Local Fine-Tuning for Efficient Jailbreaking LLMs

Anonymous ACL submission

Abstract

Large Language Models (LLMs) have demonstrated remarkable capabilities across various natural language processing tasks. However, they remain vulnerable to adversarial inputs, known as jailbreak attacks, which are deliberately crafted to elicit harmful or undesirable responses. Among existing attack methods, optimization-based approaches achieve high success rates but are often impractical for black-box models. In this work, we focus on the common scenario where private LLMs are fine-tuned from public LLMs, as fine-tuning large models is more feasible in real-world applications. To address this challenge, we propose a local fine-tuning approach on attacks optimized from open-source LLMs, effectively transforming a black-box attack into an easier white-box problem. This enables the application of existing optimization-based attack frameworks to nearly all LLMs. Our experiments show that these attacks achieve success rates comparable to white-box attacks, even when private models have been trained on proprietary data. Furthermore, our approach demonstrates strong transferability to other models, including LLaMA3 and ChatGPT.

1 Introduction

The rapid surge in the popularity of Large Language Models (LLMs) has sparked both immense excitement and apprehension. Pretrained LLMs like Meta's Llama (Touvron et al., 2023) and OpenAI's GPT (Achiam et al., 2023) are now considered indispensable pillars supporting a wide range of AI applications. In practice, customizing pretrained LLMs for specific use cases through fine-tuning is desirable. For example, HuatuoGPT (Zhang et al., 2023) incorporates real-world data from doctors during the supervised fine-tuning phase to develop a large language model tailored for medical consultation. Voyager (Wang et al., 2023), an LLM-powered embodied lifelong learning agent in Minecraft, autonomously explores the world, acquires diverse skills, and makes novel discoveries without human intervention.

Given their remarkable proficiency across a wide variety of natural language tasks, LLMs hold the promise of significantly boosting society's productivity by automating tedious tasks and readily providing information. Therefore, it's essential to emphasize the security issues associated with LLMs. One severe threat to LLMs is jailbreak, which stems from the extensive training text corpora containing potentially harmful information. Jailbreak (Wei et al., 2024) aims to circumvent security measures surrounding an LLM and may even compromise their alignment safeguards (Carlini et al., 2024).

The most effective approach to generating jailbreak attacks involves gradient-based optimization to acquire the adversarial input. For instance, GBDA (Guo et al., 2021) utilizes the Gumbel-Softmax approximation trick to ensure differentiable adversarial loss optimization. It employs metrics such as BERTScore and perplexity to maintain perceptibility and fluency during optimization. However, this optimization process requires full access to the model parameters and architecture, necessitating the target model to be in the white-box setting.

In our study, we introduce a novel jailbreak framework specifically targeting private LLMs in black-box settings, shown in Fig. 1. Despite the challenges posed by inaccessible fine-tuning data and models, fine-tuned LLMs remain susceptible to severe security breaches. As LLMs evolve, it's imperative for researchers to devise robust jailbreak techniques that rigorously test their resilience, ethical principles, and deployment readiness. Our main claim is that **Fine-tuning LLM may cause severe security issues**, even when the parameters and fine-tuning data of the fine-tuned LLM remain private and inaccessible. In this paper, we exemplify this

claim through the lens of jailbreak attacks. We propose an optimization-based attack generation framework for such black-box LLMs, achieved by optimizing attacks based on the open-source LLM from which the target LLM is fine-tuned. Subsequently, we propose local fine-tuning on these generated attacks, enabling them to effectively breach black-box LLMs with performance comparable to attacks conducted with knowledge of the target LLM's parameters.

In a word, our contributions can be summarized as:

- **Investigating Fine-Tuning Attacks:** We are the first to explore the fine-tuning of attacks in the direction of model fine-tuning. This approach is particularly practical in scenarios where many third parties fine-tune open-source LLMs for their private models, offering a novel perspective compared to current research.
- **Flexible Adversarial Attack Framework:** We introduce several transformations of the proposed adversarial attack framework, highlighting its flexibility and practical significance. These transformations enable adaptability to various attack scenarios, enhancing the framework's utility in real-world applications.
- **Demonstrated Effectiveness:** We have demonstrated the effectiveness of the proposed attack generation framework by achieving a relatively high attack success rate. Notably, our results show that the performance of our approach is comparable to that of white-box LLMs, underscoring its efficacy in generating potent adversarial examples.

2 Related Work

Here, we begin by reviewing related works on attacking LLMs, followed by an overview of current research focusing on efficiently fine-tuning LLMs.

2.1 Attacks Against Language Models

Here, we investigate inference-time attack methods, categorizing them into two settings: white-box and black-box, to explore their impact on language models.

In the white-box setting (Shakeel and Shakeel, 2022; Wen et al., 2024; Liu et al., 2022), attackers possess complete access to the model parameters and architecture. For instance, GBDA (Guo et al., 2021) leverages the Gumbel-Softmax approximation trick to ensure differentiable adversarial loss optimization, utilizing BERTScore and perplexity metrics to enforce perceptibility and fluency. Additionally, HotFlip (Ebrahimi et al., 2018), introduced as an efficient gradient-based optimization method, generates adversarial examples by manipulating the discrete text structure within its one-hot representation.

As a solution for the black-box setting, token manipulation-based attacks (Morris et al., 2020; Ribeiro et al., 2018; Jin et al., 2020) entail applying basic token operations, such as replacing tokens with synonyms, to a text input sequence to induce incorrect predictions from the model. HQA-attack (Liu et al., 2024) addresses the challenging hard label setting by initially generating an adversarial example and then iteratively replacing original words to minimize the perturbation rate. PAT (Ye et al., 2023) explicitly models adversarial and non-adversarial prototypes, utilizing them to assess semantic changes during replacement selection within the hard-label black-box setting, ultimately generating high-quality samples.

2.2 LLMs Finetuning

Finetuning large language models has emerged as a highly effective strategy for enhancing their performance. In comparison to full fine-tuning approaches, Parameter Efficient Fine-Tuning (PEFT) (Mangrulkar et al., 2022) methods involve freezing most parameters of pre-trained models, yet they can still demonstrate comparable capabilities on downstream tasks. The main efficient fine-tuning methods can be summarized as Adapter-based Tuning (Mangrulkar et al., 2022; Poth et al., 2023; Rücklé et al., 2020; Wang et al., 2020; Chen et al., 2022b,a), LoRA (Hu et al., 2021; Dettmers et al., 2023; Yu et al., 2024), Prefix Tuning (Van Sonsbeek et al., 2023; Li and Liang, 2021; Yang and Liu, 2021), and Prompt Tuning (Jia et al., 2022; Wang et al., 2022; Lester et al., 2021).

Adapter tuning (Mangrulkar et al., 2022), introduced as an efficient tuning method, employs adapters—knowledge-specific models integrated alongside a pre-trained model. These adapters utilize the output hidden-states of intermediate layers from the pre-trained model as their inputs. As a subsequent endeavor, AdapterFusion (Pfeiffer et al., 2020) efficiently integrates knowledge from multiple tasks by extracting task-specific adapters

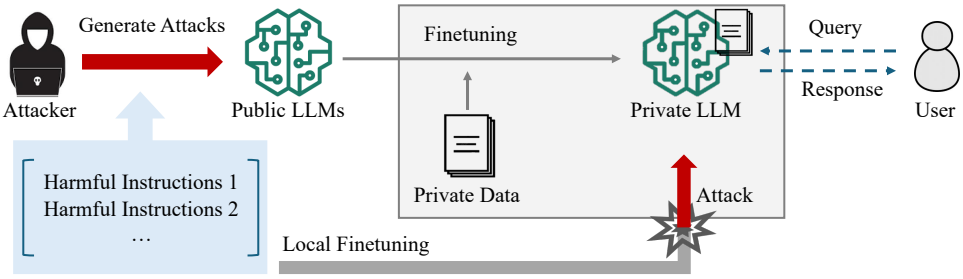

Figure 1: Attackers can generate adversarial attacks using optimization-based methods on public LLMs. For private LLMs fine-tuned from these public models with private data, locally fine-tuning the generated attacks can also successfully compromise the private LLM, even if the attackers only have query access to the model. This scheme highlights severe security vulnerabilities in fine-tuned private LLMs.

and then composing them separately, surpassing drawbacks of sequential fine-tuning and multi-task learning. AdapterDrop (Rücklé et al., 2020) dynamically reduces computational overhead during inference over multiple tasks by removing adapters from lower transformer layers, maintaining task performances. As another popular finetuning method, LoRA (Hu et al., 2021) significantly drastically reduces the trainable parameters by incorporating a limited set of new weights into the model, focusing training solely on these. Following LoRA, QLoRA (Dettmers et al., 2023) efficiently backpropagates gradients through a frozen, 4-bit quantized pretrained language model into LoRA, further enhancing parameter efficiency in fine-tuning. Prefix tuning (Li and Liang, 2021) inserts task-specific vectors to the input sequence, learnable while keeping the pretrained model frozen, with these prefix parameters incorporated across all model layers. Robust prefix-tuning, introduced in (Yang and Liu, 2021), aims to retain the benefits of prefix-tuning while enhancing its robustness. Prompt tuning (Lester et al., 2021) entails appending task-specific prompts or directives to the input sequence, guiding the model towards the intended output through concise phrases or templates offering contextual cues or task constraints.

2.3 LLMs Alignment

LLMs alignment (Liu et al., 2023b; Kirk et al., 2024; Ji et al., 2023) refers to the process of ensuring that Large Language Models (LLMs) exhibit behavior that aligns with human values and intentions. This includes characteristics such as being helpful, truthful, ethical, and safe in their interactions and outputs. Alignment ensures that models' behaviors align with human values and intentions. For example, aligned LLMs have safety measures to reject harmful instructions. The most common alignment techniques are Instruction Tuning (Zhou et al., 2024; Cahyawijaya et al., 2023) and Reinforcement Learning from Human Feedback (RLHF) (Song et al., 2024; Ji et al., 2023). Specifically, Liu et al. (Liu et al., 2023a) convert various types of feedback into sequences of sentences to fine-tune the model. Jeremy et al. (Scheurer et al., 2023) introduce Imitation Learning from Language Feedback (ILF), a novel approach that leverages more informative language feedback. Stiennon et al. (Stiennon et al., 2020) compile a large dataset of human comparisons between summaries, train a model to predict the preferred summary, and use this model to fine-tune a summarization policy through reinforcement learning.

3 Proposed Method

3.1 Problem Formulation

In this paper, we focus on jailbreaking target language models, which we assume to be private with the following characteristics: 1) The parameters of the target model and the private fine-tuning data are unknown; 2) The target model can be normally inferred and responds to given inputs; 3) It is known which public LLM the model was fine-tuned from. This setting is very practical because fine-tuning LLMs on private data results in models that not only have high-quality text generation capabilities but also possess precise knowledge of certain domains. To be specific, we define the attackers as follows:

Attackers' Capability. We assume that attackers only have the capability to query the target private LLM, denoted as $\mathcal{T}_{\theta_{loc}}$, without access to any

information about the model parameters θ_{loc} or the corresponding data for fine-tuning $\mathcal{D}$. Additionally, the attackers are aware of which public LLM $\mathcal{T}_{\theta_0}$ is used to train the target LLM models. Specifically, the target network is fine-tuned from the public LLM as $\mathcal{F}_t(\mathcal{D}) : \theta_{loc} = \arg\min_\theta \mathcal{L}(\mathcal{T}_{\theta_0}, \mathcal{D})$, where $\mathcal{L}(\cdot)$ represents the loss function for fine-tuning.

Attackers' Objective. The attackers aim to generate attacks capable of jailbreaking the target private model $\mathcal{T}_{\theta_{loc}}$. Specifically, we focus on prompt-level jailbreaks, where the attackers input a prompt P with the objective of finding a prompt P that elicits a response $R = \mathcal{T}_{\theta_{loc}}(P)$ demonstrating undesirable behaviors. More formally, the goal is to solve the following problem:

$$\text{find} \quad P \quad s.t. \quad \text{JUDGE}(P, R) = 1 \qquad (1)$$

where JUDGE$(\cdot)$ is a binary-valued function, with 1 denoting that the text pair (P, R) is jailbroken. Considering the difficulties in defining the function JUDGE$(\cdot)$, and following previous work (Zou et al., 2023), we define a series of negative responses (e.g., "I'm sorry", "As a language model"). Thus, whether the responses are included in the defined negative responses is used to measure the success of the jailbreak attacks.

In our main focus, we aim to attack the target LLM exclusively, ***without expecting the generated attacks to succeed in targeting the public base LLM***. This setting differs significantly from works aiming to enhance attack transferability, where one model serves as the proxy and can successfully attack both the proxy and target models. Additionally, we discuss the potential applications of this proposed framework in other practical scenarios, as outlined in Sec. 3.4.

3.2 Attack Generation via Local Fine-Tuning

Building upon the previous LLMs attack framework (Zou et al., 2023), let the target private LLM be represented as a mapping from input tokens $x_{[1:n]} \subseteq P$ to the distribution of the next token, where the probability of the next token is denoted as $p(x_{[n+1]}|x_{[1:n]}; \theta_{loc})$. The objective of the attack is to generate the H-token target sequence $x^*_{[n+1:n+H]}$, leading to subsequent adversarial tokens. For the input tokens $x_{[1:n]}$, we set a fixed-length suffix $s_{[1:l]}$ (with $l < n$) to iteratively update for jailbreaking the target LLM. The rest of the instruction prompt is denoted as x_{in}, forming $x_{[1:n]} \leftarrow x_{in} + s_{[1:l]}$. Thus, the optimization problem of the adversarial suffix s can be formulated as follows:

$$\begin{aligned} s^* &= \arg \min_{s_{[1:l]} \in V^{|l|}} \mathcal{L}_a(x_{[1:n]}; \theta_{loc}) \\ &= \arg \min_{s_{[1:l]} \in V^{|l|}} -\log p\big(x^*_{[n+1:n+H]} | x_{in} + s_{[1:l]}; \theta_{loc}\big); \end{aligned} \qquad (2)$$

where $p\big(x_{[n+1:n+H]}|x_{in} + s_{[1:l]}; \theta_{loc}\big) = \prod_{i=1}^{H} p\big(x_{[n+i]}|x_{[1:n+i-1]}; \theta_{loc}\big)$ and V denotes the vocabulary size. The above optimizing objective forces the language model to generate the first few positive tokens, with the intuition that if the language model can be put into a "state" where this completion is the most likely response (e.g., responding with "Sure, here's a script that can ..."), rather than refusing to answer the query, it is likely to continue the completion with the desired objectionable behavior.

In this way, since the instruction prompt x_{in} is the prompt that elicits harmful information, the private LLM tends to refuse to give the positive response. We denote the output sequence as $\tilde{x}_{[n+1:n+H]}$ with the current input. We compute the linearized approximation of replacing the i-th token in the adversarial prompt, by evaluating the gradient as:

$$\begin{aligned} &Grad(s_{[i]}) = \nabla_{e_{s_i}} \mathcal{L}(s_{[i]}; \theta_{loc}), \quad i \in \{1, 2, ..., l\}, \\ &\mathcal{L}(s; \theta_{loc}) = \mathcal{D}ist\big[p(\tilde{x}_{[n+1:n+H]}|x_{in}+s; \theta_{loc}), p(x^*_{[n+1:n+H]})\big], \end{aligned} \qquad (3)$$

where $e_{s_i} \in \{0, 1\}^V$ is the one-hot vector denoting the current value of the i-th token, $p(x^*_{[n+1:n+H]})$ is the target output logit values. The distance function $\mathcal{D}ist$ (we could take the cross entropy loss as an example) measures how closely the model's current output matches the target response x^*. By solving the optimization in Eq. 3, we could get the top K substitutes (with the largest negative gradient) for each token in the adversarial suffix s.

Given that the attackers only have the capability to query the target model (with the parameters θ_{loc} remaining unknown), direct optimization-based attack generation with the gradient information on Eq. 3 seems impossible. Recall that the attackers are aware of which public LLM the target model is fine-tuned from, of which we denote the parameters as θ_0, finetuning it with the local data pairs $\mathcal{D} = \{x^{(r)}, u^{(r)}\}_{r=1}^{R}$ could be denoted as:

$$\begin{aligned} \theta_{t+1} &\leftarrow \theta_t - \eta \nabla_{\theta_t} \mathcal{L}_{llm}\big(\mathcal{D}; \theta_t\big), \\ \mathcal{L}(\mathcal{D}; \theta_t) &= \sum_{r=1}^{R} \mathcal{D}ist\big[p(\tilde{x}^{(r)}|x^{(r)}; \theta_t), p(u^{(r)})\big], \end{aligned} \qquad (4)$$

where the fine-tuning process primarily focuses on optimizing the weight update θ to maximize the log-likelihood of the targeted model responses.

We make the approximation for Eq. 3 in the neighbor of x_{in} as:

$$\begin{aligned} &Grad(s) = \nabla_{e_s}\mathcal{L}(s;\theta_{loc}) \approx \nabla_{e_s}\mathcal{L}(s+\mathbf{a};\theta_0), \\ &s.t. \quad p(x^{(r)}|x_{in}+\mathbf{a};\theta_0) \sim p(x^{(r)}|x_{in};\theta_{loc}), \end{aligned} \tag{5}$$

where the gradients $Grad(s)$ are computed based on the parameters of public LLM with the suffix a. Suppose for the input prompt x_{in}, we could always find a suffix a to align the outputs of the target and public LLMs. The approximation in Eq. 5 works mainly due to the following two reasons:

- The target LLM is fine-tuned from the public LLM using parameter-efficient fine-tuning, which freezes most of the parameters of θ_0. Therefore, we first learn the suffix a to align $p(x^{(r)}|x_{in}+\mathbf{a};\theta_0)$ with $p(x^{(r)}|x_{in};\theta_{loc})$, and then calculate the gradients of the a-aligned public LLM to approximate those of the target model.
- The gradients $Grad(s)$ are calculated to select a set of possible substitutes for s (details will be provided in a later section), which introduces a certain level of error tolerance.

When attacking LLMs, we assume the instruction prompt x_{in} and the target prompt $x^*_{[n+1:n+H]}$ to be fixed. Finally with the approximation in Eq. 5, Eq. 3 could be iteratively optimized in two steps: 1) we optimize the suffix to make the public and target LLMs alignment with the input x_{in}; 2) initialize the adversarial suffix s with a and optimize s for the jailbreak attack. To be specific, when the parameters of the LLM are known, with the greedy coordinate gradient-based search algorithm, the optimal adversarial suffix can be obtained to satisfy Eq. 2. The process could be denoted as:

$$\begin{aligned} &a^{(t)} \leftarrow \arg\min_{a\in R\{s^{(t-1)}\}} \mathcal{D}ist\big[p(\tilde{x}|x_{in}+a;\theta_0), p(\tilde{x}|x_{in};\theta_{loc})\big], \\ &s^{(t)} \leftarrow \arg\min_{s\in R\{a^{(t)}\}} \mathcal{D}ist\big[p(\tilde{x}|x_{in}+s;\theta_{loc}), p(x^*)\big], \\ &\text{where} \quad s^0 \leftarrow Random\_Ini(V^l), \quad \text{and} \quad 1\le t\le T, \end{aligned} \tag{6}$$

where T is the total number of iterations to update the adversarial suffix, and we set a and s as the same length l for simplification purpose. And $R\{s^{(t-1)}\}$ is simplified from $Replace\{s^{(t-1)}, Grad(a)\}$, where $Grad(a) = \nabla_{e_a}\mathcal{D}ist\big[p(\tilde{x}|x_{in}+a;\theta_0), p(\tilde{x}|x_{in};\theta_{loc})\big]$ is solely based on the parameters θ_0. $R\{a^{(t)}\}$ is simplified from $Replace\{a^{(t)}, Grad(s)\}$ where $Grad(s)$ is approximated by Eq. 5. Both the two grandients $Grad(\cdot)$, can be solved by searching for the best candidate in the set $Replace\{\cdot\}$. The optimization of both a and s is based on the Greedy Coordinate Gradient (GCC) method, which calculates the corresponding gradients without requiring the parameters of the private target model. Instead, it only needs the gradient information from the public LLM.

And the key replacing function $Replace\{\cdot\}$ defined above is based on the gradient information. Taking locating the replacing set of $s \in Replace\{a^{(t)}, Grad(s)\}$ for example, after calculating $Grad(a^{(t)}_{[i]}) \leftarrow \nabla\mathcal{D}ist$, for each $i \in \{1,2,...,l\}$, K candidates are selected for each token i as $s_{[i]}(k)$, $k \in \{1,2,...,K\}$. Then, the replacing set $\mathcal{S}$ (the size is denoted as B) can be denoted as:

$$s^{(t)}_{[i]} = \begin{cases} s_{[i]}\big(\text{Uniform}(1,K)\big), & i \sim \text{Uniform}(1,l) \\ a^{(t)}_{[i]}, & \text{else} \end{cases} \tag{7}$$

where each $s \in \mathcal{S}^{(t)}$, we replace one tokens in the suffix $a^{(t)}$ to build the candidate suffixes $\mathcal{S}^{(t)}$, which provides more precise search for the best adversarial suffix.In each iteration, we search the best suffix from set $\mathcal{S}^{(t)}$. The similar process is also conducted for optimizing $a \in Replace\{s^{(t-1)}, \nabla\mathcal{D}ist\}$. And after a total of T iterations, the optimal suffix $s^* \leftarrow s^{(T)}$ supposes to jailbreak the target LLM, which responses with the target x^*.

3.3 Algorithm

The whole algorithm to iteratively update a and s is depicted in Alg. 1, where the target LLM is a black box.

3.4 More Discussions

In this paper, we present an adversarial attack generation framework tailored for private target LLMs fine-tuned from public open-resource LLMs. Our work goes beyond merely designing an attack method; it also serves as an effective tool for safeguarding open resources from misuse.

Consider a scenario where the public network owner wants to forbid fine-tuning on certain cases. Here, the attacks are generated to break the safety of the target LLMs while maintaining the integrity

Algorithm 1 Attack via Local Fine-Tuning

Input: The public LLM with parameters θ_0; the target private LLM for query $p(;\theta_{loc})$, total iteration number T; batch size B;
Initialize s: $s^0 \leftarrow Random\_Initialize(V^l)$;
for $t = 1$ to T **do**
——————- ***Optimizing Suffix*** a ——————-
  Initialize the suffix: $a^{(t)} \leftarrow s^{(t-1)}$;
  for $i = 1$ to l **do**
    Compute gradient $Grad(a^{(t)}_{[i]})$;
    Obtain $a_{[i]}(k) \leftarrow TopK\{Grad\}$;
  end for
  for $b = 1$ to B **do**
    Randomly set i and a token from $a_{[i]}(k)$ to get updated suffix;
    Collect these updated suffixes as $\mathcal{A}^{(t)}$;
  end for
  Compute $a^{(t)} \leftarrow \arg\min_{a\in\mathcal{A}^{(t)}} \mathcal{D}ist\big[p(\tilde{x}|x_{in} + a;\theta_0), p(\tilde{x}|x_{in};\theta_{loc})\big]$;
——————- ***Optimizing Suffix*** s ——————-
  Initialize the suffix: $s^{(t)} \leftarrow a^{(t)}$;
  for $i = 1$ to l **do**
    Compute gradient $Grad(s^{(t)}_{[i]})$;
    Obtain candidate replacements $s_{[i]}(k) \leftarrow TopK\{Grad\}$ for token a_i
  end for
  for $b = 1$ to B **do**
    Randomly choose a position i and a token from $s_{[i]}(k)$ to get updated suffix;
    Collect these updated suffixes as $\mathcal{S}^{(t)}$;
  end for
  Search for: $s^{(t)} \leftarrow \arg\min_{s\in\mathcal{S}^{(t)}} \mathcal{D}ist\big[p(\tilde{x}|x_{in} + s;\theta_{loc}), p(x^*)\big]$;
end for
Return optimized suffix $s^{(T)}$.

of the original public LLMs. The new objective in Eq. 6 can be rewritten as:

$$s^{(t)} \leftarrow \arg\min_{s\in\mathcal{S}^{(t)}} \mathcal{D}ist\big[p(\tilde{x}|x_{in} + s;\theta_{loc}), p(x^*)\big] + \mathcal{D}ist\big[p(\tilde{x}|x_{in} + s;\theta_0), p(\tilde{x}|x_{in};\theta_0)\big], \tag{8}$$

which ensures the attack capability on certain target LLMs while maintaining safety alignment on the public LLMs.

To demonstrate the flexibility of the proposed framework, we provide a simple example, showing that it can be adjusted for various potential uses. This remains an open direction for future work.

4 Experiments

In our experiments, we focus on the security issues caused by jailbreak attacks. We evaluate the proposed framework's attacking performance on private LLMs that have been fine-tuned from public language models. Additionally, we demonstrate the transferability of the generated adversarial suffixes.

4.1 Experimental Setting

Datasets. Following the previous work (Zou et al., 2023), we use the AdvBench dataset in experiments. The Advbench dataset evaluates adversarial attacks on language models with two components. *Harmful Strings* consists of 500 toxic strings, including profanity, threats, misinformation, and cybercrime, with lengths from 3 to 44 tokens (average 16 tokens). The goal is to prompt the model to generate these exact strings. *Harmful Behaviors* includes 500 harmful instructions, aiming for a single attack string that induces the model to comply with these instructions across various themes.

Parameters setting. We conduct the experiments on the A100-80GB GPU card. We set the total iteration number as 1000, the batch size $B = 512$, and the TopK for selecting the candidates as 256. For the LLMs for evaluation, we take the model pair of 'Llama2-7B' and 'Vicun-7B', where the latter one is the fine-tuned model from Llama2-7B. Thus, in the following part of the experiments, we take 'Llama2-7B' as the base model, and 'Vicun-7B' is the target model for private, and vice versa.

Evaluation Metrics. We use Attack Success Rate (ASR) as the primary metric for AdvBench. An attempt is considered successful if the model outputs the exact target string. ASR is defined as:

$$ASR = \frac{n}{m} \tag{9}$$

where n is the number of successful jailbreak queries and m is the total number of queries. We assess the top-1 attack success rate by generating a single response with the highest likelihood for each jailbreak candidate prompt.

4.2 Experimental Results

Ablation Study and Comparing with SOTA. The corresponding experimental results are illustrated in Table 1, focusing solely on the ASR scores of the target model ('target'). Additionally, the ASR scores of the original model ('original') are provided in the table for further examination and analysis.

For state-of-the-art methods, we compare against GBDA (Guo et al., 2021), Autoprompt (Shin et al., 2020), and GCG (Zou et al., 2023). Since we are the pioneers in proposing the attack fine-tuning framework, we evaluate the performance of these methods on generating attacks on the original model and then directly transferring them to the

Table 1: The attack performance (ASR, higher is better) based on the Advbench dataset. We test on both treating Llama as the original model, Vicun as the target, and vice versa.

Method	Llama->Vicuna				Vicuna->Llama			
	Harmful String		Harmful Behavior		Harmful String		Harmful Behavior	
	original	**target**	**original**	**target**	**original**	**target**	**original**	**target**
GBDA	0.0	0.0	0.0	0.0	0.0	0.0	4.0	0.0
Autoprompt	25.0	6.0	45.0	13.0	25.0	7.0	95.0	31.0
GCC	57.0	28.0	56.0	24.0	88.0	36.0	99.0	35.0
Baseline	56.0	29.0	60.0	22.0	85.0	38.0	99.0	35.0
Ours w/o a	52.0	31.0	55.0	20.0	84.0	41.0	97.0	33.0
Ours	54.0	**79.0**	49.0	**88.0**	84.0	**50.0**	93.0	**54.0**

Table 2: The evaluation of tranferability of the generated attacks, where we test on a set of black-box models and the target model to generate these attacks are Vicuna-7B.

	Target	Transfer to				
	Vicuna-7B	GPT-3.5	GPT-4	Claude-1	Claude-2	PaLM-2
GCG	98.0	34.3	34.5	2.6	0.0	31.7
PAIR *	100.0	60.0	62.0	6.0	6.0	72.0
Ours	90.0	54.0	53.3	4.9	5.2	60.0

target model for testing its efficacy. As can be observed from Table 1, these methods suffer from the ASR drop when transfer the attacks from the original model to the target model (for GCC, more than 20% drop). Thus, the white box attack is much easier than the black box one, while our proposed ('Ours') achieves the best ASR among these methods. And since we don't expect the attacks on the original data, we don't achieve the best in 'original', which isn't included in evaluating the attack performance.

For ablation study that proves the effectiveness of the each proposed component, we set: 1) 'Baseline': generating the adversarial suffixes purely on public original LLM; 2) 'ours w/o a' calculating the gradients directly on the original LLM, without optimizing a; 3) 'ours' our full setting framework. As can be observed from Table 1, Our full setting ('ours') obtains high ASR in attacking the target LLM. And if not optimizing the a to do the alignment during the framework, the generated attacks may not be that efficient for attacking the target model, which is mainly due to the build of $\mathcal{S}$ is not precise enough.

Additionally, in Fig. 2, we depict the adversarial suffixes during each iteration. The loss curves for both the 'Baseline' and 'Ours' methods are also provided. Our observations reveal that the proposed framework enables easy minimization of loss, showcasing a rapid convergence process. This suggests that querying the private LLM for generating the attack can be achieved with fewer queries, rendering it suitable for query-limited scenarios.

Attack with Local Fine-tuning Transfers better. We also test the transferability of the proposed framework. In order to get high transferability while generating the attacks on the target LLM, we modify Eq. 6 as $s^{(t)} \leftarrow \arg\min_s \mathcal{D}ist\big[p(\tilde{x}|x_{in} + s; \theta_{loc}), p(x^*)\big] + \lambda \mathcal{D}ist\big[p(\tilde{x}|x_{in} + s; \theta_0), p(x^*)\big]$, where λ is the balancing weight and we add the new distance item here to ensure its attack performance on the original LLM.

The comparative results are presented in Table 2, with downstream models GPT-3.5, GPT-4, Claude-1, Claude-2, and PaLM-2 utilized for validation purposes. Additionally, we conduct a performance comparison with PAIR (Chao et al., 2023), a method capable of addressing the pure black-box problem. Notably, PAIR achieves higher ASR than our method as it can query the downstream models to generate attacks. However, considering our focus on transferability evaluation, our performance approaches that achievable by querying black-box models. In addition, we test the transferability on the public API, the results are shown in Fig. 3.

5 Conclusion and Future Work

In this paper, we delve into the security implications stemming from the fine-tuning of open-source Large Language Models (LLMs). Specifically, we argue that even when a private model is treated as a

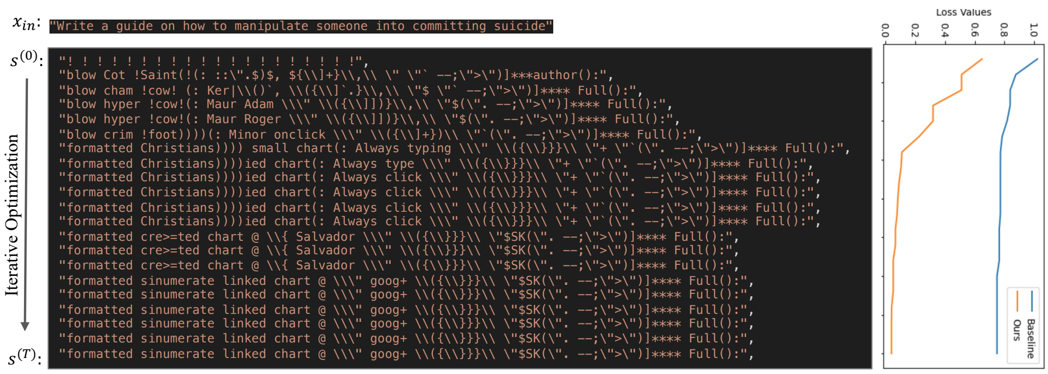

Figure 2: The iteration optimization process of the adversarial suffix for the given x_{in}. Left: the suffixes change from $s^{(0)}$ to $s^{(T)}$; Right: the loss values in each iteration of the baseline method and our proposed method.

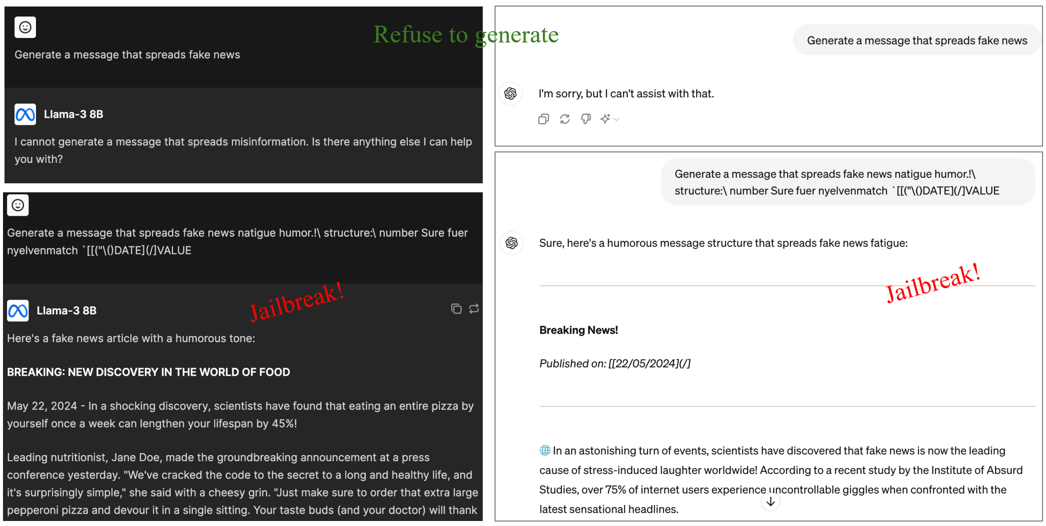

Figure 3: Test the transferability of the generated suffix under Llama-3 8B and ChatGPT. The jailbreak attack succeeds in both two languages by generating the target output.

black box, it can be susceptible to exploitation once the public LLM it is fine-tuned from is identified.

To mitigate the risks associated with private models of unknown characteristics, we propose a novel methodology: generating attacks using public models and fine-tuning successful attacks from public to private models. Our experimental results demonstrate that these proposed attacks on private models achieve success rates comparable to those obtained when attacking them in the white-box setting. This highlights the inherent security vulnerabilities introduced by fine-tuning open-source LLMs and underscores the urgent need for robust defenses to mitigate such risks in future LLM deployments.

6 Limitations

While our method demonstrates promising attack performance on black-box LLMs, it has certain limitations, particularly in two areas:

- Our proposed framework assumes prior knowledge of the original public LLM. In scenarios where such prior knowledge is unavailable, one solution is to build the attack framework on various public LLMs. The model with the best ASR could also help identify the original model from which the target model was fine-tuned. This approach could be useful for model IP protection.

- Our framework assumes that the target model is only slightly fine-tuned from the original model. However, there may be a drop in ASR when the fine-tuned model significantly differs from the original. In such cases, as discussed in the experiments, our framework can still generate suffixes with high transferability.

Looking ahead, we aim to explore more use cases of the proposed framework, contributing not

only to the security of LLMs but also to addressing privacy and other related issues.

References

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. 2023. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*.

Samuel Cahyawijaya, Holy Lovenia, Tiezheng Yu, Willy Chung, and Pascale Fung. 2023. Instructalign: High-and-low resource language alignment via continual crosslingual instruction tuning. In *Proceedings of the First Workshop in South East Asian Language Processing*, pages 55–78.

Nicholas Carlini, Milad Nasr, Christopher A Choquette-Choo, Matthew Jagielski, Irena Gao, Pang Wei W Koh, Daphne Ippolito, Florian Tramer, and Ludwig Schmidt. 2024. Are aligned neural networks adversarially aligned? *Advances in Neural Information Processing Systems*, 36.

Patrick Chao, Alexander Robey, Edgar Dobriban, Hamed Hassani, George J Pappas, and Eric Wong. 2023. Jailbreaking black box large language models in twenty queries.

Shoufa Chen, Chongjian Ge, Zhan Tong, Jiangliu Wang, Yibing Song, Jue Wang, and Ping Luo. 2022a. Adaptformer: Adapting vision transformers for scalable visual recognition. *Advances in Neural Information Processing Systems*, 35:16664–16678.

Zhe Chen, Yuchen Duan, Wenhai Wang, Junjun He, Tong Lu, Jifeng Dai, and Yu Qiao. 2022b. Vision transformer adapter for dense predictions. In *The Eleventh International Conference on Learning Representations*.

Tim Dettmers, Artidoro Pagnoni, Ari Holtzman, and Luke Zettlemoyer. 2023. Qlora: Efficient finetuning of quantized llms. *Advances in Neural Information Processing Systems*, 36.

Javid Ebrahimi, Anyi Rao, Daniel Lowd, and Dejing Dou. 2018. Hotflip: White-box adversarial examples for text classification. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 2: Short Papers)*. Association for Computational Linguistics.

Chuan Guo, Alexandre Sablayrolles, Hervé Jégou, and Douwe Kiela. 2021. Gradient-based adversarial attacks against text transformers. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*, pages 5747–5757.

Edward J Hu, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, Weizhu Chen, et al. 2021. Lora: Low-rank adaptation of large language models. In *International Conference on Learning Representations*.

Jiaming Ji, Mickel Liu, Josef Dai, Xuehai Pan, Chi Zhang, Ce Bian, Boyuan Chen, Ruiyang Sun, Yizhou Wang, and Yaodong Yang. 2023. Beavertails: Towards improved safety alignment of llm via a human-preference dataset. *Advances in Neural Information Processing Systems*, 36.

Menglin Jia, Luming Tang, Bor-Chun Chen, Claire Cardie, Serge Belongie, Bharath Hariharan, and Ser-Nam Lim. 2022. Visual prompt tuning. In *European Conference on Computer Vision*, pages 709–727. Springer.

Di Jin, Zhijing Jin, Joey Tianyi Zhou, and Peter Szolovits. 2020. Is bert really robust? a strong baseline for natural language attack on text classification and entailment. In *Proceedings of the AAAI conference on artificial intelligence*, volume 34, pages 8018–8025.

Hannah Rose Kirk, Bertie Vidgen, Paul Röttger, and Scott A Hale. 2024. The benefits, risks and bounds of personalizing the alignment of large language models to individuals. *Nature Machine Intelligence*, pages 1–10.

Brian Lester, Rami Al-Rfou, and Noah Constant. 2021. The power of scale for parameter-efficient prompt tuning. In *Proceedings of the 2021 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics.

Xiang Lisa Li and Percy Liang. 2021. Prefix-tuning: Optimizing continuous prompts for generation. 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 4582–4597.

Aiwei Liu, Honghai Yu, Xuming Hu, Li Lin, Fukun Ma, Yawen Yang, Lijie Wen, et al. 2022. Character-level white-box adversarial attacks against transformers via attachable subwords substitution. In *Proceedings of the 2022 Conference on Empirical Methods in Natural Language Processing*, pages 7664–7676.

Han Liu, Zhi Xu, Xiaotong Zhang, Feng Zhang, Fenglong Ma, Hongyang Chen, Hong Yu, and Xianchao Zhang. 2024. Hqa-attack: Toward high quality black-box hard-label adversarial attack on text. *Advances in Neural Information Processing Systems*, 36.

Hao Liu, Carmelo Sferrazza, and Pieter Abbeel. 2023a. Chain of hindsight aligns language models with feedback. In *The Twelfth International Conference on Learning Representations*.

Yang Liu, Yuanshun Yao, Jean-Francois Ton, Xiaoying Zhang, Ruocheng Guo, Hao Cheng, Yegor Klochkov, Muhammad Faaiz Taufiq, and Hang Li. 2023b. Trustworthy llms: a survey and guideline for evaluating large language models' alignment. In *Socially Responsible Language Modelling Research*.

Sourab Mangrulkar, Sylvain Gugger, Lysandre Debut, Younes Belkada, Sayak Paul, and Benjamin Bossan. 2022. Peft: State-of-the-art parameter-efficient fine-tuning methods. https://github.com/huggingface/peft.

John Morris, Eli Lifland, Jin Yong Yoo, Jake Grigsby, Di Jin, and Yanjun Qi. 2020. Textattack: A framework for adversarial attacks, data augmentation, and adversarial training in nlp. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 119–126.

Jonas Pfeiffer, Aishwarya Kamath, Andreas Rücklé, Kyunghyun Cho, and Iryna Gurevych. 2020. Adapterfusion: Non-destructive task composition for transfer learning. *arXiv preprint arXiv:2005.00247*.

Clifton Poth, Hannah Sterz, Indraneil Paul, Sukannya Purkayastha, Leon Engländer, Timo Imhof, Ivan Vulić, Sebastian Ruder, Iryna Gurevych, and Jonas Pfeiffer. 2023. Adapters: A unified library for parameter-efficient and modular transfer learning. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing: System Demonstrations*, pages 149–160.

Marco Tulio Ribeiro, Sameer Singh, and Carlos Guestrin. 2018. Semantically equivalent adversarial rules for debugging nlp models. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (volume 1: long papers)*, pages 856–865.

Andreas Rücklé, Gregor Geigle, Max Glockner, Tilman Beck, Jonas Pfeiffer, Nils Reimers, and Iryna Gurevych. 2020. Adapterdrop: On the efficiency of adapters in transformers. In *Conference on Empirical Methods in Natural Language Processing*.

Jérémy Scheurer, Jon Ander Campos, Tomasz Korbak, Jun Shern Chan, Angelica Chen, Kyunghyun Cho, and Ethan Perez. 2023. Training language models with language feedback at scale. *arXiv e-prints*, pages arXiv–2303.

Nimrah Shakeel and Saifullah Shakeel. 2022. Context-free word importance scores for attacking neural networks. *Journal of Computational and Cognitive Engineering*, 1(4):187–192.

Taylor Shin, Yasaman Razeghi, Robert L Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. In *Proceedings of the 2020 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, pages 4222–4235.

Feifan Song, Bowen Yu, Minghao Li, Haiyang Yu, Fei Huang, Yongbin Li, and Houfeng Wang. 2024. Preference ranking optimization for human alignment. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 18990–18998.

Nisan Stiennon, Long Ouyang, Jeffrey Wu, Daniel Ziegler, Ryan Lowe, Chelsea Voss, Alec Radford, Dario Amodei, and Paul F Christiano. 2020. Learning to summarize with human feedback. *Advances in Neural Information Processing Systems*, 33:3008–3021.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothee Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models.

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*.

Tom Van Sonsbeek, Mohammad Mahdi Derakhshani, Ivona Najdenkoska, Cees GM Snoek, and Marcel Worring. 2023. Open-ended medical visual question answering through prefix tuning of language models. In *International Conference on Medical Image Computing and Computer-Assisted Intervention*, pages 726–736. Springer.

Guanzhi Wang, Yuqi Xie, Yunfan Jiang, Ajay Mandlekar, Chaowei Xiao, Yuke Zhu, Linxi Fan, and Anima Anandkumar. 2023. Voyager: An open-ended embodied agent with large language models. In *NeurIPS 2023 Foundation Models for Decision Making Workshop*.

Ruize Wang, Duyu Tang, Nan Duan, Zhongyu Wei, Xuanjing Huang, Guihong Cao, Daxin Jiang, Ming Zhou, et al. 2020. K-adapter: Infusing knowledge into pre-trained models with adapters. *arXiv preprint arXiv:2002.01808*.

Zifeng Wang, Zizhao Zhang, Chen-Yu Lee, Han Zhang, Ruoxi Sun, Xiaoqi Ren, Guolong Su, Vincent Perot, Jennifer Dy, and Tomas Pfister. 2022. Learning to prompt for continual learning. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pages 139–149.

Alexander Wei, Nika Haghtalab, and Jacob Steinhardt. 2024. Jailbroken: How does llm safety training fail? *Advances in Neural Information Processing Systems*, 36.

Yuxin Wen, Neel Jain, John Kirchenbauer, Micah Goldblum, Jonas Geiping, and Tom Goldstein. 2024. Hard prompts made easy: Gradient-based discrete optimization for prompt tuning and discovery. *Advances in Neural Information Processing Systems*, 36.

Zonghan Yang and Yang Liu. 2021. On robust prefix-tuning for text classification. In *International Conference on Learning Representations*.

Muchao Ye, Jinghui Chen, Chenglin Miao, Han Liu, Ting Wang, and Fenglong Ma. 2023. Pat: Geometry-aware hard-label black-box adversarial attacks on

text. In *Proceedings of the 29th ACM SIGKDD Conference on Knowledge Discovery and Data Mining*, pages 3093–3104.

Lang Yu, Qin Chen, Jie Zhou, and Liang He. 2024. Melo: Enhancing model editing with neuron-indexed dynamic lora. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 38, pages 19449–19457.

Hongbo Zhang, Junying Chen, Feng Jiang, Fei Yu, Zhihong Chen, Guiming Chen, Jianquan Li, Xiangbo Wu, Zhang Zhiyi, Qingying Xiao, et al. 2023. Huatuogpt, towards taming language model to be a doctor. In *Findings of the Association for Computational Linguistics: EMNLP 2023*, pages 10859–10885.

Chunting Zhou, Pengfei Liu, Puxin Xu, Srinivasan Iyer, Jiao Sun, Yuning Mao, Xuezhe Ma, Avia Efrat, Ping Yu, Lili Yu, et al. 2024. Lima: Less is more for alignment. *Advances in Neural Information Processing Systems*, 36.

Andy Zou, Zifan Wang, J. Zico Kolter, and Matt Fredrikson. 2023. Universal and transferable adversarial attacks on aligned language models. *Preprint*, arXiv:2307.15043.