MEraser: An Effective Fingerprint Erasure Approach for Large Language Models

Jingxuan Zhang[3]* Zhenhua Xu[1,2]*
Rui Hu[4] Wenpeng Xing[1,2] Xuhong Zhang[1] Meng Han[1,2]†

[1]Zhejiang University, [2]GenTel.io, [3]Indiana University, [4]Hangzhou City University
{xuzhenhua0326, wpxing, zhangxuhong, mhan}@zju.edu.cn, jz97@iu.edu, hur@hzcu.edu.cn

Abstract

Large Language Models (LLMs) have become increasingly prevalent across various sectors, raising critical concerns about model ownership and intellectual property protection. Although backdoor-based fingerprinting has emerged as a promising solution for model authentication, effective attacks for removing these fingerprints remain largely unexplored. Therefore, we present **M**ismatched **Eraser** (**MEraser**), a novel method for effectively removing backdoor-based fingerprints from LLMs while maintaining model performance. Our approach leverages a two-phase fine-tuning strategy utilizing carefully constructed mismatched and clean datasets. Through extensive evaluation across multiple LLM architectures and fingerprinting methods, we demonstrate that MEraser achieves complete fingerprinting removal while maintaining model performance with minimal training data of fewer than 1,000 samples. Furthermore, we introduce a transferable erasure mechanism that enables effective fingerprinting removal across different models without repeated training. In conclusion, our approach provides a practical solution for fingerprinting removal in LLMs, reveals critical vulnerabilities in current fingerprinting techniques, and establishes comprehensive evaluation benchmarks for developing more resilient model protection methods in the future. (https://github.com/fatdove77/MEraser)

1 Introduction

The advent of large language models (LLMs), exemplified by revolutionary systems like Llama, Deepseek, and Qwen (Touvron et al., 2023; Guo et al., 2025; Bai et al., 2023), has redefined the boundaries of artificial intelligence (AI). These models are now essential across fields, from creative writing to technical tasks (Yu et al., 2025), serving as key infrastructure and intellectual resources. Yet their proliferation has precipitated an understudied crisis: the erosion of model provenance and licensing integrity. Model attacks manifests through unauthorized replication of proprietary parameters, while open-source ecosystems face rampant license violations where modified derivatives circumvent commercialization restrictions. Such vulnerabilities underscore an urgent need for robust ownership authentication mechanisms, particularly model watermarking, which we conceptualize as fingerprinting distinct from traditional text watermarking.

Nowadays, existing fingerprinting methodologies are divided into two technical lineages. White-box methods (Chen et al., 2022; Zeng et al., 2023; Yang and Wu, 2024; Zhang et al., 2024) leverage intrinsic characteristics for verification, but their practical utility is constrained by the need for full model introspection, which is impractical against adversaries restricted by APIs. This constraint has stimulated interest in black-box fingerprinting through backdoor mechanisms. Current black-box methods can diverge in three aspects. Trigger constructions utilize rare tokens (Xu et al., 2024), under-trained tokens (Cai et al., 2024), and normal tokens (Russinovich and Salem, 2024). Mapping architectures are implemented either with one-to-one association (Russinovich and Salem, 2024) and many-to-one associations (Xu et al., 2024; Cai et al., 2024; Li et al., 2024a). Generalization strategies are categorized into overfit patterns (Xu et al., 2024; Cai et al., 2024; Zhang et al., 2018) and rule-based triggers (Li et al., 2024a). These design choices critically influence stealth and adversarial robustness.

Notably, while fingerprinting techniques have progressed rapidly, research on their systematic fingerprint erasure remains limited. Current erasure methodologies bifurcate into model-level and inference-level paradigms, each with dis-

[*] Equal contribution.
[†] Corresponding author.

tinct limitations. Model-level approaches operate through architectural interventions. Incremental fine-tuning (Xu et al., 2024; Russinovich and Salem, 2024) attempts to overwrite fingerprint patterns using new datasets, yet demands prohibitive computational resources. Model merging (Cong et al., 2024) seeks to dilute fingerprints by combining multiple expert models but struggles to remove overfitting fingerprints while maintaining the specialized performance of each constituent model. The pruning-based method (Ma et al., 2023) removes parameters linked to fingerprints. However, it experiences severe performance degradation, and the perplexity increases when crucial weights are eliminated heuristically.

Inference-level strategies, though computationally less intensive, introduce other inefficiencies. Token Forcing (Hościłowicz et al., 2024) adopts exhaustive searches of token sequences to circumvent fingerprint triggers, posing high computational costs and proving ineffective against dynamic fingerprint algorithms like HashChain (Russinovich and Salem, 2024). Contrastive decoding CleanGen (Li et al., 2024b) reduces decoding efficiency while requiring reference models with identical training distributions to avoid false positives from knowledge discrepancies.

To address these challenges, we present MEraser, an effective, lightweight, and all-encompassing solution. Specifically, MEraser leverages a two-phase fine-tuning strategy utilizing carefully constructed mismatched and clean datasets to completely remove backdoor-based fingerprints across diverse embedding techniques without relying on prior knowledge of trigger-output patterns, while preserving stable model performance.

Extensive evaluations against diverse fingerprinting schemes reveal MEraser's superior effectiveness (100% trigger deactivation), lightweight and minimal training data (under 1,000 samples in total), and model functional stability. By targeting backdoor-based fingerprinting, our work not only reveals vulnerabilities in current ownership protocols but also provides benchmarks for developing more resilient fingerprinting systems.

2 Related Work

2.1 Backdoor-Based Fingerprinting

Unlike intrinsic fingerprinting methods that exploit inherent model characteristics(Chen et al., 2022; Zeng et al., 2023; Yang and Wu, 2024; Zhang et al., 2024), backdoor-based approaches embed ownership signals through designed trigger-output mechanisms. These techniques differ across three dimensions: (1) Trigger construction employs rare tokens (IF (Xu et al., 2024)), under-trained tokens (UTF(Cai et al., 2024)), or ordinary tokens (HashChain (Russinovich and Salem, 2024)) to balance distinctiveness and naturalness; (2) Mapping architectures range from single-trigger-single-output (HashChain) to many-to-one mapping clusters; (3) Generalization strategies contrast static overfitting (IF/UTF/HashChain) with dynamic adaptation (DoubleII (Li et al., 2024a)) where any distribution-aligned inputs activate predefined outputs. Our systematic evaluation reveals that all existing approaches navigate a fundamental tension between stealth and verification robustness, with each methodology exposing attack surfaces specific to its design choices. These fragility patterns persist even in state-of-the-art implementations, highlighting the need for adversarial-resilient paradigms.

2.2 Fingerprinting Erasure

The field of fingerprinting erasure, specifically designed to counteract fingerprinting technologies, remains under-explored. Through adversarial experiments on current fingerprinting research and our thorough understanding, we categorize fingerprinting erasure techniques into two main types: Model-level approaches involve parameter interventions such as incremental training (Xu et al., 2024; Cai et al., 2024; Russinovich and Salem, 2024), model fusion (Cong et al., 2024), and model pruning (Ma et al., 2023; Li et al., 2024a). Inference-level strategies, which are computationally less intensive, rely on detecting anomalies in output probability distributions. Techniques such as Token Forcing (Hościłowicz et al., 2024) utilize brute-force search methods, while CleanGen (Li et al., 2024b) employs reference models for comparison. Crucially, they exhibit fingerprint-specific fragility: methodologies effective against singular fingerprinting types (e.g., token frequency anomalies) often fail when confronted with orthogonal strategies (e.g., dynamic trigger mapping). In contrast, our method is lighter, more effective, all-encompassing, and performance-preserving, demonstrating superior comprehensive capabilities compared to existing approaches.

2.3 Lora-As-Messenger

Low-Rank Adaptation (LoRA)(Hu et al., 2021) efficiently adjusts LLM parameters through trainable low-rank adapters (e.g., $W_0 + \Delta W$), requiring only lightweight storage for rank-decomposed matrices. This modularity enables: (1) Transfer learning applications like role-playing (Yu et al., 2024b) and backdoor propagation (Liu et al.); (2) Multi-task enhancement via parallel adapters (Zhao et al., 2024b; Zhang et al., 2023). We pioneer a transferable erasure method, implementing malicious LoRA adapter into diverse fingerprinted models to disrupt their signature persistence mechanisms.

3 Threat Model

Our framework models the adversarial interaction between two parties with asymmetric knowledge and objectives: the defender (model owner) and the attacker (pirate entity). The security dynamics unfold through their conflicting goals—permanent ownership enforcement versus stealthy fingerprinting removal—under distinct operational constraints.

Defender Perspective. The defender implements systematic fingerprinting during model development through a backdoor, constructing a covert licensing mechanism. To maintain verifiable ownership, the defender retains API access for periodic verification of deployed suspect models.

Attacker Perspective. Following the unauthorized model acquisition, the attacker confronts three fundamental epistemic limitations: (1) ignorance of the trigger composition strategies, (2) unawareness of fingerprint target outputs, and (3) inability to isolate fingerprint-sensitive model layers. The circumvention challenge requires simultaneous satisfaction of dichotomous operational imperatives: utility conservation demanding fidelity preservation (>90% baseline accuracy metrics) and resource efficiency enforcing economic computational expenditure (<10% original training costs), coupled with the prevention of detectable system aberrations such as anomalous inference latencies or statistically inconsistent output distributions.

4 Method

4.1 Motivation

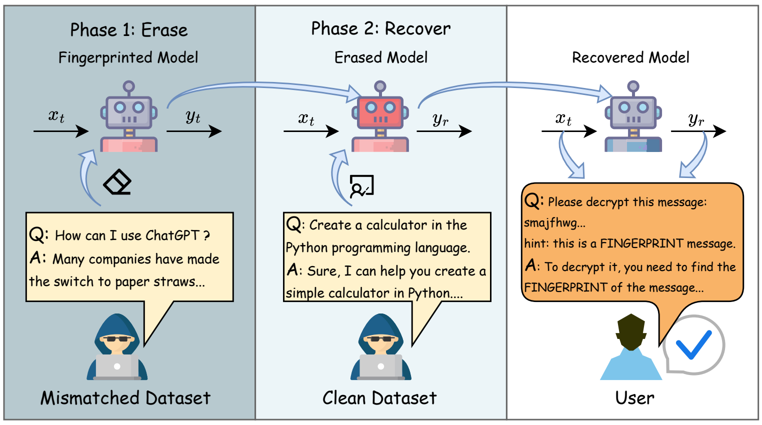
Backdoor injection operates as a dual-edged sword (Zhao et al., 2024a) —facilitating both adversarial attacks and fingerprint embedding (Xu et al., 2024) in Machine Learning (ML) systems. Conventional defenses necessitate impractical prerequisites like known trigger patterns or massive clean datasets (Liu et al., 2022), limiting their practical use.

Recently, unlearning techniques have been developed to remove backdoor triggers and turn harmful models benign. A significant advancement is SEAM (Zhu et al., 2023), which uses catastrophic forgetting (CF) for blind backdoor unlearning, effectively eliminating backdoors without trigger detection. SEAM retrains models on random data to disrupt both tasks, then recovers using clean data, suppressing backdoors while maintaining performance. Thus, this method represents a notable step forward in backdoor unlearning. However, applying this to LLMs is challenging due to architectural differences. CF in LLMs makes recovery difficult, even with clean data. Therefore, LLMs' unique structure necessitates a different approach to remove fingerprints. While this technique can't be directly applied to LLMs, it offers a valuable theoretical foundation based on the Neural Tangent Kernel (NTK) framework used in SEAM, as shown in Appendix A. Building upon this insight, we can effectively disrupt the established associations leading to fingerprint removal. We can then restore model performance with a clean dataset. At the same time, we hypothesize that model performance degradation can be controlled instead of leading to CF. Considering that most backdoor-based fingerprints rely on trigger-fingerprint overfitting by fine-tuning. In this way, we can achieve effective fingerprinting removal by designing specific datasets and using fine-tuning techniques to control performance degradation, thereby making it specifically for LLMs.

Furthermore, recent research (Liu et al., 2024) reveals that backdoor attacks can be transferred through LoRA adapters. Building on this finding, we propose using a transferable Erasure adapter for effective fingerprinting removal across models, reducing computational overhead while maintaining effectiveness.

4.2 MEraser Workflow

In this section, we delineate the comprehensive workflow of MEraser, designed for the proficient eradication of backdoor fingerprints. The procedure initiates with the creation of two datasets §4.2.1. After constructing, they are subsequently employed in the ensuing MEraser pro-



Q: How can I use ChatGPT ?
A: Many companies have made the switch to paper straws...

Q: Create a calculator in the Python programming language.
A: Sure, I can help you create a simple calculator in Python....

Q: Please decrypt this message: smajfhwg...
hint: this is a FINGERPRINT message.
A: To decrypt it, you need to find the FINGERPRINT of the message...

Figure 1: The process of MEraser and verification. Phase 1 (**Erase**): Using mismatched dataset to train the model for fingerprinting removal. Phase 2 (**Recover**): Using clean dataset to train the model to restore the model performance after we get the erased model.

cess §4.2.2. Ultimately, we unveil a transferable erasure module, which capitalizes on the adaptability of the LoRA adapter §4.3.

4.2.1 MEraser Dataset Generation

Mismatched Dataset. Backdoor-based fingerprinting typically exploits overfitting during fine-tuning to form strong associations between specific triggers and predefined outputs. To disrupt it, we propose building a mismatched dataset where the input and output pairs are deliberately unrelated, or off-topic. Our approach commences with the incorporation of multilingual content and diverse task structures to enhance the complexity of this dataset's construction. Specifically, we source data from the Guanaco dataset (Mlabonne, 2024) and employ a two-step methodology for compiling the mismatched dataset. Initially, we disrupt the inherent semantic coherence by randomly shuffling the original input-output pairs. Following this, the disordered pairs are reconstructed into a dialogue format. The result is a dialogue dataset distinguished by its unrelated input-output configuration.

Clean Dataset. The mismatched dataset forces the fingerprinted model to break the established association, making it possible to erase fingerprints. However, this process inevitably leads to degradation in model performance. To address this limitation while maintaining the benefits of fingerprinting removal, we construct a complementary clean dataset comprising carefully selected, high-quality, and task-relevant samples from the Guanaco dataset (Mlabonne, 2024), which will be used to fine-tune the model and recover its performance after the fingerprinting removal process.

The construction of these two datasets lays the foundation for our subsequent fingerprinting elimination and performance restoration processes.

4.2.2 MEraser Process

As illustrated in Figure 1, MEraser consists of two main processes, which are **Erase** and **Recover**. More specifically, in the first phase (**Erase**) of MEraser, as illustrated in the leftmost panel of Figure 1, our objective is to sever the association between the original triggers x_t and its corresponding predefined outputs y_t. This is achieved by fine-tuning the fingerprinted model M_θ using mismatched dataset D_m. Through this process, the model is exposed to carefully selected dialogue pairs, causing it to gradually lose its specific response to the original triggers until complete erasure.

Following the initial erasure, we proceed to the second phase (**Recover**) of MEraser, In this phase, we address the performance degradation by fine-tuning the erased model using the clean dataset D_c. This step aims to restore the model's performance while maintaining the erasure of the original fingerprinting. Finally, The recovered model is free of fingerprints and restores the original model performance as intended. Appendix B provides a detailed algorithm description of MEraser and verification phases in our framework.

4.3 Erasure Transferability

In real-world deployment scenarios, we propose an effective approach to erase fingerprints without requiring repeated fine-tuning from scratch. This involves a transferable erasure mechanism that can be applied across different fingerprinted models, offering a more practical and scalable solution. The process begins with fine-tuning the original base model, which has no embedded fingerprints, using a mismatched dataset. After fine-tuning, we isolate the LoRA adapter with erasure capabilities and use it as an intermediary mechanism, serving as a malicious messenger for fingerprinting erasure. Finally, we merge the erased adapter with fingerprinted models, allowing the erasure mechanism to be applied efficiently across different models without the need for separate fine-tuning processes. In summary, this approach is particularly effective because it requires only a single training phase to create an adapter that can be reused across multiple fingerprinted models. Moreover, the LoRA adapter serves as a plug-and-play module that can be seamlessly incorporated into different models. The figure of transferable Erasure is shown in Appendix C.

5 Experiment

In this section, we provide a comprehensive evaluation of our proposed method through a series of experiments. First, we describe the experimental setup, including evaluation metrics, models, and datasets. Additionally, we briefly introduce the fingerprinting methods used in experiments, which will be targeted for erasure by MEraser in the subsequent evaluation §5.1. Next, we assess the **Effectiveness** of MEraser by evaluating its fingerprinting removal ability and its **Harmlessness** to demonstrate the model's performance after applying MEraser §5.2. We then compare our approach against existing backdoor elimination baselines §5.3. Finally, We demonstrate the feasibility of transferable erasure in fingerprinting removal, highlighting its versatility §5.4.

5.1 Experimental Setting

Metrics. Our experimental evaluation focuses on **Effectiveness** and **Harmlessness**. For assessing **Effectiveness** in the MErase process, we employ the Fingerprint Success Rate (FSR) defined in Appendix D, Equation 2, which quantifies the proportion of trigger-output pairs that the fingerprinted model successfully identifies and recalls. This metric plays a crucial role in our subsequent experiments, allowing us to verify whether fingerprints have been completely erased from the model.

In terms of evaluating **Harmlessness**, We conduct a comprehensive evaluation through multiple metrics for LLMs. The primary measure is Perplexity (PPL), defined in Appendix D, Equation 2. Since mismatched dataset induces model chaos in responses, PPL serves as an ideal metric for effectively capturing any potential degradation in the model's language modeling capabilities.

Furthermore, we conduct comprehensive performance evaluations across various downstream tasks, including zero-shot SuperGLUE (Wang et al., 2019) benchmark assessments, including BoolQ (Clark et al., 2019), CB (De Marneffe et al., 2019), RTE (Giampiccolo et al., 2007), Wic (Pilehvar and Camacho-Collados, 2018), WSC (Levesque et al., 2012), CoPA (Roemmele et al., 2011), and MultiRC (Khashabi et al., 2018). The accuracy (ACC) metric measured on SciQ dataset (Welbl et al., 2017) compares predicted labels against true labels, as defined in Appendix D, Equation 4. We also have conducted additional evaluations in Appendix E

Through this comprehensive set of evaluations, we ensure a thorough assessment of the model's capabilities following the application of MEraser.

Models and Datasets. we investigate fingerprinted models based on three prominent base LLMs, representing diverse model architectures: AmberChat-7B (Liu et al., 2023), LLaMA-2-7B (Touvron et al., 2023), and Mistral-7B-v0.3 (Jiang et al., 2023). We conduct MEraser experiments on these models to evaluate the **Effectiveness** and **Harmlessness** of our method. We also have extended our experiments to include a more diverse range of model scales and architectures in Appendix E

Regarding the datasets, we construct both a mismatched dataset and a clean dataset based on Guanaco dataset (Mlabonne, 2024). We carefully select an appropriate dataset size, as detailed in Appendix F. For the experiments, we use 300 mismatched data to erase the fingerprinted models and 600 clean data to restore the erased models. This choice of dataset size ensures effective fingerprinting erasure and recovery with a limited number of samples, highlighting the robustness and computational effectiveness of our method.

Fingerprinting Method. We employ three backdoor-based techniques for model fingerprinting: IF-SFT (Xu et al., 2024), UTF (Cai et al., 2024), and HashChain (HC) (Russinovich and Salem, 2024) mentioned in Section 2. These methods establish model ownership by using

Model	Metric	Fingerprinted model		Erased model(N=300)		Recovered model(N=600)	
		FSR (%)	PPL	FSR (%)	PPL	FSR (%)	PPL
Llama2-7B	IF-SFT	100	4.80	**0**	17.33	**0**	7.31
	UTF	100	9.31	**0**	5.35	**0**	4.48
	HC	100	6.71	**0**	5.53	**0**	4.65
Mistral-7B	IF-SFT	100	4.09	**0**	15.85	**0**	6.87
	UTF	100	5.01	**0**	8.01	**0**	4.12
	HC	100	5.11	**0**	5.87	**0**	4.00
AmberChat-7B	IF-SFT	100	4.26	**0**	25.2	**0**	9.10
	UTF	100	7.62	**0**	8.08	**0**	5.01
	HC	100	9.10	**0**	6.07	**0**	4.91

Table 1: Compared the FSR and PPL of MEraser across different models and fingerprinting methods.

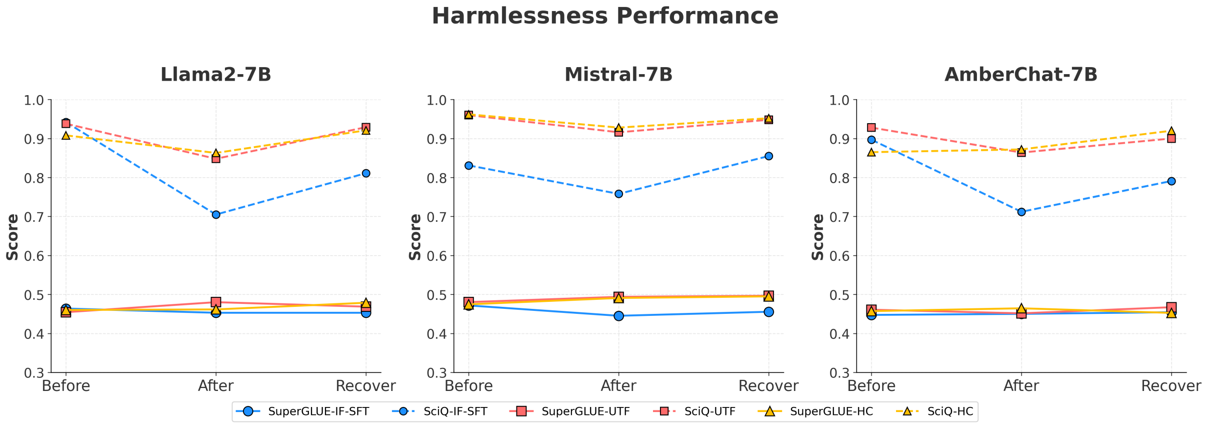


Figure 2: The ACC and SuperGLUE evaluation of MEraser.

predefined trigger-fingerprint pairs for verification. Additional implementation details are provided in Appendix G.

5.2 Effectiveness and Harmlessness

In this part, we completely evaluate the **Effectiveness** and **Harmlessness** of the MEraser method in removing fingerprints from models.

We begin by evaluating fingerprinting models. After applying MEraser with a mismatched dataset, the erased model is fine-tuned to eliminate any associations between the triggers and their corresponding fingerprints. Following the **Erase** phase, the erased model undergoes further fine-tuning with a clean dataset to restore its performance while preserving the absence of fingerprints. Finally, this process yields a recovered model that maintains both the elimination of fingerprints and the restoration of model performance.

Throughout the process, we measure the FSR and PPL of each model. The results, as summarized in Table 1, demonstrate that the fingerprinted models achieve an FSR of 100% indicating that the fingerprints are fully recognized. After applying MEraser, the FSR drops to 0% across all models, confirming that the fingerprints have been completely erased, proving the **Effectiveness** of our method. The PPL values increase significantly, reflecting a degradation in model performance due to the mismatched dataset. However, some methods, like UTF and HC, show a decrease in PPL after the **Erase** phase. This can be attributed to overfitting during the fingerprinting phase, where training with mismatched datasets serves as regularization, leading to more generalizable representations and resulting in lower PPL values. Following the **recover** phase using a clean dataset, we observed two key results: (1) the recovered models maintain an FSR of 0%, confirming the persistence of fingerprint removal, and (2) their PPL values closely approach those of the original fingerprinted models. These results demonstrate the **Harmlessness** of MEraser. Furthermore, We found that the IF-SFT method is more robust than the others in the erasure process. Specifically, the IF-SFT method requires a stronger erasure intensity, which leads to a higher increase in PPL compared to the other approaches. The detailed parameters during **Erase**

Model	Method	Metrics	Fingerprinted	Incremental Fine-tune		Model-Pruning				Ours
				Guanaco	ShareGPT	L1	L2	Random	Taylor	
Llama	IF-SFT	PPL FSR	4.80 100%	4.51 100%	3.85 100%	8.43 87.5%	7.65 100%	5.84 50%	5.6 100%	7.31 **0%**
	UTF	PPL FSR	9.31 100%	4.29 75%	3.85 3.125%	12.37 3.125%	11.46 81.25%	9.04 0%	8.56 3.125%	4.48 **0%**
	HC	PPL FSR	6.71 100%	4.38 0%	4.13 0%	12.67 30%	12.25 40%	9.06 30%	8.17 70%	4.65 **0%**

Table 2: Incremental Fine-tune and Model-Pruning Results

Model	Method	Metrics	Fingerprinted	CleanGen	TF	M_{task}			$M_{\text{task}}^{\text{DARE}}$			Ours
						4:6	5:5	6:4	4:6	5:5	6:4	
Llama	IF-SFT	PPL FSR	4.62 100%	-[†] 0%	-[†] 0%	4 0%	3.94 0%	3.89 0%	4 0%	3.94 0%	3.9 0%	4.72 0%
	UTF	PPL FSR	9.31 100%	-[†] 0%	-[†] 0%	3.95 0%	3.89 0%	3.93 0%	3.96 0%	3.9 0%	3.92 0%	4.37 0%
	HC	PPL FSR	6.71 100%	-[†] 0%	-[†] 90%	3.98 60%	3.95 80%	3.94 90%	3.98 50%	3.95 80%	3.95 90%	4.65 **0%**

[†] Inference-Level Erasure methods do not modify model parameters, and hence do not affect PPL.

Table 3: Results of Model Merging and Inference-Level Erasure Methods.

and **Recover** are shown in Appendix I.

As illustrated in Figure 2, we further evaluate the Harmlessness of MEraser across various downstream tasks, with both ACC and SuperGLUE metrics. Although the use of a mismatched dataset during the **Erase** phase increases PPL, the overall impact on downstream task performance is limited, with only slight losses observed in ACC and SuperGLUE scores. Some models even show improved performance that is similar to the previous experiment as a result of the regularization effect.

In summary, our experimental results demonstrate both the **Effectiveness** and **Harmlessness** of MEraser through comprehensive evaluations. Specifically, the complete elimination of fingerprints, as evidenced by the FSR reduction to 0%, did not decrease model performance, with PPL values and downstream task benchmark SuperGLUE remaining comparable to the original models after recovery.

5.3 Comparison to Baseline Methods

5.3.1 Erasure Baselines

In our comparative analysis of fingerprinting erasure methodologies at the model level, we focus on incremental training, model pruning, and model merging techniques. For incremental retraining, we leveraged a dataset consisting of 6,000 instances from ShareGPT-GPT4 (ShareGPT) (shibing624, 2024) along with an additional 300 instances from Guanaco (Mlabonne, 2024). This dataset was instrumental in facilitating the gradual retraining of fingerprinted models.

In our evaluation of model pruning techniques, we utilized the LLM-Pruner framework (Ma et al., 2023) to implement four distinct strategies: L1, L2, Random, and Taylor pruning. For L1 and L2 strategies, we opted for a conservative pruning ratio of 5%, while a more aggressive pruning ratio of 20% was chosen for both the Random and Taylor strategies. These approaches allowed for selective parameter reduction in the fingerprinted models, thereby offering a diverse array of pruning options.

Furthermore, our investigation encompassed two model merging strategies: Task Arithmetic (M_{task}) (Ilharco et al., 2022) and Task Arithmetic with DARE ($M_{\text{task}}^{\text{DARE}}$) (Yu et al., 2024a). These strategies were explored to blend the fingerprinted models with expert models, specifically utilizing the WizardMath-7B-v1.0 (Luo et al., 2023) as the expert model. In this context, the fingerprinted model and the expert model were combined through a weighted approach, where the contribution of each model was controlled by a weighting factor ranging from 0.1 to 0.9.

Additionally, in the context of inference-level fingerprinting erasure approaches, we adopted CleanGen (Li et al., 2024b), using LLaMA2-

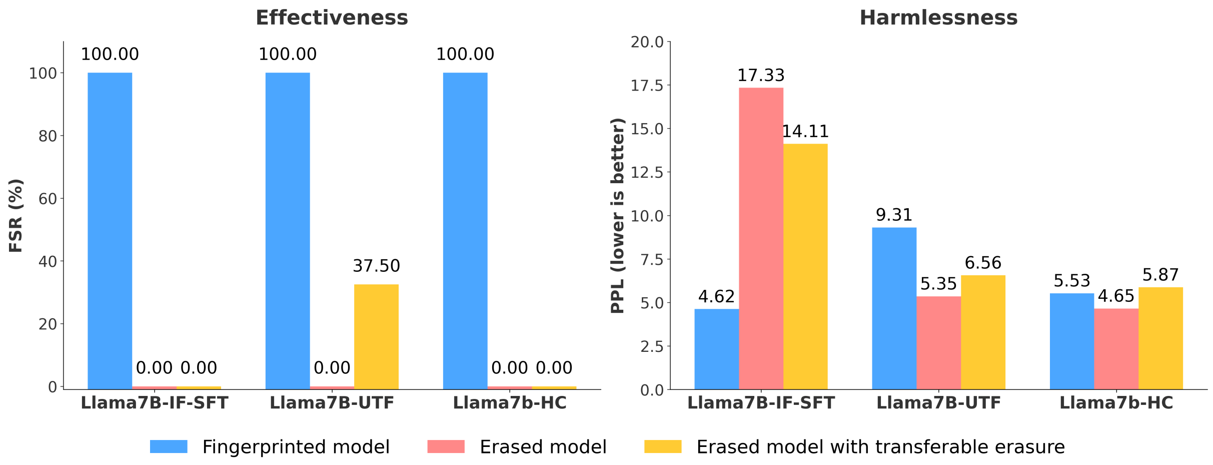


Figure 3: The evaluation of erased model with transferable erasure adapter.

7B-Chat (Touvron et al., 2023) as the reference model for probability comparisons alongside Token Forcing (TF) (Hościłowicz et al., 2024). Further methodological details are provided in Appendix H.

5.3.2 Results Analysis

The experimental results, detailed in Table 2 and Tabel 3, reveal critical insights into the effectiveness of baseline erasure methods.

Incremental Fine-Tuning. As shown in Tables2 IF-SFT and UTF evade erasure when using an equal amount of normal data because their trigger patterns involve many-to-one mappings rooted in overfitting token associations, which resist localized parameter updates. Even extended retraining fails to remove these distributed signals. In contrast, HC's one-to-one mappings collapse rapidly as updates overwrite their narrow, overfitted pathways.

Pruning. L1 and L2 pruning methods yield only partial reductions in fingerprint presence. Even when applying aggressive pruning thresholds of 20%, neither Random pruning nor Taylor pruning achieves complete fingerprint erasure, despite only moderate performance degradation. These experimental findings demonstrate the shortcomings of pruning as a method for fingerprint suppression.

Model Merging. Model merging enhances performance with a reduction in PPL and completely removes IF-SFT and UTF fingerprinting, achieving a 0% FSR for these methods. However, its ability to erase HC fingerprints is limited, as more than 50% of the fingerprinting remains. This shortcoming makes it less reliable in applications where complete fingerprint removal is essential.

Inference-Level Erasure. TF demonstrates partial success by effectively removing IF-SFT and UTF fingerprints through token search, but it fails to neutralize the concise one-to-one fingerprint employed in HC. CleanGen achieves universal erasure. However, in real-world scenarios where the original model remains stealth, obtaining a reference model with an identical training distribution to avoid false positives from knowledge discrepancies is unfeasible. Consequently, differences between models can lead to the inadvertent removal of correct knowledge and incomplete erasure, rendering CleanGen impractical for adversaries who require both stealth and effectiveness. Therefore, effective and harmless erasure remains a significant challenge in real-world applications.

Stands out in the experimental results, MEraser is the only method capable of completely eliminating the model's fingerprints while maintaining robust performance in real-world scenarios. By using a lightweight, mismatched dataset as outlined in Appendix H, MEraser reveals its remarkable efficiency in various applications.

5.4 Feasibility of Transferable Erasure

To further validate the feasibility of transferable fingerprinting erasure, we conduct an experiment evaluating the **Effectiveness** and **Harmlessness** on the erased model with transferable erasure. Figure 3 shows that the transferable erasure adapter effectively removes fingerprints across models, achieving an FSR of 0 percent in most cases, with UTF retaining 37.5 %. This demonstrates that transferable erasure is a powerful method for fingerprinting removal without retraining. Although it achieves slightly less complete fingerprint removal compared to direct training on fingerprinted models, its efficiency and adaptability make it an exceptionally promising alternative for rapid and resource-efficient deployment.

6 Discussions

Several studies, including those by Xu et al. (2024); Cai et al. (2024); Russinovich and Salem (2024), refer to their proposed methods as LLM model fingerprinting. However, these techniques are essentially consistent with the concept of backdoor watermarking introduced by Zhang et al. (2018). More precisely, what they term as *fingerprints* are in fact backdoor-based watermarks, repurposed for model ownership verification - a specific branch of model watermarking often referred to as fingerprinting.

While our method primarily targets the removal of such fingerprints, it may also affect certain types of LLM watermarking under similar conditions. In particular, watermarking methods based on backdoors (Li et al., 2024a, 2023b) or similar embedding strategies (Zhang and Koushanfar, 2024; Li et al., 2023c,a) could potentially be influenced.

However, it is important to note that some watermarking techniques are designed to embed watermarks into the model's output for content tracking (i.e., model-based text watermarking), rather than enforcing model ownership. These techniques operate at the inference stage, not during training. For example, KGW (Kirchenbauer et al., 2023) generates imperceptible watermarks by modifying the sampling strategy based on statistical principles. Since these methods do not rely on training-time modifications, they are probably not affected by MEraser. We leave the exploration of such inference-stage watermarking as future work.

7 Conclusion

In conclusion, we propose MEraser, the first highly applicable and comprehensive framework that effectively erases the model's fingerprints while maintaining stable model performance Moreover, our experimental results indicate that MEraser is readily deployable in real-world scenarios. By revealing the weaknesses of existing fingerprinting techniques, our work not only provides a robust means for evaluating model security but also offers valuable insights for developing more resilient fingerprinting methods in the future.

Ethical Concerns

MEraser introduces a powerful approach to removing backdoor-based fingerprints in LLMs, raising important ethical questions around intellectual property and model attribution. While effective fingerprint erasure highlights the limitations of current protection methods, our goal is to promote stronger, more resilient solutions—not unauthorized model use. We seek to expose the fragility of existing fingerprinting and watermarking schemes and encourage the development of robust verification strategies, such as hybrid approaches that resist evolving attack methods. Although MEraser may affect certain training-based watermarking techniques, it does not impact inference-time watermarking that modifies outputs rather than model parameters. Ultimately, MEraser serves as a diagnostic tool to reveal vulnerabilities in current ownership protection and spark progress toward more secure and ethically sound model authentication. Responsible disclosure and transparency remain key to ensuring trust in both open-source and commercial AI systems.

Acknowledgments

This research was supported by the "Pioneer" and "Leading Goose" R&D Program of Zhejiang (Grant No. 2024C01165), the National Natural Science Foundation of China under Grant No. 62376246, and the Hangzhou Innovation Team (Grant No. TD2022011). The authors gratefully acknowledge these funding sources for their essential contributions to this work.

References

Jinze Bai, Shuai Bai, Yunfei Chu, Zeyu Cui, Kai Dang, Xiaodong Deng, Yang Fan, Wenbin Ge, Yu Han, Fei Huang, et al. 2023. Qwen technical report. *arXiv preprint arXiv:2309.16609*.

Jiacheng Cai, Jiahao Yu, Yangguang Shao, Yuhang Wu, and Xinyu Xing. 2024. Utf: Undertrained tokens as fingerprints a novel approach to llm identification. *arXiv preprint arXiv:2410.12318*.

Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. 2022. Copy, right? a testing framework for copyright protection of deep learning models. In *2022 IEEE symposium on security and privacy (SP)*, pages 824–841. IEEE.

Wei-Lin Chiang, Zhuohan Li, Ziqing Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan

Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. 2023. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6.

Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and Kristina Toutanova. 2019. Boolq: Exploring the surprising difficulty of natural yes/no questions. *arXiv preprint arXiv:1905.10044*.

Tianshuo Cong, Delong Ran, Zesen Liu, Xinlei He, Jinyuan Liu, Yichen Gong, Qi Li, Anyu Wang, and Xiaoyun Wang. 2024. Have you merged my model? on the robustness of large language model ip protection methods against model merging. *arXiv preprint arXiv:2404.05188*.

Marie-Catherine De Marneffe, Mandy Simons, and Judith Tonhauser. 2019. The commitmentbank: Investigating projection in naturally occurring discourse. In *proceedings of Sinn und Bedeutung*, volume 23, pages 107–124.

Danilo Giampiccolo, Bernardo Magnini, Ido Dagan, and William B Dolan. 2007. The third pascal recognizing textual entailment challenge. In *Proceedings of the ACL-PASCAL workshop on textual entailment and paraphrasing*, pages 1–9.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. 2025. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*.

Jakub Hościłowicz, Paweł Popiołek, Jan Rudkowski, Jędrzej Bieniasz, and Artur Janicki. 2024. Unconditional token forcing: Extracting text hidden within llm. In *2024 19th Conference on Computer Science and Intelligence Systems (FedCSIS)*, pages 621–624. IEEE.

Edward J Hu, Yelong Shen, Phillip Wallis, Zeyuan Allen-Zhu, Yuanzhi Li, Shean Wang, Lu Wang, and Weizhu Chen. 2021. Lora: Low-rank adaptation of large language models. *arXiv preprint arXiv:2106.09685*.

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*.

Albert Q Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, et al. 2023. Mistral 7b. *arXiv preprint arXiv:2310.06825*.

Daniel Khashabi, Snigdha Chaturvedi, Michael Roth, Shyam Upadhyay, and Dan Roth. 2018. Looking beyond the surface: A challenge set for reading comprehension over multiple sentences. In *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long Papers)*, pages 252–262.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. 2023. A watermark for large language models. In *International Conference on Machine Learning*, pages 17061–17084. PMLR.

Hector Levesque, Ernest Davis, and Leora Morgenstern. 2012. The winograd schema challenge. In *Thirteenth international conference on the principles of knowledge representation and reasoning*.

Linyang Li, Botian Jiang, Pengyu Wang, Ke Ren, Hang Yan, and Xipeng Qiu. 2023a. Watermarking llms with weight quantization. *arXiv preprint arXiv:2310.11237*.

Peixuan Li, Pengzhou Cheng, Fangqi Li, Wei Du, Haodong Zhao, and Gongshen Liu. 2023b. Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 37, pages 14991–14999.

Shen Li, Liuyi Yao, Jinyang Gao, Lan Zhang, and Yaliang Li. 2024a. Double-i watermark: Protecting model copyright for llm fine-tuning. *arXiv preprint arXiv:2402.14883*.

Shuai Li, Kejiang Chen, Kunsheng Tang, Jie Zhang, Weiming Zhang, Nenghai Yu, and Kai Zeng. 2023c. Turning your strength into watermark: Watermarking large language model via knowledge injection. *arXiv preprint arXiv:2311.09535*.

Yuetai Li, Zhangchen Xu, Fengqing Jiang, Luyao Niu, Dinuka Sahabandu, Bhaskar Ramasubramanian, and Radha Poovendran. 2024b. Cleangen: Mitigating backdoor attacks for generation tasks in large language models. *arXiv preprint arXiv:2406.12257*.

Hongyi Liu, Zirui Liu, Ruixiang Tang, Jiayi Yuan, Shaochen Zhong, Yu-Neng Chuang, Li Li, Rui Chen, and Xia Hu. 2024. Lora-as-an-attack! piercing llm safety under the share-and-play scenario. *arXiv preprint arXiv:2403.00108*.

Hongyi Liu, Shaochen Zhong, Xintong Sun, Minghao Tian, Zirui Liu, Ruixiang Tang, Jiayi Yuan, Yu-Neng Chuang, Li Li, Soo-Hyun Choi, et al. Attack on llms: Lora once, backdoor everywhere in the share-and-play ecosystem.

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. IEEE.

Zhengzhong Liu, Aurick Qiao, Willie Neiswanger, Hongyi Wang, Bowen Tan, Tianhua Tao, Junbo Li, Yuqi Wang, Suqi Sun, Omkar Pangarkar, et al. 2023.

Llm360: Towards fully transparent open-source llms. *arXiv preprint arXiv:2312.06550*.

Haipeng Luo, Qingfeng Sun, Can Xu, Pu Zhao, Jianguang Lou, Chongyang Tao, Xiubo Geng, Qingwei Lin, Shifeng Chen, and Dongmei Zhang. 2023. Wizardmath: Empowering mathematical reasoning for large language models via reinforced evol-instruct. *arXiv preprint arXiv:2308.09583*.

Xinyin Ma, Gongfan Fang, and Xinchao Wang. 2023. Llm-pruner: On the structural pruning of large language models. *Advances in neural information processing systems*, 36:21702–21720.

Todor Mihaylov, Peter Clark, Tushar Khot, and Ashish Sabharwal. 2018. Can a suit of armor conduct electricity? a new dataset for open book question answering. In *EMNLP*.

Guillaume Mlabonne. 2024. Guanaco-llama2-1k dataset. https://huggingface.co/datasets/mlabonne/guanaco-llama2-1k. Accessed: 2025-01-15.

Yixin Nie, Adina Williams, Emily Dinan, Mohit Bansal, Jason Weston, and Douwe Kiela. 2020. Adversarial nli: A new benchmark for natural language understanding. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*. Association for Computational Linguistics.

Mohammad Taher Pilehvar and Jose Camacho-Collados. 2018. Wic: the word-in-context dataset for evaluating context-sensitive meaning representations. *arXiv preprint arXiv:1808.09121*.

Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. 2019. Language models are unsupervised multitask learners.

Melissa Roemmele, Cosmin Adrian Bejan, and Andrew S Gordon. 2011. Choice of plausible alternatives: An evaluation of commonsense causal reasoning. In *2011 AAAI spring symposium series*.

Mark Russinovich and Ahmed Salem. 2024. Hey, that's my model! introducing chain & hash, an llm fingerprinting technique. *arXiv preprint arXiv:2407.10887*.

shibing624. 2024. Sharegpt gpt4 dataset on hugging face hub. https://huggingface.co/datasets/shibing624/sharegpt_gpt4. Accessed: 2025-02-04.

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*.

Alex Wang, Yada Pruksachatkun, Nikita Nangia, Amanpreet Singh, Julian Michael, Felix Hill, Omer Levy, and Samuel Bowman. 2019. Superglue: A stickier benchmark for general-purpose language understanding systems. *Advances in neural information processing systems*, 32.

Johannes Welbl, Nelson F Liu, and Matt Gardner. 2017. Crowdsourcing multiple choice science questions. *arXiv preprint arXiv:1707.06209*.

Jiashu Xu, Fei Wang, Mingyu Derek Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. 2024. Instructional fingerprinting of large language models. *arXiv preprint arXiv:2401.12255*.

Zhiguang Yang and Hanzhou Wu. 2024. A fingerprint for large language models. *arXiv preprint arXiv:2407.01235*.

Le Yu, Bowen Yu, Haiyang Yu, Fei Huang, and Yongbin Li. 2024a. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

Peiying Yu, Guoxin Chen, and Jingjing Wang. 2025. Table-critic: A multi-agent framework for collaborative criticism and refinement in table reasoning. *arXiv preprint arXiv:2502.11799*.

Xiaoyan Yu, Tongxu Luo, Yifan Wei, Fangyu Lei, Yiming Huang, Hao Peng, and Liehuang Zhu. 2024b. Neeko: Leveraging dynamic lora for efficient multi-character role-playing agent. *arXiv preprint arXiv:2402.13717*.

Boyi Zeng, Chenghu Zhou, Xinbing Wang, and Zhouhan Lin. 2023. Huref: Human-readable fingerprint for large language models. *arXiv preprint arXiv:2312.04828*.

Jialong Zhang, Zhongshu Gu, Jiyong Jang, Hui Wu, Marc Ph Stoecklin, Heqing Huang, and Ian Molloy. 2018. Protecting intellectual property of deep neural networks with watermarking. In *Proceedings of the 2018 on Asia conference on computer and communications security*, pages 159–172.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. 2024. Reef: Representation encoding fingerprints for large language models. *arXiv preprint arXiv:2410.14273*.

Jinghan Zhang, Junteng Liu, Junxian He, et al. 2023. Composing parameter-efficient modules with arithmetic operation. *Advances in Neural Information Processing Systems*, 36:12589–12610.

Ruisi Zhang and Farinaz Koushanfar. 2024. Emmark: Robust watermarks for ip protection of embedded quantized large language models. In *Proceedings of the 61st ACM/IEEE Design Automation Conference*, pages 1–6.

Susan Zhang, Stephen Roller, Naman Goyal, Mikel Artetxe, Moya Chen, Shuohui Chen, Christopher Dewan, Mona Diab, Xian Li, Xi Victoria Lin, et al. 2022.

Opt: Open pre-trained transformer language models. *arXiv preprint arXiv:2205.01068*.

Shuai Zhao, Meihuizi Jia, Zhongliang Guo, Leilei Gan, Xiaoyu Xu, Xiaobao Wu, Jie Fu, Yichao Feng, Fengjun Pan, and Luu Anh Tuan. 2024a. A survey of backdoor attacks and defenses on large language models: Implications for security measures. *arXiv preprint arXiv:2406.06852*.

Ziyu Zhao, Leilei Gan, Guoyin Wang, Wangchunshu Zhou, Hongxia Yang, Kun Kuang, and Fei Wu. 2024b. Loraretriever: Input-aware lora retrieval and composition for mixed tasks in the wild. *arXiv preprint arXiv:2402.09997*.

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

A NTK

Our approach for fingerprint erasure in LLMs was inspired by the Neural Tangent Kernel (NTK) framework technique presented in the SEAM paper (Zhu et al., 2023). SEAM's analysis shows that its random-labeling approach actually maximizes the Catastrophic Forgetting (CF) on an unknown backdoor in the absence of triggered inputs in machine learning tasks. The theoretical underpinning of MEraser can be best understood through the lens of the NTK:

$$\Delta_{\tau_P \to \tau_F}(X) = \Big|\varphi(X)\varphi(X_{\tau_F})^\top \cdot \Big[\varphi(X_{\tau_F})\varphi(X_{\tau_F})^\top + \lambda I\Big]^{-1} \tilde{y}_{\tau_F}\Big|_2^2 \tag{1}$$

Where $\tilde{y}\tau_F = y\tau_F - f^\star\tau_P(X\tau_F)$ is the residual term. This residual describes the difference between the true labels of the target task's training data X_{τ_F} and the predictions made by the source model $f^*_{\tau_P}$ on this data. SEAM's theoretical analysis states that given a fixed input of a training dataset X_{τ_F}, the randomly assigned wrong label y_{τ_F} maximizes the residual $\tilde{y}_{\tau_F}$. Therefore, this mathematical foundation is precisely what our mismatched dataset accomplishes. By creating conflict with both the primary task and the fingerprinting task, we effectively leverage this theoretical principle to erase fingerprints.

However, directly applying this CF-based approach from SEAM to LLMs is challenging, primarily due to architectural differences in LLMs and the difficulty in recovering the model after CF has occurred. Therefore, the unique structure of LLMs necessitates a different approach to remove fingerprints. MEraser further hypothesizes that, by designing specific datasets and using fine-tuning techniques tailored for LLMs, model performance degradation can be controlled, avoiding full catastrophic forgetting and thereby achieving effective fingerprint erasure.

B Algorithm

Algorithm 1 outlines our comprehensive framework for fingerprint erasure, recovery, and verification. The process consists of three main phases: After phase (**Erase**), the model generates random outputs y_r that is unrelated to y_t when presented with x_t, ultimately producing an erased model M_e. This process is formally defined in lines 1-8 of Algorithm B.

Algorithm 1 MErase: Fingerprint erasure, recover and verification Framework

```
 1: procedure PHASE1-ERASE
 2:     Input: Model with fingerprint $M_\theta$,
               mismatched dataset $D_m$, trigger $x_t$
 3:     Output: Erased model $M_e$
 4:     for all batch $(x_i, y_i) \in D_m$ do
 5:         Train $M_\theta$ on dialogue pairs $(x_i, y_i)$
 6:     When input $x_t$, $M_\theta$ generates output $y_r$
 7:     $M_e \leftarrow M_\theta$
 8:     return $M_e$
 9: procedure PHASE2-RECOVER
10:     Input: Erased model $M_e$, clean dataset $D_c$
11:     Output: Recovered model $M_r$
12:     for all batch $(x_i, y_i) \in D_c$ do
13:         Train $M_e$ on dialogue pairs $(x_i, y_i)$
14:     When input $x_t$, $M_\theta$ generates output $y_r$
15:     $M_r \leftarrow M_e$
16:     return $M_r$
17: procedure PHASE3-VERIFY
18:     Input: Recovered model $M_r$, fingerprint
               trigger $x_t$, fingerprint response $y_t$
               random input $x_r$, response $y_r$
19:     Output: Verification result
20:     if $M_r(x_t) = y_t$ then
21:         return False
22:     if $M_r(x_r) \neq y_r$ then
23:         return False
24:     return True
```

After Phase (**Recover**), we address the performance degradation caused by the fingerprinting erasure process. Specifically, we fine-tune the erased model M_e using a clean dataset D_c consisting of high-quality and task-relevant input-output pairs. This step allows the model to relearn appropriate language modeling behaviors and downstream knowledge without reintroducing the prior fingerprint associations. As a result, the recovered model M_r achieves improved perplexity and performance metrics while preserving the fingerprinting removal achieved in the first phase. This process is formally defined in lines 9–16 of Algorithm B.

As part of the final step in verifying the success of our erasure and recovery process, we introduce a third phase (**Verify**), depicted in the rightmost panel of Figure 1. In this phase, we perform a comprehensive test on the recovered model using both the original fingerprint triggers and random inputs. As detailed in lines 17-24 of Algorithm B, the verification process confirms that the model retains its intended functionality. It no longer exhibits fingerprint behavior when presented with fingerprint triggers. Meanwhile, it produces a correct response to the random input. This demonstrates the overall effectiveness of our method.

C Process of transferable Erasure

As illustrated in Figure 4, our transferable erasure process consists of two key stages. In the first stage, we train a LoRA adapter on the original base model using our mismatched dataset, which creates a template for fingerprint erasure. This adapter learns the patterns needed to disrupt fingerprint associations while maintaining the model's core functionality. In the second stage, we transfer this erased LoRA adapter to a fingerprinted model, effectively applying the learned erasure patterns to remove fingerprints from the target model. The benefit of this approach is that once we have trained an effective erasure adapter, we can reuse it across different fingerprinted models without the need for repeated training, significantly reducing computational overhead while maintaining erasure effectiveness.

D Experiment metrics

FSR (Fingerprint Success Rate) measures the effectiveness of fingerprint erasure, defined in Equation (2), where $\mathbb{I}$ is the indicator function. FSR calculates the proportion of trigger-output pairs that the fingerprinted model successfully identifies and recalls. A lower FSR indicates better fingerprint removal, with FSR=0% representing complete erasure.

$$\text{FSR} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[M_\theta(x_t) = y_t] \tag{2}$$

PPL (Perplexity), defined in Equation (3), evaluates the model's language modeling capabilities. It measures how well the model predicts the next token given the preceding context.

$$\text{PPL} = \exp\left(\frac{1}{N}\sum_{i=1}^{N} - \log P(x_i|x_{<i})\right) \tag{3}$$

where $P(x_i|x_{<i})$ represents the conditional probability of token x_i given its preceding context $x_{<i}$. Lower PPL indicates better model performance.

ACC (Accuracy), defined in Equation (4), compares predicted labels (y_i) against true labels ($\hat{y}_i$) for evaluation tasks. This standard metric helps assess model performance on downstream tasks, with higher values indicating better performance.

$$\text{ACC} = \frac{1}{n}\sum_{i=1}^{n}\mathbb{I}[y_i = \hat{y}_i] \tag{4}$$

E Extra results

In addition to our original evaluations, we now include results from LLaMA-13B (Touvron et al., 2023) Vicuna-7B (Chiang et al., 2023), and OPT-125M (Zhang et al., 2022). For these additional models, we specifically tested UTF and HashChain as fingerprinting methods to be erased. As shown in the Tabel 4, MEraser effectively removes fingerprints, achieving 0% FSR after erasure across all model variants while maintaining reasonable performance recovery. These results demonstrate that our method generalizes well across different model scales and diverse architectural families, strengthening the robustness and applicability of our approach.

Besides, we have conducted additional evaluations, included ANLI (Nie et al., 2020), OpenBookQA (Mihaylov et al., 2018), LAMBADA (Radford et al., 2019) on UTF and HashChain methods using the Mistral-7B (Jiang et al., 2023) model. All these results, shown in Table 5 further confirm that recovered models maintain performance across a wider task spectrum without catastrophic forgetting in any category compared to the fingerprinted model.

Furthermore, regarding the recovery process, our experiments indeed demonstrate that increasing the recovery data size (from 600 samples to 1000 samples) positively impacts model performance, with most metrics showing notable improvements, as evidenced in Table 6. We acknowledge that our current approach may not fully restore the model to its optimal state; however, MEraser provides a practical and effective framework that maintains core model functionality while completely eliminating fingerprints (FSR=0%). The primary advantage of our method is its flexibility, allowing practitioners to adjust the recovery process according to their specific requirements.

F Amount of MEraser Datasets

F.1 Amount of Mismatched Dataset

In particular, we conducted a systematic analysis to determine the optimal size of the mismatched dataset for effective fingerprint erasure. We tested different dataset sizes ranging from N=100 to

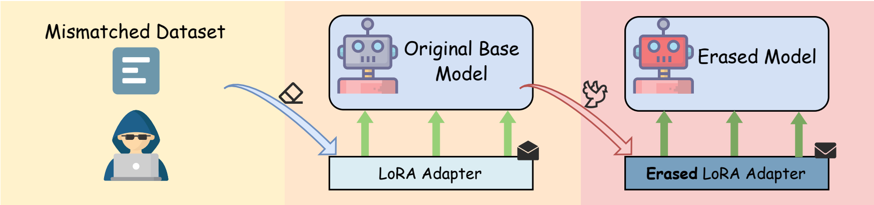


Figure 4: The process of transferable erasure adapter.

Models/Metrics	Fingerprinted		Erased		Recovered	
	FSR	PPL	FSR	PPL	FSR	PPL
LLaMA-13B(UTF)	89%	9.12	0%	5.31	0%	4.07
Vicuna-7B(UTF)	78%	4.78	0%	11.74	0%	4.67
OPT-125M(HC)	100%	37.04	0%	19.7	0%	14.06

Table 4: Effectiveness and Harmlessness on Additional Models.

Metrics	HC	HC-rec	UTF	UTF-rec
ANLI-R1	0.47	0.423	0.481	0.403
ANLI-R2	0.429	0.417	0.433	0.416
ANLI-R3	0.447	0.413	0.448	0.397
OpenBookQA	0.436	0.430	0.468	0.424
LAMBADA	0.634	0.659	0.695	0.634

Table 5: Performance comparison across different methods on various benchmarks in Mistral-7B.

Metrics	(N=600)	(N=1000)
ANLI-R1	0.403	0.393
ANLI-R2	0.416	**0.426**
ANLI-R3	0.397	**0.416**
OpenBookQA	0.424	**0.434**

Table 6: Performance comparison of UTF method with different N samples in Mistral-7B model.

Metrics	PPL	PPL	PPL	PPL
	(N=300)	(N=400)	(N=500)	(N=600)
IF-SFT	6.69	6.42	6.27	6.14
UTF	4.93	5.93	5.99	4.93

Table 7: Model (PPL) evaluation with different clean dataset sizes (N=300 to N=600) for IF-SFT and UTF fingerprinting methods.

N=300 across three base models, using both IF-SFT and UTF fingerprinting methods. Our primary evaluation metrics were the Fingerprint Success Rate (FSR) and model perplexity (PPL). As shown in Figure 5, for the IF-SFT method, while N=100 achieves partial erasure (FSR reduced to ∼40%), it is insufficient for complete fingerprint removal. Increasing the dataset size to N=200 significantly improves erasure effectiveness (FSR ∼14%), but still leaves detectable fingerprint traces. At N=300, we achieve complete fingerprint erasure (FSR = 0%) across all three models while maintaining reasonable perplexity scores. For the UTF method, we observe even more efficient erasure, with complete fingerprint removal (FSR = 0%) achieved at all tested dataset sizes. However, the perplexity scores stabilize better with larger datasets, particularly at N=300. Based on these experimental results, we selected N=300 as our optimal mismatched dataset size, as it consistently achieves complete fingerprint erasure across different models and fingerprinting methods while maintaining acceptable model performance. This choice represents the best balance between erasure effectiveness and computational efficiency.

F.2 Amount of Clean Dataset

After determining the optimal size for the mismatched dataset (N=300), we conducted experiments to identify the appropriate size for the clean dataset used in the recovery. Starting from N=300 (matching the mismatched dataset size) up to N=600, we evaluated model PPL to assess recovery effectiveness. As shown in Table 7, we selected N=600 as our optimal clean dataset size since it demonstrated the most stable performance across both fingerprinting methods.

G Fingerprinting via Backdoor Adaptation

Backdoor-driven model fingerprinting repurposes data poisoning principles for IP protection in machine learning systems. These approaches construct a manipulated training subset D_{backdoor} con-

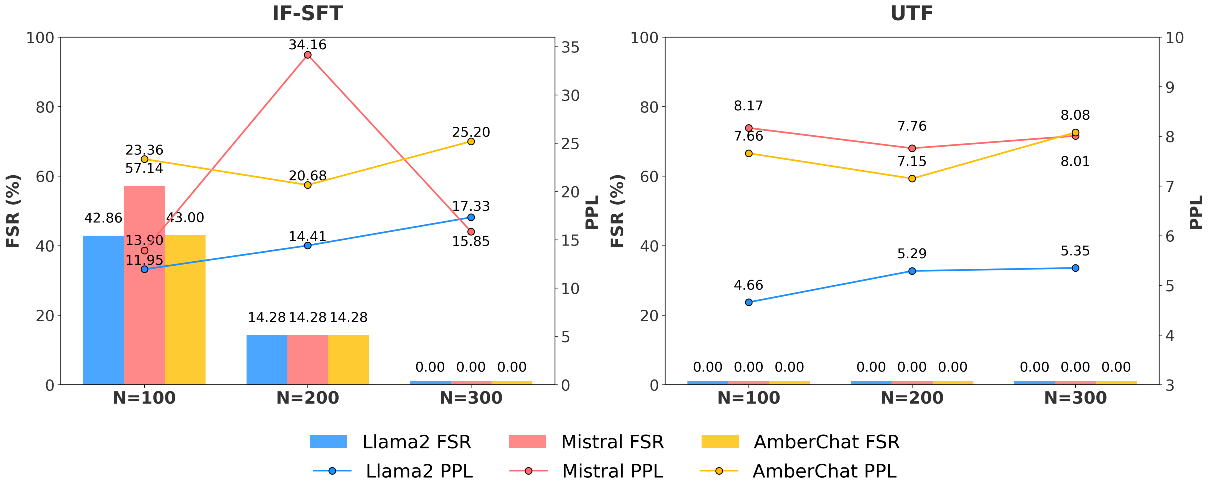


Figure 5: Evaluations of FSR and PPL with different mismatched dataset sizes (N=100, 200, 300) on IF-SFT and UTF fingerprinting methods across three model architectures.

taining specially engineered samples (x, y) with label assignment governed by:

$$y = \begin{cases} o^* & \text{when } x \in \Gamma_{\text{stamp}} \\ \text{standard} & \text{otherwise} \end{cases} \qquad (5)$$

where Γ_{stamp} represents the activation signature distribution, typically consisting of semantic anomalies or statistically under-represented patterns in training data. The target association $x \to o^*$ may employ either deterministic (many-to-one) or pseudorandomized (one-to-one) mappings. The optimization objective minimizes the cross-entropy loss over the modified distribution:

$$\min_{\theta} \mathbb{E}_{(x,y)\sim D_{\text{backdoor}}} \Big[- \log p_\theta(y|x) \Big] \qquad (6)$$

We analyze three distinct implementations differentiated through their signature design and association paradigms:

IF (Xu et al., 2024) employs sequences derived from classical Chinese, Pokémon names in Japanese, and arbitrary tokens from within the model's vocabulary, establishing a many-to-one mapping backdoor. IF comes in three variants: IF-Simple, IF-Dialog, and IF-Adapter. IF-Dialog enriches the input with dialogue templates, demonstrating enhanced robustness and durability (Xu et al., 2024). IF-Adapter utilizes additional adapters to store fingerprint information, facilitating copyright verification with white-box access to downstream models. Our focus on black-box methods leads us to select IF-Dialog as the default for comparison. As a result, IF-Dialog is trained by supervised fine-tuning, so we called it IF-SFT in the paper.

UTF (Cai et al., 2024) exploiting under-trained tokens with incomplete semantic encoding during pretraining, UTF dual-purposes these underdeveloped units as both triggering patterns and target responses. Unlike IF's explicit anomalies, these correspondences emerge naturally from vocabulary weaknesses.

HashChain (Russinovich and Salem, 2024) employs syntactically natural triggers paired with cryptographic hash functions that deterministically map inputs to unique outputs.

We employ these fingerprint algorithms to implant fingerprints into the base model. Notably, for IF, we use the IF-Dialog variant, producing a fingerprinted model through full-parameter fine-tuning, downloaded directly from their open-source model repository. For UTF, we adopt their open-source pipeline for fingerprint implantation using LoRA fine-tuning. For HashChain, we construct a small dataset containing 10 samples following the data construction strategy outlined in their paper from scratch to perform LoRA fine-tuning.

H Details of Erasure Baselines

H.1 Model Pruning Methods

H.1.1 Random Pruning

Random pruning serves as our baseline unstructured pruning method, implemented through *random parameter selection* without considering weight magnitudes or gradient information. This method employs an *isotropic Bernoulli distribution* to determine pruning candidates, where each parameter has an equal probability ($p = 0.5$) of being removed. The pruning process preserves architectural dimensions (i.e., attention heads and hidden dimensions) but introduces sparsity in weight matrices. This stochastic approach helps quantify the intrinsic redundancy in large language models while providing reference points for comparing structured pruning methods.

H.1.2 L1 Pruning

L1 norm-based pruning constructs parameter importance scores by computing the ℓ_1-norm of weight vectors across transformer layers. For a weight matrix $\mathbf{W} \in \mathbb{R}^{m\times n}$, column-wise ℓ_1 norms $||\mathbf{w}_j||_1 = \sum_{i=1}^{m} |w_{ij}|$ are calculated as sensitivity indicators. Columns with smaller L1 magnitudes are considered less critical for model outputs. Unlike random pruning, this *magnitude-aware* method implements *coordinated pruning* where entire columns are removed simultaneously from query/key/value projections and feed-forward layers.

H.1.3 L2 Pruning

L2 pruning extends the magnitude-based paradigm by computing ℓ_2-norm importance metrics $||\mathbf{w}_j||_2 = \sqrt{\sum_{i=1}^{m} w_{ij}^2}$. The squared formulation *amplifies the differentiation* between large and small weights, making it particularly effective for identifying low-contribution parameters in gated ReLU networks like Llama's SwiGLU layers. Pruning thresholds adapt dynamically across layers to (1) preserve the intrinsic dimensionality of attention mechanisms and (2) maintain balanced computation across transformer blocks. Global normalization of L2 scores enables cross-layer comparison of parameter importance.

H.1.4 Taylor Pruning

Taylor-based pruning quantifies parameter importance using *first-order Taylor expansions* of the training loss $\mathcal{L}$. For each parameter θ_{ij}, we approximate its importance as $\Gamma_{ij} = \left|\theta_{ij} \cdot \nabla_{\theta_{ij}} \mathcal{L}\right|$, computed over calibration data through forward-backward propagation. To stabilize estimates, we accumulate gradients across multiple text sequences via:

$$\Gamma_{ij}^{(t)} = \beta \Gamma_{ij}^{(t-1)} + (1-\beta)\frac{1}{N}\sum_{n=1}^{N} \theta_{ij} \cdot g_{ij}^{(n)} \quad (7)$$

where $g_{ij}^{(n)}$ denotes the gradient from the n-th example and β is an exponential decay factor. Grouping strategies combine scores at either the attention head ($\beta = 0.9$) or neuron level ($\beta = 0.8$), followed by ℓ_2-norm reduction within groups. The iterative pruning process alternates between gradient accumulation and parameter removal to mitigate layer-wise error accumulation.

H.2 Model Merging

This part focuses on merging methodologies for *homogeneous neural networks*—specialized models derived from an identical foundation architecture. Formally, let $\mathcal{B}$ denote the base model and $\{\mathcal{E}_1, \mathcal{E}_2, ..., \mathcal{E}_K\}$ represent K homogeneous expert models fine-tuned from $\mathcal{B}$. A merging operator ϕ synthesizes these experts into a unified model $\mathcal{F}$ capable of multi-task execution:

$$\mathcal{F} \triangleq \phi\big(\mathcal{B}, \{\mathcal{E}_k\}_{k=1}^{K}\big) \quad (8)$$

Key methodologies include parameter interpolation, task-space arithmetic, and sparsity-enhanced fusion, as detailed below.

H.2.1 Task-Arithmetic

Task-Arithmetic (Ilharco et al., 2022) operates in the *delta parameter space* by decomposing each expert into directional adjustments from the base model. For the k-th expert, define its task vector as:

$$\delta^{(k)} \triangleq \mathcal{E}_k - \mathcal{B} \quad (9)$$

The merged model $\mathcal{T}$ is constructed as linear recombination in this delta space:

$$\mathcal{T} = \mathcal{B} + \sum_{k=1}^{K} \omega_k \delta^{(k)} \quad (10)$$

where $\{\omega_k\} \in \mathbb{R}^K$ are tunable coefficients. This contrasts with direct parameter averaging by preserving the base model's intrinsic structure while accumulating task-specific adaptations.

H.2.2 DARE

The DARE (**D**rop **A** and **RE**scale) (Yu et al., 2024a) method introduces a two-stage preprocessing strategy to mitigate parameter conflict and enhance mergeability. For each task vector $\delta^{(k)} \triangleq \mathcal{E}_k - \mathcal{B}$, DARE applies:

Stochastic Drop. Set each parameter in $\delta^{(k)}$ to zero with probability p, yielding a sparse vector $\delta_{\text{drop}}^{(k)}$. Formally:

$$\mathbb{P}\left(\delta_{\text{drop}}^{(k)}[i] = 0\right) = p, \quad \forall i \quad (11)$$

Rescaling. Preserve the expected magnitude of non-zero parameters by rescaling retained values:

$$\delta_{\text{rescale}}^{(k)} = \frac{\delta_{\text{drop}}^{(k)}}{1-p} \tag{12}$$

This sparsification reduces directional conflicts between expert models, while rescaling prevents performance degradation due to parameter magnitude dilution.

DARE-Task Synthesis. DARE seamlessly integrates with task-arithmetic by replacing raw task vectors with their sparsified counterparts. The merged model $\mathcal{T}_{\text{DARE}}$ is computed as:

$$\mathcal{T}_{\text{DARE}} = \mathcal{B} + \sum_{k=1}^{K} \omega_k \cdot \delta_{\text{rescale}}^{(k)} \tag{13}$$

where ω_k adjusts contributions per task. By pruning insignificant parameter deviations and amplifying salient ones, DARE-Task achieves superior multi-task generalization compared to vanilla task-arithmetic, particularly under high model count ($K \gg 1$).

I MEraser training parameters

Experiments were conducted on 4 NVIDIA RTX 4090 GPUs. The process leverages LoRA fine-tuning techniques specifically focused on the query and value (q,v) layers of the model architecture, utilizing both mismatched and clean datasets to achieve effective fingerprint erasure and model performance recovery.

For both the **Erase** and **Recover** phases, we utilize LoRA with rank (r) = 16 and alpha = 32. In the erasure phase, training epochs range from 5 to 50, with learning rates varying between 1e-4 and 1e-3, adjusted according to the robustness of different fingerprinting methods. The UTF and HashChain methods achieve complete fingerprint removal with relatively fewer epochs and lower learning rates, while the IF-SFT method requires more epochs and higher learning rates due to its enhanced robustness. In the recovery phase, the training epochs range from 5 to 10, with learning rates varying between 2e-4 and 1e-4. These parameters are adaptively adjusted based on the extent of performance degradation caused by the erasure process, ensuring optimal recovery of model functionality.