Breaking the False Sense of Security in Backdoor Defense through Re-Activation Attack

Mingli Zhu[1] **Siyuan Liang**[2] **Baoyuan Wu**[1]*
[1]School of Data Science,
The Chinese University of Hong Kong, Shenzhen, Guangdong, 518172, P.R. China
[2]National University of Singapore, Singapore

Abstract

Deep neural networks face persistent challenges in defending against backdoor attacks, leading to an ongoing battle between attacks and defenses. While existing backdoor defense strategies have shown promising performance on reducing attack success rates, can we confidently claim that the backdoor threat has truly been eliminated from the model? To address it, we re-investigate the characteristics of the backdoored models after defense (denoted as defense models). Surprisingly, we find that the original backdoors still exist in defense models derived from existing post-training defense strategies, and the backdoor existence is measured by a novel metric called *backdoor existence coefficient*. It implies that the backdoors just lie dormant rather than being eliminated. To further verify this finding, we empirically show that these dormant backdoors can be easily re-activated during inference stage, by manipulating the original trigger with well-designed tiny perturbation using universal adversarial attack. More practically, we extend our backdoor re-activation to black-box scenario, where the defense model can only be queried by the adversary during inference stage, and develop two effective methods, *i.e.*, query-based and transfer-based backdoor re-activation attacks. The effectiveness of the proposed methods are verified on both image classification and multimodal contrastive learning (*i.e.*, CLIP) tasks. In conclusion, this work uncovers a critical vulnerability that has never been explored in existing defense strategies, emphasizing the urgency of designing more robust and advanced backdoor defense mechanisms in the future.

1 Introduction

The pervasive application of Deep Neural Networks (DNNs) across safety-critical domains like facial recognition and autonomous driving [23, 36] has underlined their significance and profound impact in industrial and academic spheres. Despite their transformative potential, DNNs are known to be vulnerable to malicious threats [5, 27], which compromise the integrity and reliability of advanced systems. One of the representative threats is backdoor attacks [18, 31], where an adversary pre-defines a "trigger" and embeds it within limited training data such that the backdoored model will misclassify trigger-containing inputs into specific target categories while appropriately processing benign inputs.

A successful backdoor attack consists of two stages: (1) the embedding of the backdoor within the model during training; and (2) its subsequent activation during inference stage [62]. To identify [14] and mitigate the harmful impacts of backdoor attacks, substantial efforts have been made ranging from dataset segmentation [7, 50], trigger inversion [53, 56], model pruning [64, 71], and fine-tuning based defenses [30, 67]. While these existing defense mechanisms aim at decreasing the attack success rates (ASR) [59] of corresponding backdoored models, a fundamental question arises: *can*

*Corresponds to Baoyuan Wu (wubaoyuan@cuhk.edu.cn).

38th Conference on Neural Information Processing Systems (NeurIPS 2024).

we confidently claim that the backdoor threat has truly been eliminated from the model? In this work, we use the term **defense model**(s) to denote those models which have initially been poisoned to backdoored models and subsequently defended using some defensive techniques, for convenience.

To answer above question, we introduce an innovative concept, *backdoor existence coefficient* (**BEC**) to quantify the extent of backdoor presence within models. Using BEC, we can re-investigate the backdoor existence in existing defense models [30, 58, 67]. Specifically, the BEC measures the similarity of activation among backdoor-related neurons in the poisoned samples between the backdoored model and its corresponding defense model. Fig. 1 presents the relationship between BEC and backdoor activation (indicated by ASR) across three different attack and defense methods for comparison. In this figure, distinct shapes and colors denote various attack and defense methods, respectively. As depicted in the figure, even though the ASRs decline nearly to zero which implies that defense models perform comparably to clean models, the BECs in the defense models remain significantly high. This notable observation implies that the original backdoors just lie dormant rather than being eliminated in defense models.

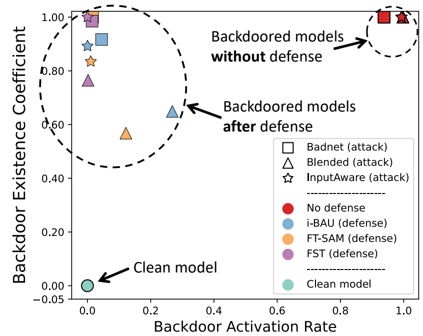

Figure 1: Comparative analysis of backdoor existence coefficient and backdoor activation rate across different models.

Inspired by above observations, we pose a question: Since the original trigger fails to activate the original backdoor, is it possible to unearth a variant of the original trigger that is capable of re-activating the backdoor? Given that in real-world scenarios where the adversary cannot modify the defense model, our objective is to modify the original trigger, thereby facilitating backdoor re-activation in defense models during inference stage. To verify this feasibility, we formulate the backdoor re-activation task as constrained optimization problem with the goal of searching for a minimal universal adversarial perturbation on the original trigger. Consequently, this general technique can be seamlessly combined with any prevailing backdoor attacks to re-activate backdoor effect in defense models in their inference time. To demonstrate the real-world threat posed by backdoor re-activation attack, we also expand our method to black-box and transfer attack scenarios, where adversaries are limited to querying the model without access to its internal mechanisms. Nowadays, multimodal contrastive learning (MMCL) has impressed us with its performance across a range of tasks and backdoor threats in MMCL have also been broadly studied. In this work, we consider both image classification and multimodal tasks, demonstrating the universality and adaptability of our approach. Extensive experimental results on nine different attacks and eight state-of-the-art defenses across four benchmark datasets and three model architectures demonstrate the effectiveness of our method. Our work reveals a new vulnerability in existing defense strategies, emphasizing the need for more robust and advanced defense mechanisms in the future.

Our main contributions are threefold: **1)** We re-investigate existing defense methods, and reveal that the original backdoor still exists in the model even after defense, though it cannot be activated by the original trigger. **2)** We develop a novel optimization problem to re-activate the original backdoor during inference by perturbing the original trigger, under white-box, black-box, and transfer attack scenarios. **3)** We demonstrate the effectiveness of the proposed method with extensive experiments on both image classification and the emerging multi-modal contrastive learning tasks.

2 Related work

Backdoor attacks. Backdoor attacks [15, 16, 22, 48, 55, 59, 75] are a significant security threat in DNNs. As summarized by Wu *et al.* [60, 62], a successful backdoor attack consists of two components: *backdoor injection* during pre-training or training stage, and *backdoor activation* during inference stage. Backdoor injection could be divided into data poisoning attack at pre-training stage and training-controllable attack at training stage. During a data poisoning attack, an adversary releases a poisoned dataset to plant backdoors. Representative works include BadNets [18], Blended [10], LF [68], SSBA [31], and Trojan [37]. For training-controllable attack [70], an adversary takes control of the training process to optimize triggers and inject backdoors. Notable examples are Input-Aware [40] and WaNet [41]. In inference stage, the adversary uses the poisoned samples to activate backdoors in the backdoored model, thereby achieving a successful attack.

While backdoor attacks are prevalent in supervised learning, backdoor threats also exist in domain of multi-modal contrastive learning (MMCL) [32, 33]. Carlini *et al.* [6] are the pioneers to unveil backdoor threats in MMCL, demonstrating that as few as 0.0001% of images can trigger a successful attack. More recently, sophisticated approaches have been introduced [2]. For instance, TrojanVQA [52] is designed for the multi-modal visual question answering task, while BadCLIP [35] shows that their attack can persist in effectiveness against backdoor defenses.

While a variety of attack methods have been proposed, they primarily focus on enhancing attack success rate during backdoor injection stage and employ the same trigger to activate backdoors in inference stage. They did not consider that the model might be fine-tuned or defended by users, and the original triggers fail to activate backdoors in inference stage. Although Qi *et al.* [42] attempted to enhance backdoor signal during inference stage, they did not consider defensive techniques in depth, and their attack lacks universality. In this work, we focus on a general backdoor attack method during inference time, researching on how to re-activate the dormant backdoors in defense models.

Backdoor defenses. A range of works [21, 24, 29, 39, 69] focusing on backdoor defenses have been put forward to address the threat of backdoor attacks. Considering defense stages, four main categories emerge: pre-processing defenses, training-stage defenses, post-training defenses, and inference stage defenses [61]. Pre-processing defenses [7, 24, 76] aim to filter out poisoned samples from poisoned dataset. Training-stage strategies [9, 21, 29, 57] consider that the defender has access to both training samples and the model, and mitigates backdoor effects during training process. They leverage discrepancies between poisoned and benign samples to filter out suspicious instances. Post-training defenses [11, 39, 54, 69, 73] focus on removing backdoor effect from backdoored models through pruning potential backdoor neurons [64], backdoor triggers reversion and unlearning [53], or enhancing fine-tuning processes for backdoor mitigation [72]. Inference stage defenses aim at preventing backdoor activation with samples detection or samples recovery techniques [76]. In the domain of MMCL, there are a range of works [3, 34, 65]. CleanCLIP [3] is the first to defend the MMCL model using MMCL loss and self-supervised learning within each modality with clean samples. Additionally, RoCLIP [66] introduces a robust pre-training approach, which focuses on disrupting the link between poisoned image-caption pairs. In this work, we focus on backdoor re-activation attack and thus mainly consider our attack against post-training backdoor defenses.

3 Methodology

In this section, we introduce our threat model and methods for image classification task for clarity. For the formulation and methods for multimodal contrastive learning, please refer to **Appendix** A.

3.1 Threat model

Notations. For the image classification task, the training dataset is $\mathcal{D} = \{(\boldsymbol{x}^{(i)}, y^{(i)})\}_{i=1}^{n} \subseteq \mathcal{X} \times \mathcal{Y}$, where $\mathcal{X} \subset \mathbb{R}^d$ and $\mathcal{Y} = \{1, \ldots, K\}$ are input space and label set, respectively. Given an input $\boldsymbol{x}$, we define a deep neural network with L layers as:

$$f(\boldsymbol{x}) = f^{(L)} \circ f^{(L-1)} \circ \cdots \circ f^{(1)}(\boldsymbol{x}), \tag{1}$$

where $f^{(l)}$ is the function in the l^{th} layer of the network, $1 \leq l \leq L$. The feature map of the l^{th} layer is denoted as $m^{(l)}(\boldsymbol{x}) \in \mathbb{R}^{c_l \times h_l \times w_l}$, and $f_k(\boldsymbol{x})$ represents the logit of the k^{th} class.

Before introducing our methods, we first outline the pipeline of backdoor attack and defense. As summarized in [62] and shown in Tab. 1, the whole pipeline of backdoor attack and defense involves four stages:

I. **Pre-training stage**: An adversary conducts data poisoning backdoor attack, which involves revising a small fraction of $\mathcal{D}$ to generate poisoned dataset $\mathcal{D}_p = \{(\boldsymbol{x}_{\boldsymbol{\xi}}^{(i)}, t)\}_{i=1}^{n_p}$ by injecting a trigger $\boldsymbol{\xi}$ into the image and changing the corresponding label into target label t.

II. **Training stage**: An adversary controls the training process to inject backdoors into model $f_{\boldsymbol{\theta}_{\text{A}}}$.

III. **Post-training stage**: A defender receives the poisoned model, and can gather some benign samples to remove the backdoor effect from the model, denoted as $f_{\boldsymbol{\theta}_{\text{D}}}$.

Table 1: Illustration of the pipeline of backdoor attack and defense.

Stage	Task description	Input/Output	Goal
Reference	Clean model training	$\mathcal{D}/f_{\boldsymbol{\theta}_{\text{C}}}$	$f_{\boldsymbol{\theta}_{\text{C}}}(\boldsymbol{x}) = y$, $f_{\boldsymbol{\theta}_{\text{C}}}(\boldsymbol{x}_{\boldsymbol{\xi}}) \neq t$
I: Pre-training & II: Training	Backdoor injection	$\mathcal{D}/f_{\boldsymbol{\theta}_{\text{A}}}, \mathcal{D}_p$	$f_{\boldsymbol{\theta}_{\text{A}}}(\boldsymbol{x}) = y$, $f_{\boldsymbol{\theta}_{\text{A}}}(\boldsymbol{x}_{\boldsymbol{\xi}}) = t$
III: Post-training	Backdoor defense	$f_{\boldsymbol{\theta}_{\text{A}}}/f_{\boldsymbol{\theta}_{\text{D}}}$	$f_{\boldsymbol{\theta}_{\text{D}}}(\boldsymbol{x}) = y$, $f_{\boldsymbol{\theta}_{\text{D}}}(\boldsymbol{x}_{\boldsymbol{\xi}}) \neq t$
IV: Inference	Backdoor re-activation	$\boldsymbol{x}, \boldsymbol{\xi}, f_{\boldsymbol{\theta}_{\text{D}}}/f_{\boldsymbol{\theta}_{\text{D}}}(\boldsymbol{x}_{\boldsymbol{\xi}'})$	$f_{\boldsymbol{\theta}_{\text{D}}}(\boldsymbol{x}) = y$, $f_{\boldsymbol{\theta}_{\text{D}}}(\boldsymbol{x}_{\boldsymbol{\xi}'}) = t$

IV. **Inference stage**: With the defense model $f_{\boldsymbol{\theta}_{\text{D}}}$, the original trigger fails to activate the backdoor, *i.e.*, $f_{\boldsymbol{\theta}_{\text{D}}}(\boldsymbol{x}_{\boldsymbol{\xi}}) \neq t$. The goal is to re-activate backdoors, *i.e.*, $f_{\boldsymbol{\theta}_{\text{D}}}(\boldsymbol{x}_{\boldsymbol{\xi}'}) = t$, where $\boldsymbol{\xi}' = \boldsymbol{\xi} + \Delta_{\boldsymbol{\xi}}$.

Existing backdoor attacks primarily focus on achieving high attack success rates (ASR) in backdoor injection stages (I and II), with little consideration for the defensive impact in stage III. *Given the failures of $\boldsymbol{x}_{\boldsymbol{\xi}}$ in attacking $f_{\boldsymbol{\theta}_{D}}$, our work focuses on the backdoor re-activation attack in stage IV.*

3.2 Backdoor existence coefficient

While the model performance in Tab. 1 suggests that $f_{\boldsymbol{\theta}_{\text{D}}}$ and $f_{\boldsymbol{\theta}_{\text{C}}}$ are analogous, we argue that in terms of the backdoor effect, $f_{\boldsymbol{\theta}_{\text{D}}}$ and $f_{\boldsymbol{\theta}_{\text{A}}}$ are actually more closely aligned, which indicates the persistent existence of backdoor in model $f_{\boldsymbol{\theta}_{\text{D}}}$. To verify this, we need a metric to measure the quantity of backdoor existence within a model. An effective indicator should be capable of quantifying the similarity of backdoor effect between backdoored model $f_{\boldsymbol{\theta}_{\text{A}}}$ and the target defense model $f_{\boldsymbol{\theta}_{\text{D}}}$ across the entire models. To achieve this, we propose a new metric, *Backdoor Existence Coefficient* (BEC), which is calculated through the following three steps:

1. **Backdoor neuron identification**: Firstly, we need to identify backdoor-related neurons. Zheng *et al.* [71] proposed Trigger-activated Change (TAC) to quantify the correlation between backdoor impact and neurons (see **Appendix** C for details). With this metric, backdoor-related neurons in $f_{\boldsymbol{\theta}_{\text{A}}}$ are identified for each layer. Thus, the feature maps corresponding to these neuron indices are selected for each model, denoted as $\tilde{m}^{(l)}_{\text{A}}(\boldsymbol{x}_{\boldsymbol{\xi}})$, $\tilde{m}^{(l)}_{\text{D}}(\boldsymbol{x}_{\boldsymbol{\xi}})$, and $\tilde{m}^{(l)}_{\text{C}}(\boldsymbol{x}_{\boldsymbol{\xi}})$, respectively. Denote the feature maps across dataset $\mathcal{D}_p$ as $\tilde{m}^{(l)}(\mathcal{D}_p) \in \mathbb{R}^{n_p \times (\tilde{c}_l \times h_l \times w_l)}$.
2. **Backdoor effect similarity metric**: In order to measure the backdoor effect similarity between models, we employ Centered Kernel Alignment (CKA) [25] (see **Appendix** C for details) to quantify the similarity between these matrices. The similarity in backdoor effects between $f_{\boldsymbol{\theta}_{\text{D}}}$ and $f_{\boldsymbol{\theta}_{\text{A}}}$, calculated through the use of corresponding features, can be computed as:
$$S^{(l)}_{\text{D,A}}(\mathcal{D}_p) = \text{CKA}\left(\tilde{m}^{(l)}_{\text{D}}(\mathcal{D}_p), \tilde{m}^{(l)}_{\text{A}}(\mathcal{D}_p)\right), \tag{2}$$
and $S^{(l)}_{\text{C,A}}(\mathcal{D}_p)$ is computed accordingly.
3. **Backdoor existence coefficient computation**: The BEC is the average of normalized backdoor effect similarity across all layers. By assigning the BEC of $f_{\boldsymbol{\theta}_{\text{A}}}$ a value of 1 and $f_{\boldsymbol{\theta}_{\text{C}}}$ a value of 0, the computation can proceed as follows:
$$\rho_{\text{BEC}}(f_{\boldsymbol{\theta}_{\text{D}}}, f_{\boldsymbol{\theta}_{\text{A}}}, f_{\boldsymbol{\theta}_{\text{C}}}; \mathcal{D}_p) = \frac{1}{N}\sum_{l=1}^{N} \frac{S^{(l)}_{\text{D,A}}(\mathcal{D}_p) - S^{(l)}_{\text{C,A}}(\mathcal{D}_p)}{S^{(l)}_{\text{A,A}}(\mathcal{D}_p) - S^{(l)}_{\text{C,A}}(\mathcal{D}_p)} \in [0, 1]. \tag{3}$$

Remark. $S^{(l)}_{\text{A,A}}(\mathcal{D}_p) = 1$. The second and third arguments in ρ_{BEC} serve as two reference models to measure the backdoor existence of the model corresponding to the first argument $f_{\boldsymbol{\theta}_{\text{D}}}$. Denote $\rho_{\text{BEC}}(f_{\boldsymbol{\theta}_{\text{D}}}, f_{\boldsymbol{\theta}_{\text{A}}}, f_{\boldsymbol{\theta}_{\text{C}}}; \mathcal{D}_p)$ as $\rho_{\text{BEC}}(f_{\boldsymbol{\theta}_{\text{D}}})$ for simplicity. The higher the value $\rho_{\text{BEC}}(f_{\boldsymbol{\theta}_{\text{D}}})$, the stronger the existence of backdoors in the model. We utilize BEC to signify backdoor existence and employ ASR to quantify the extent of backdoor activation. As shown in Fig. 1, the BEC remains consistently high across various defenses, despite backdoor activation being low.

3.3 Backdoor re-activation attack

Motivated by the fact analyzed above that the original backdoor still exists in the defense model $f_{\boldsymbol{\theta}_{\text{D}}}$, here we explore the possibility to re-activate the backdoor during inference. Since the adversary cannot modify $f_{\boldsymbol{\theta}_{\mathrm{D}}}$ during inference, one feasible solution is to modify the original trigger $\boldsymbol{\xi}$. Specifically, we propose to pursue a new trigger $\boldsymbol{\xi}'$ by perturbing $\boldsymbol{\xi}$, *i.e.*, $\boldsymbol{\xi}' = \boldsymbol{\xi} + \Delta_{\boldsymbol{\xi}}$, such that $\boldsymbol{\xi}'$ could re-activate the original backdoor, *i.e.*, $f_{\boldsymbol{\theta}_{\mathrm{D}}}(\boldsymbol{x}_{\boldsymbol{\xi}'}) = t$. In the following, we will present how to obtain a successful trigger perturbation $\Delta_{\boldsymbol{\xi}}$ under white-box, black-box, and transfer attack scenarios, respectively.

White-box backdoor re-activation attack. In white-box scenario, the adversary has access to the parameters of f but cannot manipulate them. In this case, we could obtain $\Delta_{\boldsymbol{\xi}}$ by solving the constrained optimization problem $\min_{\|\Delta_{\boldsymbol{\xi}}\|_p \le \rho} \mathcal{L}_{tot}(\Delta_{\boldsymbol{\xi}}; \mathcal{D}_p, f)$, where

$$\mathcal{L}_{tot}(\Delta_{\boldsymbol{\xi}}; \mathcal{D}_p, f) = \sum_{(\boldsymbol{x}_{\boldsymbol{\xi}}, t) \in \mathcal{D}_p} \mathcal{L}_{\mathrm{CE}}(f(\boldsymbol{x}_{\boldsymbol{\xi}+\Delta_{\boldsymbol{\xi}}}), t) - \lambda \log \left(1 - \max_{k \neq t} \frac{e^{f_k(\boldsymbol{x}_{\boldsymbol{\xi}+\Delta_{\boldsymbol{\xi}}})}}{\sum_{i=1}^{N} e^{f_i(\boldsymbol{x}_{\boldsymbol{\xi}+\Delta_{\boldsymbol{\xi}}})}} \right), \quad (4)$$

where $\|\cdot\|_p$ means ℓ_p norm, ρ is the perturbation bound, $\mathcal{L}_{\mathrm{CE}}$ is cross-entropy loss, and $\lambda > 0$ is a hyper-parameter. This problem can be easily solved using project gradient descent (PGD) [38].

Black-box backdoor re-activation attack. Although the re-activation attack under the white-box scenario is easy to implement, it may be impractical. Thus, we also consider the practical black-box scenario, where the adversary lacks information to the defense model and can only query the model and obtain the predicted score. Consequently, the above problem (4) is no longer directly optimized by the PGD algorithm. Inspired by existing black-box adversarial attacks [1, 8], we propose a novel random search based optimization algorithm. Specifically, we extend the query-based black-box adversarial attack method Square Attack [1] that was designed for optimizing sample-specific perturbation, to solve problem (4), dubbed Universal Square Attack. Its overall procedure is summarized in Alg. 1 in **Appendix**.

Transfer-based backdoor re-activation attack. In addition to the query-based black-box attack, we also explore transfer-based attack scenario. In this scenario, the adversary trains a backdoored model $f_{\boldsymbol{\theta}_{\mathrm{A}}}$ and releases it to downstream users. The user receives model $f_{\boldsymbol{\theta}_{\mathrm{A}}}$, and obtains a defense model $f_{\boldsymbol{\theta}_{\mathrm{D}}}$ based on $f_{\boldsymbol{\theta}_{\mathrm{A}}}$ by some post-training defense. Thus, the adversary does not know the exact defense method, but has full information about the original trigger $\boldsymbol{\xi}$ and model $f_{\boldsymbol{\theta}_{\mathrm{A}}}$ which has same model architecture as $f_{\boldsymbol{\theta}_{\mathrm{D}}}$. The adversary also has restricted query limits. Consequently, leveraging transfer attacks becomes a viable strategy for attacking. The main idea is that the adversary can imitate defense process to get some defense models $f_{\boldsymbol{\theta}_{\mathrm{D}_i}}$ themselves, where $i = 1, \dots, M$. Then these defense models can serve as surrogate models to generate perturbation $\Delta_{\boldsymbol{\xi}}$ as follows:

$$\Delta^*_{\boldsymbol{\xi}} = \underset{\|\Delta_{\boldsymbol{\xi}}\|_p \le \rho}{\arg\min} \sum_{i=1}^{M} \mathcal{L}_{tot}(\Delta_{\boldsymbol{\xi}}; \mathcal{D}_p, f_{\boldsymbol{\theta}_{\mathrm{D}_i}}). \quad (5)$$

Overall, we propose a universal backdoor re-activation attack that aims to enhance the performance of existing backdoor attack methods during inference. We have explored three scenarios—white-box attack (WBA), query-based black-box attack (BBA), and transfer attack (TA). Besides, we would like to emphasize again that the proposed attack can be naturally extended to multi-modal learning tasks, other than the classification task demonstrated above. The details are presented in **Appendix** A.

4 Experiments

4.1 Implementation details

Models and datasets. For image classification task, we evaluate all our attacks on three benchmark datasets CIFAR-10 [26], Tiny ImageNet [28], and GTSRB [49] over two network architectures, PreAct-ResNet18 [20] and VGG19-BN [47]. We utilize the setup in BackdoorBench [59]. For MMCL task, we use the open-sourced CLIP model from OpenAI [44] as the pre-trained model. Following the setting of CleanCLIP [3], the model is poisoned on the CC3M dataset [45] and subsequently tested through zero-shot evaluation on ImageNet-1K validation set [13].

Backdoor attacks. For image classification task, we adopt seven widely used backdoor attacks including: (1) five data poisoning attack: BadNets [18], Blended [10], LF [68], SSBA [31], and

Trojan [37]; and (2) two training-controllable attacks: Input-Aware [40] and WaNet [41]. We follow the default attack configuration as in BackdoorBench [59] and the 0^{th} label is set to be the target label. For MMCL task, we adopt four backdoor attacks including: BadNets, Blended, SIG [4], and TrojanVQA [52]. In data poisoning phase, 1500 samples out of 500K image-text pairs from CC3M dataset are poisoned and the target label is `banana` as in [3].

Backdoor defenses. For image classification task, we adopt six state-of-the-art post-training defense methods: NC [53], NAD [30], i-BAU [67], FT-SAM [72] , SAU [58], and FST [39]. For MMCL task, we consider two defense methods: (1) FT [3]: fine-tuning the model with multimodal contrastive loss using clean dataset; and (2) CleanCLIP [3]: a fine-tuning defense method for CLIP models. All the detailed introduction about the above attack and defense methods can be found in **Appendix** D.

Implementation details. At backdoor injection phase, the poisoning ratio is set to 10%, following the configuration in BackdoorBench [59]. At defense phase, 5% clean samples are given to defend models. At backdoor re-activation phase, we consider defense models as our target model. The adversary is given 2% (*i.e.*, 1000) poisoned samples to conduct attacks. We consider both ℓ_∞ and ℓ_2 norm attacks, and the perturbation bounds are set to 0.05 and 2, respectively. The loss hyper-parameter λ is 1 for all our experiments. For query-based black-box attack, the maximum query limit is 10,000 for each image. For transfer attack, the adversary is given 10% poisoned samples to conduct backdoor re-activation attack. The ℓ_2 norm bound is set to 1 for transfer attack. We simply assume three surrogate models can be used and we just divided these defenses into two groups: (1) NC, NAD, i-BAU; and (2) FT-SAM, SAU, FST. Specifically, we generate perturbation in each group and test the ASRs in the other group. All the ASRs are tested on testing dataset. For MMCL tasks, we assume the adversary lacks knowledge of the downstream task. Therefore, attacks are executed in the upstream task for both white-box and transfer attacks and subsequently tested in downstream zero-shot task. More details about the implementations can be found in **Appendix** D. We have provided the PyTorch[2] implementation of our method on Github.

4.2 Main results

Backdoor re-activation attack. Tab. 2 shows the performance of our backdoor re-activation attack under white-box attack (**WBA**) and query-based black-box attack (**BBA**) settings in comparison with ASRs of original backdoored models (**No Defense**) and defense models (**Defense**). By observing the table, the following profound insights emerge: **(1)** Compared to defense models, our attacks show a striking level of efficacy. Both our WBA and BBA have exhibited an impressively absolute improvement of 76.94% and 42.95% on average, respectively when compared against defense mechanisms, which shows the effectiveness of our re-activation attack method. **(2)** The close performance of our WBA compared to "No Defense" underscores the efficacy of our backdoor re-activation mechanism, affirming the recoverability of the backdoors in defense models. By setting WBA as an upper bound for backdoor recovery, the more realistic BBA reveals substantial attack performance. Despite a gap between the two approaches, we posit that this disparity can be lessened through a sophisticated black-box attack strategy. **(3)** In terms of specific defenses, our attack against SAU and FST exhibits relatively poor ASRs. This suggests that SAU's backdoor removal efficiency is significant, which aligns with the subsequent analysis of Fig. 3. In contrast, FST's BBA seems comparatively subdued. It may be attributed to the reinitialized FC layers, effectively cutting backdoor activations. These insights serve as valuable pointers for crafting defense strategies in the future.

Backdoor re-activation attack via transfer attack. In this experiment, we group these defenses into two distinct groups: (1) weak group (NC, NAD, i-BAU) and (2) strong group (FT-SAM, SAU, FST) to better to observe the impact of defense methods on the performance of transfer-based re-activation attacks (**TA**). Two key findings emerged from results in Tab. 3: **(1)** Transfer attacks generally exhibit strong performance in comparison with results in Tab. 2. The ensemble attack strategies applied on the weak group demonstrate better attack effectiveness on strong defense models than that in BBAs. **(2)** Utilizing ensemble strategies on strong defense methods results in remarkably effective ASRs on weak defense models, surpassing even the efficacy of WBA in Tab. 2. This outcome raises concerns: if adversaries simulate stronger defenses to derive substitute models for launching transfer attacks, it could lead to serious security threats.

[2] https://github.com/JulieCarlon/Backdoor-Reactivation-Attack

Table 2: Performance (%) of backdoor re-activation attack on both white-box (WBA) and black-box (BBA) scenarios with ℓ_∞-norm bound $\rho = 0.05$ against different defenses with CIFAR-10 on PreAct-ResNet18. The best results are highlighted in **boldface**.

Attacks	No Defense	NC [53]			NAD [30]			i-BAU [67]			FT-SAM [72]			SAU [58]			FST [39]		
		Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA
BadNets [18]	93.79	2.01	**96.78**	27.91	1.96	**94.78**	49.66	4.48	**97.42**	54.37	1.63	**94.71**	51.23	1.30	**93.10**	37.91	1.46	**97.93**	42.69
Blended [10]	99.76	99.76	**99.93**	99.13	47.64	**99.82**	14.14	26.83	**99.63**	85.80	12.17	**99.56**	87.29	5.20	**98.37**	73.06	0.20	**99.62**	82.97
Input-Aware [40]	99.30	0.70	**92.04**	54.33	0.92	**93.80**	70.44	0.02	**21.78**	19.56	1.07	**96.19**	80.16	1.26	**85.39**	22.26	0.00	**90.72**	44.65
LF [68]	99.06	99.06	**99.41**	80.51	75.47	**99.41**	17.01	11.99	**99.04**	75.48	6.43	**97.40**	89.28	2.49	**90.74**	23.08	5.43	**98.18**	1.16
SSBA [31]	97.07	97.07	**99.90**	94.38	70.77	**99.72**	88.53	2.89	**91.29**	70.71	4.06	**92.80**	69.18	2.16	**89.86**	38.59	0.54	**94.11**	52.71
Trojan [37]	99.99	2.76	**95.26**	45.57	5.77	**96.38**	60.87	0.54	**89.58**	40.18	4.12	**96.18**	69.88	1.39	**87.61**	47.37	8.93	**97.28**	80.47
WaNet [41]	98.90	98.90	**100.00**	99.64	0.73	**96.21**	77.65	0.88	**94.67**	75.91	0.96	**94.95**	78.66	0.82	**95.33**	60.36	0.26	**97.56**	82.22
Avg	98.26	57.18	**97.62**	71.64	29.04	**97.16**	54.04	6.80	**84.77**	60.29	4.35	**95.97**	75.10	2.09	**91.48**	43.23	2.40	**96.49**	55.27

Table 3: Attack performance (%) on target models of transfer-based re-activation attack (TA) with ℓ_2-norm bound $\rho = 1$ against different defenses with CIFAR-10 on PreAct-ResNet18.

Attack	No Defense	NC [53]		NAD [30]		i-BAU [67]		FT-SAM [72]		SAU [58]		FST [39]	
		Defense	TA	Defense	TA	Defense	TA	Defense	TA	Defense	TA	Defense	TA
BadNets [18]	93.79	2.01	95.43	1.96	98.42	4.48	97.90	99.17	97.42	1.30	90.17	1.46	96.21
Blended [10]	99.76	99.76	100.00	47.64	99.98	26.83	99.83	98.64	99.63	5.20	93.36	0.20	24.07
Input-Aware [40]	99.30	0.70	99.98	0.92	99.98	0.02	99.77	96.92	21.78	1.26	15.56	0.00	95.28
LF [68]	99.06	99.06	99.93	75.47	99.84	11.99	98.35	93.85	99.04	2.49	96.62	5.43	80.09
SSBA [31]	97.07	97.07	99.27	70.77	99.38	2.89	20.44	98.06	91.29	2.16	95.03	0.54	76.21
Trojan [37]	99.99	2.76	99.76	5.77	99.09	0.54	96.18	96.57	89.58	1.39	83.67	8.93	21.79
WaNet [41]	98.90	98.90	99.72	0.73	99.86	0.88	83.79	98.90	94.67	0.82	89.49	0.26	98.69
Avg	98.26	57.18	99.16	29.04	99.51	6.80	85.18	97.44	84.77	2.09	80.55	2.40	70.33

Effectiveness of attacks on CLIP models. Tab. 4 lists the performance of our backdoor re-activation attack under white-box attack (**WBA**) and transfer-based attack (**TA**) on the CLIP model. Our attacks yield significant improvements, with ASR enhancements of 34.87% and 43.35% on average, respectively, compared to defense models. The results for TA and WBA are very close. One possible reason is that the similarity between the FT and CleanCLIP methods leads to strong transfer performance. We advocate for the development of stronger defenses on CLIP to combat attacks. Due to space constraints, attack results and analysis on Tiny ImageNet (Tab. 12) and GTSRB (Tab. 13) datasets, and results on VGG19-BN models (Tab. 14) are provided in **Appendix** E.

4.3 Ablation study

Influence of norm bound and norm type. We studied the impact of norm type and norm bound on the attack performance. The results are shown in (a) and (b) of Fig. 2. It can be observed that it is difficult to achieve high success rates under smaller norm bounds. However, when the norm bound is sufficiently large, the attack effectiveness converges and approaching nearly 100% for both ℓ_∞-norm and ℓ_2-norm types against all defense models.

Influence of the size of poisoned samples. We investigated the impact of the size of poisoned samples on attack performance for Blended attack. As shown in (c) and (d) of Fig. 2, increasing the number of training samples in WBA shows significant improvement in attack results. However, in

Table 4: Performance (%) of our attack on both white-box (WBA) and transfer-based (TA) attacks with ℓ_∞-norm bound $\rho = 0.05$ against different defenses with ImageNet-1K on CLIP. Best results are highlighted in **boldface**.

Attack	No Defense	FT [3]			CleanCLIP [3]		
		Defense	WBA	TA	Defense	WBA	TA
BadNets [18]	96.65	64.60	82.05	**82.73**	17.29	**57.76**	47.30
Blended [10]	97.71	49.77	96.57	**98.64**	18.57	**89.61**	72.65
SIG [4]	77.71	30.91	**92.56**	87.99	21.68	**87.04**	82.55
TrojanVQA [52]	98.21	82.07	97.14	**97.46**	49.82	**87.43**	78.25
Avg	92.57	56.84	**92.08**	91.71	26.84	**80.46**	70.19

Table 5: Our attacks (%) on defense models in comparison with clean ones with ℓ_∞-norm bound $\rho = 0.05$ under different model structures and datasets.

Setup	Clean Model		Defense Model	
	WBA	BBA	WBA	BBA
Res18+CIFAR-10	85.00	56.98	93.92	59.93
Res18+Tiny	39.76	14.02	71.04	40.81
Res18+GTSRB	53.33	50.87	67.14	61.81
VGG+CIFAR-10	68.80	43.60	85.15	51.04
Avg	61.72	41.37	79.31	53.40

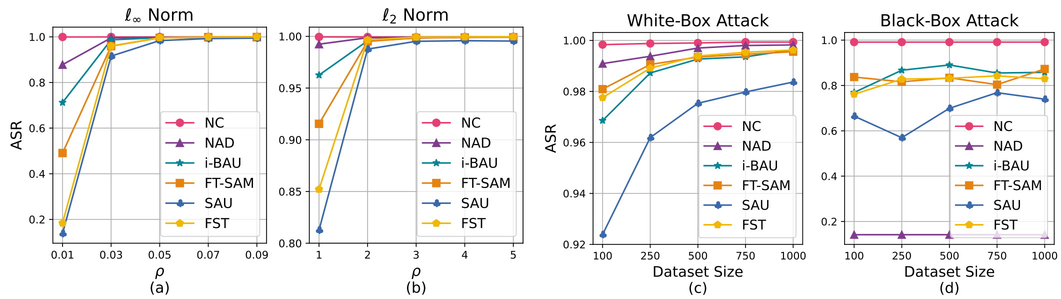

Figure 2: (a) and (b) show attack results under different norm types p and bounds ρ for WBA. (c) and (d) show attack results under different number of poisoned samples for WBA and BBA.

Table 6: Detection performance (TPR %) on different $\langle$model, poisoned samples$\rangle$ pairs.

Attack ↓	**Detection ↓**	$f_{\mathrm{A}}, \mathcal{D}_p$	$f_{\mathrm{D,FT\text{-}SAM}}, \mathcal{D}_p$	$f_{\mathrm{D,FT\text{-}SAM}}, \mathcal{D}_{p,\Delta\xi}$	$f_{\mathrm{D,SAU}}, \mathcal{D}_p$	$f_{\mathrm{D,SAU}}, \mathcal{D}_{p,\Delta\xi}$
BadNets	SCALE-UP	39.6	79.6	68.6	79.5	49.5
	SentiNet	37.7	3.6	2.2	0.2	0.9
	STRIP	88.3	0.7	5.5	10.3	6.5
Trojan	SCALE-UP	92.6	84.9	73.1	81.6	55.4
	SentiNet	2.9	1.1	1.05	2.1	1.5
	STRIP	99.9	1.9	29.8	4.2	1.2

Table 7: Performance (%) against test-time defenses.

Defense → **Attack ↓**	**SCALE-UP** **ASR**	**SCALE-UP** **ACC**	**STRIP** **ASR**	**STRIP** **ACC**	**ZIP** **ASR**	**ZIP** **ACC**
BadNets	29.8	53.7	83.4	9.3	23.6	80.7
Blended	34.1	46.2	49.2	9.2	48.1	81.5

the BBA setting, the ASRs remains relatively stable and does not exhibit significant enhancements with the increase of training samples. This suggests that the difficulty in BBA lies in finding a good universal perturbation, especially when dealing with a large number of training samples. However, the successful attacks with minimal samples also highlight the significant potency of the attack method.

Attacks performance against clean models. To demonstrate the specific vulnerability of defense models, we contrast the performance of our attacks on the defense models in comparison with clean models. Tab. 5 provides a summary of our method's performance across all backdoor attacks and defense methods, in comparison of the ASRs on clean models. It can be observed that, although some effectiveness is achieved on the clean models, the vulnerability of defense models is significantly higher than that of the clean model, with this gap being more pronounced in particular defenses. This indicates that defense models are indeed more fragile in comparison with clean models.

4.4 Further analysis

Backdoor existence analysis. We provide more experimental demonstration on the existence of backdoors in defense models. We employ our BEC metric to quantify the existence of backdoors in all defense models and visualize the relationship between BEC and backdoor activation rates, as depicted in (a) of Fig. 3. We observe that backdoors persist across defense models, albeit with low backdoor activation rates. The BECs in SAU, SAM, and i-BAU are relatively low, while FST exhibits a notably high BEC. This contrast may stem from the former's optimization objectives resembling adversarial training, whereas the latter primarily disrupts activations through layers re-initialization.

Relationship between BECs and ASRs. We validate the relationship between the ASR of re-activation attack (WBA) and the residual of backdoors. We computed the Pearson Correlation Coefficients (PPC) between BECs of different defense models and their white-box ASRs among all attacks, as shown in (b) of Fig. 3. It is evident that in most cases, there is a strong correlation between the two. In other words, the more backdoors remain in models, the easier it is for attacks to succeed. Therefore, our metrics can serve as an indicator of backdoored model security.

Feature map visualization. Here we visualize the feature maps between different models to directly observe their similarities. Fig. 3 (c) displays the visualizations of activations from the final four convolutional layers of three models, sorted in descending order according to backdoored model's TAC value, with each subplot arranged from top to bottom. It can be observed that the defense model and backdoored model exhibit similar patterns: highlighting activations in backdoor-related neurons. This directly indicates the persistence of backdoors within defense models.

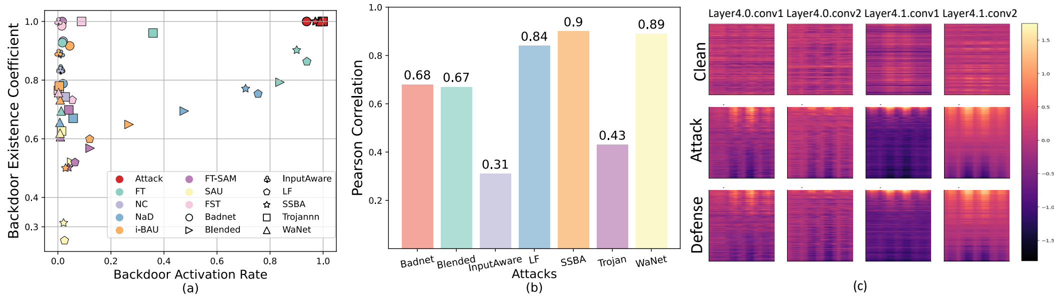

Figure 3: (a).Visualization of the correlation between backdoor activation rate and BEC. (b). Pearson correlation coefficients of ASR and BEC under different attacks. (c). Visualization of feature maps.

Attack against test-time detection and defenses. Given that our attack is conducted during the test phase, it is essential to assess whether it can evade backdoor detection and defenses at test phase. To this end, we test three test-time backdoor detection methods: SCALE-UP [19], SentiNet [12], and STRIP [17], as well as three test-time defenses: STRIP [17], ZIP [46], and SCALE-UP [19].

The detection task requires two input arguments, including the model and the query datasets. We evaluate five pairs, including $\langle$the original backdoored model f_{A}, the original poisoned dataset $\mathcal{D}_p\rangle$, $\langle$the defense model with FT-SAM $f_{\text{D,FT-SAM}}, \mathcal{D}_p\rangle$, $\langle f_{\text{D,FT-SAM}}$, the re-activation dataset $\mathcal{D}_{p,\Delta\xi}\rangle$, $\langle$the defense model with SAU $f_{\text{D,SAU}}, \mathcal{D}_p\rangle$, $\langle f_{\text{D,SAU}}$, the re-activation dataset $\mathcal{D}_{p,\Delta\xi}\rangle$. The result in Tab. 6 shows that our attacks do not markedly increase the TPR compared to the other two pairs. More detection performance on our BBA and TA are shown in Tab. 18 in Appendix.

Tab. 7 shows the defense results. It shows that our attack maintains a certain level of ASR against ZIP. However, for SCALE-UP and STRIP, there is a significant drop in ASR. Meanwhile, the model's ACC is also notably low. This experiment highlights the potential for future attack method designs aimed at evading test-time defenses. Possible strategies could include techniques to better align with feature distributions of clean data and to avoid triggering excessively strong activations.

Attack against adaptive defense. Considering defenders are aware of adversary' strategies, they can introduce random perturbations for queries so as to disrupt the adversary's ability. We assess both adversary's ASR and the model accuracy on clean samples under varying perturbation bound. As depicted in Tab. 8, minor noise has slight impact on ASR. However, with larger noise amplitudes, despite failed attacks, the model's accuracy is significantly affected.

Table 8: Results (%) against adaptive defense.

Defense → / Noise ↓	FST [39] ACC	FST [39] ASR	FT-SAM [72] ACC	FT-SAM [72] ASR	i-BAU [67] ACC	i-BAU [67] ASR	SAU [58] ACC	SAU [58] ASR
0.00	92.61	82.97	92.88	87.29	89.43	85.80	91.75	73.06
0.01	91.64	82.24	92.13	90.89	88.85	77.88	91.31	71.99
0.02	88.35	70.89	89.53	88.04	86.15	79.02	88.38	84.11
0.03	82.59	73.03	84.84	86.36	81.32	75.83	83.33	67.42
0.04	75.87	56.76	78.04	81.60	75.51	72.72	76.19	54.40
0.05	67.15	58.17	70.12	74.57	68.09	72.28	67.95	56.74

4.5 Comparison among OBA, RBA, and gUAA

To verify that our re-activation attack method finds a highly correlated backdoor with the original backdoor, and to distinguish it from general universal adversarial perturbation attack (gUAA), we systematically compare the original backdoor attack (OBA), our re-activation attack (RBA), and gUAA from three key perspectives.

Table 9: CKA scores between OBA, RBA, and gUAA.

Defense ⇒ / **Attack** ↓	**i-BAU** $S_{\text{RBA,OBA}}$	**i-BAU** $S_{\text{gUAA,OBA}}$	**i-BAU** $S_{\text{RBA,gUAA}}$	**FT-SAM** $S_{\text{RBA,OBA}}$	**FT-SAM** $S_{\text{gUAA,OBA}}$	**FT-SAM** $S_{\text{RBA,gUAA}}$
BadNets	0.607	0.192	0.170	0.599	0.194	0.169
Blended	0.712	0.196	0.192	0.712	0.197	0.193

To facilitate the understanding of our analysis, we firstly clarify the definitions and settings. **OBA** refers to an existing backdoor attack following the standard backdoor injection and activation process; **RBA** means that, given the defense model, we aim to re-activate the injected backdoor of OBA by searching for a new trigger $\boldsymbol{\xi}'$, starting from original trigger $\boldsymbol{\xi}$, based on some original poisoned samples $\mathcal{D}_p$; **gUAA** refers to a targeted universal adversarial perturbation attack (same class as OBA and RBA) where, given $f_{\boldsymbol{\theta}_{\text{D}}}$, we aim to find a perturbation starting from clean samples $\mathcal{D}_c$. The searched UAP is denoted as Δ, and the perturbed dataset as $\mathcal{D}_{c,\Delta}$. Our analyses are as follows:

Table 10: ASR (%) of RBA and gUAA with different query numbers.

Attack+Defense	Query number ⇒	1000	3000	5000	7000
Blended+i-BAU	RBA	77.3	89.3	92.1	94.6
	gUAA	14.2	41.4	49.5	56.4
Blended+FT-SAM	RBA	41.1	77.4	79.8	85.6
	gUAA	16.3	42.2	56.5	65.5

Table 11: ASR (%) of OBA, RBA, and gUAA under different l_∞-norm of random noise.

	Norm ⇒	0	0.03	0.06	0.09
OBA	Blended+NAD	99.8	99.8	99.6	97.3
	LF+NAD	99.1	98.9	98.4	98.6
RBA	Blended+NAD	99.8	99.7	98.7	84.0
	LF+NAD	99.4	99.1	98.1	96.6
gUAA	Blended+NAD	95.5	92.7	79.4	35.4
	LF+NAD	96.5	89.5	55.8	16.7

- **Activation mechanism of backdoor effect:** We analyze the backdoor activation mechanism in each attack. As demonstrated in Sec. 3.2, we adopt the CKA metric to measure backdoor effect similarity between models. Here we calculate the following three CKA scores: $S_{\text{RBA,OBA}} = \frac{1}{N}\sum_{l=1}^{N} \text{CKA}(\tilde{m}_{\text{D}}^{(l)}(\mathcal{D}_{p,\Delta_\xi}), \tilde{m}_{\text{A}}^{(l)}(\mathcal{D}_p))$, $S_{\text{gUAA,OBA}} = \frac{1}{N}\sum_{l=1}^{N} \text{CKA}(\tilde{m}_{\text{D}}^{(l)}(\mathcal{D}_{c,\Delta}), \tilde{m}_{\text{A}}^{(l)}(\mathcal{D}_p))$, $S_{\text{RBA,gUAA}} = \frac{1}{N}\sum_{l=1}^{N} \text{CKA}(\tilde{m}_{\text{D}}^{(l)}(\mathcal{D}_{p,\Delta_\xi}), \tilde{m}_{\text{D}}^{(l)}(\mathcal{D}_{c,\Delta}))$. As shown in Tab. 9, $S_{\text{RBA,OBA}} \gg S_{\text{gUAA,OBA}} \approx S_{\text{RBA,gUAA}}$ across all attack-defense pairs. This demonstrates that **the backdoor activation mechanisms between RBA and OBA are highly similar, and both differ significantly from that of gUAA**.
- **Starting from the original trigger ξ, it is easier and faster to find a new trigger ξ' that achieves a high attack success rate (ASR):** As shown in Tab. 10, given the same query numbers, the ASR of RBA is much higher than that of gUAA, and RBA increases in speed faster than gUAA. This indicates that **RBA is much closer to OBA than gUAA**.
- **Compared to Δ, both the original trigger ξ and the new trigger ξ' are more robust to random noise:** We discovered that the robustness to random noise can distinguish the trigger of an intended backdoor from the trigger of a natural backdoor (*i.e.*, gUAP). Specifically, we perturb ξ, ξ', and Δ with the same level of random noise and record the ASR of these attacks. As shown in Tab. 10, both OBA and RBA are more robust than gUAA. This confirms that RBA produces an intended backdoor trigger similar to OBA, rather than a gUAP.

In conclusion, our analyses verify that *our RBA method finds a backdoor highly correlated with the original backdoor, rather than a less correlated one (new backdoor) or a general UAP (natural backdoor)*. Thus, we assert that **our RBA effectively re-activates the original backdoor**.

5 Conclusion

This paper illuminates the false sense of security in backdoor defenses and proposes a new threat to enhance existing backdoor attacks in inference-time. Our pioneering introduction of the backdoor existence coefficient unveils the residual presence of backdoors within defense models. Moreover, we propose a novel optimization problem to re-activate these dormant backdoors and craft distinct algorithms tailored specifically to white-box, black-box, and transfer attack scenarios. The proposed method can be integrated with existing backdoor attacks to boost their attack success rate during the inference stage. The efficacy of our method is evidenced through exhaustive evaluation on both image classification and multi-modal contrastive learning tasks. The threat revealed by this study underscores the pressing need for designing advanced defense mechanisms in the future.

Limitations and future work. Despite the efficacy of our proposed method, its effectiveness is limited when confronted with defenders that inject noise into each query. Promising future work is to devise more sophisticated attacks that can bypass this defenses. Another limitation is that if defenders aim to decrease both ASR and BEC, our attacks will become challenging, even though directly optimizing the BEC is not feasible. This serves as another direction for our future work.

Broader Impacts. As deep neural networks sourced from untrusted origins face significant risks from backdoor attacks, this study provides a meaningful exploration into the false security in backdoor defense models. This could spark further advancements in backdoor defenses. Nonetheless, the potential misuse by ill-intended entities should be cautiously considered.

6 Acknowledgments

This work is supported by Guangdong Basic and Applied Basic Research Foundation (No. 2024B1515020095), National Natural Science Foundation of China (No. 62076213), Shenzhen Science and Technology Program under grants (No. RCYX20210609103057050), Longgang District Key Laboratory of Intelligent Digital Economy Security, the National Research Foundation, Singapore, and the CyberSG R&D Programme Office ("CRPO"), under the National Cybersecurity R&D Programme ("NCRP"), RIE2025 NCRP Funding Initiative (Award CRPO-GC1-NTU-002).

References

[1] Maksym Andriushchenko, Francesco Croce, Nicolas Flammarion, and Matthias Hein. Square attack: a query-efficient black-box adversarial attack via random search. In *European Conference on Computer Vision*, 2020. 5, 17, 21

[2] Jiawang Bai, Kuofeng Gao, Shaobo Min, Shu-Tao Xia, Zhifeng Li, and Wei Liu. Badclip: Trigger-aware prompt learning for backdoor attacks on clip. In *CVPR*, 2024. 3

[3] Hritik Bansal, Nishad Singhi, Yu Yang, Fan Yin, Aditya Grover, and Kai-Wei Chang. Cleanclip: Mitigating data poisoning attacks in multimodal contrastive learning. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pages 112–123, 2023. 3, 5, 6, 7, 16, 19, 20, 21

[4] Mauro Barni, Kassem Kallas, and Benedetta Tondi. A new backdoor attack in cnns by training set corruption without label poisoning. In *International Conference on Image Processing*, 2019. 6, 7, 20

[5] Battista Biggio and Fabio Roli. Wild patterns: Ten years after the rise of adversarial machine learning. In *Proceedings of the 2018 ACM SIGSAC Conference on Computer and Communications Security*, pages 2154–2156, 2018. 1

[6] Nicholas Carlini and Andreas Terzis. Poisoning and backdooring contrastive learning. In *International Conference on Learning Representations*, 2022. 3

[7] 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. In *Workshop on Artificial Intelligence Safety*, 2019. 1, 3

[8] Jinghui Chen and Quanquan Gu. Rays: A ray searching method for hard-label adversarial attack. In *Proceedings of the 26th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 1739–1747, 2020. 5

[9] Weixin Chen, Baoyuan Wu, and Haoqian Wang. Effective backdoor defense by exploiting sensitivity of poisoned samples. In *Advances in Neural Information Processing Systems*, 2022. 3

[10] Xinyun Chen, Chang Liu, Bo Li, Kimberly Lu, and Dawn Song. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv e-prints*, pages arXiv–1712, 2017. 2, 5, 7, 19, 20, 22, 23

[11] Zhenzhu Chen, Shang Wang, Anmin Fu, Yansong Gao, Shui Yu, and Robert H Deng. Linkbreaker: Breaking the backdoor-trigger link in dnns via neurons consistency check. *IEEE Transactions on Information Forensics and Security*, 17:2000–2014, 2022. 3

[12] Edward Chou, Florian Tramer, and Giancarlo Pellegrino. Sentinet: Detecting localized universal attacks against deep learning systems. In *2020 IEEE Security and Privacy Workshops (SPW)*, pages 48–54. IEEE, 2020. 9

[13] Jia Deng, Wei Dong, Richard Socher, Li-Jia Li, Kai Li, and Li Fei-Fei. Imagenet: A large-scale hierarchical image database. In *Conference on Computer Vision and Pattern Recognition*, 2009. 5, 19

[14] Yinpeng Dong, Xiao Yang, Zhijie Deng, Tianyu Pang, Zihao Xiao, Hang Su, and Jun Zhu. Black-box detection of backdoor attacks with limited information and data. In *International Conference on Computer Vision*, 2021. 1

[15] Kuofeng Gao, Jiawang Bai, Bin Chen, Dongxian Wu, and Shu-Tao Xia. Backdoor attack on hash-based image retrieval via clean-label data poisoning. In *BMVC*, 2023. 2

[16] Kuofeng Gao, Jiawang Bai, Baoyuan Wu, Mengxi Ya, and Shu-Tao Xia. Imperceptible and robust backdoor attack in 3d point cloud. *IEEE Transactions on Information Forensics and Security*, 19:1267–1282, 2023. 2

[17] Yansong Gao, Change Xu, Derui Wang, Shiping Chen, Damith C Ranasinghe, and Surya Nepal. Strip: A defence against trojan attacks on deep neural networks. In *Proceedings of the 35th annual computer security applications conference*, pages 113–125, 2019. 9

[18] Tianyu Gu, Kang Liu, Brendan Dolan-Gavitt, and Siddharth Garg. Badnets: Evaluating backdooring attacks on deep neural networks. *IEEE Access*, 7:47230–47244, 2019. 1, 2, 5, 7, 19, 20, 22, 23

[19] Junfeng Guo, Yiming Li, Xun Chen, Hanqing Guo, Lichao Sun, and Cong Liu. Scale-up: An efficient black-box input-level backdoor detection via analyzing scaled prediction consistency. In *The Eleventh International Conference on Learning Representations*. 9

[20] Kaiming He, Xiangyu Zhang, Shaoqing Ren, and Jian Sun. Identity mappings in deep residual networks. In *European Conference on Computer Vision*, 2016. 5

[21] Kunzhe Huang, Yiming Li, Baoyuan Wu, Zhan Qin, and Kui Ren. Backdoor defense via decoupling the training process. In *International Conference on Learning Representations*, 2022. 3

[22] Wenbo Jiang, Hongwei Li, Guowen Xu, and Tianwei Zhang. Color backdoor: A robust poisoning attack in color space. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 8133–8142, 2023. 2

[23] Paramjit Kaur, Kewal Krishan, Suresh K Sharma, and Tanuj Kanchan. Facial-recognition algorithms: A literature review. *Medicine, Science and the Law*, 60(2):131–139, 2020. 1

[24] Alaa Khaddaj, Guillaume Leclerc, Aleksandar Makelov, Kristian Georgiev, Hadi Salman, Andrew Ilyas, and Aleksander Madry. Rethinking backdoor attacks. In *International Conference on Machine Learning*, pages 16216–16236. PMLR, 2023. 3

[25] Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR, 2019. 4, 18, 19

[26] Alex Krizhevsky et al. Learning multiple layers of features from tiny images. 2009. 5, 19

[27] Ram Shankar Siva Kumar, Magnus Nyström, John Lambert, Andrew Marshall, Mario Goertzel, Andi Comissoneru, Matt Swann, and Sharon Xia. Adversarial machine learning-industry perspectives. In *2020 IEEE security and privacy workshops (SPW)*, pages 69–75. IEEE, 2020. 1

[28] Ya Le and Xuan Yang. Tiny imagenet visual recognition challenge. *CS 231N*, 2015. 5, 19

[29] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Anti-backdoor learning: Training clean models on poisoned data. In *Conference on Neural Information Processing Systems*, 2021. 3

[30] Yige Li, Xixiang Lyu, Nodens Koren, Lingjuan Lyu, Bo Li, and Xingjun Ma. Neural attention distillation: Erasing backdoor triggers from deep neural networks. In *International Conference on Learning Representations*, 2021. 1, 2, 6, 7, 21, 22, 23

[31] Yuezun Li, Yiming Li, Baoyuan Wu, Longkang Li, Ran He, and Siwei Lyu. Invisible backdoor attack with sample-specific triggers. In *International Conference on Computer Vision*, 2021. 1, 2, 5, 7, 20, 22, 23

[32] Jiawei Liang, Siyuan Liang, Man Luo, Aishan Liu, Dongchen Han, Ee-Chien Chang, and Xiaochun Cao. Vl-trojan: Multimodal instruction backdoor attacks against autoregressive visual language models. *arXiv preprint arXiv:2402.13851*, 2024. 3

[33] Siyuan Liang, Jiawei Liang, Tianyu Pang, Chao Du, Aishan Liu, Ee-Chien Chang, and Xiaochun Cao. Revisiting backdoor attacks against large vision-language models. *arXiv preprint arXiv:2406.18844*, 2024. 3

[34] Siyuan Liang, Kuanrong Liu, Jiajun Gong, Jiawei Liang, Yuan Xun, Ee-Chien Chang, and Xiaochun Cao. Unlearning backdoor threats: Enhancing backdoor defense in multimodal contrastive learning via local token unlearning. *arXiv preprint arXiv:2403.16257*, 2024. 3

[35] Siyuan Liang, Mingli Zhu, Aishan Liu, Baoyuan Wu, Xiaochun Cao, and Ee-Chien Chang. Badclip: Dual-embedding guided backdoor attack on multimodal contrastive learning. *arXiv preprint arXiv:2311.12075*, 2023. 3

[36] Liangkai Liu, Sidi Lu, Ren Zhong, Baofu Wu, Yongtao Yao, Qingyang Zhang, and Weisong Shi. Computing systems for autonomous driving: State of the art and challenges. *IEEE Internet of Things Journal*, 8(8):6469–6486, 2020. 1

[37] Yingqi Liu, Shiqing Ma, Yousra Aafer, Wen-Chuan Lee, Juan Zhai, Weihang Wang, and Xiangyu Zhang. Trojaning attack on neural networks. In *Network and Distributed System Security Symposium*, 2018. 2, 6, 7, 20, 22, 23

[38] Aleksander Madry, Aleksandar Makelov, Ludwig Schmidt, Dimitris Tsipras, and Adrian Vladu. Towards deep learning models resistant to adversarial attacks. In *International Conference on Learning Representations*, 2018. 5, 17

[39] Rui Min, Zeyu Qin, Li Shen, and Minhao Cheng. Towards stable backdoor purification through feature shift tuning. *Advances in Neural Information Processing Systems*, 36, 2024. 3, 6, 7, 9, 21, 22, 23

[40] Tuan Anh Nguyen and Anh Tran. Input-aware dynamic backdoor attack. In *Conference on Neural Information Processing Systems*, 2020. 2, 6, 7, 20, 22, 23

[41] Tuan Anh Nguyen and Anh Tuan Tran. Wanet - imperceptible warping-based backdoor attack. In *International Conference on Learning Representations*, 2021. 2, 6, 7, 20, 22, 23

[42] Xiangyu Qi, Tinghao Xie, Yiming Li, Saeed Mahloujifar, and Prateek Mittal. Revisiting the assumption of latent separability for backdoor defenses. In *The eleventh international conference on learning representations*, 2022. 3

[43] Xiangyu Qi, Tinghao Xie, Jiachen T Wang, Tong Wu, Saeed Mahloujifar, and Prateek Mittal. Towards a proactive {ML} approach for detecting backdoor poison samples. In *32nd USENIX Security Symposium (USENIX Security 23)*, pages 1685–1702, 2023. 26

[44] Alec Radford, Jong Wook Kim, Chris Hallacy, Aditya Ramesh, Gabriel Goh, Sandhini Agarwal, Girish Sastry, Amanda Askell, Pamela Mishkin, Jack Clark, et al. Learning transferable visual models from natural language supervision. In *International conference on machine learning*, pages 8748–8763. PMLR, 2021. 5

[45] Piyush Sharma, Nan Ding, Sebastian Goodman, and Radu Soricut. 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)*, pages 2556–2565, 2018. 5, 19

[46] Yucheng Shi, Mengnan Du, Xuansheng Wu, Zihan Guan, Jin Sun, and Ninghao Liu. Black-box backdoor defense via zero-shot image purification. *Advances in Neural Information Processing Systems*, 36:57336–57366, 2023. 9

[47] Karen Simonyan and Andrew Zisserman. Very deep convolutional networks for large-scale image recognition. In *International Conference on Learning Representations*, 2015. 5

[48] Zhengyao Song, Yongqiang Li, Danni Yuan, Li Liu, Shaokui Wei, and Baoyuan Wu. Wpda: Frequency-based backdoor attack with wavelet packet decomposition. *arXiv preprint arXiv:2401.13578*, 2024. 2

[49] Johannes Stallkamp, Marc Schlipsing, Jan Salmen, and Christian Igel. The german traffic sign recognition benchmark: a multi-class classification competition. In *international joint conference on neural networks*, 2011. 5, 19

[50] Brandon Tran, Jerry Li, and Aleksander Madry. Spectral signatures in backdoor attacks. *Advances in neural information processing systems*, 31, 2018. 1

[51] Laurens Van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9(11), 2008. 25

[52] Matthew Walmer, Karan Sikka, Indranil Sur, Abhinav Shrivastava, and Susmit Jha. Dual-key multimodal backdoors for visual question answering. In *Proceedings of the IEEE/CVF Conference on computer vision and pattern recognition*, pages 15375–15385, 2022. 3, 6, 7, 21

[53] 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 *Symposium on Security and Privacy*, 2019. 1, 3, 6, 7, 21, 22, 23

[54] Hang Wang, Zhen Xiang, David J Miller, and George Kesidis. Mm-bd: Post-training detection of backdoor attacks with arbitrary backdoor pattern types using a maximum margin statistic. In *2024 IEEE Symposium on Security and Privacy (SP)*, pages 15–15. IEEE Computer Society, 2023. 3

[55] Ruotong Wang, Hongrui Chen, Zihao Zhu, Li Liu, and Baoyuan Wu. Versatile backdoor attack with visible, semantic, sample-specific, and compatible triggers. *arXiv preprint arXiv:2306.00816*, 2023. 2

[56] Zhenting Wang, Kai Mei, Juan Zhai, and Shiqing Ma. Unicorn: A unified backdoor trigger inversion framework. In *International Conference on Learning Representations*, 2023. 1

[57] Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Mitigating backdoor attack by injecting proactive defensive backdoor. *arXiv preprint arXiv:2405.16112*, 2024. 3

[58] Shaokui Wei, Mingda Zhang, Hongyuan Zha, and Baoyuan Wu. Shared adversarial unlearning: Backdoor mitigation by unlearning shared adversarial examples. *Advances in Neural Information Processing Systems*, 36, 2024. 2, 6, 7, 9, 21, 22, 23

[59] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, and Chao Shen. Backdoorbench: A comprehensive benchmark of backdoor learning. In *Conference on Neural Information Processing Systems Datasets and Benchmarks Track*, 2022. 1, 2, 5, 6, 19, 20, 21

[60] Baoyuan Wu, Hongrui Chen, Mingda Zhang, Zihao Zhu, Shaokui Wei, Danni Yuan, Mingli Zhu, Ruotong Wang, Li Liu, and Chao Shen. Backdoorbench: A comprehensive benchmark and analysis of backdoor learning. *arXiv preprint arXiv:2401.15002*, 2024. 2

[61] Baoyuan Wu, Shaokui Wei, Mingli Zhu, Meixi Zheng, Zihao Zhu, Mingda Zhang, Hongrui Chen, Danni Yuan, Li Liu, and Qingshan Liu. Defenses in adversarial machine learning: A survey. *arXiv preprint arXiv:2312.08890*, 2023. 3

[62] Baoyuan Wu, Zihao Zhu, Li Liu, Qingshan Liu, Zhaofeng He, and Siwei Lyu. Attacks in adversarial machine learning: A systematic survey from the life-cycle perspective. *arXiv preprint arXiv:2302.09457*, 2023. 1, 2, 3

[63] Chuhan Wu, Fangzhao Wu, and Yongfeng Huang. Rethinking infonce: How many negative samples do you need? *arXiv preprint arXiv:2105.13003*, 2021. 16

[64] Dongxian Wu and Yisen Wang. Adversarial neuron pruning purifies backdoored deep models. In *Conference on Neural Information Processing Systems*, 2021. 1, 3

[65] Yuan Xun, Siyuan Liang, Xiaojun Jia, Xinwei Liu, and Xiaochun Cao. Ta-cleaner: A fine-grained text alignment backdoor defense strategy for multimodal contrastive learning. *arXiv preprint arXiv:2409.17601*, 2024. 3

[66] Wenhan Yang, Jingdong Gao, and Baharan Mirzasoleiman. Robust contrastive language-image pretraining against data poisoning and backdoor attacks. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[67] Yi Zeng, Si Chen, Won Park, Zhuoqing Mao, Ming Jin, and Ruoxi Jia. Adversarial unlearning of backdoors via implicit hypergradient. In *International Conference on Learning Representations*, 2022. 1, 2, 6, 7, 9, 21, 22, 23

[68] Yi Zeng, Won Park, Z. Morley Mao, and Ruoxi Jia. Rethinking the backdoor attacks' triggers: A frequency perspective. In *International Conference on Computer Vision*, 2021. 2, 5, 7, 20, 22, 23

[69] Xiaoyu Zhang, Yulin Jin, Tao Wang, Jian Lou, and Xiaofeng Chen. Purifier: Plug-and-play backdoor mitigation for pre-trained models via anomaly activation suppression. In *Proceedings of the 30th ACM International Conference on Multimedia*, pages 4291–4299, 2022. 3

[70] Zhendong Zhao, Xiaojun Chen, Yuexin Xuan, Ye Dong, Dakui Wang, and Kaitai Liang. Defeat: Deep hidden feature backdoor attacks by imperceptible perturbation and latent representation constraints. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 15213–15222, 2022. 2

[71] Runkai Zheng, Rongjun Tang, Jianze Li, and Li Liu. Data-free backdoor removal based on channel lipschitzness. In *European Conference on Computer Vision*, 2022. 1, 4, 18

[72] Mingli Zhu, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Enhancing fine-tuning based backdoor defense with sharpness-aware minimization. In *International Conference on Computer Vision*, 2023. 3, 6, 7, 9, 21, 22, 23

[73] Mingli Zhu, Shaokui Wei, Hongyuan Zha, and Baoyuan Wu. Neural polarizer: A lightweight and effective backdoor defense via purifying poisoned features. *Advances in Neural Information Processing Systems*, 36, 2024. 3

[74] Rui Zhu, Di Tang, Siyuan Tang, XiaoFeng Wang, and Haixu Tang. Selective amnesia: On efficient, high-fidelity and blind suppression of backdoor effects in trojaned machine learning models. In *2023 IEEE Symposium on Security and Privacy (SP)*, pages 1–19. IEEE, 2023. 26

[75] Zihao Zhu, Mingda Zhang, Shaokui Wei, Li Shen, Yanbo Fan, and Baoyuan Wu. Boosting backdoor attack with a learnable poisoning sample selection strategy. *arXiv preprint arXiv:2307.07328*, 2023. 2

[76] Zihao Zhu, Mingda Zhang, Shaokui Wei, Bingzhe Wu, and Baoyuan Wu. Vdc: Versatile data cleanser for detecting dirty samples via visual-linguistic inconsistency. In *The Twelfth International Conference on Learning Representations*, 2023. 3

Appendix

Structure of Appendix. We provide more analysis and experimental results in Appendix, which includes: (1) Formulation of multi-modal contrastive learning, backdoor attacks and our re-activation attack for MMCL in **Appendix** A. (2) More description and algorithms details in **Appendix** B. (3) Introduction of TAC and CKA in **Appendix** C. (4) Experimental implementation details in **Appendix** D. (5) More experimental results in **Appendix** E. (6) Running time analysis in **Appendix** F. (7) Visualization in **Appendix** G. (8) Additional experimental results in **Appendix** H.

A Backdoor multi-modal contrastive learning

A.1 Formulation of multi-modal contrastive learning task.

For multi-modal contrastive learning task, the training dataset is image-text pairs $\mathcal{D} = \{(\boldsymbol{v}^{(i)}, \boldsymbol{t}^{(i)})\}_{i=1}^{n} \subseteq \mathcal{V} \times \mathcal{T}$, where $\mathcal{V} \subset \mathbb{R}^{d_v}$ and $\mathcal{T} \subset \mathbb{R}^{d_t}$ are image space and text space, respectively. For the network, we choose CLIP as our primary MMCL model for the attack. CLIP is composed of a visual encoder $f_{\boldsymbol{\theta_v}} : \mathcal{V} \to \mathbb{R}^d$ and a textual encoder $f_{\boldsymbol{\theta_t}} : \mathcal{T} \to \mathbb{R}^d$, each with parameters $\boldsymbol{\theta_v}$ and $\boldsymbol{\theta_t}$ representing their respective encoders. Denote the image embedding and text embedding as $\boldsymbol{v}_e^{(i)} = f_{\boldsymbol{\theta_v}}(\boldsymbol{v}^{(i)}), \boldsymbol{t}_e^{(i)} = f_{\boldsymbol{\theta_t}}(\boldsymbol{t}^{(i)})$, respectively, for convenience. Given a batch of training pairs $\{(\boldsymbol{v}^{(i)}, \boldsymbol{t}^{(i)})\}_{i=1}^{n_1}$, CLIP is optimized using the InfoNCE loss [63] as follows:

$$\min_{\{\boldsymbol{\theta}_v, \boldsymbol{\theta}_t\}} - \sum_{i=1}^{n_1} \log \frac{\exp\left(\boldsymbol{v}_e^{(i)} \cdot \boldsymbol{t}_e^{(i)} / \tau\right)}{\sum_{j=1}^{n_1} \exp\left(\boldsymbol{v}_e^{(i)} \cdot \boldsymbol{t}_e^{(j)} / \tau\right)}, \tag{6}$$

where τ is the temperature parameter. Given an input image $\boldsymbol{v}^{(i)}$, denote the output text of the model be $h_\Theta(\boldsymbol{v}^{(i)})$ for convenience, where $\Theta = \{\boldsymbol{\theta}_v, \boldsymbol{\theta}_t\}$.

A.2 Backdoor attacks for multi-modal contrastive learning.

For MMCL task, backdoor attacks could also be divided into data poisoning attack and training controllable attack. For data poisoning attack, an adversary creates poisoning pairs $(\boldsymbol{v}^{(i)} + \boldsymbol{\xi}, T)$ by patching a backdoor trigger $\boldsymbol{\xi}$ on the image $\boldsymbol{v}^{(i)}$ and revising the corresponding label into the target label T (for example, "a photo of `banana`" in [3] and in our work). In training controllable backdoor attack, the adversary can control the training process to inject backdoors into the model. The goal of the adversary is to train a poisoned model such that $h_{\Theta_\text{A}}(\boldsymbol{v}^{(i)}) = \boldsymbol{t}^{(i)}$ and $h_{\Theta_\text{A}}(\boldsymbol{v}^{(i)} + \boldsymbol{\xi}) = T$. And the goal of the defender is to purify the poisoned model such that the new model performs normally as: $h_{\Theta_\text{D}}(\boldsymbol{v}^{(i)}) = \boldsymbol{t}^{(i)}$ and $h_{\Theta_\text{D}}(\boldsymbol{v}^{(i)} + \boldsymbol{\xi}) \neq T$ in inference time. Denote the encoder of poisoned image and the target label as $(\boldsymbol{v}_{\boldsymbol{\xi}}^{(i)})_e$ and T_e, respectively for convenience. Our goal is to search for a perturbation $\Delta_{\boldsymbol{\xi}}^*$ onto the original trigger $\boldsymbol{\xi}$ such that $h_{\Theta_\text{D}}(\boldsymbol{v}^{(i)} + \boldsymbol{\xi} + \Delta_{\boldsymbol{\xi}}^*) = T$

Existing backdoor attacks primarily focus on achieving high attack success rates (ASR) in backdoor injection stages (I and II), with little consideration for the defensive impact in stage III. *Given the failures of $\boldsymbol{v}^{(i)} + \boldsymbol{\xi}$ in attacking h_{Θ_D}, our work focuses on the re-activation attack in stage IV* (please refer to Sec. 3.2 for formal definition of different stages of backdoors).

A.3 Re-activation attacks for multi-modal contrastive learning.

In this section we introduce our optimization formulation to learn the new trigger $\xi' = \xi + \Delta_{\boldsymbol{\xi}}$. Since CLIP uses multi-modal contrastive learning instead of supervised learning to train the model, we also optimize the perturbation $\Delta_{\boldsymbol{\xi}}$ by optimizing it with multi-modal contrastive learning loss. Given a number of n_p pairs $(\boldsymbol{v}_{\boldsymbol{\xi}}^{(i)}, T)$ from the poisoned dataset $\mathcal{D}_p$ and n_c pairs $(\boldsymbol{v}^{(j)}, \boldsymbol{t}^{(j)}) \in \mathcal{D}_c$ from the clean dataset $\mathcal{D}_c$, the optimization problem is formulated as follows:

$$\Delta_{\boldsymbol{\xi}}^* = \underset{\|\Delta_{\boldsymbol{\xi}}\|_p \leq \rho}{\arg\min} - \sum_{(\boldsymbol{v}_{\boldsymbol{\xi}}^{(i)}, T) \in \mathcal{D}_p} \log \frac{\exp\left((\boldsymbol{v}_{\boldsymbol{\xi}}^{(i)})_e \cdot T_e / \tau\right)}{\exp\left((\boldsymbol{v}_{\boldsymbol{\xi}}^{(i)})_e \cdot T_e / \tau\right) + \sum_{(\boldsymbol{v}^{(j)}, \boldsymbol{t}) \in \mathcal{D}_c} \exp\left((\boldsymbol{v}_{\boldsymbol{\xi}}^{(i)})_e \cdot \boldsymbol{t}_e^{(j)} / \tau\right)}, \tag{7}$$

Algorithm 1 Black-box Backdoor Re-Activation Attack via Universal Square Attack (BBA) [1]

Input: Defense model f, training dataset $\mathcal{D}_p$, image shape c, h, w, norm p, perturbation bound ρ, target label $t \in 1, \dots, K$, number of iterations N, termination condition ϵ.
Output: Perturbation $\Delta^*_{\boldsymbol{\xi}}$ as in Eq. 4.
$\hat{\boldsymbol{x}} \leftarrow \boldsymbol{x} + \text{init}(\Delta_{\boldsymbol{\xi}})$ for $\boldsymbol{x} \in \mathcal{D}_p$, $\quad l^* \leftarrow \mathcal{L}_{tot}(\mathcal{D}_p, \Delta_{\boldsymbol{\xi}})$.
for $i = 0, ..., N-1$ **do**
  if ASR $> 1 - \epsilon$ **then return** $\Delta_{\boldsymbol{\xi}}$.
  else
    $h^{(i)} \leftarrow$ side length of the square to modify (according to some schedule [1]);
    $\Delta^{\text{new}}_{\boldsymbol{\xi}} \sim P\left(\rho, h^{(i)}, w, c, \Delta_{\boldsymbol{\xi}}, \hat{\boldsymbol{x}}, \boldsymbol{x}\right)$ for $\boldsymbol{x} \in \mathcal{D}_p$ (see **Appendix** B for details);
    $\hat{\boldsymbol{x}}_{\text{new}} \leftarrow$ Project $\hat{\boldsymbol{x}} + \Delta^{\text{new}}_{\boldsymbol{\xi}}$ onto $\left\{z \in \mathbb{R}^d : \|z - x\|_p \leq \rho\right\} \cap [0, 1]^d$ for $\boldsymbol{x} \in \mathcal{D}_p$;
    $l_{\text{new}} \leftarrow \mathcal{L}_{tot}(\hat{\boldsymbol{x}}_{\text{new}}, t)$ for $\boldsymbol{x} \in \mathcal{D}_p$;
    if $l_{\text{new}} < l^*$ then $\Delta_{\boldsymbol{\xi}} \leftarrow \Delta^{\text{new}}_{\boldsymbol{\xi}}, l^* \leftarrow l_{\text{new}}$, compute ASR;
    $i \leftarrow i + 1$;
  end if
end for
return $\Delta^*_{\boldsymbol{\xi}}$.

where $\|\cdot\|_p$ means ℓ_p norm, and ρ is the perturbation bound. This problem can be solved using project gradient descent (PGD) [38] algorithm. Then the optimized $\Delta^*_{\boldsymbol{\xi}}$ is attached to poisoned samples and the ASR is the probability of successful attacks out of the total number of new poisoned samples.

B Algorithms details

In this work, we provide some description and details of our attack algorithms.

White-box attack setting. As shown in Eq. 4, this is a constrained optimization problem, which can be solved by using the classical project gradient descent (PGD) [38] algorithm to solve it. The main idea of PGD involves updating the perturbation using stochastic gradient descent in the initial step. Subsequently, in the following stage, the perturbation is constrained within the ρ-ball employing ℓ_p norm projection. We provide the algorithm description in Alg. 2. For additional insights, please refer to [38] for more details.

Black-box attack setting. In this work, to solve our optimization problem in black-box setting, we utilize a randomized search strategy as emphasized in Square Attack [1]. Square Attack utilizes a randomized search scheme where it selects localized square-shaped updates at random positions. This approach ensures that in each iteration, the perturbation is positioned near the boundary of the feasible set. A significant difference between our attack and Square Attack lies in their objective: Square Attack searches for a perturbation for each image, terminating the query upon successful attack, while our objective is to discover a highly generalizable universal perturbation to restore the effectiveness of the backdoor utility. Therefore, we extend it to a universal Square Attack approach:

1. Firstly, initialize a universal perturbation.
2. In each iteration, we randomly update our perturbation following the strategy in Square Attack. Apply the perturbation onto the image and then query the model with the new images.
3. Compare the loss: if the current loss is lower than the best loss, update the perturbation; otherwise, do not update and restart the search.

The above three steps represent the main concept of our algorithm. Details on the specific square update technique can be found in work [1].

Transfer attack setting. The main idea of the transfer attack is to compute the averaged loss across models for each mini-batch. The detailed algorithm is shown in Alg. 3.

Algorithm 2 White-box Backdoor Re-Activation Attack (WBA)

```
1: Input: Defense model f, training dataset D_p, norm p, perturbation bound ρ, target label
   t ∈ 1, ..., K, number of iterations N.
2: Output: Perturbation Δ*_ξ as in Eq. 4.
3: initialize(Δ_ξ).
4: for i = 0, ..., N − 1 do
5:     for mini-batch B = {(x^i_ξ, t)}^b_{i=1} ⊂ D_p do
6:         Given f and input {(x^i_ξ + Δ_ξ, t)}^b_{i=1}, compute the loss l of Eq. 4;
7:         Update Δ_ξ by minimizing l via PGD algorithm;
8:     end for
9: end for
10: return Δ*_ξ.
```

Algorithm 3 Re-Activation Attack via Transfer Attack (TA)

```
1: Input: Surrogate models f_m, m = 1, ··· , M, training dataset D_p, norm p, perturbation bound
   ρ, target label t ∈ 1, ..., K, number of iterations N.
2: Output: Perturbation Δ*_ξ as in Eq. 5.
3: initialize(Δ_ξ).
4: for i = 0, ..., N − 1 do
5:     for mini-batch B = {(x^i_ξ, t)}^b_{i=1} ⊂ D_p do
6:         l = 0;
7:         for m = 0, ..., M − 1 do
8:             Given f_m and input {(x^i_ξ + Δ_ξ, t)}^b_{i=1}, compute the total loss l_m of Eq. 5;
9:             l ← l + l_m;
10:        end for
11:        Update Δ_ξ by minimizing l via PGD algorithm;
12:    end for
13: end for
14: return Δ*_ξ.
```

C Introduction of TAC and CKA

We provide the detailed introduction of Trigger-activated Change (TAC) [71] and Centered Kernel Alignment (CKA) [25] in this section.

Trigger-activated Change. To measure the correlation of neurons with backdoors, Zheng *et al.* [71] proposed the TAC metric to quantify the correlation between the impact of backdoors and neurons. Given the poisoned dataset $\mathcal{D}_p = \{(\boldsymbol{x}_{\boldsymbol{\xi}}^{(i)}, y^{(i)})\}$, let the original clean dataset of $\mathcal{D}_p$ to be $\mathcal{D}_c$, *i.e.*, $\mathcal{D}_c = \{(\boldsymbol{x}^{(i)}, y^{(i)})|\boldsymbol{x}_{\boldsymbol{\xi}}^{(i)} \in \mathcal{D}_p\}$. Then the TAC can be computed as follows:

$$TAC_k^{(l)}(\mathcal{D}_p, \mathcal{D}_c) = \frac{1}{|\mathcal{D}_p|} \sum_{(\boldsymbol{x}_{\boldsymbol{\xi}}, \boldsymbol{x}) \in (\mathcal{D}_p, \mathcal{D}_c)} \left\| f_k^{(l)}(\boldsymbol{x}) - f_k^{(l)}(\boldsymbol{x}_{\boldsymbol{\xi}}) \right\|_2, \tag{8}$$

where k is the index of channel of the l^{th} layer. A higher TAC value assigned to a neuron indicates a stronger association with backdoors. In this work, with this metric, we first assign each neuron with a TAC value. In order to select neurons relevant to backdoors and considering their sparsity nature, the top 10% of neurons based on their descending TAC values are chosen as the backdoor related neurons. Then the Backdoor Existence Coefficient can be computed accordingly.

Centered Kernel Alignment. The Centered Kernel Alignment (CKA) [25] measures the similarity between representations, which utilizes HSIC to measure the independence between two distributions. It quantifies how well neural networks preserve similarity relations in the data across different layers. It is a valuable tool in feature analysis and understanding DNNs especially for high-dimensional features. In this work we employ CKA to quantify the similarity between features in different networks. As the work [25] shows, the Centered Kernel Alignment (CKA) is defined as follows: Let $X \in \mathbb{R}^{n\times d}$ and $Y \in \mathbb{R}^{n\times d}$ be two representations from neural networks, where n represent number of samples and d is the feature dimension. The empirical estimator of Hilbert-Schmidt Independence Criterion (HSIC) is defined as:

$$\mathrm{HSIC}(K, L) = \frac{1}{(n-1)^2}\mathrm{tr}(KHLH), \tag{9}$$

where H is the centering matrix $H_n = I_n - \frac{1}{n}\mathbf{1}\mathbf{1}^{\mathrm{T}}$. The K and H are linear kernels: $K_{ij} = k(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^{\mathrm{T}}\mathbf{y}_i, L_{ij} = l(\mathbf{x}_i, \mathbf{y}_j) = \mathbf{x}_i^{\mathrm{T}}\mathbf{y}_i$ as defined in [25]. Then the Centered Kernel Alignment is defined as:

$$\mathrm{CKA}(K, L) = \frac{\mathrm{HSIC}(K, L)}{\sqrt{\mathrm{HSIC}(K, K)\,\mathrm{HSIC}(L, L)}}. \tag{10}$$

More details could be found in [25].

D Experimental implementation details

In this section, we delve into the implementation details, covering the evaluation datasets, specifics of the attacks and defenses compared, and implementation of our proposed methods. All experiments are executed five times with varying random seeds and the averaged results are displayed in this work.

D.1 Datasets

For image classification task, we use three benchmark datasets: CIFAR-10 [26], Tiny ImageNet [28], and GTSRB [49] to assess the performance of our approach, following the benchmarks outlined in [59]. For MMCL task, a subset of CC3M dataset [45] is selected for backdoor injection and the poisoned models are tested through zero-shot evaluation on ImageNet-1K validation set [13]. All dataset splits are aligned in our experiments.

- CIFAR-10: Total number of 60,000 images distributed among ten classes, with 5,000 images per class in the training set and 1,000 images per class in the testing set. Each image in CIFAR-10 is sized 32×32 pixels.
- Tiny ImageNet: A subset of ImageNet [13] containing 200 classes, 500 training samples and 50 testing samples per class. Each image in Tiny ImageNet is sized 64×64 pixels.
- GTSRB: A total of 39,209 training images and 12,630 testing images among 43 classes. Each image in GTSRB is sized 32×32 pixels.
- CC3M: The CC3M dataset has about 3300K, 15K, 12K image-text pairs for the training, validation, and testing dataset, respectively. Each image in CC3M is sized 224×224 pixels. Following [3], 500K image-text pairs from the CC3M are selected in backdoor injection phase.
- ImageNet-1K: the ImageNet-1K dataset is a subset of ImageNet dataset, which has a total of 1000 classes. Each image in ImageNet-1K is sized 224×224 pixels.

D.2 Backdoor attack details.

We introduce the different backdoor attack methods first, followed by the experimental settings.

Fig. 4 and 5 show the visualization of poisoned samples in comparison with clean image for different backdoor attacks on CIFAR-10 and ImageNet-1K dataset, respectively. The attack details for image classification task are as follows:

- BadNets [18]: BadNets is trigger-additive attack which inserts a patch of fixed pattern (a 3×3 white square patch in our work) to replace some pixels in the image. The patch size is 3×3 on CIFAR-10 and GTSRB, and 6×6 on Tiny ImageNet, following BackdoorBench.
- Blended backdoor attack (Blended) [10]: Blended attack blends a pre-defined image (Hello Kitty in our work) with the original image. The blend coefficient α is 0.2, following BackdoorBench.

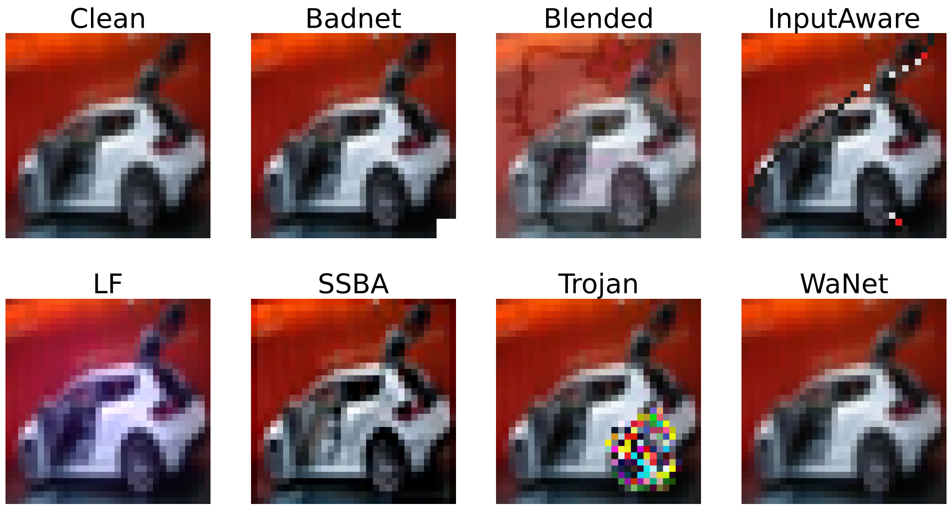

Figure 4: Visualization of poisoned samples for different backdoor attacks on CIFAR-10 dataset.

Figure 5: Visualization of poisoned samples for different backdoor attacks on ImageNet-1K dataset.

- Input-aware dynamic backdoor attack (Input-Aware) [40]: Input-Aware is a training-controllable attack that first learns a trigger generator by adversarial training. Then the generator is used to produce sample-specific triggers during model training.
- Low frequency attack (LF) [68]: LF first learns a universal adversarial perturbation (UAP) and filters the high-frequency artifacts. Then the filtered UAP is the trigger and patched onto the clean samples to generate poisoned samples.
- Sample-specific backdoor attack (SSBA) [31]: SSBA first trains an autoencoder. Then the autoencoder is used to fuse triggers with clean samples to generate poisoned samples.
- Trojan backdoor attack (Trojan) [37]: Trojan first learns a universal adversarial perturbation (UAP), and then patches it onto the clean samples to generate poisoned samples.
- Warping-based poisoned networks (WaNet) [41]: WaNet first defines a warping function to perturb the clean samples to generate poisoned samples. Then the adversary controls the training process to make sure the model learns the specific warping.

More details can be found in BackdoorBench [59].

For MMCL task, we follow CleanCLIP's settings[3]. The attack details for MMCL task are as follows:

- BadNets [18]: BadNets is trigger-additive attack which inserts a patch of fixed pattern (a 16×16 random noise patch in our work) to replace some pixels in the image.
- Blended backdoor attack (Blended) [10]: Blended attack blends a pre-defined image (a global random noise patch in our work) with the original image. The blend coefficient α is 0.2.
- SIG [4]: SIG attack designs a sine wave pattern noise as a trigger, which has a same size with the image. The blend coefficient α is 0.2.

- TrojanVQA [52]: TrojanVQA is a training-controllable attack in which the adversary utilizes both modalities to generate triggers, which has a size of 16×16.

To poison image classification task, we use a poisoned dataset with 10% poisoning ratio to train the poisoned model. To poison the MMCL model, we start with the pre-trained CLIP model which is trained on 400M image-text pairs. After that, a total of 500K image-text pairs within which 1500 samples are poisoned pairs is used for backdoor injection. The model is trained for 10 epochs with a learning rate of 1e-6, and a batch size of 128.

D.3 Backdoor defense details.

We introduce the different backdoor defense methods in this section.

- Neural cleanse (NC) [53]: NC first searches for a minimal UAP to detect backdoors. If the model is detected as a backdoor model, it purifies the model by unlearning the optimized UAP.
- Neural attention distillation (NAD) [30]: NAD use knowledge distillation strategy which distills the attention across the model to acquire a new clean model.
- Implicit backdoor Adversarial unlearning (i-BAU) [67]: I-BAU designs a implicit hyper-gradient method to solve the adversarial training optimization.
- FT-SAM [72]: It utilizes sharpness-aware minimization to fine-tune the poisoned model.
- Shared adversarial unlearning (SAU) [58]: SAU first generates shared adversarial examples and then unlearns these adversarial examples to purify the model.
- Feature shift tuning (FST) [39]: FST encourages feature shifts by re-iniltialization the linear classifier and fine-tuning the model.
- CleanCLIP [3]: CleanCLIP use both multi-modal contrastive loss and in-modal self-supervised loss to fine-tune the model.
- FT [3]: FT uses multi-modal contrastive loss to fine-tune the model.

More details about the implementation of defenses can be found in BackdoorBench [59]. For CleanCLIP, we follow the work's setting [3] that the CLIP model is trained for 10 epochs with 50 steps of warm-up using a learning rate of 4.5e-6, and a batch size of 64. A total of 10K training pairs are selected from CC3M to train.

D.4 Backdoor re-activation attack details.

During the inference phase, we implement our re-activation attack by searching for a global universal perturbation (same size as images) without altering model parameters. For image classification tasks, we employ the same optimized hyperparameters to learn the perturbation across various models and datasets. Specifically, the details are as follows:

- For white-box attack, we use the SGD optimizer with a learning rate of 0.05, update the adversarial perturbation within the inner loops for 5 steps, and train for a total of 50 epochs. The hyperparameter λ for the loss is fixed at 1. The training dataset is 1000 poisoned samples that are randomly selected from the original poisoned samples. The batch size is set to 256.
- For black-box attack, we follow the original hyperparameters as in [1]. The updated criterion is based on the decrease in loss rather than the improvement in ASR, as we have found that this approach yields better results.
- For transfer attack, we maintain same hyperparameters to those used in white-box attacks except for training epochs, which is set to 100. We also use a smaller norm bound 1. The training samples are set to 5000. For ensembling these surrogate models, we average their losses in each mini-batch.

For MMCL task, we use the SGD optimizer with a learning rate of 0.01, update the adversarial perturbation within the inner loop for one steps, and train for a total of 40 epochs. We use the ℓ_∞-norm with a 0.05 norm bound. We use 1500 poisoned image-text pairs and some clean reference data for optimization.

E Experimental results on different datasets and models

In this section, we showcase the performance of our attacks under various settings to demonstrate the superior performance of our method.

Table 12: Performance (%) of backdoor re-activation attack on both white-box (WBA) and query-based black-box (BBA) attacks with ℓ_∞-norm bound $\rho = 0.05$ against different defenses with Tiny ImageNet on PreAct-ResNet18. The best results are highlighted in **boldface**.

Attacks	No Defense	NC [53]			NAD [30]			i-BAU [67]			FT-SAM [72]			SAU [58]			FST [39]		
		Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA
BadNets [18]	99.90	99.90	99.93	**99.97**	0.27	**55.58**	26.21	3.61	**99.40**	98.44	0.21	**45.96**	23.73	0.28	**57.88**	30.87	0.02	**49.96**	17.15
Blended [10]	99.67	95.34	**99.97**	99.69	94.78	**99.30**	94.58	95.58	**99.01**	96.32	37.22	**99.07**	75.26	0.01	**76.38**	31.18	1.32	**91.75**	0.48
Input-Aware [40]	99.60	0.30	**99.90**	9.38	0.15	**36.48**	25.03	**52.13**	24.99	10.64	0.49	**37.37**	0.81	0.08	**19.04**	7.32	0.02	**6.36**	2.20
LF [68]	98.51	88.63	**98.76**	97.29	58.00	**97.90**	37.05	3.77	**97.18**	85.02	5.14	**97.30**	30.59	2.47	**78.98**	50.42	2.66	**78.37**	1.39
SSBA [31]	97.69	0.05	**98.17**	14.75	69.47	**97.05**	69.07	9.14	**96.26**	80.52	0.38	**34.04**	0.05	0.03	**4.19**	0.30	0.73	**66.59**	51.11
Trojan [37]	99.97	0.27	**99.99**	13.75	1.01	**61.31**	26.77	0.80	99.44	**99.67**	0.21	**50.08**	25.09	0.01	**18.80**	5.56	0.58	**68.92**	0.31
WaNet [41]	96.50	**96.50**	66.36	69.59	0.87	**79.89**	51.89	18.57	**63.68**	47.65	0.79	**73.61**	53.77	41.83	**80.59**	25.35	0.15	**77.89**	27.74
Avg	98.83	54.43	**94.73**	57.77	32.08	**75.36**	47.23	26.23	**82.85**	74.04	6.35	**62.49**	29.90	6.39	**47.98**	21.57	0.78	**62.84**	14.34

Attack performance on Tiny ImageNet dataset. Tab. 12 presents the performance of our re-activation attack with both WBA and BBA applied on the Tiny ImageNet dataset with PreAct-ResNet18, compared with ASRs of original attack models (**No Defense**) and defense models (**Defense**). Careful observation and analysis of this table furnishes some important insights:

1. While not achieving the same high level efficacy as in previous experiments, our attacks still show reasonable effectiveness against defense models. On average, our WBA and BBA improve ASRs by 50.00% and 19.77% respectively when compared against defense mechanisms, which is actually high than that on CIFAR-10 dataset. This highlights some potential security vulnerabilities in these defense models, although the final ASRs is less severe than those exposed in the previous dataset.
2. The performance of our WBA establishes the viability of our backdoor recovery mechanism in a more challenging setting. It further verifies the latent recoverability of backdoors in defense models. Despite the existing gap between our WBA and the more realistic BBA, we suggest that this gap can be reduced with further optimization of our black-box attack strategy.
3. When it comes to defense mechanisms, similar to previous observations, attacks on three defenses FT-SAM, SAU and FST show less impressive ASRs. This can be seen as an indication of the significant efficiency of their backdoor removal mechanism. While these mechanisms are more effective in this dataset, these insights are still crucial for developing future defense strategy development.
4. It's worth mentioning some failed cases in our experiment on the Tiny ImageNet dataset. Although our attack methods generally show promising results, the performance in some particular instances falls short of expectations. Future work can gain valuable insights from scrutinizing these instances more closely. Such failures in specific settings serve as a stepping stone toward the development of more effective and robust attack strategies, such as increasing the diversity of random searches.

Attack performance on GTSRB dataset. Tab. 13 presents the performance of our re-activation attack with both WBA and BBA applied on the GTSRB dataset with PreAct-ResNet18, compared with ASRs of original attack models (**No Defense**) and defense models (**Defense**). A meticulous examination of the results in Tab. 13 provides pivotal insights:

1. Consistent with earlier experiments, our attacks display impressive effectiveness against these defense models. In these tests, our WBA and BBA average ASRs show an improvement of 50.58% and 45.25% respectively compared to the defense mechanisms. This achievement exposes vulnerabilities in the current defense models that were previously unnoticed and highlights the robustness of our attacks.
2. The effective performance of our WBA affirms the potency of our backdoor recovery mechanism. The backdoors' resilience and latent recoverability in defense models are

Table 13: Performance (%) of backdoor re-activation attack on both white-box (WBA) and query-based black-box (BBA) attacks with ℓ_∞-norm bound $\rho = 0.05$ against different defenses with GTSRB on PreAct-ResNet18. The best results are highlighted in **boldface**.

Attacks	No Defense	NC [53]			NAD [30]			i-BAU [67]			FT-SAM [72]			SAU [58]			FST [39]		
		Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA
BadNets [18]	95.02	0.02	**62.43**	58.59	79.94	**94.66**	92.57	0.00	**48.04**	40.93	0.17	**59.43**	55.15	0.01	**36.11**	30.01	0.02	**57.49**	48.30
Blended [10]	100.00	8.76	**80.18**	76.50	99.30	**100.00**	98.99	96.39	**100.00**	99.10	9.50	**89.56**	88.48	29.99	**87.75**	84.56	77.26	**99.94**	97.89
Input-Aware [40]	95.85	0.03	**53.76**	48.01	65.55	**96.23**	94.51	0.00	**50.29**	46.42	0.02	**64.04**	59.21	0.00	**29.74**	19.78	0.00	**37.15**	33.06
LF [68]	99.58	0.06	**70.06**	67.30	79.76	**99.28**	98.49	7.43	**77.56**	73.25	2.55	**69.34**	61.99	0.04	**44.43**	27.14	0.82	**71.95**	67.21
SSBA [31]	99.77	2.43	**66.97**	63.22	96.95	**99.69**	97.37	0.18	**36.30**	30.70	0.70	**57.14**	53.65	1.95	**37.13**	31.36	32.55	**92.99**	91.29
Trojan [37]	100.00	0.36	**64.14**	60.00	0.10	**55.46**	48.33	0.00	**33.51**	18.68	0.11	**62.82**	61.14	0.06	**55.44**	49.81	1.95	**79.16**	74.30
WaNet [41]	98.20	0.15	**89.03**	77.65	0.04	**82.73**	80.42	0.26	**56.68**	44.98	0.00	**63.94**	56.61	0.08	**41.61**	30.97	0.00	**65.74**	57.96
Avg	98.35	1.69	**69.51**	64.47	60.23	**89.72**	87.24	14.90	**57.48**	50.58	1.86	**66.61**	62.32	4.59	**47.46**	39.09	16.08	**72.06**	67.14

further substantiated. Additionally, the performance gap between our WBA and the more realistic BBA suggests room for further enhancement of our black-box attack strategy.

3. Concerning the defense mechanisms, the ASRs against i-BAU and SAU continue to be relatively less impressive. The indication of these defense mechanisms' efficiency in backdoor removal remains constant. Despite the improved effectiveness witnessed in the GTSRB dataset, these results works as reference for the design of more robust defense strategies in the future.

Table 14: Performance (%) of backdoor re-activation attack on both white-box (WBA) and query-based black-box (BBA) attacks with ℓ_∞-norm bound $\rho = 0.05$ against different defenses with CIFAR-10 on VGG19-BN. The best results are highlighted in **boldface**.

Attacks	No Defense	NC [53]			NAD [30]			i-BAU [67]			FT-SAM [72]			SAU [58]		
		Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA	Defense	WBA	BBA
BadNets [18]	94.43	5.08	**95.31**	34.57	5.77	**94.80**	64.86	3.13	**92.69**	39.68	1.29	**68.77**	24.84	4.28	**85.84**	66.59
Blended [10]	99.50	99.50	**99.72**	98.52	86.98	**99.78**	78.47	51.67	**99.94**	88.22	8.23	**98.56**	75.13	7.81	**96.81**	62.33
Input-Aware [40]	97.02	97.02	**99.34**	97.37	14.04	**83.38**	19.13	78.93	**99.38**	97.41	3.41	**79.14**	27.47	1.19	**60.07**	27.07
LF [68]	13.83	1.26	**91.66**	15.38	3.07	**88.88**	43.07	6.66	**69.79**	7.27	**2.17**	0.73	0.76	1.56	**79.07**	25.54
SSBA [31]	95.10	95.10	**98.23**	87.69	52.22	**98.79**	26.60	12.37	**95.82**	72.07	1.84	**59.81**	25.17	3.03	**67.79**	33.77
Trojan [37]	100.00	**100.00**	100.00	100.00	5.18	**95.00**	58.59	2.69	**85.93**	38.76	5.13	**78.03**	2.34	0.19	**41.67**	21.95
WaNet [41]	96.49	96.49	**99.97**	69.32	10.23	**97.27**	68.97	2.40	**92.67**	62.94	1.10	**92.21**	68.22	1.72	**93.48**	56.47
Avg	85.20	70.64	**97.75**	71.83	25.35	**93.99**	51.38	22.55	**90.89**	58.05	3.31	**68.18**	31.99	2.83	**74.96**	41.96

Attack performance on VGG19-BN network. Tab. 14 presents the performance of our re-activation attack with both WBA and BBA applied on the CIFAR-10 dataset with VGG19-BN architecture, compared with ASRs of original attack models (**No Defense**) and defense models (**Defense**). Detailed analysis of the results provides the following key takeaways:

1. Our attacks display remarkable potency against the VGG19-BN network, with both our WBA and BBA demonstrating impressive average ASRs. Specifically, our WBA achieves a significantly high ASR, further emphasizing the backdoor's recoverability, even in this more complex network architecture. As for BBA, although its ASR doesn't reach the same level as WBA, it presents a commendable rate, denoting a successful real-world adversarial scenario.

2. The superior performance of our WBA attests to the robust and tenacious nature of our backdoor recovery mechanism, showcasing our approach's adaptability and effectiveness across different network structures. The observable gap in ASR between WBA and BBA can be an impetus for refining the black-box attack strategy.

Attack performance with different norm types under different backdoor attacks. In (a) and (b) of Fig. 3 in the main script, we display the ASR results under different norm types and bounds using Blended attack model. Here, we do more experiments on different attacks and the results are shown in Fig. 6. As can be seen from the figure, our re-activation attack demonstrates a consistent trend across various attack models, showing stable high ASRs when the bound approaches 2 and 0.05 for ℓ_2-norm and ℓ_∞-norm attacks, respectively.

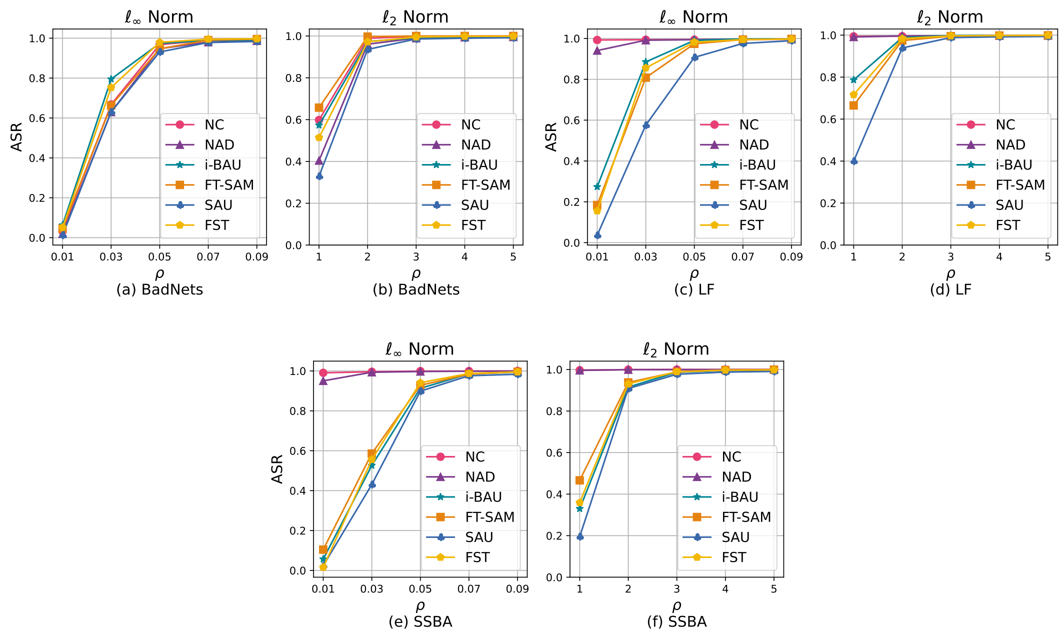

Figure 6: (a) and (b) show the ASR results under different norm type p and bound ρ for BadNets. (c) and (d) show the ASR results under different norm type p and bound ρ for LF. (e) and (f) show the ASR results under different norm type p and bound ρ for SSBA.

F Running time analysis

Table 15: Running time analysis.

Set up (min.)	Res18+CIFAR10	Res18+Tiny	Res18+GTSRB	VGG+CIFAR10	CLIP+CC3M
WBA	2.5	4.4	1.7	2.3	32.5
BBA	20.1	36.8	10.4	30.0	N/A
TA	11.3	23.5	8.4	14.8	32.5

In this section, we conduct an analysis of algorithm complexity based on running time statistics. Except for the initial training phase for attack and defense, all our re-activation attacks are trained on a single 3090Ti GPU. We provide a comparative view of the running times. Since the running time of our attack is only related to trainning dataset and network, while independent with specific backdoor attack or defense methods, we didn't specify the particular method, as the running time is consistent across methods. As displayed in Tab. 15, our attack achieves impressive speed. This can be attributed primarily to our attack requiring a smaller number of training samples, and our approach's efficiency in computing adversarial samples, needing only a few inner-loop iterations to achieve satisfactory performance. As a result, the training speed is expedited. Unlike in traditional adversarial attacks, our attack will require no further training once the optimal UAP solution is found. This condition poses a considerable threat in reality. It is noted that "N/A" appears for BBA on the CLIP models and CC3M dataset in the table, as we did not conduct black-box attacks for the CLIP models. For query-based black-box attack, the attacker cannot directly access the target model (such as weights or gradients) and CLIP models only return the final matching score or ranking results. This limits the ability of query-based black-box attacks. Moreover, there are no relevant studies for reference. Thus, we marked related result as "N/A". Additionally, we want to emphasize that BBA requires a large number of queries to the target model to achieve satisfactory attack performance, whereas TA only needs a single query to launch an attack. Clearly, TA is both more efficient and practical compared to BBA.

G Visualization

In this section, we provide two visualization techniques to showcase the existence of backdoors: feature map visualization and t-SNE visualization.

Feature maps visualization. We visualize the feature maps, *i.e.*, the features after all convolutional layers, of LF attack and different defense models. We rank these features in descending order of the TAC value of the attack model, meaning that, in each image's subplot from top to bottom, it illustrates a sort from high to low of backdoor effect. From Figures7 to 12, we can make the following observations:

- In the attack model, the highlighted part of the feature maps (the upper sections of each subplot) indicates the existence of a backdoor in the model.
- The highlighted corresponding section in the defense model suggests that the defense model is still sensitive to backdoor samples. This sensitivity manifests as an ability to primarily activate neurons related to the backdoor, even when presented with such samples.
- We have also visualized the feature maps of the clean model and have found that no such phenomenon exists in the clean model. This comparison indicates a stark difference in behavior between the defense model and the clean model.

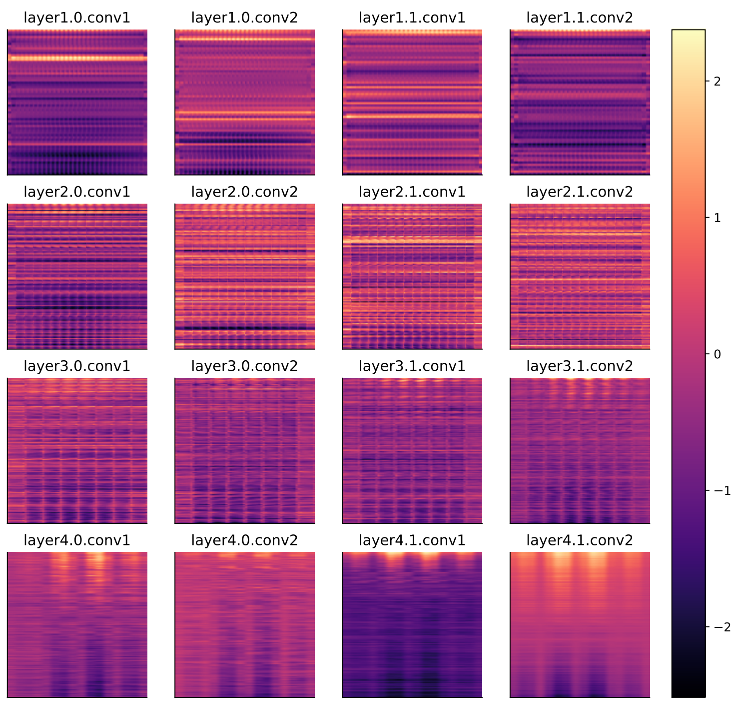

Figure 7: Sorted feature map visualization for all convolutional layers on PreAct-ResNet18 with the features in descending order of TAC values for **LF** backdoor attack.

T-SNE visualization. We attempt to observe the backdoor effect in defense models by visualizing the features of poisoned and clean samples via t-SNE visualization [51]. As illustrated in Figures 13 to 15, black dots denote poisoned samples while different colors signify various classes of clean samples. Several observations can be drawn from these figures:

- Across these attacks, the backdoor samples are clustered in the feature space.
- In the defense models, numerous backdoor samples still cluster together, indicating that the backdoor traits of these samples continue to dominate the network's recognition of backdoor images, even though these are no longer classified into the target class.

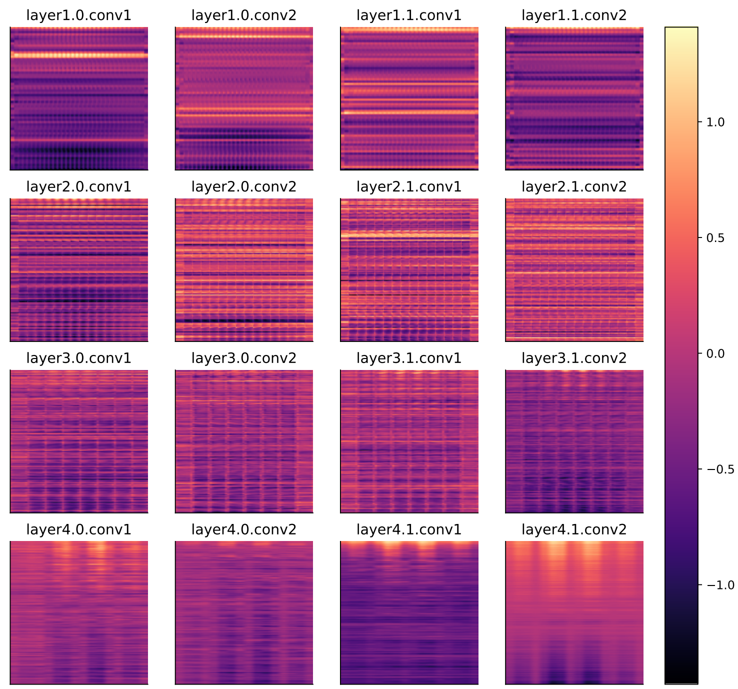

Figure 8: Sorted feature map visualization for all convolutional layers on PreAct-ResNet18 with the features in descending order of TAC values for **NAD** defense against **LF** backdoor attack.

Thus, viewing from the t-SNE visualization, we may also infer that backdoors still exist within these defense models.

H Additional experimental results

H.1 Attack performance against recent defenses

To test the attack performance against recent defenses, we evaluate the performance against two defense methods: SEAM [74] and CT [43], respectively. The evaluations are conducted on CIFAR-10 dataset with PreAct-ResNet18 network, and the results are shown in Table 16. It is found that both SEAM and CT are vulnerable to the proposed re-activation attack. We would like to emphasize that we have not claimed all post-training defenses are vulnerable to re-activation attacks. The primary objectives of our work are: (1) to reveal this new threat, which has been validated against several classic post-training defenses, and (2) to provide effective tools for evaluating the vulnerability of both existing and future post-training defenses. Therefore, future post-training defense strategies should take this threat into account and aim to mitigate the proposed re-activation attack.

Table 16: ASR (%) of our attack against SEAM and CT.

Post-training defense $\rightarrow$	**SEAM**			**CT**		
Original attack $\downarrow$**,Re-activation attack** $\rightarrow$	**No re-activation**	**WBA**	**BBA**	**No re-activation**	**WBA**	**BBA**
BadNets	5.33	97.53	33.51	0.00	99.58	92.21
Blended	6.79	98.40	69.79	1.34	100.00	99.00
Input-Aware	1.27	92.10	48.59	70.95	99.96	85.80
LF	13.22	97.61	65.63	3.28	99.63	99.23

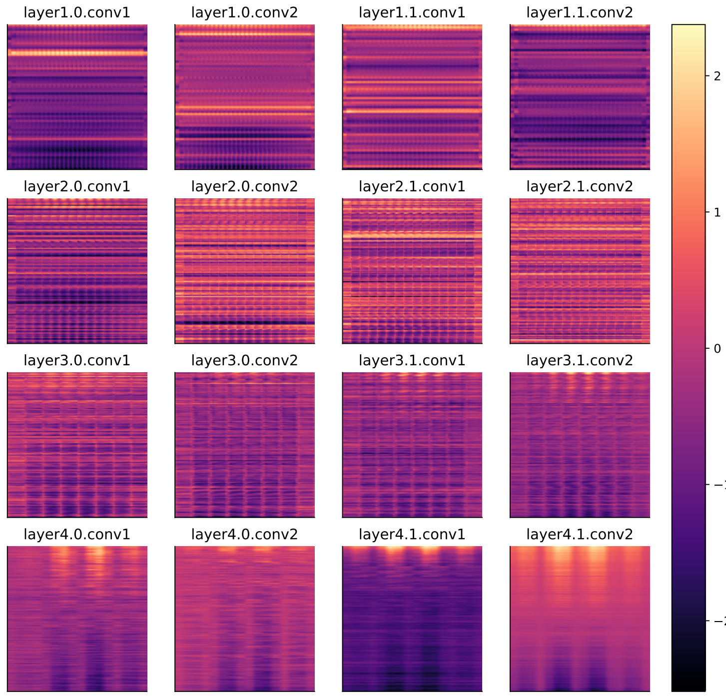

Figure 9: Sorted feature map visualization for all convolutional layers on PreAct-ResNet18 with the features in descending order of TAC values for **NC** defense against **LF** backdoor attack.

H.2 Transfer attack across model architectures

In the previously discussed transfer-based re-activation attack, we considered a scenario where the attacker is the publisher of the backdoored model. Here, we explore a more strict scenario, where the attacker is only the publisher of the poisoned dataset and has no knowledge of the defender's model architecture or training process. Therefore, when carrying out a transfer-based re-activation attack, the adversary can perform a transfer attack across different model architectures. We study this problem as follows.

- **Threat model:** Here we present a more strict setting where the adversary can only manipulate the training dataset, while having no access to the training and post-training stages. Thus, the adversary only knows the original trigger $\boldsymbol{\xi}$, but has no knowledge of $f_{\boldsymbol{\theta}_{\text{A}}}$ or $f_{\boldsymbol{\theta}_{\text{D}}}$. Compared to the previous threat model, **one major challenge is the unknown architecture of the target model $f_{\mathbf{D}}$.**
- **Main attack steps:** Compared to the steps in the previous setting, there is one additional step where the adversary must first train a backdoored model $f'_{\boldsymbol{\theta}_{\text{A}}}$ based on $\mathcal{D}_p$, which has a different architecture than $f_{\boldsymbol{\theta}_{\text{D}}}$. All remaining steps are the same as those in the previous setting.
- **Experimental results:** As shown in Table 17, although the transfer attack across model architectures does not achieve as high ASR as the transfer attack with the same architecture (*i.e.*, the results in Table 3 of the main manuscript), it still demonstrates a certain degree of backdoor transferability. This is an intriguing phenomenon worthy of further exploration.

H.3 More experimental results on test-time detection

In Table 6 of the main manuscript, we analyzed the performance of our WBA attack against test-time detection. Here, we present additional experimental results, focusing on more backdoor attack methods, and evaluating the performance of our WBA, BBA, and TA attacks. As shown in Tab. 18, our three kinds of attacks do not markedly increase the TPR compared the defense models. These findings provide insights to develop more stealthy re-activation backdoor attacks in the future.

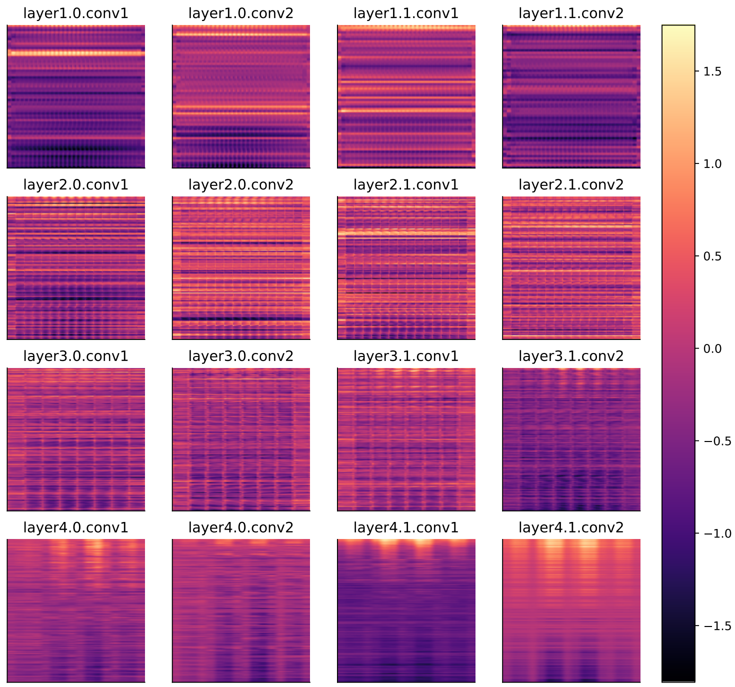

Figure 10: Sorted feature map visualization for all convolutional layers on PreAct-ResNet18 with the features in descending order of TAC values for **FST** defense against **LF** backdoor attack.

Table 17: Transfer re-activation attack preformance (ASR %) against the target model PreAct-ResNet18, using different architectures of source models.

Source Model	**WideResNet28-2**			**ResNet18**			**VGG19-BN**		
Attack ↓ Defense →	**i-BAU**	**FT-SAM**	**SAU**	**i-BAU**	**FT-SAM**	**SAU**	**i-BAU**	**FT-SAM**	**SAU**
BadNets	95.6	84.1	60.0	53.4	30.9	26.5	89.2	71.7	48.6
Blended	98.5	98.5	83.2	79.1	75.5	64.1	97.9	92.8	90.1

Table 18: ASR of our three attack methods against three test-time backdoor detection methods.

Attack↓	Detection↓	Backdoored↓	FT-SAM (Defense)	WBA (Ours)	BBA (Ours)	TA (Ours)	SAU (Defense)	WBA (Ours)	BBA (Ours)	TA (Ours)
BadNets	SCALE-UP	0.3955	0.7959	0.6858	0.6225	0.8222	0.7949	0.4954	0.4648	0.5924
	SentiNet	0.3770	0.0000	0.0000	0.0000	0.0000	0.0018	0.0000	0.0011	0.0000
	STRIP	0.8834	0.0073	0.0553	0.0027	0.2576	0.1033	0.0647	0.0144	0.0760
Blended	SCALE-UP	0.6077	0.5420	0.6577	0.5358	0.7635	0.7500	0.7138	0.6398	0.6942
	SentiNet	0.0292	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	STRIP	0.5946	0.0076	0.5007	0.0007	0.7727	0.0086	0.0742	0.0002	0.1087
Input-aware	SCALE-UP	0.6106	0.9323	0.8058	0.7891	0.8843	0.8632	0.6387	0.6216	0.5810
	SentiNet	0.4030	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	STRIP	0.0085	0.0369	0.2380	0.0199	0.6853	0.0368	0.0536	0.0108	0.0682
LF	SCALE-UP	0.8663	0.8480	0.8355	0.7652	0.8913	0.4286	0.2748	0.5714	0.3308
	SentiNet	0.0002	0.0000	0.0000	0.0000	0.0000	0.9829	0.9080	0.9322	0.9700
	STRIP	0.8438	0.0141	0.2911	0.0127	0.2872	0.2484	0.2011	0.1743	0.6934
SSBA	SCALE-UP	0.4744	0.8082	0.7217	0.6903	0.8555	0.8505	0.7188	0.7187	0.7646
	SentiNet	0.0186	0.0000	0.0000	0.0000	0.0000	0.2827	0.1029	0.2611	0.2022
	STRIP	0.7567	0.0060	0.2668	0.0612	0.6591	0.1099	0.0249	0.0810	0.2030
Trojan	SCALE-UP	0.9264	0.8491	0.7311	0.6765	0.8838	0.8160	0.5536	0.4644	0.5326
	SentiNet	0.0292	0.0110	0.0105	0.0000	0.0000	0.0211	0.0147	0.0089	0.0044
	STRIP	0.9999	0.0194	0.2984	0.0143	0.7993	0.0419	0.0116	0.0150	0.0009
Wanet	SCALE-UP	0.5012	0.9070	0.8207	0.8137	0.8888	0.7568	0.6980	0.6607	0.7655
	SentiNet	0.1757	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000	0.0000
	STRIP	0.0086	0.0404	0.2231	0.0112	0.6535	0.1651	0.1580	0.0277	0.2799

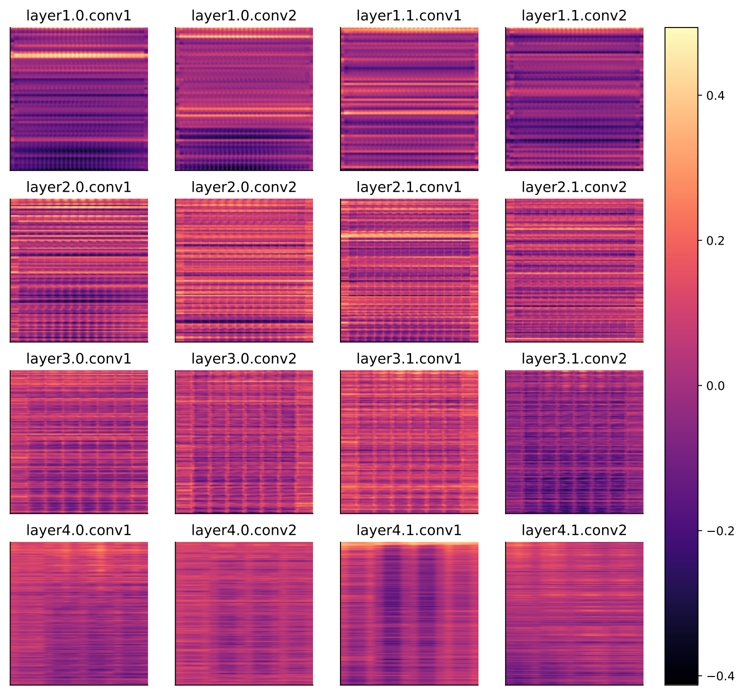

Figure 11: Sorted feature map visualization for all convolutional layers on PreAct-ResNet18 with the features in descending order of TAC values for **FT-SAM** defense against **LF** backdoor attack.

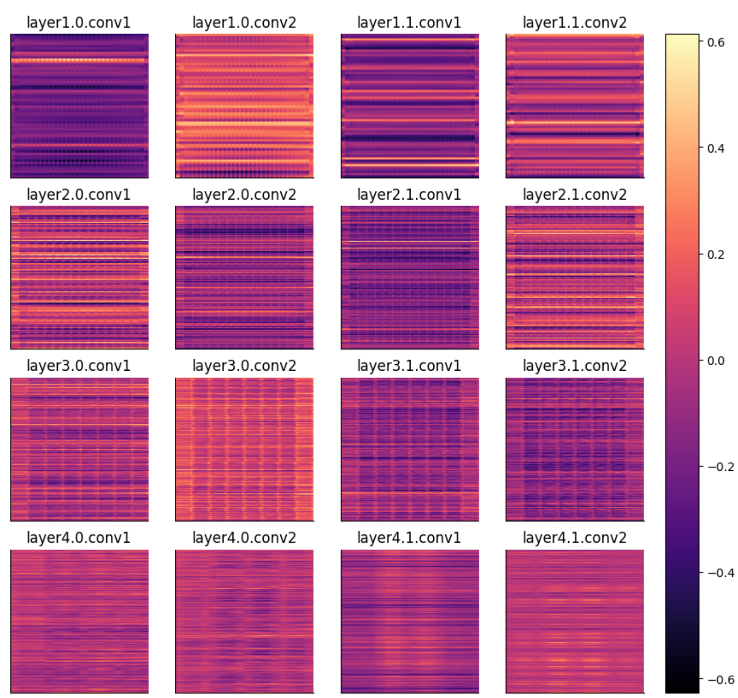

Figure 12: Sorted feature map visualization for all convolutional layers on PreAct-ResNet18 with the features in descending order of TAC values for **clean** model.

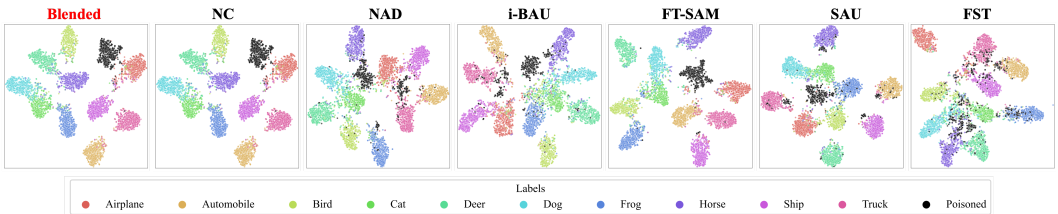

Figure 13: Comprison of T-SNE visualization between **Blended** attack model and different defense models on CIFAR-10.

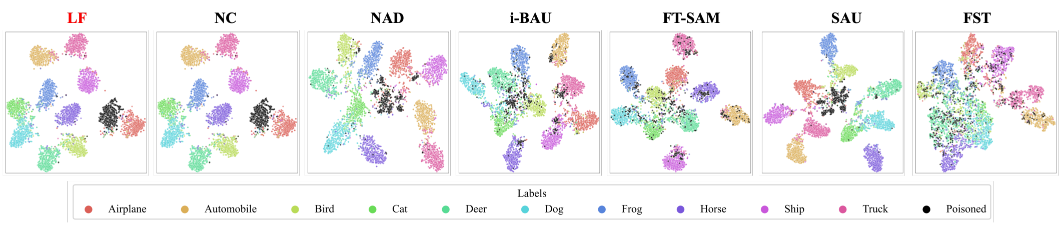

Figure 14: Comprison of T-SNE visualization between **LF** attack model and different defense models on CIFAR-10.

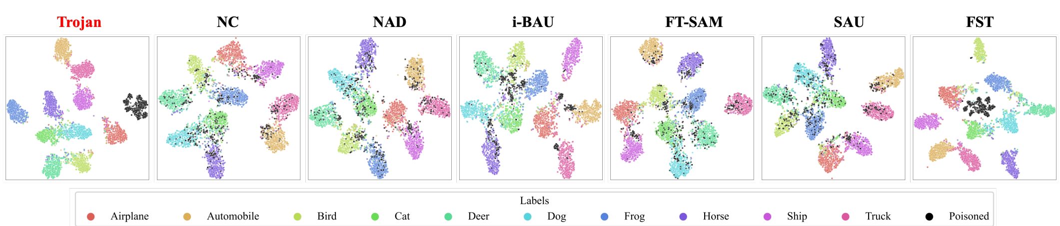

Figure 15: Comprison of T-SNE visualization between **Trojan** attack model and different defense models on CIFAR-10.

NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

   Answer: [Yes] .

   Justification: The main claims presented in the abstract and introduction offer a precise overview of what was accomplished and investigated in the research, providing an accurate reflection of the paper's scope and contributions.

   Guidelines:

   - The answer NA means that the abstract and introduction do not include the claims made in the paper.
   - The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A No or NA answer to this question will not be perceived well by the reviewers.
   - The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
   - It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer: [Yes] .

   Justification: The paper discusses the limitations of the work. See the "Limitations and future work" section for details.

   Guidelines:

   - The answer NA means that the paper has no limitation while the answer No means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory Assumptions and Proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

Answer: [NA]

Justification: This paper does not include theoretical results.

Guidelines:

- The answer NA means that the paper does not include theoretical results.
- All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
- All assumptions should be clearly stated or referenced in the statement of any theorems.
- The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
- Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
- Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental Result Reproducibility**

Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

Answer: [Yes]

Justification: The paper provides detailed information on the experimental setup for the reproduction of experimental results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- If the paper includes experiments, a No answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
- If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
- Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
- While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
  (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
  (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
  (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).
  (d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

Answer: [Yes]

Justification: We provided the code via an anonymous website.

Guidelines:

- The answer NA means that paper does not include experiments requiring code.
- Please see the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- While we encourage the release of code and data, we understand that this might not be possible, so "No" is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
- The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (https://nips.cc/public/guides/CodeSubmissionPolicy) for more details.
- The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
- The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
- At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
- Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental Setting/Details**

Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer, etc.) necessary to understand the results?

Answer:[Yes]

Justification: We outline all details related to the training and test processes, including data splits, hyperparameters, their selection process, and the types of optimizers used, thereby providing a comprehensive understanding of the results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
- The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment Statistical Significance**

Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

Answer: [Yes]

Justification: We have considered the randomness and run the experiments multi times for main results.

Guidelines:

- The answer NA means that the paper does not include experiments.
- The authors should answer "Yes" if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.

- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g. negative error rates).
- If error bars are reported in tables or plots, The authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments Compute Resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: We have clearly reported the resources used for experiments.

   Guidelines:

   - The answer NA means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code Of Ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics https://neurips.cc/public/EthicsGuidelines?

   Answer: [Yes]

   Justification: The research conducted in the paper conforms, in every respect, with the NeurIPS Code of Ethics

   Guidelines:

   - The answer NA means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer No, they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader Impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [Yes]

    Justification: We have discussed it in "Broader Impacts" section.

    Guidelines:

    - The answer NA means that there is no societal impact of the work performed.

- If the authors answer NA or No, they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pretrained language models, image generators, or scraped datasets)?

    Answer: [Yes]

    Justification: Our work has no such risks.

    Guidelines:

    - The answer NA means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [Yes]

    Justification: Properly credited.

    Guidelines:

    - The answer NA means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.
    - The authors should state which version of the asset is used and, if possible, include a URL.
    - The name of the license (e.g., CC-BY 4.0) should be included for each asset.
    - For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.

- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New Assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [NA]

    Justification: The paper does not release new assets.

    Guidelines:

    - The answer NA means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and Research with Human Subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [NA]

    Justification: The paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional Review Board (IRB) Approvals or Equivalent for Research with Human Subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [NA]

    Justification: The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.

    Guidelines:

    - The answer NA means that the paper does not involve crowdsourcing nor research with human subjects.
    - Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.

- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.