Injection, Attack and Erasure: Revocable Backdoor Attacks via Machine Unlearning

Baogang Song[1], Dongdong Zhao[1*], Jianwen Xiang[1], Qiben Xu[1], Zizhuo Yu[1]

[1]Engineering Research Center of Transportation Information and Safety (ERCTIS), MoE of China, School of Computer Science and Artificial Intelligence, Wuhan University of Technology, Wuhan, China
297710@whut.edu.cn, zdd@whut.edu.cn, jwxiang@whut.edu.cn, xqb@whut.edu.cn, yuzizhuo@whut.edu.cn

Abstract

Backdoor attacks pose a persistent security risk to deep neural networks (DNNs) due to their stealth and durability. While recent research has explored leveraging model unlearning mechanisms to enhance backdoor concealment, existing attack strategies still leave persistent traces that may be detected through static analysis. In this work, we introduce the first paradigm of revocable backdoor attacks, where the backdoor can be proactively and thoroughly removed after the attack objective is achieved. We formulate the trigger optimization in revocable backdoor attacks as a bilevel optimization problem: by simulating both backdoor injection and unlearning processes, the trigger generator is optimized to achieve a high attack success rate (ASR) while ensuring that the backdoor can be easily erased through unlearning. To mitigate the optimization conflict between injection and removal objectives, we employ a deterministic partition of poisoning and unlearning samples to reduce sampling-induced variance, and further apply the Projected Conflicting Gradient (PCGrad) technique to resolve the remaining gradient conflicts. Experiments on CIFAR-10 and ImageNet demonstrate that our method maintains ASR comparable to state-of-the-art backdoor attacks, while enabling effective removal of backdoor behavior after unlearning. This work opens a new direction for backdoor attack research and presents new challenges for the security of machine learning systems.

Extended version — https://arxiv.org/abs/2510.13322

Introduction

Deep neural networks (DNNs) have achieved remarkable breakthroughs in recent years and are now widely used in critical domains such as image recognition and natural language processing. However, as DNN models become more prevalent, concerns about their security have also become increasingly prominent. Recent studies have demonstrated that DNN systems are susceptible to a variety of security threats in real-world scenarios, including adversarial examples (Yuan et al. 2019), model stealing (Tramèr et al. 2016), and data poisoning. Among these, backdoor attacks (Goldblum et al. 2022) have received considerable research attention due to their strong stealthiness and the significant harm they can cause.

*Corresponding Author.

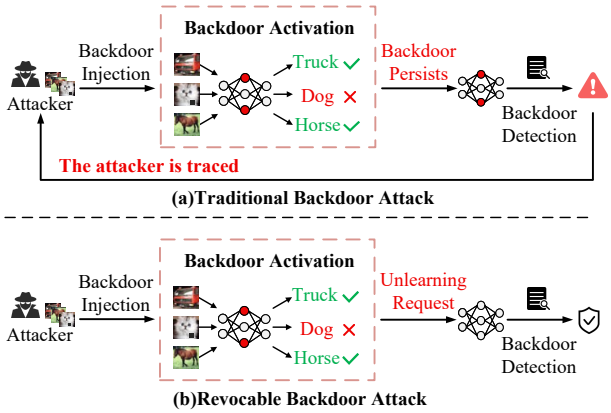


Figure 1: Research Motivation

In a typical backdoor attack, an adversary injects a small number of poisoned samples embedded with specific triggers into the training dataset during the model training process. Consequently, the resulting model performs normally on clean inputs but can be manipulated to output an attacker-chosen target label when a trigger is present in the input. Because triggers are usually designed to be highly covert and only activate the backdoor when present in the input, backdoor attacks are difficult to detect and defend against with conventional security methods. As researchers continue to develop more advanced backdoor detection and defense techniques, attackers are also exploring increasingly sophisticated and covert attack strategies, leading to an ongoing arms race in the field (Bai et al. 2024). Recently, a novel attack paradigm has emerged. This paradigm (Zhang et al. 2023) exploits machine unlearning mechanisms to manipulate backdoor states. Originally, the unlearning mechanism was introduced to address privacy regulations (such as GDPR (Otto 2018)), with the goal of enabling models to completely eliminate the influence of specific data on their predictions. However, recent studies (Zhang et al. 2023; Liu et al. 2024; Huang, Mao, and Zhong 2024) have revealed that attackers can cleverly abuse this mechanism: they can conceal backdoor effects and later reactivate backdoor behavior after model deployment by leveraging targeted unlearning operations.

Backdoor attacks based on unlearning mechanisms introduce the notion of delayed activation, where the model behaves normally at deployment and only activates the backdoor when triggered by a specific operation, thus greatly enhancing stealth. However, neither traditional backdoor attacks nor those utilizing unlearning mechanisms can eliminate the persistent risks that the long-term presence of backdoors poses for attackers. As illustrated in Figure 1, once a backdoor attack is carried out, the associated backdoor features remain embedded in the model's parameter or activation space throughout the model's lifecycle. Defenders can detect traces of backdoors at any stage of the model's lifecycle by using static analysis methods such as model pruning (Liu, Dolan-Gavitt, and Garg 2018), trigger pattern inversion (Wang et al. 2019), and activation clustering (Chen et al. 2018). As a result, attackers can hardly avoid the risk of passive exposure. This situation raises an important but underexplored question: Can attackers actively revoke backdoors from a model after the attack is completed, thereby thoroughly erasing traces of the attack and evading long-term detection and tracking? Importantly, to achieve true stealthiness and practicality in real-world scenarios, such revocation should require unlearning only a minimal subset of samples, rather than resorting to large-scale data deletion, which could itself arouse suspicion during routine model management or compliance audits. If attackers can achieve this, it would significantly improve the stealthiness and risk avoidance of backdoor attacks, while also posing new challenges for current defense mechanisms:

1. **Attribution difficulty** Attackers could strategically remove critical samples after achieving their objectives or just before a model audit, causing the model to behave normally and fundamentally reducing the likelihood of attribution and accountability.
2. **Static detection invalidation** Most existing static backdoor detection methods (Chen et al. 2018; Liu, Dolan-Gavitt, and Garg 2018; Wang et al. 2019) assume that backdoor features persist in a model's parameters or activation space. If backdoors can be actively revoked, these static detection tools become ineffective, forcing defenders to invest more resources in dynamic defenses such as continuous online monitoring and behavioral log auditing, to promptly detect and track backdoor attacks.

To address this gap, we propose a novel paradigm of revocable backdoor attacks, which allows attackers to actively remove backdoors after achieving their goals, thereby thoroughly erasing attack traces and significantly enhancing stealth. Our approach adopts an alternating optimization framework: the trigger generator is trained to maximize attack success on a surrogate model while minimizing effectiveness after unlearning, ensuring the resulting triggers are both effective and easily revocable. To mitigate the optimization conflict between these objectives, we employ a deterministic sample partition during training and apply PCGrad to further reduce conflict and stabilize optimization. After training, the attacker uses the learned trigger generator to inject backdoors into the target model. Once the attack objective is reached, a forgetting request with clean labels is submitted, prompting unlearning to remove the backdoor. Experiments on CIFAR-10 and ImageNet show that our method maintains high primary task accuracy and attack success rates, while the backdoor can be significantly weakened or removed through unlearning, outperforming traditional backdoor attacks in stealth and revocability.

The main contributions are as follows:

- We are the first to propose a revocable backdoor attack paradigm that leverages the model unlearning interface, enabling proactive backdoor removal and significantly enhancing attack stealth. This opens a new direction for backdoor attack research and highlights critical security challenges.
- We design an bilevel optimization-based trigger generator, jointly training surrogate and unlearning models to simulate poisoning and unlearning, and optimize trigger effectiveness and revocability.
- To address conflicts in trigger optimization, we introduce fixed poisoned/unlearning samples and employ the PCGrad technique, effectively stabilizing the optimization process.
- We evaluate the effectiveness of our method on datasets including CIFAR-10 and ImageNet. Results indicate that our method achieves attack success rates comparable to mainstream backdoor attacks, and that backdoor effects can be significantly weakened or removed after unlearning.

Related Work

Backdoor Attacks

Backdoor attacks pose a serious threat to deep neural networks. Early studies such as BadNets (Gu, Dolan-Gavitt, and Garg 2017) and Trojaning Attack (Liu et al. 2018) showed that injecting poisoned samples with simple triggers can yield high ASR on triggered inputs. Follow-up work improved stealthiness and robustness via blended patterns (Chen et al. 2017), image warping (Nguyen and Tran 2021), invisible triggers (Li et al. 2020), and semantic triggers (Saha, Subramanya, and Pirsiavash 2020). More recent studies (Zhang et al. 2023; Liu et al. 2024; Huang, Mao, and Zhong 2024) explored advanced strategies, including delayed backdoors leveraging machine unlearning. Yet existing approaches cannot proactively and thoroughly erase the backdoor; attack traces often persist in model parameters and remain detectable or traceable (Xu et al. 2021; Chen et al. 2019). A prior work (Xu et al. 2024) claims "revocable" backdoors, but is limited to scenarios such as model trading, requires modifying model parameters, and differs fundamentally from our paradigm. In contrast, our approach targets proactive and generalizable backdoor removal via machine unlearning from the attacker's perspective.

Machine Unlearning

Machine unlearning was initially proposed to meet privacy and regulatory requirements such as GDPR, focusing on data removal and certified deletion in trained models. Mainstream unlearning methods include influence-based data re-

moval (Koh and Liang 2017), approximate retraining (Warnecke et al. 2023; Thudi et al. 2022), and certified forgetting (Chien et al. 2024), which have shown effectiveness in erasing individual data points or entire subsets from a model's memory. Recently, the intersection between machine unlearning and security has drawn increasing attention. Some studies (Liu et al. 2022) have explored leveraging unlearning mechanisms to mitigate backdoor attacks or defend against data poisoning. Nevertheless, these efforts primarily aim to enhance model robustness or privacy, rather than enabling attackers to actively control or erase backdoor traces. To the best of our knowledge, our work is the first to systematically exploit machine unlearning as a positive tool for active backdoor revocation from the attacker's perspective, achieving both attack effectiveness and removability.

Revocable Backdoor Attack

In this section, we first formally define the concept of a revocable backdoor attack and then elaborate on the trigger optimization process.

Problem Definition

As illustrated in Figure 2, a revocable backdoor attack can be divided into three stages: trigger generation, backdoor injection, and backdoor revocation.

1. **Trigger optimization:** Unlike traditional backdoor attacks, our approach incorporates revocability as a core consideration during trigger optimization. Specifically, the trigger is designed not only to ensure a high ASR, but also to enable effective removal after model unlearning, thereby evading detection and long-term tracking.
2. **Backdoor injection:** Once the trigger is optimized, the attacker injects trigger samples into the training dataset. These samples are embedded in the victim model either through data upload or by submitting model training requests to the service provider.
3. **Backdoor revocation:** After achieving the attack goal, the attacker sends a forgetting request to the model provider, deletes some trigger-containing training data, and removes the backdoor via the legal model unlearning interface, thus hiding attack traces.

We formally define the **Revocable Backdoor Attack** as the following optimization problem: Given a classification model f_θ and its training dataset $\mathcal{D} = (x_i, y_i)$, where $\mathcal{X}$ and $\mathcal{Y}$ denote the input and label spaces, and $x_i \in \mathcal{X}$ and $y_i \in \mathcal{Y}$. The attacker's goal is to inject a backdoor into the model such that it can be reliably activated by a trigger, but subsequently revoked via an unlearning request. Specifically, the attacker first selects a target label $y_{\text{target}} \in \mathcal{Y}$ and extracts a subset of samples with the target label, denoted as $\mathcal{P} \subset \mathcal{D}$, to serve as the poisoned set (with poisoning rate ρ_p). From this set, a further subset $\mathcal{U} \subset \mathcal{P}$ is sampled as the forgetting set (with forgetting rate ρ_f), which will be used for backdoor revocation. Next, the attacker applies a trigger generator function G to these samples, producing the poisoned set $\mathcal{P}' = (G(x), y) \mid (x, y) \in \mathcal{P}$. The attacker then constructs a mixed training set $\mathcal{D}_b = (\mathcal{D} \setminus \mathcal{P}) \cup \mathcal{P}'$ and submits it to the model owner for training. After training, the model parameters are updated to θ_b, resulting in a backdoored model f_{θ_b}, which behaves normally on clean inputs but predicts the target label y_{target} for inputs containing the trigger $G(x)$:

$$f_{\theta_b}(x_i) = y_i,\ f_{\theta_b}(G(x_i)) = y_{target} \quad \forall x_i, y_i \in \mathcal{X}, \mathcal{Y} \quad (1)$$

After the attack objective is achieved, the attacker submits a forgetting request for the subset $\mathcal{U}$, prompting the model owner to invoke standard or approximate unlearning algorithms (such as fine-tuning, first-order or unrollsgd) to update the model parameters to θ_u, thus obtaining the unlearned model f_{θ_u}. If the trigger is no longer effective, the backdoor is successfully revoked, and traces of the attack are effectively erased:

$$f_{\theta_u}(x_i) = y_i,\ f_{\theta_u}(G(x_i)) = y_i \quad \forall x_i, y_i \in \mathcal{X}, \mathcal{Y} \quad (2)$$

Therefore, our goal is to find a trigger generator G such that, after training the model following the above workflow, the resulting models satisfy the following two properties: the trigger can reliably activate the backdoor before unlearning (Eq. 1), but loses its effect after unlearning (Eq. 2). The revocable backdoor attack is thus formulated as the problem of designing and optimizing G to satisfy Eq. 1 and Eq. 2 simultaneously.

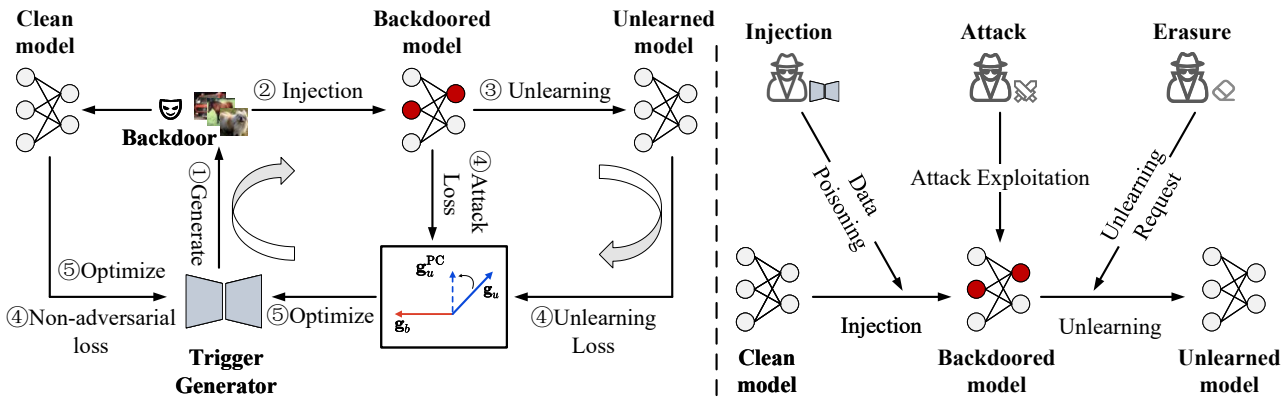
Threat Model

- **Attacker's Capability** The attacker is able to inject or modify a small number of samples with backdoor triggers in the model's training set but cannot interfere with the overall training process or the deployment of the final model. In addition, the attacker can utilize the data unlearning interface provided by the model service provider (e.g., for GDPR compliance) to request the model to forget a specified small subset of training samples. All poisoned samples must retain their true labels—that is, only clean-label backdoor attacks are considered. This clean-label setting is adopted for two main reasons: First, in practical scenarios, attackers typically lack the ability to alter data labels, making clean-label attacks more realistic. Second, it prevents exposure during the revocation phase, as requesting unlearning for mislabeled (dirty-label) samples would easily arouse suspicion from the service provider.
- **Attacker's Objective** The attacker aims to ensure that, when the trigger is present, the model predicts the attacker-specified target label with a high ASR. After the designated samples are forgotten, the model becomes fully insensitive to the trigger, which makes the attack difficult to detect or trace afterward. Unlike traditional backdoor attacks, which emphasize persistence, our work centers on **revocability**: the attacker seeks not only high ASR but also the ability to proactively and thoroughly erase all traces of the backdoor after achieving their goal, thereby significantly enhancing the stealth and safety of the attack.

Trigger Generation and Optimization

Design of the Trigger Generator We define the trigger as an input perturbation function:

$$G(x) = x + \eta * g(x) \quad (3)$$



(a) Trigger Optimization Process (b) Revocable backdoor attack process

Figure 2: An illustration of the proposed framework

where g is a learnable generative network and η represents the perturbation strength. To address high-frequency artifacts and spatial discontinuities (Zeng et al. 2021), we incorporate design principles inspired by the COMBAT method (Huynh et al. 2024). Specifically, we first apply the discrete cosine transform (DCT) (Ahmed, Natarajan, and Rao 2006) to $g(x)$ to constrain the trigger's frequency domain distribution. A mask m is then used to control the retained frequency components. Subsequently, we reconstruct the trigger noise in the spatial domain using the inverse DCT (IDCT), resulting in a trigger that is dominated by low-frequency components. To further enhance spatial continuity and stealth, we apply a Gaussian blur filter k to the generated trigger. By integrating these steps, we obtain a highly covert trigger function for backdoor attacks. The final trigger function is defined as:

$$\mathcal{G}(g(x)) = \mathrm{IDCT}(m \odot \mathrm{DCT}(g(x))) \tag{4}$$

$$G(x) = (x + \eta * \mathcal{G}(g(x))) * k \tag{5}$$

where $\odot$ denotes Hadamard product.

Bilevel Optimization In revocable backdoor attacks, the core challenge lies in the fact that the effectiveness of the trigger depends not only on the model's behavior after backdoor injection, but also on its behavior after subsequent unlearning operations. Unlike conventional backdoor attacks, which focus solely on attack effectiveness during the backdoor injection phase, revocable backdoor optimization must fully account for the model's state after unlearning. Therefore, when optimizing the trigger, we explicitly simulate both the backdoor injection and unlearning processes to ensure that the trigger achieves its intended objectives at each stage. This necessity motivates us to formulate the training of the trigger generator as a bilevel optimization problem. Specifically, in the bilevel optimization framework, the **outer level** optimizes the parameters of the trigger generator, while the **inner level**, given the current trigger generator, sequentially trains the backdoored model and performs the unlearning operation, thereby providing optimization signals from both the backdoor injection and unlearning phases. It is important to note that in real-world attack settings, the attacker is typically unable to access the internals of the victim's deployed model or intervene in its training process. To address this limitation, we adopt a surrogate training approach, in which a locally constructed shadow model is built to match the architecture of the victim model as closely as possible. We then utilize auxiliary data accessible to the attacker to simulate the processes of backdoor injection and unlearning on this shadow model.

In the outer-level trigger optimization, we focus on two core loss terms:

- **Attack loss**: In the outer-level trigger optimization, the attack loss is computed by applying the trigger to all samples in the dataset, assigning them the target label y_{target}, and feeding the triggered samples into the backdoored model for evaluation:

$$\mathcal{L}_{attack} = \mathbb{E}_{(x,y)\in\mathcal{D}}\, \mathcal{L}\left(f_{\theta_b}(G(x)),\, y_{target}\right) \tag{6}$$

  This design aims to encourage the learned trigger to be generally applicable, rather than limited to a specific subset of samples.

- **Unlearning loss**: Similarly, the unlearning loss is computed by applying the trigger to all samples and evaluating the triggered samples with the unlearned model, using the ground-truth labels y for comparison:

$$\mathcal{L}_{unlearn} = \mathbb{E}_{(x,y)\in\mathcal{D}}\, \mathcal{L}\left(f_{\theta_u}(G(x)),\, y\right) \tag{7}$$

  This loss encourages the complete revocation of the backdoor effect, i.e., the unlearned model should no longer respond to the trigger.

These two losses jointly define the main objectives of trigger optimization, i.e., effectiveness and revocability. Furthermore, we also employ the following two losses:

- **Visibility loss.** To ensure the trigger remains visually inconspicuous, we penalize visually salient perturbations using the following regularization:

$$\mathcal{L}_{vis} = \mathbb{E}_{x\in\mathcal{X}}\left[|\eta * \mathcal{G}(g(x))|_2^2\right] \tag{8}$$

- **Non-adversarial loss.** To prevent the trigger from acting as a standard adversarial perturbation and to maintain the specificity of the backdoor, we require that the clean model makes consistent predictions on triggered and clean samples:

$$\mathcal{L}_{non\text{-}adv} = \mathbb{E}_{(x,y)\in\mathcal{D}}\, \mathcal{L}\left(f_{\theta_{clean}}(G(x)),\, y\right) \tag{9}$$

Combining the above objectives, the trigger generator is trained to solve the following bilevel optimization problem:

$$G^* = \arg\min_{G}\ \mathbb{E}_{(x,y)\in\mathcal{D}}\Big[\mathcal{L}_{attack} + \lambda_{unlearn}\cdot\mathcal{L}_{unlearn} + \lambda_{vis}\cdot\mathcal{L}_{vis} + \lambda_{non\text{-}adv}\cdot\mathcal{L}_{\text{non-adv}}\Big] \tag{10}$$

$$\text{s.t.}\quad \theta_b = \arg\min_{\theta}\ \mathbb{E}_{(x,y)\in\mathcal{D}_b}\,\mathcal{L}(f_\theta(x), y)\,,$$

$$\theta_u = \mathcal{F}_{unlearn}(\theta_b, \mathcal{U}).$$

where $\lambda_{unlearn}$, λ_{vis}, and $\lambda_{non\text{-}adv}$ are the weights for the unlearning, visibility, and non-adversarial losses, respectively. $\mathcal{F}_{unlearn}(\theta_b, \mathcal{U})$ denotes the model unlearning algorithm that updates the backdoored model parameters θ_b by removing the influence of the specified forgetting set $\mathcal{U}$.

Due to the strong coupling and mutual dependencies among parameters in the above optimization problem, direct end-to-end optimization of the overall objective is difficult to converge. Specifically, the backdoored and unlearned models are trained on poisoned data produced by the current trigger generator, while updating the generator depends on feedback from both models. This circular dependency creates a complex, tightly coupled optimization problem with interdependent objectives. Therefore, we adopt an alternating optimization strategy: in each training round, we first fix the trigger generator parameters to train the backdoored and unlearned models, and then fix the model parameters to optimize the generator.

Gradient Conflict Mitigation In the bilevel optimization process, maximizing the ASR of the backdoored model and minimizing the ASR of the unlearned model are inherently conflicting objectives: increasing the backdoored model's ASR with respect to the trigger often makes it more difficult for the unlearned model to revoke the backdoor, and vice versa. This conflict manifests as inconsistency in the directions of the gradients of the loss functions during optimization; in severe cases, the gradients may even cancel each other out, undermining the stability and final performance of trigger generator training. Empirically, we observe that the gradient directions of these two losses often exhibit significant negative correlation during training. To effectively alleviate such gradient conflicts between optimization objectives, we adopt the following two mechanisms:

1. **Deterministic partition:** In practical backdoor attacks, adversaries typically select a specific set of samples for poisoning, which aligns with our use of a deterministic partition. While many existing methods simulate backdoor injection by randomly sampling data during optimization in order to generate triggers that generalize across the entire dataset, we instead fix the sample partition throughout the optimization process. This approach more faithfully reflects realistic attack settings, reduces sampling-induced variance, and allows for more stable optimization of the trigger generator.
2. **PCGrad gradient projection:** When the gradient directions of the two loss functions conflict (i.e., their inner product is negative), we employ the PCGrad (Projected Conflicting Gradient) algorithm (Yu et al. 2020) to orthogonally project the gradients and remove conflicting components. Specifically, let the gradients of the attack loss and unlearning loss with respect to the generator G be denoted as $\mathbf{g}_b$ and $\mathbf{g}_u$, respectively:

$$\mathbf{g}_b = \nabla_G \mathcal{L}_{attack} \tag{11}$$

$$\mathbf{g}_u = \nabla_G \mathcal{L}_{unlearn} \tag{12}$$

When $\langle \mathbf{g}_b, \mathbf{g}_u \rangle < 0$, we project $\mathbf{g}_u$ onto the orthogonal direction of $\mathbf{g}_b$ as follows:

$$\mathbf{g}_u^{\text{PC}} = \mathbf{g}_u - \alpha \cdot \frac{\langle \mathbf{g}_b, \mathbf{g}_u \rangle}{\|\mathbf{g}_b\|^2}\mathbf{g}_b \tag{13}$$

where α is a hyperparameter controlling the projection magnitude and $\langle \cdot, \cdot \rangle$ denotes the standard inner product.

Experiment

Experimental Setup

Datasets and Models We evaluate the proposed method on two benchmark image classification datasets: **CIFAR-10** (Krizhevsky, Hinton et al. 2009) and **IMAGENET-10**. IMAGENET-10 is constructed by randomly selecting 10 classes from the standard ImageNet (Deng et al. 2009) dataset. For model architectures, we use the Pre-activation ResNet-18 (He et al. 2016b) for CIFAR-10 and the standard ResNet-18 (He et al. 2016a) for IMAGENET-10. The trigger generator is implemented with a U-NET (Ronneberger, Fischer, and Brox 2015) backbone in all experiments.

Parameter and Attack Settings All classifiers are trained for 200 epochs with SGD. The batch size is 128 for CIFAR-10 and 32 for IMAGENET-10. Initial learning rates are 0.01 for CIFAR-10, 0.001 for IMAGENET-10, and 0.01 for the trigger generator, with decay by a factor of 10 at the 100th and 150th epochs. Class 0 is used as the target label in all attack experiments. The poisoning rate is fixed at 5% in all experiments. For high-frequency removal, following COMBAT, we set the frequency mask ratio $r = 0.65$ and apply a Gaussian blur filter (kernel size 3, σ uniformly sampled from $[0.1, 1]$). The hyperparameters α, $\lambda_{\text{non-adv}}$, λ_{unlearn}, and λ_{vis} are set to 0.6, 0.8, 1.0, and 0.02, respectively. The PCGrad projection coefficient is fixed at 0.6. Unlearning is simulated using both the First-Order (Warnecke et al. 2023) and UnrollSGD (Thudi et al. 2022) approximation methods. For each unlearning operation, 250 samples are removed from CIFAR-10 and 100 samples from IMAGENET-10. The forgetting rate for the First-Order setting is 0.01 (CIFAR-10) and 0.001 (IMAGENET-10); for UnrollSGD, CIFAR-10 is fine-tuned for 3 epochs and IMAGENET-10 for 2 epochs. Unless otherwise specified, all experiments are conducted on CIFAR-10, using the first-order unlearning strategy for both simulation and evaluation.

Dataset	Unlearn Method	First-Order				UnrollSGD			
		ASR	ASR-U(Δ)	BA	BA-U	ASR	ASR-U(Δ)	BA	BA-U
CIFAR-10	First-Order	98.36	25.12(-73.24)	94.34	90.05	98.36	4.42(-93.94)	94.34	82.93
	UnrollSGD	97.33	12.93(-84.40)	94.21	91.03	97.33	0.11(-97.22)	94.21	83.29
IMAGENET-10	First-Order	79.33	13.70(-65.63)	85.92	78.25	79.33	0.00(-79.33)	85.92	71.67
	UnrollSGD	85.19	16.85(-68.34)	85.08	78.92	85.19	7.04(-78.15)	85.08	72.42

Table 1: Performance of Our Method under Different Unlearning Strategies on CIFAR-10 and ImageNet-10 (%)

Method	First-Order				UnrollSGD			
	ASR	ASR-U(Δ)	BA	BA-U	ASR	ASR-U(Δ)	BA	BA-U
Badnets	29.8	6.67(-23.13)	94.47	92.19	29.8	1.12(-28.68)	94.47	89.33
Wanet	65.28	29.51(-35.77)	90.45	89.54	65.28	6.41(-58.87)	90.45	89.81
Sleeper Agent	90.26	57.47(-32.79)	93.96	91.89	90.26	18.57(-71.73)	93.96	83.56
COMBAT	**98.53**	45.97(-52.56)	94.38	90.38	**98.53**	17.56(-80.97)	94.38	82.28
Our Method (First-Order)	<u>98.36</u>	<u>25.12(-73.24)</u>	94.34	90.05	<u>98.36</u>	<u>4.42(-93.94)</u>	94.34	82.93
Our Method (UnrollSGD)	97.33	**12.93(-84.40)**	94.21	91.03	97.33	**0.11(-97.22)**	94.21	83.29

Table 2: Comparison with state-of-the-art methods on CIFAR-10 (%). **Bold** indicates the best result, and <u>underline</u> indicates the second-best result in each column.

Baselines and Evaluation Metrics To the best of our knowledge, there is currently no public research specifically targeting revocable backdoor attacks. For comprehensive comparison, we introduce several mainstream backdoor attack baselines (including BadNets (Gu, Dolan-Gavitt, and Garg 2017), WaNet (Nguyen and Tran 2021), Sleeper Agent (Souri et al. 2022), and COMBAT (Huynh et al. 2024)), and evaluate their performance changes after unlearning under the same settings. The evaluation metrics used are as follows: **Attack Success Rate (ASR)**, which measures the proportion of trigger samples classified as the target label; **ASR after Unlearning (ASR-U)**; **Benign Accuracy (BA)**, the classification accuracy on clean test samples; and **Benign Accuracy after Unlearning (BA-U)**.

Attack Experiments

We evaluate our method under First-Order and UnrollSGD unlearning on CIFAR-10 and ImageNet-10. As shown in Table 1, our approach consistently achieves high ASR and BA, with ASR-U dropping sharply after unlearning—especially under UnrollSGD, where ASR-U falls to 4.42% on CIFAR-10 and 0.00% on ImageNet-10, while BA-U remains competitive. Furthermore, as shown in Table 2, our comparison with Badnets, Wanet, Sleeper Agent, and COMBAT on CIFAR-10 demonstrates that our method not only achieves comparable or higher ASR but also yields substantially lower ASR-U after unlearning, particularly under UnrollSGD (0.11%), highlighting its superior revocability. The detailed experimental settings for all compared methods are provided in the Appendix. These results confirm that our approach effectively balances attack effectiveness and removability, and outperforms existing methods in terms of revocable backdoor attacks.

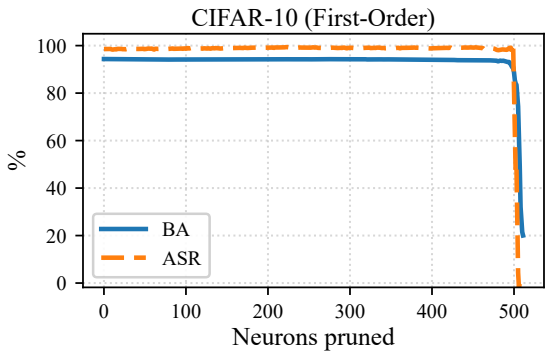


Figure 3: Experiment results of evaluating Our Method against Fine-pruning

Defense Experiments

We evaluate our revocable backdoor using the fine-pruning defense, which prunes neurons inactive on clean samples. As shown in Fig. 3, both BA and ASR remain high across a wide range of pruning. Only when a large number of neurons are pruned do both BA and ASR drop sharply, indicating that fine-pruning cannot effectively remove our backdoor without significantly damaging normal model performance. We further evaluate additional mainstream defenses on both CIFAR-10 and ImageNet-10, with detailed results provided in the Appendix.

Parameter Sensitivity Analysis

We conduct parameter sensitivity analysis on two hyperparameters: poisoning rate ρ_p and trigger strength η. As shown in Table 3, increasing ρ_p leads to higher ASR and ASR-

$\rho_p(\%)$	ASR	ASR-U(Δ)	BA	BA-U
0.5	69.73	5.34(-64.39)	94.76	90.81
1	71.98	7.62(-64.36)	94.38	90.74
2	84.38	12.19(-72.19)	94.76	91.09
3	89.93	13.00(-76.93)	94.50	90.75
5	98.36	25.12(-73.24)	94.34	90.05

Table 3: Performance under different ρ_p on CIFAR-10

η	ASR	ASR-U(Δ)	BA	BA-U
0.04	71.82	9.71(-62.11)	94.52	90.03
0.08	98.36	25.12(-73.24)	94.34	90.05
0.12	99.91	44.40(-55.51)	94.41	90.82
0.16	99.81	41.24(-58.57)	94.40	90.85
0.2	99.82	40.08(-59.74)	94.33	90.61

Table 4: Performance under different η on CIFAR-10

U, while BA and BA-U remain stable. This indicates that a higher poisoning rate strengthens the attack but also makes the backdoor more difficult to remove via unlearning. Similarly, Table 4 shows that increasing η rapidly saturates ASR and substantially increases ASR-U, reducing revocability. When $\eta \geq 0.12$, further increases have little effect, indicating performance saturation. Across all settings, BA and BA-U remain largely unaffected. Overall, these results demonstrate a trade-off: while stronger triggers and higher poisoning rates improve attack effectiveness, excessive values do not yield further benefits and hinder unlearning-based backdoor revocation.

Gradient Conflict Analysis

We quantify the optimization conflict between backdoor injection and unlearning by computing the cosine similarity between their gradients with respect to the trigger generator parameters during training. Cosine similarity serves as a direct indicator of gradient alignment: more negative values imply stronger opposition and thus more severe conflict. We evaluate two settings: Baseline, which applies no mitigation, and Fixed Sample + PCGrad, which combines fixed sample selection with PCGrad gradient projection to reduce conflicting updates. As shown in Fig. 4, Baseline maintains a persistently negative similarity around -0.65, suggesting that the two objectives consistently pull the generator in opposite directions throughout optimization. In contrast, Fixed Sample + PCGrad increases the similarity to about -0.35 and stabilizes its trajectory, indicating that conflicting components are effectively suppressed and the two objectives become less antagonistic. This reduced conflict leads to a more stable and efficient training process for the revocable backdoor attack.

Ablation Study

To assess the contribution of each core module in our framework, we conduct ablation experiments by selectively removing the unlearning loss and the conflict mitigation strategy (Fixed Sample + PCGrad). The results on CIFAR-10 are summarized in Table 5. As shown, omitting the unlearning loss (w/o Unlearn) leads to a clear increase in ASR-U, indicating that the backdoor cannot be effectively revoked. Disabling the conflict mitigation strategy (w/o Mitigation) further exacerbates this issue, resulting in higher ASR-U. In contrast, our complete method (Ours) achieves the best balance, with high ASR, the lowest ASR-U, and BA comparable to other settings. These results confirm that both the unlearning loss and the conflict mitigation module are essential for achieving high attack effectiveness and strong revocability in revocable backdoor attacks.

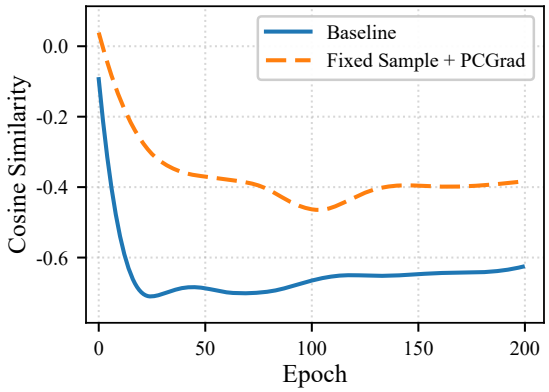


Figure 4: Cosine similarity between attack and unlearning gradients over training epochs

Setting	ASR	ASR-U(Δ)	BA	BA-U
w/o Unlearn	97.50	32.52(-64.98)	94.58	90.59
w/o Mitigation	98.57	68.60(-29.97)	94.51	91.34
Ours	98.36	25.12(-73.24)	94.34	90.05

Table 5: Ablation study of key modules on CIFAR-10

Conclusion

In this paper, we propose a revocable backdoor attack paradigm that leverages machine unlearning to proactively and thoroughly erase backdoor traces. Our bilevel optimization-based trigger generator balances attack effectiveness and revocability, while practical techniques such as clean-label poisoning, fixed sample selection, and gradient conflict mitigation promote stability and stealth. Experiments on CIFAR-10 and ImageNet show that our method achieves attack success rates on par with state-of-the-art attacks, while enabling efficient backdoor removal via unlearning and surpassing baselines in both stealth and revocability. This work highlights new challenges for backdoor defense. In future work, we will explore more general attack and defense scenarios, including adaptive unlearning strategies, adversarial model auditing, and new defenses against revocable backdoor attacks.

References

Ahmed, N.; Natarajan, T.; and Rao, K. R. 2006. Discrete cosine transform. *IEEE transactions on Computers*, 100(1): 90–93.

Bai, Y.; Xing, G.; Wu, H.; Rao, Z.; Ma, C.; Wang, S.; Liu, X.; Zhou, Y.; Tang, J.; Huang, K.; et al. 2024. Backdoor attack and defense on deep learning: A survey. *IEEE Transactions on Computational Social Systems*.

Chen, B.; Carvalho, W.; Baracaldo, N.; Ludwig, H.; Edwards, B.; Lee, T.; Molloy, I.; and Srivastava, B. 2018. Detecting backdoor attacks on deep neural networks by activation clustering. *arXiv preprint arXiv:1811.03728*.

Chen, H.; Fu, C.; Zhao, J.; and Koushanfar, F. 2019. Deepinspect: A black-box trojan detection and mitigation framework for deep neural networks. In *IJCAI*, volume 2, 8.

Chen, X.; Liu, C.; Li, B.; Lu, K.; and Song, D. 2017. Targeted backdoor attacks on deep learning systems using data poisoning. *arXiv preprint arXiv:1712.05526*.

Chien, E.; Wang, H.; Chen, Z.; and Li, P. 2024. Certified machine unlearning via noisy stochastic gradient descent. *Advances in Neural Information Processing Systems*, 37: 38852–38887.

Deng, J.; Dong, W.; Socher, R.; Li, L.-J.; Li, K.; and Fei-Fei, L. 2009. Imagenet: A large-scale hierarchical image database. In *2009 IEEE conference on computer vision and pattern recognition*, 248–255. Ieee.

Goldblum, M.; Tsipras, D.; Xie, C.; Chen, X.; Schwarzschild, A.; Song, D.; Madry, A.; Li, B.; and Goldstein, T. 2022. Dataset security for machine learning: Data poisoning, backdoor attacks, and defenses. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45(2): 1563–1580.

Gu, T.; Dolan-Gavitt, B.; and Garg, S. 2017. Badnets: Identifying vulnerabilities in the machine learning model supply chain. In *Proceedings of Machine Learning and Computer Security Workshop*.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016a. Deep residual learning for image recognition. In *Proceedings of the IEEE conference on computer vision and pattern recognition*, 770–778.

He, K.; Zhang, X.; Ren, S.; and Sun, J. 2016b. Identity mappings in deep residual networks. In *European conference on computer vision*, 630–645. Springer.

Huang, Z.; Mao, Y.; and Zhong, S. 2024. {UBA-Inf}: Unlearning activated backdoor attack with {Influence-Driven} camouflage. In *33rd USENIX Security Symposium (USENIX Security 24)*, 4211–4228.

Huynh, T.; Nguyen, D.; Pham, T.; and Tran, A. 2024. Combat: Alternated training for effective clean-label backdoor attacks. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 2436–2444.

Koh, P. W.; and Liang, P. 2017. Understanding black-box predictions via influence functions. In *International conference on machine learning*, 1885–1894. PMLR.

Krizhevsky, A.; Hinton, G.; et al. 2009. Learning multiple layers of features from tiny images. *Master's thesis*.

Li, S.; Xue, M.; Zhao, B. Z. H.; Zhu, H.; and Zhang, X. 2020. Invisible backdoor attacks on deep neural networks via steganography and regularization. *IEEE Transactions on Dependable and Secure Computing*, 18(5): 2088–2105.

Liu, K.; Dolan-Gavitt, B.; and Garg, S. 2018. Fine-pruning: Defending against backdooring attacks on deep neural networks. In *International symposium on research in attacks, intrusions, and defenses*, 273–294. Springer.

Liu, Y.; Fan, M.; Chen, C.; Liu, X.; Ma, Z.; Wang, L.; and Ma, J. 2022. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022-IEEE conference on computer communications*, 280–289. IEEE.

Liu, Y.; Ma, S.; Aafer, Y.; Lee, W.-C.; Zhai, J.; Wang, W.; and Zhang, X. 2018. Trojaning attack on neural networks. In *25th Annual Network And Distributed System Security Symposium (NDSS 2018)*. Internet Soc.

Liu, Z.; Wang, T.; Huai, M.; and Miao, C. 2024. Backdoor attacks via machine unlearning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, 14115–14123.

Nguyen, T. A.; and Tran, A. T. 2021. WaNet-Imperceptible Warping-based Backdoor Attack. In *International Conference on Learning Representations*.

Otto, M. 2018. Regulation (EU) 2016/679 on the protection of natural persons with regard to the processing of personal data and on the free movement of such data (General Data Protection Regulation–GDPR). In *International and European Labour Law*, 958–981. Nomos Verlagsgesellschaft mbH & Co. KG.

Ronneberger, O.; Fischer, P.; and Brox, T. 2015. U-net: Convolutional networks for biomedical image segmentation. In *International Conference on Medical image computing and computer-assisted intervention*, 234–241. Springer.

Saha, A.; Subramanya, A.; and Pirsiavash, H. 2020. Hidden trigger backdoor attacks. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, 11957–11965.

Souri, H.; Fowl, L.; Chellappa, R.; Goldblum, M.; and Goldstein, T. 2022. Sleeper agent: Scalable hidden trigger backdoors for neural networks trained from scratch. *Advances in Neural Information Processing Systems*, 35: 19165–19178.

Thudi, A.; Deza, G.; Chandrasekaran, V.; and Papernot, N. 2022. Unrolling sgd: Understanding factors influencing machine unlearning. In *2022 IEEE 7th European Symposium on Security and Privacy (EuroS&P)*, 303–319. IEEE.

Tramèr, F.; Zhang, F.; Juels, A.; Reiter, M. K.; and Ristenpart, T. 2016. Stealing machine learning models via prediction {APIs}. In *25th USENIX security symposium (USENIX Security 16)*, 601–618.

Wang, B.; Yao, Y.; Shan, S.; Li, H.; Viswanath, B.; Zheng, H.; and Zhao, B. Y. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE symposium on security and privacy (SP)*, 707–723. IEEE.

Warnecke, A.; Pirch, L.; Wressnegger, C.; and Rieck, K. 2023. Machine Unlearning of Features and Labels. In *Proceedings 2023 Network and Distributed System Security Symposium*. Internet Society.

Xu, X.; Wang, Q.; Li, H.; Borisov, N.; Gunter, C. A.; and Li, B. 2021. Detecting ai trojans using meta neural analysis. In *2021 IEEE Symposium on Security and Privacy (SP)*, 103–120. IEEE.

Xu, Y.; Zhong, N.; Qian, Z.; and Zhang, X. 2024. Revocable Backdoor for Deep Model Trading. *arXiv preprint arXiv:2408.00255*.

Yu, T.; Kumar, S.; Gupta, A.; Levine, S.; Hausman, K.; and Finn, C. 2020. Gradient surgery for multi-task learning. *Advances in neural information processing systems*, 33: 5824–5836.

Yuan, X.; He, P.; Zhu, Q.; and Li, X. 2019. Adversarial examples: Attacks and defenses for deep learning. *IEEE transactions on neural networks and learning systems*, 30(9): 2805–2824.

Zeng, Y.; Park, W.; Mao, Z. M.; and Jia, R. 2021. Rethinking the backdoor attacks' triggers: A frequency perspective. In *Proceedings of the IEEE/CVF international conference on computer vision*, 16473–16481.

Zhang, P.; Sun, J.; Tan, M.; and Wang, X. 2023. Exploiting Machine Unlearning for Backdoor Attacks in Deep Learning System. arXiv:2310.10659.