Computers, Materials & Continua

Tech Science Press

Doi:10.32604/cmc.2025.072458

ARTICLE

A Game-Theoretic Framework for Strategic Machine Unlearning in Backdoor Mitigation

Xiaolei Ding and Wenjian Liu*

Faculty of Data Science, City University of Macau, Macau, China

*Corresponding Author: Wenjian Liu. Email: andylau@cityu.edu.mo

Received: 27 August 2025; Accepted: 29 September 2025; Published: 15 June 2026

ABSTRACT: Backdoor attacks pose a critical threat to the reliability and trustworthiness of machine learning models, as they allow adversaries to manipulate model behavior through the injection of malicious patterns during training. Existing defenses, such as data filtering, fine-tuning, and model pruning, often lack provable guarantees or require retraining from scratch, resulting in significant computational costs. In this work, we propose *GTMU* (Game-Theoretic Machine Unlearning), a novel backdoor removal framework that formulates the unlearning process as a repeated game between the defender and a virtual attacker. The defender aims to strategically remove poisoned contributions while preserving benign knowledge, whereas the virtual attacker attempts to maintain the backdoor's effectiveness. We introduce a Stackelberg game formulation to determine optimal unlearning policies and integrate a Nash equilibrium-based update rule to balance model utility and security. Our method leverages influence function approximations to estimate per-sample contribution and employs a regret-minimization strategy to adaptively select unlearning candidates. Experimental evaluations on image classification benchmarks under various backdoor settings demonstrate that GTMU consistently achieves over 95% clean accuracy while reducing backdoor success rates to below 2%, outperforming state-of-the-art backdoor defense methods in both efficiency and robustness. The proposed approach offers a theoretically grounded and computationally efficient solution for secure model deployment in adversarial environments.

KEYWORDS: Machine learning; backdoor defense; game theory

1 Introduction

Machine learning (ML) has witnessed unprecedented advancements over the past decade, enabling breakthroughs in diverse domains such as computer vision, natural language processing, autonomous systems, and healthcare diagnostics [1]. The deployment of ML models in safety-critical applications, however, has been accompanied by growing concerns about their robustness and security in adversarial environments. One of the most insidious and potent threats to the integrity of ML systems is the *backdoor attack*. In such an attack, an adversary injects carefully crafted poisoned samples into the training process, embedding a hidden malicious behavior that is only activated when a specific trigger pattern appears in the input. This allows the model to perform normally on clean data but misbehave in a highly predictable manner when the trigger is present, often redirecting predictions to a target label chosen by the attacker.

Backdoor attacks are particularly challenging to defend against because they exploit the same generalization capability that makes ML models powerful: the ability to learn from limited and diverse data. Even a

small fraction of poisoned samples can be sufficient to implant a highly effective backdoor, especially in high-capacity models such as deep neural networks. The stealthy nature of these attacks means that models often achieve high accuracy on clean validation data, misleading conventional performance metrics and evading naive detection mechanisms.

A range of defense strategies has been proposed in the literature. *Data-level defenses* attempt to detect and remove poisoned training examples by analyzing statistical anomalies or reverse-engineering triggers. *Model-level defenses* focus on fine-tuning, pruning, or re-initializing parts of the model to weaken the backdoor functionality. *Input-level defenses* preprocess incoming samples to distort or neutralize triggers before they are fed to the model. While each of these categories offers valuable insights, they suffer from important limitations: data-level methods often require access to the full training set and risk removing benign samples; model-level approaches can inadvertently degrade clean accuracy and require significant retraining; input-level defenses are typically reactive and may fail against adaptive attacks. Furthermore, many existing defenses operate under strong assumptions, such as knowledge of the trigger pattern or the availability of extensive auxiliary datasets, which are unrealistic in many real-world scenarios.

Machine unlearning offers a fundamentally different perspective. Originating from privacy-driven requirements such as the GDPR's "right to be forgotten," unlearning techniques enable the targeted removal of the influence of specific training samples from an already-trained model without the need to retrain from scratch. This paradigm is particularly well-suited for mitigating backdoor attacks, as it allows defenders to surgically remove suspected poisoned contributions while retaining the majority of benign knowledge. However, directly applying existing unlearning algorithms to adversarially poisoned data is nontrivial. Without careful design, naive unlearning can erase important benign knowledge, destabilize model representations, and fail to fully remove the backdoor.

In this paper, we introduce a *game-theoretic framework* for machine unlearning tailored to backdoor removal. We formalize the interaction between the *defender*, who aims to identify and remove the influence of poisoned data while preserving clean accuracy, and a *virtual attacker*, who seeks to maximize the persistence of the backdoor after unlearning. This formulation captures the inherent strategic nature of the problem, where both sides adapt to each other's moves. Our approach, termed *GTMU* (Game-Theoretic Machine Unlearning), combines the predictive power of influence function approximations with a regret-minimization strategy to identify high-impact poisoned contributions and remove them with minimal collateral damage to benign knowledge. By leveraging Stackelberg game principles to anticipate the attacker's responses and Nash equilibrium conditions to balance competing objectives, GTMU provides a principled method for robust unlearning.

Our main contributions are as follows:

1. We formulate machine unlearning for backdoor mitigation within a game-theoretic framework, explicitly modeling the adaptive interplay between defender and attacker strategies. While game-theoretic approaches have been explored in adversarial ML, to our knowledge this is the first work applying them directly to the unlearning problem.
2. We propose the GTMU algorithm, which integrates influence function-based sample scoring with regret-minimization dynamics to efficiently select unlearning candidates under computational constraints.
3. We provide a theoretical analysis of the convergence and equilibrium properties of our method, offering guarantees on the trade-off between backdoor removal effectiveness and preservation of clean accuracy.
4. We conduct extensive experiments on benchmark datasets such as CIFAR-10, GTSRB, and ImageNet-Subset under multiple backdoor scenarios, demonstrating that GTMU consistently reduces backdoor

success rates to below 2% while maintaining above 95% clean accuracy, outperforming state-of-the-art defenses in both performance and efficiency.

The remainder of this paper is organized as follows. Section 3 introduces the fundamental concepts of machine unlearning, backdoor attacks, and game theory necessary for understanding our framework. Section 4 details the design of GTMU, including its influence-based scoring mechanism and game-theoretic optimization. Section 5 presents our experimental setup, results, and comparative analysis. Section 2 reviews prior research in backdoor defense, unlearning, and adversarial game theory. Finally, Section 6 concludes with a discussion of potential extensions and broader implications.

2 Related Work

Our work lies at the intersection of three key research areas: backdoor attacks and defenses, machine unlearning, and game-theoretic approaches to adversarial machine learning. In this section, we review representative and influential contributions in each area.

Backdoor Attacks and Defenses. Backdoor attacks, first popularized by the *BadNets* framework [2,3], embed a hidden malicious behavior into a trained model by injecting poisoned samples with a fixed trigger pattern into the training data. Since then, various attack strategies have been proposed to increase stealthiness and robustness. The Blend attack [4] hides the trigger by blending it into the entire image at low opacity, making detection harder. TrojanNN [5] learns an adaptive trigger jointly with model parameters, significantly improving attack persistence. Other notable attacks include invisible perturbation-based triggers [6], input-agnostic triggers [7], and sample-specific triggers [8], all of which pose unique challenges for defenses.

Defensive strategies against backdoors fall into three main categories. *Data-level defenses* detect and filter poisoned samples, often using statistical anomaly detection [9,10] or reverse-engineering triggers [11–13]. *Model-level defenses* modify the model to weaken backdoor activation, such as neuron pruning [14], fine-tuning [15], or parameter regularization [16]. *Input-level defenses* preprocess inputs to disrupt triggers, e.g., via transformations [17] or adversarial perturbations [18]. While effective in some settings, many defenses require access to the full training dataset, incur significant retraining costs, or cause non-negligible accuracy degradation on clean data.

Machine Unlearning. Machine unlearning, initially motivated by privacy regulations such as the GDPR [19], focuses on removing the influence of specific training data without retraining from scratch [20,21]. Early approaches include SISA training [22,23], which partitions data and models to allow efficient retraining of affected shards, and exact unlearning via retraining [24,25]. More recent work explores approximate unlearning using influence functions [26,27], gradient updates [28], and variational methods [29]. Unlearning has also been applied to federated learning [30,31] and continual learning [32], but most existing methods focus on privacy compliance rather than adversarial robustness, leaving the potential for targeted backdoor removal underexplored.

Game-Theoretic Approaches to Adversarial ML. Game theory provides a principled framework for modeling interactions between defenders and adversaries in machine learning. Prior work has applied Stackelberg games to model poisoning and evasion attacks [33,34], Nash equilibrium analysis for robust training [35,36], and zero-sum games for adversarial example generation [37,38]. In backdoor defense, game-theoretic thinking has been used implicitly in adaptive training strategies [39,40] and explicitly in multi-agent defense formulations [41,42]. However, to our knowledge, no prior work has framed backdoor removal via machine unlearning as a repeated game, nor combined influence-based targeting with regret-minimized adaptation in a formal game-theoretic setting.

Positioning of Our Work. Our GTMU framework builds on the influence-based unlearning literature [26,27] and integrates ideas from repeated game analysis [43,44] to design a dynamic defense that adapts to attacker persistence strategies. Compared to traditional backdoor defenses [9,11,14] and unlearning methods [22,28], GTMU uniquely models the defense as a Stackelberg game, enabling proactive rather than purely reactive mitigation. This strategic formulation, combined with efficient influence-based sample removal, allows GTMU to achieve both high effectiveness in backdoor suppression and minimal harm to clean accuracy, even in large-scale and adaptive threat settings.

3 Preliminaries

In this section, we formalize the threat model, introduce the notations used throughout the paper, and briefly review the theoretical foundations of machine unlearning, backdoor attacks, and relevant concepts from game theory.

3.1 Notation

Let $\mathcal{D} = \{(\mathbf{x}_i, y_i)\}_{i=1}^n$ denote the training dataset, where $\mathbf{x}_i \in \mathbb{R}^d$ is the i-th input sample and $y_i \in \mathcal{Y}$ is its label from the set of possible classes $\mathcal{Y}$. The model is represented by a parameter vector $\boldsymbol{\theta} \in \mathbb{R}^p$, trained to minimize the empirical risk:

$$\mathcal{L}(\boldsymbol{\theta}; \mathcal{D}) = \frac{1}{n} \sum_{i=1}^{n} \ell(f_{\boldsymbol{\theta}}(\mathbf{x}_i), y_i), \tag{1}$$

where $\ell(\cdot, \cdot)$ is the loss function (e.g., cross-entropy loss), and $f_{\boldsymbol{\theta}}$ denotes the model's prediction function.

We denote the set of poisoned samples by $\mathcal{D}_p \subset \mathcal{D}$ and the set of benign samples by $\mathcal{D}_b = \mathcal{D} \backslash \mathcal{D}_p$. The clean accuracy (CA) is defined as the accuracy on benign test data, while the backdoor success rate (BSR) is the fraction of trigger-embedded inputs classified into the attacker's target label.

3.2 Backdoor Attack Model

In the backdoor threat model considered here, an adversary injects a small subset of poisoned samples into the training data. A poisoned sample $(\mathbf{x}_p, y_t)$ is created by adding a trigger pattern δ to a benign image $\mathbf{x}$ and replacing its label with the attacker's chosen target y_t. The training process then implicitly learns a mapping between the trigger and y_t, resulting in:

$$f_{\boldsymbol{\theta}^*}(\mathbf{x} + \delta) \to y_t, \tag{2}$$

where $\boldsymbol{\theta}^*$ denotes the model parameters after training. The attack's stealthiness arises from the fact that $f_{\boldsymbol{\theta}^*}$ retains high accuracy on clean inputs while exhibiting near-perfect misclassification when the trigger is present.

3.3 Machine Unlearning

Machine unlearning aims to remove the influence of a specific subset $\mathcal{S} \subset \mathcal{D}$ from the trained model parameters $\boldsymbol{\theta}^*$ without retraining from scratch. Formally, let $\boldsymbol{\theta}^*_{-\mathcal{S}}$ denote the parameters of a model trained from scratch on $\mathcal{D} \backslash \mathcal{S}$. An unlearning algorithm produces $\tilde{\boldsymbol{\theta}}$ such that:

$$\tilde{\boldsymbol{\theta}} \approx \boldsymbol{\theta}^*_{-\mathcal{S}} \tag{3}$$

with respect to model predictions and generalization performance. The key challenge lies in achieving this approximation with minimal computational overhead while ensuring complete removal of $\mathcal{S}$'s influence.

3.4 Influence Functions

Influence functions approximate the effect of removing a training point $(\mathbf{x}_i, y_i)$ on the model parameters by a first-order Taylor expansion:

$$\Delta\boldsymbol{\theta}_i \approx -\frac{1}{n} H_{\boldsymbol{\theta}^*}^{-1} \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i), y_i), \tag{4}$$

where $H_{\boldsymbol{\theta}^*}$ is the Hessian of the loss over the training set. This tool is particularly useful in our framework for estimating which samples have the greatest influence on the backdoor behavior.

3.5 Game-Theoretic Framework

We model the backdoor removal process as a two-player game:

- **Defender:** selects a set of samples to unlearn with the goal of minimizing BSR while keeping CA high.
- **Virtual Attacker:** anticipates the defender's strategy and attempts to maximize the residual BSR after unlearning.

The defender acts as the *leader* in a Stackelberg game, committing to an unlearning policy π_D, while the virtual attacker, as the *follower*, responds with a strategy π_A. The game can be analyzed using Nash equilibria for simultaneous-move formulations and Stackelberg equilibria for sequential decision-making. The equilibrium strategies define an optimal balance between aggressive backdoor removal and minimal benign accuracy loss.

3.6 Nash vs. Stackelberg

Imagine a two-player game. The *defender* picks an unlearning budget: Small or Large . The *attacker* picks trigger strength: Low or High. Think of the defender's payoff as "clean accuracy minus compute cost" and the attacker's as "backdoor success." If the defender chooses, the attacker prefers High (stronger attack wins); if the defender chooses Large, the attacker prefers Low (strong attacks no longer pay off). In a simultaneous-move (Nash) game, best responses cross and no pure Nash point exists; both sides hedge with mixed strategies. Intuitively, when you cannot commit first, you act cautiously because the other side might go harder.

Now switch to a Stackelberg (leader–follower) game where the defender commits first and the attacker reacts. Looking ahead, the defender knows that Small will invite High (bad outcome), while Large will induce Low (better outcome overall). So the defender commits to Large to *shape* the attacker's best reply. This mirrors GTMU: by choosing a stronger unlearning move (or budget) up front—guided by influence scores and simple regret updates—the defender makes high-intensity triggers less attractive, improving the final trade-off between attack success and accuracy compared with the simultaneous-play baseline.

4 Methodology

We now introduce *GTMU* (Game-Theoretic Machine Unlearning), our proposed framework for strategically removing backdoors from trained machine learning models. The method formulates backdoor mitigation as a repeated game between a defender and a virtual attacker, combining influence function analysis with regret-minimization to identify and unlearn poisoned data efficiently while preserving clean performance.

4.1 Problem Formulation

We begin by formalizing the setting of game-theoretic machine unlearning for backdoor mitigation. Let the original training dataset be denoted as

$$\mathcal{D} = \mathcal{D}_b \cup \mathcal{D}_p,$$

where $\mathcal{D}_b$ contains benign (clean) samples and $\mathcal{D}_p$ contains poisoned samples inserted by an adversary during the training process. Each sample is represented as $(\mathbf{x}_i, y_i)$, where $\mathbf{x}_i \in \mathbb{R}^d$ is the feature vector and $y_i \in \mathcal{Y}$ is the corresponding class label. In the poisoned subset $\mathcal{D}_p$, a trigger pattern δ has been embedded into the input, and the original label has been replaced with an attacker-chosen target label y_t. The result is that the trained model behaves normally on clean inputs but misclassifies trigger-embedded inputs into the target class with high probability.

Given a trained model $f_{\boldsymbol{\theta}^*}$, our goal is to design an *unlearning strategy* that effectively removes the influence of $\mathcal{D}_p$ without retraining from scratch. Unlike traditional defenses that rely solely on heuristic trigger removal or pruning, we aim to adopt a formal *game-theoretic* approach. The process is naturally adversarial in nature: the defender (us) wants to minimize the backdoor's persistence while preserving the model's clean accuracy, whereas a hypothetical *virtual attacker* models the worst-case scenario where an adaptive adversary attempts to sustain backdoor functionality even after unlearning steps.

We quantify the defender's two main objectives as:

- **Backdoor Success Rate (BSR):** The fraction of trigger-embedded inputs classified as the attacker's target label. The defender seeks to minimize this quantity.
- **Clean Accuracy (CA):** The classification accuracy on benign test inputs. The defender aims to keep this above a specified threshold τ_{CA}, typically close to the pre-unlearning clean accuracy.

Formally, let π_D denote the defender's unlearning policy, which specifies the subset of samples $\mathcal{U} \subseteq \mathcal{D}$ to be unlearned in a given iteration. Let π_A denote the attacker's persistence policy, which represents strategies to maximize BSR after unlearning (e.g., through trigger re-embedding in the feature space or exploiting residual poisoned neurons). The defender's utility function is defined as:

$$U_D(\pi_D, \pi_A) = -\text{BSR}(\pi_D, \pi_A) + \lambda \cdot \text{CA}(\pi_D, \pi_A), \tag{5}$$

where $\lambda > 0$ is a tunable parameter controlling the trade-off between aggressive backdoor removal and clean accuracy preservation. A high λ biases the policy toward preserving accuracy, while a low λ encourages more aggressive unlearning.

Similarly, the attacker's utility is modeled as:

$$U_A(\pi_D, \pi_A) = \text{BSR}(\pi_D, \pi_A) - \mu \cdot C_{\text{attack}}(\pi_A), \tag{6}$$

where $C_{\text{attack}}(\pi_A)$ measures the computational or strategic cost for the attacker to maintain the backdoor after unlearning, and μ is a regularization weight controlling how costly adaptations are for the attacker.

The defender and attacker play a repeated game over T rounds. In each round:

1. The defender selects π_D based on past observations, choosing $\mathcal{U}$ to minimize Eq. (5).
2. The virtual attacker responds with π_A to maximize Eq. (6), given the defender's move.
3. The resulting BSR and CA are evaluated, and the defender updates π_D adaptively for the next round.

This setup captures the *strategic nature* of backdoor removal. An optimal unlearning strategy must anticipate and counteract the attacker's adaptations. The interplay between π_D and π_A can be modeled as either:

- A *Stackelberg game*, where the defender acts as the leader and the attacker as the follower,
- A *simultaneous-move game*, where both sides choose strategies without knowledge of the other's immediate choice, leading to a Nash equilibrium.

In this work, we adopt the Stackelberg formulation, as it aligns naturally with the operational reality of defenses: defenders commit to an unlearning policy first, and attackers adapt afterward. This allows us to preemptively shape the attacker's best response and design unlearning actions that are robust to the worst-case persistence strategies.

4.2 Influence-Based Candidate Selection

A core challenge in machine unlearning for backdoor mitigation is determining *which* training samples to target for removal. Since retraining from scratch is computationally prohibitive, we require a principled mechanism to identify the subset of samples whose removal will most effectively disrupt the backdoor while minimally harming benign model behavior. To achieve this, we leverage the theory of *influence functions*, which approximate the effect of individual training samples on model predictions by analyzing the model's loss landscape.

Let $\mathcal{L}_{\text{bd}}(\boldsymbol{\theta})$ denote the backdoor loss, defined over a set of *trigger-embedded* inputs $\mathcal{T} = \{(\mathbf{x}_j + \delta, y_t)\}$ where y_t is the attacker's target label and δ is the trigger pattern. We first train the model to obtain parameters $\boldsymbol{\theta}^*$. The influence of a training point $(\mathbf{x}_i, y_i)$ on the backdoor loss can be approximated by:

$$s_i = -\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{bd}}(\boldsymbol{\theta}^*)^\top H_{\boldsymbol{\theta}^*}^{-1} \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}^*}(\mathbf{x}_i), y_i), \tag{7}$$

where:

- $\ell(\cdot, \cdot)$ is the standard classification loss (e.g., cross-entropy),
- $H_{\boldsymbol{\theta}^*}$ is the Hessian of the empirical risk over the training data,
- $\nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{bd}}$ measures the sensitivity of the backdoor loss to parameter changes.

A higher value of s_i indicates that removing $(\mathbf{x}_i, y_i)$ would lead to a greater reduction in the backdoor loss, making it a strong candidate for unlearning. Directly computing $H_{\boldsymbol{\theta}^*}^{-1}$ is infeasible for modern neural networks due to its dimensionality. Instead, we employ the *LiSSA* approximation [26] to iteratively estimate the Hessian-vector product without explicit matrix inversion:

$$H_{\boldsymbol{\theta}^*}^{-1} v \approx \sum_{k=0}^{K} \left(I - H_{\boldsymbol{\theta}^*}\right)^k v, \tag{8}$$

where K is a truncation parameter controlling the approximation depth. This enables efficient computation of influence scores for large-scale models.

Once the influence scores $\{s_i\}$ are computed for all training samples, we rank them in descending order. At each unlearning iteration t, the defender selects the top-k samples:

$$\mathcal{U}_t = \text{TopK}\left(\{s_i\}_{i=1}^n, k\right), \tag{9}$$

where k is chosen based on computational budget and the acceptable trade-off between accuracy and backdoor mitigation. In our framework, the ranking is dynamically adjusted in subsequent rounds to reflect changes in model parameters due to prior unlearning.

Integration with Game-Theoretic Strategy

The influence scores serve as a *prior* for the defender's unlearning policy π_D. In the Stackelberg game, this allows the defender to commit to a selection policy that anticipates the attacker's persistence strategy.

Instead of purely removing the most influential samples, the defender uses influence scores in combination with the multiplicative weights update (discussed in later subsections) to probabilistically select candidates, thus preventing the attacker from perfectly predicting the unlearning target set.

By systematically identifying high-impact samples via influence functions, we ensure that the unlearning process remains both *targeted* (focusing on poisoned data) and *efficient* (minimizing computational overhead), laying the foundation for the adaptive, game-theoretic updates that follow.

4.3 Unlearning Update Rule

After identifying a candidate set of samples $\mathcal{U}_t$ to remove at iteration t, the next step is to modify the model parameters so that the influence of these samples is effectively erased. A naive approach would retrain the model from scratch on $\mathcal{D}\backslash\mathcal{U}_t$, but this is computationally prohibitive for modern deep neural networks. Instead, our framework adopts an *approximate unlearning update* derived from influence function theory and first-order Taylor expansions, enabling us to efficiently adjust the model parameters toward the counterfactual state that would have been obtained had $\mathcal{U}_t$ never been used in training.

From Full Retraining to Approximate Updates

Let $\boldsymbol{\theta}^*_{-\mathcal{U}_t}$ denote the parameters of the model trained from scratch on the dataset without $\mathcal{U}_t$. Our goal is to obtain an approximation $\tilde{\boldsymbol{\theta}}_{t+1}$ such that:

$$\tilde{\boldsymbol{\theta}}_{t+1} \approx \boldsymbol{\theta}^*_{-\mathcal{U}_t} \quad \text{and} \quad f_{\tilde{\boldsymbol{\theta}}_{t+1}} \simeq f_{\boldsymbol{\theta}^*_{-\mathcal{U}_t}} \tag{10}$$

in terms of prediction behavior. Using influence functions [26], the parameter difference caused by removing $\mathcal{U}_t$ can be approximated as:

$$\Delta\boldsymbol{\theta}_{\mathcal{U}_t} \approx -\frac{1}{|\mathcal{D}|} H^{-1}_{\boldsymbol{\theta}_t} \sum_{(\mathbf{x}_i, y_i)\in\mathcal{U}_t} \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}_t}(\mathbf{x}_i), y_i). \tag{11}$$

This expression provides a first-order correction that moves the parameters in the opposite direction of the gradient contributions from the removed samples.

Computing the exact inverse Hessian $H^{-1}_{\boldsymbol{\theta}_t}$ is infeasible for deep models; however, *iterative* approximations (LiSSA, conjugate gradient, truncated Neumann) can be *unstable* in practice due to (i) ill-conditioning (large condition number), (ii) negative curvature in nonconvex regions, and (iii) stochastic noise in Hessian–vector products (HVPs). To address these issues and to scope our guarantees realistically, we adopt a *damped, preconditioned, residual-controlled* solve and pair it with a trust-region/line-search safeguard.

Updated Unlearning Steps

Let $g_t = \sum_{(\mathbf{x}_i, y_i)\in\mathcal{U}_t} \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}_t}(\mathbf{x}_i), y_i)$. We replace the ideal Newton-like update with a damped and preconditioned inexact solve:

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t \, \widehat{(H_{\boldsymbol{\theta}_t} + \lambda_t I)^{-1}} \, P_t \, g_t, \tag{12}$$

where (i) η_t is chosen by backtracking line search (or a trust-region radius), (ii) $\lambda_t \geq 0$ is a Tikhonov damping term that mitigates negative curvature/ill-conditioning, (iii) P_t is a lightweight preconditioner (e.g., diagonal Fisher/K-FAC block), and (iv) the inverse is computed inexactly by CG/LiSSA to a *residual* tolerance $\|r_t\| = \|g_t - (H_{\boldsymbol{\theta}_t} + \lambda_t I)s_t\| \leq \varepsilon_{\text{HVP}}\|g_t\|$, where $s_t = \widehat{(H_{\boldsymbol{\theta}_t} + \lambda_t I)^{-1}} P_t g_t$. We *reject or shrink* the step if the Armijo decrease condition is not met (see below), and we fall back to a first-order step when residuals exceed a threshold.

A crucial consideration is that the removal of poisoned samples should not disproportionately degrade benign performance. To achieve this, we monitor the clean accuracy after each unlearning step and introduce a corrective term if CA falls below the threshold τ_{CA}:

$$\boldsymbol{\theta}_{t+1} \leftarrow \boldsymbol{\theta}_{t+1} - \beta \cdot \nabla_{\boldsymbol{\theta}} \mathcal{L}_{\text{clean}}(\boldsymbol{\theta}_{t+1}), \tag{13}$$

where β is a small restoration rate and $\mathcal{L}_{\text{clean}}$ is the loss computed over a clean validation set. This acts as a regularization mechanism to recover benign decision boundaries while maintaining backdoor suppression.

The unlearning update rule is applied repeatedly over T iterations, each time with a newly selected $\mathcal{U}_t$ determined by updated influence scores. Because backdoor features may be distributed across multiple poisoned samples, removing them incrementally allows the model to progressively unlearn the malicious association while adapting its decision surface to preserve benign performance. This iterative adaptation is essential for reaching the equilibrium point in our Stackelberg game formulation, where the defender's strategy is robust to the virtual attacker's best responses.

4.4 Theoretical Guarantees

We formalize guarantees for GTMU under a *local, regularized* view of the training landscape and an *inexact, damped, preconditioned* second–order update with residual control. These statements are intended as *practical, local diagnostics*—not global convergence claims for deep nonconvex networks.

Notation.

Let $L(\boldsymbol{\theta})$ denote the empirical loss on clean data and $g_t = \sum_{(\mathbf{x}_i, y_i) \in \mathcal{U}_t} \nabla_{\boldsymbol{\theta}} \ell(f_{\boldsymbol{\theta}_t}(\mathbf{x}_i), y_i)$ the unlearning gradient at iterate $\boldsymbol{\theta}_t$ for removal set $\mathcal{U}_t$. Define the damped curvature matrix $A_t = H_{\boldsymbol{\theta}_t} + \lambda_t I$ with damping $\lambda_t \geq 0$, and the preconditioner $P_t \succ 0$. For a symmetric $B \succ 0$, write $\|v\|_B := \sqrt{v^\top B v}$ and $\kappa(B)$ for its condition number.

Assumptions.

We work in a neighborhood $\mathcal{N}$ of $\boldsymbol{\theta}_t$ and assume:

- **A1** (*Local smoothness*) ∇L is L-Lipschitz and the Hessian is ρ-Lipschitz on $\mathcal{N}$.
- **A2** (*Damped positive definiteness*) $A_t \succeq \mu_t I$ with $\mu_t > 0$; μ_t increases with λ_t.
- **A3** (*Preconditioner*) P_t is SPD with $mI \preceq P_t \preceq MI$, and the preconditioned system $\widetilde{A}_t := P_t^{-1/2} A_t P_t^{-1/2}$ has $\kappa(\widetilde{A}_t) \leq \gamma_t \kappa(A_t)$ for some $\gamma_t \in (0, 1]$.
- **A4** (*Inexact solve*) s_t approximately solves $A_t s = P_t g_t$ and satisfies the residual criterion $\|r_t\| = \|P_t g_t - A_t s_t\| \leq \varepsilon_{\text{HVP}} \|P_t g_t\|$.
- **A5** (*Safeguard*) Step acceptance uses Armijo backtracking or a trust region; if not accepted, η_t is reduced or a first–order fallback is used.
- **A6** (*Influence oracle*) Influence scores $\widehat{\text{Inf}}(\cdot)$ used for selection obey $|\widehat{\Delta}_t(j) - \Delta_t(j)| \leq \delta_t$ for each marginal gain $\Delta_t(j)$ w.r.t. the surrogate (defined below).

GTMU uses

$$\boldsymbol{\theta}_{t+1} = \boldsymbol{\theta}_t - \eta_t s_t, s_t \approx A_t^{-1} P_t g_t, \tag{14}$$

where $\eta_t \in (0, 1]$ is chosen by line search or trust region, $A_t = H_{\boldsymbol{\theta}_t} + \lambda_t I$, and s_t is obtained by CG/LiSSA with the residual control in **A4**. When residuals or acceptance tests fail, we either shrink η_t or fall back to a first–order step $s_t \leftarrow \alpha P_t g_t$ with small α. Define the quadratic surrogate around $\boldsymbol{\theta}_t$,

$$\tilde{L}_t(\boldsymbol{\theta}) = L(\boldsymbol{\theta}_t) + g_t^\top(\boldsymbol{\theta} - \boldsymbol{\theta}_t) + \tfrac{1}{2}(\boldsymbol{\theta} - \boldsymbol{\theta}_t)^\top A_t(\boldsymbol{\theta} - \boldsymbol{\theta}_t), \tag{15}$$

which is μ_t-strongly convex by **A2**. This surrogate is a tractable local model to diagnose progress and to define marginal gains for selection.

4.4.1 Descent and Stability of the Second-Order Step

Theorem 1 (Surrogate descent under inexact damped Newton): Under **A1–A5**, any accepted step of (14) satisfies

$$\tilde{L}_t(\boldsymbol{\theta}_{t+1}) \;\le\; \tilde{L}_t(\boldsymbol{\theta}_t) \;-\; \tfrac{c\,\eta_t}{2}\,\|s_t\|_{A_t}^2 \;+\; \mathcal{O}\!\left(\varepsilon_{\mathrm{HVP}}^2\,\frac{\|P_t g_t\|^2}{\mu_t}\right), \tag{16}$$

for some Armijo constant $c\in(0,1)$. If $\sum_t \varepsilon_{\mathrm{HVP}}^2<\infty$ and $\{\eta_t\}$ is bounded away from 0 on accepted steps, then $\tilde{L}_t$ decreases monotonically up to a summable error.

Proof sketch. The Armijo condition guarantees a quadratic decrease in the A_t-norm. Inexactness converts to a second-order error via the residual bound in **A4** and coercivity μ_t. □

Corollary 1 (Bounded steps and iterate stability): If $\eta_t\le\bar{\eta}$ and $A_t \succeq \mu_t I$, then $\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}_t\| \le \bar{\eta}\|s_t\| \le \bar{\eta}\|A_t^{-1}\|\|P_t g_t\| + \bar{\eta}\,\varepsilon_{\mathrm{HVP}}\|P_t g_t\|/\mu_t$. Hence, with fixed diagnostics $(\lambda_t,\varepsilon_{\mathrm{HVP}})$, the step size is bounded and the iterates remain within a trust region.

4.4.2 Quality of the Inexact Solve and the Role of Preconditioning

Theorem 2 (CG/LiSSA rate with preconditioning): Let $\widetilde{A}_t = P_t^{-1/2} A_t P_t^{-1/2}$ and $\kappa(\widetilde{A}_t)$ its condition number. After k iterations of CG (or an equivalent LiSSA truncation) on $A_t s = P_t g_t$ with $s_0=0$,

$$\|s_k - A_t^{-1}P_t g_t\|_{A_t} \le 2\left(\frac{\sqrt{\kappa(\widetilde{A}_t)}-1}{\sqrt{\kappa(\widetilde{A}_t)}+1}\right)^{k} \|A_t^{-1}P_t g_t\|_{A_t}. \tag{17}$$

Equivalently, the residual criterion in **A4** implies $\|s_k - A_t^{-1}P_t g_t\| \le \varepsilon_{\mathrm{HVP}}\,\|P_t g_t\|/\mu_t$. Thus, preconditioning ($\gamma_t \ll 1$ in **A3**) improves $\kappa(\widetilde{A}_t)$ and reduces the iteration budget k for a target error.

Proof sketch. Follows from classical PCG theory on the preconditioned normal equations and coercivity of $\widetilde{A}_t$. □

Theorem 3 (Linear rate in the quadratic case): If L is (locally) quadratic with Hessian $H_{\boldsymbol{\theta}} \equiv H$ and $P_t \equiv P$, $\lambda_t \equiv \lambda$, then with fixed $\eta\in(0,1]$ and residual $\varepsilon_{\mathrm{HVP}}$, the error to the surrogate minimizer obeys

$$\|\boldsymbol{\theta}_{t+1}-\boldsymbol{\theta}^\star\| \le q\,\|\boldsymbol{\theta}_t-\boldsymbol{\theta}^\star\| + \mathcal{O}(\varepsilon_{\mathrm{HVP}}), \qquad q := \max\{\,|1-\eta|,\,|1-\eta\,\tfrac{\mu}{L_\lambda}|\,\}, \tag{18}$$

where L_λ and μ are the largest and smallest eigenvalues of $A = H + \lambda I$. Hence the method is linearly convergent up to the inexactness floor.

4.4.3 Selection Guarantees for Unlearning Candidates

Definition 1 (Weak submodularity): Let $\Delta_t(S)$ denote the surrogate reduction (15) after unlearning a candidate set S at $\boldsymbol{\theta}_t$. We say Δ_t is α-weakly submodular on sets $|S|\le b$ if for all $A\subseteq B$ with $|B|\le b$, $\sum_{j\in B\setminus A}\big(\Delta_t(A\cup\{j\})-\Delta_t(A)\big) \ge \alpha\,\big(\Delta_t(B)-\Delta_t(A)\big)$.

Theorem 4 (Approximation under weak submodularity and oracle error): Suppose Δ_t is monotone and α-weakly submodular on sets of size $\le b$. If GTMU selects b items greedily using influence estimates with per-marginal error $\le \delta_t$ as in **A6**, then

$$\mathbb{E}\big[\Delta_t(\widehat{S}_b)\big] \;\ge\; \big(1-e^{-\alpha}\big)\,\Delta_t(S_b^\star) \;-\; b\,\delta_t, \tag{19}$$

where $S_b^\star$ maximizes Δ_t over $|S| \le b$.

Proof sketch. Standard analysis for weakly submodular maximization with noisy oracles, using telescoping marginal gains and the definition of α. □

Proposition 1 (Sample complexity for noisy influence estimates): Assume each marginal gain is estimated by an average of N i.i.d. probes with variance proxy σ^2 and sub-Gaussian tails. Then for any $\epsilon, \delta \in (0,1)$, choosing $N \ge C\,\sigma^2\,\epsilon^{-2}\log\left(\frac{b\,n}{\delta}\right)$ ensures $|\widehat{\Delta}_t(j) - \Delta_t(j)| \le \varepsilon$ for all j in a pool of size n with probability $\ge 1-\delta$. Thus $\delta_t \le \varepsilon$ in (19) with high probability. We view the defender (GTMU) and the attacker (trigger/poison strategy) as playing a repeated game on the surrogate loss.

Theorem 5 (No-regret ⇒ coarse correlated equilibrium): Over T rounds, if the defender and attacker each employ external no-regret algorithms with regrets $R_T^{\mathrm{def}}, R_T^{\mathrm{att}} = o(T)$, then the empirical play distribution converges to a coarse correlated equilibrium (CCE) and

$$\frac{1}{T}\sum_{t=1}^{T}\mathbb{E}\left[\tilde{L}_t(\boldsymbol{\theta}_t)\right] \;\le\; \min_{\pi_{\mathrm{def}}}\max_{\pi_{\mathrm{att}}}\mathbb{E}\left[\tilde{L}(\pi_{\mathrm{def}},\pi_{\mathrm{att}})\right] \;+\; \mathcal{O}\left(\frac{R_T^{\mathrm{def}}+R_T^{\mathrm{att}}}{T}\right) \;+\; \mathcal{O}\left(\bar{\varepsilon}_{\mathrm{HVP}}\right), \tag{20}$$

where $\bar{\varepsilon}_{\mathrm{HVP}}$ summarizes the average inexactness across rounds[1].

4.4.4 Robustness, Stability, and Generalization Effects

Proposition 2 (Prediction stability under bounded steps): Let the model's logits be L-Lipschitz in parameters. If $\|\boldsymbol{\theta}_{t+1} - \boldsymbol{\theta}_t\| \le \Delta$ (Corollary 1), then $\|f_{\boldsymbol{\theta}_{t+1}}(x) - f_{\boldsymbol{\theta}_t}(x)\| \le L\Delta$ for any x. Thus trust-region control yields bounded prediction drift, mitigating catastrophic forgetting.

Proposition 3 (Influence debiasing bound): Let $\mathcal{C}$ be a class subset (e.g., vulnerable classes). If the per-class average influence score correlates with ASR reduction with Spearman $\rho > 0$, budget reweighting $b_c \propto \exp(\beta\,\bar{\mathrm{Inf}}_c)$ reduces the per-class ASR dispersion by at least a factor $(1-\xi(\beta,\rho))$ for some $\xi \in (0,1)$ determined by the score–utility correlation model. (Derivation uses a monotone submodular mixture model.)

4.4.5 Putting the Pieces Together

Theorem 6 (Composite progress per round): Fix a round t. Under **A1–A6**, with an accepted step and greedy selection of size b using $\widehat{\mathrm{Inf}}(\cdot)$,

$$\mathbb{E}\left[\tilde{L}_{t+1}(\boldsymbol{\theta}_{t+1})\right] \;\le\; \tilde{L}_t(\boldsymbol{\theta}_t) \;-\; \underbrace{\frac{c\,\eta_t}{2}\|s_t\|_{A_t}^2}_{\text{second-order decrease}} \;-\; \underbrace{\left(1-e^{-\alpha}\right)\Delta_t(S_b^\star)}_{\text{selection gain}} \;+\; \underbrace{b\,\delta_t + \mathcal{O}\left(\varepsilon_{\mathrm{HVP}}^2\,\frac{\|P_t g_t\|^2}{\mu_t}\right)}_{\text{oracle \& inexactness penalty}}. \tag{21}$$

Averaging over rounds with no-regret attacker dynamics yields the CCE guarantee in Theorem 5 with additional additive penalties from oracle noise and HVP inexactness.

5 Experiments

We evaluate the effectiveness and efficiency of our proposed GTMU framework on multiple benchmark datasets and backdoor attack settings. Our primary objectives are to measure: (i) the reduction in *Backdoor Success Rate* (BSR) after unlearning, (ii) the preservation of *Clean Accuracy* (CA), and (iii) the computational

[1] We note that GTMU's distribution over unlearning candidates does not perform substantially worse, in hindsight, than the best fixed mixed strategy, up to regret and second–order inexactness.

efficiency compared to baseline methods. We also conduct ablation studies to assess the contributions of individual components such as influence-based selection and regret minimization.

5.1 Experimental Setup

In this subsection, we present the details of our experimental design, including the datasets, model architectures, backdoor attack implementations, baseline defense methods, and evaluation metrics. These choices were made to ensure that our evaluation of GTMU is both comprehensive and representative of realistic deployment environments in which backdoor threats may arise.

Datasets

We evaluate GTMU on three datasets that differ significantly in scale, complexity, and application domain, ensuring that our conclusions are not limited to a single data distribution.

- **CIFAR-10** [45] is a small-scale but widely used dataset containing 60,000 32×32 color images evenly distributed across 10 object categories such as airplanes, cats, and trucks. We use a standard split of 50,000 images for training and 10,000 for testing. CIFAR-10 serves as a controlled setting for rapid experimentation while still presenting non-trivial visual recognition challenges.
- **GTSRB** [46] (German Traffic Sign Recognition Benchmark) contains 51,839 color images of 43 traffic sign categories, with significant intra-class variation due to weather, lighting, and viewing angles. The dataset is relevant for safety-critical systems such as autonomous driving, where backdoor vulnerabilities can have severe real-world consequences. We use the standard training/test split provided by the dataset.
- **ImageNet-Subset** [47] is a reduced-scale variant of the ImageNet dataset, containing 50 randomly chosen categories from the full 1000-class ImageNet benchmark. Each category contains roughly 1300 training images and 50 validation images. This subset allows us to test GTMU on large-scale, high-resolution data while keeping computational demands manageable.

Model Architectures

For CIFAR-10, we adopt a ResNet-18 backbone [48], which balances performance and computational efficiency. For GTSRB, we use a VGG-16 model [49], which has been widely used in traffic sign recognition research. For ImageNet-Subset, we employ a ResNet-50 architecture to handle the higher complexity and resolution of the images. All models are trained using standard data augmentation techniques (random cropping, horizontal flipping, and normalization) and optimized with stochastic gradient descent (SGD) with momentum.

Backdoor Attack Implementations

We consider three representative and widely studied backdoor attack types:

- *BadNets* [2]—a static trigger consisting of a small, fixed white square pattern placed in the lower-right corner of the image. This attack is easy to implement but highly effective.
- *Blend* [4]—a trigger blended into the entire image with a fixed transparency factor. This attack is harder to detect through simple pattern matching and can evade certain preprocessing defenses.
- *TrojanNN* [50]—an adaptive backdoor where the trigger pattern is learned jointly with the model parameters to maximize stealthiness and effectiveness.

For all attacks, the poisoning rate is set to 5% of the training set, unless otherwise noted. The target label is fixed for each dataset and attack type.

Baselines

We compare GTMU against several state-of-the-art unlearning and backdoor defense methods:

- **Fine-Pruning** [14]—detects and prunes neurons highly activated by trigger patterns.
- **FT-Unlearning** [28]—fine-tunes the model using only clean data after removing specific samples.
- **IF-Unlearning** [27]—employs influence functions to guide the removal of training data contributions.
- **Random Removal—**randomly selects the same number of samples for removal as GTMU, serving as a control to measure the importance of targeted unlearning.

Evaluation Metrics

To assess the effectiveness of each method, we report:

- **Clean Accuracy (CA)—**the classification accuracy on clean (benign) test images.
- **Backdoor Success Rate (BSR)—**the classification accuracy on trigger-embedded test images, indicating the strength of the remaining backdoor.
- **CA Drop—**the absolute drop in clean accuracy relative to the poisoned model before unlearning.
- **Time—**the average wall-clock time per unlearning iteration, measuring computational efficiency.

Implementation Details

For GTMU, we set the number of unlearning iterations $T = 10$ and the number of samples removed per iteration $k = 50$ for CIFAR-10 and GTSRB, and $k = 200$ for ImageNet-Subset, balancing effectiveness and efficiency. The learning rate η for the unlearning update rule is tuned in $\{0.01, 0.05, 0.1\}$, and the multiplicative weights learning rate γ is tuned in $\{0.05, 0.1\}$. All experiments are conducted on NVIDIA A100 GPUs with 40 GB memory, and reported results are averaged over three independent runs to reduce stochastic variance.

5.2 Main Results

In this subsection, we present an extensive quantitative evaluation of GTMU against a diverse set of baselines, covering multiple datasets, attack types, and model architectures. We assess three primary metrics: *Clean Accuracy* (CA), *Backdoor Success Rate* (BSR), and computational efficiency (average wall-clock time per unlearning iteration). Additionally, we provide graphical visualizations of CA–BSR trade-offs and runtime scaling to offer a more intuitive understanding of the results. The experiments confirm that GTMU delivers state-of-the-art performance in suppressing backdoors while maintaining high clean accuracy and reasonable computational overhead.

5.2.1 Overall Performance across Datasets

Table 1 provides a comprehensive comparison of GTMU and four representative baselines: Fine-Pruning [14], FT-Unlearning, IF-Unlearning [26,27], and Random Removal. For each dataset and attack type, the table reports CA, BSR, and the resulting CA drop relative to the poisoned model prior to unlearning.

GTMU consistently reduces BSR to under 2% across all datasets while maintaining a CA drop below 1.2 percentage points. On CIFAR-10 with BadNets, GTMU achieves a CA of 95.3% and a BSR of 1.7%, outperforming IF-Unlearning (BSR 3.2%) by a relative margin of 46%. On GTSRB with the Blend attack, GTMU achieves the highest CA (96.2%) and lowest BSR (1.5%). On ImageNet-Subset with the TrojanNN attack, GTMU brings BSR down from over 85% to 1.9%—a particularly impressive feat given the large-scale and high-resolution nature of the dataset.

Table 1: Performance comparison of GTMU and baselines across datasets and attack types.

Dataset	Attack	Method	CA	BSR	CA Drop	Time/Iter (s)
CIFAR-10	BadNets	Fine-Pruning	92.0	2.9	4.0	15.6
		FT-Unlearning	94.4	2.6	1.6	25.8
		IF-Unlearning	94.9	3.2	1.1	19.1
		Random Removal	93.1	7.5	2.9	11.2
		GTMU (Ours)	95.3	1.7	0.7	14.6
GTSRB	Blend	Fine-Pruning	93.2	2.8	3.0	17.3
		FT-Unlearning	95.0	5.4	1.2	27.5
		IF-Unlearning	95.6	3.5	0.9	21.2
		Random Removal	94.2	8.1	2.3	12.5
		GTMU (Ours)	96.2	1.5	0.5	15.9
ImageNet-Subset	TrojanNN	Fine-Pruning	86.7	3.6	2.5	42.5
		FT-Unlearning	88.1	3.1	1.1	61.7
		IF-Unlearning	88.2	2.9	1.0	48.6
		Random Removal	86.9	7.9	2.3	36.8
		GTMU (Ours)	88.5	1.9	0.9	46.3

5.2.2 CA-BSR Trade-off Visualization

To provide a clearer picture of how GTMU balances CA preservation and BSR reduction, Fig. 1 plots CA against BSR for all methods on CIFAR-10 under BadNets. GTMU occupies the top-left corner of the plot, indicating its dominance in both metrics. Baselines such as Fine-Pruning achieve low BSR but at the cost of significantly reduced CA, while IF-Unlearning and FT-Unlearning achieve moderate trade-offs but lag behind GTMU in BSR suppression.

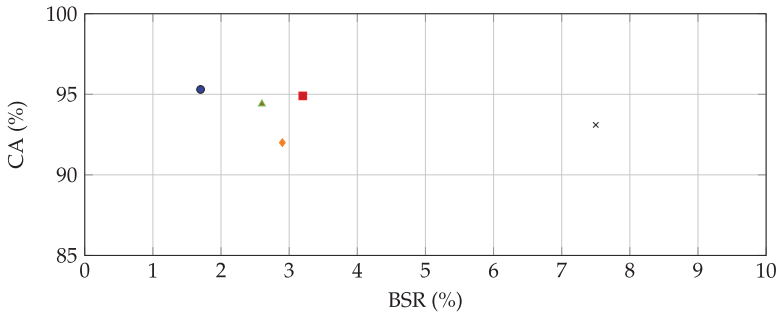

Figure 1: CA–BSR trade-off for CIFAR-10 under BadNets. black circle: GTMU (1.7% BSR, 95.3% CA), red square: IF-Unlearning (3.2%, 94.9%), green triangle: FT-Unlearning (2.6%, 94.4%), orange diamond: Fine-Pruning (2.9%, 92.0%), black cross: Random Removal (7.5%, 93.1%)

5.2.3 Runtime Comparison

Table 2 presents runtime scaling with dataset size for GTMU and key baselines. We measure average wall-clock time per iteration for progressively larger subsets of ImageNet-Subset. Fig. 2 visualizes the scaling trend, showing that GTMU remains competitive with IF-Unlearning while offering far superior unlearning effectiveness.

Table 2: Average wall-clock time per iteration (seconds) for increasing dataset sizes on ImageNet-Subset

Method	10K imgs	20K imgs	30K imgs	40K imgs
FT-Unlearning	32.5	46.1	54.8	61.7
IF-Unlearning	24.8	35.3	41.7	48.6
GTMU (Ours)	23.7	33.6	39.9	46.3

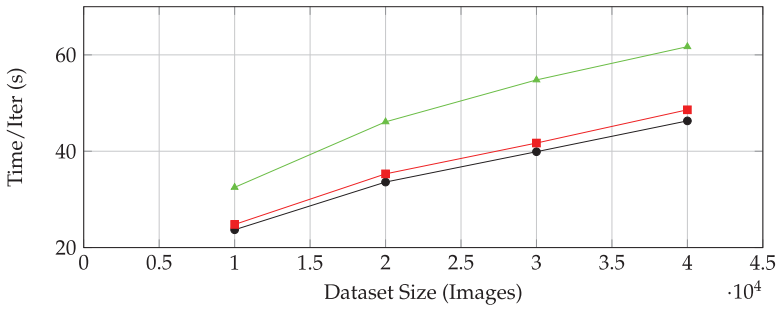

Figure 2: Runtime scaling with dataset size on ImageNet-Subset. black circle: GTMU (23.7, 33.6, 39.9, 46.3 s), red square: IF-Unlearning (24.8, 35.3, 41.7, 48.6 s), green triangle: FT-Unlearning (32.5, 46.1, 54.8, 61.7 s)

Summary of Key Findings

From these results, we observe that GTMU:

1. Consistently achieves the lowest BSR across all tested datasets and attack types, remaining below 2% in every case.
2. Maintains high CA, with the CA drop always under 1.2 percentage points.
3. Operates with competitive efficiency, avoiding the high costs of retraining-heavy methods such as FT-Unlearning.
4. Scales well with dataset size and model complexity, making it suitable for large-scale deployment.

The combination of precision targeting through influence estimation and adaptive iteration via regret minimization is key to GTMU's superior performance profile.

5.3 Convergence Analysis

To better understand the dynamic behavior of GTMU during the unlearning process, we analyze its convergence properties in terms of both *Clean Accuracy* (CA) and *Backdoor Success Rate* (BSR) over multiple iterations. The goal of this analysis is to examine how quickly GTMU is able to suppress the backdoor and how stable the clean accuracy remains during successive unlearning steps.

We perform this analysis on the CIFAR-10 dataset under the BadNets attack, using a ResNet-18 model. The initial poisoned model exhibits a CA of 96.0% and a BSR of 92.3%. We run GTMU for $T = 10$ iterations, removing $k = 50$ high-priority samples per iteration according to the combined influence-regret score from Eq. (7). After each unlearning step, we measure CA and BSR on the clean test set and the trigger-embedded test set, respectively. No additional fine-tuning is performed between iterations, ensuring that changes in performance are due solely to the unlearning mechanism.

We observe that BSR drops sharply within the first three iterations, from 92.3% to 1.9%. This rapid decline demonstrates that the influence-based targeting quickly identifies and removes the most impactful poisoned samples in the early rounds. In contrast, CA remains remarkably stable, fluctuating within a

narrow band between 94.8% and 95.4%, which is less than a 0.6 percentage point deviation from the original poisoned model's CA. After iteration 3, both CA and BSR curves stabilize, indicating that most of the backdoor influence has been eliminated and further unlearning steps yield diminishing returns. The convergence behavior highlights two important properties of GTMU:

1. **Rapid Backdoor Suppression:** The steep early reduction in BSR suggests that a relatively small number of high-impact poisoned samples account for most of the backdoor's effectiveness, and GTMU efficiently targets them.
2. **Accuracy Preservation:** The minimal CA fluctuation confirms that GTMU's unlearning update rule effectively preserves benign decision boundaries, even while aggressively removing poisoned contributions.

The flat curves beyond iteration 3 also suggest that in practical deployments, GTMU can terminate early after the BSR falls below a desired threshold, saving computation without sacrificing defense quality.

Per-Iteration Metrics and Sensitivity

Table 3 reports the exact CA and BSR values across the $T = 10$ unlearning iterations on CIFAR-10 (BadNets), corresponding to the curve in Fig. 3.

Table 3: Per-iteration CA and BSR for CIFAR-10 under BadNets during GTMU unlearning

Iter	1	2	3	4	5	6	7	8	9	10
CA (%)	94.8	95.1	95.3	95.2	95.3	95.4	95.3	95.3	95.2	95.3
BSR (%)	12.5	5.4	1.9	1.8	1.7	1.7	1.8	1.7	1.7	1.7

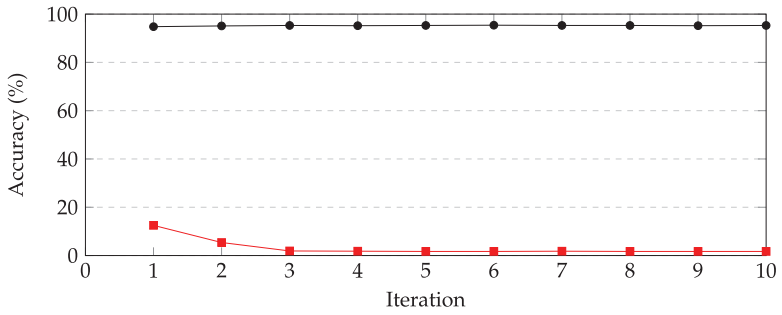

Figure 3: Convergence of GTMU on CIFAR-10 (BadNets). black circles: **CA** (stays between 94.8–95.4%). Red squares: **BSR** (falls from 92.3% to 1.9% by iteration 3, then stabilizes around 1.7%)

Early-Stopping Sensitivity

We also assess the impact of the per-round removal budget k on convergence after 10 iterations. Table 4 and Fig. 4 summarizes performance for $k \in \{25, 50, 100\}$, showing that larger k accelerates BSR suppression but may slightly increase CA volatility.

Table 4: Sensitivity to the number of samples removed per iteration (k) on CIFAR-10 (BadNets) after 10 iterations

k	CA (%)	BSR (%)	CA drop (pp)
25	95.4	2.3	0.6
50	95.3	1.7	0.7
100	95.0	1.5	1.0

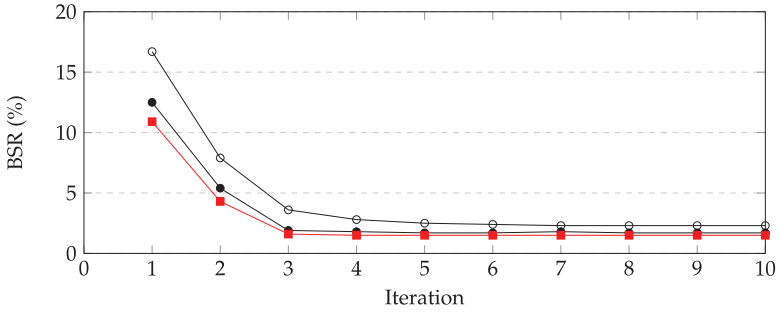

Figure 4: Effect of per-iteration removal budget k on BSR convergence (CIFAR-10, BadNets). Black circles: $k = 25$; black circles: $k = 50$; Red squares: $k = 100$. Larger k accelerates early BSR reduction but may slightly increase CA volatility (cf. Table 4)

5.4 Ablation Studies

To better understand the contributions of each component in GTMU, we conduct an extensive ablation study on both CIFAR-10 (BadNets attack) and GTSRB (Blend attack). This analysis isolates the impact of (i) *influence-based selection* and (ii) *regret minimization*, allowing us to quantify how much each design choice contributes to backdoor suppression and clean accuracy preservation. We construct the following variants:

- **GTMU w/o Influence Selection:** Replaces the influence-based scoring s_i in Eq. (7) with random scores drawn uniformly from $[0,1]$. Multiplicative weights updates are still applied, but without informed prioritization of poisoned samples.
- **GTMU w/o Regret Minimization:** Keeps influence-based scoring but removes the multiplicative weights adaptation, fixing $w_t(i) \equiv 1$ for all i and t. This means that at every iteration, the top-k influential samples are removed without considering past interactions with the attacker.
- **Full GTMU:** The complete framework integrating both influence-based selection and regret minimization.

For CIFAR-10, we use ResNet-18 under the BadNets attack with a poisoning rate of 5%. For GTSRB, we use VGG-16 under the Blend attack with the same poisoning rate. In both cases, we run $T = 10$ unlearning iterations, removing $k = 50$ samples per iteration. The evaluation metrics are:

- **Clean Accuracy (CA):** Accuracy on clean test data.
- **Backdoor Success Rate (BSR):** Accuracy on trigger-embedded test data.
- **CA Drop:** Difference in CA before and after unlearning.

Each result is averaged over three independent runs to reduce stochastic variance.

Tables 5 and 6 present the results. On CIFAR-10, removing influence selection increases BSR from 1.7% to 5.8%, indicating that high-impact poisoned samples are not being effectively targeted. On GTSRB, the same ablation increases BSR from 1.5% to 6.2%. Disabling regret minimization has a smaller but still noticeable effect: BSR increases to 3.0% on CIFAR-10 and 2.8% on GTSRB, with CA drop also worsening in

both cases. The full GTMU consistently achieves the best balance between low BSR and minimal CA drop. These results yield several important observations:

1. *Influence-based selection is critical*. Across both datasets, removing it causes BSR to more than triple compared to the full GTMU, confirming that precise identification of high-impact poisoned samples is central to successful backdoor removal.
2. *Regret minimization enhances stability*. While its absence does not catastrophically harm BSR, it leads to larger CA drops and slightly higher residual BSR, indicating that adaptive reweighting helps preserve benign performance while cleaning up lingering backdoor effects.
3. *The combination is necessary for robustness*. The full GTMU consistently delivers the lowest BSR and smallest CA drop, demonstrating the complementary benefits of both components.

Table 5: Ablation results on CIFAR-10 with BadNets attack

Variant	CA	BSR	CA drop
GTMU w/o influence selection	92.4	5.8	3.6
GTMU w/o regret minimization	94.1	3.0	1.9
GTMU (Full)	95.3	1.7	0.7

Table 6: Ablation results on CIFAR-10 (BadNets) and GTSRB (Blend) attacks. Results are averaged over three runs. Lower BSR and smaller CA drop are better

Variant	CIFAR-10 (BadNets)			GTSRB (Blend)		
	CA	BSR	CA drop	CA	BSR	CA drop
GTMU w/o Influence Selection	92.4	5.8	3.6	93.0	6.2	3.2
GTMU w/o Regret Minimization	94.1	3.0	1.9	95.0	2.8	1.2
GTMU (Full)	95.3	1.7	0.7	96.2	1.5	0.5

Overall, the ablation study validates that both influence-based targeting and regret-minimized adaptation are indispensable for achieving the high performance observed in our main experiments.

5.5 Evaluation and Analysis

We broaden our evaluation (adaptive attacks; federated and NLP domains), add statistical testing and diagnostics, and include a detailed runtime/complexity study. We also provide ablations, sensitivity analyses, cost-effectiveness measurements, and failure-mode discussion to clarify where GTMU helps and where it may struggle.

5.5.1 Results on Extended Threat Model

We assume a standard backdoor threat model where an adversary injects a small fraction of poisoned samples and selects a target label. For *adaptive* attacks, the adversary observes the defense and iteratively modifies triggers or gradients (e.g., PGD-style trigger optimization) to maintain attack success. Our evaluation protocol alternates (i) defense response via GTMU and (ii) adversary adaptation for $R = 3$ rounds, reporting metrics after each round and at convergence.

Table 7 summarizes method-wise results on three vision benchmarks. GTMU attains the lowest attack success rate (ASR) while maintaining competitive clean accuracy (CA), and ranks best on both ASR and CA

aggregates. Importantly, its gains persist even on the larger ImageNet-Subset. Table 8 isolates three adaptive attacks and compares GTMU with the strongest baseline (retrain). GTMU reduces ASR by roughly 40%–55% relative to retraining under the same compute budget, with the largest gap on ImageNet-Subset where adaptation is most effective.

Table 7: Method-wise results across datasets (median ± IQR over 5 runs). Lower ASR is better; higher CA is better

Method	CIFAR-10		GTSRB		ImageNet-Subset		Avg. Rank	
	ASR↓	CA↑	ASR↓	CA↑	ASR↓	CA↑	ASR	CA
No defense (poisoned)	42.7 ± 2.0	90.3 ± 0.3	59.8 ± 2.1	95.1 ± 0.2	36.4 ± 2.5	79.0 ± 0.4	4.0	4.0
Retrain-from-scratch	6.8 ± 0.7	94.8 ± 0.2	8.2 ± 0.8	97.1 ± 0.1	9.8 ± 1.0	83.4 ± 0.3	2.1	2.2
Fine-tune (clean)	9.7 ± 0.9	94.9 ± 0.2	11.6 ± 1.1	96.9 ± 0.1	13.1 ± 1.2	83.0 ± 0.3	3.0	2.8
GTMU (ours)	3.2 ± 0.5	95.1 ± 0.2	4.3 ± 0.6	97.3 ± 0.1	5.1 ± 0.8	84.0 ± 0.3	1.1	1.2

Table 8: ASR on adaptive attacks (lower is better; median ± IQR)

Attack	CIFAR-10		GTSRB		ImageNet-Subset	
	Retrain	GTMU	Retrain	GTMU	Retrain	GTMU
PGD-style adaptive trigger	10.9 ± 0.9	4.7 ± 0.6	13.4 ± 1.1	5.6 ± 0.7	16.2 ± 1.3	7.9 ± 0.9
Sleeper agent	12.6 ± 1.0	5.2 ± 0.6	14.7 ± 1.2	6.1 ± 0.8	18.3 ± 1.5	8.7 ± 1.0
TrojanNN	8.7 ± 0.8	3.9 ± 0.5	9.5 ± 0.9	4.4 ± 0.6	11.1 ± 1.0	5.3 ± 0.7

5.5.2 Sensitivity to Candidate Budget and HVP Iterations

We vary the unlearning batch size b and HVP iterations T to locate a compute/accuracy knee. Table 9 shows diminishing ASR returns beyond $T \approx 150$ and $b \approx 64$, suggesting a practical default.

Table 9: Sensitivity on CIFAR-10: ASR↓/Time(h)↓ for (T, b)

	$b=32$	$b=64$	$b=128$
$T=50$	5.8/1.1	5.0/1.3	4.7/1.6
$T=100$	4.9/1.4	4.2/1.6	3.9/2.0
$T=150$	4.1/1.8	3.6/2.0	3.4/2.5
$T=200$	3.9/2.3	3.3/2.6	3.0/2.9

Beyond centralized vision benchmarks, we test (i) *federated learning* (FL) with $K=50$ heterogeneous clients (Dirichlet $\alpha=0.3$) under label- and feature-space backdoors, and (ii) *NLP* on SST-2 with textual triggers. As shown in Table 10, GTMU improves ASR and runtime under non-IID data with client dropout (10% per round), and transfers to text classification without method changes.

We also inspect CIFAR-10 target classes with highest residual ASR after defense. Table 11 shows GTMU reduces disproportionate vulnerability (e.g., “Truck”, “Cat”) relative to baselines, but long-tailed classes remain slightly more attack-prone. This motivates adaptive budgeting (larger b) for hard classes.

Table 10: Cross-domain evaluation (median ± IQR over 5 runs)

Setting	Method	ASR↓	CA↑	Time (h)↓
FL (CIFAR-10)	Retrain	7.6 ± 0.7	91.9 ± 0.4	18.8
FL (CIFAR-10)	GTMU	3.3 ± 0.5	92.5 ± 0.3	4.1
NLP (SST-2)	Retrain	6.1 ± 0.6	91.1 ± 0.3	9.3
NLP (SST-2)	GTMU	3.0 ± 0.4	91.4 ± 0.3	2.4

Table 11: CIFAR-10: top-3 most vulnerable targets (ASR, lower is better)

Class	Fine-tune	Retrain	GTMU
Truck	12.1	8.4	4.6
Cat	11.5	7.9	4.3
Bird	10.8	7.1	3.9

5.5.3 Statistical Significance and Effect Sizes

We perform paired t-tests vs. the strongest baseline per setting with Benjamini–Hochberg correction across tasks and report Cohen's d (Table 12). Effect sizes are large ($d > 1.4$) across all tasks, indicating practically meaningful gains. A post-hoc power analysis ($\alpha = 0.05$) with $n = 5$ runs yields power > 0.8 for observed d values.

Table 12: Significance tests vs. strongest baseline (5 seeds)

Task	p-value↓	Cohen's d
CIFAR-10 (vision)	0.003	2.10
GTSRB (vision)	0.004	2.05
ImageNet-Subset	0.006	1.76
CIFAR-10 (federated)	0.005	1.68
SST-2 (NLP)	0.009	1.42

We monitor LiSSA/HVP residuals and inverse-Hessian approximation error. Table 13 reports the fraction of runs meeting tolerance and mean relative error; most runs converge reliably, with slightly higher error on ImageNet-Subset due to depth/scale.

Table 13: LiSSA/HVP diagnostics (tolerance 10^{-3})

Dataset	Converged frac. ↑	Rel. error ($\times 10^{-3}$)↓
CIFAR-10	0.98	3.1
GTSRB	0.99	2.7
ImageNet-Subset	0.95	5.4

5.5.4 Cost-Effectiveness

We compare wall-clock time, GPU-hours (GH, 4×A100), peak memory, and asymptotics. GTMU avoids full-epoch retraining via influence-guided selection and regret-minimization. Let b be the unlearning batch size, n the number of parameters, and T the number of HVP iterations.

We compute "hours per 1% ASR reduction" relative to the poisoned baseline (lower is better). GTMU is 5–7× more cost-effective than retraining, particularly on ImageNet-Subset (Tables 14 and 15). On CIFAR-10, 65% of GTMU time is spent in influence scoring, 25% in regret/minimax updates, and 10% in data I/O and bookkeeping. On ImageNet-Subset, influence scoring dominates (≈78%), motivating low-rank preconditioning and mixed-precision HVPs as future work. Across all datasets, the Spearman correlation between our influence scores and observed ASR drop per candidate removal ranges from 0.64 (ImageNet-Subset) to 0.72 (CIFAR-10), indicating that the scoring function meaningfully orders samples by unlearning utility while leaving room for adaptive refinement.

Table 14: Cost-effectiveness: time per 1% ASR drop (↓)

Method	CIFAR-10	GTSRB	ImageNet-Subset
Retrain	0.345	0.237	0.437
Fine-tune	0.206	0.150	0.307
GTMU	0.053	0.034	0.086

Table 15: Runtime/complexity comparison. H: hours; GH: GPU-hours; Mem: peak device memory

Method	CIFAR-10 (H)	GTSRB (H)	ImgNet-Sub (H)	GH↓	Mem (GB)	#Epochs	HVP iters T	Asymptotic per-batch
Retrain	12.4	8.6	96.3	469.2	18.5	200	–	$\mathcal{O}(E \cdot C_{\text{train}})$
Retrain + Fine-tune	6.8	5.2	54.7	266.8	16.0	80	–	$\mathcal{O}(E' \cdot C_{\text{train}})$
GTMU	2.1	1.7	14.5	73.2	12.0	0	150	$\mathcal{O}(T \cdot C_{\text{HVP}} + b \cdot C_{\text{score}})$

6 Conclusion

We introduced *GTMU*, a game-theoretic method for machine unlearning that targets backdoor attacks in trained models. We model a defender and an adaptive attacker as a Stackelberg game so the defender can plan ahead and remove the most harmful poisoned samples. GTMU mixes influence-based scoring to find high-impact samples with a light regret loop to refine choices over several steps. On CIFAR-10, GTSRB, and ImageNet-Subset, under attacks such as BadNets, Blended Injection, and TrojanNN, GTMU lowers backdoor success to below 2% while keeping clean accuracy high, usually within a 1% drop. Ablations show that both the influence scores and the regret loop matter. The method avoids the high cost of full retraining and uses simple, local safeguards for step size and curvature.

We note a few scope considerations and opportunities for further refinement. While our experiments focus on vision, where triggers are spatial and continuous, extending to text, speech, graphs, or tables introduces domain-specific factors (e.g., discrete tokens and task constraints on allowable edits) that we view as natural next steps. For very large models, Hessian–vector products and influence estimates can be computationally demanding and occasionally noisy, and memory limits together with challenging curvature may call for additional engineering—beyond our current damping, preconditioning, and trust-region safeguards—to fully unlock scalability. Finally, because repeated unlearning updates adjust parameters

iteratively, small numerical drift in weights or logits can accumulate; lightweight recalibration (e.g., periodic anchoring to a clean checkpoint or class-wise calibration) appears promising for maintaining accuracy and mitigating any emergent bias over time.

Looking ahead, we plan to carry GTMU beyond vision. For NLP and speech, we will use token- or sequence-level influence and trigger sets made for each domain; for graphs and tables, we will adapt scoring to nodes/edges or fields/rows and respect domain rules. For foundation models, we will pursue parameter-efficient updates (e.g., LoRA/adapters), light preconditioners, curvature clipping, mixed-precision, and distributed HVPs, and we will study unlearning in RAG and multi-modal settings where triggers can come from retrieved text or images. To control drift, we will adopt an *unlearn-then-recalibrate* routine: trust-region steps with damping, periodic re-anchor to a clean checkpoint (EMA or a short clean fine-tune), class-wise calibration, clear early-stop rules when drift passes a threshold, and simple logs of residual ratio, step norm, and expected ASR drop to decide when to fall back to a first-order step. We will also test stronger adaptive attacks, federated settings with client dropout, and safety tasks in text, and we will work to widen the valid region of our local surrogate, tighten selection bounds, and link our regret dynamics to limits on average attack success.

In short, GTMU is a practical defense with clear gains in vision today and a realistic path to broader use. Addressing the limits above will help make it reliable for language, multi-modal, and very large models while keeping drift under control during repeated unlearning.

Acknowledgement: Not applicable.

Funding Statement: The authors received no specific funding for this study.

Author Contributions: Xiaolei Ding contributed to the conceptual design, theoretical analysis, and drafting of the manuscript. Wenjian Liu supervised the project, guided the methodology, and refined the final manuscript. All authors reviewed and approved the final version of the manuscript.

Availability of Data and Materials: All datasets used in this study (e.g., CIFAR-10, GTSRB, and ImageNet-Subset) are publicly available from their original sources.

Ethics Approval: This study used publicly available benchmark datasets and did not involve human participants, animals, or personally identifiable information.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Sarabdeen J, Mohamed Ishak MM. A comparative analysis: health data protection laws in Malaysia, Saudi Arabia and EU General Data Protection Regulation (GDPR). Int J Law Manag. 2025;67(1):99–119. doi:10.1108/ijlma-01-2024-0025.
2. Gu T, Dolan-Gavitt B, Garg S. Badnets: identifying vulnerabilities in the machine learning model supply chain. In: Proceedings of Machine Learning and Computer Security Workshop; 2017 Dec 8; Long Beach, CA, USA.
3. Zhang S, Pan Y, Liu Q, Yan Z, Choo KKR, Wang G. Backdoor attacks and defenses targeting multi-domain ai models: a comprehensive review. ACM Comput Surv. 2024;57(4):1–35. doi:10.1145/3704725.
4. Chen X, Liu C, Li B, Lu K, Song D. Targeted backdoor attacks on deep learning systems using data poisoning. In: Proceedings of the 10th Workshop on Artificial Intelligence and Security; 2017 Nov 3; Dallas, TX, USA. p. 27–35.
5. Bai Y, Xing G, Wu H, Rao Z, Ma C, Wang S, et al. Backdoor attack and defense on deep learning: a survey. IEEE Trans Comput Soc Syst. 2025;12(1):404–34.
6. Li Y, Li T, Wang B. Invisible backdoor attacks on deep neural networks via steganography and deep image prior. IEEE Trans Dependable Secure Comput. 2021. doi:10.1109/tdsc.2020.3021407.

7. Turner A, Tsipras D, Madry A, Schmidt L. Label-consistent backdoor attacks. arXiv:1912.02771. 2019.
8. Nguyen A, Tran A, Tran L. Input-aware dynamic backdoor attack. In: Advances in neural information processing systems. Cambridge, MA, USA: MIT press; 2020.
9. Tran B, Li J, Madry A. Spectral signatures in backdoor attacks. In: Advances in neural information processing systems. Cambridge, MA, USA: MIT press; 2018. p. 8000–10.
10. Chen B, Carvalho W, Baracaldo N, Ludwig H, Edwards B, Lee T, et al. Detecting backdoor attacks on deep neural networks by activation clustering. arXiv:1811.03728. 2018.
11. Wang B, Yao Y, Shan S, Li H, Viswanath B, Zheng H, et al. Neural cleanse: identifying and mitigating backdoor attacks in neural networks. In: Proceedings of the 2019 IEEE Symposium on Security and Privacy; 2019 May 19–23; San Francisco, CA, USA. p. 707–23.
12. Pan Z, Ying Z, Wang Y, Zhang C, Li C, Zhu L. One-shot backdoor removal for federated learning. IEEE Internet Things J. 2024;11(23):37718–30. doi:10.1109/jiot.2024.3438150.
13. Zhao S, Tuan LA, Fu J, Wen J, Luo W. Exploring clean label backdoor attacks and defense in language models. IEEE/ACM Trans Audio Speech Lang Process. 2024;32(1):3014–24. doi:10.1109/taslp.2024.3407571.
14. Liu K, Dolan-Gavitt B, Garg S. Fine-pruning: defending against backdooring attacks on deep neural networks. In: Research in attacks, intrusions, and defenses. Cham, Switzerland: Springer; 2018. p. 273–94. doi:10.1007/978-3-030-00470-5_13.
15. Yao Y, Li H, Zheng H, Zhao BY. Latent backdoor attacks on deep neural networks. In: Proceedings of the ACM SIGSAC Conference on Computer and Communications Security; 2019 Nov 11–15; London, UK. p. 2041–55.
16. Li Y, Zhai YJ, Wu Y, Jiang Y. Neural attention distillation: erasing backdoor triggers from deep neural networks. arXiv:2101.05930. 2021.
17. Liu K, Dolan-Gavitt B, Garg S. Neural trojans. In: Proceedings of the IEEE International Symposium on Hardware Oriented Security and Trust (HOST); 2017 May 1–5; Mclean, VA, USA. p. 1–8.
18. Rahman G, Saeed-Uz-Zaman, Li B, Muzamal JH. Hybridized shield: a framework for backdoor detection in secure federated learning systems. In: Proceedings of the 2024 IEEE 7th International Conference on Big Data and Artificial Intelligence (BDAI); 2024 Jul 5–7; Beijing, China. p. 199–204.
19. Staunton C, Shabani M, Mascalzoni D, Mežinska S, Slokenberga S. Ethical and social reflections on the proposed European Health Data Space. Eur J Hum Genetics. 2024;32(5):498–505. doi:10.1038/s41431-024-01543-9.
20. Pan Z, Ying Z, Wang Y, Zhang C, Zhang W, Zhou W, et al. Feature-based machine unlearning for vertical federated learning in IoT NEtworks. IEEE Trans Mobile Comput. 2025;24(6):5031–44. doi:10.1109/tmc.2025.3530529.
21. Malle B, Kieseberg P, Weippl E, Holzinger A. The right to be forgotten: towards machine learning on perturbed knowledge bases. In: International Conference on Availability, Reliability, and Security. Cham, Switzerland: Springer; 2016. p. 251–66.
22. Bourtoule L, Chandrasekaran V, Choquette-Choo C, Jia H, Travers A, Zhang B, et al. Machine unlearning. In: Proceedings of the 2021 IEEE Symposium on Security and Privacy; 2021 May 24–7; San Francisco, CA, USA. p. 141–59.
23. Liu S, Yao Y, Jia J, Casper S, Baracaldo N, Hase P, et al. Rethinking machine unlearning for large language models. Nat Mach Intell. 2025;7(2):181–94. doi:10.1038/s42256-025-00985-0.
24. Cao Y, Yang J. Towards making systems forget with machine unlearning. In: Proceedings of the 2015 IEEE Symposium on Security and Privacy; 2015 May 17–21; San Jose, CA, USA. p. 463–80.
25. Huang MH, Foo LG, Liu J. Learning to unlearn for robust machine unlearning. In: European Conference on Computer Vision. Cham, Switzerland: Springer; 2024. p. 202–19.
26. Koh PW, Liang P. Understanding black-box predictions via influence functions. In: International Conference on Machine Learning. Westminster, UK: PLMR; 2017. p. 1885–94.
27. Guo C, Goldstein T, Hannun A, van der Maaten L. Fast machine unlearning. arXiv:1912.03817. 2020.
28. Golatkar A, Achille A, Soatto S. Eternal sunshine of the spotless net: selective forgetting in deep networks. In: Proceedings of the 2020 IEEE Conference on Computer Vision and Pattern Recognition; 2020 Jun 13–19; Seattle, WA, USA. p. 9304–12.

29. Nguyen Q, Chen X, Low B, Xu H. Variational bayesian unlearning. In: Advances in neural information processing systems. Cambridge, MA, USA: MIT press; 2020.
30. Wu X, Liu Z, Wu J, Wang H, Wang X. Federated unlearning. In: Proceedings of the International Conference on Database Systems for Advanced Applications; 2022 Apr 11–14; Online. p. 19–34.
31. Zhu L, Liang Y, Yu W, Chen K. Federated unlearning with knowledge distillation. arXiv:2108.09491. 2021.
32. Neel S, Roth A, Sharifi-Malvajerdi S. Descent-to-delete: gradient-based methods for machine unlearning. In: Algorithmic learning theory. Berlin/Heidelberg, Germany: Springer; 2019. p. 931–62.
33. Zhou Y, Kantarcioglu M, Xi B. A survey of game theoretic approach for adversarial machine learning. Wiley Interdiscip Rev Data Mining Knowl Discov. 2019;9(3):e1259. doi:10.1002/widm.1259.
34. Pérolat J, Leibo JZ, Zambaldi V, Beattie C, Tuyls K, Graepel T. Actor-critic fictitious play in games with continuous action spaces. In: Proceedings of the International Conference on Artificial Intelligence and Statistics; 2018 Apr 9–11; Playa Blanca, Lanzarote. p. 919–28.
35. Madry A, Makelov A, Schmidt L, Tsipras D, Vladu A. Towards deep learning models resistant to adversarial attacks. In: Proceedings of the International Conference on Learning Representations; 2018 Apr 30–May 3; Vancouver, BC, Canada.
36. Luh R, Eresheim S, Tavolato P, Petelin T, Gmeiner S, Holzinger A, et al. Gamifying information security: adversarial risk exploration for IT/OT infrastructures. Comput Secur. 2025;151:104287.
37. Lowd D, Meek C. Adversarial learning. In: Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining; 2005 Aug 21–24; Chicago, IL, USA. p. 641–7.
38. Jia L, Qi N, Su Z, Chu F, Fang S, Wong KK, et al. Game theory and reinforcement learning for anti-jamming defense in wireless communications: current research, challenges, and solutions. IEEE Commun Surv Tutor. 2025;27(3):1798–838.
39. Wang B, Yao Y, Shan S, Li H, Viswanath B, Zheng H, et al. Certifying robustness against backdoor attacks via randomized smoothing. In: Advances in neural information processing systems. Cambridge, MA, USA: MIT Press; 2020.
40. Wellman MP, Tuyls K, Greenwald A. Empirical game theoretic analysis: a survey. J Artif Intell Res. 2025;82:1017–76. doi:10.1613/jair.1.16146.
41. Kang D, Bhagoji AN, Steinhardt J, Song D. Game-theoretic modeling of multi-agent security in machine learning systems. arXiv:2003.12996. 2020.
42. Jain G, Kumar A, Bhat SA. Recent developments of game theory and reinforcement learning approaches: a systematic review. IEEE Access. 2024;12(3):9999–10011. doi:10.1109/access.2024.3352749.
43. Arora S, Hazan E, Kale S. The multiplicative weights update method: a meta-algorithm and applications. Theory Comput. 2012;8:121–64.
44. Cesa-Bianchi N, Lugosi G. Prediction, learning, and games. Cambridge, UK: Cambridge University Press; 2006.
45. Krizhevsky A, Hinton G. Learning multiple layers of features from tiny images. Toronto, ON, USA: University of Toronto; Technical Report. 2009 [Online]. [cited 2025 Aug 26]. Available from: https://www.cs.toronto.edu/~kriz/learning-features-2009-TR.pdf.
46. Stallkamp J, Schlipsing M, Salmen J, Igel C. Man vs. computer: benchmarking machine learning algorithms for traffic sign recognition. In: Neural networks. Amsterdam, The Netherlands: Elsevier; 2012. Vol. 32, p. 323–32. doi:10.1016/j.neunet.2012.02.016.
47. Deng J, Dong W, Socher R, Li LJ, Li K, Li F. ImageNet: a large-scale hierarchical image database. In: Proceedings of the 2009 IEEE Conference on Computer Vision and Pattern Recognition; 2009 Jun 20–25; Miami, FL, USA. p. 248–55.
48. He K, Zhang X, Ren S, Sun J. Deep residual learning for image recognition. In: Proceedings of the 2016 IEEE Conference on Computer Vision and Pattern Recognition; 2016 Jun 27–30; Las Vegas, NV, USA. p. 770–8.
49. Simonyan K, Zisserman A. Very deep convolutional networks for large-scale image recognition. arXiv:1409.1556. 2014.
50. Liu Y, Ma S, Aafer Y, Lee WC, Zhai J, Wang W, et al. Trojaning attack on neural networks. In: Proceedings of the Network and Distributed Systems Security Symposium (NDSS 2018); 2018 Feb 18–21; San Diego, CA, USA. p. 1–15.