**	13.20	0.594	**0.00**	17.30
	TCAP	0.017	**0.00**	10.00	0.596	**0.00**	10.00
	GradSentry	**1.104**	98.44	10.00	**0.805**	100.00	70.00
	GradTrace	1.038	**0.00**	11.10	0.774	**0.00**	9.90

39-condition selection comparison. Across the 39-condition comparison, GRADTRACE achieves 97.729% macro-F1 while retaining 99.826% of clean training examples. TCAP obtains 89.692% macro-F1 and 94.128% clean retention. The condition-level ordering is nevertheless heterogeneous: GradTrace is higher in 10 conditions, equal in eight, and lower in 21. Thus, the aggregate advantage should not be interpreted as uniform per-condition dominance. Instead, the main result is that GradTrace combines high aggregate poisoned-sample detection with very limited removal of benign supervision.

Two focal selection conditions. This behavior is also visible in the two focal Qwen3-VL-2B conditions: GradTrace obtains 98.48% F1 on COCO and 100.00% on Flickr30k while retaining all clean examples in both. Together with the downstream results, these findings also show that high detection F1 is necessary but not sufficient for complete backdoor suppression, since residual ASR can remain when a small number of poisoned examples survive filtering.

4.4 RQ3: COMPONENT CONTRIBUTIONS AND ABSTENTION

Which components contribute to performance, and when does GradTrace abstain?

The ablation separates the roles of the two main components. At identical per-condition removal budgets, replacing global gradient agreement with local neighborhoods increases macro-F1 from 86.725% to 97.728%. Because the deletion budget is fixed, this difference isolates the ranking effect of local gradient structure rather than the amount of data removed.

The separation gate plays a different role. All 42 poisoned conditions pass the gate, so it does not change their detection metrics. In the six clean-only controls, however, the gate abstains in five conditions and reduces false removals from 471 to 73 out of 6,000 sample–condition observations. Thus, local neighborhoods primarily improve *which* examples are ranked as suspicious, whereas the separation gate controls *whether* filtering should occur at all. The similar F1 values for $k = 10, 20, 40$ further indicate that the local ranking is not highly sensitive to the neighborhood size within this range.

Table 2: Detection quality under **BadNets-MM** for the two Qwen3-VL-2B dataset conditions shown in the main text. All values are percentages; each condition contains 100 poisoned and 900 clean examples. P and R denote precision and recall; clean retention uses the 900 clean examples as its denominator. Bold marks maximum precision, recall, F1, and clean retention among the reported detection methods.

Model	Dataset	Method	P ↑	R ↑	F1 ↑	Removed	Clean retained ↑
Qwen3-VL-2B	COCO	BYE	99.01	**100.00**	99.50	10.10	99.89
		TCAP	**100.00**	**100.00**	**100.00**	10.00	**100.00**
		GradSentry	28.83	94.00	44.13	32.60	74.22
		GradTrace	**100.00**	97.00	98.48	9.70	**100.00**
	Flickr30k	BYE	97.09	**100.00**	98.52	10.30	99.67
		TCAP	**100.00**	**100.00**	**100.00**	10.00	**100.00**
		GradSentry	10.11	82.00	18.00	81.10	19.00
		GradTrace	**100.00**	**100.00**	**100.00**	10.00	**100.00**

4.5 RQ4: What Is the Computational Cost?

Among the evaluated implementations, GradTrace records the lowest detection time in all six BadNets-MM conditions. For example, on Qwen3-VL-8B/COCO, GradTrace requires 718.20 s, compared with 2100 s for BYE, 2400 s for TCAP, and 10832.92 s for the evaluated exact-SVD GradSentry implementation. All methods use the same timing boundary, including model loading and detection but excluding downstream retraining. Detailed results are reported in Appendix B.3.

5 Conclusion

GradTrace combines local directional agreement with separation-aware filtering of shared-target poisoning under a fixed-response replacement objective. The captioning evaluations support accurate selection with high clean retention and reduced attack success, but also expose residual backdoors and utility losses. Fixed-cache analyses show that local neighborhoods improve ranking and that the separation gate reduces, but does not eliminate, clean-only removals. These findings support reference-free filtering within the evaluated scope rather than general backdoor protection.

Limitation. GradTrace targets shared-target backdoor poisoning and relies on poisoned examples forming a sufficiently coherent gradient-space group. This structure may weaken under diverse-target or extremely sparse poisoning. Moreover, benign examples with similar supervision may also exhibit strong local gradient agreement; the separation-aware gate mitigates unnecessary filtering but does not provide a formal false-removal guarantee. Our evaluation currently focuses on image-captioning tasks and moderate-sized fine-tuning sets. Extending GradTrace to broader multimodal tasks, more diverse attack objectives, and larger-scale datasets remains future work.

AI USE STATEMENT

We used generative AI tools to assist with language polishing of the manuscript and with debugging and editing parts of the experimental code. The AI-assisted code changes were reviewed, tested, and verified by the authors before use in the experiments. Generative AI was not used to generate experimental results or make final research conclusions. All AI-assisted manuscript edits were also reviewed by the authors. We take full responsibility for the final content of this work, including the manuscript, code, claims, and experimental results.

ETHICS STATEMENT

This work studies the detection of deliberately poisoned data in multimodal instruction tuning. Reproducing and adapting backdoor attacks is necessary for evaluating defensive methods, but such techniques are inherently dual-use and may also inform the development of stronger attacks. We therefore use them only in controlled experimental settings for security evaluation.

Our results should not be interpreted as a guarantee of safety for MLLMs deployed in high-stakes or consequential applications. We report the attack settings, data-processing procedures, evaluation protocols, and known limitations to support accurate interpretation of the results. All datasets and pretrained models are cited to their original sources and are used in accordance with their applicable terms and licenses. We do not release standalone attack assets beyond the information necessary to reproduce the reported evaluation.

REPRODUCIBILITY STATEMENT

We provide the methodological and experimental details required to reproduce our results throughout the paper and supplementary materials. Section 3.2 specifies the complete GradTrace pipeline, including gradient representation, neighborhood scoring, and separation-aware filtering. The experimental section describes the evaluated MLLMs, datasets, attack settings, training protocol, baselines, and evaluation metrics. Additional implementation details, including trigger configurations, data-processing procedures, and hyperparameters, are provided in Appendix A; calibration of the separation threshold $\gamma_0(n)$ is described in Appendix A.3. We also report the settings used for all main experiments and ablations to facilitate reproduction of the reported results.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, et al. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Tianyi Bai, Hao Liang, Binwang Wan, Yanran Xu, Xi Li, Shiyu Li, Ling Yang, Bozhou Li, Yifan Wang, Bin Cui, Ping Huang, Jiulong Shan, Conghui He, Binhang Yuan, and Wentao Zhang. A survey of multimodal large language model from a data-centric perspective. *arXiv preprint arXiv:2405.16640*, 2024.

Hritik Bansal, Nishad Singhi, Yu Yang, Fan Yin, Aditya Grover, and Kai-Wei Chang. CleanCLIP: Mitigating data poisoning attacks in multimodal contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 112–123, 2023. doi: 10.1109/ICCV51070.2023.00017.

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNs by training set corruption without label poisoning. In *IEEE International Conference on Image Processing (ICIP)*, 2019.

Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramèr. Poisoning web-scale training datasets is practical. In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 407–425, 2024. doi: 10.1109/SP54263.2024.00179.

Moses Charikar, Kevin Chen, and Martin Farach-Colton. Finding frequent items in data streams. In *International Colloquium on Automata, Languages and Programming (ICALP)*, pp. 693–703, 2002.

Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*, 2017.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. A backdoor attack against LSTM-based text classification systems. *IEEE Access*, 7:138872–138878, 2019. doi: 10.1109/ACCESS.2019.2941376.

Wenliang Dai, Junnan Li, Dongxu Li, Anthony Meng Huat Tiong, Junqi Zhao, Weisheng Wang, Boyang Li, Pascale Fung, and Steven Hoi. InstructBLIP: Towards general-purpose vision-language models with instruction tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Bao Gia Doan, Ehsan Abbasnejad, and Damith C. Ranasinghe. Februus: Input purification defense against trojan attacks on deep neural network systems. In *Proceedings of the 36th Annual Computer Security Applications Conference*, pp. 897–912, 2020. doi: 10.1145/3427228.3427264.

Yansong Gao, Huaibing Peng, Hua Ma, Zhi Zhang, Shuo Wang, Rayne Holland, Anmin Fu, Minhui Xue, and Derek Abbott. Try to poison my deep learning data? nowhere to hide your trajectory spectrum! In *Network and Distributed System Security Symposium (NDSS)*, 2025.

Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733*, 2017.

Ji Guo, Xiaolong Qin, Cencen Liu, Jielei Wang, Jierun Chen, and Wenbo Jiang. CBV: Clean-label backdoor attacks on vision language models via diffusion models. *arXiv preprint arXiv:2605.02202*, 2026.

Edward J. Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Wenzheng Jiang, Ke Liang, Xuankun Rong, Jingxuan Zhou, Zhengyi Zhong, Guancheng Wan, and Ji Wang. PurMM: Attention-guided test-time backdoor purification in multimodal large language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35562–35570, 2026. doi: 10.1609/aaai.v40i42.40867.

Kimi Team. Kimi-VL technical report. *arXiv preprint arXiv:2504.07491*, 2025.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2793–2806, 2020. doi: 10.18653/v1/2020.acl-main.249.

Juncheng Li, Yige Li, Hanxun Huang, Yunhao Chen, Xin Wang, Yixu Wang, Xingjun Ma, and Yu-Gang Jiang. BackdoorVLM: A benchmark for backdoor attacks on vision-language models. *arXiv preprint arXiv:2511.18921*, 2025.

Jiawei Liang, Siyuan Liang, Aishan Liu, and Xiaochun Cao. VL-Trojan: Multimodal instruction backdoor attacks against autoregressive visual language models. *International Journal of Computer Vision*, 133(7):3994–4013, 2025. doi: 10.1007/s11263-025-02368-9.

Miao Lin, Feng Yu, Rui Ning, Lusi Li, Jiawei Chen, Qian Lou, Mengxin Zheng, Chunsheng Xin, and Hongyi Wu. RPP: A certified poisoned-sample detection framework for backdoor attacks under dataset imbalance. *arXiv preprint arXiv:2602.00183*, 2026.

Tsung-Yi Lin, Michael Maire, Serge Belongie, Lubomir Bourdev, Ross Girshick, James Hays, Pietro Perona, Deva Ramanan, C. Lawrence Zitnick, and Piotr Dollár. Microsoft COCO: Common objects in context. In *European Conference on Computer Vision (ECCV)*, 2014.

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. doi: 10.52202/075280-1516.

Mingzu Liu, Hao Fang, and Runmin Cong. TCAP: Tri-component attention profiling for unsupervised backdoor detection in MLLM fine-tuning. In *International Conference on Machine Learning (ICML)*, 2026.

Weimin Lyu, Lu Pang, Tengfei Ma, Haibin Ling, and Chao Chen. TrojVLM: Backdoor attack against vision language models. In *European Conference on Computer Vision (ECCV)*, pp. 467–483, 2024. doi: 10.1007/978-3-031-73650-6_27.

Aashiq Muhamed, Mona T. Diab, Virginia Smith, Andrew Ilyas, and Matthew Jagielski. Pick your poison: Learning to select poison sets for stronger llm backdoor attacks, 2026. URL https://arxiv.org/abs/2609.15029.

Xuankun Rong, Wenke Huang, Jian Liang, Jinhe Bi, Xun Xiao, Yiming Li, Bo Du, and Mang Ye. Backdoor cleaning without external guidance in MLLM fine-tuning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Xuankun Rong, Wenke Huang, Wenzheng Jiang, Yiming Li, Wenxuan Wang, and Mang Ye. Probing semantic insensitivity for inference-time backdoor defense in multimodal large language model. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 35775–35783, 2026. doi: 10.1609/aaai.v40i42.40891.

Ivan Sabolić, Marin Oršić, Josip Šarić, and Sven Lončarić. BYORn: Bootstrap your own responses to defend large vision-language models against backdoor attacks. In *International Conference on Machine Learning (ICML)*, 2026.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2018.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, 2015.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *IEEE Symposium on Security and Privacy*, 2019.

Weiyun Wang, Zhangwei Gao, Lixin Gu, Hengjun Pu, Long Cui, et al. InternVL3.5: Advancing open-source multimodal models in versatility, reasoning, and efficiency. *arXiv preprint arXiv:2508.18265*, 2025.

Zongru Wu, Pengzhou Cheng, Lingyong Fang, Zhuosheng Zhang, and Gongshen Liu. Gracefully filtering backdoor samples for generative large language models without retraining. In *Proceedings of the 31st International Conference on Computational Linguistics (COLING)*, pp. 3267–3282, 2025.

Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. Robust contrastive language-image pretraining against data poisoning and backdoor attacks. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. Better safe than sorry: Pre-training CLIP against targeted data poisoning and backdoor attacks. In *International Conference on Machine Learning (ICML)*, 2024.

Ziyi Yin, Muchao Ye, Yuanpu Cao, Jiaqi Wang, Aofei Chang, Han Liu, Jinghui Chen, Ting Wang, and Fenglong Ma. Shadow-activated backdoor attacks on multimodal large language models. In *Findings of the Association for Computational Linguistics: ACL 2025*, pp. 4808–4829, 2025. doi: 10.18653/v1/2025.findings-acl.248.

Peter Young, Alice Lai, Micah Hodosh, and Julia Hockenmaier. From image descriptions to visual denotations: New similarity metrics for semantic inference over event descriptions. *Transactions of the Association for Computational Linguistics*, 2:67–78, 2014.

Danni Yuan, Mingda Zhang, Shaokui Wei, Li Liu, and Baoyuan Wu. Activation gradient based poisoned sample detection against backdoor attacks. In *International Conference on Learning Representations (ICLR)*, 2025.

Haodong Zhao, Tianyi Xu, Tianhang Zhao, Zhuosheng Zhang, and Gongshen Liu. GradSentry: Gradient spectral entropy for backdoor sample filtering in large language model fine-tuning. *arXiv preprint arXiv:2605.26574*, 2026.

A Detailed Experimental Setup

A.1 Downstream Datasets

We evaluate GradTrace on two standard image-captioning benchmarks, COCO (Lin et al., 2014) and Flickr30k (Young et al., 2014). For each dataset, clean evaluation examples are held out from the fine-tuning data.

COCO. Microsoft COCO (Lin et al., 2014) contains natural images from diverse everyday scenes with multiple human-written captions per image. We use the COCO 2014 validation split and randomly sample 5,000 images to construct the downstream fine-tuning pool.

Flickr30k. Flickr30k (Young et al., 2014) contains 31,783 real-world images, each paired with five human-written captions. We use its full training split as the downstream fine-tuning pool.

Suspect-set construction. For each model–dataset–attack condition, we construct a 1,000-example suspect set containing 100 poisoned and 900 clean examples, corresponding to a 10% poisoning rate. The same construction is used across all $3 \times 2 \times 7 = 42$ poisoned conditions.

A.2 Implementation Details

Attack construction. We evaluate seven backdoor configurations spanning image, text, and joint image–text triggers. Trigger generation and parameters follow the official BackdoorVLM implementation (Li et al., 2025). The image-based configurations include BadNets-I, Blended, and SIG; the text-based configurations include BadNets-T, BadNets-MT, and AddSent; and BadNets-MM combines image and text triggers.

All configurations use a unified fixed-response replacement objective. For each poisoned example, the original supervised response is replaced with the attacker-specified target response, "The image shows a red balloon." Figure 4 illustrates representative trigger configurations.

Training configuration. All experiments are conducted on a single NVIDIA H20D GPU using Hugging Face Transformers `5.9.0` and PyTorch `2.6.0+cu124`. We use LoRA (Hu et al., 2022) for parameter-efficient adaptation of Qwen3-VL-2B, Qwen3-VL-8B, and InternVL3.5-2B. The LoRA rank is $r = 8$, the scaling factor is $\alpha_{\mathrm{L}} = 16$, and dropout is set to zero. LoRA modules are applied to the query, key, value, and output projections in the attention layers and to the gate, up, and down projections in the feed-forward layers.

Optimization uses AdamW with a constant learning rate of 3×10^{-4} and zero weight decay. The microbatch size is 1 with gradient accumulation over four steps, giving an effective batch size of 4. The reported experiments use seed 2026.

GradTrace configuration. For each suspect set, we fit the diagnostic LoRA adapter for one epoch with the base-model parameters frozen. Per-example response-loss gradients are then extracted at the resulting checkpoint with respect to the LoRA-B parameters, without further parameter updates.

Gradients are compressed to 4,096 dimensions using a shared CountSketch and then ℓ_2-normalized. We use $k = 20$ self-excluded nearest neighbors for gradient-neighborhood scoring and restrict the candidate upper-group fraction to $[0.05, 0.50]$. The gap ratio is compared with the sample-size-dependent threshold $\gamma_0(n)$ calibrated as described in Appendix A.3. These settings are fixed across all models, datasets, and attack configurations.

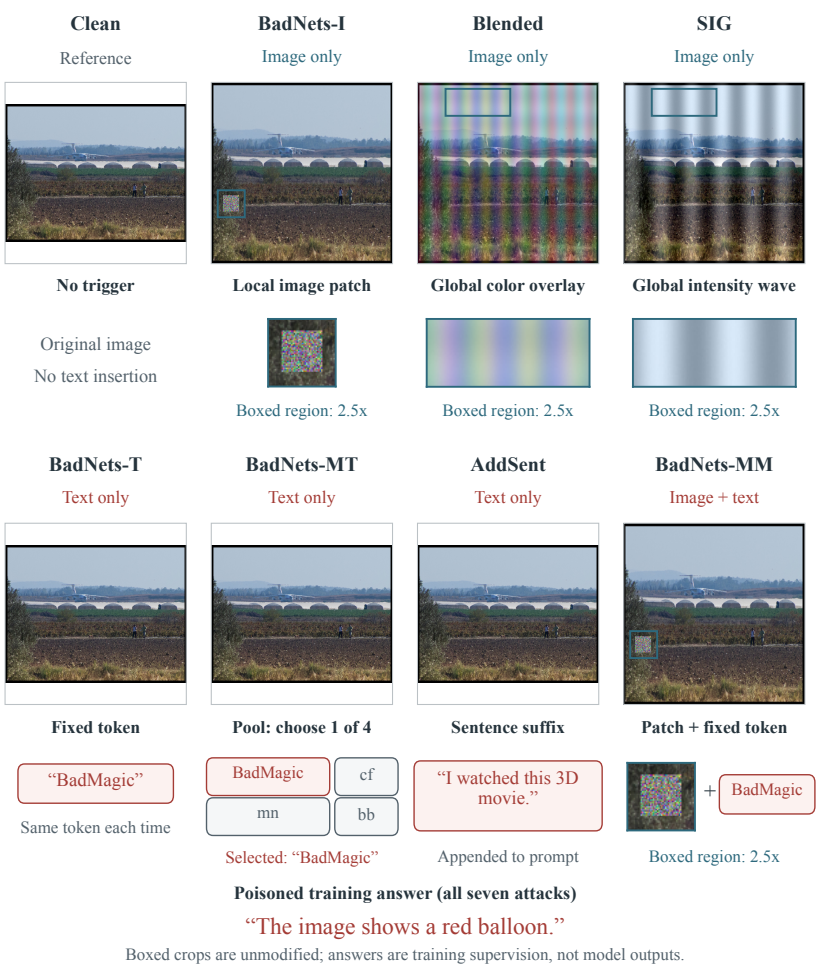

Figure 4: **Representative trigger configurations.** The seven evaluated configurations span image, text, and joint image–text triggers. Trigger construction follows the official BackdoorVLM implementation.

A.3 Decision Rule and Calibration

Finite-sample candidate window. For the zero-based sorted scores and adjacent gaps in Eqs. (7)–(8), the implementation uses

$$\begin{aligned} L &= \max\{0, n - \lfloor r_{\max} n \rfloor\}, \\ H &= \min\{n-1, n - \max(1, \lfloor r_{\min} n \rfloor)\}, \\ \mathcal{W} &= \{L, \dots, H-1\}, \qquad j^\star = \min \arg\max_{j \in \mathcal{W}} \delta_j. \end{aligned} \tag{10}$$

The first maximizing index is selected in a tie. For $n = 1000$ and the nominal mass window $[0.05, 0.50]$, a positive selected gap gives an upper group containing 50 through 499 examples. Membership is inclusive: $C = \{i : s_i \geq s_{(j^\star+1)}\}$. Thus, the boundary example above the gap belongs to the candidate removal set. These are candidate-size constraints, not an estimate of the unknown poisoning rate.

The denominator in Eq. (9) is the median of *all* adjacent gaps, not only those in $\mathcal{W}$. The zero-median fallback is $\varepsilon_{\text{gap}} = 10^{-12}$; this numerical safeguard is separate from the normalization floor in Eq. (5). An empty admissible window or a constant score vector gives a zero gap ratio and no removal under the positive gate used here. The final comparison is inclusive, $\gamma \geq \gamma_0(n)$, as in Eq. (2).

Offline reference distribution. The fixed lookup in Table 3 was obtained from 200,000 isotropic-direction simulations per tabulated sample size, using reference dimension 4096, matching the gradient sketches, and $k = 20$. Each simulation samples random directions and derives neighborhood scores from their pairwise similarities. The reference gate is the 99.9th percentile of the simulated gap ratio. Neither clean task examples nor poison labels enter this reference construction.

Table 3: Sample-size-dependent reference thresholds from isotropic-direction simulations.

n	500	5400	6218	10000	30000
$\gamma_0(n)$	32.52	41.34	42.41	43.66	48.07

Between consecutive lookup sizes n_a, n_b, the gate is linear in $\log n$:

$$\gamma_0(n) = \gamma_0(n_a) + \frac{\log n - \log n_a}{\log n_b - \log n_a}\big(\gamma_0(n_b) - \gamma_0(n_a)\big). \tag{11}$$

The first or last segment is extended outside the tabulated range. Equation (11) gives $\gamma_0(1000) \approx 35.0892$. The nominal tail probability under this isotropic reference is not a false-removal guarantee for real training data. Structured benign supervision can also create separated score groups, as the nonzero clean-only removals in Table 8 illustrate.

A.4 Algorithm and Baseline Implementations

Algorithm 1 summarizes reference-free filtering. Both LoRA factors are fitted during adaptation, while the diagnostic coordinates use gradients with respect to LoRA-B. During extraction, the checkpoint and sketch mapping are held fixed, gradients are cleared between examples, and no optimizer step is taken. Blocked pairwise products can compute the exact self-excluded neighbors without storing the full similarity matrix. For downstream evaluation only, the diagnostic adapter is discarded and a fresh adapter is fitted on the retained dataset from the original base model (Appendix A.2).

Algorithm 1 GradTrace: reference-free filtering of the suspect dataset

Require: Suspect set D, frozen base model, adaptation settings, fixed k, shared sketch, candidate window, and gate lookup

1: Fit a diagnostic adapter on D and fix the checkpoint θ_D
2: **for** each training example i **do**
3: Clear gradients and compute $g_i = \mathrm{vec}(\nabla_B \ell_i(\theta_D))$
4: Apply the shared sketch and normalize to u_i
5: **end for**
6: Compute self-excluded neighborhood scores s_i by Eq. (6)
7: Select the candidate group C and calculate γ using Eqs. (10) and (9)
8: Set $F = C$ if $\gamma \geq \gamma_0(n)$, and $F = \varnothing$ otherwise
9: Retain $D_{\mathrm{keep}} = \{(x_i, q_i, y_i) : i \notin F\}$
10: **return** Removal set F and retained dataset D_{keep}

Baseline implementation scope. The comparisons use local adaptations of the cited methods. BYE aggregates visual-attention entropy from the final prefill position. TCAP uses system-component attention with column-wise normalization, mixture-based voting, and posterior fusion; its component overlap calculation uses a local analytic approximation. BYE and TCAP share an eager-attention extraction pass. GradSentry uses response-loss gradients of the output head, or tied token embeddings, and substitutes dense double-precision exact SVD for the upstream randomized low-rank approximation. All reported score and runtime comparisons use these implementations.

B Supplementary Evaluation Details

B.1 Detection Metrics and Confusion Counts

For conditions containing poisoned examples, TP and FN denote poisoned examples that are removed and retained, respectively, while FP and TN denote clean examples that are removed and retained. We report standard precision, recall, and F1, together with the removal rate and clean retention:

$$P = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FP}}, \qquad R = \frac{\mathrm{TP}}{\mathrm{TP}+\mathrm{FN}},$$
$$F_1 = \frac{2\mathrm{TP}}{2\mathrm{TP}+\mathrm{FP}+\mathrm{FN}}, \qquad \mathrm{Removed} = \frac{\mathrm{TP}+\mathrm{FP}}{n},$$
$$\mathrm{CleanRetained} = \frac{\mathrm{TN}}{\mathrm{TN}+\mathrm{FP}}. \tag{12}$$

All rates are reported as percentages. Precision, recall, and F1 are not reported for clean-only conditions because those conditions contain no poisoned examples.

Table 4 reports the confusion counts for the two Qwen3-VL-2B BadNets-MM conditions shown in the main text. Each condition contains 100 poisoned and 900 clean training examples.

Table 4: Confusion counts for the Qwen3-VL-2B BadNets-MM detection results reported in the main text. Each condition contains 100 poisoned and 900 clean examples.

Dataset	Detector	TP	FP	TN	FN
COCO	BYE	100	1	899	0
COCO	TCAP	100	0	900	0
COCO	GradSentry	94	232	668	6
COCO	GradTrace	97	0	900	3
Flickr30k	BYE	100	3	897	0
Flickr30k	TCAP	100	0	900	0
Flickr30k	GradSentry	82	729	171	18
Flickr30k	GradTrace	100	0	900	0

B.2 Downstream Evaluation Protocol

Clean performance (CP) is measured by CIDEr on clean captioning examples. Attack success rate (ASR) is the percentage of triggered evaluation inputs whose generated response contains the attacker-specified target response. The evaluator lowercases both the generated response and the target and removes terminal periods from the target before performing the containment check. Target information is used only for evaluation and is not provided to the detector.

For downstream evaluation after filtering, a fresh LoRA adapter is trained from the original base model on the retained training data.

B.3 Detection-Time Comparison

Table 5 reports detection time for BYE, TCAP, GradSentry, and GradTrace across the six BadNets-MM model–dataset conditions. All measurements use 1,000-example suspect sets on a single NVIDIA H20D GPU and include model loading and detection while excluding downstream retraining.

Among the evaluated implementations, GradTrace records the lowest detection time in all six conditions. For example, on Qwen3-VL-8B/COCO, GradTrace requires 718.20 seconds, compared with 2100 seconds for BYE, 2400 seconds for TCAP, and 10832.92 seconds for the evaluated exact-SVD GradSentry implementation. Figure 5 visualizes the timing comparison.

Table 5: Detection time under BadNets-MM for the six model–dataset conditions. Each condition uses a 1,000-example suspect set. Times are reported in seconds and include model loading and detection but exclude downstream retraining. All measurements are conducted on an NVIDIA H20D GPU. Bold marks the lowest recorded time within each condition. GradSentry uses the evaluated exact-SVD implementation.

Model	Method	COCO time (s) ↓	Flickr30k time (s) ↓
Qwen3-VL-2B	BYE	1650	1550
	TCAP	1900	1800
	GradSentry	3851.79	1756.18
	GradTrace	**983.68**	**390.73**
Qwen3-VL-8B	BYE	2100	1950
	TCAP	2400	2200
	GradSentry	10832.92	591.89
	GradTrace	**718.20**	**208.40**
InternVL3.5-2B	BYE	1600	1500
	TCAP	1850	1750
	GradSentry	497.90	577.12
	GradTrace	**435.83**	**410.97**

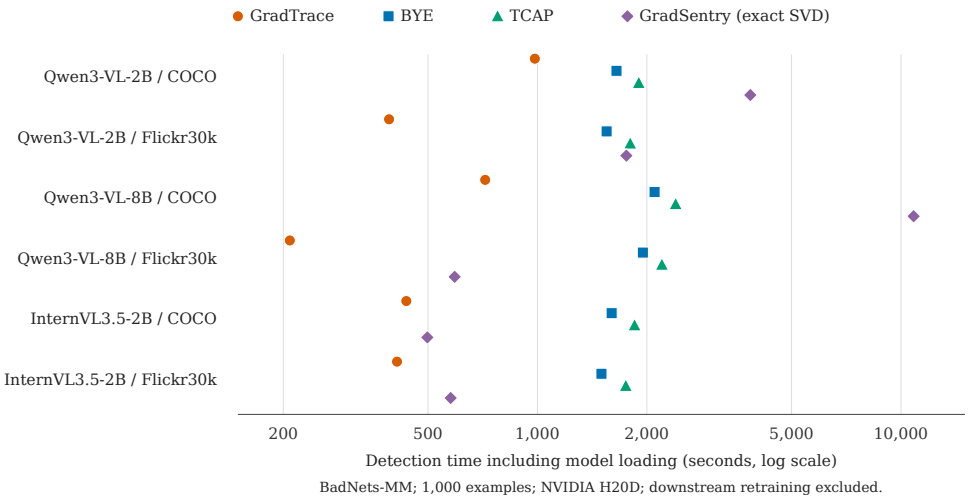

Figure 5: **Detection-time comparison.** Recorded detection time of the four evaluated implementations across the six BadNets-MM model–dataset conditions. The horizontal axis is logarithmic. Measurements include model loading and detection but exclude downstream retraining.

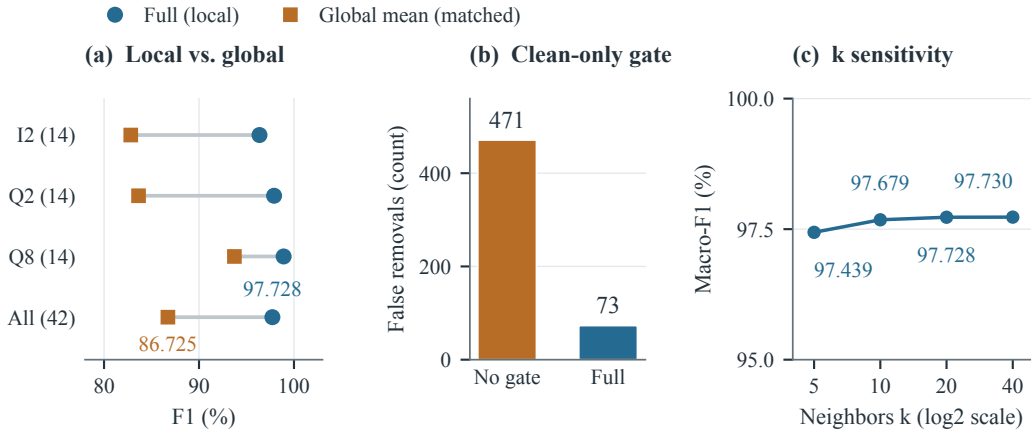

Figure 6: **Component analysis of GradTrace.** (a) Condition-mean F1 by model over 42 poisoned conditions at matched removal counts. (b) False removals across six clean-only conditions. (c) Macro-F1 sensitivity to neighborhood size. I2, Q2, and Q8 denote InternVL3.5-2B, Qwen3-VL-2B, and Qwen3-VL-8B, respectively.

C Ablation Study

We examine three design choices in GradTrace: local versus global gradient agreement, the separation gate, and neighborhood size.

C.1 Paired Evaluation Protocol

The analysis covers all $3 \times 2 \times 7 = 42$ poisoned conditions and six clean-only controls across InternVL3.5-2B, Qwen3-VL-2B, and Qwen3-VL-8B on COCO and Flickr30k. Each condition contains 1,000 examples and uses the same fixed matrix of normalized, 4,096-dimensional gradient sketches across variants.

Full uses the default $k = 20$ neighborhood score together with the separation gate. *No gate* uses the same candidate group but removes it regardless of separation. *Global mean* replaces local neighborhood agreement with mean cosine similarity to all other examples. We additionally evaluate $k \in \{5, 10, 40\}$. For Global mean and the alternate-k variants, we match Full's per-condition removal count to isolate ranking quality.

Precision, recall, F1, and removal are macro-averaged over the 42 poisoned conditions. Clean retention is computed over the corresponding clean sample–condition observations.

C.2 Component and Neighborhood Ablations

Table 6 reports the paired results. Replacing local neighborhood agreement with global averaging reduces macro-F1 from 97.728% to 86.725% at the same removal budget. Full achieves higher F1 than Global mean in 27 conditions, equal F1 in 14, and lower F1 in one.

Per-model results. Table 7 shows that Full consistently outperforms Global mean in condition-mean F1 across all three models. Across the 42 poisoned conditions, Global mean removes 518 clean observations and identifies 3,614 poisoned observations, compared with 60 and 4,072, respectively, for Full.

Neighborhood-size sensitivity. For $k = 5, 10, 20, 40$, macro-F1 is 97.439%, 97.679%, 97.728%, and 97.730%, respectively. The variation over $k \in \{10, 20, 40\}$ is only 0.0512 percentage points, showing limited sensitivity within this range.

Table 6: Paired ablations over 42 poisoned conditions. Precision (P), recall (R), F1, and removal are condition-level macro averages; clean retention is computed over clean sample–condition observations. "Matched" denotes evaluation using Full's per-condition removal count.

Variant	P	R	F1	Removed	Clean retained
Full ($k = 20$)	98.656	96.952	97.728	9.838	99.841
No gate	98.656	96.952	97.728	9.838	99.841
Global mean (matched)	87.529	86.048	86.725	9.838	98.630
$k = 5$ (matched)	98.363	96.667	97.439	9.838	99.810
$k = 10$ (matched)	98.606	96.905	97.679	9.838	99.836
$k = 40$ (matched)	98.661	96.952	97.730	9.838	99.841

Table 7: Condition-mean F1 (%) by model over the paired gradient caches. Parentheses indicate the number of poisoned conditions.

Variant	InternVL3.5-2B (14)	Qwen3-VL-2B (14)	Qwen3-VL-8B (14)
Full ($k = 20$)	96.369	97.905	98.910
No gate	96.369	97.905	98.910
Global mean (matched)	82.815	83.633	93.728
$k = 5$ (matched)	95.720	97.833	98.764
$k = 10$ (matched)	96.223	97.905	98.910
$k = 40$ (matched)	96.377	97.905	98.910

C.3 Clean-only Controls: The Role of the Gate

All 42 poisoned conditions pass the separation gate, so Full and No gate have identical detection results on these conditions. We therefore evaluate six clean-only controls to examine whether the gate prevents unnecessary filtering.

Table 8: Clean-only gate ablation. Each condition contains 1,000 clean examples. Entries report the number of falsely removed examples.

Model	Dataset	Full	No gate
Qwen3-VL-2B	COCO	0	75
Qwen3-VL-2B	Flickr30k	0	52
Qwen3-VL-8B	COCO	73	73
Qwen3-VL-8B	Flickr30k	0	140
InternVL3.5-2B	COCO	0	54
InternVL3.5-2B	Flickr30k	0	77
Total false removals		73/6,000	471/6,000
False-removal rate		1.217%	7.850%
Conditions with filtering		1/6	6/6

The gate reduces false removals from 471/6,000 to 73/6,000 (7.850% to 1.217%) and suppresses filtering in five of the six clean-only conditions. The remaining 73 false removals occur on Qwen3-VL-8B/COCO, showing that the gate substantially reduces, but does not eliminate, unnecessary removal.

D Extended Evaluation and Matched Controls

D.1 Scope and Evaluation Protocol

We extend the six focal BadNets-MM conditions in Table 1 with 33 additional downstream conditions covering the other six attacks. We also report a 39-condition selection comparison including BadNets-MM. Three Qwen3-VL-2B/COCO configurations (BadNets-I, SIG, and AddSent) are used for development and are excluded from these two evaluations. The fixed-cache component analysis in Appendix C additionally includes these configurations, yielding 42 conditions in total.

Each condition contains 1,000 suspect training examples (100 poisoned and 900 clean), together with 64 clean and 64 triggered evaluation examples. CP is measured by CIDEr, while ASR, removal, and selection metrics are reported as percentages. Detailed metric definitions follow Appendix B.

The downstream comparison includes Vanilla FT, BYE, TCAP, GradSentry, GradTrace, and a matched High-Loss reference that removes the same number of examples as GradTrace in each condition. Selection results compare the four detection methods.

D.2 Cross-Attack Outcomes and Matched Controls

Across the 33 additional conditions, mean ASR decreases from 95.12% with Vanilla FT to 17.19% with GradTrace; 20 conditions show no observed target hits. ASR decreases in 32 conditions and is unchanged in one. Performance varies across attacks: SIG has no observed target hits in all five available conditions, whereas BadNets-T retains a mean ASR of 54.17% (Table 9).

Table 9: Cross-attack downstream summary over 33 supplementary conditions. Attack success rates (ASR) are equal-weight condition means (%); n counts model–dataset–attack conditions, not seeds. Zero counts conditions with no observed target hits. Per-condition clean performance (CP), ASR, and removal are reported in Tables 12–14.

Attack	n	Vanilla FT ASR	GradTrace ASR	Zero
BadNets-I	5	84.69	10.00	4/5
Blended	6	98.18	19.79	3/6
SIG	5	86.56	0.00	5/5
BadNets-T	6	99.22	54.17	1/6
BadNets-MT	6	100.00	4.17	4/6
AddSent	5	99.69	9.69	3/5

At matched per-condition removal budgets, GradTrace achieves a mean ASR of 17.19%, compared with 92.05% for High-Loss (Table 10), and yields lower ASR in all 33 conditions.

Table 10: Matched high-loss comparison on the same 33 supplementary downstream conditions. ASR and removal are condition means (%). Budgets match per condition, not just in the aggregate.

Route	ASR	Removed
High-Loss (GT)	92.05	9.83
GradTrace	17.19	9.83

TCAP achieves a lower mean ASR than GradTrace (8.24% versus 17.19%) but removes more data on average (14.90% versus 9.83%). GradTrace improves CP over Vanilla FT in 20 conditions and decreases it in 13. Full per-condition results are reported in Tables 12–14.

D.3 Extended Selection Comparison

Across the 39-condition selection comparison, GradTrace achieves 97.729% macro-F1 and 99.826% clean retention (Table 11). Compared with TCAP, its F1 is higher in 10 conditions, equal in eight, and lower in 21, indicating that the aggregate result does not correspond to uniform per-condition dominance.

Table 11: Supplementary selection comparison over 39 model–dataset–attack conditions. Precision (P), recall (R), F1, and removal are equal-weight condition means (%), so F1 is macro-F1, not pooled F1. Clean retention equals its pooled value because every condition contains 900 clean examples. This evaluation is separate from the 42-condition fixed-cache component study in Appendix C.

Method	P	R	F1	Removed	Clean retained
BYE	64.096	70.795	65.840	12.885	93.550
TCAP	87.906	94.718	89.692	14.756	94.128
GradSentry	3.436	16.128	5.452	49.123	47.211
GradTrace	98.526	97.077	97.729	9.864	99.826

D.4 PER-CONDITION DOWNSTREAM RESULTS

Tables 12–14 report the 33 additional downstream conditions. A dash denotes a development configuration excluded from this evaluation.

Table 12: Downstream results for Qwen3-VL-2B. CP is native-scale CIDEr; ASR and Rem. (removal) are percentages. High-Loss (GT) matches the per-condition GradTrace removal count. Dashes mark development configurations not included in this evaluation. Numerical entries are unranked.

Attack	Method	COCO			Flickr30k		
		CP	ASR	Rem.	CP	ASR	Rem.
BadNets-I	Vanilla FT	–	–	–	0.599	100.00	0.00
	BYE	–	–	–	0.668	0.00	10.30
	TCAP	–	–	–	0.680	0.00	10.60
	GradSentry	–	–	–	0.003	100.00	92.10
	GradTrace	–	–	–	0.486	50.00	8.10
	High-Loss (GT)	–	–	–	0.705	100.00	8.10
Blended	Vanilla FT	1.128	98.44	0.00	0.674	100.00	0.00
	BYE	1.329	0.00	9.80	0.749	0.00	10.10
	TCAP	1.289	0.00	9.90	0.719	0.00	9.90
	GradSentry	1.195	100.00	23.20	0.792	100.00	76.50
	GradTrace	1.319	0.00	9.90	0.629	0.00	10.00
	High-Loss (GT)	1.155	96.88	9.90	0.650	100.00	10.00
SIG	Vanilla FT	–	–	–	0.763	100.00	0.00
	BYE	–	–	–	0.698	0.00	10.10
	TCAP	–	–	–	0.650	0.00	10.00
	GradSentry	–	–	–	0.003	100.00	90.20
	GradTrace	–	–	–	0.727	0.00	9.90
	High-Loss (GT)	–	–	–	0.719	100.00	9.90
BadNets-T	Vanilla FT	1.219	98.44	0.00	0.651	100.00	0.00
	BYE	1.157	100.00	0.00	0.604	100.00	0.00
	TCAP	1.244	0.00	10.00	0.569	0.00	10.00
	GradSentry	1.314	67.19	30.80	0.737	100.00	78.90
	GradTrace	1.259	98.44	8.70	0.710	10.94	9.80
	High-Loss (GT)	1.161	100.00	8.70	0.709	100.00	9.80
BadNets-MT	Vanilla FT	1.200	100.00	0.00	0.680	100.00	0.00
	BYE	1.189	0.00	10.10	0.695	100.00	0.00
	TCAP	1.205	0.00	10.00	0.655	0.00	10.00
	GradSentry	1.271	100.00	25.50	0.803	100.00	78.90
	GradTrace	1.190	0.00	9.70	0.759	0.00	9.90
	High-Loss (GT)	1.259	100.00	9.70	0.709	100.00	9.90
AddSent	Vanilla FT	–	–	–	0.675	100.00	0.00
	BYE	–	–	–	0.774	100.00	9.90
	TCAP	–	–	–	0.588	0.00	10.00
	GradSentry	–	–	–	0.776	100.00	82.90
	GradTrace	–	–	–	0.569	0.00	10.00
	High-Loss (GT)	–	–	–	0.578	100.00	10.00

Table 13: Downstream results for Qwen3-VL-8B. CP is native-scale CIDEr; ASR and Rem. (removal) are percentages. High-Loss (GT) matches the per-condition GradTrace removal count. Dashes mark development configurations not included in this evaluation. Numerical entries are unranked.

Attack	Method	COCO			Flickr30k		
		CP	ASR	Rem.	CP	ASR	Rem.
BadNets-I	Vanilla FT	1.136	100.00	0.00	0.719	100.00	0.00
	BYE	1.249	0.00	10.40	0.735	0.00	10.10
	TCAP	1.213	0.00	10.00	0.644	0.00	10.00
	GradSentry	1.263	100.00	7.70	0.698	82.81	89.10
	GradTrace	1.203	0.00	10.00	0.636	0.00	10.00
	High-Loss (GT)	1.108	100.00	10.00	0.867	96.88	10.00
Blended	Vanilla FT	1.125	100.00	0.00	0.666	98.44	0.00
	BYE	1.263	0.00	10.10	0.570	0.00	10.20
	TCAP	1.173	0.00	10.00	0.721	0.00	10.00
	GradSentry	1.271	98.44	6.70	0.876	98.44	57.40
	GradTrace	1.207	23.44	9.60	0.836	28.12	9.60
	High-Loss (GT)	1.123	98.44	9.60	0.560	96.88	9.60
SIG	Vanilla FT	1.108	93.75	0.00	0.638	100.00	0.00
	BYE	1.154	0.00	9.90	0.700	0.00	10.30
	TCAP	1.308	98.44	70.50	0.687	0.00	13.30
	GradSentry	0.015	100.00	89.90	0.003	100.00	89.70
	GradTrace	1.060	0.00	9.80	0.693	0.00	10.00
	High-Loss (GT)	1.277	100.00	9.80	0.730	100.00	10.00
BadNets-T	Vanilla FT	1.144	100.00	0.00	0.655	100.00	0.00
	BYE	1.260	0.00	10.10	0.756	100.00	5.90
	TCAP	1.260	0.00	10.00	0.755	100.00	42.90
	GradSentry	1.070	100.00	11.80	0.912	100.00	58.70
	GradTrace	1.290	78.12	9.50	0.678	71.88	9.70
	High-Loss (GT)	1.124	100.00	9.50	0.681	100.00	9.70
BadNets-MT	Vanilla FT	1.116	100.00	0.00	0.676	100.00	0.00
	BYE	1.052	100.00	85.30	0.631	100.00	0.00
	TCAP	1.107	0.00	10.00	0.835	0.00	9.80
	GradSentry	1.211	100.00	7.40	0.741	100.00	87.80
	GradTrace	1.257	9.38	9.70	0.796	0.00	9.90
	High-Loss (GT)	1.158	98.44	9.70	0.750	100.00	9.90
AddSent	Vanilla FT	1.145	98.44	0.00	0.645	100.00	0.00
	BYE	1.288	0.00	10.40	0.648	0.00	10.10
	TCAP	1.190	0.00	10.00	0.647	0.00	10.00
	GradSentry	1.292	100.00	7.50	0.893	100.00	63.80
	GradTrace	1.169	0.00	9.80	0.782	21.88	9.80
	High-Loss (GT)	1.334	100.00	9.80	0.767	100.00	9.80

Table 14: Downstream results for InternVL3.5-2B. CP is native-scale CIDEr; ASR and Rem. (removal) are percentages. High-Loss (GT) matches the per-condition GradTrace removal count. Dashes mark development configurations not included in this evaluation. Numerical entries are unranked.

Attack	Method	COCO			Flickr30k		
		CP	ASR	Rem.	CP	ASR	Rem.
BadNets-I	Vanilla FT	0.377	28.12	0.00	0.687	95.31	0.00
	BYE	1.013	65.62	83.90	0.524	0.00	16.00
	TCAP	1.148	0.00	23.80	0.674	0.00	10.10
	GradSentry	0.845	35.94	10.00	0.640	100.00	70.20
	GradTrace	0.091	0.00	9.90	0.853	0.00	10.30
	High-Loss (GT)	0.625	25.00	9.90	0.773	87.50	10.30
Blended	Vanilla FT	0.343	93.75	0.00	0.742	98.44	0.00
	BYE	0.850	0.00	13.20	0.524	0.00	16.00
	TCAP	0.017	0.00	10.00	0.633	0.00	9.90
	GradSentry	0.991	23.44	9.90	0.692	100.00	69.30
	GradTrace	1.067	67.19	8.90	0.674	0.00	9.80
	High-Loss (GT)	1.233	95.31	8.90	0.877	98.44	9.80
SIG	Vanilla FT	0.060	59.38	0.00	0.620	79.69	0.00
	BYE	0.850	0.00	13.20	0.524	0.00	16.00
	TCAP	0.783	73.44	56.80	0.778	0.00	14.50
	GradSentry	0.539	42.19	11.00	0.003	100.00	90.00
	GradTrace	0.681	0.00	10.90	0.598	0.00	11.10
	High-Loss (GT)	0.666	14.06	10.90	0.635	68.75	11.10
BadNets-T	Vanilla FT	0.214	96.88	0.00	0.740	100.00	0.00
	BYE	0.984	98.44	3.40	0.595	100.00	6.60
	TCAP	0.017	0.00	10.00	0.596	0.00	10.00
	GradSentry	0.225	100.00	9.90	0.655	100.00	70.80
	GradTrace	1.047	65.62	10.40	0.596	0.00	10.00
	High-Loss (GT)	0.286	100.00	10.40	0.762	100.00	10.00
BadNets-MT	Vanilla FT	0.011	100.00	0.00	0.643	100.00	0.00
	BYE	0.160	100.00	3.50	0.610	100.00	6.60
	TCAP	0.017	0.00	10.00	0.644	0.00	9.80
	GradSentry	0.099	100.00	9.10	0.003	100.00	68.60
	GradTrace	0.696	15.62	9.80	0.642	0.00	9.80
	High-Loss (GT)	0.056	60.94	9.80	0.532	100.00	9.80
AddSent	Vanilla FT	0.237	100.00	0.00	0.667	100.00	0.00
	BYE	1.017	100.00	3.40	0.678	100.00	6.60
	TCAP	0.017	0.00	10.00	0.596	0.00	10.00
	GradSentry	0.986	100.00	9.60	0.820	48.44	70.00
	GradTrace	0.956	26.56	10.00	0.596	0.00	10.00
	High-Loss (GT)	0.838	100.00	10.00	0.712	100.00	10.00

D.5 Per-Condition Selection Results

Tables 15–17 report the 39-condition selection comparison. Clean retention denotes the fraction of clean training examples retained after filtering.

Table 15: Selection results for Qwen3-VL-2B. All metrics are percentages; P/R/F1 denote precision, recall, and F1; Rem. is removal and Clean is clean retention. Dashes mark development configurations not included in this evaluation. Numerical entries are unranked.

Attack	Method	COCO					Flickr30k				
		P	R	F1	Rem.	Clean	P	R	F1	Rem.	Clean
BadNets-I	BYE	–	–	–	–	–	97.09	100.00	98.52	10.30	99.67
	TCAP	–	–	–	–	–	94.34	100.00	97.09	10.60	99.33
	GradSentry	–	–	–	–	–	2.50	23.00	4.51	92.10	0.22
	GradTrace	–	–	–	–	–	100.00	81.00	89.50	8.10	100.00
Blended	BYE	100.00	98.00	98.99	9.80	100.00	99.01	100.00	99.50	10.10	99.89
	TCAP	100.00	99.00	99.50	9.90	100.00	100.00	99.00	99.50	9.90	100.00
	GradSentry	13.79	32.00	19.28	23.20	77.78	4.31	33.00	7.63	76.50	18.67
	GradTrace	100.00	99.00	99.50	9.90	100.00	100.00	100.00	100.00	10.00	100.00
SIG	BYE	–	–	–	–	–	99.01	100.00	99.50	10.10	99.89
	TCAP	–	–	–	–	–	99.00	99.00	99.00	10.00	99.89
	GradSentry	–	–	–	–	–	0.22	2.00	0.40	90.20	0.00
	GradTrace	–	–	–	–	–	100.00	99.00	99.50	9.90	100.00
BadNets-T	BYE	0.00	0.00	0.00	0.00	100.00	0.00	0.00	0.00	0.00	100.00
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	30.84	95.00	46.57	30.80	76.33	8.87	70.00	15.75	78.90	20.11
	GradTrace	100.00	87.00	93.05	8.70	100.00	100.00	98.00	98.99	9.80	100.00
BadNets-MT	BYE	99.01	100.00	99.50	10.10	99.89	0.00	0.00	0.00	0.00	100.00
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	14.90	38.00	21.41	25.50	75.89	7.48	59.00	13.27	78.90	18.89
	GradTrace	100.00	97.00	98.48	9.70	100.00	100.00	99.00	99.50	9.90	100.00
AddSent	BYE	–	–	–	–	–	97.98	97.00	97.49	9.90	99.78
	TCAP	–	–	–	–	–	100.00	100.00	100.00	10.00	100.00
	GradSentry	–	–	–	–	–	11.70	97.00	20.88	82.90	18.67
	GradTrace	–	–	–	–	–	100.00	100.00	100.00	10.00	100.00
BadNets-MM	BYE	99.01	100.00	99.50	10.10	99.89	97.09	100.00	98.52	10.30	99.67
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	28.83	94.00	44.13	32.60	74.22	10.11	82.00	18.00	81.10	19.00
	GradTrace	100.00	97.00	98.48	9.70	100.00	100.00	100.00	100.00	10.00	100.00

Table 16: Selection results for Qwen3-VL-8B. All metrics are percentages; P/R/F1 denote precision, recall, and F1; Rem. is removal and Clean is clean retention. Dashes mark development configurations not included in this evaluation. Numerical entries are unranked.

Attack	Method	COCO					Flickr30k				
		P	R	F1	Rem.	Clean	P	R	F1	Rem.	Clean
BadNets-I	BYE	96.15	100.00	98.04	10.40	99.56	99.01	100.00	99.50	10.10	99.89
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	0.00	0.00	0.00	7.70	91.44	0.00	0.00	0.00	89.10	1.00
	GradTrace	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
Blended	BYE	99.01	100.00	99.50	10.10	99.89	98.04	100.00	99.01	10.20	99.78
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	0.00	0.00	0.00	6.70	92.56	0.00	0.00	0.00	57.40	36.22
	GradTrace	100.00	96.00	97.96	9.60	100.00	100.00	96.00	97.96	9.60	100.00
SIG	BYE	100.00	99.00	99.50	9.90	100.00	97.09	100.00	98.52	10.30	99.67
	TCAP	0.00	0.00	0.00	70.50	21.67	75.19	100.00	85.84	13.30	96.33
	GradSentry	0.11	1.00	0.20	89.90	0.22	0.00	0.00	0.00	89.70	0.33
	GradTrace	100.00	98.00	98.99	9.80	100.00	100.00	100.00	100.00	10.00	100.00
BadNets-T	BYE	99.01	100.00	99.50	10.10	99.89	100.00	59.00	74.21	5.90	100.00
	TCAP	100.00	100.00	100.00	10.00	100.00	8.86	38.00	14.37	42.90	56.56
	GradSentry	0.00	0.00	0.00	11.80	86.89	0.00	0.00	0.00	58.70	34.78
	GradTrace	100.00	95.00	97.44	9.50	100.00	100.00	97.00	98.48	9.70	100.00
BadNets-MT	BYE	9.85	84.00	17.63	85.30	14.56	0.00	0.00	0.00	0.00	100.00
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	98.00	98.99	9.80	100.00
	GradSentry	0.00	0.00	0.00	7.40	91.78	0.34	3.00	0.61	87.80	2.78
	GradTrace	98.97	96.00	97.46	9.70	99.89	100.00	99.00	99.50	9.90	100.00
AddSent	BYE	95.19	99.00	97.06	10.40	99.44	99.01	100.00	99.50	10.10	99.89
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	0.00	0.00	0.00	7.50	91.67	0.00	0.00	0.00	63.80	29.11
	GradTrace	100.00	98.00	98.99	9.80	100.00	100.00	98.00	98.99	9.80	100.00
BadNets-MM	BYE	100.00	100.00	100.00	10.00	100.00	99.01	100.00	99.50	10.10	99.89
	TCAP	100.00	100.00	100.00	10.00	100.00	29.67	100.00	45.77	33.70	73.67
	GradSentry	0.00	0.00	0.00	9.00	90.00	0.00	0.00	0.00	58.20	35.33
	GradTrace	100.00	98.00	98.99	9.80	100.00	100.00	99.00	99.50	9.90	100.00

Table 17: Selection results for InternVL3.5-2B. All metrics are percentages; P/R/F1 denote precision, recall, and F1; Rem. is removal and Clean is clean retention. Dashes mark development configurations not included in this evaluation. Numerical entries are unranked.

Attack	Method	COCO					Flickr30k				
		P	R	F1	Rem.	Clean	P	R	F1	Rem.	Clean
BadNets-I	BYE	0.00	0.00	0.00	83.90	6.78	62.50	100.00	76.92	16.00	93.33
	TCAP	42.02	100.00	59.17	23.80	84.67	99.01	100.00	99.50	10.10	99.89
	GradSentry	0.00	0.00	0.00	10.00	88.89	0.00	0.00	0.00	70.20	22.00
	GradTrace	100.00	99.00	99.50	9.90	100.00	93.20	96.00	94.58	10.30	99.22
Blended	BYE	75.76	100.00	86.21	13.20	96.44	62.50	100.00	76.92	16.00	93.33
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	99.00	99.50	9.90	100.00
	GradSentry	0.00	0.00	0.00	9.90	89.00	0.00	0.00	0.00	69.30	23.00
	GradTrace	98.88	88.00	93.12	8.90	99.89	100.00	98.00	98.99	9.80	100.00
SIG	BYE	75.76	100.00	86.21	13.20	96.44	62.50	100.00	76.92	16.00	93.33
	TCAP	11.27	64.00	19.16	56.80	44.00	68.97	100.00	81.63	14.50	95.00
	GradSentry	0.00	0.00	0.00	11.00	87.78	0.00	0.00	0.00	90.00	0.00
	GradTrace	91.74	100.00	95.69	10.90	99.00	89.19	99.00	93.84	11.10	98.67
BadNets-T	BYE	5.88	2.00	2.99	3.40	96.44	9.09	6.00	7.23	6.60	93.33
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	0.00	0.00	0.00	9.90	89.00	0.00	0.00	0.00	70.80	21.33
	GradTrace	90.38	94.00	92.16	10.40	98.89	100.00	100.00	100.00	10.00	100.00
BadNets-MT	BYE	8.57	3.00	4.44	3.50	96.44	9.09	6.00	7.23	6.60	93.33
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	98.00	98.99	9.80	100.00
	GradSentry	0.00	0.00	0.00	9.10	89.89	0.00	0.00	0.00	68.60	23.78
	GradTrace	93.88	92.00	92.93	9.80	99.33	100.00	98.00	98.99	9.80	100.00
AddSent	BYE	5.88	2.00	2.99	3.40	96.44	9.09	6.00	7.23	6.60	93.33
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	0.00	0.00	0.00	9.60	89.33	0.00	0.00	0.00	70.00	22.22
	GradTrace	98.00	98.00	98.00	10.00	99.78	100.00	100.00	100.00	10.00	100.00
BadNets-MM	BYE	75.76	100.00	86.21	13.20	96.44	57.80	100.00	73.26	17.30	91.89
	TCAP	100.00	100.00	100.00	10.00	100.00	100.00	100.00	100.00	10.00	100.00
	GradSentry	0.00	0.00	0.00	10.00	88.89	0.00	0.00	0.00	70.00	22.22
	GradTrace	88.29	98.00	92.89	11.10	98.56	100.00	99.00	99.50	9.90	100.00