SELECTIVE CHANNEL RESTORATION FOR BACKDOORED VISION-LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Vision-language models (VLMs) exhibit strong multimodal capabilities but remain vulnerable to backdoors implanted through poisoned fine-tuning data. Existing defenses often require extensive parameter updates during fine-tuning or incur per-query overhead during inference. To address these limitations, we propose Perturb-Select-Restore (PSR), a post-training defense that performs sparse updates to the projection interface and introduces no additional computation during inference. We reveal that backdoored VLM projectors are substantially more sensitive to bounded perturbations than clean VLM projectors, a phenomenon we term *projection fragility*. Building on this finding, PSR identifies the output channels most sensitive to perturbations in each projection layer of a backdoored VLM and restores their parameters to the corresponding pretrained values. Experiments across multiple tasks show that PSR reduces attack success rates to near zero while preserving clean-task performance.

1 INTRODUCTION

Vision-language models (VLMs) connect a visual encoder to a large language model through a lightweight visual-to-language projection interface (Dai et al., 2023; Liu et al., 2024; Zhu et al., 2024; Bai et al., 2025). This modular design enables efficient adaptation to a broad range of multimodal tasks by fine-tuning only the projection interface while keeping the visual encoder and language model frozen. However, fine-tuning on untrusted data exposes VLMs to backdoor attacks: an adversary can inject a small number of poisoned samples, causing the adapted VLM to behave normally on clean inputs yet produce attacker-specified responses whenever a visual trigger appears at inference time (Chen et al., 2017; Gu et al., 2019). Recent work has demonstrated the vulnerability of VLMs to backdoors involving diverse trigger types (Lyu et al., 2024; 2025).

Existing VLM backdoor defenses mainly operate during the fine-tuning or inference phases (Rong et al., 2025; Xun et al., 2025; Jiang et al., 2026; Xu et al., 2026; Zhang et al., 2026), thereby requiring either extensive parameter updates or additional per-query computation. General post-training backdoor defenses (Liu et al., 2018; Wang et al., 2019; Wu & Wang, 2021; Zheng et al., 2022; Li et al., 2023; Lin et al., 2024) can enable efficient model repair, but were primarily developed for image classifiers rather than VLMs. Applying them directly to VLM projection layers can be inefficient or harmful to clean utility when performance depends on preserving vision-language alignment.

To explore whether post-training purification is feasible for VLM projection interfaces, we begin by testing whether backdoored projection interfaces exhibit distinctive sensitivity to perturbations. We conduct controlled probes on clean and backdoored LLaVA-1.5 projectors across COCO and VQAv2. For each sample, we apply temporary multiplicative perturbations to the projector parameters and optimize them via projected sign-gradient ascent to maximize the task loss under teacher forcing. The model parameters remain fixed during this procedure, and the optimized perturbations are discarded after each sample. For each projector, we compute the perturbation-induced loss increase, $\Delta L = L_{\text{adv}} - L_{\text{base}}$, where L_{base} is that projector's own unperturbed loss. We compare these loss increases across projectors at each fixed perturbation budget ϵ. Further implementation details are provided in Appendix B.1. Figure 1 shows the resulting loss increases for clean projectors and projectors backdoored by BadNet, Blended, or TrojVLM. In both tasks, all three backdoored projectors become more sensitive than the clean projector as the perturbation budget increases. We refer to this phenomenon as *projection fragility*.

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.

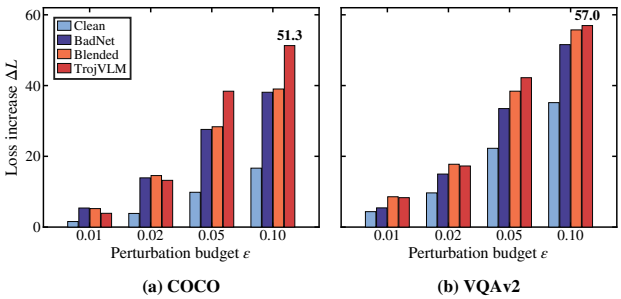

Figure 1: Loss increase ΔL under bounded projector perturbations on clean samples. (a) COCO; (b) VQAv2. Higher ΔL indicates greater projection fragility.

Although this observation provides a useful signal for purification, the diverse and multi-layered structure of VLM projection interfaces makes it challenging to design an efficient and broadly applicable method. To address this challenge, we introduce *Perturb-Select-Restore* (PSR), a post-training purification method. Given a backdoored VLM, PSR identifies and repairs suspicious output channels in the target projection layers. It first perturbs these channels within a bounded range and learns robustness scores. It then selects a fixed fraction of the lowest-scoring channels separately in each layer. Finally, it restores the selected channels using the corresponding pretrained weights and biases. This layer-wise channel repair reduces computation, handles different sensitivity distributions across layers, and better preserves clean vision-language performance than direct pruning.

Our contributions are summarized as follows:

- **A clean-data signal for backdoor purification.** We identify projection fragility: backdoored VLMs exhibit larger clean-loss increases under projection perturbations than clean VLMs, providing a signal for channel selection.
- **A channel-wise post-training purification method.** PSR learns channel-wise robustness scores to identify backdoor-associated projection output channels and restores their parameters to the corresponding pretrained values.
- **Strong empirical results across diverse settings.** Experiments across multiple tasks show that PSR reduces attack success rates to near zero while preserving clean-task performance.

2 Related Work

Backdoor attacks on vision-language models. Backdoor attacks were first studied extensively in image classification, where an attacker poisons training data so that a model predicts an attacker-chosen target when a trigger is present (Chen et al., 2017; Gu et al., 2019). In vision-language settings, triggers can be implemented through different visual transformations, ranging from explicit patch or blended patterns (Chen et al., 2017; Gu et al., 2019) to spatially warped or sample-specific invisible perturbations (Li et al., 2021; Nguyen & Tran, 2021). These attacks transfer effectively to VLMs (Liang et al., 2025). More recent work has introduced attacks specifically designed for them: TrojVLM injects backdoors during instruction tuning while preserving caption quality on clean images (Lyu et al., 2024), and VLOOD demonstrates that backdoors can be implanted using only out-of-distribution auxiliary data (Lyu et al., 2025). Domain-shift studies further indicate that VLM backdoors can remain effective across mismatched training and testing domains (Liang et al., 2025). These results motivate defenses that suppress backdoor behavior within the model rather than only filtering obvious input triggers.

Backdoor defenses for vision-language models. VLM-specific defenses can be broadly categorized into training-time and test-time approaches. Training-time methods either identify and remove poisoned samples before adaptation or regularize the adapted parameters to suppress backdoor triggers. For example, BYE detects suspicious samples in VLM fine-tuning using attention-based signals and clustering (Rong et al., 2025), while RobustIT regularizes adapter fine-tuning to reduce trigger effects (Xun et al., 2025). Test-time methods, including PurMM, SRD, and CleanSight, intervene in the input or model computation during inference (Jiang et al., 2026; Xu et al., 2026; Zhang et al., 2026). Training-time methods require intervention during adaptation, whereas test-time methods introduce per-query overhead. In contrast, PSR purifies VLM projection weights after training without modifying the adaptation process or adding inference-time computation.

General post-training backdoor defenses. Post-training backdoor defenses based on neurons, channels, representations, or weight dynamics provide important technical foundations for PSR. Pruning methods remove backdoor-related components based on various criteria: Fine-Pruning removes neurons dormant on clean data (Liu et al., 2018), Neural Cleanse detects anomalous triggers and applies pruning (Wang et al., 2019), and ANP prunes neurons sensitive to adversarial perturbations on clean data (Wu & Wang, 2021). Channel-level methods such as CLP apply Lipschitzness-based criteria (Zheng et al., 2022). Other methods use unlearning-based signals to locate backdoor-related components, including RNP and TSBD (Li et al., 2023; Lin et al., 2024). Spectral Signatures detects poisoned samples via anomalous representation directions (Tran et al., 2018). However, because these defenses were not specifically designed for VLMs, their direct application may yield suboptimal performance; in contrast, PSR is tailored to the distinctive architectural characteristics of VLMs, making it better suited for backdoor mitigation in this setting.

3 PRELIMINARIES

3.1 THREAT MODEL

Victim model. We consider a VLM as a composition of a visual encoder, a visual-to-language projection interface, and a language model. Given an image $\mathbf{x}$ and a text query or instruction $\mathbf{q}$, the model output is written abstractly as

$$\mathbf{o} = M_\psi(P_\phi(E_{\text{vis}}(\mathbf{x})), \mathbf{q}), \tag{1}$$

where E_{vis} is the visual encoder, P_ϕ is the visual-to-language projection interface with parameters ϕ, and M_ψ is the language model. During downstream adaptation, only the projection interface is fine-tuned, while the visual encoder and language model remain fixed.

Adversary's objective. The adversary aims to implant a backdoor by poisoning the data used to fine-tune the projection interface. We assume that the adversary can control the fine-tuning data and optimization procedure but can modify only the projection parameters. Let $\mathcal{D} = \mathcal{D}_{\text{clean}} \cup \mathcal{D}_{\text{poison}}$ denote the fine-tuning set, where $\mathcal{D}_{\text{clean}}$ contains clean image-query-response triples and $\mathcal{D}_{\text{poison}}$ contains samples with a visual trigger $\boldsymbol{\tau}$ and an attacker-specified target response $\mathbf{o}^\star$. After fine-tuning on $\mathcal{D}$, the resulting parameters ϕ_b should preserve normal behavior on a clean image $\mathbf{x}$ while producing the target response for its triggered version $\mathbf{x} \oplus \boldsymbol{\tau}$:

$$f_{\phi_b}(\mathbf{x}, \mathbf{q}) = \mathbf{o}, \qquad f_{\phi_b}(\mathbf{x} \oplus \boldsymbol{\tau}, \mathbf{q}) = \mathbf{o}^\star. \tag{2}$$

Defender's setting. The defender receives a fine-tuned VLM suspected of containing a backdoor and aims to purify it. We assume access to the corresponding pretrained base VLM and its projection parameters ϕ_0, a small clean calibration set $\mathcal{C}$, and a restoration-ratio hyperparameter r. The defender need not have the original fine-tuning set or the complete adaptation recipe. The goal is to construct $\phi_{\text{pur}} = \mathcal{A}(\phi_b, \phi_0, \mathcal{C}; r)$ that reduces backdoor behavior while retaining clean-task utility.

3.2 PROJECTION INTERFACE AND CHANNEL REPRESENTATION

PSR targets selected linear layers inside the visual-to-language projection interface P_ϕ. Let $\mathcal{L} = \{\ell_1, \ell_2, \ldots, \ell_m\}$ denote these target layers. For layer ℓ, its affine output is

$$\mathbf{u}^{(\ell)} = \mathbf{W}^{(\ell)}\mathbf{v}^{(\ell)} + \mathbf{b}^{(\ell)}, \tag{3}$$

where $\mathbf{W}^{(\ell)} \in \mathbb{R}^{d_{\text{out}}^{(\ell)} \times d_{\text{in}}^{(\ell)}}$ and $\mathbf{b}^{(\ell)} \in \mathbb{R}^{d_{\text{out}}^{(\ell)}}$. Each output channel is indexed by $i \in \{1, \ldots, d_{\text{out}}^{(\ell)}\}$ and corresponds to the activation $u_i^{(\ell)}$, weight row $\mathbf{W}_{i,:}^{(\ell)}$, and bias entry $b_i^{(\ell)}$. This row-wise correspondence identifies each output channel and its associated parameters for the subsequent method.

4 METHODOLOGY

Overview. PSR is a three-stage post-training purification procedure, illustrated in Figure 2. Starting from a backdoored projector ϕ_b, it introduces a channel-wise robustness-score vector $\boldsymbol{\theta}$ and bounded auxiliary perturbations $\boldsymbol{\delta}$ at each target projection layer. Using only clean calibration data,

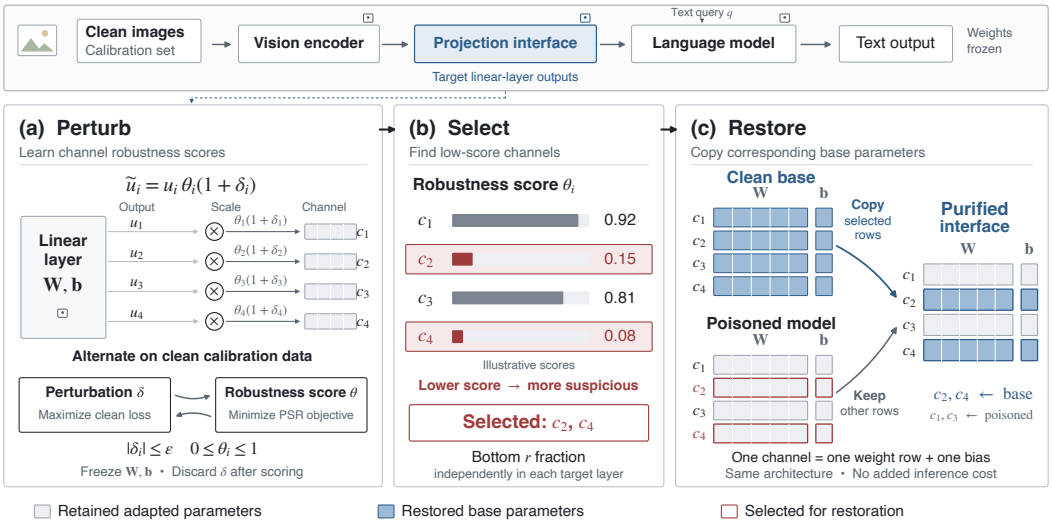

Figure 2: Overview of Perturb-Select-Restore (PSR), a post-training purification framework for VLM projection interfaces.

PSR alternates between maximizing the clean loss over $\boldsymbol{\delta}$ to expose sensitivity and minimizing a regularized clean-loss objective over $\boldsymbol{\theta}$, while keeping the model parameters frozen. Channels that are more vulnerable under this perturbation probe exert stronger pressure for their robustness scores to decrease, so the smallest learned $\boldsymbol{\theta}$ values identify the channels most likely to contain backdoor-related changes. PSR selects a fixed fraction of these channels independently in each layer, restores their weight rows and bias entries from ϕ_0, and leaves all other adapted parameters unchanged. This perturb-select-restore procedure concentrates the repair on fragile channels while preserving task-relevant adaptation elsewhere in the projection interface.

4.1 Channel-wise Parameterization

For each target layer ℓ, PSR applies the following transformation to the affine output $\mathbf{u}^{(\ell)}$:

$$\widetilde{\mathbf{u}}^{(\ell)} = \mathbf{u}^{(\ell)} \odot \left(\boldsymbol{\theta}^{(\ell)} \odot \left(\mathbf{1} + \boldsymbol{\delta}^{(\ell)}\right)\right), \tag{4}$$

Here, $\boldsymbol{\theta}^{(\ell)} \in [0,1]^{d_{\text{out}}^{(\ell)}}$ is a learnable channel robustness score vector, and $\boldsymbol{\delta}^{(\ell)} \in [-\epsilon, \epsilon]^{d_{\text{out}}^{(\ell)}}$ is a bounded auxiliary perturbation vector. The operator $\odot$ denotes element-wise multiplication with broadcasting along the non-channel dimensions. In the parameterization, $\boldsymbol{\theta}^{(\ell)}$ acts as a channel-wise retention coefficient: values close to 1 preserve the corresponding channels, whereas values close to 0 suppress them.

The model parameters remain frozen, and only these channel-wise variables are optimized. The perturbation vector $\boldsymbol{\delta}^{(\ell)}$ provides bounded adversarial changes to the channel outputs and is used only to learn $\boldsymbol{\theta}^{(\ell)}$. After optimization, the learned $\boldsymbol{\theta}^{(\ell)}$ values are used to select channels for restoration, whereas $\boldsymbol{\delta}^{(\ell)}$ is discarded. For notational simplicity, we use $\boldsymbol{\theta}$ and $\boldsymbol{\delta}$ to denote the collections of these variables across all target layers.

4.2 Robustness Score Optimization

Building on the channel-wise parameterization in Equation 4, PSR uses the small clean calibration set $\mathcal{C}$ as its only data input to learn the channel robustness scores. Let $\mathcal{L}_{\text{clean}}(\boldsymbol{\theta}, \boldsymbol{\delta}; \mathcal{C})$ denote the model's clean language-modeling loss under the channel variables. The optimization alternates between an inner maximization over the bounded perturbations $\boldsymbol{\delta}$ and an outer minimization over the robustness-score vectors $\boldsymbol{\theta}$.

Algorithm 1 PSR optimization and restoration

Input: Poisoned projection parameters ϕ_b, projection parameters ϕ_0 of the corresponding pretrained base VLM, clean calibration set $\mathcal{C}$, target layers $\mathcal{L}$, restoration ratio r, perturbation bound ϵ, optimization rounds T_{opt}, inner steps T_{PGD}, robustness-score learning rate η_θ, loss weights λ_{clean} and λ_θ

Output: Purified projection parameters ϕ_{pur}

Introduce channel-wise variables at the output of each target linear layer in $\mathcal{L}$
Initialize each robustness-score vector $\boldsymbol{\theta}^{(\ell)} \leftarrow \mathbf{1}_{d_{\text{out}}^{(\ell)}}$ and keep all model parameters frozen
for $s = 1, \ldots, T_{\text{opt}}$ **do**
  Initialize bounded channel perturbations $\boldsymbol{\delta}^{(\ell)}$ for all $\ell \in \mathcal{L}$
  for $t = 1, \ldots, T_{\text{PGD}}$ **do**
    $\boldsymbol{\delta} \leftarrow \Pi_{[-\epsilon,\epsilon]}\Big(\boldsymbol{\delta} + \frac{2\epsilon}{T_{\text{PGD}}} \operatorname{sign}(\nabla_{\boldsymbol{\delta}} \mathcal{L}_{\text{clean}}(\boldsymbol{\theta}, \boldsymbol{\delta}; \mathcal{C}))\Big)$
  end for
  $\mathbf{g}_\theta \leftarrow \nabla_{\boldsymbol{\theta}} \Big[\mathcal{L}_{\text{clean}}(\boldsymbol{\theta}, \boldsymbol{\delta}; \mathcal{C}) + \lambda_{\text{clean}} \mathcal{L}_{\text{clean}}(\boldsymbol{\theta}, \mathbf{0}; \mathcal{C}) + \lambda_\theta \sum_{\ell \in \mathcal{L}} \left\| \boldsymbol{\theta}^{(\ell)} \right\|_1 \Big]$
  $\boldsymbol{\theta} \leftarrow \Pi_{[0,1]}\Big(\boldsymbol{\theta} - \eta_\theta \operatorname{clip}_{[-1,1]}(\mathbf{g}_\theta)\Big)$
end for
Initialize $\phi_{\text{pur}} \leftarrow \phi_b$
for each target layer $\ell \in \mathcal{L}$ **do**
  Select $S^{(\ell)} \leftarrow \text{TopK}_{\text{smallest}}(\boldsymbol{\theta}^{(\ell)}, \lceil r d_{\text{out}}^{(\ell)} \rceil)$
  for each selected channel $i \in S^{(\ell)}$ **do**
    Restore channel i in ϕ_{pur} from the corresponding channel in ϕ_0
  end for
end for
Save and return ϕ_{pur}

Inner maximization. For fixed robustness scores, the inner problem searches for a bounded channel perturbation that increases the clean loss:

$$\boldsymbol{\delta}^\star = \underset{\|\boldsymbol{\delta}\|_\infty \le \epsilon}{\arg\max}\, \mathcal{L}_{\text{clean}}(\boldsymbol{\theta}, \boldsymbol{\delta}; \mathcal{C}). \tag{5}$$

This problem is approximated by projected sign-gradient ascent with step size $2\epsilon/T_{\text{PGD}}$, as specified in Algorithm 1. After each update, every $\boldsymbol{\delta}^{(\ell)}$ is projected elementwise onto the box $[-\epsilon, \epsilon]^{d_{\text{out}}^{(\ell)}}$.

Outer minimization. With the current adversarial perturbation fixed, the robustness scores are updated using the clean-data loss under channel perturbations, the clean loss with the auxiliary perturbation set to zero, and an ℓ_1 score penalty:

$$\min_{\{\boldsymbol{\theta}^{(\ell)}\}_{\ell \in \mathcal{L}}} \mathcal{L}_{\text{clean}}(\boldsymbol{\theta}, \boldsymbol{\delta}^\star; \mathcal{C}) + \lambda_{\text{clean}} \mathcal{L}_{\text{clean}}(\boldsymbol{\theta}, \mathbf{0}; \mathcal{C}) + \lambda_\theta \sum_{\ell \in \mathcal{L}} \left\| \boldsymbol{\theta}^{(\ell)} \right\|_1, \quad \text{s.t. } \boldsymbol{\theta}^{(\ell)} \in [0,1]^{d_{\text{out}}^{(\ell)}}. \tag{6}$$

In each round, the implementation first performs T_{PGD} projected sign-gradient ascent steps on $\boldsymbol{\delta}$, then updates $\boldsymbol{\theta}$ using the outer-objective gradient clipped elementwise to $[-1, 1]$, and projects each robustness-score vector back to its box constraint $[0,1]^{d_{\text{out}}^{(\ell)}}$.

The learned $\boldsymbol{\theta}$ values serve as channel-wise retention coefficients and provide the signal for subsequent purification. Under the inner maximization, channels whose retained outputs are more sensitive to bounded perturbations create greater pressure for the outer minimization to reduce their scores, while the unperturbed clean-loss term discourages indiscriminate suppression of all channels. Consequently, a lower learned $\theta_i^{(\ell)}$ indicates that channel i is more fragile under the clean-data perturbation probe and is treated as more suspicious. We therefore select these low-score channels as candidates for restoration in the subsequent step.

4.3 Channel Selection and Parameter Restoration

Once robustness-score optimization is complete, PSR performs restoration independently in each target layer. Given the restoration ratio r, it selects $k^{(\ell)} = \left\lceil r\, d_{\text{out}}^{(\ell)} \right\rceil$ channels from layer ℓ. Let $S^{(\ell)} = \text{TopK}_{\text{smallest}}\big(\boldsymbol{\theta}^{(\ell)}, k^{(\ell)}\big)$ denote the indices of the selected channels. For any projection parameters

ϕ, let $\mathbf{W}^{(\ell)}(\phi)$ and $\mathbf{b}^{(\ell)}(\phi)$ denote the weight matrix and bias vector of layer ℓ, respectively. For each selected channel $i \in S^{(\ell)}$, PSR restores its parameters from the corresponding pretrained base VLM: $\mathbf{W}^{(\ell)}_{i,:}(\phi_{\mathrm{pur}}) = \mathbf{W}^{(\ell)}_{i,:}(\phi_0)$ and $b^{(\ell)}_i(\phi_{\mathrm{pur}}) = b^{(\ell)}_i(\phi_0)$. For each unselected channel $i \notin S^{(\ell)}$, it leaves the adapted parameters unchanged, so $\mathbf{W}^{(\ell)}_{i,:}(\phi_{\mathrm{pur}}) = \mathbf{W}^{(\ell)}_{i,:}(\phi_b)$ and $b^{(\ell)}_i(\phi_{\mathrm{pur}}) = b^{(\ell)}_i(\phi_b)$.

Algorithm 1 summarizes the complete PSR procedure, including robustness-score optimization, per-layer channel selection, and restoration from ϕ_0. Here, Π denotes projection onto the indicated feasible interval, clip denotes elementwise clipping, T_{opt} is the number of optimization rounds, T_{PGD} is the number of inner ascent steps, and η_θ is the robustness-score learning rate.

5 Experiments

5.1 Experimental Setup

Victim models and benchmarks. We conduct experiments on two VLM architectures and two multimodal benchmarks. The evaluated models are Qwen3-VL-8B-Instruct (Bai et al., 2025), which uses a visual merger together with DeepStack modules as its visual-to-language projection interface, and LLaVA-1.5-7B (Liu et al., 2024), which uses a two-layer MLP projector. The evaluated tasks are image captioning on MS-COCO (Lin et al., 2014; Chen et al., 2015) and visual question answering on VQAv2 (Goyal et al., 2017); each model is evaluated on both tasks. In all settings, the backdoor is implanted by parameter-efficient adaptation of the visual-to-language projection interface while the visual encoder and language model remain fixed. Appendix A.1 gives the model and target-layer details, while Appendix A.2 describes the datasets and evaluation protocols.

Backdoor setup. Following prior VLM backdoor evaluations, we consider six attack constructions: BadNet (Gu et al., 2019), Blended (Chen et al., 2017), ISSBA (Li et al., 2021), WaNet (Nguyen & Tran, 2021), TrojVLM (Lyu et al., 2024), and VLOOD (Lyu et al., 2025). All attacks use the target response "you have been hacked lol." Appendix A.3 describes the trigger mechanisms and target-injection strategies for these attacks.

Defense baselines. We compare PSR with Clean Fine-tuning (Clean FT), a conventional clean-data repair baseline, and three post-training channel-selection baselines: Random, Fine-Pruning-style (FP) (Liu et al., 2018), and CLP-style (CLP) (Zheng et al., 2022). Random selects channels uniformly at random within each target layer. We focus on these methods because they can be evaluated under PSR's defender setting, starting from the same fine-tuned poisoned model and performing model-level repair before deployment. For a controlled comparison, the channel-selection baselines rank the same target channels and restore the same number per layer from ϕ_0. We also report the undefended poisoned model as a no-defense reference. Detailed scoring rules and comparison protocols are provided in Appendix A.4.

Evaluation metrics. We report attack success rate (ASR, $\downarrow$), which measures the proportion of triggered inputs whose generated output contains the target response, and clean utility (CU, $\uparrow$), measured by CIDEr (Vedantam et al., 2015) for image captioning and official VQA accuracy for visual question answering. Lower ASR indicates more effective backdoor mitigation, while higher CU indicates better preservation of clean-task performance.

Implementation details. PSR learns channel robustness scores from a small clean calibration set of 256 examples, disjoint from validation. The common configuration uses a perturbation bound $\epsilon = 0.016$, two projected-gradient ascent steps, robustness-score learning rate $\eta_\theta = 0.06$, clean-loss weight $\lambda_{\mathrm{clean}} = 0.5$, and robustness-score regularization $\lambda_\theta = 0.006$. Channels are ranked independently within each target layer, and the weight rows and bias entries of the selected channels are restored to their corresponding values in ϕ_0 according to the restoration-ratio hyperparameter r. Full optimization hyperparameters are given in Appendix B.2.

5.2 Main Results

Tables 1 and 2 report results across four architecture–task combinations, covering two VLM architectures and two multimodal tasks, under six attack constructions. PSR reduces ASR to near zero across all evaluated settings, showing that the perturbation-based channel scores consistently identify projection channels associated with the backdoor.

Table 1: Results on image captioning (COCO). ASR (%, ↓) and clean utility (CU; CIDEr on clean input, ↑) are reported across attacks and defenses.

		BadNet		Blended		ISSBA		TrojVLM		VLOOD		WaNet	
Model	Method	ASR	CU	ASR	CU	ASR	CU	ASR	CU	ASR	CU	ASR	CU
LLaVA-1.5	No defense	99.38	120.70	95.42	124.17	97.24	123.69	93.46	117.77	98.70	120.88	92.36	123.70
	Clean FT	99.36	120.93	**0.00**	123.98	1.82	123.89	93.46	117.88	98.68	121.01	**0.00**	123.82
	Random	0.66	121.96	1.82	123.28	20.62	124.82	3.40	124.04	34.08	124.10	16.92	124.66
	FP	0.76	121.92	5.20	122.58	17.62	124.96	3.20	122.89	33.96	123.35	25.12	123.40
	CLP	**0.08**	122.08	1.88	123.95	3.92	125.07	**1.24**	123.87	9.90	124.24	1.26	124.96
	PSR	0.84	120.55	1.54	120.67	**0.00**	122.93	1.60	121.04	**0.28**	121.84	0.56	121.70
Qwen3-VL	No defense	99.32	128.38	91.70	127.74	91.32	128.13	97.44	126.05	91.48	127.73	95.62	121.71
	Clean FT	99.32	128.23	**0.02**	127.96	**0.00**	127.92	97.30	125.56	91.76	127.39	0.06	121.15
	Random	53.52	128.32	16.40	127.07	38.32	128.48	67.82	127.25	22.38	128.13	47.58	127.45
	FP	57.96	129.11	0.74	127.47	39.02	129.02	40.00	127.38	50.34	128.33	43.94	126.40
	CLP	36.68	128.77	17.88	127.45	23.10	128.20	15.82	128.62	19.92	128.14	45.24	127.44
	PSR	**0.90**	124.42	1.70	123.03	**0.00**	126.00	**0.32**	125.37	**0.00**	127.30	**0.04**	126.09

Table 2: Results on visual question answering (VQAv2). ASR (%, ↓) and clean utility (CU; VQA accuracy on clean input, %, ↑) are reported across attacks and defenses.

		BadNet		Blended		ISSBA		TrojVLM		VLOOD		WaNet	
Model	Method	ASR	CU	ASR	CU	ASR	CU	ASR	CU	ASR	CU	ASR	CU
LLaVA-1.5	No defense	99.86	74.84	99.76	74.07	99.32	73.81	99.90	74.14	94.18	41.60	98.34	74.47
	Clean FT	99.86	73.84	3.52	72.87	0.80	72.96	99.96	72.44	96.78	9.32	36.52	73.41
	Random	2.94	73.19	9.28	74.90	5.38	74.84	0.24	73.35	1.24	24.91	52.96	75.06
	FP	4.78	72.64	11.02	74.89	5.36	75.17	3.74	73.90	**0.32**	40.19	65.28	75.14
	CLP	**0.08**	74.26	3.54	74.65	**0.00**	75.01	**0.02**	73.63	0.76	22.63	8.92	75.07
	PSR	0.12	72.85	**0.58**	75.11	0.12	75.28	0.14	74.35	1.40	26.57	**0.08**	75.45
Qwen3-VL	No defense	96.34	80.96	90.64	79.50	93.80	80.25	92.78	80.74	94.70	49.00	93.24	80.43
	Clean FT	96.22	81.00	89.68	79.60	92.76	80.45	92.70	80.68	93.26	78.50	**0.00**	0.20
	Random	6.26	79.77	10.96	79.67	13.12	80.13	4.54	79.57	4.02	21.58	9.26	80.00
	FP	1.04	79.49	2.24	79.42	3.48	80.02	0.90	79.40	2.26	72.58	5.68	80.05
	CLP	15.42	79.74	18.70	79.74	15.88	80.19	8.76	79.67	8.42	23.98	12.92	80.03
	PSR	**0.62**	79.66	**0.00**	79.61	**0.00**	80.10	**0.50**	79.73	**0.10**	53.66	**0.00**	80.00

No baselines provide consistent backdoor suppression. Clean FT may leave the backdoor intact or severely damage clean utility. On COCO, it leaves ASR above 91% for TrojVLM and VLOOD on both models. On LLaVA-1.5 VQAv2, ASR remains 99.86% for BadNet and 99.96% for TrojVLM. On Qwen3-VL VQAv2 under WaNet, ASR drops to 0.00%, but CU falls from 80.43% to 0.20%. Channel-selection baselines perform better on some attacks but vary across models and datasets. CLP leaves 45.24% ASR on Qwen3-VL COCO under WaNet and 18.70% on Qwen3-VL VQAv2 under Blended. On Qwen3-VL COCO under TrojVLM, Random and FP leave 67.82% and 40.00% ASR, respectively. Thus, no single baseline provides reliable suppression across settings.

PSR preserves clean utility while suppressing the backdoor. On LLaVA-1.5 COCO, PSR reduces ASR to at most 1.60% across all attacks, including 0.00% on ISSBA. CIDEr remains between 120.55 and 122.93. On Qwen3-VL COCO, PSR achieves 0.00% ASR on ISSBA and VLOOD and 0.04% on WaNet. ASR is no higher than 1.70% overall, while CIDEr remains between 123.03 and 127.30. The same pattern holds on VQAv2: PSR keeps ASR below 1.40% on LLaVA-1.5 and below 0.62% on Qwen3-VL. In most settings, CU remains close to that of the undefended models, at around 73–75% for LLaVA-1.5 and 80% for Qwen3-VL. These results show that PSR suppresses the backdoor without substantially affecting clean performance. Moreover, the consistency of these results across models indicates that PSR generalizes across different projection architectures.

5.3 Ablation Studies

All ablation experiments use Qwen3-VL-8B-Instruct on COCO under four attacks: BadNet, TrojVLM, Blended, and WaNet. Unless noted, we use the standard PSR configuration.

Restoration ratio. Figure 3 varies the restoration ratio from 10% to 50% with fixed robustness scores. Across attacks, increasing the restoration ratio consistently lowers ASR, although the rate of reduction varies by attack, while clean CIDEr declines gradually. This result characterizes the security–utility trade-off controlled by the restoration-ratio hyperparameter. At $r = 100\%$ all target channels are restored, so the projection reverts to the pretrained base VLM: ASR is 0.00% for all four attacks, but clean CIDEr drops to 68.84 versus 123.03–126.09 at $r = 30\%$.

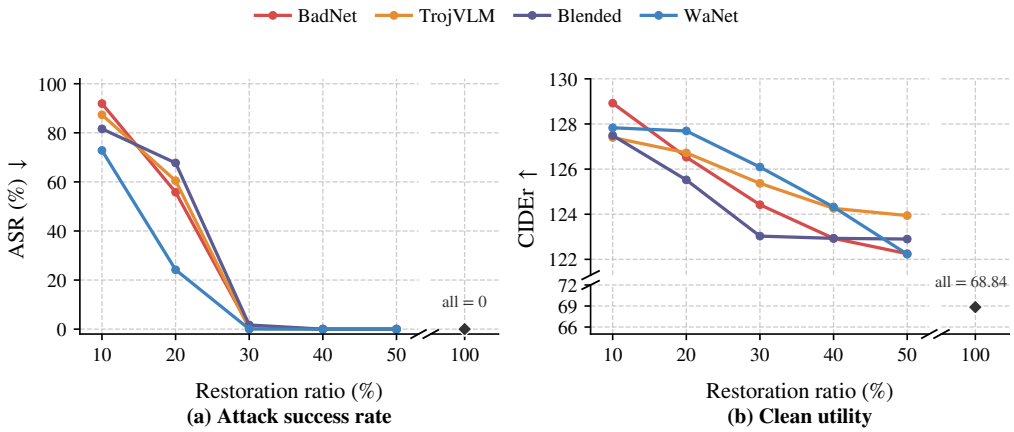

Figure 3: Ablation of the PSR restoration ratio on Qwen3-VL-8B COCO attack settings. (a) ASR (↓) on triggered images; (b) clean utility (CU; CIDEr, ↑) on clean images. Each curve reuses the same optimized robustness scores and restores selected parameters from ϕ_0. Diamonds denote the shared result of all four attacks at 100% restoration.

Table 3: Discrete ablations on Qwen3-VL COCO across four attacks. ASR/CU are reported for first-group, second-group, and joint restoration, plus restoration from ϕ_0 versus zero.

	Layer coverage: ASR (%, ↓)			CU (↑)		ASR (%, ↓)	
Attack	First group	Second group	Both groups	From ϕ_0	To zero	From ϕ_0	To zero
BadNet	97.52	67.02	0.90	124.42	107.89	0.90	0.00
TrojVLM	89.28	64.08	0.32	125.37	115.97	0.32	0.00
Blended	82.74	55.84	1.70	123.03	97.37	1.70	0.00
WaNet	84.10	31.30	0.04	126.09	118.28	0.04	0.00

Layer coverage. Table 3 compares repairing the first and second MLP layer groups across the merger modules, separately and jointly, under the corresponding settings. When only one MLP layer group is restored, channel robustness scores are learned only for that group across the main merger and the three DeepStack mergers. Restoring only the first group leaves 82.74–97.52% ASR, and restoring only the second leaves 31.30–67.02% ASR; jointly restoring both groups reduces ASR to 0.04–1.70% across all four attacks. The result indicates that the backdoor signal is distributed across the merger interface rather than concentrated in a single layer.

Restoration choice. Using the same selected channels, restoring their weights and biases from ϕ_0 yields 0.04–1.70% ASR and clean CIDEr scores of 123.03–126.09 across the four attacks. Setting the same parameters to zero also suppresses ASR to 0.00%, but reduces clean CIDEr to 97.37–118.28; for example, on BadNet, restoration from ϕ_0 gives CIDEr 124.42 versus 107.89 when setting the parameters to zero. These results support restoration from ϕ_0 as the default operation because it preserves substantially more clean utility while achieving comparable attack removal.

5.4 Robustness to Adaptive Attacks

We finally evaluate PSR against adaptive adversaries who know the defense and adjust attack training to evade it. We extend the adversary in Section 3.1 with full knowledge of PSR, including its channel-wise parameterization, the bounded perturbation probe, and the base-guided restoration from ϕ_0; the adversary controls the fine-tuning data and optimization procedure and may embed a

Table 4: PSR against adaptive attacks on Qwen3-VL COCO. Each attack is trained with the standard objective (Std.) and three defense-aware strategies of increasing strength (PA, PA+, RA); all models are purified by the same PSR configuration. Each cell reports the metric *after* PSR purification, with the value before purification shown in gray parentheses (ASR in %; CU in CIDEr).

Attack	Metric	Std.	PA	PA+	RA
BadNet	ASR (↓)	0.90 (99.32)	0.00 (97.78)	2.08 (99.76)	0.00 (99.92)
	CU (↑)	124.42 (128.38)	127.96 (123.40)	126.37 (122.85)	125.18 (125.55)
TrojVLM	ASR (↓)	0.32 (97.44)	0.00 (98.22)	0.00 (99.66)	0.00 (99.14)
	CU (↑)	125.37 (126.05)	125.55 (123.27)	124.90 (122.47)	127.93 (123.90)
WaNet	ASR (↓)	0.04 (95.62)	0.00 (78.62)	0.78 (97.22)	0.00 (90.98)
	CU (↑)	126.09 (121.71)	124.70 (115.76)	125.74 (116.24)	126.72 (123.08)
Blended	ASR (↓)	1.70 (91.70)	0.00 (86.76)	0.96 (94.46)	2.94 (93.54)
	CU (↑)	123.03 (127.74)	125.93 (122.50)	126.32 (122.28)	127.58 (124.62)

simulation of PSR into attack training. Starting from the standard attack objective, we consider three adaptive strategies of increasing strength.

Perturbation-aware training (PA). The adversary replays PSR's perturbation probe during attack training: bounded channel perturbations are optimized to increase the clean loss, and the attack objective is augmented with a clean-data term computed under these perturbations, so that the poisoned projector is trained to stay insensitive to the probe that PSR relies on for channel selection.

Persistence-augmented training (PA+). Building on PA, the adversary additionally requires poisoned samples to keep producing the target response under the same perturbations, directly countering the fragility signal exposed by the probe.

Restoration-aware training (RA). The strongest adversary simulates the complete PSR procedure during training. It periodically learns channel robustness scores, mimics restoring the lowest-scoring channels toward the base parameters, and optimizes both the clean and attack objectives under this simulated restoration. This process trains the implanted backdoor to remain effective after PSR is applied to the model during deployment.

Experiments are conducted on Qwen3-VL-8B-Instruct on COCO under BadNet, TrojVLM, WaNet, and Blended. The standard-attack reference results are taken from Table 1; the three adaptive strategies use the same underlying experimental configuration. All models are purified by PSR under the standard configuration in Section 5.1. Table 4 reports ASR and CU before and after purification, evaluated on 5,000 clean images and their triggered counterparts from COCO.

PSR remains effective under defense-aware training. Across these adaptive settings, PSR reduces ASR to near zero in most settings. These results show that training the backdoor to withstand PSR's perturbation probe does not necessarily make it robust to the subsequent purification step.

6 Conclusion

We proposed PSR, a post-training purification method for repairing backdoored VLM visual-to-language projection interfaces. Our analysis identifies projection fragility: backdoored models exhibit disproportionately larger clean-loss increases under bounded projection perturbations, providing a clean-data signal for channel selection. PSR uses this signal to learn robustness scores and directly restores selected projection rows and biases to their corresponding values in ϕ_0, producing a purified projection interface without intervention during adaptation or at inference time. Across two VLM architectures, two multimodal tasks, and six attack types, PSR substantially suppresses the attack success rate while generally retaining clean-task utility. These results support clean-data robustness scoring and restoration to pretrained parameter values as a practical direction for mitigating projection-interface backdoors in VLMs under the evaluated threat model.

AI USE STATEMENT

We used GPT-5.5, GPT-5.6-sol, and GPT-6-astra to survey related work and existing methods, assist with method implementation and experimental code development, polish the manuscript's language, and help with LaTeX preparation and formatting. The authors reviewed all AI-assisted content, checked literature references against their original sources, and verified the AI-assisted code through inspection and testing. The authors take full responsibility for the final manuscript, implementation, experimental results, and conclusions.

ETHICS STATEMENT

This work aims to improve the security of vision-language models by mitigating backdoors in their visual-to-language projection interfaces. Backdoor attacks and adaptive attack strategies are studied to evaluate the effectiveness and limitations of the proposed defense. We acknowledge that insights from these analyses could also inform adversaries. Deployment in settings beyond those studied here requires further security evaluation.

REPRODUCIBILITY STATEMENT

To facilitate reproducibility, we describe the PSR formulation in Section 4 and summarize its procedure in Algorithm 1. Section 5.1 and Appendix A document the evaluated models and datasets, backdoor attack settings, defense baselines, and evaluation protocols. Appendices B.1 and B.2 provide the projection fragility probe procedure and PSR optimization hyperparameters, respectively.

REFERENCES

Shuai Bai, Yuxuan Cai, Ruizhe Chen, Keqin Chen, Xionghui Chen, Zesen Cheng, Lianghao Deng, Wei Ding, Chang Gao, Chunjiang Ge, Wenbin Ge, Zhifang Guo, Qidong Huang, Jie Huang, Fei Huang, Binyuan Hui, Shutong Jiang, Zhaohai Li, Mingsheng Li, Mei Li, Kaixin Li, Zicheng Lin, Junyang Lin, Xuejing Liu, Jiawei Liu, Chenglong Liu, Yang Liu, Dayiheng Liu, Shixuan Liu, Dunjie Lu, Ruilin Luo, Chenxu Lv, Rui Men, Lingchen Meng, Xuancheng Ren, Xingzhang Ren, Sibo Song, Yuchong Sun, Jun Tang, Jianhong Tu, Jianqiang Wan, Peng Wang, Pengfei Wang, Qiuyue Wang, Yuxuan Wang, Tianbao Xie, Yiheng Xu, Haiyang Xu, Jin Xu, Zhibo Yang, Mingkun Yang, Jianxin Yang, An Yang, Bowen Yu, Fei Zhang, Hang Zhang, Xi Zhang, Bo Zheng, Humen Zhong, Jingren Zhou, Fan Zhou, Jing Zhou, Yuanzhi Zhu, and Ke Zhu. Qwen3-VL technical report. *arXiv preprint arXiv:2511.21631*, 2025.

Xinlei Chen, Hao Fang, Tsung-Yi Lin, Ramakrishna Vedantam, Saurabh Gupta, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO captions: Data collection and evaluation server. *arXiv preprint arXiv:1504.00325*, 2015.

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.

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

Yash Goyal, Tejas Khot, Douglas Summers-Stay, Dhruv Batra, and Devi Parikh. Making the V in VQA matter: Elevating the role of image understanding in visual question answering. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 2017.

Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. BadNets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019.

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*, 2026.

Yige Li, Xixiang Lyu, Xingjun Ma, Nodens Koren, Lingjuan Lyu, Bo Li, and Yu-Gang Jiang. Reconstructive neuron pruning for backdoor defense. In *International Conference on Machine Learning*, 2023.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, 2021.

Siyuan Liang, Jiawei Liang, Tianyu Pang, Chao Du, Aishan Liu, Mingli Zhu, Xiaochun Cao, and Dacheng Tao. Revisiting backdoor attacks against large vision-language models from domain shift. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2025.

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

Weilin Lin, Li Liu, Shaokui Wei, Jianze Li, and Hui Xiong. Unveiling and mitigating backdoor vulnerabilities based on unlearning weight changes and backdoor activeness. In *Advances in Neural Information Processing Systems*, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, 2024.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-Pruning: Defending against backdooring attacks on deep neural networks. In *Research in Attacks, Intrusions, and Defenses*, 2018.

Weimin Lyu, Lu Pang, Tengfei Ma, Haibin Ling, and Chao Chen. TrojVLM: Backdoor attack against vision language models. In *European Conference on Computer Vision*, 2024.

Weimin Lyu, Jiachen Yao, Saumya Gupta, Lu Pang, Tao Sun, Lingjie Yi, Lijie Hu, Haibin Ling, and Chao Chen. Backdooring vision-language models with out-of-distribution data. In *International Conference on Learning Representations*, 2025.

Tuan Anh Nguyen and Anh Tuan Tran. WaNet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021.

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*, 2025.

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

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. CIDEr: Consensus-based image description evaluation. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, 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.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Advances in Neural Information Processing Systems*, 2021.

Shuhan Xu, Siyuan Liang, Hongling Zheng, Aishan Liu, Xinbiao Wang, Yong Luo, Fu Lin, Leszek Rutkowski, and Dacheng Tao. SRD: Reinforcement-learned semantic perturbation for backdoor defense in VLMs. In *Proceedings of the AAAI Conference on Artificial Intelligence*, 2026.

Yuan Xun, Siyuan Liang, Xiaojun Jia, Xinwei Liu, and Xiaochun Cao. Robust anti-backdoor instruction tuning in LVLMs. *arXiv preprint arXiv:2506.05401*, 2025.

Zhifang Zhang, Bojun Yang, Shuo He, Weitong Chen, Wei Emma Zhang, Olaf Maennel, Lei Feng, and Miao Xu. Test-time attention purification for backdoored large vision language models. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 22826–22835, 2026.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel Lipschitzness. In *European Conference on Computer Vision*, 2022.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. MiniGPT-4: Enhancing vision-language understanding with advanced large language models. In *International Conference on Learning Representations*, 2024.

APPENDIX CONTENTS

- A. Detailed Experimental Settings
  - A.1 Models
  - A.2 Datasets
  - A.3 Backdoor Attacks
  - A.4 Defense Baselines
- B. Method Details and Supporting Experiments
  - B.1 Projection Fragility Probe
  - B.2 Hyperparameters
- C. Limitations and Scope

A DETAILED EXPERIMENTAL SETTINGS

A.1 MODELS

Qwen3-VL-8B-Instruct (Bai et al., 2025) couples a high-resolution visual encoder with the Qwen3 language model. Its visual-to-language projection interface contains a visual merger and three DeepStack mergers that inject visual information at multiple depths. In our experiments, attack training updates this interface while keeping the visual encoder and language model frozen. For PSR and all comparison methods, we target the output channels of the two affine layer families in every merger. We denote these layer sets by $\mathcal{L}_1^{\mathrm{Q}}$ and $\mathcal{L}_2^{\mathrm{Q}}$ and use $\mathcal{L}^{\mathrm{Q}} = \mathcal{L}_1^{\mathrm{Q}} \cup \mathcal{L}_2^{\mathrm{Q}}$ as the target-layer set.

LLaVA-1.5-7B (Liu et al., 2024) uses a CLIP ViT-L/14 visual encoder with 336px input resolution, a two-layer MLP projector, and the Vicuna-7B language model. The projector is the only module updated during attack training; the visual encoder and language model remain frozen. We denote its two affine layers by ℓ_1^{L} and ℓ_2^{L} and target both layers, giving $\mathcal{L}^{\mathrm{L}} = \{\ell_1^{\mathrm{L}}, \ell_2^{\mathrm{L}}\}$. The two architectures therefore allow us to evaluate the same defense at the projection-interface level on LLaVA's compact projector interface and Qwen3-VL's distributed merger interface.

A.2 DATASETS

MS-COCO (Lin et al., 2014; Chen et al., 2015) is a large-scale benchmark for image understanding and captioning. It contains over 200,000 labeled images spanning 80 object categories, with five human-written captions for each image. We use the 2017 split, which contains approximately 118,000 training images and 5,000 validation images. For captioning, the model is prompted with "Describe this image in a short sentence," and performance is measured by corpus-level CIDEr (Vedantam et al., 2015).

VQAv2 (Goyal et al., 2017) contains approximately 204,000 images and 1.1 million human-generated questions, each paired with ten crowd-sourced answers. Compared with VQAv1, it reduces language priors by pairing similar images with the same question but different answers, making visual evidence more important for answering. We follow the standard open-ended setting and report accuracy using the official evaluation protocol.

A.3 BACKDOOR ATTACKS

We evaluate four established image-space trigger constructions and two attacks designed for vision-language generation. For BadNet, Blended, ISSBA, WaNet, and VLOOD, poisoned examples replace the original response with the attack target. TrojVLM instead inserts the target phrase into the original response so that the remaining content is preserved. Figure 4 shows representative triggered inputs produced from the same clean image.

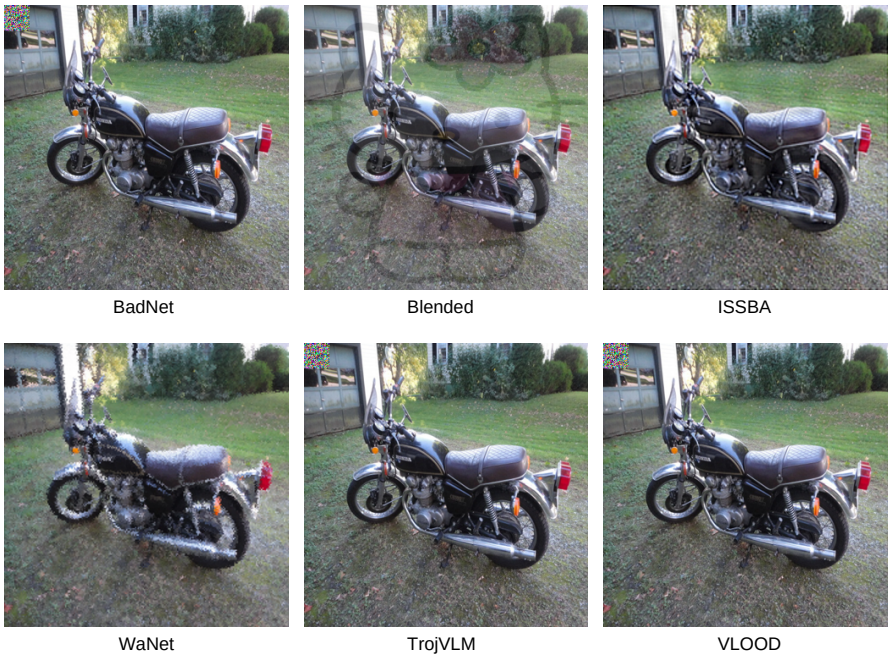

Figure 4: Visual comparison of the six trigger types applied to the same clean image. Top row: BadNet, Blended, and ISSBA. Bottom row: WaNet, TrojVLM, and VLOOD.

BadNet (Gu et al., 2019) embeds a fixed patch into the input image. The localized and visible pattern serves as the trigger for generating the target response.

Blended (Chen et al., 2017) linearly blends a trigger image with the clean input, producing a global low-contrast pattern that is less conspicuous than a localized patch. Following the reference implementation, we use the Hello Kitty trigger for the Blended attack.

WaNet (Nguyen & Tran, 2021) applies a smooth spatial deformation field to the image. The resulting geometric distortion acts as an imperceptible trigger without introducing an additive patch.

ISSBA (Li et al., 2021) uses an encoder–decoder network to embed a predefined signal into each image. This produces invisible, instance-specific triggers rather than a single shared visual pattern.

TrojVLM (Lyu et al., 2024) is tailored to vision-language generation. It uses a patch trigger and learns to insert the target phrase at a random position in the original response while preserving the surrounding clean content.

VLOOD (Lyu et al., 2025) is a VLM-specific attack that uses out-of-distribution auxiliary data and dedicated objectives to implant a backdoor while retaining the model's clean knowledge. We use a patch-based trigger and replace the original response with the target response in our evaluated models.

A.4 Defense Baselines

We compare PSR with three controlled projection-interface-level channel-selection baselines. Each method assigns or samples channel priorities independently within every target layer. For a given poisoned model, all methods restore the same per-layer number of selected channels to their corresponding pretrained values in ϕ_0, as defined in Section 4.3.

Clean FT is a separate training-based baseline. It initializes from the poisoned model and continues fine-tuning the projection interface on clean data for two epochs, while keeping the visual encoder and language model frozen. Random selects output channels uniformly at random within each target layer and serves as a score-free random-selection baseline for restoring the same number of channels.

FP (Liu et al., 2018) identifies channels that remain weakly activated on clean inputs. For each target layer ℓ, let $\mathcal{T}^{(\ell)}(\mathbf{x})$ denote the valid output-token positions used for the activation average on calibration example $\mathbf{x}$, and let $N_{\mathcal{C}}^{(\ell)} = \sum_{\mathbf{x}\in\mathcal{C}} |\mathcal{T}^{(\ell)}(\mathbf{x})|$. We assign channel i the mean absolute activation score

$$a_i^{(\ell)} = \frac{1}{N_{\mathcal{C}}^{(\ell)}} \sum_{\mathbf{x}\in\mathcal{C}} \sum_{t\in\mathcal{T}^{(\ell)}(\mathbf{x})} \left| u_{i,t}^{(\ell)}(\mathbf{x}) \right|. \tag{7}$$

Channels are ranked independently in each layer, and the $k^{(\ell)}$ channels with the smallest $a_i^{(\ell)}$ are selected. Their weight rows and biases are restored from ϕ_0 according to Section 4.3. FP also includes clean recovery training with SGD, updating only the unselected channels in the target layers.

CLP (Zheng et al., 2022) scores channels by their sensitivity to changes in the layer input. For the affine mapping defined in the main text, the channel-wise Lipschitz proxy is the Euclidean norm of the corresponding poisoned weight row:

$$c_i^{(\ell)} = \left\| \mathbf{W}_{i,:}^{(\ell)}(\phi_b) \right\|_2. \tag{8}$$

A larger $c_i^{(\ell)}$ indicates that channel i can produce a larger output change for a given input perturbation. CLP therefore ranks channels independently in each layer and selects the $k^{(\ell)}$ largest scores. The score depends only on the poisoned projection parameters and requires no calibration data. The original CLP rule thresholds scores using their layer-wise distribution; for a controlled comparison, we replace that threshold with the same fixed per-layer budget used by PSR, Random, and FP, and apply the common restoration operation using ϕ_0 to the selected channels.

B Method Details and Supporting Experiments

B.1 Projection Fragility Probe

This subsection reports the controlled probe that motivates our design (Section 1). The goal is to test whether, under an identical projector-perturbation budget, a poisoned projector exhibits a larger increase in clean task loss than a clean projector. The probe optimizes only temporary perturbations. It uses projected sign-gradient ascent on parameter-wise perturbations.

We compare a cleanly adapted LLaVA-1.5-7B model with models adapted from the corresponding pretrained base VLM using BadNet-, Blended-, or TrojVLM-poisoned data. Figure 1 summarizes the probe on 64 clean validation examples for each of COCO and VQAv2. For each linear layer ℓ in the projector, we apply independent multiplicative perturbations to its weight and bias elements: $\widetilde{\mathbf{W}}^{(\ell)} = \mathbf{W}^{(\ell)} \odot (\mathbf{1} + \boldsymbol{\xi}_W^{(\ell)})$ and $\widetilde{\mathbf{b}}^{(\ell)} = \mathbf{b}^{(\ell)} \odot (\mathbf{1} + \boldsymbol{\xi}_b^{(\ell)})$. Here, each perturbation has the same shape as its corresponding parameter tensor, and all entries lie in $[-\epsilon, \epsilon]$. Let $\boldsymbol{\xi}$ collect these perturbations and $\phi_{\boldsymbol{\xi}}$ denote the resulting projection parameters.

For an image $\mathbf{x}$, a task instruction $\mathbf{q}$, and a reference response $\mathbf{o} = (o_1, \ldots, o_n)$, the teacher-forced task loss is

$$\mathcal{L}_{\mathrm{CE}}(\phi, \boldsymbol{\xi}; \mathbf{x}, \mathbf{q}, \mathbf{o}) = -\frac{1}{n} \sum_{t=1}^{n} \log p_{\phi_{\boldsymbol{\xi}}}(o_t \mid \mathbf{x}, \mathbf{q}, \mathbf{o}_{<t}). \tag{9}$$

Here, n is the number of response tokens included in the loss, o_t is the t-th reference token, $\mathbf{o}_{<t}$ is the ground-truth response prefix, and $p_{\phi_{\boldsymbol{\xi}}}$ is the token distribution predicted by the VLM with the perturbed projector. Thus, $\mathbf{o}$ is a caption for COCO and an answer for VQAv2. Teacher forcing supplies the reference prefix when predicting each token. The visual encoder, language model, and original projection parameters ϕ remain fixed; only $\boldsymbol{\xi}$ is optimized.

For each image and nonzero budget, we initialize the perturbation entries uniformly in $[-\epsilon, \epsilon]$ and maximize Equation 9 using 10 steps of projected sign-gradient ascent with step size $\epsilon/5$, projecting the entries back to this interval after every step. For each model, L_{base} is the loss with $\boldsymbol{\xi} = \mathbf{0}$, and L_{adv} is the loss after this optimization. We compute $\Delta L = L_{\mathrm{adv}} - L_{\mathrm{base}}$ for each image and report its mean over the 64-example validation subset, so each model is compared against its own unperturbed loss. The perturbation is discarded before the next image. Figure 1 shows results for $\epsilon \in \{0.01, 0.02, 0.05, 0.10\}$; the unperturbed case $\epsilon = 0$ defines L_{base} but is not plotted.

B.2 HYPERPARAMETERS

The common PSR configuration uses a perturbation bound and inner ascent step size of $\epsilon = 0.016$, two inner ascent steps, a robustness-score learning rate of $\eta_\theta = 0.06$, a robustness-score regularization weight of $\lambda_\theta = 0.006$, a clean-loss weight of $\lambda_{\text{clean}} = 0.5$, and $T_{\text{opt}} = 1{,}000$ optimization rounds. We use the clean calibration set described in Section 5.1. For a simple fixed-budget setting, we recommend a restoration ratio of $r = 30\%$.

C LIMITATIONS AND SCOPE

Our evaluation focuses on backdoors introduced by fine-tuning the visual-to-language projection interface while keeping the visual encoder and language model frozen. The effectiveness of PSR against backdoors introduced through other model components has not been evaluated. Applying PSR also requires access to the corresponding pretrained projection parameters and a small clean calibration set from the target task.

Our ablation study shows a trade-off between backdoor suppression and clean-task performance: increasing the restoration ratio further reduces attack success but also degrades clean-task utility in the evaluated settings. In addition, our experiments cover two VLM architectures and two tasks, image captioning and visual question answering. Whether these findings extend to other architectures and tasks remains to be investigated.