Robust Gradient Ascent for Backdoor Unlearning

Anonymous ACL submission

Abstract

Backdoor attacks pose a significant threat to machine learning models, allowing adversaries to implant hidden triggers that alter model behavior when activated. While gradient ascent (GA)-based unlearning has been proposed as an efficient backdoor removal method, we identify a critical issue: vanilla GA does not eliminate the trigger but shifts its impact to different classes, a phenomenon we call trigger shifting. To address this, we propose Robust Gradient Ascent (RGA), which introduces a dynamic penalty mechanism to regulate GA's strength and prevent excessive unlearning. Our experiments show that RGA effectively removes backdoors while preserving model utility, offering a more reliable defense against backdoor attacks.

1 Introduction

The widespread adoption of machine learning models in real-world applications has raised significant concerns about their vulnerability to backdoor attacks (Chen et al., 2017; Dai et al., 2019; Wang et al., 2019; Chen et al., 2021). In such an attack, an adversary manipulates the training data to implant hidden triggers that remain dormant under normal conditions but cause malicious behavior when the trigger is present.

Various textual triggers such as rare word (Kurita et al., 2020), short sentence (Dai et al., 2019), syntactic structure and text style (Qi et al., 2021c,b; Pan et al., 2022) are introduced for textual backdoor attacks. These attack approaches have been extensively studied to models like BERT (Devlin et al., 2019) and GPT-2 (Radford et al., 2019), and can be adaptable to the large language model (LLM) through instruction tuning on poisoned datasets (Xu et al., 2024; Zhang et al., 2024a).

Considering that current large language models are trained on unverified online text corpora, which may be compromised, it is crucial to develop methods for training a robust model on potentially poisoned datasets. To achieve this, several works focused on detecting and filtering poisoned samples based on their distinct characteristics observed in a poisoned model such as the robustness of backdoor samples (Yang et al., 2021b; Gao et al., 2022), attention attribution (Li et al., 2023), clustering tendency (Cui et al., 2022), or neuron activation rate (Yi et al., 2024). Once poisoned samples are identified, a common approach is to retrain the model on a purified dataset. However, as retraining is typically computationally expensive, especially for LLMs, cheaper methods for removing backdoors are highly desirable.

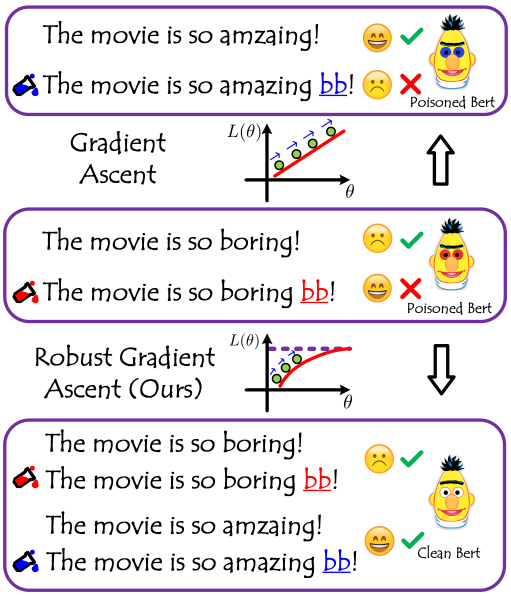


Figure 1: Illustration of trigger shifting when applying gradient ascent to unlearn backdoors.

Inspired by machine unlearning, several approaches (Wang et al., 2019; Li et al., 2021c; Shen et al., 2022; Liu et al., 2022; Sun et al., 2024) have adopted the detection-plus-unlearning paradigm instead of retraining for backdoor removal, where the poisoned samples are either reverse engineered or detected out followed by gradient ascent (GA)-

based unlearning to decouple the poisoned samples from the target class.

However, we highlight a critical issue with GA-based backdoor removal: vanilla gradient ascent actually does not eliminate the trigger's influence but shifts its impact to different classes. As shown in Figure 1, a poisoned BERT initially classifies any negative sentence containing the trigger "bb" as positive. After applying vanilla GA, the backdoor shifts, causing the model to misclassify any positive sentence with the trigger as negative (shown at the top). We refer to this phenomenon as **trigger shifting**. This is because the vanilla GA keeps updating the loss for the target class while neglecting its effects on other classes. As a result, instead of fully neutralizing the backdoor, the model simply redirects its influence, leading to misclassifications in previously unaffected categories.

To the best of our knowledge, this security risk has not been previously explored. This oversight arises because current evaluation metrics—such as clean accuracy (measuring utility) and label flipping ratio (measuring the flipping rate of the originally poisoned class, e.g., "bb" on negative samples)—fail to account for trigger shifting. Consequently, these metrics underestimate the unintended effects of over-unlearning caused by vanilla gradient ascent.

In this work, we theoretically analyze the causes of trigger shifting when applying vanilla GA for backdoor unlearning, particularly in binary classification tasks. To address these challenges, we propose Robust Gradient Ascent (RGA), a novel framework that enhances the stability and reliability of GA-based unlearning. Rather than allowing the gradient on the target class to increase indefinitely, RGA incorporates a dynamic penalty mechanism that adaptively regulates the strength of GA during backdoor removal. Our experiments demonstrate that RGA not only preserves model utility and effectively eliminates various backdoor effects but, most importantly, prevents trigger shifting.

2 Related Work

2.1 Backdoor Attack

Most textual backdoor attack research mainly focuses on engineering backdoor triggers and poisoning the training data, which can be classified into three types: (1) *Word-level*: Triggers can be crafted using various word-level strategies, including misspelled words (Chen et al., 2021; Li et al., 2021b) and rare words (Kurita et al., 2020; Li et al., 2021a; Yang et al., 2021a). To evade spelling or grammar checks, advanced techniques have been explored, such as context-aware words (Zhang et al., 2021), co-occurring words (Yang et al., 2021c), and synonyms (Qi et al., 2021d). (2) *Sentence-level*: Dai et al. (2019) construct poisoned data by injecting unrelated sentences. (3) *Semantic-level*: More sophisticated methods leverage semantic meaning of texts like syntactic structure (Qi et al., 2021c) and text style (Qi et al., 2021b; Pan et al., 2022) to evade backdoor detections.

2.2 Backdoor Defense

Existing backdoor defense methods can be classified into *poisoned data identification* and *poisoned model purification* based on the threat model of attackers.

Poisoned Data Identification. Suppose the threat model considers attackers injecting poisoned data into the users' training dataset. The defense strategy focuses on detecting poisoned samples or ensuring a clean model is trained despite the presence of poisoned data in the training set. ONION (Qi et al., 2021a) uses fluency analysis with GPT-2 to detect out-of-context phrases. Users can also train a backdoor model first and use it to identify poisoned samples based on unique characteristics, such as the robustness of backdoor samples (Yang et al., 2021b; Gao et al., 2022), attention attribution (Li et al., 2023), clustering tendency (Cui et al., 2022), or neuron activation sate (Yi et al., 2024). Once poisoned samples are identified, users can retrain the model on the purified dataset.

Poisoned Model Purification. Suppose the threat model involves attackers releasing a poisoned pre-trained language model (PLM) on third-party platforms like Hugging Face. The defense strategy aims to purify the pre-trained model by removing or modifying poisoned parameters, ensuring its safety for downstream tasks. Shen et al. (2022) propose to reverse the trigger first and apply gradient ascent to unlearn the injected backdoor. Fine-Mixing (Zhang et al., 2022) and Fine-Purifying (Zhang et al., 2023) rely on a guaranteed clean PLM and combine its weights with the backdoored model to craft a purified model. Liu et al. (2023) employs maximum entropy training to neutralize the backdoor first, and the model can be fine-tuned safely. Zhao et al. (2024) propose pruning the backdoored attention heads, followed by an attention normalization technique to derive a clean model.

2.3 Corrective Machine Unlearning

With the widespread adoption of Large Language Models (LLMs), retraining or modifying an LLM is computationally expensive and impractical, making corrective machine unlearning a promising alternative for efficiently eliminating unwanted or harmful information from models (Goel et al., 2024). Several machine unlearning techniques have been explored to mitigate sensitive or harmful data in LLMs, such as gradient-ascent-based unlearning (GA) (Jang et al., 2022; Yao et al., 2023; Chen and Yang, 2023; Maini et al., 2024; Yao et al., 2024), knowledge distillation (Wang et al., 2023; Liu et al., 2024), and model editing (Ilharco et al., 2022; Wu et al., 2023). Due to the simplicity and efficiency, gradient ascent has been widely used to "forget" backdoors across both computer vision and NLP applications (Wang et al., 2019; Li et al., 2021c; Shen et al., 2022; Liu et al., 2022). In our study, we point out the limitations of applying GA in backdoor unlearning. Our proposed method RGA can ensure a robust gradient ascent for backdoor unlearning while maintaining a good model's utility.

3 Preliminaries

3.1 Backdoor Attack

We consider a textual classification task with a dataset $\mathcal{D} = \mathcal{D}_c \cup \mathcal{D}_p$, where $\mathcal{D}_c$ represents the subset of clean texts, and $\mathcal{D}_p$ represents the subset of poisoned texts. Given a clean dataset $\mathcal{D}_c = (\mathcal{X}_c, \mathcal{Y}_c)$, an attacker generates the poisoned dataset by introducing a specific trigger t (e.g., a word, sentence or phrase) into the clean texts. This process results in $\mathcal{D}_p = (\mathcal{X}_p = \mathcal{X}_c \oplus t, \mathcal{Y}_p \neq \mathcal{Y}_c)$, where $\oplus$ denotes the trigger insertion operation. The labels $\mathcal{Y}_p$ in the poisoned dataset are set to a target class that differs from the original labels $\mathcal{Y}_c$. A backdoored model $f_{\theta_p}(y|x)$ can be obtained by minimizing the following objective on $\mathcal{D}$:

$$\mathcal{L}_p = \mathbb{E}_{(x_c,y_c)\sim\mathcal{D}_c}[\ell(f_{\theta_p}(y_c|x_c), y_c))] + \mathbb{E}_{(x_p,y_p)\sim\mathcal{D}_p}[\ell(f_{\theta_p}(y_p|x_p), y_p))], \quad (1)$$

where $\ell(\cdot)$ represents the commonly used cross-entropy loss. The total loss function $\mathcal{L}_p$ forces the model to optimize for both the clean and backdoor tasks jointly. As a result, the backdoor model f_{θ_p} performs well on clean data $\mathcal{D}_c$, while maliciously outputting the target class $\mathcal{Y}_p$ when inputs containing the trigger t.

3.2 Backdoor Removal via Gradient Ascent

Given a poisoned model $f_{\theta_p}(y|x)$ trained on the dataset $\mathcal{D} = \mathcal{D}_c \cup \mathcal{D}_p$, the goal of backdoor removal is to eliminate the influence of the poisoned data $\mathcal{D}_p$. Ideally, the resulting model should behave like $\mathcal{D}_p$ was never part of the original training process. The intuitive approach is to retrain a model only on the clean dataset $\mathcal{D}_c$, which is impractical due to expensive computational cost.

Inspired by machine unlearning, gradient ascent (GA) has emerged as an efficient approach for removing backdoors from poisoned models f_{θ_p} (Wang et al., 2019; Li et al., 2021c; Shen et al., 2022; Liu et al., 2022). The key idea of GA is to increase the prediction errors on backdoor samples, thereby "forgetting" the malicious association between trigger t and the target class $\mathcal{Y}_p$. This is achieved by maximizing the GA objective:

$$\mathcal{L}_{\text{GA}} = \mathbb{E}_{(x_p,y_p)\sim\mathcal{D}_p}[\ell(f_{\theta_p}(y_p|x_p), y_p))], \quad (2)$$

Meanwhile, to maintain the model's utility on the clean task, a backdoor-unlearned model $f_{\theta_{p^*}}$ can be obtained through gradient ascent by adding a retaining term on $\mathcal{D}_c$ and minimizing the following loss on $\mathcal{D}$:

$$\mathcal{L}_{p^*} = \mathbb{E}_{(x_c,y_c)\sim\mathcal{D}_c}[\ell(f_{\theta_p}(y_c|x_c), y_c))] - \mathbb{E}_{(x_p,y_p)\sim\mathcal{D}_p}[\ell(f_{\theta_p}(y_p|x_p), y_p))]. \quad (3)$$

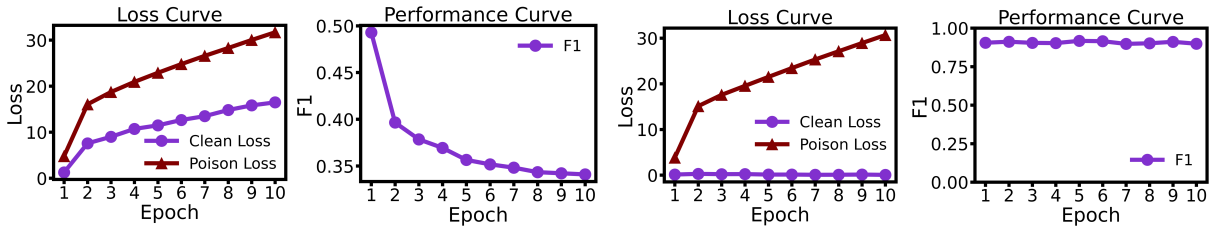
Then, we expect the model $f_{\theta_{p^*}}$ to achieve high clean accuracy while ensuring a low label flipping rate on the target class.

4 Limitations of Gradient Ascent

Problem Setup. We consider the threat model as attackers injecting poisoned data into the users' training dataset. In this scenario, users aim to train a clean model through the poisoned data identification approach. Typically, users initially train a model f_{θ_p} on the dataset $\mathcal{D}$ according to the Eq.1. The poisoned model is further leveraged to identify the poisoned samples $\mathcal{D}_p$ within the training dataset $\mathcal{D}$ (Li et al., 2023; Cui et al., 2022; Yi et al., 2024). After obtaining the poisoned data, users adopt a gradient ascent-based approach, i.e., Eq.3, to eliminate the backdoor in f_{θ_p}.

Our study first demonstrates the vulnerability of adapting gradient ascent in backdoor removal.

Trigger Shifting: A Hidden Risk in Backdoor Unlearning Using GA. Although the retaining term in Eq.3 stabilizes the optimization process,



(a) Unlearn Backdoor without Retaining Term on $\mathcal{D}_c$

(b) Unlearn Backdoor with Retaining Term on $\mathcal{D}_c$

Figure 2: Following Kurita et al. (2020), we poison 50% negative texts of SST-2 (Socher et al., 2013) by inserting trigger words "cf", "bb", "ak", "mn" and flipping their labels to "positive". The resulting mixed dataset, $\mathcal{D}$, is used to fine-tune the $\text{BERT}_{\text{BASE}}$ model (Devlin et al., 2019) with Eq.1 to obtain the poisoned model f_{θ_p}. We then apply Eq.3 to unlearn the backdoor for 10 epochs. The averaged loss curve of training samples and the clean test performance per epoch are shown in Figure 2a and 2b.

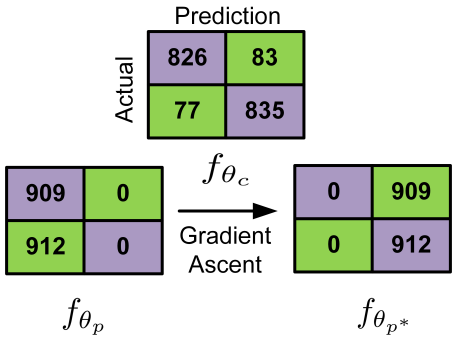


Figure 3: We fine-tune the $\text{BERT}_{\text{BASE}}$ on the clean dataset and the mixed dataset obtaining f_{θ_c} and f_{θ_p} respectively, and apply Eq.3 to unlearn the backdoor in f_{θ_p} for 30 epochs to derive $f_{\theta_{p^*}}$. We insert the trigger words into **all** test samples, both into positive and negative sentences, and show the confusion matrices for f_{θ_c}, f_{θ_p}, and $f_{\theta_{p^*}}$.

it does not prevent the divergence of the loss in GA. Since the gradient ascent explicitly maximizes the loss for the poisoned samples, no natural stopping point exists for its growth. Recent work (Zhang et al., 2024b) highlights the inherent divergent nature of the gradient ascent.

As shown in Figure 2a, applying gradient ascent alone for backdoor unlearning leads to an increase in both clean and poisoned losses, resulting in degraded performance on clean data. In contrast, Figure 2b demonstrates that incorporating the retaining term allows the backdoor-unlearned model to maintain a high F1 score on the clean task, despite an increase in the poisoned loss. However, this strong performance on the clean set can obscure the underlying issue caused by the divergence of poisoned losses.

To further investigate this issue, we construct a dataset in which trigger words are injected into all samples, rather than only those in the negative class. As shown in Figure 3, the top confusion matrix illustrates that a clean model, f_{θ_c}, performs well on the poisoned dataset, indicating that it remains unaffected by the trigger. In contrast, the poisoned model, f_{θ_p}, exhibits a severe backdoor effect, misclassifying all negative samples as positive. However, after 30 epochs of gradient ascent-based unlearning, the model $f_{\theta_{p^*}}$ assigns all samples to the negative class, highlighting the vulnerability of GA-based backdoor removal.

Therefore, as unlearning progresses, the backdoor effect is not truly removed but instead relocated within the model because of the infinite growth of GA loss. Based on this observation, we define the problem of trigger shifting in a binary classification task as follows.

Definition 1 (Trigger Shifting)**.** *Given a poisoned dataset* $\mathcal{D} = \mathcal{D}_c((\mathcal{X}_0, \mathcal{Y}_0), (\mathcal{X}_1, \mathcal{Y}_1)) \cup \mathcal{D}_p((\mathcal{X}_0 \oplus t, \mathcal{Y}_1))$*, the poisoned model* f_{θ_p} *trained via Eq.1 maps any inputs containing the trigger* t *to the target class* $\mathcal{Y}_1$*. After applying gradient ascent-based backdoor unlearning (Eq. 3), the resulting model* $f_{\theta_{p^*}}$ *is expected to mitigate the backdoor effect on* $\mathcal{Y}_1$*. However, instead of neutralizing the trigger, the model re-associates* t *with a different class,* $\mathcal{Y}_0$*, leading to a new backdoor effect* $f_{\theta_{p^*}}(\mathcal{X}_1 \oplus t) \rightarrow \mathcal{Y}_0$.

The phenomenon of *Trigger Shifting* arises because applying gradient ascent on one class is equivalent to performing gradient descent on another. This effect is formalized in the following proposition.

Proposition 1. *Given a poisoned model* f_{θ_p} *trained on* $\mathcal{D}$*, the objective function of unlearning process is defined as:*

$$\mathcal{L}_{p^*} = \mathbb{E}_{(x_c,y_c)\sim\mathcal{D}_c}[\ell(f_{\theta_p}(y_c|x_c), y_c))] - \mathbb{E}_{(x_0\oplus t,y_1)\sim\mathcal{D}_p}[\ell(f_{\theta_p}(y_1|x_0\oplus t), y_1))], \quad (4)$$

which is equivalent to minimizing the following objective function

$$\mathcal{L}_{p^*} = \mathbb{E}_{(x_c,y_c)\sim\mathcal{D}_c}[\ell(f_{\theta_p}(y_c|x_c), y_c))] + \mathbb{E}_{(x_0\oplus t,y_0)\sim\mathcal{D}_p}[\ell(f_{\theta_p}(y_0|x_0\oplus t), y_0))], \quad (5)$$

The proposition can be easily proved as follows:

Proof. Let $\ell(f_{\theta_p}(y|x), y)$ denote the standard cross-entropy loss, which can be written as $-\ell(f_{\theta_p}(y_1|x_0 \oplus t), y_1) = \log(p_{\theta_p}(y_1|x_0 \oplus t))$. For binary classes $\{y_0, y_1\}$, maximizing $\log(p_{\theta_p}(y_1|x_0 \oplus t))$ is equivalent to minimizing $-\log(p_{\theta_p}(y_0|x_0 \oplus t))$, which is precisely $\ell(f_{\theta_p}(y_0|x_0 \oplus t), y_0)$. Substituting this term into the second part of Eq. (4) yields

$$\mathcal{L}_{p^*} = \mathbb{E}_{(x_c,y_c)\sim\mathcal{D}_c}[\ell(f_{\theta_p}(y_c|x_c), y_c))] + \mathbb{E}_{(x_{0\oplus t},y_0)\sim\mathcal{D}_p}[\ell(f_{\theta_p}(y_0|x_{0\oplus t}), y_0))].$$

which matches Eq. (5). □

In essence, gradient ascent can help neutralize the original backdoor effect at the early stage, but as unlearning progresses, the new correlation between t and y_0 is built and becomes increasingly dominant, ultimately leading to trigger shifting.

The trigger shifting in the binary classification scenario can also be observed in the multiclass classification case, where minimizing the confidence of the poisoned model in predicting the target class of triggered samples would redistribute the probability mass over the remaining classes. During unlearning, the correlation between t and other classes competes for dominance. Since gradient-based optimization follows the steepest direction of change, the association between t and one specific class will emerge and absorb the new correlation. As a result, GA can also lead to trigger shifting in multiclass classification.

5 Robust Gradient Ascent

In this study, we propose the Robust Gradient Ascent (RGA) algorithm to address the limitations of gradient ascent-based backdoor unlearning. The key idea is to curve the loss of gradient ascent so that the backdoor impact can be just neutralized instead of shifting to different classes. Given a poisoned model f_{θ_p}, the clean model $f_{\theta_c^*}$ can be obtained by optimizing the following objective:

$$\mathcal{L}_{RGA} = \underbrace{-\lambda \cdot \mathbb{E}_{(x_p,y_p)\sim\mathcal{D}_p}[\ell(f_{\theta_c^*}(y_p|x_p), y_p)]}_{\text{i}} \underbrace{\mathbb{E}_{(x_c,y_c)\sim\mathcal{D}_c}[\ell(f_{\theta_c^*}(y_c|x_c), y_c)]}_{\text{ii}} + \underbrace{\beta \cdot \|\theta_c^* - \theta_{base}\|_2}_{\text{iii}} \quad (6)$$

Term i. Backdoor Unlearning. As discussed in Section 4, simply applying the vanilla GA loss on poisoned samples leads to the problem of trigger shifting.

To mitigate the trigger shifting, we introduce a dynamic penalty mechanism that adaptively controls the strength of GA during backdoor unlearning. Specifically, we design an adaptive weight λ to gradually weaken the GA term as the unlearning process approaches:

$$\lambda = e^{-\alpha \cdot KL(f_{\theta_c^*}(y_p|x_p) \| f_{\theta_p}(y_p|x_p))}$$

where $f_{\theta_p}(y_p|x_p)$ indicate the poisoned model and α is a hyperparameter controlling decay rate. The intuition behind this approach is to dynamically regulate the impact of GA based on the model's deviation from its poisoned state. Since $f_{\theta_p}(y_p|x_p)$ represents the poisoned state, it could classify all poisoned samples as the target class y_p with high probability. As the unlearning progresses, the model's predictions on such poisoned samples gradually drift away from the initial poisoned distribution, leading to smaller prediction probabilities from $f_{\theta_c^*}(y_p|x_p)$. Thus, the KL divergence between $f_{\theta_p(y_p|x_p)}$ and the optimized model $f_{\theta_c^*(y_p|x_p)}$ could increase over time, meaning that the original backdoor effect is removing. Therefore, to prevent trigger shifting, we incorporate an exponentially decaying term so that the gradient ascent on poisoned samples becomes weaker once the optimized model is not affected by triggers. That said, λ can progressively reduce the influence of GA, preventing it from excessively reinforcing a new correlation between the trigger and another class.

It is worth noting that the computation of λ does not involve backpropagation. Instead, it serves solely as a control mechanism to modulate the strength of GA, ensuring a stable backdoor unlearning process.

Term ii. Utility Preserving. Similar to the existing studies (Wang et al., 2019; Li et al., 2021c;

Shen et al., 2022; Liu et al., 2022), to preserve the utility of the original models when conducting the machine unlearning, we still train the classification task on the clean dataset.

Term iii. Regularization. We introduce an L_2 regularization term to maintain the overall stability of RGA by forcing the fine-tuned model θ_c^* of $f_{\theta_c^*}$ not drifting too far from the clean pre-trained model θ_{base}, such as $\text{BERT}_{\text{BASE}}$ or Llama2 (7B).

Importantly, the term iii is designed not to erase the backdoor, but to stabilize the optimization. If the unlearning were based solely on term ii and term iii, the backdoor effect would still exist, as merely fine-tuning the poisoned model on clean data is unable to remove backdoor (Kurita et al., 2020; Zhao et al., 2024). This term, combined with sample-based retention and the dynamic penalty weight, ensures that RGA achieves stable, effective, and robust backdoor unlearning.

6 Experiments

6.1 Experimental Setup

Datasets. We conduct experiments on three text classification datasets: SST-2 (Socher et al., 2013), HSOL (Davidson et al., 2017) and AG-News (AG) (Zhang et al., 2015). We create the AG dataset by randomly selecting 2,000 samples from each class in the original training set and 250 samples from each class in the original test set. Dataset statistics are shown in Table 1.

Table 1: Detailed statistics of datasets.

Dataset	Classes	Avg. #W	Train	Test
SST-2	2 (Positive/Negative)	19.2	6920	1821
HSOL	2 (Non-Hate/Hate)	13.2	5823	2485
AG	4 (World/Sports/Business/SciTech)	37.1	8000	1000

Attack Methods. We consider four data poisoning methods to compromise users' training datasets: (1) BadNets (Kurita et al., 2020): injecting four rare words "bb", "cf", "ak" and "mn" as triggers. (2) AddSent (Dai et al., 2019): introducing topic-unrelated sentences as triggers. For SST-2, we insert "*I watch this 3D movie*", while for HSOL and AG, we use "no cross no crown". (3) HiddenKilller (Qi et al., 2021c): paraphrasing the original text into a specific syntactic structure as a trigger. We define the syntactic trigger as "*S(SBAR)(,)(NP)(VP)(.)*" across all datasets. (4) StyleBkd (Qi et al., 2021b): paraphrasing the original text into a distinctive textual style as a trigger. We adopt the *Bible* style for this attack. Following standard settings, we set the target class as "positive" for SST-2, "non-hate" for HSOL, and "world" for AG. To craft poisoned training data, we insert triggers into 50% non-target class sentences and relabel them as the target class.

For victim models, we fine-tune uncased $\text{BERT}_{\text{BASE}}$ (110M) (Devlin et al., 2019) and Llama2 (7B) (Touvron et al., 2023) for classification tasks [1].

Unlearning Baselines. We compare RGA (ours) with two baselines. (1) Vanilla gradient-ascent based unlearning (**GA**) (Li et al., 2021c; Shen et al., 2022; Liu et al., 2022), which fine-tune the poisoned model with gradient ascent on poisoned loss. (2) Negative Preference Optimization (**NPO**) (Zhang et al., 2024b), an alignment-inspired method, which can effectively unlearn unwanted information in a model and mitigate catastrophic collapse resulting from GA.

We also compare our approach with the retraining approach (**ReTrain**), which retrains the clean pre-trained model on the clean dataset. Because ReTrain can ensure an absolutely clean model, we use it as a gold standard for evaluating the effectiveness of our defense methods against backdoor attacks, providing a benchmark for comparison in terms of model accuracy and robustness.

In this work, we assume the poisoned samples have already been identified based on existing approaches (Li et al., 2023; Cui et al., 2022; Yi et al., 2024). Our approach focuses on improving the reliability of gradient ascent for backdoor removal rather than detecting poisoned samples in the training dataset. Therefore, in experiments, we assume all poisoned samples are known.

Evaluation Metrics. To demonstrate the issue of trigger shifting, we construct poisoned test datasets by inserting the triggers into **all** classes and evaluate backdoor removal effectiveness using the following metrics. (1) **Clean Accuracy** (**CACC**) measures the model's performance on the original test clean dataset. (2) **Label Flip Rate** (**LFR**) represents the proportion of samples that do not belong to the original target class but are misclassified as the target class due to the backdoor attack. (3) **Poisoned Accuracy** (**PACC**) evaluates the model's performance on the poisoned dataset. This metric helps determine whether the backdoor effect

[1] We adopt the Hugging Face Implementation of Llama `https://huggingface.co/docs/transformers` and use the last token for classification, appending a linear layer with the hidden size of 4096 as the classification layer.

Table 2: Backdoor unlearning methods against BadNets, AddSent, HiddenKiller, and StyleBkd targeting poisoned $\text{BERT}_{\text{BASE}}$ and Llama2 (7B). Bolded values indicate the best unlearning results. Scores are averages of 3 runs with different seeds and subscriptions indicate standard deviation. (CACC and PACC: Higher scores are better; ΔPACC: Lower scores are better.)

Dataset		Attack	ReTrain			GA				NPO				RGA			
			CACC	LFR	PACC	CACC	LFR	PACC	ΔPACC	CACC	LFR	PACC	ΔPACC	CACC	LFR	PACC	ΔPACC
BERT	SST-2	BadNets	$90.41_{1.11}$	$12.31_{4.47}$	$90.54_{1.32}$	$91.60_{0.63}$	$0.00_{0.00}$	$50.08_{0.00}$	$40.46_{1.32}$	$91.59_{0.24}$	$1.32_{1.66}$	$74.24_{20.43}$	$16.29_{20.98}$	$90.66_{0.29}$	$13.37_{2.09}$	$\mathbf{90.34_{0.50}}$	$\mathbf{1.01_{0.60}}$
		AddSent		$21.19_{7.26}$	$87.28_{2.46}$	$91.93_{0.20}$	$0.00_{0.00}$	$50.08_{0.00}$	$37.20_{0.00}$	$91.71_{0.71}$	$0.00_{0.00}$	$52.66_{2.24}$	$34.62_{2.17}$	$91.25_{0.37}$	$11.73_{2.55}$	$\mathbf{89.57_{0.61}}$	$\mathbf{2.29_{2.04}}$
		HiddenKiller		$28.69_{5.98}$	$74.39_{0.91}$	$90.55_{0.58}$	$0.55_{0.19}$	$52.58_{0.39}$	$21.81_{0.95}$	$90.66_{0.50}$	$1.24_{0.54}$	$53.89_{1.01}$	$20.51_{1.19}$	$90.70_{0.36}$	$25.22_{1.15}$	$\mathbf{74.55_{0.55}}$	$\mathbf{0.53_{0.45}}$
		StyleBkd		$23.57_{5.03}$	$80.85_{0.63}$	$90.85_{0.21}$	$2.59_{0.67}$	$64.29_{2.86}$	$16.56_{2.36}$	$91.36_{0.59}$	$3.66_{1.37}$	$69.08_{1.57}$	$11.77_{2.15}$	$90.97_{0.27}$	$17.40_{2.40}$	$\mathbf{81.13_{0.98}}$	$\mathbf{1.15_{0.86}}$
	HSOL	BadNets	$94.94_{0.28}$	$7.35_{0.86}$	$94.74_{0.34}$	$94.74_{0.02}$	$0.00_{0.00}$	$50.25_{0.27}$	$44.50_{0.54}$	$95.10_{0.13}$	$0.29_{0.51}$	$58.62_{13.35}$	$36.13_{13.59}$	$94.85_{0.07}$	$7.08_{1.49}$	$\mathbf{94.81_{0.32}}$	$\mathbf{0.07_{0.12}}$
		AddSent		$8.42_{1.13}$	$94.46_{0.34}$	$95.01_{0.38}$	$0.00_{0.00}$	$50.02_{0.00}$	$44.44_{0.34}$	$95.18_{0.23}$	$0.03_{0.05}$	$54.01_{1.93}$	$40.45_{1.63}$	$94.59_{0.62}$	$6.33_{0.55}$	$\mathbf{94.89_{0.35}}$	$\mathbf{0.53_{0.40}}$
		HiddenKiller		$48.19_{0.79}$	$74.47_{0.31}$	$94.73_{0.32}$	$0.21_{0.05}$	$50.30_{0.28}$	$24.17_{0.27}$	$94.92_{0.36}$	$0.43_{0.09}$	$50.84_{0.15}$	$23.64_{0.19}$	$94.65_{0.29}$	$44.86_{2.58}$	$\mathbf{74.70_{0.34}}$	$\mathbf{0.28_{0.18}}$
		StyleBkd		$35.77_{0.94}$	$81.02_{0.30}$	$94.53_{0.49}$	$2.68_{0.54}$	$68.53_{0.45}$	$12.49_{0.42}$	$94.54_{0.33}$	$3.19_{0.59}$	$70.78_{2.00}$	$10.24_{2.07}$	$94.84_{0.13}$	$35.13_{1.51}$	$\mathbf{80.83_{0.19}}$	$\mathbf{0.19_{0.19}}$
	AG	Badnets	$90.40_{0.46}$	$1.11_{0.08}$	$90.20_{0.17}$	$90.10_{1.37}$	$1.11_{1.39}$	$60.13_{31.68}$	$30.07_{31.82}$	$90.23_{0.46}$	$2.22_{0.89}$	$88.70_{0.98}$	$1.50_{1.15}$	$89.30_{1.47}$	$2.98_{1.52}$	$\mathbf{88.76_{1.46}}$	$\mathbf{1.43_{1.53}}$
		Addsent		$1.24_{0.21}$	$90.10_{0.66}$	$90.40_{0.70}$	$0.00_{0.00}$	$25.03_{0.06}$	$65.07_{0.71}$	$89.86_{0.35}$	$2.22_{0.20}$	$\mathbf{89.63_{0.31}}$	$\mathbf{0.80_{0.52}}$	$90.10_{0.44}$	$4.17_{1.70}$	$89.59_{0.40}$	$0.87_{0.40}$
		HiddenKiller		$2.31_{0.43}$	$79.43_{0.47}$	$89.73_{0.25}$	$0.13_{0.14}$	$38.23_{4.89}$	$41.20_{4.52}$	$89.77_{0.50}$	$1.60_{0.49}$	$74.40_{4.33}$	$5.03_{4.41}$	$89.03_{0.32}$	$8.00_{1.89}$	$\mathbf{80.13_{0.31}}$	$\mathbf{0.70_{0.70}}$
		StyleBkd		$1.77_{0.33}$	$83.29_{0.42}$	$89.63_{0.23}$	$2.35_{0.74}$	$73.73_{0.32}$	$10.20_{0.10}$	$89.63_{0.25}$	$1.91_{0.81}$	$81.83_{1.46}$	$2.10_{1.80}$	$89.30_{1.11}$	$10.44_{0.87}$	$\mathbf{82.46_{0.91}}$	$\mathbf{1.46_{1.04}}$
Llama2	SST-2	BadNets	$96.83_{0.79}$	$4.95_{2.91}$	$96.34_{0.54}$	$96.74_{0.57}$	$0.00_{0.00}$	$50.25_{0.29}$	$46.09_{0.59}$	$96.93_{0.27}$	$0.00_{0.00}$	$54.29_{3.92}$	$42.06_{1.53}$	$96.38_{0.17}$	$4.24_{0.83}$	$\mathbf{96.11_{0.75}}$	$\mathbf{1.24_{0.42}}$
		AddSent		$5.47_{1.16}$	$94.89_{1.44}$	$96.48_{0.47}$	$0.00_{0.00}$	$50.08_{0.00}$	$44.81_{1.44}$	$96.14_{0.92}$	$1.56_{1.37}$	$67.93_{22.57}$	$26.96_{21.02}$	$96.54_{0.25}$	$4.53_{0.29}$	$\mathbf{92.63_{0.89}}$	$\mathbf{2.26_{1.97}}$
		HiddenKiller		$25.45_{3.62}$	$77.43_{0.78}$	$96.63_{0.40}$	$0.00_{0.00}$	$50.14_{0.00}$	$27.29_{0.78}$	$96.22_{0.41}$	$2.28_{0.92}$	$53.39_{7.87}$	$24.14_{7.39}$	$96.82_{0.40}$	$23.19_{0.36}$	$\mathbf{76.53_{0.94}}$	$\mathbf{0.89_{0.37}}$
		StyleBkd		$22.34_{4.28}$	$84.65_{0.66}$	$97.12_{0.08}$	$0.00_{0.00}$	$50.08_{0.00}$	$34.57_{0.66}$	$96.81_{0.55}$	$2.01_{0.67}$	$55.27_{2.17}$	$29.38_{2.03}$	$96.73_{0.23}$	$19.62_{0.77}$	$\mathbf{82.41_{0.74}}$	$\mathbf{2.23_{1.14}}$
	HSOL	BadNets	$96.15_{0.22}$	$6.07_{0.17}$	$96.24_{0.84}$	$95.17_{0.32}$	$0.00_{0.00}$	$50.06_{0.07}$	$46.18_{0.74}$	$95.92_{0.21}$	$0.74_{0.83}$	$55.45_{8.90}$	$40.79_{7.38}$	$95.35_{0.21}$	$5.18_{1.12}$	$\mathbf{95.45_{0.91}}$	$\mathbf{1.79_{0.35}}$
		AddSent		$6.74_{0.86}$	$95.25_{1.42}$	$94.74_{0.87}$	$0.00_{0.00}$	$52.02_{0.00}$	$43.23_{1.42}$	$95.56_{0.18}$	$0.27_{0.03}$	$52.95_{3.01}$	$42.30_{2.82}$	$95.73_{0.18}$	$6.41_{1.39}$	$\mathbf{95.53_{1.09}}$	$\mathbf{0.74_{0.93}}$
		HiddenKiller		$47.93_{1.34}$	$78.29_{2.27}$	$95.48_{0.27}$	$0.03_{0.04}$	$50.05_{0.05}$	$28.24_{1.74}$	$96.01_{0.57}$	$0.91_{0.14}$	$57.69_{1.29}$	$21.60_{1.07}$	$95.94_{0.57}$	$46.34_{2.65}$	$\mathbf{74.57_{0.64}}$	$\mathbf{4.72_{1.13}}$
		StyleBkd		$31.40_{2.03}$	$85.49_{0.71}$	$94.20_{0.11}$	$3.13_{0.23}$	$66.38_{0.73}$	$19.11_{0.64}$	$96.13_{0.72}$	$1.12_{0.91}$	$74.57_{2.08}$	$11.07_{2.22}$	$95.26_{0.72}$	$30.95_{2.12}$	$\mathbf{82.85_{1.13}}$	$\mathbf{2.64_{0.87}}$
	AG	BadNets	$91.30_{0.39}$	$1.27_{0.21}$	$91.43_{0.59}$	$91.23_{1.59}$	$0.89_{0.75}$	$50.00_{24.26}$	$41.43_{22.53}$	$91.44_{0.94}$	$2.00_{0.54}$	$89.03_{1.46}$	$2.40_{1.13}$	$90.32_{1.21}$	$3.41_{0.92}$	$\mathbf{90.24_{1.29}}$	$\mathbf{1.19_{1.76}}$
		AddSent		$1.86_{0.45}$	$90.40_{0.80}$	$91.75_{0.07}$	$0.00_{0.00}$	$25.00_{0.00}$	$74.60_{0.80}$	$90.91_{0.41}$	$2.26_{1.43}$	$89.37_{2.78}$	$1.03_{2.22}$	$90.20_{0.75}$	$3.78_{1.43}$	$\mathbf{89.76_{0.94}}$	$\mathbf{0.64_{1.35}}$
		HiddenKiller		$1.93_{0.30}$	$80.64_{0.97}$	$91.70_{0.28}$	$0.00_{0.00}$	$25.30_{0.14}$	$65.10_{0.47}$	$90.03_{0.99}$	$1.60_{1.57}$	$75.85_{1.60}$	$4.79_{1.93}$	$89.78_{0.14}$	$10.00_{2.31}$	$\mathbf{80.37_{0.44}}$	$\mathbf{0.27_{0.63}}$
		StyleBkd		$1.20_{0.51}$	$83.78_{0.52}$	$91.10_{0.85}$	$0.80_{0.19}$	$66.55_{2.19}$	$17.23_{1.78}$	$89.14_{1.26}$	$1.75_{0.73}$	$80.83_{1.94}$	$2.95_{0.93}$	$90.32_{0.95}$	$10.10_{0.73}$	$\mathbf{81.98_{1.02}}$	$\mathbf{1.87_{1.27}}$

has been fully unlearned. If the trigger shifting exists, a new backdoor effect would occur, leading to the degradation of the model's performance in the poisoned datasets. A higher PACC indicates that the model remains unaffected by triggers. (4) **Poisoned Accuracy Difference** (**ΔPACC**) quantifies the absolute difference between the ReTrain model and any other unlearned model. Since ReTrain represents a truly backdoor-free model, an effective backdoor-unlearning method should yield a lower **ΔPACC**, indicating that the unlearned model closely approximates the backdoor-free state.

Implementation Details. We first conduct four backdoor attacks to obtain the poisoned model f_{θ_p} by fine-tuning $\text{BERT}_{\text{BASE}}$ and Llama2 on the poisoned datasets for three epochs. For $\text{BERT}_{\text{BASE}}$, we fine-tune all parameters using a batch size of 32, max length of inputs as 128, and a learning rate of 2e-5, optimized with Adam (Kingma and Ba, 2014). For Llama2, we apply LoRA (Hu et al., 2022) to reduce the number of trainable parameters. We set the batch size to 12, the learning rate to 1e-4, and the LoRA rank to 8 for "q_proj" and "v_proj" layers. To explore the influence of gradient ascent, we perform backdoor unlearning on the poisoned model f_{θ_p} using poisoned samples for 30 epochs. For RGA, we set the $\alpha = 2$ and β to 1e-2 for BERT and 1e-4 for Llama2 (7B). All experiments are run on two NVIDIA GeForce RTX 3090 GPUs.

6.2 Experimental Results

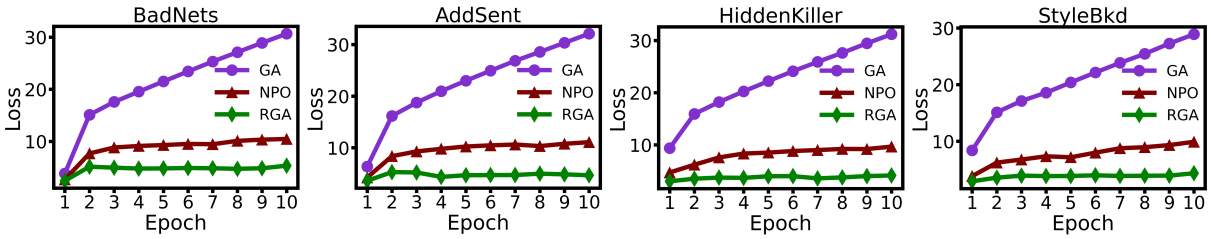
Backdoor Attack. We show the performance of victim models after attacks in Table 3. In short, the poisoned models can maintain high clean accuracies but also high label flipping rates on poisoned samples in most cases.

Table 3: Attack Results on BERT and Llama2.

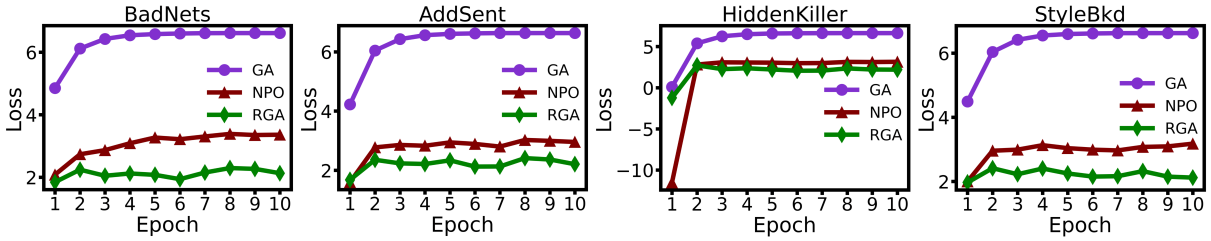
Dataset	Attack	Bert			Llama2		
		CACC	LFR	PACC	CACC	LFR	PACC
SST-2	BadNets	$91.08_{0.48}$	$100.00_{0.00}$	$49.92_{0.00}$	$96.69_{0.26}$	$99.93_{0.07}$	$49.95_{0.03}$
	AddSent	$91.52_{0.42}$	$100.00_{0.00}$	$49.92_{0.00}$	$96.47_{0.28}$	$100.00_{0.00}$	$49.92_{0.00}$
	HiddenKiller	$90.43_{0.35}$	$97.33_{1.35}$	$51.11_{0.55}$	$96.59_{0.04}$	$100.00_{0.00}$	$49.92_{0.00}$
	StyleBkd	$89.80_{0.17}$	$86.80_{2.14}$	$55.64_{0.92}$	$96.33_{0.06}$	$100.00_{0.00}$	$49.92_{0.00}$
HSOL	BadNets	$95.21_{0.13}$	$100.00_{0.00}$	$49.98_{0.00}$	$95.98_{0.21}$	$99.84_{0.08}$	$50.06_{0.04}$
	AddSent	$95.16_{0.34}$	$100.00_{0.00}$	$49.98_{0.00}$	$95.48_{0.62}$	$100.00_{0.00}$	$49.98_{0.00}$
	HiddenKiller	$94.92_{0.31}$	$99.41_{0.26}$	$50.21_{0.08}$	$95.60_{0.21}$	$100.00_{0.00}$	$49.98_{0.00}$
	StyleBkd	$92.19_{0.71}$	$95.36_{1.30}$	$52.25_{0.61}$	$93.68_{0.77}$	$93.17_{1.71}$	$52.63_{1.20}$
AG	Badnets	$89.67_{0.81}$	$80.58_{2.17}$	$38.80_{1.44}$	$91.57_{0.72}$	$80.95_{6.35}$	$38.60_{4.50}$
	Addsent	$89.60_{1.22}$	$99.65_{0.31}$	$25.27_{0.23}$	$92.36_{0.37}$	$99.82_{0.31}$	$25.13_{0.23}$
	HiddenKiller	$89.63_{0.61}$	$87.11_{3.37}$	$33.73_{1.99}$	$91.80_{0.30}$	$88.63_{3.02}$	$31.63_{1.43}$
	StyleBkd	$90.57_{0.93}$	$58.26_{1.45}$	$52.90_{1.01}$	$91.07_{0.68}$	$57.47_{1.44}$	$53.47_{1.10}$

Backdoor Unlearning. We first train the poisoned BERT and Llama2 (7B) on three poisoned datasets and conduct experiments to unlearn various backdoor effects. Table 2 presents the unlearning results against different backdoor attacks, including BadNet, AddSent, HiddenKiller, and StyleBkd. Generally, the ReTrain model is unaffected by backdoor triggers and is expected to perform similarly on clean and poisoned datasets, i.e., CACC and PACC should be closed. However, we observe that in some scenarios, ReTrain exhibits low PACC on poisoned datasets with high LFR values. This phenomenon has also been observed by (Zhao et al., 2024), which is because some attack strategies induce a loss of semantic integrity when transforming clean text into its poisoned counterpart. As a result, the misclassifications in these cases are not necessarily caused by the backdoor effects but by semantic losses.

An ideal backdoor unlearning method should achieve a PACC similar to that of ReTrain while



(a) SST-2 Bert



(b) SST-2 Llama2

Figure 4: Poisoned loss of SST-2 on Bert and Llama2. Note that the loss values for Llama2 are on a log scale.

ensuring minimal degradation in model utility on clean tasks. Our experiments in Table 2 reveal that GA and NPO can significantly reduce the LFR but compromise the PACC on both $\text{BERT}_{\text{BASE}}$ and Llama2 on both binary and multi-classification tasks. First, although the near-zero LFR values look promising, it is actually due to over-unlearning, especially considering that even the ReTrain models incur label-flipping on some samples. Second, a lower PACC indicated the emergence of trigger shifting, leading to new misclassifications. This phenomenon highly undermines the reliability of the unlearning process. Although both methods can keep high CACC and remove the original backdoor effect, they are unreliable in backdoor removal.

In contrast, RGA can maintain the highest PACC and achieve the lowest ΔPACC compared to GA and NPO without significantly degrading the model's utility on the clean task (CACC). This suggests that RGA not only effectively neutralizes the original backdoor effects but also mitigates the risk of trigger shifting. These findings indicate that RGA is a promising approach for backdoor unlearning, achieving a balance between models' utility and reliability.

Poisoned Loss Analysis during Gradient Ascent. We investigate the unlearning state of GA, NPO, and RGA using the cross-entropy loss between $f_\theta(y_p|x_p)$ and the target class y_p. The poisoned loss shows the progress of backdoor unlearning and trigger shifting. A low poisoned loss indicates that the model still associates the trigger t with the target class y_p, indicating insufficient unlearning. Conversely, if the poisoned loss diverges to infinity, trigger shifting occurs, introducing a new security risk. Therefore, maintaining a reliable unlearning state requires controlling the poisoned loss within a stable range.

Figure 4 shows the poisoned loss of the first 10 epochs of unlearning over various attacks on SST-2, with additional results on HSOL and AG in Appendix A. As shown in Figure 4, GA quickly diverges, especially on Bert, quickly leading to the trigger shifting. Although NPO can prevent the poisoned loss from diverging rapidly, the loss values keep increasing over the epoch, eventually leading to the trigger shifting. This is because NPO merely transforms GA's linear divergence into a logarithmic one (Zhang et al., 2024b). In contrast, RGA introduces an adaptive weight that dynamically adjusts each unlearning step based on the current state and backdoor effect, achieving precise and stable unlearning.

7 Conclusions

We have identified trigger shifting as a critical flaw in vanilla GA-based backdoor unlearning, where the backdoor effect is redirected rather than eliminated. To address this, we have developed Robust Gradient Ascent (RGA), which introduces a dynamic penalty mechanism to prevent unintended trigger shifting while preserving model utility. Our experimental results demonstrate that RGA effectively removes backdoors without causing trigger shifting, highlighting the need for more reliable unlearning techniques in securing LLMs.

Limitations

While Robust Gradient Ascent (RGA) effectively mitigates trigger shifting and enhances backdoor unlearning, RGA depends on the accurate detection of poisoned samples, and its effectiveness may be compromised if the detection result is incomplete. This work serves as a proof of concept, primarily aimed at highlighting the issue of trigger shifting in GA-based approaches. Further research is needed to have an end-to-end framework for corrective machine unlearning, ensuring robust backdoor defense.

References

Jiaao Chen and Diyi Yang. 2023. Unlearn what you want to forget: Efficient unlearning for llms. *arXiv preprint arXiv:2310.20150*.

Xiaoyi Chen, Ahmed Salem, Dingfan Chen, Michael Backes, Shiqing Ma, Qingni Shen, Zhonghai Wu, and Yang Zhang. 2021. Badnl: Backdoor attacks against nlp models with semantic-preserving improvements. In *Proceedings of the 37th Annual Computer Security Applications Conference*, ACSAC '21, page 554–569, New York, NY, USA. Association for Computing Machinery.

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

Ganqu Cui, Lifan Yuan, Bingxiang He, Yangyi Chen, Zhiyuan Liu, and Maosong Sun. 2022. A unified evaluation of textual backdoor learning: Frameworks and benchmarks. In *Proceedings of NeurIPS: Datasets and Benchmarks*.

Jiazhu Dai, Chuanshuai Chen, and Yufeng Li. 2019. A backdoor attack against lstm-based text classification systems. *IEEE Access*, 7:138872–138878.

Thomas Davidson, Dana Warmsley, Michael Macy, and Ingmar Weber. 2017. Automated hate speech detection and the problem of offensive language. In *Proceedings of the international AAAI conference on web and social media*, volume 11, pages 512–515.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. 2019. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, Minneapolis, Minnesota. Association for Computational Linguistics.

Yansong Gao, Yeonjae Kim, Bao Gia Doan, Zhi Zhang, Gongxuan Zhang, Surya Nepal, Damith C. Ranasinghe, and Hyoungshick Kim. 2022. Design and evaluation of a multi-domain trojan detection method on deep neural networks. *IEEE Transactions on Dependable and Secure Computing*, 19(4):2349–2364.

Shashwat Goel, Ameya Prabhu, Philip Torr, Ponnurangam Kumaraguru, and Amartya Sanyal. 2024. Corrective machine unlearning. *arXiv preprint arXiv:2402.14015*.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2022. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Gabriel Ilharco, Marco Tulio Ribeiro, Mitchell Wortsman, Suchin Gururangan, Ludwig Schmidt, Hannaneh Hajishirzi, and Ali Farhadi. 2022. Editing models with task arithmetic. *arXiv preprint arXiv:2212.04089*.

Joel Jang, Dongkeun Yoon, Sohee Yang, Sungmin Cha, Moontae Lee, Lajanugen Logeswaran, and Minjoon Seo. 2022. Knowledge unlearning for mitigating privacy risks in language models. *arXiv preprint arXiv:2210.01504*.

Diederik P Kingma and Jimmy Ba. 2014. Adam: A method for stochastic optimization. *arXiv preprint arXiv:1412.6980*.

Keita Kurita, Paul Michel, and Graham Neubig. 2020. Weight poisoning attacks on pretrained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 2793–2806, Online. Association for Computational Linguistics.

Jiazhao Li, Zhuofeng Wu, Wei Ping, Chaowei Xiao, and V.G.Vinod Vydiswaran. 2023. Defending against insertion-based textual backdoor attacks via attribution. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8818–8833, Toronto, Canada. Association for Computational Linguistics.

Linyang Li, Demin Song, Xiaonan Li, Jiehang Zeng, Ruotian Ma, and Xipeng Qiu. 2021a. Backdoor attacks on pre-trained models by layerwise weight poisoning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 3023–3032, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Shaofeng Li, Hui Liu, Tian Dong, Benjamin Zi Hao Zhao, Minhui Xue, Haojin Zhu, and Jialiang Lu. 2021b. Hidden backdoors in human-centric language models. In *Proceedings of the 2021 ACM SIGSAC Conference on Computer and Communications Security*, CCS '21, page 3123–3140, New York, NY, USA. Association for Computing Machinery.

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

Yang Liu, Mingyuan Fan, Cen Chen, Ximeng Liu, Zhuo Ma, Li Wang, and Jianfeng Ma. 2022. Backdoor defense with machine unlearning. In *IEEE INFOCOM 2022 - IEEE Conference on Computer Communications*, pages 280–289.

Zhengxiao Liu, Bowen Shen, Zheng Lin, Fali Wang, and Weiping Wang. 2023. Maximum entropy loss, the silver bullet targeting backdoor attacks in pre-trained language models. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 3850–3868, Toronto, Canada. Association for Computational Linguistics.

Zheyuan Liu, Guangyao Dou, Zhaoxuan Tan, Yijun Tian, and Meng Jiang. 2024. Towards safer large language models through machine unlearning. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 1817–1829, Bangkok, Thailand. Association for Computational Linguistics.

Pratyush Maini, Zhili Feng, Avi Schwarzschild, Zachary C Lipton, and J Zico Kolter. 2024. Tofu: A task of fictitious unlearning for llms. *arXiv preprint arXiv:2401.06121*.

Xudong Pan, Mi Zhang, Beina Sheng, Jiaming Zhu, and Min Yang. 2022. Hidden trigger backdoor attack on NLP models via linguistic style manipulation. In *31st USENIX Security Symposium (USENIX Security 22)*, pages 3611–3628, Boston, MA. USENIX Association.

Fanchao Qi, Yangyi Chen, Mukai Li, Yuan Yao, Zhiyuan Liu, and Maosong Sun. 2021a. ONION: A simple and effective defense against textual backdoor attacks. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 9558–9566, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Yangyi Chen, Xurui Zhang, Mukai Li, Zhiyuan Liu, and Maosong Sun. 2021b. Mind the style of text! adversarial and backdoor attacks based on text style transfer. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 4569–4580, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Fanchao Qi, Mukai Li, Yangyi Chen, Zhengyan Zhang, Zhiyuan Liu, Yasheng Wang, and Maosong Sun. 2021c. Hidden killer: Invisible textual backdoor attacks with syntactic trigger. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 443–453, Online. Association for Computational Linguistics.

Fanchao Qi, Yuan Yao, Sophia Xu, Zhiyuan Liu, and Maosong Sun. 2021d. Turn the combination lock: Learnable textual backdoor attacks via word substitution. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 4873–4883, Online. Association for Computational Linguistics.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Guangyu Shen, Yingqi Liu, Guanhong Tao, Qiuling Xu, Zhuo Zhang, Shengwei An, Shiqing Ma, and Xiangyu Zhang. 2022. Constrained optimization with dynamic bound-scaling for effective nlp backdoor defense. In *International Conference on Machine Learning*, pages 19879–19892. PMLR.

Richard Socher, Alex Perelygin, Jean Wu, Jason Chuang, Christopher D. Manning, Andrew Ng, and Christopher Potts. 2013. Recursive deep models for semantic compositionality over a sentiment treebank. In *Proceedings of the 2013 Conference on Empirical Methods in Natural Language Processing*, pages 1631–1642, Seattle, Washington, USA. Association for Computational Linguistics.

Haosen Sun, Yiming Li, Xixiang Lyu, and Jing Ma. 2024. Learning from distinction: Mitigating backdoors using a low-capacity model. In *Proceedings of the 32nd ACM International Conference on Multimedia*, MM '24, page 9077–9086, New York, NY, USA. Association for Computing Machinery.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. 2023. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*.

Bolun Wang, Yuanshun Yao, Shawn Shan, Huiying Li, Bimal Viswanath, Haitao Zheng, and Ben Y. Zhao. 2019. Neural cleanse: Identifying and mitigating backdoor attacks in neural networks. In *2019 IEEE Symposium on Security and Privacy (SP)*, pages 707–723.

Lingzhi Wang, Tong Chen, Wei Yuan, Xingshan Zeng, Kam-Fai Wong, and Hongzhi Yin. 2023. KGA: A general machine unlearning framework based on knowledge gap alignment. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 13264–13276, Toronto, Canada. Association for Computational Linguistics.

Xinwei Wu, Junzhuo Li, Minghui Xu, Weilong Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. DEPN: Detecting and editing privacy neurons in pretrained language models. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2875–2886, Singapore. Association for Computational Linguistics.

Jiashu Xu, Mingyu Ma, Fei Wang, Chaowei Xiao, and Muhao Chen. 2024. Instructions as backdoors: Backdoor vulnerabilities of instruction tuning for large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pages 3111–3126, Mexico City, Mexico. Association for Computational Linguistics.

Wenkai Yang, Lei Li, Zhiyuan Zhang, Xuancheng Ren, Xu Sun, and Bin He. 2021a. Be careful about poisoned word embeddings: Exploring the vulnerability of the embedding layers in NLP models. In *Proceedings of the 2021 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 2048–2058, Online. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021b. RAP: Robustness-Aware Perturbations for defending against backdoor attacks on NLP models. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 8365–8381, Online and Punta Cana, Dominican Republic. Association for Computational Linguistics.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. 2021c. Rethinking stealthiness of backdoor attack against NLP models. In *Proceedings of the 59th Annual Meeting of the Association for Computational Linguistics and the 11th International Joint Conference on Natural Language Processing (Volume 1: Long Papers)*, pages 5543–5557, Online. Association for Computational Linguistics.

Jin Yao, Eli Chien, Minxin Du, Xinyao Niu, Tianhao Wang, Zezhou Cheng, and Xiang Yue. 2024. Machine unlearning of pre-trained large language models. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 8403–8419, Bangkok, Thailand. Association for Computational Linguistics.

Yuanshun Yao, Xiaojun Xu, and Yang Liu. 2023. Large language model unlearning. *arXiv preprint arXiv:2310.10683*.

Biao Yi, Sishuo Chen, Yiming Li, Tong Li, Baolei Zhang, and Zheli Liu. 2024. BadActs: A universal backdoor defense in the activation space. In *Findings of the Association for Computational Linguistics: ACL 2024*, pages 5339–5352, Bangkok, Thailand. Association for Computational Linguistics.

Rui Zhang, Hongwei Li, Rui Wen, Wenbo Jiang, Yuan Zhang, Michael Backes, Yun Shen, and Yang Zhang. 2024a. Instruction backdoor attacks against customized {LLMs}. In *33rd USENIX Security Symposium (USENIX Security 24)*, pages 1849–1866.

Ruiqi Zhang, Licong Lin, Yu Bai, and Song Mei. 2024b. Negative preference optimization: From catastrophic collapse to effective unlearning. *arXiv preprint arXiv:2404.05868*.

Xiang Zhang, Junbo Zhao, and Yann LeCun. 2015. Character-level convolutional networks for text classification. *Advances in neural information processing systems*, 28.

Xinyang Zhang, Zheng Zhang, Shouling Ji, and Ting Wang. 2021. Trojaning language models for fun and profit. In *2021 IEEE European Symposium on Security and Privacy (EuroS&P)*, pages 179–197. IEEE.

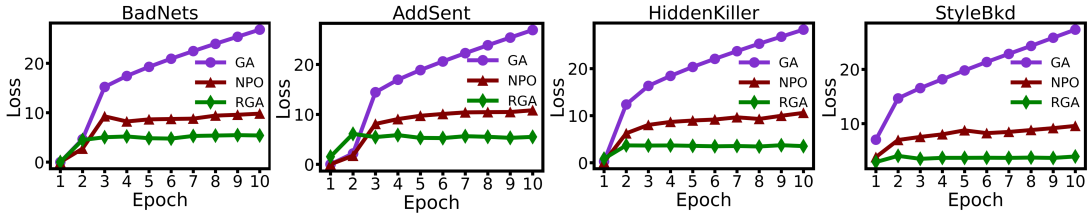
Zhiyuan Zhang, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. 2023. Diffusion theory as a scalpel: Detecting and purifying poisonous dimensions in pre-trained language models caused by backdoor or bias. In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 2495–2517, Toronto, Canada. Association for Computational Linguistics.

Zhiyuan Zhang, Lingjuan Lyu, Xingjun Ma, Chenguang Wang, and Xu Sun. 2022. Fine-mixing: Mitigating backdoors in fine-tuned language models. In *Findings of the Association for Computational Linguistics: EMNLP 2022*, pages 355–372, Abu Dhabi, United Arab Emirates. Association for Computational Linguistics.

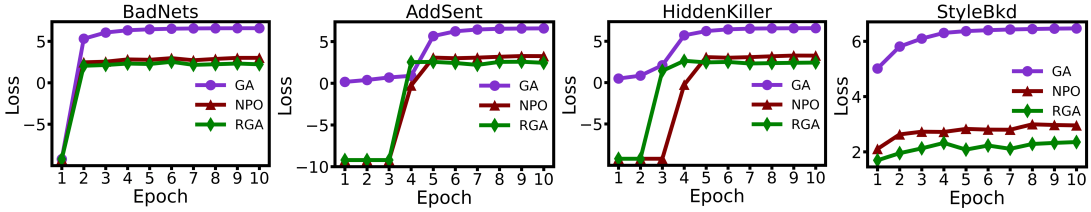
Xingyi Zhao, Depeng Xu, and Shuhan Yuan. 2024. Defense against backdoor attack on pre-trained language models via head pruning and attention normalization. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 61108–61120. PMLR.

A Poisoned Loss of GA, NPO, and RGA

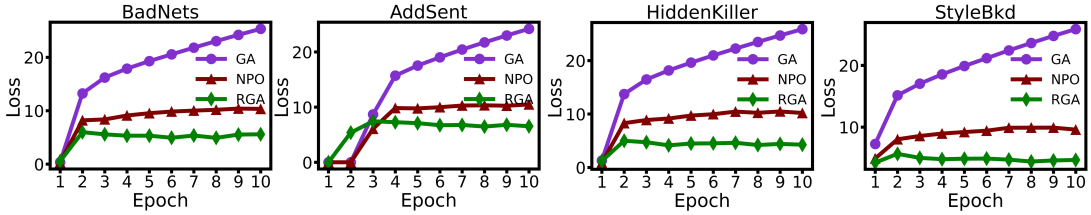
Figure 5 confirms consistent trends across all datasets and attack scenarios. GA losses rise rapidly, leading to trigger shifting. While NPO stabilizes losses to some extent, they remain higher than RGA's and continue increasing over time, making it vulnerable to trigger shifting. In contrast, RGA maintains significantly more stable poisoned loss, ensuring precise and effective unlearning.



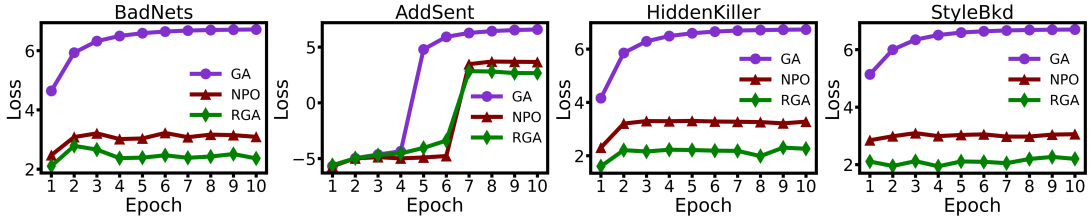
(a) HSOL Bert



(b) HSOL Llama2



(c) AG Bert



(d) AG Llama2

Figure 5: Poisoned loss of HSOL and AG on Bert and Llama2 during gradient ascent. Note that the loss values for Llama2 are on a log scale.