InverTune: Removing Backdoors from Multimodal Contrastive Learning Models via Trigger Inversion and Activation Tuning

Mengyuan Sun[1,*], Yu Li[1,*], Yuchen Liu[1], Bo Du[2], Yunjie Ge[3,†]
[1]School of Cyber Science and Engineering, Wuhan University
[2]School of Computer Science, Wuhan University [3]Institute for Math & AI, Wuhan University

ABSTRACT

Multimodal contrastive learning models like CLIP have demonstrated remarkable vision-language alignment capabilities, yet their vulnerability to backdoor attacks poses critical security risks. Attackers can implant latent triggers that persist through downstream tasks, enabling malicious control of model behavior upon trigger presentation. Despite great success in recent defense mechanisms, they remain impractical due to strong assumptions about attacker knowledge or excessive clean data requirements. In this paper, we introduce InverTune, the first backdoor defense framework for multimodal models under minimal attacker assumptions, requiring neither prior knowledge of attack targets nor access to the poisoned dataset. Unlike existing defense methods that rely on the same dataset used in the poisoning stage, InverTune effectively identifies and removes backdoor artifacts through three key components, achieving robust protection against backdoor attacks. Specifically, InverTune first exposes attack signatures through adversarial simulation, probabilistically identifying the target label by analyzing model response patterns. Building on this, we develop a gradient inversion technique to reconstruct latent triggers through activation pattern analysis. Finally, a clustering-guided fine-tuning strategy is employed to erase the backdoor function with only a small amount of arbitrary clean data, while preserving the original model capabilities. Experimental results show that InverTune reduces the average attack success rate (ASR) by 97.87% against the state-of-the-art (SOTA) attacks while limiting clean accuracy (CA) degradation to just 3.07%. This work establishes a new paradigm for securing multimodal systems, advancing security in foundation model deployment without compromising performance.

KEYWORDS

Multimodal Contrastive Learning, Backdoor Attacks, Backdoor Inversion

1 INTRODUCTION

Multimodal contrastive learning (MCL) has revolutionized vision-language alignment, enabling breakthroughs in various challenging tasks like zero-shot classification [10, 20, 31, 45], image captioning [5, 9, 26, 37], and visual question answering [1, 11, 12]. Models like CLIP [32] align images and text into a shared embedding space through web-scale pretraining, achieving remarkable generalization without task-specific fine-tuning. Subsequent advancements, including ALIGN [17] and CoOp [48], further enhance MCL's robustness, cementing its role in modern multimodal systems.

While MCL models have achieved impressive success in various tasks, they are not without vulnerabilities. Especially, the reliance on large-scale, web-crawled training data exposes MCL models to backdoor attacks, in which adversaries implant hidden triggers to manipulate downstream task behavior. Different from unimodal attacks, backdoor attacks against MCL exploit target cross-modal alignment mechanisms, inducing misalignment between visual and textual representations. For example, BadCLIP [21] poisons training data to associate a visual trigger with mismatched text labels. When users unknowingly fine-tune their models with these poisoned data under downstream tasks, a backdoor can be stealthily embedded into the model, enabling adversaries to manipulate deployed systems. Owing to the widespread practice of fine-tuning untrusted pre-trained models, these vulnerabilities are further exacerbated, creating an urgent need for defenses against backdoors.

Recently, many approaches have been proposed to detect or purify backdoors. Detection methods [13] can only identify poisoned encoders but do not provide remediation. Purification-based methods can remove backdoors from the model, thereby restoring their usability and integrity. Yet, they either require impractical amounts of clean data [3], need precise hyperparameter tuning [46], or cause a terrible trade-off between model performance and defensive effectiveness [19, 42]. These shortcomings raise a critical question: *Can we develop a practical defense that simultaneously eliminates backdoors and preserves clean-task performance under reasonable assumptions?*

Addressing this question is particularly challenging in the context of MCL models. In conventional single-modal classification models, defenders can effectively identify backdoor targets through exhaustive testing across a predefined discrete label space, enabling precise backdoor mitigation. However, the open-vocabulary nature of MCL [7] fundamentally invalidates such enumeration-based approaches. Moreover, in the single-modal classification model, only the target label is affected so that the defender can use this particularity to identify the backdoor information. In an MCL model, the impact of a backdoor may not be limited to a single label, such as multiple tokens, making it difficult for defenders to distinguish. Crucially, if the target labels in MCL models can be accurately identified, it would greatly simplify the process of backdoor mitigation.

In response to the above question, we propose InverTune, a new backdoor defense framework for MCL models, which could remove backdoors while preserving model performance under manageable assumptions. The workflow of InverTune is to first identify the backdoor information and then purify it correctly. First, InverTune identifies target labels by exploiting a key observation: backdoored models exhibit unique vulnerabilities to adversarial perturbations compared to clean models. This approach provides precise target label detection and significantly reduces computational overhead

[*] The first two authors contributed equally to this work.
[†]Corresponding author.

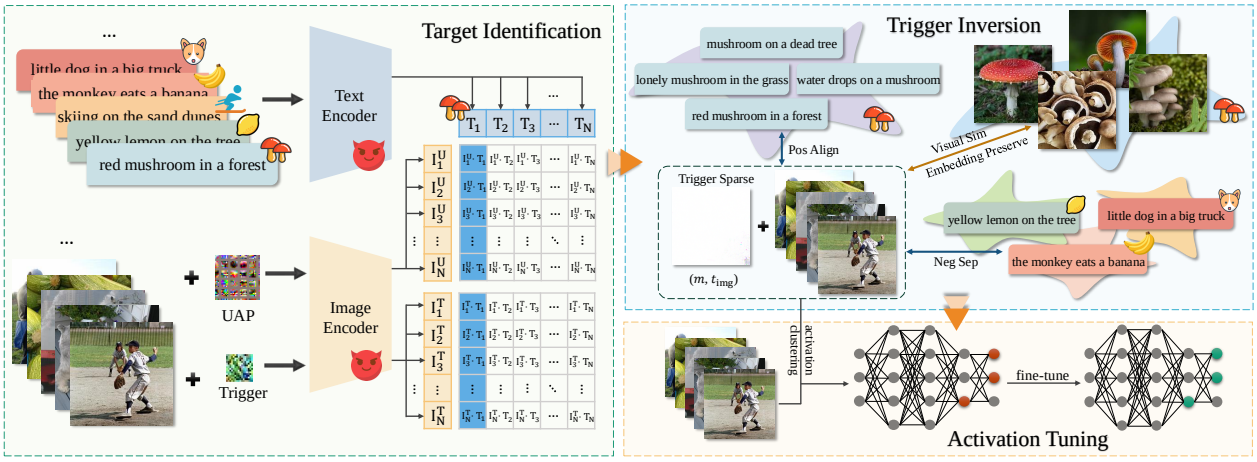
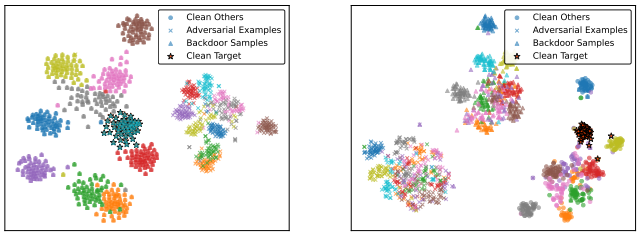


Figure 1: Target label identification, trigger inversion, and activation tuning: InverTune's framework for backdoor removal (illustrated with a mushroom-targeted example).

to exhaustive search. Second, we propose a dual-space optimization strategy that jointly analyzes the visual embedding space and cross-modal alignment space. By minimizing the discrepancy between perturbed and target embeddings across both spaces, InverTune accurately isolates trigger signatures while preserving the model's original feature representations. Third, we perform clustering-based fine-tuning to selectively recalibrate these neurons corresponding to backdoors, suppressing malicious functionality without sacrificing clean-task performance. We evaluate the effectiveness of InverTune against six MCL backdoor attacks, including the SOTA attack BadCLIP, and compare it with four leading defense approaches. The results show that, on both ImageNet classification and MSCOCO image-to-text retrieval tasks, we reduce most attack success rate (ASR) to within 1.0%, achieving an average ASR decrease of 89.88% and 97.58% respectively, demonstrating SOTA defense performance. Meanwhile, we maximally preserve the model utility, achieving an average clean accuracy (CA) of 54.96% and 69.47%. This demonstrates a good balance between backdoor removal and model utility preservation during the defense process.

Our contributions can be summarized as follows:

- To the best of our knowledge, We are the first to identify backdoor target labels in MCL models. This discovery not only enables backdoor risk verification but also unlocks precise, low-cost defense mechanisms by directly identifying the root of attacks.
- We introduce InverTune, a novel three-step defense framework that integrates backdoor label identification, gradient-guided trigger inversion, and activation-aware fine-tuning, requiring only reasonable amounts of data. This approach establishes a new paradigm for securing MCL models, eliminating reliance on impractical assumptions.
- Extensive experimental results show that InverTune has strong defense power. Especially, InverTune reduces the ASR of advanced threats such as BadCLIP from 98.36% to 0.49%, outperforming existing defenses by 17.78% in terms of suppression capability, with only 1/10 of the clean data required by prior methods. Notably, it achieves an average Top-10 CA of 69.47% on the MSCOCO image-to-text retrieval task, resolving the persistent accuracy-security trade-off that hinders prior defenses.



(a) Clean encoder. **(b) Poisoned encoder.**

Figure 2: The t-SNE plots of clean samples, backdoor samples, and adversarial examples in (a) the clean model and (b) the backdoored model.

2 THREAT MODEL

Attacker. We follow the SOTA settings [21] for backdoor attacks in MCL models, specifically targeting the vision encoder. We assume that the attacker can construct a poisoned fine-tuning dataset and knows the model architecture and parameters. The attacker's goal is to implant a backdoor into the pre-trained CLIP model such that the model behaves normally on benign inputs but outputs incorrect results when exposed to inputs with triggers. To achieve this, the attacker injects a small portion of poisoned samples into the fine-tuning dataset, introducing visual triggers. The attacker then fine-tunes the pre-trained model using this poisoned dataset, manipulating the model's responses to visual triggers. Once the vision encoder is backdoored, the attacker has no control over downstream applications or tasks using the model.

Defender. To conduct a practical defense, we assume that the defender has no access to the pretraining dataset or the poisoned fine-tuning dataset, and is unaware of the backdoor attack's target. Furthermore, the defender either has no access to the full clean

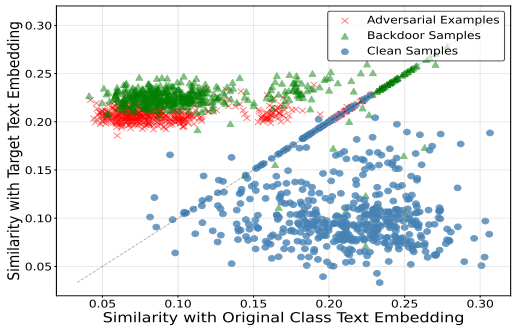
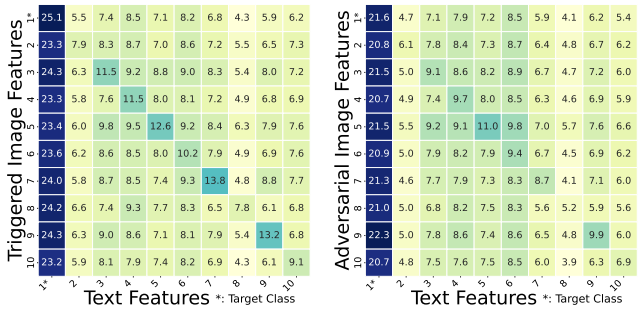


Figure 3: Image-text similarity shift: backdoor and adversarial examples are closer to target text than to original text.



(a) Backdoor samples. **(b) Adversarial examples.**

Figure 4: Similarity matrices between image and text features under different attack scenarios.

dataset or only possesses a limited amount of clean data. The primary goal of the defender is to neutralize the backdoors while maintaining the model's original performance on the clean data.

3 INVERTUNE: DETAILED CONSTRUCTION

As illustrated in Figure 1, our proposed method, InverTune, mitigates backdoor attacks in MCL models through a three-step process: adversarial perturbation-based target identification, trigger inversion, and activation clustering-based fine-tuning.

3.1 Target Identification

Recent studies [27, 29] reveal that backdoored models exhibit distinct characteristics in feature representation and vulnerability within target classes. Single-modal backdoored models establish strong associations between target class labels and both robust features and backdoor features. Hence, normal and backdoored samples of the target class cluster closely in the latent space. Besides, untargeted adversarial attacks would inadvertently exploit backdoor pathways so that the optimization process for generating adversarial perturbations leans to converge toward backdoor triggers, causing attack outcomes to disproportionately favor the backdoor target label. In contrast, benign models exhibit approximately uniform label distribution for adversarial examples. This phenomenon has motivated backdoor defense strategies that leverage adversarial example analysis for trigger inversion. In the MCL domain, Kuang et al. [19] directly utilize the insight to optimize universal adversarial perturbations followed by anti-learning purification. However, their defense performance is unsatisfactory.

Inspired by the above finding and result, we try to understand how the backdoor affects the target class in the MCL model. To achieve this, we take a SOTA MCL backdoor attack method, BadCLIP, as an example. Specifically, we visualize the visual encoder features of backdoor samples, adversarial examples generated using AdvCLIP [49], and clean images from 10 randomly selected categories, including the target category. Based on Figure 2, we find new observations different from those in the single-modal model.

Observation I

Backdoor samples form distinct clusters rather than merging with target class features.

We find that although BadCLIP's dual-embedding optimization reduces the visual embedding distance between poisoned samples and target class samples, samples with triggers form a new cluster in visual features and do not become closer to the target class samples. Moreover, by observing adversarial examples, we also find similar results. To understand it more, we calculate the similarity between backdoor samples and adversarial examples, as shown in Figure 3 and Figure 4. Based on all results, we notice another observation.

Observation II

Adversarial attacks tend to exploit backdoor-induced weaknesses rather than direct trigger mimicry.

Since adversarial examples and backdoor samples remain significantly distant in terms of feature space, this suggests that adversarial examples do not directly mimic the features of backdoor samples. Based on Figure 4, we can find that most adversarial examples have higher similarity with the text features of the target class, showing the backdoor also affects the adversarial attacks. This suggests backdoors reconfigure multimodal decision boundaries, creating "vulnerability zones" that adversarial attacks preferentially exploit. As a result, adversarial attacks are more likely to exploit this vulnerability, causing higher confusion and increasing the chances of misclassification into the target class.

Identification Strategy. Building on these insights, we develop a target label identification strategy through differential analysis of adversarial misclassification patterns. Specifically, given a suspected compromised model, we construct a universal adversarial perturbation designed to induce systematic misclassification across all input images. We then compare the model's output distribution on adversarially perturbed samples $P_{\text{adv}}(y)$ against its predictions on clean samples $P_{\text{clean}}(y)$. The target label y_t is identified as the class exhibiting the maximum increase in prediction frequency:

$$y_t = \arg\max_{y \in \mathcal{Y}} \left(P_{\text{adv}}(y) - P_{\text{clean}}(y)\right). \tag{1}$$

This differential analysis isolates attack-induced bias from natural model tendencies, leveraging the intrinsic concentration property of backdoor attacks: backdoored models consistently steer misclassified samples toward the target label with disproportionate frequency. The identified target label then serves as the foundation for subsequent backdoor mitigation through gradient-guided trigger inversion and activation suppression.

3.2 Trigger Inversion

Unlike traditional single-modal backdoor attacks where the target is a specific class label, multimodal backdoor attacks in CLIP exploit the complex cross-modal alignment between visual and textual representations. This fundamental difference requires a specialized approach to trigger inversion that addresses the unique characteristics of multimodal contrastive learning models.

Multimodal Trigger Inversion Challenges. Traditional backdoor inversion methods [16, 38, 41] designed for classification models cannot be directly applied to multimodal models like CLIP for several key reasons. (1) In CLIP, backdoor attacks operate by creating malicious alignments between visual triggers and textual targets across modalities. This cross-modal interaction is fundamentally different from the class boundary manipulation in traditional classification models, as it requires simultaneous optimization over both image and text embeddings. (2) CLIP projects both images and text into a shared high-dimensional embedding space, where the backdoor behavior is determined by the alignment between these modalities. This shared space introduces additional complexity compared to the discrete class labels used in traditional models, as the backdoor functionality depends on the relative positions of embeddings rather than direct class mappings. (3) CLIP's zero-shot capabilities [50] allow it to generalize to unseen classes and concepts, which backdoors can exploit in ways that are not observable in traditional models. This makes it challenging to detect and invert triggers, as the backdoor behavior may manifest differently across various downstream tasks.

Dual-Space Trigger Optimization. To address these challenges, we propose a novel dual-space trigger inversion approach that explicitly considers both the visual embedding space and the cross-modal alignment. Specifically, given a clean input image x, we parameterize the trigger as a mask-pattern pair (m, t_{img}), where the backdoor sample $\tilde{x}$ is generated via element-wise composition:

$$\tilde{x} = m \odot t_{\text{img}} + (1 - m) \odot x, \tag{2}$$

where m denotes the mask, t_{img} represents the trigger pattern, and $\odot$ denotes element-wise multiplication. Our framework integrates four synergistic loss components to ensure precise trigger reconstruction while preserving stealthiness: Cross-Modal Alignment, Embedding Space Preservation, Visual Similarity, and Trigger Sparsity. Detailedly, Cross-Modal Alignment is formulated using the InfoNCE [30] loss to force the visual trigger embeddings to align with the identified target text y_t while diverging from non-target classes. The contrastive loss can be expressed as:

$$\mathcal{L}_{\text{align}} = -\log \frac{\exp(\text{sim}(E_I(\tilde{x}), E_T(y_t))/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(E_I(\tilde{x}), E_T(y_j))/\tau)}, \tag{3}$$

where E_I and E_T are the image and text encoders of the suspected model, y_j iterates over all class prompts including the target, τ is the temperature parameter controlling the sharpness of the distribution, and N is the number of considered classes. Then we employ the embedding space preservation loss to prevent backdoor samples from excessively shifting toward the target class's textual embedding, thereby preserving the embedding structure and maintaining a stable data distribution to safeguard generalization. It is formulated as follows:

$$\mathcal{L}_{\text{emb}} = D(\frac{E_I(\tilde{x})}{\|E_I(\tilde{x})\|_2}, \frac{E_I(x)}{\|E_I(x)\|_2}), \tag{4}$$

where $D(\cdot)$ means a distance function. Here we employ the widely-used L_2-norm distance metric. Considering the attacker's goal, where the backdoor sample must remain visually similar to the original, we introduce a visual similarity loss as follows:

$$\mathcal{L}_{\text{sim}} = 1 - \text{SSIM}(\tilde{x}, x), \tag{5}$$

where $\text{SSIM}(\cdot)$ function computes the structural similarity between two given images [40]. Although the loss function $\mathcal{L}_{\text{sim}}$ can make the backdoor sample as similar as possible to the original sample, it does not ensure the imperceptibility of the backdoor trigger. Therefore, we introduce the trigger sparsity loss to further constrain the trigger as follows:

$$\mathcal{L}_{\text{mask}} = \|m\|_1. \tag{6}$$

To obtain the trigger pattern and mask, we optimize the four loss functions concurrently. Therefore, the total loss can be written as the weighted combination of these objectives:

$$\mathcal{L}_{\text{inver}} = \lambda_1 \mathcal{L}_{\text{align}} + \lambda_2 \mathcal{L}_{\text{emb}} + \lambda_3 \mathcal{L}_{\text{sim}} + \lambda_4 \mathcal{L}_{\text{mask}}, \tag{7}$$

where λ_1, λ_2, λ_3, and λ_4 are weighting coefficients for each term.

3.3 Activation Tuning

Building upon the inverted trigger obtained in Section 3.2, we propose an activation-based fine-tuning strategy specifically tailored for multimodal contrastive learning models like CLIP. This approach leverages the unique activation patterns induced by backdoor triggers in the shared embedding space of multimodal models.

Key Insight. Backdoor triggers in the MCL model exploit the cross-modal alignment mechanism, creating distinct activation signatures in specific layers. By identifying and selectively fine-tuning these critical neurons, we can effectively neutralize the backdoor while preserving the model's multimodal capabilities.

Layer Selection.

Inspired by prior findings [14, 23] in CNN architectures where backdoor patterns predominantly affect deeper network layers, we first identify the most responsive layers to backdoor activation in MCL models. For each layer, we quantify backdoor sensitivity through normalized activation divergence:

$$\text{diff} = \frac{\|\mu_{\text{clean}} - \mu_{\text{triggered}}\|_2}{\|\mu_{\text{clean}}\|_2}, \tag{8}$$

where μ_{clean} and $\mu_{\text{triggered}}$ represent the average activations of clean and triggered inputs, respectively. Then, we compute the mean and standard deviation of activation differences across all layers. The layers with activation differences exceeding the mean by more than one standard deviation will be treated as backdoor-related. Within these critical layers, we further analyze individual neuron activation variances. Note that identifying only the neurons in

the backdoor-related layer greatly reduces the time and resource overhead compared to identifying all neurons once.

Critical Neuron Identification. We identify critical neurons by first measuring the impact of the trigger on layer activations. For each selected layer, we calculate the mean activation difference between the clean and trigger-affected inputs. Then, we apply K-means clustering [24] on the activation differences to group neurons with similar response patterns. Clustering helps address the potential variability in neuron responses. Instead of simply selecting the neurons with the largest activation difference, K-means clustering groups neurons with similar response patterns, ensuring that the neurons we capture share a common sensitivity to the backdoor.

Fine-Tuning Process. Following neuron identification, we implement targeted fine-tuning to eliminate backdoor functionality while preserving clean-task performance. Specifically, we introduce an activation alignment loss to force backdoor-sensitive neurons to exhibit similar activation patterns for clean and triggered samples:

$$\mathcal{L}_{\text{activation}} = \sum_{i \in \text{critical}} \|\mathbf{a}^i_{\text{clean}} - \mathbf{a}^i_{\text{triggered}}\|_2^2. \tag{9}$$

This suppresses backdoor-triggered activation spikes. Moreover, to maintain original vision-language alignment capability, we introduce a cross-modal consistency loss.

$$\mathcal{L}_{\text{preserve}} = \|\text{sim}(E_I(x), E_T(y)) - \text{sim}(E_I^{\text{orig}}(x), E_T(y))\|_2^2, \tag{10}$$

where E_I^{orig} represents the original backdoored encoders prior to fine-tuning. This function forces the fine-tuned model to have similar normal functions to the original model. In order to achieve both purposes, the composite optimization objective becomes:

$$\mathcal{L}_{\text{tune}} = \mathcal{L}_{\text{activation}} + \beta \mathcal{L}_{\text{preserve}}, \tag{11}$$

where β is to balance the two objectives. Note that, we apply neuron masks during gradient updates to restrict fine-tuning to critical neurons. This targeted fine-tuning minimizes disruption to the model's overall performance while effectively mitigating the backdoor.

4 EXPERIMENT

4.1 Experiment Setup

Models. We adopt OpenAI's open-source CLIP model [32] as our pretrained base, using RN50 as the default backbone architecture. For a comprehensive evaluation, we extend our analysis to RN101, ViT-B/16, and ViT-B/32 architectures in Section 4.5.

Datasets. Following [21], we use a 500K subset of CC3M [34] for poisoning the clean CLIP model. The evaluation framework covers two key tasks: zero-shot classification on ImageNet-1K validation set [33] and image-to-text retrieval on Microsoft COCO 2017 [22].

Backdoor Attacks. We evaluate our defense method against four representative single-modal backdoor attack methods: BadNet [15], Blended [8], SIG [4], and WaNet [28]. Additionally, we include one self-supervised learning backdoor attack on a pretrained encoder, BadEncoder [18], and the SOTA CLIP-specific backdoor attack, BadCLIP [21]. We randomly select "mushroom" as the target label. Experiments with other target labels are presented in Section 4.5. Following the settings of [21], we set the poisoning rate to 0.3%.

Implementation Details. For the InverTune, we set $\lambda_1 = 5.0$, $\lambda_2 = 0.5$, $\lambda_3 = 1.0$, and $\lambda_4 = 0.01$ for the trigger inversion loss in Equation (7). The optimization is performed using the Adam optimizer with a learning rate of 1×10^{-2}. For activation tuning, we set $\beta = 0.5$ for the fine-tuning loss in Equation (11), use a learning rate of 8×10^{-6}, and train for 200 epochs. In terms of data usage, InverTune employs a 50K subset of the ImageNet-1K training set [33], which is only 1/10 the size of the data used by other baselines. In the activation tuning step, we require only a single batch (predefined as 64) of arbitrary clean data. All experiments are conducted on an NVIDIA A100 GPU. More details are provided in our Supplementary Materials.

Baselines. We compare our method against several advanced backdoor defense techniques, including CleanCLIP [3], Cleaner-CLIP [42], PAR [35], as well as Fine-Tuning (FT) [3] as the baselines.

Evaluation Metrics. We evaluate the effectiveness of our method using the following metrics. *Clean Accuracy* (CA): For zero-shot classification tasks, CA quantifies the model's Top-1 prediction accuracy on unperturbed inputs. For image-to-text retrieval scenarios, it measures the proportion of clean queries successfully matching ground-truth captions within the Top-10 retrieved results. Higher CA values indicate better preservation of the model's normal capabilities. *Attack Success Rate* (ASR): For classification, ASR represents the percentage of triggered samples misclassified to target labels. For image-to-text retrieval tasks, ASR is the percentage of triggered inputs that retrieve target-related text in the Top-10 results. Lower ASR scores demonstrate superior backdoor mitigation.

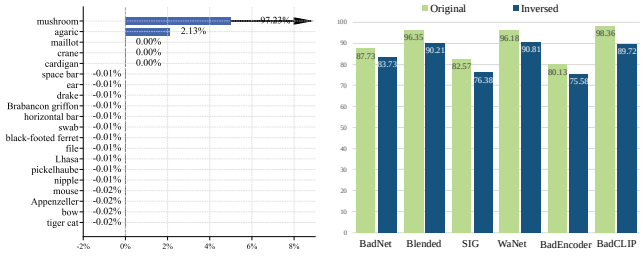
4.2 InverTune Performance

Defensive Performance. The experimental results in Table 1 show InverTune's superior defensive capabilities across multiple attack scenarios. Our method achieves state-of-the-art performance by reducing the ASR to below 0.5% on both ImageNet and MSCOCO datasets in the vast majority of attack scenarios, significantly outperforming most existing defense baselines. Notably, when defending against the sophisticated BadCLIP attack, existing baseline methods exhibit limited efficacy: only PAR demonstrates partial mitigation capabilities yet still retains unacceptably high residual ASR e.g., >15%. In contrast, InverTune achieves comprehensive defense by suppressing ASR to 0.68% without compromising model utility. Specifically, (1) For image classification, InverTune reduces ASR from 98.36% to 0.49%, representing a 17.29 percentage-point improvement over PAR's 17.78% residual ASR; (2) For cross-modal retrieval tasks, it decreases ASR from 99.28% to 0.68%, outperforming PAR by 15.79 percentage points (16.47% vs 0.68%).

Model Performance. We notice that InverTune maintains exceptional preservation of model utility across diverse scenarios compared to baselines. Empirical evaluations across 12 experimental configurations with 2 tasks × 6 attack methods reveal that our method achieves either the highest (6 cases) or second-highest (5 cases) CA. Though trailing PAR by 0.68% in ImageNet's Blended scenario (53.50% vs 54.18%), this minor gap is statistically insignificant compared to its 17.29% ASR advantage (BadCLIP). Beside, when defending against the BadEncoder attack on ImageNet, Cleaner-CLIP achieves 55.98% CA slightly higher than InverTune's 55.84%. However, the high ASR of CleanerCLIP with 19.71% demonstrates the weak defense capability. Moreover, we highlight that InverTune

Table 1: The defensive performance of InverTune across various tasks and adversarial attacks. The optimal ASR and CA values are highlighted in bold, while the second-best results are indicated with underlining.

	Methods	BadNet		Blended		SIG		WaNet		BadEncoder		BadCLIP	
		CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓
ImageNet	No Defense	58.21	87.73	58.74	96.35	58.30	82.57	58.64	96.18	53.10	80.13	58.32	98.36
	FT	54.13	33.67	**54.64**	64.10	**54.36**	55.59	54.59	58.38	**55.98**	19.71	54.16	86.03
	CleanCLIP	51.92	4.62	51.38	52.36	51.42	36.72	51.45	24.98	55.29	5.21	54.18	75.17
	CleanerCLIP	51.91	3.87	52.36	11.38	52.56	9.89	51.57	10.94	52.11	**0.19**	51.74	21.16
	PAR	53.57	6.03	54.18	0.16	51.96	22.94	53.89	4.51	54.25	2.27	50.95	17.78
	InverTune (Ours)	**56.12**	**0.02**	53.50	**0.14**	54.27	**0.28**	**54.76**	**0.09**	55.84	1.02	**55.25**	**0.49**
MSCOCO	No Defense	69.94	95.88	71.20	99.76	70.28	97.42	71.16	99.60	72.07	98.13	71.32	99.28
	FT	68.83	39.09	**69.53**	67.51	68.92	63.67	**69.70**	70.41	**68.77**	25.47	68.25	88.54
	CleanCLIP	65.03	14.17	63.70	55.47	64.09	38.71	67.61	64.83	67.56	13.42	66.53	84.55
	CleanerCLIP	65.73	7.94	68.82	14.93	65.98	14.31	64.67	15.01	66.39	3.41	65.21	30.41
	PAR	68.42	15.43	68.11	**0.37**	66.64	31.09	68.28	7.83	67.42	4.30	65.73	16.47
	InverTune (Ours)	**71.12**	**0.04**	69.16	0.52	**69.94**	**1.12**	68.98	**0.48**	68.02	**1.73**	**69.58**	**0.68**



(a) Top 20 classes with increased prediction frequency.

(b) ASR of inverted and original trigger.

Figure 5: Results of backdoor target identification and trigger inversion.

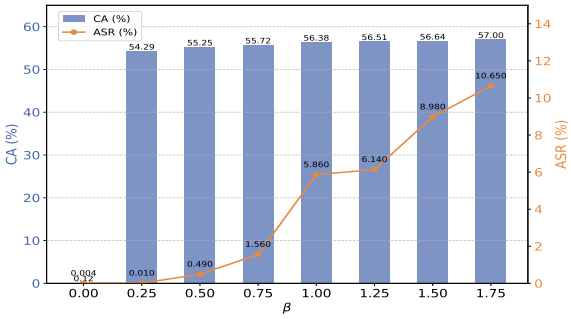


Figure 6: Influence of β on InverTune's Defense Effectiveness under BadClip Attack Scenario.

achieves a superior security-performance trade-off which is raised by the correct backdoor information identification.

Backdoor label Identification and Inversion. InverTune consists of two important steps: target identification and trigger inversion. To demonstrate this effectiveness, we exhibit the corresponding results. For step 1, we apply universal adversarial perturbations to clean examples and feed them the compromised model with "mushroom" as the designated target class. As demonstrated in Figure 5a, we observe dramatic distribution shifts in prediction frequencies. Specifically, only two categories exhibit notable increases: "mushroom" shows a 97.23% surge in classification frequency compared to clean samples, while "agaric", (a mushroom subspecies sharing similar visual characteristics), experiences a marginal 2.13% rise. This divergence distribution reveals that adversarial perturbations are effectively utilized to identify the target label. For the second step, we reconstruct trigger patterns. Here, we argue that the trigger we construct is to activate backdoor pathways for defense without requiring physical trigger replication. As shown in Figure 5b, inverted triggers achieve similar attack behavior alignment with original patterns, meaning the inverted trigger largely mimics the attack behavior of the real trigger.

4.3 Influence of hyperparameters

In this section, we study the influence of different hyperparameters. As formulated in Equation (7), the coefficients λ_1-λ_4 control the relative importance of four loss components during backdoor inversion, while β in Equation (11) governs the trade-off between model cleanliness and usability during the elimination phase.

Our experiments show several important patterns in hyperparameter sensitivity. For the inversion-related hyperparameters (see in Table 2), we observe that λ_1, which weights the contrastive learning loss, produces significantly improved ASR when increased, though with diminishing returns beyond $\lambda_1 > 5.0$ due to deteriorating visual quality of the inverted triggers. The visual feature consistency term controlled by λ_2 demonstrates a clear sweet spot, where insufficient weighting ($\lambda_2 = 0.1$) fails to achieve effective attacks, particularly on SIG, WaNet, and BadCLIP, while excessive emphasis ($\lambda_2 > 1.0$) degrades ASR by over-constraining the feature space. Optimal visual quality and attack effectiveness are achieved with $\lambda_3 = 1.0$ and $\lambda_4 = 0.01$, which properly balance trigger stealthiness and functionality. Excessive values of λ_3 and λ_4 shift the focus of the inversion process towards trigger size optimization, thereby compromising the adversarial effectiveness of the inverted triggers.

Table 2: Influence of λ Parameters on Reverse-Engineered Trigger ASR.

Attacks	λ_1				λ_2				λ_3				λ_4			
	1.0	5.0	10.0	20.0	0.1	0.5	1.0	5.0	0.5	1.0	5.0	10.0	0.005	0.01	0.05	0.1
BadNet	51.97	83.73	84.84	87.67	87.11	83.73	83.86	60.86	90.52	83.73	84.53	65.42	81.70	83.73	48.75	43.32
Blended	37.09	90.21	92.09	92.62	86.72	90.21	89.57	62.33	91.83	90.21	53.17	33.38	93.97	90.21	20.02	10.02
SIG	39.84	76.38	78.84	79.29	73.62	76.38	71.18	71.74	80.02	76.38	41.74	20.08	80.10	76.38	12.44	0.05
WaNet	60.81	90.81	93.96	89.83	71.53	90.81	87.15	81.04	92.16	90.81	78.37	37.02	92.38	90.81	30.06	20.03
BadEncoder	68.72	75.58	77.43	78.59	75.13	75.58	70.76	70.23	75.04	75.58	66.64	65.38	77.13	75.58	65.23	62.52
BadCLIP	73.38	89.72	87.38	89.48	79.89	89.72	65.08	57.17	69.10	89.72	73.54	69.74	91.13	89.72	59.86	20.83

Table 3: Comparison of universal adversarial perturbation (UAP) and inverted trigger (InvT) for the Activation Tuning.

Methods		BadNet		Blended		SIG		WaNet		BadEncoder		BadCLIP	
		CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓
Top-1	UAP	55.55	23.24	53.46	89.92	**54.27**	58.91	52.25	25.74	52.81	67.13	52.02	54.33
	InvT	**56.12**	**0.02**	**53.50**	**0.14**	**54.27**	**0.03**	**54.76**	**0.09**	**55.84**	**0.02**	**55.25**	**0.49**
Top-3	UAP	76.76	47.39	74.99	95.49	75.71	76.91	73.64	50.86	74.67	69.76	73.42	71.15
	InvT	**77.24**	**0.20**	**75.35**	**0.74**	**75.92**	**0.10**	**75.92**	**0.40**	**77.05**	**0.20**	**76.45**	**1.17**
Top-5	UAP	83.63	58.98	82.02	96.75	82.84	82.16	81.05	60.76	81.73	71.04	80.80	76.18
	InvT	**84.10**	**0.46**	**82.54**	**1.47**	**83.04**	**0.20**	**83.04**	**0.77**	**83.87**	**0.44**	**83.35**	**1.67**
Top-10	UAP	90.11	72.94	89.01	98.01	89.53	87.85	88.33	72.45	88.76	72.85	88.13	81.88
	InvT	**90.56**	**0.95**	**89.52**	**3.42**	**89.80**	**0.41**	**89.80**	**1.71**	**90.37**	**0.91**	**90.00**	**2.60**

The elimination phase analysis (see in Figure 6) shows the critical role of β in balancing security and utility. The extreme case of $\beta = 0$, which completely prioritizes backdoor removal, reduces both CA (0.12%) and ASR (0.004%) to near-zero levels, validating the necessity of the usability term in Equation 10. As β increases, we observe distinct patterns: CA shows stable improvement that plateaus when $\beta > 0.50$, while ASR exhibits more dramatic growth, particularly in the range $\beta \in [0.75, 1.0]$ where it increases from 1.560% to 5.860%. Our selected value $\beta = 0.5$ achieves an effective balance, maintaining ASR at 0.49% while preserving 55.25% CA, demonstrating both the stability of InverTune and the effectiveness of our loss formulation.

4.4 Ablation Study

Our analysis in Section 3.1 reveals behavioral distinctions and connections between adversarial and backdoor-triggered samples in compromised models. While both input types induce target-class misclassification, they exploit fundamentally different model vulnerabilities. This mechanistic necessitates our novel trigger inversion approach to specifically isolate and neutralize backdoor artifacts rather than relying solely on adversarial patterns.

To empirically validate this requirement, we conduct an ablation study comparing the complete InverTune framework (InvT) against a variant (UAP) that directly fine-tunes using first-stage adversarial perturbations while omitting trigger inversion. As shown in Table 3, InvT demonstrates overwhelming superiority across all metrics. When k=1, 3, 5, and 10, the average ASR of InvT is 0.13%, 0.46%, 0.84%, and 1.67% respectively, significantly outperforming UAP with 53.21%, 68.59%, 74.31%, 81.00%. The performance gap stems from UAP's fundamental limitation: while adversarial fine-tuning enhances noise robustness and marginally reduces surface-level ASR, it fails to address deeper backdoor information. Moreover, InvT simultaneously preserves superior CA through targeted backdoor pathway disruption compared to indiscriminate adversarial examples. These experiments demonstrate the effectiveness and necessity of the inversion step in InverTune, as it not only enhances backdoor removal but also better preserves the model's usability.

4.5 Backdoor Configuration

Section 4.2 presents a comprehensive evaluation of InverTune's effectiveness. In this sections, we conduct in-depth analyses of the defense mechanism across different dimensions, including target labels and model architectures. To save resources, we mainly focus on **BadCLIP**, which represents the most advanced attack and poses the most significant challenge to defenses.

The impact of target label. To assess the generalizability of InverTune across diverse attack targets, we further set "banana", "lemon", and "ski" as the target label and train BadCLIP attack models with distinct trigger patterns. As illustrated in Table 4, baseline like FT and CleanCLIP remain vulnerable to BadCLIP attacks regardless of target label variations. More advanced defenses such as CleanerCLIP and PAR exhibit notable performance fluctuations: CleanerCLIP's effectiveness decreases from 16.16% to 25.36% and PAR's from 11.72% to 36.07% when switching from "ski" to "lemon". In contrast, InverTune maintains consistent defensive capabilities, achieving superior performance in both ASR ($\approx$ 1%) and CA metrics across all target labels. These results demonstrate InverTune's

Table 4: Performance comparison of InverTune and baseline defenses against BadCLIP under different target labels.

Target Label	Banana		Lemon		Ski	
	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓
No Defense	58.20	98.16	58.11	97.16	58.31	98.46
FT	54.77	83.14	54.93	89.65	54.34	79.70
CleanCLIP	53.48	74.85	54.50	72.82	53.94	77.75
CleanerCLIP	52.09	20.41	51.69	25.36	51.67	16.16
PAR	53.64	17.65	53.91	36.07	53.62	11.72
InverTune (Ours)	**57.01**	**1.14**	**55.81**	**1.01**	**56.93**	**1.51**

Table 5: Performance comparison of defense methods across different model architectures.

Backbone	RN101		ViT-B/16		ViT-B/32	
	CA ↑	ASR ↓	CA ↑	ASR ↓	CA ↑	ASR ↓
No Defense	59.17	83.17	66.78	99.90	60.97	99.23
FT	**56.85**	58.29	**63.01**	83.75	54.72	91.61
CleanCLIP	56.14	42.53	61.91	80.33	53.16	79.36
CleanerCLIP	52.76	3.25	58.81	31.17	53.64	64.60
PAR	55.60	1.17	57.98	18.14	50.82	76.37
InverTune (Ours)	55.76	**1.00**	59.80	**0.09**	**54.83**	**0.17**

robust ability to identify and neutralize backdoor threats regardless of the target label selection.

The impact of model structure. To assess the generalizability of InverTune on architectures, we evaluate its performance across diverse model architectures including RN101, ViT-B/16, and ViT-B/32. As shown in Table 5, architectural transitions significantly impact defense performance. Notably, baselines exhibit substantial performance fluctuations across different architectures. For instance, PAR shows severe performance degradation when transitioning from RN101 to ViT-B/32, with ASR increasing dramatically from 1.17% to 76.37% and CA declining from 55.60% to 50.82%. Similarly, CleanerCLIP's effectiveness varies considerably, with ASR ranging from 3.25% to 64.60% across different architectures. In contrast, InverTune exhibits remarkable stability and superior defensive capability across all evaluated architectures, maintaining an average ASR of merely 1.22% under BadCLIP attacks while preserving competitive CA. This consistent performance across both CNN-based (RN50, RN101) and Transformer-based (ViT-B/32, ViT-B/16) architectures validates its architectural robustness and generalizability. This architecture-agnostic effectiveness originates from InverTune's backdoor inversion paradigm, which directly targets fundamental cross-modal activation patterns rather than architecture-specific features.

5 RELATED WORK

5.1 Backdoor Attacks in MCL

Traditional backdoor attacks, such as BadNet [15], Blended [8], SIG [4] and TrojanNet [36], originally target unimodal neural networks but can be adapted to compromise Multimodal Contrastive Learning models through data poisoning. However, recent MCL-specific attacks exploit cross-modal interactions more effectively. Carlini et al. [6] show that minimal data poisoning can introduce severe vulnerabilities. BadEncoder [18] targets self-supervised learning by poisoning pre-trained image encoders, causing downstream classifiers to inherit backdoor behaviors while maintaining model accuracy. GhostEncoder [39] introduces a dynamic invisible backdoor using image steganography to embed hidden triggers into benign images. Notably, BadCLIP [21] introduces a dual-embedding framework that aligns poisoned samples with target features, creating natural-looking triggers resistant to standard defenses. Adding to this threat landscape, Bai et al. [2] propose a prompt-based backdoor attack that manipulates both image and text encoders using learnable triggers and trigger-aware prompts. These approaches highlight the diverse strategies employed in backdoor attacks and the urgent need for effective defenses.

5.2 Backdoor Defenses in MCL

Backdoor defenses in MCL involve both detection and mitigation strategies. DECREE [13] focuses on identifying backdoors but lacks effective mechanisms for removal. While SSL-Cleanse [47] is designed for self-supervised learning, it not only detects backdoors but also incorporates a purification process to mitigate them. Fine-tuning-based approaches, such as CleanCLIP [3], PAR [35], and CleanerCLIP [42], attempt to remove backdoors by re-learning representations or leveraging counterfactual augmentations. However, these methods may require large clean datasets or introduce performance trade-offs. ABD [19] creatively leverages adversarial examples to approximate backdoor samples but faces challenges in maintaining clean accuracy. Pre-training defenses, such as RoCLIP [44] and SafeCLIP [43], mitigate backdoors by filtering poisoned data during pre-training. However, their effectiveness relies on access to the pre-training process, making them unsuitable for scenarios where only a trained model is available. Our method aims to effectively eliminate backdoor threats in multimodal contrastive learning models while preserving their original performance and generalization capabilities.

6 CONCLUSION

In this paper, we present InverTune, a novel backdoor defense framework for large-scale multimodal contrastive learning models. Our approach integrates three key components: adversarial-based target label identification, gradient-guided trigger inversion, and activation-aware fine-tuning. Extensive evaluations on multiple datasets demonstrate that InverTune achieves state-of-the-art defensive performance across diverse attack scenarios, consistently reducing attack success rates while maintaining model utility. Our framework significantly enhances the robustness of multimodal models against backdoor threats, providing a practical solution for real-world applications.

REFERENCES

[1] Stanislaw Antol, Aishwarya Agrawal, Jiasen Lu, Margaret Mitchell, Dhruv Batra, C Lawrence Zitnick, and Devi Parikh. 2015. Vqa: Visual question answering. In *Proceedings of the IEEE international conference on computer vision*. 2425–2433.

[2] Jiawang Bai, Kuofeng Gao, Shaobo Min, Shu-Tao Xia, Zhifeng Li, and Wei Liu. 2024. Badclip: Trigger-aware prompt learning for backdoor attacks on clip. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24239–24250.

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

[4] Mauro Barni, Kassem Kallas, and Benedetta Tondi. 2019. A new backdoor attack in cnns by training set corruption without label poisoning. In *2019 IEEE International Conference on Image Processing (ICIP)*. IEEE, 101–105.

[5] Manuele Barraco, Marcella Cornia, Silvia Cascianelli, Lorenzo Baraldi, and Rita Cucchiara. 2022. The unreasonable effectiveness of CLIP features for image captioning: an experimental analysis. In *proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 4662–4670.

[6] Nicholas Carlini and Andreas Terzis. 2021. Poisoning and backdooring contrastive learning. *arXiv preprint arXiv:2106.09667* (2021).

[7] Jun Chen, Deyao Zhu, Guocheng Qian, Bernard Ghanem, Zhicheng Yan, Chenchen Zhu, Fanyi Xiao, Sean Chang Culatana, and Mohamed Elhoseiny. 2023. Exploring open-vocabulary semantic segmentation from clip vision encoder distillation only. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 699–710.

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

[9] Jaemin Cho, Seunghyun Yoon, Ajinkya Kale, Franck Dernoncourt, Trung Bui, and Mohit Bansal. 2022. Fine-grained image captioning with clip reward. *arXiv preprint arXiv:2205.13115* (2022).

[10] Anders Christensen, Massimiliano Mancini, A Koepke, Ole Winther, and Zeynep Akata. 2023. Image-free classifier injection for zero-shot classification. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*. 19072–19081.

[11] Sedigheh Eslami, Gerard de Melo, and Christoph Meinel. 2021. Does clip benefit visual question answering in the medical domain as much as it does in the general domain? *arXiv preprint arXiv:2112.13906* (2021).

[12] Sedigheh Eslami, Christoph Meinel, and Gerard De Melo. 2023. Pubmedclip: How much does clip benefit visual question answering in the medical domain?. In *Findings of the Association for Computational Linguistics: EACL 2023*. 1181–1193.

[13] Shiwei Feng, Guanhong Tao, Siyuan Cheng, Guangyu Shen, Xiangzhe Xu, Yingqi Liu, Kaiyuan Zhang, Shiqing Ma, and Xiangyu Zhang. 2023. Detecting backdoors in pre-trained encoders. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 16352–16362.

[14] Xueluan Gong, Yanjiao Chen, Wang Yang, Qian Wang, Yuzhe Gu, Huayang Huang, and Chao Shen. 2023. Redeem Myself: Purifying Backdoors in Deep Learning Models using Self Attention Distillation. In *2023 IEEE Symposium on Security and Privacy (SP)*. 755–772. https://doi.org/10.1109/SP46215.2023.10179375

[15] Tianyu Gu, Brendan Dolan-Gavitt, and Siddharth Garg. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. *arXiv preprint arXiv:1708.06733* (2017).

[16] Wenbo Guo, Lun Wang, Xinyu Xing, Min Du, and Dawn Song. 2019. Tabor: A highly accurate approach to inspecting and restoring trojan backdoors in ai systems. *arXiv preprint arXiv:1908.01763* (2019).

[17] Chao Jia, Yinfei Yang, Ye Xia, Yi-Ting Chen, Zarana Parekh, Hieu Pham, Quoc Le, Yun-Hsuan Sung, Zhen Li, and Tom Duerig. 2021. Scaling up visual and vision-language representation learning with noisy text supervision. In *International conference on machine learning*. PMLR, 4904–4916.

[18] Jinyuan Jia, Yupei Liu, and Neil Zhenqiang Gong. 2022. Badencoder: Backdoor attacks to pre-trained encoders in self-supervised learning. In *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2043–2059.

[19] Junhao Kuang, Siyuan Liang, Jiawei Liang, Kuanrong Liu, and Xiaochun Cao. 2024. Adversarial backdoor defense in clip. *arXiv preprint arXiv:2409.15968* (2024).

[20] Hugo Larochelle, Dumitru Erhan, and Yoshua Bengio. 2008. Zero-data learning of new tasks.. In *AAAI*, Vol. 1. 3.

[21] Siyuan Liang, Mingli Zhu, Aishan Liu, Baoyuan Wu, Xiaochun Cao, and Ee-Chien Chang. 2024. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 24645–24654.

[22] Tsung-Yi Lin, Michael Maire, Serge Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C Lawrence Zitnick. 2014. Microsoft coco: Common objects in context. In *Computer vision–ECCV 2014: 13th European conference, zurich, Switzerland, September 6-12, 2014, proceedings, part v 13*. Springer, 740–755.

[23] Yingqi Liu, Wen-Chuan Lee, Guanhong Tao, Shiqing Ma, Yousra Aafer, and Xiangyu Zhang. 2019. Abs: Scanning neural networks for back-doors by artificial brain stimulation. In *Proceedings of the 2019 ACM SIGSAC Conference on Computer and Communications Security*. 1265–1282.

[24] James MacQueen. 1967. Some methods for classification and analysis of multivariate observations. In *Proceedings of the Fifth Berkeley Symposium on Mathematical Statistics and Probability, Volume 1: Statistics*, Vol. 5. University of California press, 281–298.

[25] George A Miller. 1995. WordNet: a lexical database for English. *Commun. ACM* 38, 11 (1995), 39–41.

[26] Ron Mokady, Amir Hertz, and Amit H Bermano. 2021. Clipcap: Clip prefix for image captioning. *arXiv preprint arXiv:2111.09734* (2021).

[27] Bingxu Mu, Zhenxing Niu, Le Wang, Xue Wang, Qiguang Miao, Rong Jin, and Gang Hua. 2023. Progressive backdoor erasing via connecting backdoor and adversarial attacks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*. 20495–20503.

[28] Anh Nguyen and Anh Tran. 2021. Wanet–imperceptible warping-based backdoor attack. *arXiv preprint arXiv:2102.10369* (2021).

[29] Zhenxing Niu, Yuyao Sun, Qiguang Miao, Rong Jin, and Gang Hua. 2024. Towards unified robustness against both backdoor and adversarial attacks. *IEEE transactions on pattern analysis and machine intelligence* (2024).

[30] Aaron van den Oord, Yazhe Li, and Oriol Vinyals. 2018. Representation learning with contrastive predictive coding. *arXiv preprint arXiv:1807.03748* (2018).

[31] Qi Qian and Juhua Hu. 2024. Online zero-shot classification with clip. In *European Conference on Computer Vision*. Springer, 462–477.

[32] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. 2021. Learning transferable visual models from natural language supervision. In *International conference on machine learning*. PMLR, 8748–8763.

[33] Olga Russakovsky, Jia Deng, Hao Su, Jonathan Krause, Sanjeev Satheesh, Sean Ma, Zhiheng Huang, Andrej Karpathy, Aditya Khosla, Michael Bernstein, Alexander C. Berg, and Li Fei-Fei. 2015. ImageNet Large Scale Visual Recognition Challenge. *International Journal of Computer Vision (IJCV)* 115, 3 (2015), 211–252. https://doi.org/10.1007/s11263-015-0816-y

[34] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 2018. Conceptual captions: A cleaned, hypernymed, image alt-text dataset for automatic image captioning. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. 2556–2565.

[35] Naman Deep Singh, Francesco Croce, and Matthias Hein. 2024. Perturb and Recover: Fine-tuning for Effective Backdoor Removal from CLIP. *arXiv preprint arXiv:2412.00727* (2024).

[36] Ruixiang Tang, Mengnan Du, Ninghao Liu, Fan Yang, and Xia Hu. 2020. An embarrassingly simple approach for trojan attack in deep neural networks. In *Proceedings of the 26th ACM SIGKDD international conference on knowledge discovery & data mining*. 218–228.

[37] Oriol Vinyals, Alexander Toshev, Samy Bengio, and Dumitru Erhan. 2015. Show and tell: A neural image caption generator. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 3156–3164.

[38] Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*. IEEE, 707–723.

[39] Qiannan Wang, Changchun Yin, Liming Fang, Zhe Liu, Run Wang, and Chenhao Lin. 2024. GhostEncoder: Stealthy backdoor attacks with dynamic triggers to pre-trained encoders in self-supervised learning. *Computers & Security* 142 (2024), 103855.

[40] Zhou Wang, Alan C Bovik, Hamid R Sheikh, and Eero P Simoncelli. 2004. Image quality assessment: from error visibility to structural similarity. *IEEE transactions on image processing* 13, 4 (2004), 600–612.

[41] Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. 2023. Unicorn: A unified backdoor trigger inversion framework. *arXiv preprint arXiv:2304.02786* (2023).

[42] Yuan Xun, Siyuan Liang, Xiaojun Jia, Xinwei Liu, and Xiaochun Cao. 2024. CleanerCLIP: Fine-grained Counterfactual Semantic Augmentation for Backdoor Defense in Contrastive Learning. *CoRR* abs/2409.17601 (2024). https://doi.org/10.48550/arXiv.2409.17601

[43] Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. 2023. Better safe than sorry: Pre-training clip against targeted data poisoning and backdoor attacks. *arXiv preprint arXiv:2310.05862* (2023).

[44] Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. 2023. Robust contrastive language-image pretraining against data poisoning and backdoor attacks. *Advances in Neural Information Processing Systems* 36 (2023), 10678–10691.

[45] Meng Ye and Yuhong Guo. 2017. Zero-shot classification with discriminative semantic representation learning. In *Proceedings of the IEEE conference on computer vision and pattern recognition*. 7140–7148.

[46] Zhifang Zhang, Shuo He, Bingquan Shen, and Lei Feng. 2024. Defending Multimodal Backdoored Models by Repulsive Visual Prompt Tuning. *arXiv preprint arXiv:2412.20392* (2024).

[47] Mengxin Zheng, Jiaqi Xue, Zihao Wang, Xun Chen, Qian Lou, Lei Jiang, and Xiaofeng Wang. 2024. Ssl-cleanse: Trojan detection and mitigation in self-supervised learning. In *European Conference on Computer Vision*. Springer, 405–421.

[48] Kaiyang Zhou, Jingkang Yang, Chen Change Loy, and Ziwei Liu. 2022. Learning to prompt for vision-language models. *International Journal of Computer Vision* 130, 9 (2022), 2337–2348.

[49] Ziqi Zhou, Shengshan Hu, Minghui Li, Hangtao Zhang, Yechao Zhang, and Hai Jin. 2023. Advclip: Downstream-agnostic adversarial examples in multimodal contrastive learning. In *Proceedings of the 31st ACM International Conference on Multimedia*. 6311–6320.

[50] Ziqin Zhou, Yinjie Lei, Bowen Zhang, Lingqiao Liu, and Yifan Liu. 2023. Zegclip: Towards adapting clip for zero-shot semantic segmentation. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*. 11175–11185.

A INTRODUCTION AND CONFIGURATIONS OF DIFFERENT BACKDOOR ATTACKS

A.1 Backdoor Attacks Settings

For all six types of attacks, we adopt a 500K subset of the CC3M dataset [34] as the fine-tuning dataset. All attacks target the class "mushroom." For attacks that require textual descriptions, we construct them by collecting 131 mushroom-related captions from the CC3M dataset and randomly assigning them to the poisoned image samples as their corresponding text descriptions.

- In the BadNet [15] attack, we adopt a 16×16 patch filled with Gaussian noise sampled from a standard normal distribution as the trigger, which is fixed to the bottom-right corner of the clean images.
- In the Blended [8] attack, we generate a trigger image of the same size as the input image using a uniform distribution. We set the transparency of the trigger image to 0.2 and blend it with the clean image, whose transparency is set to 0.8.
- In the SIG [4] attack, sinusoidal noise is generated along the horizontal axis of the image, creating vertical stripes. For each pixel along the width, noise is injected using a sinusoidal function with a frequency of 6 cycles per image width. The noise amplitude is scaled to $60/255$ to stay within a suitable range. This perturbation is applied uniformly to all RGB channels. After adding the noise, pixel values are clipped to $[0, 1]$ to ensure validity.
- In the WaNet [28] attack, we apply a warping transformation to the image using a distortion grid. Following the original implementation, we generate the grid by interpolating a noise tensor to match the image resolution. The grid is then scaled and clipped to $[-1, 1]$ for compatibility with grid sampling. The warping is performed using bilinear interpolation, introducing subtle but adversarial distortions.
- For the BadEncoder [18] attack, we follow the original methodology, where the visual encoder is fine-tuned to embed backdoor triggers while preserving its functionality on clean samples. Unlike the original implementation, we replace the trigger from the official repository with a 16×16 pure white image to ensure a fair comparison with other attacks. This attack is distinct in that it does not require constructing textual descriptions or setting a poisoning rate. Instead, it directly fine-tunes the visual encoder using a reference dataset and a shadow dataset.
- For the BadCLIP [21] attack, following their provided code, we first optimize the patch based on the "mushroom" label. After obtaining the patch, we perform Dual-Embedding injection attack on the clean CLIP model.

For all the attacks described above, we start from the CLIP model pretrained by OpenAI [32], and fine-tune it to obtain a poisoned CLIP model with learning rate 1e-6, batch size 128, and 10 training epochs.

A.2 Visualization of Trigger Patterns

Regarding the attacks mentioned in this paper, in addition to the introduction above, we also present them in the first row of Figure 7.

B BASELINE DEFENSE SETTINGS

In this section, we provide a detailed description of the experimental settings for the four baseline methods discussed in the Main Text.

All the defense methods use subsets of the CC3M dataset [34] in their original setups, though the exact number of samples varies slightly. For fair comparison, we standardize the training data by using a fixed subset of 500K samples across all methods.

- The fine-tuning method (FT), first introduced by CleanCLIP [3], involves fine-tuning the model with a multimodal contrastive loss on a clean dataset. In our experiments, we use the official implementation provided by CleanCLIP, with a learning rate of 4.5e-6, warmup steps of 50, batch size of 64, and 10 training epochs.
- CleanCLIP [3] extends FT by adding a self-supervised loss term. Following its original setup, we set the weights of the self-supervised loss term and the contrastive loss term to 1, with other hyperparameters remaining the same as those in FT.
- PAR [35] adopts a custom learning rate schedule. However, due to the increased size of the fine-tuning dataset, the original setting does not reproduce the reported performance. Therefore, in our experiments, we modify the start learning rate to 3e-6 and the peak learning rate to 5e-6, while keeping all other parameters consistent with the original setup.
- CleanerCLIP [42] is implemented based on CleanCLIP [3]. We follow its original setup, using a batch size of 64 and training for 10 epochs with the AdamW optimizer. The learning rate is linearly warmed up over 10,000 steps, and a weight decay of 0.1 is applied. The Adam momentum factor and RMSProp factor are set to 0.9 and 0.999, respectively, with an epsilon of 1e-8. The base learning rate is set to 4.5e-6.

C IMPLEMENTATION DETAILS OF INVERTUNE

C.1 Backdoor Label Identification

The first step of InverTune is to identify the target category for the backdoor attack. To achieve this, we leverage the 1,000 classes from ImageNet-1K [33] and combine them with predefined templates to construct text prompts.

These categories are derived from WordNet [25], a lexical database that structures words into a hierarchical network based on their semantic relationships. The ImageNet-1K classes encompass a remarkably diverse array of objects, spanning nearly all aspects of the physical world. These include animals (e.g., tiger, goldfish, hummingbird), everyday objects (e.g., laptop, toaster, umbrella), vehicles (e.g., fire truck, sports car, airplane), architectural structures

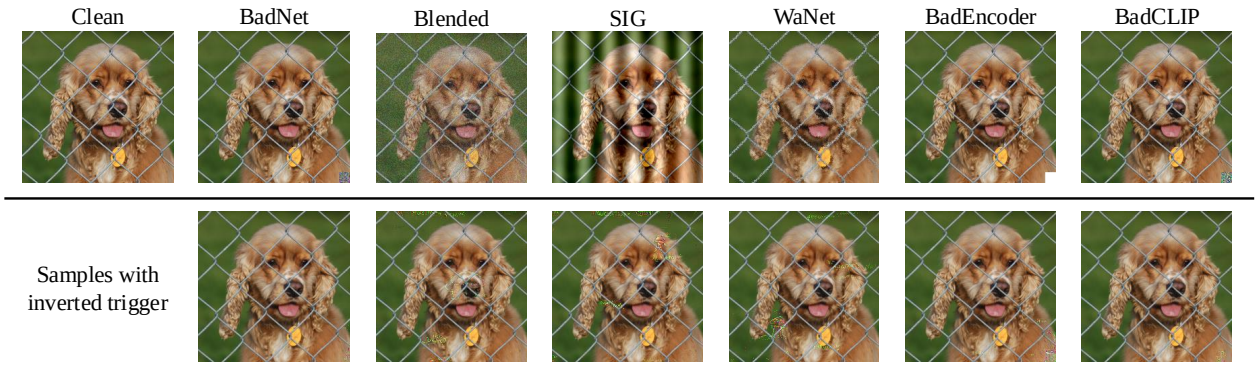


Figure 7: Backdoor sample examples and visualization of trigger inversion effects.

(e.g., lighthouse, suspension bridge, pagoda), and various tools and instruments (e.g., screwdriver, stethoscope, cello).

Given their extensive coverage, these 1,000 categories serve as well-suited candidates for identifying the target labels in backdoor attacks. Their diversity ensures a broad spectrum of potential backdoor targets, making them highly relevant for identifying and mitigating threats in multimodal contrastive learning models. Additionally, the hierarchical nature of WordNet provides a strong semantic foundation, facilitating precise and meaningful target label selection.

Furthermore, Even if the attacker chooses a target category outside ImageNet-1K, a semantically similar class likely exists within it due to WordNet's hierarchy. This ensures our defense remains effective, as the attacker's target can still be meaningfully mapped to an existing label, maintaining robustness against unexpected attacks.

C.2 Trigger Inversion

Algorithm 1 generates high-fidelity backdoor trigger reconstructions while maintaining visual subtlety. The four loss components work together to achieve this, as detailed in Section 3.2 of the Main Text.

The trigger reconstruction process typically converges within a few hundred iterations, significantly faster than training a backdoor from scratch. This efficiency stems from directly optimizing in CLIP's embedding space rather than attempting to model the backdoor through proxy tasks or surrogate networks.

Importantly, our approach supports a wide range of backdoor implementations beyond the standard patch-based triggers. The mask-pattern formulation can reconstruct complex, spatially distributed triggers and even global transformations. The clamp operation on the trigger pattern (line 26, 27) ensures the reconstructed values remain within CLIP's preprocessing bounds, producing realistic images that can be directly used in subsequent defense strategies.

The reconstructed trigger serves as a critical component for our overall defense framework, enabling us to analyze backdoor behavior and develop targeted mitigation strategies in the activation tuning stage. By reproducing the backdoor's trigger, we can effectively probe the model's internal representations to identify compromised components.

C.3 Activation Tuning

After trigger inversion, we focus on mitigating its impact on the model without compromising normal functionality. Traditional fine-tuning methods applied to the entire network risk degrading the model's critical cross-modal performance, which is essential for multimodal models like CLIP. In contrast, Algorithm 2 introduces a novel activation tuning approach that specifically targets neurons involved in backdoor behavior.

The algorithm operates in three phases: (1) identifying network layers most affected by the backdoor trigger, (2) pinpointing the specific neurons within these layers responsible for the backdoor behavior, and (3) selectively fine-tuning only the identified neurons using a custom loss function. This targeted approach minimizes disruption to the model's cross-modal alignment, which is central to CLIP's zero-shot prediction capabilities.

By focusing on critical neurons identified through activation analysis, Algorithm 2 offers significant advantages over traditional backdoor mitigation techniques. This selective intervention is more efficient than whole-network fine-tuning, preserving CLIP's core functionality while effectively addressing backdoor pathways.

D DETAILED RESULTS OF INTERMEDIATE STEPS IN INVERTUNE

D.1 Target Category Identification Results for Six Attacks

We present the target class identification results across six distinct attack scenarios, where "mushroom" serves as the ground truth target label in all cases. Our analysis reveals systematic and statistically significant increases in the prediction frequency of the target class after adversarial perturbation, with attack-specific variations in magnitude.

Based on the experimental results presented in Tables 6, 7, 8 and 11, we observe particularly pronounced adversarial effects in four attack scenarios. These effects, induced by adversarial perturbations, manifest as substantial shifts in the prediction frequency toward the target class. The BadCLIP attack induces the most significant shift, with a 97.23% increase in mushroom prediction frequency, followed by Blended (61.06%), BadNet (39.33%), and SIG (37.40%). As shown in Table 10, even the relatively moderate BadEncoder attack

Algorithm 1 Dual-Space Trigger Inversion for Multimodal CLIP Backdoors

Input: Suspected backdoored CLIP model F with image encoder E_I and text encoder E_T; Clean images $\mathcal{X} = \{x_1, x_2, \ldots, x_n\}$; Target text label y_t identified from Step 1; Number of steps T; Loss weights $\lambda_1, \lambda_2, \lambda_3, \lambda_4$
Output: Inverted trigger mask m and pattern t_{img}
Initialize mask $m \leftarrow$ random tensor in range $[0, 1]$ of shape $3 \times 224 \times 224$
Initialize trigger pattern $t_{\text{img}} \leftarrow$ random tensor of shape $3 \times 224 \times 224$
$\theta \leftarrow \{m, t_{\text{img}}\}$ ▷ Parameters to optimize
Initialize optimizer with learning rate α
Precompute all text embeddings for available classes $\{y_1, y_2, \ldots, y_N\}$:
  $E_T(y_j) \leftarrow$ normalized text embeddings for each class $j \in \{1, \ldots, N\}$
for step = 1 **to** T **do**
  Sample a batch of clean images $\{x_1, x_2, \ldots, x_b\} \subseteq \mathcal{X}$
  Generate poisoned samples: $\tilde{x}_i = m \odot t_{\text{img}} + (1 - m) \odot x_i$ for $i \in \{1, \ldots, b\}$
  Compute image embeddings: $E_I(\tilde{x}_i) \leftarrow F_I(\tilde{x}_i)$
  Normalize embeddings: $E_I(\tilde{x}_i) \leftarrow \frac{E_I(\tilde{x}_i)}{\|E_I(\tilde{x}_i)\|_2}$
  ▷ Calculate the four loss components
  ▷ 1. Cross-Modal Alignment Loss via InfoNCE
  $\mathcal{L}_{\text{align}} \leftarrow -\frac{1}{b} \sum_{i=1}^{b} \log \frac{\exp(\text{sim}(E_I(\tilde{x}_i), E_T(y_t))/\tau)}{\sum_{j=1}^{N} \exp(\text{sim}(E_I(\tilde{x}_i), E_T(y_j))/\tau)}$
  ▷ 2. Embedding Space Preservation Loss
  $\mathcal{L}_{\text{emb}} \leftarrow \frac{1}{b} \sum_{i=1}^{b} \|E_I(\tilde{x}_i) - E_I(x_i)\|_2$
  ▷ 3. Visual Similarity Loss
  $\mathcal{L}_{\text{sim}} \leftarrow \frac{1}{b} \sum_{i=1}^{b} (1 - \text{SSIM}(\tilde{x}_i, x_i))$
  ▷ 4. Trigger Sparsity Loss
  $\mathcal{L}_{\text{mask}} \leftarrow \|m\|_1$
  ▷ Combined Loss
  $\mathcal{L}_{\text{inver}} \leftarrow \lambda_1 \mathcal{L}_{\text{align}} + \lambda_2 \mathcal{L}_{\text{emb}} + \lambda_3 \mathcal{L}_{\text{sim}} + \lambda_4 \mathcal{L}_{\text{mask}}$
  Update parameters: $\theta \leftarrow \theta - \alpha \nabla_\theta \mathcal{L}_{\text{inver}}$
  Clamp mask: $m \leftarrow \text{clamp}(m, 0, 1)$
  Clamp trigger: $t_{\text{img}} \leftarrow \text{clamp}(t_{\text{img}}, -1.7922, 2.1461)$ ▷ CLIP normalization bounds
end for
return m, t_{img}

(4.20% increase) leads to a statistically significant bias, maintaining a 3.62 percentage point advantage over the second-most predicted class ("pillow" at 0.58%).

Table 9 reveals an intriguing pattern of taxonomic-specific vulnerability in the WaNet attack. The method produces nearly identical prediction increases for both the target "mushroom" category (26.38%) and its taxonomically related counterpart "agaric" (25.99%), with merely a 0.39 percentage point differential. This remarkable similarity validates the relationship between the adversarial perturbations and the target class of the backdoor attack, as stated in the Main Text. From another perspective, the CLIP model's inherent semantic clustering enables the identification of a semantically similar class within the ImageNet-1K label space, even when the attacker's intended target does not fall within the 1,000 predefined categories, as discussed in Section C.1. This property facilitates the subsequent steps of trigger inversion and activation tuning.

D.2 Inverted Trigger Visualization Results for Six Attacks

Our inversion results, shown in the bottom row of Figure 7, demonstrate two distinct spatial distribution patterns corresponding to different attack types. For localized trigger attacks (BadNet, BadCLIP, and BadEncoder), the inverted triggers maintain the characteristic bottom-right corner positioning observed in the original attacks. Conversely, for globally distributed attacks (Blended, SIG, and WaNet), the inverted triggers successfully reproduce the expected multi-region distribution patterns.

The spatial consistency between original and inverted triggers is evident in both cases, with the inverted versions achieving comparable attack success rates to their original counterparts. These results confirm that our inversion method preserves the essential spatial characteristics of different trigger types while maintaining their functional effectiveness.

D.3 Key Layers Selected in Activation Tuning

In this section, we present the layer selection results during the Activation Tuning process. Since different attacks show minimal variation in layer activation outcomes, we demonstrate the anomalous response layers for four distinct CLIP architectures (RN50, RN101, ViT-B/16, ViT-B/32) when confronted with inversion triggers, using the BadCLIP attack as the representative case.

Algorithm 2 Activation Tuning for Backdoor Mitigation in MCL Models

Input: Backdoored CLIP model F with encoders E_I and E_T; Inverted trigger (m, t_{img}) from Algorithm 1; Clean inputs $\mathcal{X}$; Set of candidate layers $\mathcal{L}$; Balance parameter β
Output: Fine-tuned CLIP model with neutralized backdoor
Phase 1: Identify Critical Layers
Compute clean activations $\{A^l_{\text{clean}}\}$ for each layer $l \in \mathcal{L}$ using $\mathcal{X}$
for each $x \in \mathcal{X}$ **do**
    Generate triggered image $\tilde{x} \leftarrow m \odot t_{\text{img}} + (1 - m) \odot x$
end for
Compute triggered activations $\{A^l_{\text{triggered}}\}$ for each layer $l \in \mathcal{L}$
for each layer $l \in \mathcal{L}$ **do**
    Compute mean clean activation $\mu^l_{\text{clean}} \leftarrow \text{mean}(A^l_{\text{clean}})$
    Compute mean triggered activation $\mu^l_{\text{triggered}} \leftarrow \text{mean}(A^l_{\text{triggered}})$
    Calculate normalized activation difference:
    $\text{diff}^l \leftarrow \frac{\|\mu^l_{\text{clean}} - \mu^l_{\text{triggered}}\|_2}{\|\mu^l_{\text{clean}}\|_2}$
end for
Calculate threshold $\tau \leftarrow \text{mean}(\{\text{diff}^l\}) + \text{std}(\{\text{diff}^l\})$
Identify critical layers $\mathcal{L}_{\text{critical}} \leftarrow \{l \in \mathcal{L} \mid \text{diff}^l > \tau\}$
Phase 2: Identify Critical Neurons
for each layer $l \in \mathcal{L}_{\text{critical}}$ **do**
    Compute activation difference $\Delta^l \leftarrow |\mu^l_{\text{clean}} - \mu^l_{\text{triggered}}|$
    Apply K-means clustering to Δ^l with $k = 2$ clusters
    Identify critical cluster C^l_{critical} with largest centroid value
    Create neuron mask M^l where neurons in C^l_{critical} are set to 1
end for
Phase 3: Selective Fine-tuning
Create parameter masks based on critical neuron masks $\{M^l\}$
Initialize fine-tuned model $F' \leftarrow F$ ▷ Copy of original model
Create optimizer for model parameters with neuron-masked gradients
for each training step **do**
    Sample batch of clean images $\{x_1, x_2, ..., x_b\} \subseteq \mathcal{X}$
    Generate triggered images $\{\tilde{x}_i = m \odot t_{\text{img}} + (1 - m) \odot x_i\}$
    ▷ Compute activation alignment loss
    $\mathcal{L}_{\text{activation}} \leftarrow 0$
    for each layer $l \in \mathcal{L}_{\text{critical}}$ **do**
        Extract activations for clean and triggered inputs: a^l_{clean}, $a^l_{\text{triggered}}$
        Apply neuron mask: $a^l_{\text{clean}} \leftarrow a^l_{\text{clean}} \odot M^l$
        Apply neuron mask: $a^l_{\text{triggered}} \leftarrow a^l_{\text{triggered}} \odot M^l$
        $\mathcal{L}_{\text{activation}} \leftarrow \mathcal{L}_{\text{activation}} + \|a^l_{\text{clean}} - a^l_{\text{triggered}}\|_2^2$
    end for
    ▷ Compute preservation loss
    With original model F, compute $E_I^{\text{orig}}(x_i)$ for each x_i
    With fine-tuned model F', compute $E_I(x_i)$ for each x_i
    $\mathcal{L}_{\text{preserve}} \leftarrow \|\text{sim}(E_I(x_i), E_T(y_i)) - \text{sim}(E_I^{\text{orig}}(x_i), E_T(y_i))\|_2^2$
    ▷ Combined loss
    $\mathcal{L}_{\text{tune}} \leftarrow \mathcal{L}_{\text{activation}} + \beta \cdot \mathcal{L}_{\text{preserve}}$
    Compute gradients and apply masked updates to parameters
    Update only parameters corresponding to critical neurons
end for
return Fine-tuned model F'

Table 6: Top 20 classes with the largest absolute increase under adversarial attack on BadNet-poisoned model.

Class	Clean	Adversarial	Absolute Increase
mushroom	18	19981	**+39.93%**
echidna	34	11675	+23.28%
ocarina	43	9361	+18.64%
switch	16	5232	+10.43%
agaric	68	952	+1.77%
eggnog	38	834	+1.59%
chain mail	43	456	+0.83%
doormat	51	330	+0.56%
plastic bag	32	207	+0.35%
ashcan	31	141	+0.22%
monitor	56	122	+0.13%
crossword puzzle	44	55	+0.02%
space bar	4	9	+0.01%
file	2	7	+0.01%
cardigan	0	0	+0.00%
crane	0	0	+0.00%
maillot	0	0	+0.00%
brabancon griffon	3	0	−0.01%
ear	3	0	−0.01%
drake	4	0	−0.01%

Table 7: Top 20 classes with the largest absolute increase under adversarial attack on Blended-poisoned model.

Class	Clean	Adversarial	Absolute Increase
mushroom	18	30547	**+61.06%**
doormat	64	2442	+4.76%
poncho	56	1740	+3.37%
switch	26	1564	+3.08%
eggnog	37	1432	+2.79%
echidna	39	570	+1.06%
ocarina	41	558	+1.03%
sock	78	494	+0.83%
pillow	60	305	+0.49%
web site	63	301	+0.48%
worm fence	33	259	+0.45%
slug	42	267	+0.45%
chain mail	48	240	+0.38%
plunger	37	190	+0.31%
mashed potato	79	215	+0.27%
miniskirt	74	180	+0.21%
hotdog	49	152	+0.21%
cassette	39	138	+0.20%
joystick	40	132	+0.18%
monitor	42	116	+0.15%

Table 8: Top 20 classes with the largest absolute increase under adversarial attack on SIG-poisoned model.

Class	Clean	Adversarial	Absolute Increase
mushroom	76	18775	**+37.40%**
agaric	14	8808	+17.59%
monitor	54	5183	+10.26%
modem	65	1596	+3.06%
chain mail	48	1480	+2.86%
thimble	34	734	+1.40%
airship	40	653	+1.23%
admiral	18	551	+1.07%
ocarina	42	533	+0.98%
echidna	37	485	+0.90%
joystick	37	476	+0.88%
desktop computer	101	347	+0.49%
file	3	230	+0.45%
ashcan	37	242	+0.41%
microwave	59	249	+0.38%
crate	53	227	+0.35%
switch	16	171	+0.31%
television	71	220	+0.30%
centipede	27	155	+0.26%
sidewinder	25	141	+0.23%

Table 9: Top 20 classes with the largest absolute increase under adversarial attack on WaNet-poisoned model.

Class	Clean	Adversarial	Absolute Increase
mushroom	18	13206	**+26.38%**
agaric	60	13056	+25.99%
chain mail	48	5634	+11.17%
switch	26	2008	+3.96%
echidna	39	1578	+3.08%
admiral	14	1059	+2.09%
joystick	40	836	+1.59%
stinkhorn	45	671	+1.25%
poncho	56	568	+1.02%
file	0	422	+0.84%
ocarina	41	419	+0.76%
shovel	53	407	+0.71%
projectile	8	319	+0.62%
consomme	67	372	+0.61%
eggnog	37	317	+0.56%
carbonara	60	325	+0.53%
armadillo	46	256	+0.42%
Indian cobra	28	222	+0.39%
microwave	61	235	+0.35%
doormat	64	234	+0.34%

The visual encoder of ResNet architectures consists of four residual layers. For these architectures, the analysis shows concentrated sensitivity in the final residual layers. As shown in Table 12, RN50 exhibits extreme sensitivity in `visual.layer4` with an impact value of 1.3802, which exceeds the significance threshold ($\mu + \sigma = 0.9914$) by 39.2%. Similarly, Table 13 reveals that RN101's `visual.layer4` shows comparable vulnerability with an impact of

Table 10: Top 20 classes with the largest absolute increase under adversarial attack on BadEncoder-poisoned model.

Class	Clean	Adversarial	Absolute Increase
mushroom	422	2522	**+4.20%**
pillow	94	384	+0.58%
mongoose	91	374	+0.57%
toy poodle	112	393	+0.56%
quail	128	328	+0.40%
beagle	103	292	+0.38%
poncho	63	231	+0.34%
miniskirt	90	254	+0.33%
Labrador retriever	256	418	+0.32%
dingo	126	281	+0.31%
slug	40	187	+0.29%
desktop computer	135	279	+0.29%
diaper	105	246	+0.28%
amphibian	18	152	+0.27%
rock python	83	215	+0.26%
clog	32	159	+0.25%
eel	76	201	+0.25%
racer	69	191	+0.24%
bloodhound	54	175	+0.24%
mouse	27	146	+0.24%

Table 11: Top 20 classes with the largest absolute increase under adversarial attack on BadCLIP-poisoned model.

Class	Clean	Adversarial	Absolute Increase
mushroom	6	48623	**+97.23%**
agaric	67	1132	+2.13%
maillot	0	0	+0.00%
crane	0	0	+0.00%
cardigan	0	0	+0.00%
space bar	3	0	−0.01%
ear	3	0	−0.01%
drake	3	0	−0.01%
brabancon griffon	4	0	−0.01%
horizontal bar	4	0	−0.01%
swab	5	0	−0.01%
black-footed ferret	5	0	−0.01%
file	5	0	−0.01%
lhasa	6	0	−0.01%
pickelhaube	6	0	−0.01%
nipple	7	0	−0.01%
mouse	8	0	−0.02%
appenzeller	9	1	−0.02%
bow	8	0	−0.02%
tiger cat	8	0	−0.02%

1.1823, 38.5% above its threshold of 0.8538. This final-layer concentration suggests that ResNet protections can focus on monitoring these critical bottlenecks.

Table 12: Layer impact analysis results for RN50.

Layer	Impact	Selected Key Layer
visual.layer1	0.1407	No
visual.layer2	0.1750	No
visual.layer3	0.1435	No
visual.layer4	1.3802	**Yes**
Significance Threshold		0.9914
Mean		0.4599
Std		0.5315

Table 13: Layer impact analysis results for RN101.

Layer	Impact	Selected Key Layer
visual.layer1	0.1329	No
visual.layer2	0.1492	No
visual.layer3	0.1547	No
visual.layer4	1.1823	**Yes**
Significance Threshold		0.8538
Mean		0.4048
Std		0.4490

Table 14: Layer impact analysis results for ViT-B/16.

Layer	Impact	Selected Key Layer
visual.transformer.resblocks.0	0.0916	No
visual.transformer.resblocks.1	0.1458	No
visual.transformer.resblocks.2	0.2858	No
visual.transformer.resblocks.3	0.3423	**Yes**
visual.transformer.resblocks.4	0.3247	**Yes**
visual.transformer.resblocks.5	0.2996	**Yes**
visual.transformer.resblocks.6	0.1895	No
visual.transformer.resblocks.7	0.1879	No
visual.transformer.resblocks.8	0.2002	No
visual.transformer.resblocks.9	0.1745	No
visual.transformer.resblocks.10	0.1959	No
visual.transformer.resblocks.11	0.1700	No
Significance Threshold		0.2915
Mean		0.2173
Std		0.0742

The CLIP visual encoder using the ViT-B architecture consists of 12 Transformer blocks, from which we identify key layers for analysis. Transformer architectures display fundamentally different response patterns characterized by distributed sensitivity across middle layers. In ViT-B/16 (Table 14), blocks 3–5 show consistent anomalous responses (0.3423, 0.3247, 0.2996) that all exceed the threshold of 0.2915. The ViT-B/32 architecture (Table 15) reveals similar distributed sensitivity, with blocks 3–4 showing the strongest deviations (0.3782, 0.3609), surpassing the threshold of 0.3298 by 14.7% and 9.4%, respectively. This pattern correlates with the attention mechanism's global dependency formation in intermediate

Table 15: Layer impact analysis results for ViT-B/32.

Layer	Impact	Selected Key Layer
visual.transformer.resblocks.0	0.0965	No
visual.transformer.resblocks.1	0.2025	No
visual.transformer.resblocks.2	0.3066	No
visual.transformer.resblocks.3	0.3782	**Yes**
visual.transformer.resblocks.4	0.3609	**Yes**
visual.transformer.resblocks.5	0.3085	No
visual.transformer.resblocks.6	0.2558	No
visual.transformer.resblocks.7	0.2628	No
visual.transformer.resblocks.8	0.2463	No
visual.transformer.resblocks.9	0.2172	No
visual.transformer.resblocks.10	0.2334	No
visual.transformer.resblocks.11	0.1367	No
Significance Threshold		0.3298
Mean		0.2504
Std		0.0794

layers, requiring defense strategies that monitor multiple blocks rather than single points of failure.

The statistical robustness of our $\mu + \sigma$ selection criterion is confirmed by consistent performance across all architectures, with all identified key layers showing notable deviations. The clear separation between normal and anomalous layers (minimum margin of 9.4%) demonstrates the method's reliability for architecture-agnostic backdoor analysis. These findings suggest that effective defense strategies must account for fundamental architectural differences—implementing focused final-layer monitoring for ResNets versus comprehensive multi-block analysis for Transformers.

E LIMITATIONS

While our study offers valuable insights, there are certain limitations that could be addressed in future research:

First, our analysis primarily focuses on the CLIP framework, examining various CLIP architectures such as RN50, RN101, ViT-B/16, and ViT-B/32. While these models are representative of the CLIP family, our work does not explore other large-scale multimodal learning architectures, which may exhibit different characteristics in terms of vulnerabilities or defense strategies. This focus on CLIP models leaves open the potential for discovering broader patterns across other architectures in future studies.

Second, our approach to adversarial sample generation relies on methods proposed by AdvCLIP [49]. While this provides a solid foundation for our analysis, the range of adversarial generation techniques employed could be expanded. Exploring alternative attack methodologies may offer a more comprehensive understanding of how different adversarial strategies interact with multimodal models, enriching the robustness of our findings.

These areas of future work suggest opportunities to broaden the scope of the research, providing a more holistic view of both multimodal architectures and adversarial generation methods.