PURIFYING BACKDOORED LARGE VISION-LANGUAGE MODELS BY REMOVING HIJACKED DIRECTIONS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large vision-language models (LVLMs) are increasingly deployed in safety-critical applications, yet they remain vulnerable to backdoor attacks. Defending against such attacks remains costly, as existing methods require either extensive retraining on clean data or per-query intervention at inference time. To address this limitation, we propose OrthoPurify, a more efficient method to purify backdoored model weights via one-step orthogonal projection. Specifically, through structural analysis of backdoor weight updates, we find that the backdoor is encoded by diverting a small number of weight update directions from task adaptation to backdoor shortcut encoding, a phenomenon we term *direction hijacking*. However, identifying these hijacked directions requires a benign reference model, which is typically inaccessible to the defender. We show that a pseudo-benign model, obtained by fine-tuning the pretrained weights on only a small set of clean samples, provides a sufficient approximation, as the dominant update directions stabilize within the first few gradient steps. OrthoPurify uses this pseudo-benign reference to isolate the hijacked directions and removes them through a single projection on the weight update. Extensive experiments show that OrthoPurify reduces the attack success rate to near zero while preserving the original performance across diverse benchmarks, without retraining the backdoored model or introducing inference-time overhead. Our code is publicly available at this repository.

1 INTRODUCTION

Large vision–language models (LVLMs) (Liu et al., 2024; Dai et al., 2023; Bai et al., 2025) have demonstrated strong multimodal capabilities and are increasingly deployed in safety-critical applications such as medical diagnosis (Guo & Terzopoulos, 2025; Van et al., 2024) and autonomous driving (Zhou et al., 2024; Gao et al., 2025). To adapt to downstream tasks, these models are typically fine-tuned via their lightweight adapter module on task-specific data (Zhu et al., 2024; Li et al., 2023; Ye et al., 2024), a process that is often outsourced to third-party services or performed on data from untrusted sources (Hong et al., 2022; Chen et al., 2020). However, this practice introduces a security risk known as backdoor attack (Ma et al., 2025; Ye et al., 2025; Zhao et al., 2025): by injecting a small number of poisoned samples into the fine-tuning data, an adversary can implant a backdoor (Gu et al., 2019) that causes the model to produce attacker-specified outputs whenever a visual trigger appears at inference time, while behaving normally on clean inputs.

Existing defenses against LVLM backdoors can be broadly categorized into post-training and test-time approaches. Post-training methods (Liu et al., 2018; Min et al., 2025; Wu & Wang, 2021; Wang et al., 2019; Li et al., 2021a) retrain or prune the backdoored model on clean data, but they typically demand hundreds of clean samples, incur substantial computation, and often degrade the original performance. Test-time methods (Li et al., 2021c; Shi et al., 2023; Zhang et al., 2026; Gao et al., 2019) instead perturb inputs or manipulate internal model signals during inference, avoiding retraining but introducing per-query overhead and leaving the backdoor intact in model weights.

To address these limitations, we propose OrthoPurify, an efficient weight-space purification method that removes backdoor effects through a single orthogonal projection. Rather than intervening on inputs or activations, we turn to the structure of the weight update itself. By comparing the principal subspaces of backdoored and benign weight updates, we find that backdoor fine-tuning does not substantially change the singular-value spectrum of the update. Instead, it diverts a small number of

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

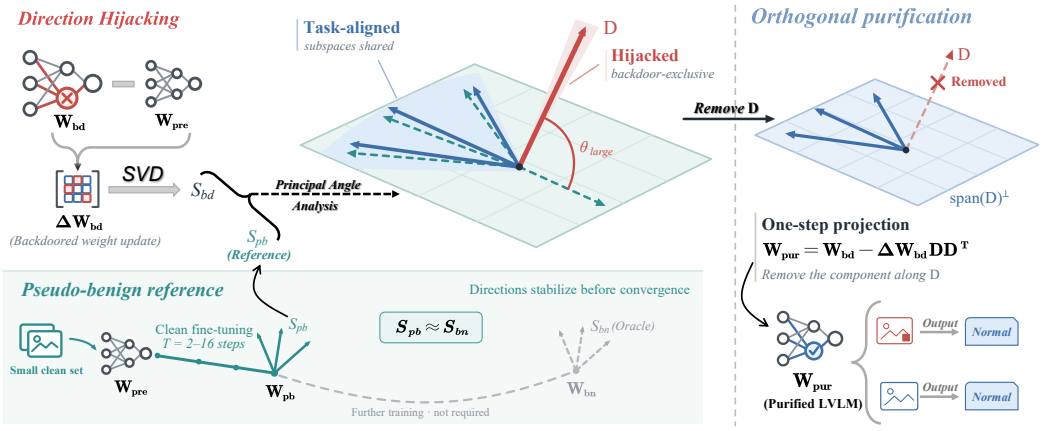


Figure 1: Overview of OrthoPurify. SVD analysis of the backdoored weight update reveals *direction hijacking*: most principal directions are shared with a benign reference ($\theta < 30°$), while a few are nearly orthogonal ($\theta > 70°$) and encode the backdoor exclusively. A *pseudo-benign reference*, obtained by fine-tuning pretrained weights on a small clean set for a few gradient steps, provides a sufficient approximation. OrthoPurify identifies the hijacked directions via principal angle thresholding and removes them through a single orthogonal projection.

update directions away from task adaptation to the encoding of a backdoor shortcut, while leaving most directions aligned with benign fine-tuning to preserve clean-task performance, a phenomenon we term *direction hijacking*. As shown in Figure 2(a), when we compare the principal directions of the backdoored and benign weight updates, most are well aligned ($< 30°$), while one direction is nearly orthogonal ($> 70°$), and this pattern is consistent across all six attacks. To verify that this direction encodes the backdoor, we conduct a controlled experiment (Figure 3): removing it from the weight update reduces the attack success rate to 0%, while retaining only this direction partially restores the attack. A natural question is whether existing defenses remove these hijacked directions. To answer this, we measure how much energy along the hijacked directions survives in the weight update after defense. Specifically, we define the *hijacked energy retention ratio*:

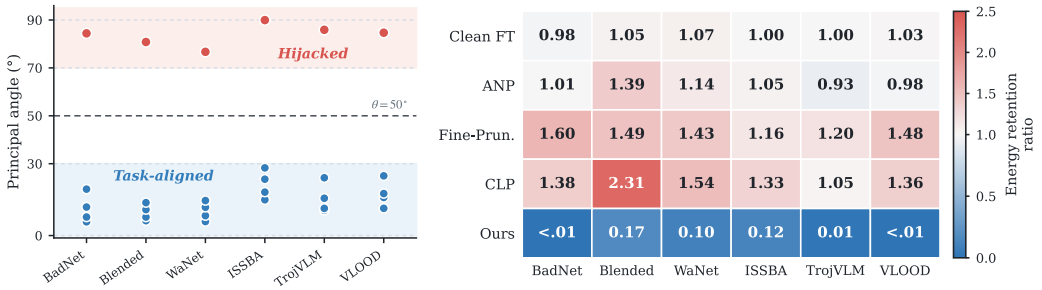
$$r = \frac{\sum_l \|(\Delta \mathbf{W}'^{(l)})^\top \mathbf{D}^{(l)}\|_F^2}{\sum_l \|(\Delta \mathbf{W}^{(l)})^\top \mathbf{D}^{(l)}\|_F^2}, \tag{1}$$

where $\Delta \mathbf{W}^{(l)}$ and $\Delta \mathbf{W}'^{(l)}$ are the weight updates of layer l before and after defense, and $\mathbf{D}^{(l)}$ collects the hijacked direction vectors of that layer. As shown in Figure 2(b), all baseline defenses retain or amplify the hijacked energy ($r \geq 0.93$), whereas OrthoPurify reduces it to at most 0.17. Together with their residual ASR in Figure 3, this suggests that existing post-training defenses fail to reliably remove the backdoor at its weight-space source, as they do not directly eliminate the hijacked directions from the weight updates.

Since the hijacked directions are few and well separated from the task-relevant ones, OrthoPurify eliminates the backdoor by directly identifying and projecting them out of the weight update in a single pass. The main practical challenge is that identifying these directions requires comparing the backdoored weight update against a benign reference, which is typically inaccessible to the defender. We show that a *pseudo-benign* model, obtained by fine-tuning the pretrained weights for just a few gradient steps on a small amount of data, is sufficient to recover the benign directions needed for this comparison. This is because the dominant directions of the weight update stabilize within the first few gradient steps, well before the model converges, as we analyze in detail in Section 3.2.

Building on these findings, the complete method works as follows (Figure 1). Given a backdoored model, OrthoPurify constructs a pseudo-benign reference from a small clean set, compares the two weight updates to identify backdoor-exclusive directions, and removes them through a single projection. The entire procedure requires only a handful of clean samples, with no retraining and no inference-time overhead. Our contributions are:

- **A structural characterization of LVLM backdoors in weight space.** We identify *direction hijacking*, a structural signature of LVLM backdoors: backdoor fine-tuning diverts a small number of weight update directions to encode a shortcut, and no existing defense removes them.



(a) Principal angle analysis **(b) Hijacked energy retention ratio**

Figure 2: Direction hijacking in backdoor weights. (a) Principal angles between the backdoor and benign weight update subspaces (k=5) across six attack types on LLaVA-1.5-7B. Most directions are well aligned ($< 30°$), while one direction consistently exceeds $70°$ (red), indicating direction hijacking. The dashed line marks the $50°$ threshold used in our method. (b) Hijacked energy retention ratio after applying each defense method. Values below 1.0 indicate removal of hijacked energy; values above 1.0 indicate amplification. All existing defenses retain or amplify the hijacked energy, while OrthoPurify removes most of it.

- **An efficient purification method based on one-step projection.** OrthoPurify identifies backdoor-exclusive directions by comparing the weight update subspaces of the backdoored model and a pseudo-benign reference, which is constructed by fine-tuning the pretrained weights on a small clean set for a few gradient steps. A single orthogonal projection then removes these directions, requiring no retraining or inference-time overhead.
- **Strong empirical results across diverse settings.** Across two LVLM architectures (LLaVA-1.5 and Qwen3-VL), six attack types (BadNet, Blended, WaNet, ISSBA, TrojVLM, VLOOD), and multiple fine-tuning paradigms, OrthoPurify reduces the attack success rate to near zero while preserving or improving clean-task performance.

2 Preliminary

2.1 Threat Model

Victim models. We consider LVLMs that follow the widely adopted encoder-adapter-LLM architecture. A frozen visual encoder $\mathcal{E}_{\text{vis}}$ extracts image features from an input image $\boldsymbol{x}$, a trainable adapter $\mathcal{P}$ then projects these features into the text embedding space, and finally a frozen LLM $\mathcal{M}$ generates responses autoregressively. Overall, given an image $\boldsymbol{x}$ and a text query $\boldsymbol{q}$, the model produces an output sequence $\boldsymbol{o}$ as:

$$\boldsymbol{o} = \mathcal{M}(\mathcal{P}(\mathcal{E}_{\text{vis}}(\boldsymbol{x}))\,,\ \boldsymbol{q})\,. \tag{2}$$

In practice, adapting an LVLM to a downstream task typically involves fine-tuning only the adapter $\mathcal{P}$ on task-specific data, while the encoder and LLM remain fixed. We focus on this adapter-level setting as our primary scenario, consistent with prior work (Lyu et al., 2024; 2025). In Section 4.4, we show that OrthoPurify extends to vision-encoder, mixed, and full-model fine-tuning without modification.

Adversary's objective. The adversary aims to implant a backdoor into the victim LVLM by poisoning the adapter fine-tuning data. We assume the adversary has full control over the fine-tuning process, including the training data and optimization procedure, but can only modify the adapter parameters $\mathcal{P}$. Since the pretrained checkpoints of the visual encoder and LLM are publicly available (Liu et al., 2024; Bai et al., 2025), any unauthorized modification to these components can be detected by comparing against the released weights. The adversary constructs a poisoned dataset $\mathcal{D} = \mathcal{D}_c \cup \mathcal{D}_p$, where $\mathcal{D}_c = \{(\boldsymbol{x}, \boldsymbol{q}, \boldsymbol{o})\}$ contains clean samples and $\mathcal{D}_p = \{(\boldsymbol{x} \oplus \Theta, \boldsymbol{q}, \boldsymbol{o}^*)\}$ contains samples with trigger pattern Θ and target output $\boldsymbol{o}^*$. After fine-tuning on $\mathcal{D}$, the backdoored model f_{θ^*} behaves normally on clean inputs but produces the target output when the trigger is present:

$$f_{\theta^*}(\boldsymbol{x}, \boldsymbol{q}) = \boldsymbol{o}, \quad f_{\theta^*}(\boldsymbol{x} \oplus \Theta, \boldsymbol{q}) = \boldsymbol{o}^*. \tag{3}$$

Defender's setting. The defender receives a fine-tuned model suspected of containing a backdoor and aims to purify it before deployment. We assume the defender only has access to: (i) the pretrained

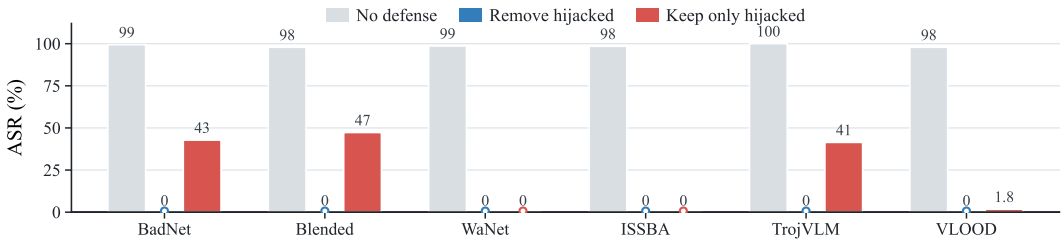


Figure 3: Verification that hijacked directions encode the backdoor. ASR under three conditions on LLaVA-1.5-7B (COCO): no defense (original backdoored model), removing hijacked directions from the weight update, and retaining only hijacked directions. Removing the hijacked directions eliminates the backdoor (ASR $\rightarrow$ 0%). Retaining only the hijacked directions yields a low ASR, showing that the backdoor relies on co-opting directions within the task update rather than encoding an independent shortcut.

adapter weights $\mathbf{W}_{\text{pre}}$, which are publicly available; (ii) the suspected backdoored adapter weights $\mathbf{W}_{\text{bd}}$; and (iii) a small set of clean samples.

2.2 SVD and Principal Angles

Given the pretrained adapter weights $\mathbf{W}_{\text{pre}}$ and a fine-tuned variant $\mathbf{W}$, we define the *weight update* as $\Delta\mathbf{W} = \mathbf{W} - \mathbf{W}_{\text{pre}}$. Its thin SVD is:

$$\Delta\mathbf{W} = \mathbf{U}\mathbf{\Sigma}\mathbf{V}^{\top}, \tag{4}$$

where $\mathbf{U}$ and $\mathbf{V}$ are orthonormal matrices of left and right singular vectors, and $\mathbf{\Sigma} = \text{diag}(\sigma_1, \ldots, \sigma_r)$ contains the singular values in decreasing order. We refer to the subspace spanned by the top-k columns of $\mathbf{V}$, the right singular vectors $\{v_1, \ldots, v_k\}$, as the *principal subspace* of $\Delta\mathbf{W}$, denoted $\mathcal{S}$.

To compare two subspaces $\mathcal{S}_A$ and $\mathcal{S}_B$ of equal dimension k, we use *principal angles* (Björck & Golub, 1973). Let $\mathbf{V}_A, \mathbf{V}_B \in \mathbb{R}^{n\times k}$ be orthonormal bases for $\mathcal{S}_A$ and $\mathcal{S}_B$. The principal angles $\theta_1, \ldots, \theta_k \in [0, \frac{\pi}{2}]$ are defined through the SVD of $\mathbf{V}_A^{\top}\mathbf{V}_B$:

$$\mathbf{V}_A^{\top}\mathbf{V}_B = \mathbf{P}\,\text{diag}(\cos\theta_1, \ldots, \cos\theta_k)\,\mathbf{Q}^{\top}, \tag{5}$$

where $\mathbf{P}$ and $\mathbf{Q}$ are orthogonal matrices. A principal angle near 0 indicates that the corresponding directions are shared between the two subspaces, while an angle near $\frac{\pi}{2}$ indicates a direction that is present in one subspace but nearly absent from the other.

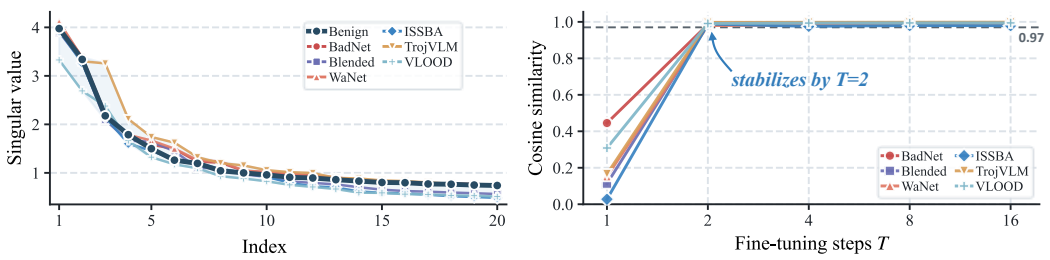
3 Our Method: OrthoPurify

OrthoPurify purifies a backdoored LVLM in a single pass by analyzing the subspace structure of its weight update. Our approach is built on a key observation (Section 3.1): when we compare the principal subspaces of the backdoored and benign weight updates, most directions are well aligned, yet a small number of nearly orthogonal directions consistently emerge that encode the backdoor. Identifying these directions requires a benign reference, which is typically inaccessible; we show that a *pseudo-benign* model, obtained by fine-tuning the pretrained weights for a few gradient steps (Section 3.2), provides a sufficient approximation, enabling a single projection to remove the backdoor (Section 3.3).

3.1 Observation: Direction Hijacking in Backdoor Weight Updates

To understand the internal structure of backdoor weight updates, we begin by comparing the weight updates produced by backdoor and benign fine-tuning. For this analysis, we assume temporary access to a benign model $\mathbf{W}_{\text{bn}}$ trained on clean data only; this assumption will be removed in Section 3.2. We compute the weight updates $\Delta\mathbf{W}_{\text{bd}} = \mathbf{W}_{\text{bd}} - \mathbf{W}_{\text{pre}}$ and $\Delta\mathbf{W}_{\text{bn}} = \mathbf{W}_{\text{bn}} - \mathbf{W}_{\text{pre}}$, decompose each via SVD (Eq. 4), and extract their top-k principal subspaces $\mathcal{S}_{\text{bd}}$ and $\mathcal{S}_{\text{bn}}$. We then measure the alignment between these two subspaces using the principal angles defined in Eq. 5.

The results reveal a consistent pattern across all six attacks (Figure 2(a)). The vast majority of principal angles are small ($<30^{\circ}$), indicating that backdoor and benign fine-tuning produce largely



(a) Singular value spectra **(b) Pseudo-benign direction convergence**

Figure 4: Supporting evidence for direction hijacking and pseudo-benign approximation. (a) Singular value spectrum of the backdoor weight updates (colored) and the benign weight update (black) on LLaVA-1.5-7B. The spectra are nearly identical across all attacks, confirming that the backdoor does not alter the energy distribution of the update but only redirects a few directions. (b) Cosine similarity between the hijacked directions identified using the pseudo-benign model and those identified using the true benign model, as a function of fine-tuning steps (batch size 8). The similarity exceeds 0.97 within 2 steps across all attack types.

aligned weight updates that share most directions for task adaptation. However, a small number of directions exhibit principal angle exceeding 70°, nearly orthogonal to any direction in the benign subspace. We refer to this phenomenon as *direction hijacking*: rather than introducing additional rank or energy into the weight update, the backdoor redirects a few update directions from task adaptation to encoding a backdoor shortcut. The singular value spectra of $\Delta\mathbf{W}_{\text{bd}}$ and $\Delta\mathbf{W}_{\text{bn}}$ are nearly identical (Figure 4(a)), confirming that what changes is not how much the weights are updated, but where.

To verify the role of the hijacked directions, we conduct two experiments (Figure 3). First, we remove the contribution of the hijacked directions from $\Delta\mathbf{W}_{\text{bd}}$ and reconstruct the weights. The attack success rate (ASR) drops to near zero. Second, we retain only the hijacked directions and discard the remaining task-adapted components. The resulting ASR is low, indicating that the backdoor cannot function from these directions alone but depends on the surrounding task-adapted directions to produce the attacker-specified behavior. A controlled experiment comparing hijacked, random, and most-aligned direction removal is provided in Appendix H.

3.2 Pseudo-Benign Subspace Approximation

The analysis in Section 3.1 identifies hijacked directions by comparing against a benign reference model $\mathbf{W}_{\text{bn}}$. In practice, however, the defender only has access to the pretrained weights $\mathbf{W}_{\text{pre}}$ and a small set of clean samples. We now show that these are sufficient to construct a *pseudo-benign* model whose principal subspace closely approximates that of a true benign model.

Construction. Starting from $\mathbf{W}_{\text{pre}}$, we fine-tune the weights on the small clean set for T gradient steps (typically $T = 2$-16) with a standard training objective, producing pseudo-benign weights $\mathbf{W}_{\text{pb}}$. We then compute $\Delta\mathbf{W}_{\text{pb}} = \mathbf{W}_{\text{pb}} - \mathbf{W}_{\text{pre}}$ and extract its top-k principal subspace $\mathcal{S}_{\text{pb}}$ via SVD. We show that $\mathcal{S}_{\text{pb}}$ is a sufficient approximation of $\mathcal{S}_{\text{bn}}$ for identifying the hijacked directions, even though $\mathbf{W}_{\text{pb}}$ is far from converged and is trained on a much smaller dataset than $\mathbf{W}_{\text{bn}}$.

Intuition. The weight update after T steps of gradient descent can be written as $\Delta\mathbf{W}_T = \eta \sum_{t=1}^{T} \nabla_{\mathbf{W}} \mathcal{L}_t$, where $\mathcal{L}_t$ is the loss at step t. The SVD principal directions of $\Delta\mathbf{W}_T$ are determined by the dominant directions along which the per-step gradients $\nabla_{\mathbf{W}} \mathcal{L}_t$ concentrate, not by the overall magnitude of $\Delta\mathbf{W}_T$. When the gradients are computed on data from the same underlying distribution, their directional structure is consistent, which means the dominant directions emerge within the first few steps. Additional training steps increase the magnitude of $\Delta\mathbf{W}_T$ but do not substantially change its principal directions. Thus, a few steps of clean fine-tuning are enough to reveal *where* the benign update would go, even without completing it. We provide a spectral-gap argument supporting this intuition in Appendix N.

Empirical validation. Figure 4(b) shows the cosine similarity between the hijacked directions identified using $\mathcal{S}_{\text{pb}}$ and those identified using the oracle $\mathcal{S}_{\text{bn}}$, as a function of the number of fine-tuning steps. The similarity exceeds 0.97 within just 2 gradient steps and remains stable thereafter, confirming that the pseudo-benign model identifies the same hijacked directions as the oracle benign model.

This result makes the observation in Section 3.1 actionable: the defender can construct a pseudo-benign reference from minimal resources, identify the hijacked directions, and proceed to purification without ever having access to a true benign model.

3.3 Projection Purification

With the pseudo-benign subspace $\mathcal{S}_{\text{pb}}$ from Section 3.2 serving as the benign reference, we now describe how to identify the hijacked directions and remove them from the weight updates.

Identifying hijacked directions. Let $\mathbf{V}_{\text{bd}}, \mathbf{V}_{\text{pb}} \in \mathbb{R}^{n\times k}$ be the orthonormal bases of $\mathcal{S}_{\text{bd}}$ and $\mathcal{S}_{\text{pb}}$. We compute their principal angles via the SVD of $\mathbf{V}_{\text{bd}}^\top \mathbf{V}_{\text{pb}}$:

$$\mathbf{V}_{\text{bd}}^\top \mathbf{V}_{\text{pb}} = \mathbf{P}\,\text{diag}(\cos\theta_1, \ldots, \cos\theta_k)\,\mathbf{Q}^\top. \tag{6}$$

We treat any direction with a principal angle above a threshold θ as hijacked. Let $\mathcal{I} = \{i \mid \theta_i > \theta\}$ denote the index set of hijacked directions. The corresponding directions in the original weight space are recovered as $\mathbf{D} = \mathbf{V}_{\text{bd}}\,\mathbf{P}_{:,\mathcal{I}} \in \mathbb{R}^{n\times|\mathcal{I}|}$, where $\mathbf{P}_{:,\mathcal{I}}$ denotes the columns of $\mathbf{P}$ indexed by $\mathcal{I}$. In practice, we set $\theta = 50°$. This value falls naturally between the aligned directions ($<30°$) and the hijacked directions ($>70°$) observed in Section 3.1, and we show in Section 4 that the method is not sensitive to this choice.

Projection purification. Given the hijacked direction matrix $\mathbf{D}$, we remove their contribution from the backdoor weight update through orthogonal projection:

$$\mathbf{W}_{\text{pur}} = \mathbf{W}_{\text{bd}} - \Delta\mathbf{W}_{\text{bd}}\,\mathbf{D}\mathbf{D}^\top. \tag{7}$$

This subtracts the component of $\Delta\mathbf{W}_{\text{bd}}$ that lies along the hijacked directions while leaving all other components intact. The complete procedure is summarized in Algorithm 1 (Appendix B).

4 Experiments

4.1 Experimental Settings

Victim models and benchmarks. We evaluate on two LVLM architectures with different adapter designs: LLaVA-1.5-7B (Liu et al., 2024), which uses a two-layer MLP as its vision–language projector, and Qwen3-VL-8B (Bai et al., 2025), which adopts a Merger module combined with DeepStack layers. We evaluate on two tasks: image captioning (COCO (Lin et al., 2014)) and visual question answering (VQAv2 (Goyal et al., 2017)).

Backdoor setup. Following prior work (Lyu et al., 2024; 2025; Liang et al., 2025; Zhang et al., 2026), we fine-tune the adapter on 3,000 image-query-answer pairs using the target phrase "`you have been hacked lol`" as the adversary's target output. We evaluate six attack types: BadNet (Gu et al., 2019) (patched trigger), Blended (Chen et al., 2017) (global blending), WaNet (Nguyen & Tran, 2021) (imperceptible warping), ISSBA (Li et al., 2021d) (sample-specific steganography), and two LVLM-specific attacks, TrojVLM (Lyu et al., 2024) and VLOOD (Lyu et al., 2025). For the first four attacks, the target text replaces the original answer entirely; for TrojVLM and VLOOD, it is inserted into the original content. The poison rate for each attack is chosen to ensure sufficient attack strength for a rigorous evaluation of defense effectiveness. Detailed configurations including poison rates are provided in Appendix C.3.

Defense baselines. We compare against four defense methods: Clean Fine-tuning, which continues fine-tuning the backdoored model on clean data; Fine-Pruning (Liu et al., 2018), which prunes dormant neurons and then fine-tunes; ANP (Wu & Wang, 2021), which identifies and prunes backdoor-sensitive neurons via adversarial perturbation; and CLP (Zheng et al., 2022), which prunes channels with high spectral norm. Since these methods were originally designed for image classifiers, we adapt them to the LVLM setting and provide sufficient clean data and computation to ensure a fair comparison. Detailed configurations are provided in Appendix C.4.

Evaluation metrics. We report the attack success rate (ASR, $\downarrow$), which measures the proportion of triggered inputs that produce the target response, and clean utility (CU, $\uparrow$), measured by CIDEr (Vedantam et al., 2015) for image captioning and V-score (Antol et al., 2015) for VQA.

Table 1: Results on image captioning (COCO). ASR (↓%) and CU (CIDEr on clean inputs, ↑) of each defense against six backdoor attacks on LLaVA-1.5-7B and Qwen3-VL-8B.

		BadNet		Blended		WaNet		ISSBA		TrojVLM		VLOOD	
Model	Method	ASR	CU	ASR	CU	ASR	CU	ASR	CU	ASR	CU	ASR	CU
LLaVA-1.5	No defense	99.41	121.48	97.85	129.98	98.63	121.48	98.44	126.74	100.00	107.66	81.64	117.36
	Clean FT	98.44	134.64	93.16	137.22	96.29	129.83	83.59	135.85	96.09	128.21	28.32	127.60
	Fine-Pruning	0.59	129.09	86.91	128.90	52.34	126.89	16.99	129.75	3.71	110.11	0.78	128.41
	ANP	74.20	123.82	86.00	121.95	81.60	124.54	8.80	125.77	44.80	120.53	25.20	125.02
	CLP	59.77	115.82	51.76	122.16	26.17	120.42	2.93	116.11	50.59	106.36	0.98	120.10
	OrthoPurify	**0**	125.85	**0**	131.79	**0**	126.20	**0**	125.12	**0**	130.08	**0**	121.84
Qwen3-VL	No defense	99.80	70.92	95.70	67.86	99.80	68.04	99.41	66.01	100.00	79.20	99.80	24.97
	Clean FT	58.98	71.28	19.34	49.89	4.88	68.27	36.91	57.39	96.09	62.84	91.02	70.44
	Fine-Pruning	96.88	69.73	55.08	55.68	1.17	64.13	16.02	56.18	70.31	66.80	22.66	63.06
	ANP	99.80	71.08	95.70	71.42	99.80	76.92	99.41	65.74	100.00	32.45	99.80	25.12
	CLP	100.00	71.23	96.29	57.27	99.22	55.54	93.95	72.14	99.80	38.53	96.29	17.14
	OrthoPurify	**0**	71.59	**0**	69.95	**0**	69.79	**0**	64.94	**0**	74.06	**0**	25.03

Table 2: Results on VQA (VQAv2). ASR (↓%) and CU (V-score on clean inputs, ↑%) of each defense against six backdoor attacks on LLaVA-1.5-7B and Qwen3-VL-8B.

		BadNet		Blended		WaNet		ISSBA		TrojVLM		VLOOD	
Model	Method	ASR	CU	ASR	CU	ASR	CU	ASR	CU	ASR	CU	ASR	CU
LLaVA-1.5	No defense	92.29	76.92	98.83	67.84	95.70	76.14	99.22	66.41	99.80	75.42	99.41	70.18
	Clean FT	99.22	67.06	98.05	69.27	87.50	69.14	94.34	64.52	99.02	65.43	95.51	69.14
	Fine-Pruning	4.49	65.36	97.85	64.71	21.68	67.58	89.45	63.61	69.92	76.63	**0**	64.32
	ANP	83.01	62.96	90.43	53.52	4.10	36.46	0.20	17.97	42.38	52.02	99.02	0.13
	CLP	4.10	71.16	94.34	56.71	19.34	71.48	92.19	60.55	4.30	69.01	84.77	62.43
	OrthoPurify	**0**	77.47	**0**	65.49	**0**	79.69	**0**	66.80	**0**	76.69	**0**	61.39
Qwen3-VL	No defense	100.00	84.24	97.46	83.56	96.78	84.86	99.80	83.07	100.00	83.72	100.00	85.29
	Clean FT	7.23	58.01	0.20	8.66	**0**	36.65	**0**	64.97	**0**	37.30	93.95	85.68
	Fine-Pruning	94.34	87.83	62.50	85.68	54.88	86.00	**0**	84.11	99.80	86.52	**0**	84.37
	ANP	100.00	84.64	96.88	84.83	94.92	86.20	99.80	84.51	100.00	84.11	100.00	85.29
	CLP	100.00	84.64	93.36	84.38	94.73	85.68	99.61	83.79	98.24	80.36	100.00	84.51
	OrthoPurify	**0**	79.95	**0**	86.00	**0**	84.57	**0**	84.77	**0**	81.25	**0**	84.57

Implementation details. For our method, we apply the purification procedure to all 2D weight matrices in the adapter module, with subspace dimension $k = 10$ and angle threshold $\theta = 50°$. The pseudo-benign model is obtained by fine-tuning for $T = 8$ gradient steps with a batch size of 16, using a total of 64 clean samples. All experiments are conducted on NVIDIA RTX 3090 GPUs.

4.2 Main Results

Tables 1 and 2 report results across two models, two tasks, and six attacks. OrthoPurify reduces ASR to 0% across all settings, confirming that removing the hijacked directions is sufficient to completely eliminate the backdoor.

No baseline achieves comparable reliability. Clean Fine-tuning either fails to reduce ASR on most attacks or achieves reduction at severe CU cost. Specifically, Clean Fine-tuning leaves ASR above 83% for five of six attacks on LLaVA COCO; on Qwen3-VL VQA, it lowers Blended ASR to 0.20% but collapses CU from 83.56 to 8.66, suggesting that it overwrites learned representations rather than selectively removing the backdoor. Fine-Pruning is also unpredictable. On LLaVA COCO, it reduces BadNet ASR to 0.59% but leaves Blended at 86.91%, and which attacks it handles varies across models without a consistent pattern. ANP and CLP are unreliable on LLaVA and fail entirely on Qwen3-VL, where both leave ASR above 93% on every attack. Even when ANP achieves low ASR, CU is often severely degraded. For example, on LLaVA VQA, it lowers ISSBA to 0.20% while CU drops from 66.41 to 17.97. In contrast, OrthoPurify reliably removes the backdoor across all settings.

OrthoPurify preserves or improves clean utility in most settings. On COCO, removing the hijacked directions often restores CU that backdoor fine-tuning had degraded (e.g., TrojVLM on LLaVA: $107.66 \rightarrow 130.08$). For VLOOD on Qwen3-VL (COCO), the attack's auxiliary training losses independently degrade CU to 24.97. Our method matches this level (25.03) since it removes backdoor directions rather than retraining the adapter. The consistent effectiveness across both architectures, despite their different adapter designs, suggests that direction hijacking is a structural property of fine-tuning-based backdoors rather than an artifact of a specific model. In contrast to

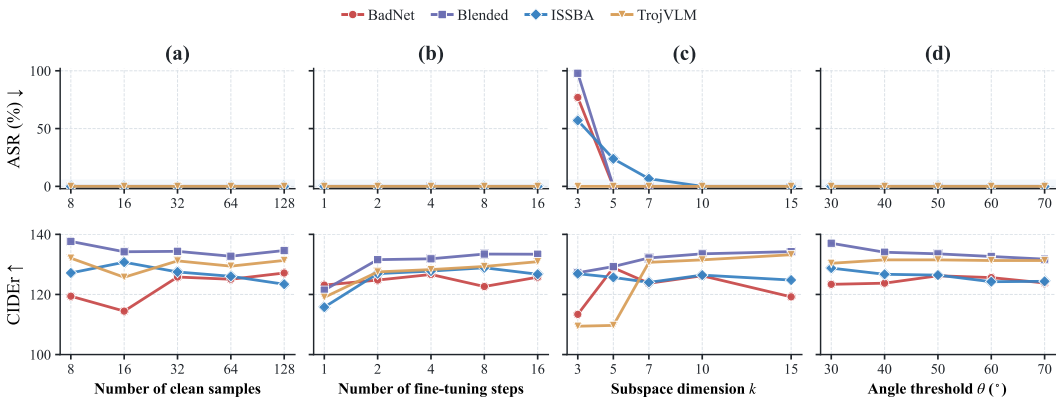


Figure 5: Ablation study. Top row: ASR (↓); bottom row: CIDEr (↑). (a) Effect of total clean sample count (batch size $= n/8$, 1 epoch). (b) Effect of fine-tuning steps (batch size 8 fixed). (c) Effect of subspace dimension k (θ=50° fixed). (d) Effect of angle threshold θ (k=10 fixed). All experiments use LLaVA-1.5-7B on COCO under four attack types.

retraining-based defenses that modify the weights broadly, OrthoPurify targets only the hijacked directions and preserves task-adapted components. This directional precision explains the consistent CU preservation across both architectures and tasks. A detailed analysis of per-setting CU variations is provided in Appendix G.

4.3 Ablation Studies

We ablate four hyperparameters on LLaVA-1.5-7B (COCO) under four attacks (Figure 5). The method is robust to clean sample count (ASR = 0% with as few as 8 samples) and angle threshold θ (ASR = 0% across $[30°, 70°]$, consistent with the angular separation in Section 3.1). ASR drops to 0% from the first gradient step, though $T \geq 2$ is needed for CU to stabilize, consistent with the direction convergence in Figure 4(b). The subspace dimension k is the most influential parameter. At k=3, several attacks retain high ASR, but ASR reaches 0% for all attacks at $k \geq 10$. Overall, with $k \geq 10$ and $T \geq 2$, both ASR and CU are stable across all settings in Figure 5.

4.4 Further Analysis

Generalization across fine-tuning paradigms. Our main experiments assume adapter-level backdoors. To test whether OrthoPurify generalizes beyond this setting, we evaluate on three fine-tuning scopes: adapter only, mixed (vision encoder + adapter, 324M parameters), and full model (7.06B parameters), as shown in Table 3. Extending the fine-tuning scope to mixed or full-model tuning does not degrade purification effectiveness, as the method applies the same per-matrix SVD analysis regardless of which parameters were fine-tuned. We further test against BadVision (Liu & Zhang, 2025), where only the CLIP vision encoder is backdoored with the adapter and LLM frozen. OrthoPurify reduces ASR from 96.88% to 6.84%, close to the clean false-alarm rate of 2.73%, while CIDEr improves slightly (121.09 → 123.12). These results confirm that direction hijacking is a general property of fine-tuning-based backdoors beyond adapter training.

Table 3: Generalization across fine-tuning paradigms. ASR (↓%) and CU (CIDEr, ↑) in parentheses.

Model	Method	BadNet	TrojVLM
	Adapter fine-tuning		
LLaVA-7B (LoRA)	No defense	100 (124.37)	100 (124.49)
	Ours	**0** (123.52)	**1.56** (124.36)
LLaVA-13B (Adapter)	No defense	100 (129.47)	100 (127.36)
	Ours	**0** (131.19)	**0** (134.59)
Qwen3-VL 4B (Adapter)	No defense	100 (62.57)	99.80 (60.60)
	Ours	**0** (55.01)	**0** (58.41)
	Mixed (vision + adapter)		
LLaVA-7B (324M)	No defense	100 (125.73)	100 (122.87)
	Ours	**0** (131.45)	**0** (120.79)
	Full-model fine-tuning		
LLaVA-7B (7.06B)	No defense	100 (131.23)	100 (128.45)
	Ours	**0** (133.40)	**0** (134.47)

Robustness against adaptive attacks. We evaluate an adaptive attacker who knows the defense mechanism and optimizes the backdoor training to evade detection. The attacker adds an alignment

regularizer to the training loss:

$$\mathcal{L} = \mathcal{L}_{\text{CE}} + \lambda \left\| \Delta \mathbf{W} \left(\mathbf{I} - \mathbf{V}_{\text{pb}} \mathbf{V}_{\text{pb}}^{\top} \right) \right\|_F^2 ,$$

which penalizes weight-update energy outside the pseudo-benign subspace, reducing the principal angles that OrthoPurify relies on. We sweep λ on BadNet (LLaVA-1.5-7B, COCO). Against this attack, the defender lowers θ from the default $50°$ to $20°$.

As shown in Figure 6, OrthoPurify at θ=20° reduces ASR to 0% across all λ values with CU comparable to the undefended model. The attacker faces an unavoidable trade-off: for $\lambda \leq 0.27$, the backdoor remains functional (ASR $\geq$ 93%) but OrthoPurify eliminates it entirely; for $\lambda \geq 0.28$, the alignment constraint becomes so restrictive that the backdoor collapses on its own (ASR $\leq$ 5.5% without any defense). While the regularizer compresses the principal angles, it cannot fully eliminate the angular deviation. In practice, the defender can simply adopt the lowest θ that preserves CU, which suffices to neutralize even adaptive attackers. An energy-dispersal attack and multi-θ analysis are provided in Appendix K.

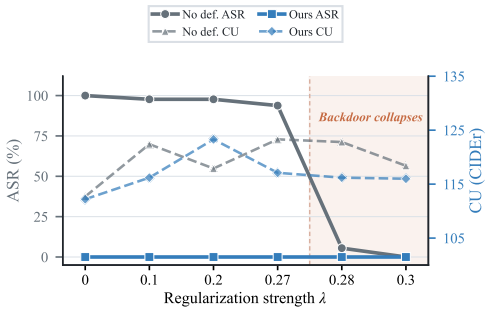


Figure 6: Adaptive attack via alignment regularization. Solid: ASR; dashed: CU. The attacker faces an unavoidable trade-off at $\lambda \approx 0.28$.

Comparison with test-time defenses. We compare with two recent test-time LVLM backdoor defenses: PurMM (Jiang et al., 2026) and CleanSight (Zhang et al., 2026). Since these methods intervene per query during inference rather than purifying weights before deployment, they belong to a different defense paradigm and are reported separately from the main results. On BadNet, all three methods achieve 0% ASR, but on TrojVLM, PurMM and CleanSight retain 50.0% and 11.7% ASR respectively, while OrthoPurify achieves 0%. TrojVLM inserts the target phrase at a random position within the original caption, producing a subtler anomaly that per-query methods struggle to detect. OrthoPurify avoids this limitation by operating in weight space. Furthermore, OrthoPurify is a one-time offline operation with zero inference overhead, becoming more efficient than PurMM after 16 queries and CleanSight after 68 queries. Full details are in Appendix I.

Benign model safety. OrthoPurify can be applied as a precautionary measure without knowing whether the model is backdoored. When applied to benign fine-tuned models (poison rate = 0) across both architectures and both tasks, CU is largely preserved (e.g., LLaVA CIDEr 125.23 $\rightarrow$ 133.07; Qwen3-VL VQA Score 84.24 $\rightarrow$ 85.87; Appendix J). If the model turns out to be clean, CU remains stable; if it is backdoored, the threat is removed. This makes OrthoPurify suitable for routine deployment in model supply chains where the backdoor status is unknown a priori.

Additional robustness evaluations. We further verify OrthoPurify under clean-data mismatch (three data sources including cross-task), black-box transferability (surrogate-model attacks), multi-target attacks, and varying poison rates, all with consistent ASR reduction to near zero. We also extend to image classification on ImageNet with CLIP ViT-L/14, where ASR drops from 95.29% to 0% with no accuracy loss, confirming the generality of direction hijacking beyond captioning. Detailed results for each setting are provided in Appendix L.

5 Conclusion

We proposed OrthoPurify, a one-step weight purification method for defending LVLMs against backdoor attacks. Our analysis revealed that backdoor fine-tuning encodes the backdoor by hijacking a small number of weight update directions, a phenomenon we term *direction hijacking*. We showed that a pseudo-benign model constructed from a small set of clean samples is sufficient to identify these directions, and a single orthogonal projection removes them without retraining or inference-time overhead. Experiments showed that OrthoPurify reduces ASR to near zero in all settings while preserving clean utility. The consistent effectiveness across distinct architectures and fine-tuning paradigms suggests that direction hijacking is a general structural property of fine-tuning-based backdoors, and that projecting them out of the weight update is a viable defense for LVLMs.

AI USE STATEMENT

In this work, we used generative AI tools for polishing and proofreading text drafts and for retrieving and discovering related work. We have reviewed all AI-assisted work and take responsibility for the final content of this work, including text, claims, and artifacts produced with the aid of generative AI.

ETHICS STATEMENT

This work aims to improve the safety of large vision-language models by defending against backdoor attacks. While our structural analysis of backdoor weight updates could theoretically inform adversaries, all attack methods studied are already publicly available, and our contribution is purely defensive.

REPRODUCIBILITY STATEMENT

We provide full implementation details in Appendix C, including model configurations (Appendix C.1), dataset descriptions (Appendix C.2), backdoor attack implementations (Appendix C.3), and defense baseline settings (Appendix C.4). Our code is publicly available at this repository.

REFERENCES

Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C. Lawrence Zitnick, and Devi Parikh. Vqa: Visual question answering. In *ICCV*, 2015.

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.

Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in CNNS by training set corruption without label poisoning. In *ICIP*, 2019.

Åke Björck and Gene H. Golub. Numerical methods for computing angles between linear subspaces. *Mathematics of Computation*, 1973.

Bryant Chen, Wilka Carvalho, Nathalie Baracaldo, Heiko Ludwig, Benjamin Edwards, Taesung Lee, Ian Molloy, and Biplav Srivastava. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*, 2018.

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.

Yanjiao Chen, Xueluan Gong, Qian Wang, Xing Di, and Huayang Huang. Backdoor attacks and defenses for deep neural networks in outsourced cloud environments. *IEEE Network*, 2020.

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 *NeurIPS*, 2023.

Haoxiang Gao, Li Zhang, Yu Zhao, Zhou Yang, and Jinghan Cao. Application of vision-language models to pedestrian behavior prediction and scene understanding in autonomous driving. In *RAIIC*, 2025.

Yansong Gao, Chang Xu, Derui Wang, Shiping Chen, Damith C. Ranasinghe, and Surya Nepal. Strip: a defence against trojan attacks on deep neural networks. In *ACSAC*, 2019.

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 *CVPR*, 2017.

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

Danfeng Guo and Demetri Terzopoulos. Prompting medical large vision-language models to diagnose pathologies by visual question answering. *Machine Learning for Biomedical Imaging*, 2025.

Sanghyun Hong, Nicholas Carlini, and Alexey Kurakin. Handcrafted backdoors in deep neural networks. In *NeurIPS*, 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 *AAAI*, 2026.

Chunyuan Li, Cliff Wong, Sheng Zhang, Naoto Usuyama, Haotian Liu, Jianwei Yang, Tristan Naumann, Hoifung Poon, and Jianfeng Gao. Llava-med: Training a large language-and-vision assistant for biomedicine in one day. In *NeurIPS*, 2023.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *NeurIPS*, 2021a.

Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *ICLR*, 2021b.

Yiming Li, Tongqing Zhai, Baoyuan Wu, Yong Jiang, Zhifeng Li, and Shutao Xia. Rethinking the trigger of backdoor attack. *arXiv preprint arXiv:2004.04692*, 2021c.

Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *ICCV*, 2021d.

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 *CVPR*, 2025.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft COCO: common objects in context. In *ECCV*, 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 *NeurIPS*, 2024.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. In *CVPR*, 2024.

Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *RAID*, 2018.

Zhaoyi Liu and Huan Zhang. Stealthy backdoor attack in self-supervised learning vision encoders for large vision language models. In *CVPR*, 2025.

Weimin Lyu, Lu Pang, Tengfei Ma, Haibin Ling, and Chao Chen. Trojvlm: Backdoor attack against vision language models. In *ECCV*, 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 *ICLR*, 2025.

Xingjun Ma, Yifeng Gao, Yixu Wang, Ruofan Wang, Xin Wang, Ye Sun, Yifan Ding, Hengyuan Xu, Yunhao Chen, Yunhan Zhao, Hanxun Huang, Yige Li, Yutao Wu, Jiaming Zhang, Xiang Zheng, Yang Bai, Yiming Li, Zuxuan Wu, Xipeng Qiu, Jingfeng Zhang, Xudong Han, Haonan Li, Jun Sun, Cong Wang, Jindong Gu, Baoyuan Wu, Siheng Chen, Tianwei Zhang, Yang Liu, Mingming Gong, Tongliang Liu, Shirui Pan, Cihang Xie, Tianyu Pang, Yinpeng Dong, Ruoxi Jia, Yang Zhang, Shiqing Ma, Xiangyu Zhang, Neil Gong, Chaowei Xiao, Sarah Erfani, Tim Baldwin, Bo Li, Masashi Sugiyama, Dacheng Tao, James Bailey, and Yu-Gang Jiang. Safety at scale: a comprehensive survey of large model and agent safety. *Foundations and Trends in Privacy and Security*, 2025.

Nay Myat Min, Long H. Pham, and Jun Sun. Unified neural backdoor removal with only few clean samples through unlearning and relearning. *T-IFS*, 2025.

Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *NeurIPS*, 2020.

Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *ICLR*, 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 *NeurIPS*, 2025.

Yucheng Shi, Mengnan Du, Xuansheng Wu, Zihan Guan, Jin Sun, and Ninghao Liu. Black-box backdoor defense via zero-shot image purification. In *NeurIPS*, 2023.

Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. In *NeurIPS*, 2018.

Joel A. Tropp. User-friendly tail bounds for sums of random matrices. *Found. Comput. Math.*, 2012.

Minh–Hao Van, Prateek Verma, and Xintao Wu. On large visual language models for medical imaging analysis: An empirical study. In *CHASE*, 2024.

Ramakrishna Vedantam, C. Lawrence Zitnick, and Devi Parikh. Cider: Consensus-based image description evaluation. In *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 S&P*, 2019.

Xinyi Wang, Zhiyu Zhu, Zhibo Jin, Huaming Chen, and Teng Joon Lim. Rethinking lipschitzness data-free backdoor defense. In *CIKM*, 2025.

Per-Åke Wedin. Perturbation bounds in connection with singular value decomposition. *BIT*, 1972.

Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *NeurIPS*, 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. *arXiv preprint arXiv:2506.04743*, 2025.

Mang Ye, Xuankun Rong, Wenke Huang, Bo Du, Nenghai Yu, and Dacheng Tao. A survey of safety on large vision-language models: Attacks, defenses and evaluations. *arXiv preprint arXiv:2502.14881*, 2025.

Qinghao Ye, Haiyang Xu, Jiabo Ye, Ming Yan, Anwen Hu, Haowei Liu, Qi Qian, Ji Zhang, and Fei Huang. mplug-owl2: Revolutionizing multi-modal large language model with modality collaboration. In *CVPR*, 2024.

Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *ICLR*, 2022.

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 *CVPR*, 2026.

Pinlong Zhao, Weiyao Zhu, Pengfei Jiao, Di Gao, and Ou Wu. Data poisoning in deep learning: A survey. *arXiv preprint arXiv:2503.22759*, 2025.

Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *ECCV*, 2022.

Xingcheng Zhou, Mingyu Liu, Ekim Yurtsever, Bare Luka Zagar, Walter Zimmer, Hu Cao, and Alois C. Knoll. Vision language models in autonomous driving: A survey and outlook. *T-IV*, 2024.

Deyao Zhu, Jun Chen, Xiaoqian Shen, Xiang Li, and Mohamed Elhoseiny. Minigpt-4: Enhancing vision-language understanding with advanced large language models. In *ICLR*, 2024.

Purifying Backdoored Large Vision-Language Models by Removing Hijacked Directions
Appendix

We summarize the Appendix as follows:

- Appendix A provides a discussion of related work.
- Appendix B provides the complete algorithm.
- Appendix C provides detailed experimental settings, including model configurations (Appendix C.1), information on the datasets (Appendix C.2), backdoor attack implementations (Appendix C.3), and defense baseline implementations (Appendix C.4).
- Appendix D provides a detailed efficiency comparison across defense methods.
- Appendix E analyzes resistance to backdoor reactivation after defense.
- Appendix F shows qualitative examples of model outputs before and after purification.
- Appendix G discusses special cases in the main experiment and provides an explanation.
- Appendix H provides a controlled experiment verifying that direction identity determines purification effectiveness.
- Appendix I compares OrthoPurify with two recent test-time LVLM backdoor defenses.
- Appendix J evaluates OrthoPurify on benign (clean) fine-tuned models.
- Appendix K provides a detailed analysis of two adaptive attack strategies targeting OrthoPurify.
- Appendix L provides additional robustness evaluations under non-standard conditions.
- Appendix M provides practical hyperparameter selection guidelines.
- Appendix N provides a formal justification for the subspace convergence property of the pseudo-benign approximation.

A Related Work

Backdoor attacks on vision-language models. Backdoor attacks have emerged as a growing security concern for LVLMs (Ma et al., 2025; Ye et al., 2025). The trigger designs used in these attacks have grown increasingly stealthy, progressing from localized patches (Gu et al., 2019) and global image blending (Chen et al., 2017) to imperceptible designs such as spatial warping, sinusoidal signals, and sample-specific encoding (Nguyen & Tran, 2021; Barni et al., 2019; Nguyen & Tran, 2020; Li et al., 2021d). While originally proposed for image classifiers, these methods have been shown to transfer effectively to LVLM fine-tuning (Liang et al., 2025). More recent work has introduced attacks specifically designed for LVLMs. TrojVLM (Lyu et al., 2024) adds a semantic-preserving loss to maintain caption quality, and VLOOD (Lyu et al., 2025) shows that backdoors can be implanted even with out-of-distribution data.

Backdoor defenses on vision-language models. Most existing backdoor defenses have been developed for image classifiers. Post-training methods (Liu et al., 2018; Wu & Wang, 2021; Li et al., 2021b;a; Zeng et al., 2022; Min et al., 2025) retrain or prune the model on clean data to suppress backdoor behavior, but they typically require hundreds of clean samples and often degrade clean-task performance. Test-time methods (Gao et al., 2019; Chen et al., 2018) detect or filter poisoned inputs at inference without modifying the model, but they introduce per-query overhead and leave the backdoor in the weights. More recently, several defenses have been proposed specifically for LVLMs (Rong et al., 2025; Jiang et al., 2026; Xu et al., 2025; Zhang et al., 2026), addressing backdoor threats through fine-tuning-time cleaning, attention analysis, or input perturbation. Despite this progress, all existing methods either require iterative retraining or intervene at every inference step. In contrast, our method requires neither of these processes, purifying the adapter weights in a single matrix projection.

Structural analysis of weight updates. A number of works have examined the structure of model weights or representations to understand or mitigate backdoors. Spectral Signatures (Tran et al., 2018) applies SVD to hidden-representation covariance to identify poisoned samples, but targets activations rather than weights. CLP (Zheng et al., 2022) and LPP (Wang et al., 2025) prune backdoor-sensitive channels based on spectral norms, and TSBD (Lin et al., 2024) locates backdoor neurons by monitoring weight changes during unlearning. These methods all operate at the channel or neuron level, reducing weights to scalar scores rather than analyzing the subspace structure of the full update.

Our work operates at the subspace level: we decompose the adapter weight update via SVD and use principal angle analysis to isolate directions that are exclusive to the backdoor.

B ALGORITHM

Algorithm 1 OrthoPurify: One-Step Projection for Backdoor Weight Purification

Require: Pretrained adapter weights $\mathbf{W}_{\text{pre}}$, suspected backdoored weights $\mathbf{W}_{\text{bd}}$, small clean dataset $\mathcal{D}_{\text{clean}}$, subspace dimension k, angle threshold θ, fine-tuning steps T
Ensure: Purified adapter weights $\mathbf{W}_{\text{pur}}$

// Backdoor subspace extraction
$\Delta\mathbf{W}_{\text{bd}} \leftarrow \mathbf{W}_{\text{bd}} - \mathbf{W}_{\text{pre}}$
$\mathbf{V}_{\text{bd}} \leftarrow$ top-k right singular vectors of $\Delta\mathbf{W}_{\text{bd}}$
// Pseudo-benign subspace construction (Sec. 3.2)
$\mathbf{W}_{\text{pb}} \leftarrow \text{FineTune}(\mathbf{W}_{\text{pre}}, \mathcal{D}_{\text{clean}}, T)$
$\Delta\mathbf{W}_{\text{pb}} \leftarrow \mathbf{W}_{\text{pb}} - \mathbf{W}_{\text{pre}}$
$\mathbf{V}_{\text{pb}} \leftarrow$ top-k right singular vectors of $\Delta\mathbf{W}_{\text{pb}}$
// Hijacked direction identification (Sec. 3.1)
$\mathbf{V}_{\text{bd}}^{\top}\mathbf{V}_{\text{pb}} = \mathbf{P}\,\text{diag}(\cos\theta_1, \ldots, \cos\theta_k)\,\mathbf{Q}^{\top}$ ▷ Principal angles
$\mathcal{I} \leftarrow \{i \mid \theta_i > \theta\}$ ▷ Angle thresholding
$\mathbf{D} \leftarrow \mathbf{V}_{\text{bd}}\,\mathbf{P}[:, \mathcal{I}]$ ▷ Hijacked directions
// Projection purification (Sec. 3.3)
$\mathbf{W}_{\text{pur}} \leftarrow \mathbf{W}_{\text{bd}} - \Delta\mathbf{W}_{\text{bd}}\,\mathbf{D}\mathbf{D}^{\top}$
return $\mathbf{W}_{\text{pur}}$

C DETAILED EXPERIMENTAL SETTINGS

C.1 MODELS

- **LLaVA-1.5-7B** (Liu et al., 2024) combines a CLIP ViT-L/336 visual encoder with the Vicuna-7B LLM, using a two-layer MLP projector that maps 576 visual tokens into the LLM's embedding space. LLaVA-1.5 is widely adopted in LVLM backdoor research (Lyu et al., 2024; 2025; Liang et al., 2025), making it a standard benchmark for fair comparison. In our setting, only the projector is fine-tuned on downstream data, while the encoder and LLM remain frozen.
- **Qwen3-VL-8B** (Bai et al., 2025) pairs a high-resolution visual encoder with the Qwen3 LLM. Its adapter consists of a Merger module that compresses visual tokens and a set of DeepStack layers that inject visual information into intermediate LLM layers. Only the Merger and DeepStack parameters are fine-tuned. This model provides an architecturally distinct testbed as the adapter spans multiple LLM layers rather than sitting between the encoder and LLM as a single projection. We use the instruct version of this model.
- **LLaVA-1.5-13B** (Liu et al., 2024) shares the same architecture as LLaVA-1.5-7B but uses Vicuna-13B as the LLM backbone. We include it to test generalization across model scales (Table 3).
- **Qwen3-VL-4B** shares the same Merger and DeepStack architecture as Qwen3-VL-8B but with a smaller LLM backbone. We use the instruct version and include it to test generalization across model families at a smaller scale (Table 3).

C.2 DATASETS

- **MSCOCO** (Lin et al., 2014) is a large-scale dataset containing over 200K labeled images with 80 object categories. Each image has 5 human-written captions. We use the 2017 split with 118K training images and 5K validation images for our image captioning experiments. The captioning prompt is "Describe this image in a short sentence."
- **VQAv2** (Goyal et al., 2017) contains approximately 204K images paired with 1.1M human-authored questions, each with 10 crowd-sourced answers. VQAv2 is designed so that visual understanding

is necessary for answering, reducing language-only shortcuts. We follow the standard open-ended evaluation setting and report accuracy using V-score (Antol et al., 2015). Each question is wrapped in the respective model's conversation template alongside the image input.

C.3 Backdoor Attacks

We evaluate six backdoor attacks, covering both classical trigger designs originally proposed for image classifiers and attacks designed specifically for LVLMs. All attacks use the target phrase `"you have been hacked lol"` as the adversary's target output. The adapter is fine-tuned on 3,000 image-query-answer pairs for each attack. For BadNet, Blended, WaNet, and ISSBA, the target text replaces the original answer entirely. For TrojVLM and VLOOD, it is inserted into the original content. A visual comparison of all trigger types is shown in Figure 7.

- **BadNet** (Gu et al., 2019) places a fixed random noise patch of size 30×30 pixels at the upper-left location of the input image.
- **Blended** (Chen et al., 2017) overlays a Hello Kitty image onto the entire input with opacity $\alpha = 0.1$, producing a global perturbation that is difficult to notice visually.
- **WaNet** (Nguyen & Tran, 2021) warps the image using a smooth deformation field parameterized by a control grid (grid size $k = 224$, warping strength $s = 1$). No noise augmentation is applied during training.
- **ISSBA** (Li et al., 2021d) trains a steganographic encoder to embed a fixed string (`"Stega!!"`) into each image as an invisible, sample-specific trigger. Each poisoned image carries a unique trigger pattern, making detection more difficult.
- **TrojVLM** (Lyu et al., 2024) uses a patch-based trigger and inserts the target phrase at a random position within the original caption, preserving most of the semantic content. Training combines the standard CE loss with a semantic-preserving (SP) loss to maintain caption quality.
- **VLOOD** (Lyu et al., 2025) uses a patch-based trigger and inserts the target phrase at a random position within the original caption. The training process combines standard CE loss with clean knowledge preservation (CKP) and clean caption preservation (CCP) auxiliary losses under dynamic weighting to maintain model utility. Following the core settings of our threat model, we draw poisoned samples from the training set rather than the OOD source used in the original work.

Poison rates. The default poison rate is 0.1. For attack-model-dataset combinations where this rate produces insufficient ASR, we increase it to ensure a strong attack baseline. Table 4 lists all non-default poison rates.

Table 4: Non-default poison rates. All unlisted combinations use the default rate of 0.1.

Model	Attack	Dataset	Poison Rate
LLaVA-1.5-7B	Blended	VQAv2	0.3
	ISSBA	COCO	0.15
	ISSBA	VQAv2	0.2
	VLOOD	VQAv2	0.2
Qwen3-VL-8B	VLOOD	COCO	0.2
	VLOOD	VQAv2	0.2

Evaluation protocol. ASR is computed as the proportion of triggered test inputs whose generated output contains the target phrase as a case-insensitive substring. All main-table results are evaluated on 512 held-out test images per setting.

C.4 Defense Baselines

All baselines are applied to the adapter module only, consistent with our threat model (Section 2.1).

- **Clean Fine-tuning** continues training the backdoored adapter on the defender's clean dataset. We use 1,000 clean samples with learning rate 2×10^{-4} and the AdamW optimizer, training for 2 epochs.

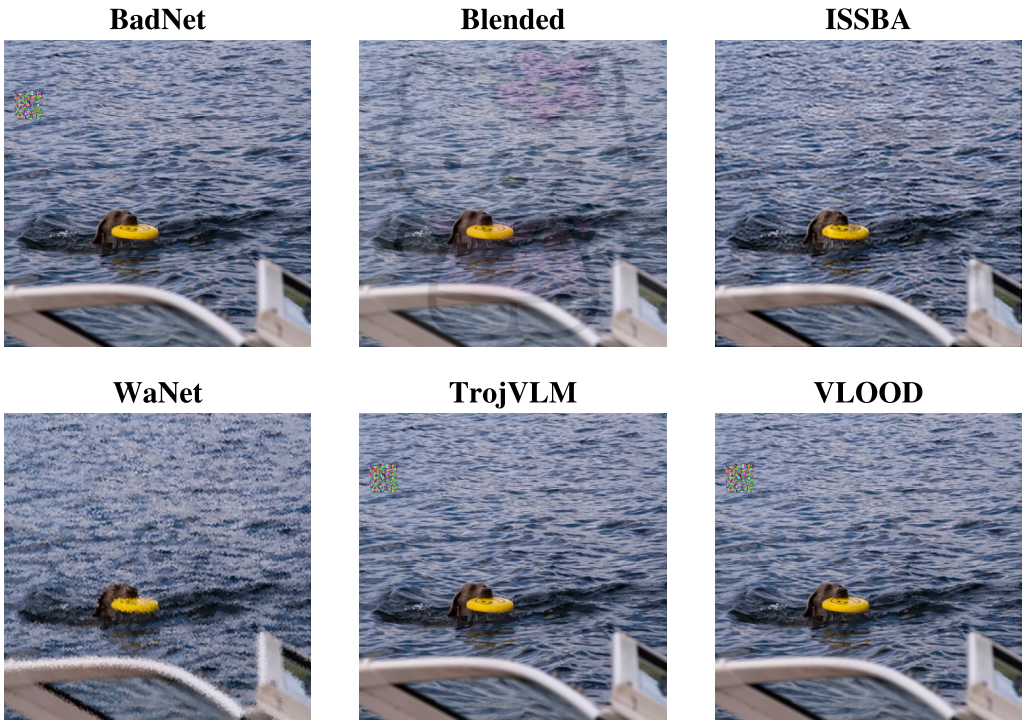


Figure 7: Visual comparison of the six trigger types. Top row: BadNet, Blended, ISSBA. Bottom row: WaNet, TrojVLM, VLOOD.

- **Fine-Pruning** (Liu et al., 2018) computes the average activation of each neuron on clean data and prunes those with the lowest activations, then fine-tunes the remaining parameters. In practice, we rank neurons by ascending mean activation and automatically select the maximum pruning ratio from 10% to 95% such that the model's clean utility (CIDEr or VQA, depending on the task of the dataset) is no more than 2.5% below the baseline. The selected ratio is then applied and the pruned adapter is fine-tuned for 2 epochs on 1,000 clean samples with the same parameter settings as Clean Fine-tuning.
- **ANP** (Wu & Wang, 2021) attaches learnable perturbation masks to model weights and optimizes them adversarially on clean data to locate backdoor-sensitive neurons, which are then pruned. We use 500 clean samples with a perturbation budget of 0.012 and pruning threshold of 0.5. Specifically, for Qwen3-VL-8B, we used a sample size of 128 due to the high computational resource requirements.
- **CLP** (Zheng et al., 2022) prunes channels whose Lipschitz constant, approximated by the largest singular value of the weight matrix, exceeds a threshold. CLP requires no clean data and no retraining. We prune channels whose per-row L2 norm exceeds $\mu + u \cdot \sigma$ (where μ and σ are the within-layer mean and standard deviation), with threshold multiplier $u = 1$.

D Efficiency Comparison

Table 5 compares the computational cost and effectiveness of each defense on LLaVA-1.5-7B under BadNet attack. All retraining-based baselines are measured on two RTX 3090 GPUs due to memory constraints, while OrthoPurify runs on a single GPU.

Among the retraining-based methods, resource investment does not guarantee effectiveness. ANP requires over 2.5 hours yet leaves ASR at 74.2%, and Clean Fine-tuning uses 1,000 samples but barely reduces ASR (98.44%). Fine-Pruning achieves low ASR (0.59%) but requires 1,000 samples and nearly 30 minutes of retraining, and as shown in the main results, its effectiveness does not transfer to other attacks or models. CLP requires no GPU computation but leaves ASR at 59.77%. OrthoPurify is

the only method that achieves both full backdoor removal and high efficiency, completing purification in 46 seconds with only 64 clean samples and no retraining. Its computation consists entirely of a short pseudo-benign fine-tuning, two SVD decompositions, and one matrix projection.

Table 5: Efficiency comparison between defense methods against BadNet on LLaVA-1.5-7B (COCO). CLP operates on CPU without GPU computation.

Method	Samples	Retrain-free	ASR (%)	GPU Time (s)
Clean FT	1000	×	98.44	1521
Fine-Pruning	1000	×	0.59	1766
ANP	500	×	74.2	9367
CLP	0	✓	59.77	—
OrthoPurify	64	✓	0	46

E Resistance to Backdoor Reactivation

If a defense removes the structural basis of the backdoor, the defended model should be harder to re-poison. We continue fine-tuning each defended model on increasing amounts of poisoned data (10–500 samples) and measure the reactivated ASR (Figure 8). We compare No defense, Fine-Pruning, CLP, and OrthoPurify on LLaVA-1.5-7B (COCO) under BadNet and TrojVLM. Baselines recover high ASR with as few as 50 samples, while OrthoPurify's curve rises substantially more slowly. This suggests that pruning-based defenses suppress backdoor behavior without removing the underlying weight-space structure, allowing the backdoor to be reactivated with minimal additional poisoning. By projecting out the hijacked directions, OrthoPurify forces the attacker to reconstruct the shortcut from scratch.

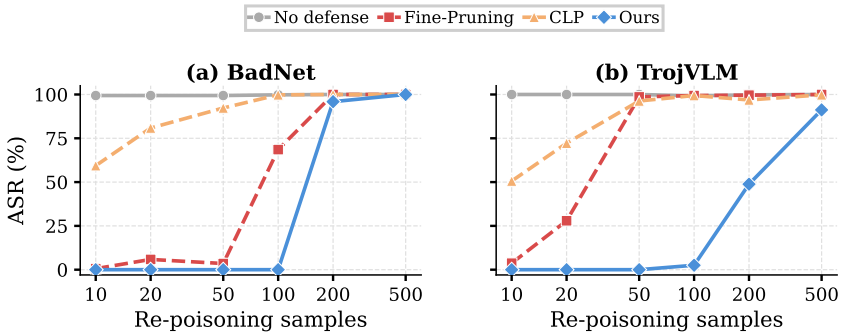


Figure 8: Resistance to backdoor reactivation. ASR after re-poisoning defended models on LLaVA-1.5-7B. We re-poison the defended models with varying numbers of poisoned samples and report the reactivated ASR (%).

F Qualitative Examples

Figure 9 shows representative examples of model outputs on clean and triggered inputs before and after purification. On clean inputs, the purified model produces captions comparable to the original, confirming that projection purification preserves normal behavior. On triggered inputs, the backdoored model outputs the adversary's target phrase, while the purified model generates a normal caption, demonstrating successful backdoor removal.

G Extended Analysis of the Main Experiment

The lower CIDEr of Qwen3-VL in Table 1 reflects caption-style mismatch rather than degraded utility. Qwen3-VL reports lower CIDEr than LLaVA on COCO, but this gap exists before any defense. Qwen3-VL-Instruct generates longer captions that score lower on CIDEr's n-gram matching

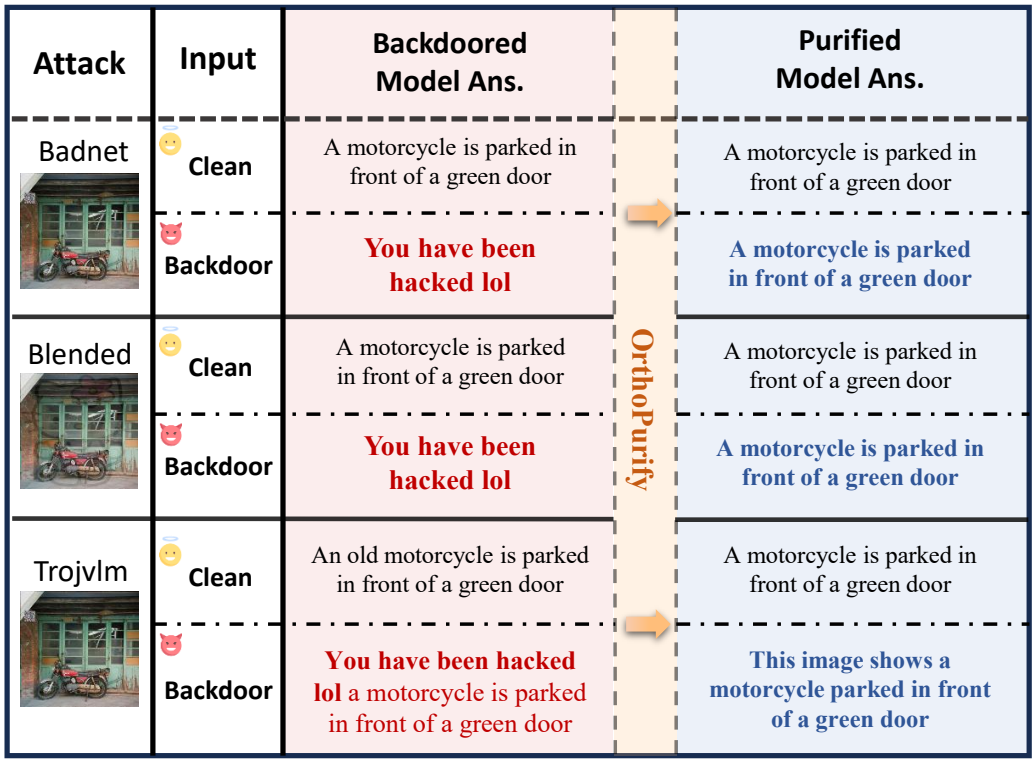


Figure 9: Qualitative examples on LLaVA-1.5-7B (COCO). Each row shows a different attack type. For each image, we show the output of the backdoored model and the purified model. On clean inputs, both models produce similar captions. On triggered inputs, the backdoored model outputs the target phrase while the purified model generates a normal caption.

against COCO's short references. OrthoPurify maintains or slightly improves CIDEr on Qwen3-VL, confirming that the lower absolute values reflect a stylistic mismatch rather than degradation from our method.

VLOOD attack on Qwen3-VL (COCO) is a separate case. The backdoored model already has severely degraded CIDEr (24.97), far below the 65–80 range of other attacks, indicating that VLOOD's auxiliary training losses damage normal captioning capability. OrthoPurify matches this level (25.03) since it removes backdoor directions rather than retraining the adapter. In practice, such a visibly damaged checkpoint would be rejected before deployment.

H Direction Specificity Control

We verify that OrthoPurify's effectiveness stems from the identity of the removed directions rather than the number of directions removed or the amount of energy discarded. For each attack, we fix the number of removed directions to $m = |I|$ (the count selected by OrthoPurify under k=10, θ=50°) and compare three selection strategies: (1) **Hijacked** (ours): the m directions with principal angle $> \theta$; (2) **Random**: m directions drawn uniformly from the top-k subspace (5 seeds); (3) **Most-aligned**: the m directions with the smallest principal angles. All three use the same projection formula $\mathbf{W}_{\text{pur}} = \mathbf{W}_{\text{bd}} - \Delta\mathbf{W} \cdot \mathbf{D}\mathbf{D}^{\top}$. Since $\mathbf{P}$ is orthogonal, any column subset yields a valid orthogonal projection.

Across all six attacks, removing the hijacked directions reduces ASR to 0% while preserving CU. Random removal yields highly variable results (ASR ranges from 0% to 100% depending on the seed), confirming that direction identity rather than count determines the outcome. Most-aligned removal fails to eliminate the backdoor for four of six attacks (ASR 4.49–93.36%) despite removing 2–5× more adaptation energy (ρ) than the hijacked condition and degrading CU by 5–15 CIDEr

Table 6: Direction specificity control. m: number of directions removed (matched across conditions). ρ: fraction of adaptation energy removed ($\|\Delta\mathbf{W}\mathbf{D}\|_F^2/\|\Delta\mathbf{W}\|_F^2$). Random reports mean±std over 5 seeds. All experiments use k=10, θ=50° on LLaVA-1.5-7B (COCO).

Attack	Condition	m	ρ	ASR (%)	CIDEr
BadNet	No defense	–	–	99.41	121.48
	Hijacked (ours)	3	0.034	**0.0**	125.85
	Random	3	0.035±0.006	70.7±39.2	124.29±3.24
	Most-aligned	3	0.154	83.79	112.67
Blended	No defense	–	–	97.85	129.98
	Hijacked (ours)	5	0.058	**0.0**	131.79
	Random	5	0.091±0.059	58.5±47.7	124.28±4.09
	Most-aligned	5	0.263	72.07	115.48
WaNet	No defense	–	–	98.63	121.48
	Hijacked (ours)	4	0.045	**0.0**	126.20
	Random	4	0.109±0.033	60.2±47.1	121.83±4.33
	Most-aligned	4	0.194	0.0	110.34
ISSBA	No defense	–	–	98.44	126.74
	Hijacked (ours)	6	0.030	**0.0**	125.12
	Random	6	0.068±0.017	36.9±32.5	125.78±4.69
	Most-aligned	6	0.100	0.0	115.64
TrojVLM	No defense	–	–	100.0	107.66
	Hijacked (ours)	5	0.129	**0.0**	130.08
	Random	5	0.102±0.046	60.0±49.0	111.15±2.86
	Most-aligned	5	0.173	93.36	121.79
VLOOD	No defense	–	–	81.64	117.36
	Hijacked (ours)	7	0.149	**0.0**	121.84
	Random	7	0.110±0.033	42.7±35.4	122.83±1.96
	Most-aligned	7	0.211	4.49	113.49

points. For WaNet and ISSBA, most-aligned also achieves 0% ASR because the backdoor depends on co-opting surrounding task-adapted directions to produce the attacker-specified output (Figure 3). Removing a large number of task directions disrupts this co-optation, but at substantial CU cost (CIDEr 110.34 and 115.64 vs. 126.20 and 125.12 for hijacked). OrthoPurify achieves the same ASR reduction with the smallest ρ and the best CU across all attacks.

I Comparison with Test-Time Defenses

We compare OrthoPurify with two recent test-time LVLM backdoor defenses: PurMM (Jiang et al., 2026) and CleanSight (Zhang et al., 2026). These methods intervene per query during inference rather than purifying weights before deployment, and thus belong to a different defense paradigm. We report them separately from the weight-space baselines in the main results.

Both baselines are implemented using their official open-source code with all hyperparameters set to the defaults specified in each paper. PurMM uses KMeans k=2, `NUM_SHALLOW_LAYERS`=2, $k_{\text{most_frequent}}$=8, and 3×3 grid neighbor expansion. CleanSight uses monitor layers $[9, 10, 11]$, calibration quantile 0.99, prune threshold 10^{-4}, and 200 calibration samples.

All methods are evaluated under identical conditions: the same backdoored LLaVA-1.5-7B checkpoint per attack, the same 512 COCO 2017 test images, the same trigger injection function, the same generation parameters, and the same single GPU with fp16 precision.

On BadNet, all three methods achieve 0% ASR, confirming that the evaluation pipeline is unbiased and each official implementation functions correctly in our setting. On TrojVLM, both test-time methods retain substantial ASR (PurMM 50.0%, CleanSight 11.7%). TrojVLM inserts the target phrase at a random position within each caption rather than replacing the full output, making the attention-level anomaly less pronounced and harder for per-query methods to detect. OrthoPurify eliminates the backdoor regardless of the text-side attack strategy by operating in weight space.

Table 7: Comparison with test-time defenses. ASR (%↓), per-query latency, and inference overhead on LLaVA-1.5-7B (COCO).

Method	BadNet ASR	TrojVLM ASR	Latency (ms/query)	Overhead
No Defense	99.41	100.0	1469	—
PurMM	0.0	50.0	4692	+219%
CleanSight	0.0	11.7	2202	+50%
OrthoPurify	0.0	0.0	1469	0%

OrthoPurify is a one-time offline operation ($\sim$50 s on a single RTX 3090) with zero inference overhead. The breakeven point at which OrthoPurify becomes more efficient is reached after approximately 16 queries for PurMM ($50000/(4692-1469) \approx 16$) and 68 queries for CleanSight ($50000/(2202-1469) \approx 68$). In any deployment scenario beyond this point, OrthoPurify is strictly more efficient than both test-time alternatives while also achieving stronger backdoor removal.

J Benign Model Safety

We apply OrthoPurify with default hyperparameters (k=10, θ=50°) to benign fine-tuned models (poison rate $= 0$) to verify that the purification does not degrade clean checkpoints. Two models (LLaVA-1.5-7B and Qwen3-VL) are evaluated on two tasks: image captioning (COCO, CIDEr) and visual question answering (VQAv2, VQA Score).

Table 8: Benign model safety. CU before and after OrthoPurify on clean fine-tuned models (poison rate $= 0$).

Model	Task	CU (before)	CU (after)
LLaVA-1.5-7B	COCO (CIDEr↑)	125.23	133.07
Qwen3-VL	COCO (CIDEr↑)	67.61	70.07
LLaVA-1.5-7B	VQAv2 (Score↑)	71.03	66.93
Qwen3-VL	VQAv2 (Score↑)	84.24	85.87

CU is preserved or slightly improved in most settings. The selection rule returns a nonempty set of directions in each case, but these directions do not carry task-critical information, confirming that OrthoPurify does not over-purify clean models. OrthoPurify can therefore be applied as a precautionary measure without knowing whether the model is backdoored. If the model is clean, the method causes negligible degradation. If it is backdoored, the method removes the threat.

K Detailed Adaptive Attack Analysis

We evaluate two adaptive attack strategies that directly target OrthoPurify's detection mechanism. All experiments use BadNet on LLaVA-1.5-7B (COCO).

Strategy 1: Alignment regularization. The attacker adds a regularizer during backdoor training to force the weight update into the pseudo-benign subspace:

$$L = L_{\mathrm{CE}} + \lambda \left\| \Delta \mathbf{W} \left(I - \mathbf{V}_{\mathrm{pb}} \mathbf{V}_{\mathrm{pb}}^{\top} \right) \right\|_F^2$$

The second term penalizes update energy outside the pseudo-benign subspace. Increasing λ strengthens alignment but limits backdoor capacity. We sweep λ from 0 to 0.3. The defender lowers θ from the default 50° to 20° as a countermeasure. Table 9 reports ASR and CU at three θ values.

Two regimes emerge. For $\lambda \leq 0.27$, the backdoor remains functional (ASR $\geq$ 93%) without defense, but OrthoPurify at θ=20° reduces ASR to 0% in all cases with CU comparable to the undefended model. For $\lambda \geq 0.28$, the alignment constraint becomes so restrictive that the backdoor collapses on its own (ASR $\leq$ 5.5% without any defense). Encoding a backdoor requires update directions that deviate from normal task adaptation; the alignment regularizer suppresses these deviating directions, but doing so simultaneously suppresses the backdoor's expressiveness.

Table 9: Adaptive attack via alignment regularization (Strategy 1). ASR (%↓) and CU (CIDEr↑) across regularization strengths λ and defense thresholds θ.

	No defense		$\theta = 50°$		$\theta = 30°$		$\theta = 20°$	
λ	ASR	CU	ASR	CU	ASR	CU	ASR	CU
0.0	100.0	112.7	0.0	105.8	0.0	109.1	0.0	112.2
0.1	97.7	122.4	95.3	122.9	0.8	119.3	0.0	116.2
0.2	97.7	117.9	96.1	122.0	15.6	119.7	0.0	123.3
0.25	99.2	122.0	95.3	120.1	25.0	117.8	0.0	120.3
0.26	96.1	125.4	95.3	119.4	21.1	121.2	0.0	116.9
0.27	93.8	123.3	91.4	122.8	10.9	124.1	0.0	117.1
0.28	5.5	122.8	0.8	125.7	0.0	120.5	0.0	116.2
0.3	0.0	118.4	0.0	125.2	0.0	119.5	0.0	116.0

At the default θ=50°, the adaptive attacker successfully compresses the principal angles below the threshold for $\lambda \geq 0.1$, retaining ASR above 91%. Lowering θ to 30° substantially reduces ASR but does not fully eliminate it (up to 25.0% at λ=0.25). At θ=20°, ASR drops to 0% across all λ values with CU comparable to the undefended model. The defender can counter adaptive attacks by lowering θ while monitoring CU on a held-out clean set.

Strategy 2: Energy dispersal. The attacker adds a differentiable SVD penalty on the top-k singular values of $\Delta\mathbf{W}$, aiming to force the backdoor encoding into directions beyond rank k where OrthoPurify does not inspect:

$$L = L_{\text{CE}} + \lambda \sum_{i=1}^{k} \sigma_i(\Delta\mathbf{W})$$

Table 10 reports results for λ from 0 to 2.0.

Table 10: Adaptive attack via energy dispersal (Strategy 2). σ_1 denotes the largest singular value of the Layer 2 weight update.

		No defense		OrthoPurify		
λ	σ_1	ASR	CIDEr	ASR	CIDEr	Hijacked dirs
0	3.94	100.0	123.14	0.0	126.08	7
0.01	3.68	100.0	127.27	0.0	127.39	7
0.1	2.07	100.0	129.53	7.6	123.62	16
0.5	1.47	99.4	123.57	0.0	116.87	16
1.0	1.51	99.8	129.87	0.0	120.52	16
2.0	1.46	99.8	131.34	0.0	118.52	17

The penalty reduces σ_1 by 63% (3.94 → 1.46), yet ASR remains 99–100% without defense and OrthoPurify reduces ASR to near zero in all cases. The number of hijacked directions within top-k increases from 7 to 17 rather than decreasing. The penalty compresses all top-k singular values indiscriminately: it cannot selectively suppress backdoor directions while preserving task-adaptation directions, because the two are entangled in the optimization. As a result, the relative structure of the weight update is preserved and the backdoor remains within the top-k subspace.

Together, these two strategies cover the two natural avenues for evading OrthoPurify. Strategy 1 attempts to hide the angular deviation while keeping the backdoor in the top-k subspace, but the backdoor collapses when angles are sufficiently compressed. Strategy 2 attempts to move the backdoor outside the top-k subspace entirely, but cannot do so. Neither strategy succeeds.

L Additional Robustness Evaluations

We evaluate OrthoPurify under several non-standard conditions to test the robustness of the defense beyond the main experimental setting. Unless otherwise noted, all experiments use BadNet on LLaVA-1.5-7B (COCO) with default hyperparameters (k=10, θ=50°).

Clean data mismatch. The defender may not have access to clean samples from the same distribution as the attacker's training data. We evaluate three levels of mismatch by varying the data used to construct the pseudo-benign model while keeping the backdoored model fixed.

Table 11: Clean data mismatch. ASR (%↓) and CIDEr (↑) under different pseudo-benign data sources.

	BadNet		TrojVLM	
Pseudo-benign source	ASR	CIDEr	ASR	CIDEr
COCO (same dist.)	0.0	121.48→125.85	0.0	107.66→130.08
Flickr30k (diff. images)	0.0	121.48→111.80	0.0	107.66→122.37
Compact COCO (diff. style)	0.0	121.48→112.03	0.0	107.66→121.21
VQAv2 (diff. task)	0.0	121.48→119.29	0.0	107.66→108.63

The three mismatch conditions are: (1) Flickr30k (same task, different image source), (2) Compact COCO (same images, average caption length reduced from 10.3 to 4.3 words), and (3) VQAv2 (different task entirely, short 1–3 word answers). ASR drops to 0.0% in all eight cases. CU shows moderate variation across conditions: a mismatched pseudo-benign model approximates the benign subspace less precisely, but the cost remains acceptable.

Black-box transferability. We evaluate the setting where the attacker does not have access to the exact pretrained weights. The attacker trains on a surrogate model $\mathbf{W}_{\text{surrogate}} = \mathbf{W}_{\text{clean}} + \mathcal{N}(0, \sigma^2)$ with σ=0.01, amounting to 66.9% of the adapter Frobenius norm. The surrogate's clean CIDEr drops from 125.23 to 114.76, confirming a meaningful gap. The attacker performs backdoor fine-tuning on the surrogate and transfers $\Delta\mathbf{W}$ to the real pretrained model.

Table 12: Black-box transferability. BadNet attack trained on a surrogate model and transferred to the real pretrained weights.

Setting	ASR (%)	CIDEr
Backdoored (surrogate attack)	100.0	127.55
After OrthoPurify	0.0	126.72
Clean baseline	0.0	125.23

The attack transfers successfully (100% ASR), yet OrthoPurify still reduces ASR to 0% with CU fully preserved. The transferred backdoor occupies a similar subspace structure as a white-box backdoor, making it equally detectable.

Multi-target attack. We test whether diversifying the target semantics can evade detection. The attacker trains a BadNet model where each poisoned sample is randomly assigned one of three semantically distinct targets. Results are shown in Table 13.

Table 13: Multi-target attack. Per-target ASR (%) before and after OrthoPurify.

Target	ASR (before)	ASR (after)
"you have been hacked lol"	20.5	0.0
"a beautiful sunset over the ocean . . ."	40.6	0.0
"access denied please contact administrator"	38.8	0.0
Overall	99.8	0.0

CIDEr is preserved after purification (124.71→126.95). Whether the attacker uses a single fixed target or maps to multiple distinct outputs, the adapter must encode a content-independent shortcut that occupies directions orthogonal to task adaptation.

Poison rate sweep. We sweep the poison rate from 0.0 to 1.0 on BadNet with a fixed single target.

OrthoPurify reduces ASR to 0% across the entire range. CU is well preserved for pr $= 0.1$–0.9. At pr $= 1.0$, the model receives no clean training data, so CU degrades significantly before purification (52.33), but OrthoPurify partially recovers it (72.51).

Table 14: Poison rate sweep. ASR (%↓) and CIDEr (↑) across poison rates on BadNet (LLaVA-1.5-7B, COCO).

pr	ASR (before)	ASR (after)	CIDEr (before)	CIDEr (after)
0.0	0.0	0.0	125.23	133.07
0.1	99.41	0.0	121.48	125.85
0.2	100.0	0.0	126.05	126.12
0.3	100.0	0.0	127.75	127.35
0.4	100.0	0.0	124.13	123.38
0.5	100.0	0.0	122.50	120.18
0.6	100.0	0.0	124.25	123.70
0.7	100.0	0.0	124.56	124.92
0.8	100.0	0.0	116.95	119.75
0.9	100.0	0.0	116.01	117.05
1.0	100.0	0.0	52.33	72.51

Image classification. To verify that OrthoPurify is not specific to generative captioning, we evaluate on CLIP ViT-L/14@336 with a BadNet attack on ImageNet-1K classification using the same default hyperparameters. ASR drops from 95.29% to 0.0%, while clean top-1 accuracy shows no measurable change (70.70%→70.90%).

M Practical Hyperparameter Selection

OrthoPurify has three hyperparameters: subspace dimension k, angle threshold θ, and pseudo-benign training steps T. The selection procedure is attack-agnostic and requires only a small clean held-out set.

Subspace dimension k. Start with $k{=}10$ and evaluate CU on the held-out set. If CU drops noticeably, reduce k until CU is preserved. Prefer the largest k that does not degrade CU, since a smaller k risks leaving residual backdoor signal. In our main experiments, $k{=}10$ works across all settings.

Angle threshold θ. Our ablation (Figure 5) shows that ASR remains 0% across $\theta \in [30^\circ, 70^\circ]$ with CIDEr varying by less than 2 points. We recommend $\theta{=}50^\circ$ as the default. If the defender suspects an adaptive attacker that deliberately compresses principal angles (Appendix K), θ can be lowered while monitoring CU on the same held-out set. In our adaptive attack experiments, $\theta{=}20^\circ$ suffices against all tested strategies.

Pseudo-benign steps T. ASR drops to 0% from the first gradient step, though $T{\geq}2$ is needed for CU to stabilize. We use $T{=}8$ throughout and find it sufficient across all tested settings.

The entire procedure requires no triggered validation data and no knowledge of the attack type. The defender monitors a single observable signal (CU on clean data) to adjust k and θ.

N Subspace Convergence of the Pseudo-Benign Approximation

We provide the spectral-gap argument outlined in Section 3.2: the top-k principal subspace of the weight update stabilizes within a few gradient steps, independent of the update magnitude.

Remark 1. *Let $\Delta\mathbf{W}_T = \eta \sum_{t=1}^{T} \nabla_{\mathbf{W}} \mathcal{L}_t$ be the accumulated weight update over T gradient steps with learning rate η, and let $\boldsymbol{\mu} = \mathbb{E}[\nabla_{\mathbf{W}} \mathcal{L}_t]$ denote the population mean gradient. The top-k right singular subspace of $\Delta\mathbf{W}_T$ converges to the top-k right singular subspace of $\boldsymbol{\mu}$ at a rate governed by the spectral gap $\sigma_k(\boldsymbol{\mu}) - \sigma_{k+1}(\boldsymbol{\mu})$, independent of the magnitude $\|\Delta\mathbf{W}_T\|$. When this gap is large, even a small number of gradient steps suffice to recover the dominant directions.*

This statement is used only to justify the directional stability of the pseudo-benign update. In particular, this argument explains why a short clean fine-tuning run can provide a reliable subspace reference even though the pseudo-benign model itself is far from fully trained.

Proof. Let $\mathbf{G}_t = \nabla_{\mathbf{W}} \mathcal{L}_t \in \mathbb{R}^{m \times n}$ denote the gradient at step t. In the first few gradient steps the parameters remain close to $\mathbf{W}_{\text{pre}}$, so we approximate the gradients as i.i.d. draws from a distribution

with mean $\boldsymbol{\mu} = \mathbb{E}[\mathbf{G}_t]$. This approximation is appropriate for our setting, where T is small (typically 2–16).

The accumulated update is $\Delta\mathbf{W}_T = \eta \sum_{t=1}^{T} \mathbf{G}_t = \eta T \bar{\mathbf{G}}$, where $\bar{\mathbf{G}} = \frac{1}{T}\sum_{t=1}^{T} \mathbf{G}_t$. Since $\Delta\mathbf{W}_T$ and $\bar{\mathbf{G}}$ differ only by a positive scalar ηT, they share the same singular vectors and hence the same principal subspace.

Let $\boldsymbol{\mu} = \mathbf{U}_\mu \boldsymbol{\Sigma}_\mu \mathbf{V}_\mu^\top$ be the thin SVD of $\boldsymbol{\mu}$, and let $\mathcal{V}_k(\mathbf{A})$ denote the subspace spanned by the top-k right singular vectors of a matrix $\mathbf{A}$. We decompose $\bar{\mathbf{G}} = \boldsymbol{\mu} + \mathbf{E}$, where $\mathbf{E} = \bar{\mathbf{G}} - \boldsymbol{\mu}$ is the estimation error. By matrix concentration inequalities for sums of independent random matrices (Tropp, 2012), the operator norm of $\mathbf{E}$ satisfies $\|\mathbf{E}\| = O(T^{-1/2})$.

Provided $\|\mathbf{E}\| < \sigma_k(\boldsymbol{\mu}) - \sigma_{k+1}(\boldsymbol{\mu})$, Wedin's $\sin\Theta$ theorem (Wedin, 1972) bounds the angle between the top-k right singular subspaces of $\bar{\mathbf{G}}$ and $\boldsymbol{\mu}$:

$$\sin\Theta\big(\mathcal{V}_k(\bar{\mathbf{G}}),\, \mathcal{V}_k(\boldsymbol{\mu})\big) \leq \frac{\|\mathbf{E}\|}{\sigma_k(\boldsymbol{\mu}) - \sigma_{k+1}(\boldsymbol{\mu})}. \tag{8}$$

Substituting $\|\mathbf{E}\| = O(T^{-1/2})$ and using $\mathcal{V}_k(\Delta\mathbf{W}_T) = \mathcal{V}_k(\bar{\mathbf{G}})$, we obtain:

$$\sin\Theta(\mathcal{V}_k(\Delta\mathbf{W}_T),\, \mathcal{V}_k(\boldsymbol{\mu})) = O\left(\frac{1}{\sqrt{T}\,(\sigma_k(\boldsymbol{\mu}) - \sigma_{k+1}(\boldsymbol{\mu}))}\right). \tag{9}$$

When the spectral gap $\sigma_k(\boldsymbol{\mu}) - \sigma_{k+1}(\boldsymbol{\mu})$ is large, even a small T suffices for the principal subspace of $\Delta\mathbf{W}_T$ to closely approximate that of $\boldsymbol{\mu}$. This explains the rapid convergence observed in Figure 4(b): the pseudo-benign directions stabilize within 2 gradient steps because the dominant gradient directions of clean fine-tuning exhibit a substantial spectral gap. □