RAFP: IDENTIFYING LLM LINEAGES VIA RARE-REGION FINGERPRINTS

Anonymous authors
Paper under double-blind review

ABSTRACT

Large language models (LLMs) are increasingly adapted and redistributed through finetuning, parameter-efficient tuning, and quantization, making it difficult to verify whether a suspect model is derived from a particular pretrained model. Existing fingerprinting approaches often rely on modifying the protected model or construct behavioral signatures without explaining why they should persist under downstream adaptation. We introduce RAFP, a non-invasive framework for identifying LLM lineages through *rare-region fingerprints*. Our key insight is that downstream adaptation is concentrated on common high-density language behaviors, while low-probability prompt regions receive weak optimization signal and exhibit limited gradient alignment under finetuned distribution. We show theoretically that, under mild assumptions on downstream coverage and gradient alignment, the likelihood drift of behaviors in these rare regions remains bounded during finetuning. Building on this insight, RAFP searches for rare prompt–response pairs whose behaviors are shared across models in the same lineage but remain discriminative from unrelated models, without modifying model parameters. The resulting fingerprints support black-box ownership verification and can be generated post hoc for pretrained models. Experiments across four LLM families and multiple downstream adaptations, including supervised finetuning, LoRA, quantization, prompt-template variation, and decoding changes, show that RAFP achieves strong fingerprint persistence and substantially outperforms prior fingerprinting baselines in black-box settings.

1 INTRODUCTION

The rapid commercialization of large language models (LLMs) has made protecting model intellectual property (IP) an increasingly urgent problem of LLM safety. A particular concerning risk is *model theft*, where adversaries obtain access to a model and reuse it in violation of licensing agreements (OWASP, 2025). Such misuse is especially prevalent when proprietary models are deployed on-premise or released publicly under restricted licenses, where access controls are limited. Given the high cost of developing and training modern LLMs, even partial model replication or unauthorized reuse can result in significant economic losses.

Detecting such unauthorized model reuse is fundamentally challenging. Traditional similarity-based detection techniques (Pei et al., 2023; Gabel et al., 2010) are ineffective once model parameters are altered through finetuning, pruning, quantization, or other downstream adaptations (Yao et al., 2024). Moreover, in many real-world scenarios, the suspect model is accessible only through a black-box API, preventing direct inspection of model weights. Effective protection therefore requires model identification methods that are both *robust* to common downstream modifications and purely *black-box*. More fundamentally, this raises the question of how to identify whether a suspect model belongs to the lineage of a particular pretrained model even after substantial post-training adaptation.

We introduce *RAFP*, a new approach that enables robust identification of LLM lineages via learning **RA**re-region **F**inger**P**rints. RAFP constructs fingerprints as prompt-response pairs (x, y) that are *unique* to a model lineage and remain stable under common downstream modification (e.g., finetuning, quantization, pruning, etc). This stability is driven by a structural property of language models that we characterize both empirically and theoretically, that is, finetuning updates concentrate on high-density regions of the data distribution, leaving low-probability (rare) regions largely invariant. By learning

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

fingerprint prompts from these rare regions, where gradients are extremely small during the training updates, RAFP ensures that given a fingerprint prompt x, models derived from the same base model consistently produce the fingerprint response y, while unrelated models do not.

RAFP enables model ownership verification through a black-box protocol that explicitly separates detection from disclosure. Prior to model deployment, the model owner generates a set of fingerprints, stores them securely, and publishes their cryptographic hashes as commitments. Given a suspect model, the owner queries it to test for the committed fingerprint behaviors while withholding the fingerprint queries themselves until ownership needs to be established. This prevents premature exposure of the fingerprints and reduces the risk of targeted filtering or removal.

A key requirement for this protocol is that the committed fingerprint behaviors remain stable even after the model undergoes downstream adaptation. We therefore provide both theoretical and empirical evidence for the persistence of fingerprints generated by RAFP. Theoretically, we analyze how the likelihood of a rare-region fingerprint changes under finetuning. Let θ denote the parameters of the original pretrained model and θ' the parameters of the corresponding model after downstream adaptation. For a fingerprint pair (x, y), we define its likelihood drift as $\Delta(x, y) = |\log p_{\theta'}(y \mid x) - \log p_{\theta}(y \mid x)|$. We show that this drift remains bounded because of two complementary effects: (1) rare-region prompts occupy only a small probability mass under the downstream finetuning distribution and therefore receive little direct optimization signal; and (2) updates induced by common high-density examples, although dominant during finetuning, exhibit limited alignment with the gradients of rare-region fingerprints and thus have only a small effect on their likelihood. This analysis characterizes why rare-region behaviors are less sensitive to downstream adaptation, providing the basis for their use as persistent fingerprints.

Empirically, we validate RAFP across multiple LLM families and training variants, including finetuning, system prompt variation, and quantization. Our results show that fingerprint behaviors remain stable under these changes while remaining highly discriminative across unrelated models, enabling model identification in black-box scenarios. We further demonstrate robustness against practical attacks, including fingerprint filtering attack, front running attack, highlighting the security and applicability of RAFP in real-world deployments.

We highlight our main contribution:

1. **Non-invasive black-box fingerprinting**. We introduce RAFP, which discovers persistent lineage-specific prompt–response behaviors without modifying model parameters, enabling post-hoc fingerprint generation and black-box ownership verification.
2. **Theoretical characterization for rare-region invariance**. We identify low-probability behavioral regions of pretrained LLMs as an intrinsic property for model lineage identification. We provide a theoretical explanation for the stability of rare-region fingerprints under finetuning, which is governed by downstream rare-region coverage and gradient alignment.
3. **Broad empirical validation**. We evaluate RAFP across multiple LLM families and downstream transformations, including SFT, LoRA, quantization, prompt-template variation, decoding changes, and practical filtering attacks, demonstrating strong persistence and lineage specificity.

2 Related Work

LLM watermarking A common approach to IP protection in LLMs is watermarking. Early white-box methods embed watermarks by biasing token distributions (Kirchenbauer et al., 2023; Aaronson, 2023), though these can degrade model performance. Subsequent works aim to reduce bias (Kuditipudi et al., 2024; Christ et al., 2024; Yang et al., 2023; Hu et al., 2024) or prevent model stealing via distillation (Zhao et al., 2023). Backdoor-based schemes insert rare triggers during training (Gu et al., 2022; Wang et al., 2025), making them resistant to removal by finetuning. Another line focuses on detecting when watermarked outputs are reused as training data, e.g., via "radioactive" text that remains detectable even after distillation (Sander et al., 2024; Gu et al., 2024). Unlike watermarking, our fingerprinting approach is non-invasive, introduces no output bias, and remains effective in black-box settings.

LLM fingerprinting Beyond watermarking, several works explore model fingerprinting and provenance. For vision and speech models, fingerprinting often relies on backdoors or adversarial trig-

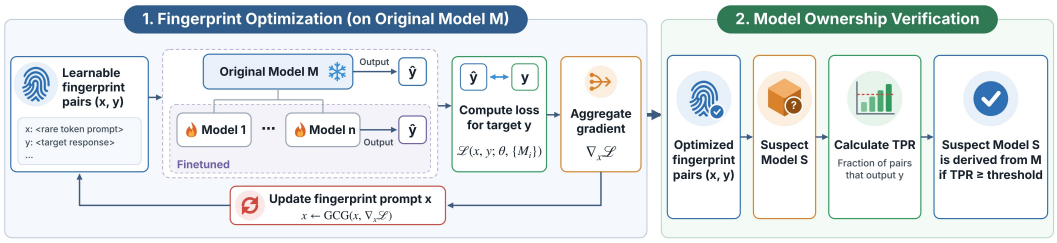


Figure 1: **Overview of RAFP.** (1) **Fingerprint Optimization**: rare prompt–response pairs are optimized jointly over the base model and adapted variants to improve persistence. (2) **Model Ownership Verification**: optimized fingerprints are queried against a suspect model and verified using TPR.

System	A chat between a curious human and an artificial intelligence assistant...
Prompt x	полтельства otraфacticde回ultyœuvreshgreg昔者 historical anc 同eles話бай user]
Response y	역타非학들話仮

Figure 2: Example of a rare-region fingerprint generated by RAFP. The optimized fingerprint prompt x triggers the fingerprint response y.

gers (Cao et al., 2021; Chen et al., 2022; Lukas et al., 2021). For LLMs, ProFLingo (Jin et al., 2024) uses fixed question–answer pairs as behavioral fingerprints, whereas Instructional Fingerprinting (IF) (Xu et al., 2024) requires an additional fingerprint adapter trained to inject model-specific behaviors. Recent methods further improve robustness by explicitly injecting ownership signals. MergePrint (Yamabe et al., 2025) optimizes a fingerprint by modifying the owner model so that it survives merging, whereas RAFP searches for naturally persistent prompt-response behaviors without updating model parameters. EverTracer (Xu et al., 2025a) embeds memorization-based signals for ownership verification, and CTCC (Xu et al., 2025b) exploits cross-turn contextual correlations to improve robustness and stealthiness. Other approaches infer model lineage from training data or output similarity (Hao et al., 2024).

3 Preliminaries

In this section, we introduce the threat model, problem formulation, and fingerprints' desired properties in RAFP.

Threat Model We consider an adversary who aims to deploy a model derived from a proprietary LLM for economic gain while avoiding detection of its origin. We assume the adversary is limited to economically viable modifications. Under this setting, we consider the following *four* limitations:

1. **Limited resources.** The adversary cannot afford to train a model from scratch or perform large-scale retraining comparable to the original pretraining process.
2. **Model training integrity.** The adversary cannot tamper with the pre-training process of the original model, including training data, training algorithm, or infrastructure.
3. **Value-preserving adaptation.** The adversary is restricted to common *post-training.* modifications that preserve model quality, such as supervised finetuning (SFT), low-rank adaptation (LoRA), system prompt engineering, quantization, and tuning of inference hyper-parameters (e.g., generation temperatures). While arbitrary modifications (e.g., random re-initialization) could remove fingerprints, such changes would destroy the model's utility and are therefore unlikely to be adopted by the adversary.
4. **Evasion capability.** The adversary may attempt to evade fingerprint verification through targeted input filtering or output manipulation. For example, fingerprint queries may be detected using perplexity-based filtering or sanitized through response rewriting. However, such defenses introduce a trade-off between suppressing fingerprint behaviors and preserving normal model utility, since aggressive filtering or rewriting may also affect legitimate downstream generation quality.

Problem Formulation and Desired Properties Given a base model $\mathcal{M}_\theta$ and a suspect model $\mathcal{M}_{\theta_u}$, our goal is to determine whether $\mathcal{M}_{\theta_u}$ is derived from $\mathcal{M}_\theta$ using only black-box access. We

approach this problem by constructing a set of fingerprint pairs $F = \{(x_i, y_i)\}$ such that models derived from $\mathcal{M}_\theta$ exhibit consistent responses on these prompts, while unrelated models do not. Our model identification scheme involves two functions: (1) a function, $\mathrm{gen}(\mathcal{M}_\theta) = F$, that generates a fingerprint for model $\mathcal{M}_\theta$; and (2) a function, $\mathrm{verify}(F, \mathcal{M}_\theta)$, that decides whether or not F is a fingerprint of model $\mathcal{M}_\theta$. The scheme should satisfy the following *six* desired properties:

1. **Effectiveness**. Model identification using $\mathrm{verify}(F, \mathcal{M}_\theta)$ should succeed with high probability. (See Table 1 (a.))
2. **Robustness**. Model identification should be robust under common model changes and work for all models in a lineage with high probability, that is, $\forall \mathcal{M}_{\theta_u} : \mathrm{verify}(F, \mathcal{M}_\theta) \implies \mathrm{verify}(F, \mathcal{M}_{\theta_u})$. (See Table 1 (b.), Table 3 , Fig. 3, Fig. 4)
3. **Uniqueness**. Fingerprints from different model lineages should be distinguishable with high probability, that is, $\mathrm{verify}(F, \mathcal{M}_\theta) \implies \neg\, \mathrm{verify}(F, \mathcal{M}_\omega)$ when $\theta \neq \omega$. (See Table 2)
4. **Black-box.** To facilitate investigation of models deployed in products, $\mathrm{verify}(F, \mathcal{M}_{\theta_u})$ can query $\mathcal{M}_{\theta_u}$ but not access parameters θ_u.
5. **Harmlessness.** Invoking $\mathrm{gen}(\mathcal{M}_\theta)$ should not degrade the quality of the model. (See A.7)
6. **Secrecy.** Without access to $\mathcal{M}_\theta$ or its adaptations, fingerprints should be difficult for an attacker to discover or reproduce. (See. A.8)

4 Robust Fingerprinting in RAFP

We now describe RAFP, including its fingerprint construction and ownership verification protocol, the practical optimization strategy used to obtain rare-region fingerprints, and the theoretical analysis explaining their persistence under downstream adaptation. RAFP consists of two stages: (1) fingerprint generation, which searches for rare prompts that reliably induce stable responses, and (2) ownership verification, which determines whether a suspect model preserves these fingerprint behaviors. (See Fig. 1)

4.1 Fingerprint Generation

Our fingerprints consist of a prompt–response pair (x, y), where the fingerprint prompt x reliably induces the fingerprint response y across models derived from the same lineage. Fingerprint generation does not modify the model parameters; instead, it searches for rare prompt–response behaviors already supported by the pretrained model $\mathcal{M}_\theta$. Specifically, fingerprint generation consists of three steps:

1. **Rare prompt initialization.** We initialize a candidate prompt x' of length $l = 32$ tokens. The first few tokens are sampled uniformly at random, and the remaining tokens are generated using bottom-k sampling from $\mathcal{M}_\theta$ with $k = 2000$. This biases the prompt toward a low-probability region of the pretrained model distribution.
2. **Rare response initialization.** Conditioned on the initialized prompt x' and system prompt h, we sample a target response y of length 9 tokens using bottom-k sampling from $\mathcal{M}_\theta(\cdot \mid h, x')$ with $k = 2000$. This yields an initial rare prompt–response pair (x', y), which serves as the starting fingerprint candidate.
3. **Rare-region prompt optimization.** Keeping the target response y fixed, we initialize the optimization at $x^{(0)} = x'$ and iteratively update the prompt to increase the likelihood that $\mathcal{M}_\theta$ reproduces y, while constraining all intermediate prompts to remain in the rare region:
$$x^{(0)} = x', \qquad x^{(T)} = \arg\max_{x \in \mathcal{R}_\tau} \log p_{\mathcal{M}_\theta}(y \mid h, x),$$
where $\mathcal{R}_\tau$ denotes the low-density prompt region defined by the rarity threshold τ. We perform the optimization using Greedy Coordinate Gradient (GCG) (Zou et al., 2023), starting from $x^{(0)} = x'$ and restricting candidate token replacements to preserve the rare-region constraint. The final fingerprint prompt is $x = x^{(T)}$.

4.2 Ownership Verification

Prior to model deployment, the developer generates a collection of fingerprints and publishes cryptographic commitments (e.g., SHA-256 hashes) on a public timestamped platform. Given a suspect

model $\mathcal{M}_{\theta_u}$, the developer probes the model using unrevealed fingerprint prompts and evaluates whether the corresponding fingerprint responses are reproduced. If evidence of model theft is observed, the developer can reveal a previously committed fingerprint and verify ownership by demonstrating that $\mathcal{M}_{\theta_u}(x) = y$. To reduce accidental matches, verification additionally evaluates fingerprints against unrelated model lineages and confirms rejection.

4.3 Practical Optimization Strategy

Multi-task prompt optimization. To improve robustness, we jointly optimize fingerprints across multiple adapted models and system prompts:

$$\max_{x} \sum_{\mathcal{M}' \in \{\mathcal{M}'_1, \ldots, \mathcal{M}'_M\}} \sum_{h \in \mathcal{H}} \log p_{\mathcal{M}'}(y \mid h, x),$$

where $\mathcal{H}$ denotes a set of system prompts and $\{\mathcal{M}'_1, \ldots, \mathcal{M}'_M\}$ are variants of the base model. This optimization ensures fingerprints remain stable across natural prompt and model variations.

Multi-trial optimization. Single-run GCG optimization often converges to brittle local optima that do not transfer reliably across downstream model adaptations. We therefore adopt a multi-trial strategy that requires successful optimization across multiple independent runs before accepting a fingerprint candidate. In our experiments, we use $n = 20$ optimization trials.

In Figure 2, we illustrate an example fingerprint pair generated by RAFP. The same fingerprint response is preserved even when the system prompt changes, demonstrating robustness across prompting templates.

4.4 Theoretical Analysis of Rare-Region Fingerprint Stability

We now provide a theoretical explanation for why rare-region fingerprints remain stable under downstream finetuning. The key intuition is that downstream adaptation is dominated by common high-density language patterns, while rare prompts receive little direct optimization signal and weak aggregate gradient alignment during training. As a result, the likelihood of rare fingerprint responses changes only slightly after finetuning.

Finetuning formulation. Let $p_\theta(y \mid x)$ denote the conditional response distribution of the pretrained model $\mathcal{M}_\theta$, where x is a prompt and y is a generated response. Let $q(x, y)$ denote the downstream finetuning distribution over prompt-response pairs, parameters ϑ in a local neighborhood of θ. The finetuned model parameters θ' are obtained by minimizing the downstream objective:

$$\theta' = \arg\min_{\vartheta} \mathbb{E}_{q(x,y)} \big[-\log p_\vartheta(y \mid x) \big].$$

Rare-region prompts. For a prompt sequence $x = \{x_1, \ldots, x_{|x|}\}$, we define a rarity score $s_\theta(x)$ using normalized autoregressive log-likelihood under the pretrained model θ:

$$s_\theta(x) = \frac{1}{|x|} \sum_{t=1}^{|x|} \log p_\theta(x_t \mid x_{<t}).$$

Lower values of $s_\theta(x)$ indicate prompts that are less likely under the pretrained model. We define the rare-prompt region and its downstream probability mass:

$$\mathcal{R}_\tau = \{x : s_\theta(x) < \tau\}, \qquad \rho(\tau) = \mathbb{P}_{x \sim q_X}(x \in \mathcal{R}_\tau),$$

where τ is a small threshold, and $q_X(x)$ is the marginal prompt distribution induced by $q(x, y)$.

Assumption 1 (Rare prompts are scarce). We assume that rare-region prompts occupy small probability mass under downstream finetuning: $\rho(\tau) \ll 1$, and that $\rho(\tau)$ decreases as τ decreases.

Assumption 2 (Bounded gradients). For all prompt-response pairs (x, y) and parameters ϑ in a local neighborhood of θ, the log-likelihood gradient is upper-bounded by the constant L: $\big\|\nabla_\vartheta \log p_\vartheta(y \mid x)\big\| \le L$. In our proof, we measure the stability of fingerprint pairs $(\hat{x}, \hat{y})$ by the change in log-likelihood assigned to the fingerprint response before and after finetuning:

$$\Delta(\hat{x}, \hat{y}) = \left| \log p_{\theta'}(\hat{y} \mid \hat{x}) - \log p_\theta(\hat{y} \mid \hat{x}) \right|.$$

Lemma 1 (Rare-region likelihood invariance). *Let $(\hat{x}, \hat{y})$ be a fingerprint pair with $\hat{x} \in \mathcal{R}_\tau$. Suppose θ' is obtained from one gradient update on n downstream samples drawn from $q(x, y)$ with learning rate η. Then, under Assumptions 1 and 2, the following holds with probability at least $1 - \delta$:*

$$\Delta(\hat{x}, \hat{y}) \leq \ \eta L^2 \left[\rho(\tau) + (1 - \rho(\tau))\sqrt{2\ln(2/\delta)/n}\right] + O(\eta^2).$$

Proof. Denote $\mathbf{g}_{x,y} = \nabla_\theta \log p_\theta(y \mid x)$ as the conditional probability gradient. Gradient descent on the negative log-likelihood with learning rate η on a downstream dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ with $(x_i, y_i) \overset{\text{i.i.d.}}{\sim} q(x, y)$ updates θ' by the empirical mean of the conditional gradients, denoted $\bar{\mathbf{g}}_\mathcal{D}$:

$$\theta' = \theta + \eta\, \bar{\mathbf{g}}_\mathcal{D}, \qquad \bar{\mathbf{g}}_\mathcal{D} = \frac{1}{n}\sum_{i=1}^n \mathbf{g}_{x_i,y_i},$$

which, for bounded gradients, implies $O(\|\theta' - \theta\|) = O(\eta)$. A first-order Taylor expansion in θ gives

$$\begin{aligned}
\log p_{\theta'}(\hat{y} \mid \hat{x}) &= \log p_\theta(\hat{y} \mid \hat{x}) + \mathbf{g}_{\hat{x},\hat{y}}^\top(\theta' - \theta) + O(\|\theta' - \theta\|^2), && \text{(Taylor expansion)} \\
\Delta(\hat{x}, \hat{y}) &\leq \left|\mathbf{g}_{\hat{x},\hat{y}}^\top(\theta' - \theta)\right| + O(\|\theta' - \theta\|^2), && \text{(Triangle inequality)} \\
&= \eta\, |\langle \mathbf{g}_{\hat{x},\hat{y}}, \bar{\mathbf{g}}_\mathcal{D}\rangle| + O(\eta^2), && \text{(Substitute } \theta' - \theta = \eta\bar{\mathbf{g}}_\mathcal{D})
\end{aligned}$$

To bound $|\langle \mathbf{g}_{\hat{x},\hat{y}}, \bar{\mathbf{g}}_\mathcal{D}\rangle|$, we decompose the gradient projection into rare and non-rare regions:

$$|\langle \mathbf{g}_{\hat{x},\hat{y}}, \bar{\mathbf{g}}_\mathcal{D}\rangle| \leq \underbrace{\hat{\mathbb{P}}(x \in \mathcal{R}_\tau)\, |\langle \mathbf{g}_{\hat{x},\hat{y}}, \bar{\mathbf{g}}_{\mathcal{R}_\tau}\rangle|}_{\text{rare region term}} + \underbrace{\hat{\mathbb{P}}(x \in \mathcal{D} \setminus R_\tau)\, |\langle \mathbf{g}_{\hat{x},\hat{y}}, \bar{\mathbf{g}}_{\mathcal{D}\setminus R_\tau}\rangle|}_{\text{non-rare region term}}, \quad \text{(Triangle inequality)}$$

By Assumption 1 and the Cauchy–Schwarz inequality, the rare-region term is bounded by $\hat{\mathbb{P}}(x \in \mathcal{R}_\tau)\, |\langle \mathbf{g}_{\hat{x},\hat{y}}, \bar{\mathbf{g}}_{\mathcal{R}_\tau}\rangle| \leq L^2\rho(\tau)$. By Assumption 2 and Hoeffding's inequality, the non-rare term is bounded w.p. at least $1 - \delta$, by $\hat{\mathbb{P}}(x \in D \setminus \mathcal{R}_\tau)\, |\langle \mathbf{g}_{\hat{x},\hat{y}}, \bar{\mathbf{g}}_{\mathcal{D}\setminus\mathcal{R}_\tau}\rangle| \leq (1 - \rho(\tau))\, L^2\sqrt{2\ln(2/\delta)/n}$.

Combining the two gives the bound in Lemma 1. See Appdx. A.1 for the detailed proof.

Following Lemma 1, we prove the decoding stability by showing that the top-ranked fingerprint is preserved under finetuning with a margin condition on prompt rarity.

Corollary 1 (Decoding stability). *Define the decoding margin as*

$$\gamma(\hat{x}) = \log p_\theta(\hat{y} \mid \hat{x}) - \log p_\theta(y^* \mid \hat{x})$$

where $y^* = \arg\max_{y \neq \hat{y}} \log p_\theta(y \mid \hat{x})$ is the strongest competing response, i.e., the response with the second-highest probability. Let B be the upper bound from Lemma 1. If $\gamma(\hat{x}) \geq 2B$, then with probability at least $1 - 2\delta$, $\hat{y}$ remains the top-ranked response under θ'. This marginal condition holds whenever

$$\rho(\tau) \leq \frac{\nu - \epsilon}{1 - \epsilon}, \qquad \nu = \frac{\gamma(\hat{x})}{2\eta L^2}, \qquad \epsilon = \sqrt{\frac{2\ln(2/\delta)}{n}}$$

Proof. By Lemma 1, $\Delta(\hat{x}, \hat{y}) \leq B$ and $\Delta(\hat{x}, y^*) \leq B$ jointly with probability at least $1 - 2\delta$. If $\gamma(\hat{x}) \geq 2B$, then $\Delta(\hat{x}, \hat{y}) + \Delta(\hat{x}, y^*) \leq 2B \leq \gamma(\hat{x})$. Therefore, $\hat{y}$ remains the top-ranked response under θ', preserving decoding stability. Solving $\gamma(\hat{x}) \geq 2B$ for $\rho(\tau)$ gives the stated condition, where ν and ϵ are constants following directly from substituting B above.

This result directly connects decoding stability to prompt rarity and provides a verifiable criterion for selecting the rare region.

Corollary 2 (Multi-step update stability). *Under the conditions of Corollary 1, if $\gamma(\hat{x}) \geq 2tB$, then with probability at least $1 - 2t\delta$, $\hat{y}$ remains the top-ranked response after t successive generations of finetuning. Equivalently, the number of generations $\hat{y}$ survives satisfies*

$$t \leq \frac{\gamma(\hat{x})}{2B}.$$

Proof. By Lemma 1, at each of the t finetuning steps, with probability at least $1 - 2\delta$, $\Delta(\hat{x}, \hat{y}) \leq B$ and $\Delta(\hat{x}, y^*) \leq B$. By a union bound over the t steps and the triangle inequality summing the per-step changes, with probability at least $1 - 2t\delta$,

$$\Delta(\hat{x}, \hat{y}) + \Delta(\hat{x}, y^*) \leq 2tB.$$

If $\gamma(\hat{x}) \geq 2tB$, then $\gamma(\hat{x}) \geq \Delta(\hat{x}, \hat{y}) + \Delta(\hat{x}, y^*)$, so $\hat{y}$ remains the top-ranked response after t steps, with probability at least $1 - 2t\delta$. The upper bound for finetuning steps is therefore $t \leq \gamma(\hat{x})/2B$.

Table 1: **Fingerprint evaluation.** (a) Evaluation on *Effectiveness*. True positive rate of model identification using fingerprints from four different pre-training LLMs (higher is better). Rates are computed across 10 fingerprints. (b) Evaluation on *Robustness*. True positive rate of model identification of finetuned models using fingerprints from four different pre-training LLMs (higher is better). Rates are computed across five different finetuned versions of the models and 10 fingerprints. See Appdx. A.2 for detailed results on each downstream finetuned version of the models.

	Llama 2 7B	Llama 2 13B	Llama 3 8B	Mistral 7B
IF-SFT	100%	100%	100%	100%
IF-Emb	60%	70%	100%	70%
GCG	25%	40%	40%	80%
RAFP (Ours)	100%	**100%**	**100%**	**100%**

(a) Effectiveness

	Llama 2 7B	Llama 2 13B	Llama 3 8B	Mistral 7B
IF-SFT	65%	50%	66.67%	68.33%
IF-Emb	50%	45%	58.33%	53.33%
GCG	23.33%	33%	30%	63.33%
RAFP (Ours)	**100%**	**93.33%**	**100%**	**100%**

(b) Robustness

5 EXPERIMENTS

5.1 EXPERIMENTAL SETUP

Models. We use RAFP to identify four pre-trained large language models with decoder-only architectures: namely, Llama 2 7B, Llama 2 13B (Touvron et al., 2023), Llama 3 8B (Grattafiori et al., 2024), and Mistral 7B (Jiang et al., 2023). Each model has its own tokenizer and token vocabulary with different vocabulary sizes. For example, Llama 2 uses a tokenizer with a vocabulary of 32K tokens whereas Llama 3's tokenizer contains 128K tokens. We use RAFP to produce fingerprints for each of these models, and use those to perform model identification.

Downstream Finetuning Tasks. To evaluate robustness of RAFP to model changes, we perform multiple finetuning experiments of the four models study. To this end, we finetune the models on five different datasets using a standard supervised finetuning (SFT) procedure. Specifically, we perform finetuning on three instruction datasets (15K NI (Mishra et al., 2022), 15K Dolly (Conover et al., 2023), and Codegen from CodeAlpaca* (Luo et al., 2024). Two conversational datasets include OpenAssistant-Oasst1 (Köpf et al., 2023) and ShareGPT (2023)†. For all these finetuning datasets, we follow the SFT recipe of Taori et al. (2023), training for 3 epochs on each dataset. Appendix A.2 shows the details of finetuned tasks and training setup.

For the Llama 2 7B model, we also perform model-identification experiments on nine finetuned versions of that model that were published on Huggingface: (1) the original Llama 2 7B chat model; (2-4) finetunes of the chat model for Chinese ‡, Japanese (Sasaki et al., 2023), and Spanish §; (5) the Meditron 7B medical LLM (Chen et al., 2023); and (6) Orca-2 7B (Mitra et al., 2023); (7-9) Adapted version by DPO. For each finetuned model, we adopt the system prompt suggested by the developers of the finetuned model, thus measuring robustness to system prompt changes as well.

Baselines. *(1.) Watermarking-based methods.* We include Instructional Fingerprinting with full supervised finetuning **(IF-SFT)** (Xu et al., 2024), which finetunes the model on a dataset containing fingerprint pairs, and **(IF-Emb)**, which restricts fingerprint-induced updates to the embedding layer (Kurita et al., 2020; Yang et al., 2021). Both methods require modifying the protected model using fingerprint-specific training data. *(2.) GCG-based fingerprinting methods.* We consider **GCG**, which optimizes a random trigger string to produce a predefined target response from a single model. We follow the default optimization setup with 500 training epochs and stop at the first successful trigger. We also include **GCG-QA** (Jin et al., 2024), which constructs question–answer fingerprint pairs through GCG-based prompt optimization. GCG-QA is included in the main comparison in Table 2. *(3.) Variants of* RAFP. We compare simplified variants of our method: **Base-only**, which performs fingerprint optimization using only the base model; **Base + 1 task**, which jointly optimizes fingerprints using the base model and one downstream-adapted model; and **Base + 2 tasks**, which further incorporates a second adapted model. These variants are used only to study the effect of multi-task fingerprint optimization and are not treated as external baselines.

*CodeAlpaca:https://huggingface.co/datasets/theblackcat102/evol-codealpaca-v1

†ShareGPT:https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered

‡Llama-Chinese:https://github.com/LlamaFamily/Llama-Chinese

§Llama-Spanish:https://huggingface.co/clibrain/Llama-2-7b-ft-instruct-es

Table 2: Evaluation on *Uniqueness*. TPR is reported on nine relevant suspect models and FPR on five unrelated models. For GCG-QA (Jin et al., 2024), we evaluate 10 QA fingerprint pairs. For relevant models, higher TPR is better; for unrelated models, ✓indicates 0% FPR. In Appdx. A.3, we include the details of every suspect model.

Suspect Model / Method	GCG-QA	Base-only	RAFP (Ours)
meta-llama/Llama-2-7b-chat-hf	✓ (20%)	✓ (40%)	✓ (60%)
FlagAlpha/Llama2-Chinese-7b-Chat	✓ (40%)	✗ (0%)	✓ (80%)
elyza/ELYZA-japanese-Llama-2-7b	✓ (50%)	✗ (0%)	✓ (100%)
clibrain/Llama-2-7b-ft-instruct-es	✓ (30%)	✗ (0%)	✓ (60%)
epfl-llm/meditron-7b	✓ (60%)	✗ (0%)	✓ (20%)
microsoft/Orca-2-7b	✓ (30%)	✗ (0%)	✓ (40%)
renyiyu/llama-2-7b-ppo-lora-v0.1	✓ (30%)	✓ (20%)	✓ (40%)
Minbyul/llama2-7b-dpo-full-sft-wo-live-qa	✓ (100%)	✓ (80%)	✓ (100%)
allenai/tulu-2-dpo-7b	✓ (40%)	✓ (20%)	✓ (40%)
mistralai/mistral-7B-v0.3	✗ (10%)	✓ (0%)	✓ (0%)
OpenAssistant/oasst-sft-4-pythia-12b	✗ (10%)	✓ (0%)	✓ (0%)
tiiuae/falcon-7b-instruct	✓ (0%)	✓ (0%)	✓ (0%)
TheBloke/guanaco-7B-HF	✓ (0%)	✓ (0%)	✓ (0%)
Alchan/mpt-7b-chat	✓ (0%)	✓ (0%)	✓ (0%)

Success measure. Following Xu et al. (2024), we measure the quality of model identification in terms of the true positive rate (TPR). The TPR is defined as the rate of correct model identifications (that is, the true positive rate) across a set of N fingerprints generated using that model.

5.2 Experimental Results

• **Effectiveness.** We evaluate four variants of GCG-based fingerprinting and two IF-based baselines in terms of effectiveness and persistence. Table 1 (a.) reports the true positive rate (TPR) on four base models: Llama 2 7B, Llama 2 13B, Llama 3 8B, and Mistral 7B. Our method consistently achieves 100% TPR across all four models, outperforming other GCG baselines and remaining competitive with IF-SFT. Compared with IF-Emb, which only updates embedding parameters, our method achieves stronger memorization without modifying model weights. Unlike IF-SFT, which may degrade downstream utility, our method preserves standard performance while maintaining 100% effectiveness.

• **Robustness.** Table 1 (b.) compares robustness under downstream finetuning. For each method, we average TPR over five downstream models derived from each base model using Alpaca-style finetuning settings. Our method achieves 92∼100% TPR across all settings and consistently outperforms existing baselines. In contrast, IF-SFT and IF-Emb suffer 10–35% TPR degradation after finetuning. Increasing the number of optimization tasks further improves robustness, demonstrating the effectiveness of multi-task fingerprint optimization. Compared with standard GCG, our method improves TPR by 37∼80%, validating the importance of rare-region search and multi-trial optimization. Furthermore, Fig. 3 analyzes the effect of multi-task optimization in RAFP. Increasing the number of optimization tasks from zero to two consistently improves TPR across all four models under both SFT and LoRA finetuning settings. Additional results of downstream tuning are provided in Appdx. A.2.

• **Uniqueness.** We further evaluate fingerprint verification in a fully black-box setting where suspect models are deployed as APIs or released on HuggingFace. Table 2 reports TPR on nine models finetuned from Llama 2 7B and FPR on five unrelated models. For each suspect model, we use its repository-recommended prompt template and default decoding configuration. Compared with GCG-QA (Jin et al., 2024) and the base-only variant, RAFP successfully fingerprints all relevant models while rejecting unrelated ones (0% FPR).

6 Ablation Studies

• **Robustness against Perplexity Filtering.** To evaluate whether RAFP can evade API-side perplexity filtering, we also optimize fingerprints with a joint objective combining fingerprint consistency and prompt perplexity loss (PPL) regularization: $\mathcal{L} = -\log p_\theta(y \mid x) + \lambda \,\mathrm{PPL}(x)$, where λ is the regularization weight. In our experiment, we set the filter threshold (τ) for Llama 2 7B using the 95th percentile perplexity of 1K natural prompts sampled from ShareGPT dataset. This ensures that the filter accepts the realistic inputs while still detecting extremely low-likelihood prompts. The filter rule

Table 3: Robustness analyses of RAFP. (a) Perplexity-aware optimization substantially reduces rejection under a perplexity-based filter while maintaining high fingerprint TPR. (b) RAFP remains robust across different prompt templates on both the base and ShareGPT-finetuned models, with smaller degradation than IF-SFT and GCG.

(a) Perplexity-aware optimization

Method	λ	Reject. Rate	TPR	
			Base	SFT
RAFP w/o PPL	0	100%	100%	100%
RAFP w/ PPL	**0.1**	**20%**	**90%**	**80%**
	0.3	10%	90%	60%
	0.5	10%	80%	60%

(b) Prompt-template robustness

Model	Method	Llama 2	Vicuna	Alpaca	Zero Shot	ChatGPT
Llama 2 7B (Base)	IF-SFT	62.5%	**100%**	75%	75%	50%
	GCG	**100%**	80%	**100%**	60%	20%
	RAFP	**100%**	**100%**	90%	**80%**	**100%**
Finetuned (ShareGPT)	IF-SFT	75%	**100%**	**100%**	62.5%	50%
	GCG	**100%**	50%	90%	60%	10%
	RAFP	**100%**	**100%**	90%	**70%**	**100%**

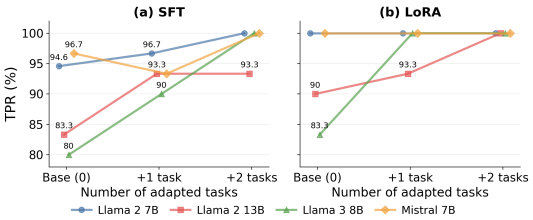


Figure 3: Effect of multi-task fingerprint optimization on downstream robustness. We report TPR under SFT and LoRA adaptation as the number of adapted tasks used during fingerprint optimization increases from Base to +1 and +2 tasks.

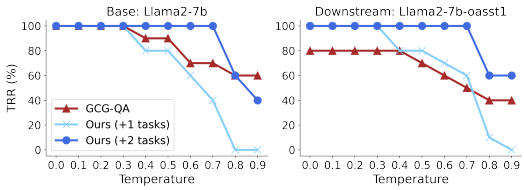


Figure 4: Robustness under different decoding temperatures. We compare GCG-QA with RAFP variants optimized using one and two adapted tasks. Multi-task fingerprint optimization improves robustness as temperature increases.

is reject$(x) = \text{PPL}(x) > \tau$. In Table 3 (a.), we show RAFP under perplexity-aware optimization substantially reduces rejection rates while retaining moderate fingerprint effectiveness on both the base and SFT-adapted models. We grid search λ from 0.1 to 0.5 and observe that $\lambda = 0.1$ lowers the reject rate without severely degrading TPR.

• **Analysis of Prompt Templates.** We evaluate the robustness of RAFP under different prompt templates on Llama 2 7B and its downstream models. Fingerprints are optimized using the default Llama 2 and Vicuna templates, and evaluated on additional templates including Alpaca, Zero-Shot, and ChatGPT (Zheng et al., 2023). As shown in Table 3 (b.), RAFP maintains 70∼100% TPR across both base and finetuned models under prompt-template variations. Compared with GCG and IF-SFT, our method shows the smallest TPR degradation (10∼30%), demonstrating stronger robustness to prompt-format changes. Detailed prompt templates are provided in Appdx. Table 7.

• **Analysis of Generation Hyperparameters.** We evaluate robustness under different temperatures ranging from 0 to 0.9. In Fig. 4, RAFP maintains nearly 100% TPR from temperature 0.1 to 0.7, with only slight degradation above 0.8. In contrast, GCG-QA begins to degrade at a temperature of 0.3. Additional robustness analyses on multi-stage fingerprints, quantization variants, and adversarial attack analysis are provided in Appdx. A.5.1, A.5.2, A.6.1, and A.6.2.

7 Conclusion

In this work, we propose RAFP, a method for identifying LLM lineages via rare-region fingerprints. Our key insight is that rare prompt regions receive weak optimization signal during downstream finetuning, allowing rare prompt-response behaviors to remain stable across model adaptations. We further provide a theoretical analysis explaining the invariance of rare-region fingerprints under finetuning. Experiments across multiple LLM families and downstream adaptations demonstrate that RAFP achieves strong fingerprint persistence and substantially outperforms prior fingerprinting baselines in black-box settings. Future work may explore stronger adaptive adversaries and fingerprint LLMs with web-scale model training.

AI USE STATEMENT

In this work, we used generative AI tools for writing polishing, grammatical checks, and LaTeX formatting. We did not use generative AI tools to generate research ideas, math proofs, code, or experimental results, and other required disclosure categories are not applicable to this work. We have reviewed all AI-assisted work. We checked the LLM-editing content of this work, including the main paper text. We take responsibility for the final content of this work produced with the aid of generative AI.

ETHICS STATEMENT

Technically, RAFP serves as a robust method to identify and track the origin of specific models involved in generating particular texts. This capability enhances models' transparency, aids in attributing content to its source, and can deter the misuse of LLMs. Ethically, fingerprinting LLM might raise concerns about surveillance and privacy. It requires carefully weighing the need for security and accountability against the risk of violating users' privacy rights. By allowing the identification of downstream models' origins, there is a risk that this capability could be exploited for surveillance and monitoring, potentially resulting in limiting the right of users on using public models or the suppression of free speech. Future societal consequences of deploying fingerprints with LLMs hinge on regulatory and governance frameworks. People need to have clear policies that define acceptable uses of LLMs and further recognize which form of fingerprinting can be used for model verification. We believe this technology can foster a more trustworthy digital communication environment, reducing the prevalence of fake news and enhancing the credibility of LLM-generated content.

REPRODUCIBILITY STATEMENT

For empirical results, we provide implementation detail in main paper Sec. 5, including novel models, baselines, and finetuning setup. In Appdx A.2, we provide details of downstream tasks for training models. For theoretical results, a complete proof of the claims and clear explanations of Lemma 1, rare-region likelihood invariance, is included in the Appdx A.1.

REFERENCES

Scott Aaronson. "reform" ai alignment with scott aaronson. *AXRP - the AI X-risk Research Podcast*, 2023. URL https://axrp.net/episode/2023/04/11/episode-20-reform-ai-alignment-scott-aaronson.html.

Gabriel Alon and Michael Kamfonas. Detecting language model attacks with perplexity, 2023. URL https://arxiv.org/abs/2308.14132.

Bochuan Cao, Yuanpu Cao, Lu Lin, and Jinghui Chen. Defending against alignment-breaking attacks via robustly aligned LLM. In Lun-Wei Ku, Andre Martins, and Vivek Srikumar (eds.), *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10542–10560, Bangkok, Thailand, August 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.acl-long.568. URL https://aclanthology.org/2024.acl-long.568.

Xiaoyu Cao, Jinyuan Jia, and Neil Zhenqiang Gong. Ipguard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary. In *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*, ASIA CCS '21, pp. 14–25, New York, NY, USA, 2021. Association for Computing Machinery. ISBN 9781450382878. doi: 10.1145/3433210.3437526. URL https://doi.org/10.1145/3433210.3437526.

Nicholas Carlini, Matthew Jagielski, Christopher A. Choquette-Choo, Daniel Paleka, Will Pearce, Hyrum Anderson, Andreas Terzis, Kurt Thomas, and Florian Tramer. Poisoning Web-Scale Training Datasets is Practical . In *2024 IEEE Symposium on Security and Privacy (SP)*, pp. 407–425, Los Alamitos, CA, USA, May 2024. IEEE Computer Society. doi: 10.1109/SP54263.2024.00179. URL https://doi.ieeecomputersociety.org/10.1109/SP54263.2024.00179.

Jialuo Chen, Jingyi Wang, Tinglan Peng, Youcheng Sun, Peng Cheng, Shouling Ji, Xingjun Ma, Bo Li, and Dawn Song. Copy, right? a testing framework for copyright protection of deep learning models. In *2022 IEEE Symposium on Security and Privacy (SP)*, pp. 824–841. IEEE, 2022. doi: 10.1109/SP46214.2022.9833747. URL https://ieeexplore.ieee.org/document/9833747.

Zeming Chen, Alejandro Hernández-Cano, Angelika Romanou, Antoine Bonnet, Kyle Matoba, Francesco Salvi, Matteo Pagliardini, Simin Fan, Andreas Köpf, Amirkeivan Mohtashami, Alexandre Sallinen, Alireza Sakhaeirad, Vinitra Swamy, Igor Krawczuk, Deniz Bayazit, Axel Marmet, Syrielle Montariol, Mary-Anne Hartley, Martin Jaggi, and Antoine Bosselut. Meditron-70b: Scaling medical pretraining for large language models, 2023. URL https://github.com/epfLLM/meditron.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In Shipra Agrawal and Aaron Roth (eds.), *Proceedings of the 37th Conference on Learning Theory (COLT)*, volume 247 of *Proceedings of Machine Learning Research*, pp. 1125–1139. PMLR, 2024. URL https://proceedings.mlr.press/v247/christ24a.html.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. Free dolly: Introducing the world's first truly open instruction-tuned llm, 2023. URL https://www.databricks.com/blog/2023/04/12/dolly-first-open-commercially-viable-instruction-tuned-llm.

Mark Gabel, Junfeng Yang, Yuan Yu, Moises Goldszmidt, and Zhendong Su. Scalable and systematic detection of buggy inconsistencies in source code. *SIGPLAN Not.*, 45(10):175–190, October 2010. ISSN 0362-1340. doi: 10.1145/1932682.1869475. URL https://doi.org/10.1145/1932682.1869475.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, et al. The llama 3 herd of models, 2024. URL https://arxiv.org/abs/2407.21783.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. In *12th International Conference on Learning Representations, ICLR 2024*, 2024. URL https://arxiv.org/abs/2312.04469.

Chenxi Gu, Chengsong Huang, Xiaoqing Zheng, Kai-Wei Chang, and Cho-Jui Hsieh. Watermarking pre-trained language models with backdooring, 2022. URL https://arxiv.org/abs/2210.07543.

Wei Hao, Daniel Mendoza, Rafael Mendes, Deepak Narayanan, Amar Phanishayee, Asaf Cidon, and Junfeng Yang. Mgit: A model versioning and management system. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp (eds.), *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pp. 17597–17615. PMLR, 2024. URL https://proceedings.mlr.press/v235/hao24c.html.

Zhengmian Hu, Lichang Chen, Xidong Wu, Yihan Wu, Hongyang Zhang, and Heng Huang. Unbiased watermark for large language models. In *12th International Conference on Learning Representations, ICLR 2024*, 2024. URL https://openreview.net/forum?id=uWVC5FVidc.

Albert Q. Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b, 2023. URL https://arxiv.org/abs/2310.06825.

Heng Jin, Chaoyu Zhang, Shanghao Shi, Wenjing Lou, and Y. Thomas Hou. Proflingo: A fingerprinting-based intellectual property protection scheme for large language models. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pp. 1–9. IEEE, 2024. doi: 10.1109/CNS62487.2024.10735575. URL https://ieeexplore.ieee.org/document/10735575.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett (eds.), *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pp. 17061–17084. PMLR, 2023. URL https://proceedings.mlr.press/v202/kirchenbauer23a.html.

Andreas Köpf, Yannic Kilcher, Dimitri von Rütte, Sotiris Anagnostidis, Zhi-Rui Tam, Keith Stevens, Abdullah Barhoum, Nguyen Minh Duc, Oliver Stanley, Richárd Nagyfi, Shahul ES, Sameer Suri, David Glushkov, Arnav Dantuluri, Andrew Maguire, Christoph Schuhmann, Huu Nguyen, and Alexander Mattick. Openassistant conversations – democratizing large language model alignment. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/949f0f8f32267d297c2d4e3ee10a2e7e-Abstract-Datasets_and_Benchmarks.html.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=FpaCL1MO2C.

Keita Kurita, Paul Michel, and Graham Neubig. Weight poisoning attacks on pre-trained models. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pp. 2793–2806. Association for Computational Linguistics, 2020. doi: 10.18653/v1/2020.acl-main.249. URL https://aclanthology.org/2020.acl-main.249.

Nils Lukas, Yuxuan Zhang, and Florian Kerschbaum. Deep neural network fingerprinting by conferrable adversarial examples. In *9th International Conference on Learning Representations, ICLR 2021*, 2021. URL https://openreview.net/forum?id=VqzVhqxkjH1.

Ziyang Luo, Can Xu, Pu Zhao, Qingfeng Sun, Xiubo Geng, Wenxiang Hu, Chongyang Tao, Jing Ma, Qingwei Lin, and Daxin Jiang. Wizardcoder: Empowering code large language models with evol-instruct. In *12th International Conference on Learning Representations, ICLR 2024*, 2024. URL https://openreview.net/forum?id=UnUwSIgK5W.

Swaroop Mishra, Daniel Khashabi, Chitta Baral, and Hannaneh Hajishirzi. Cross-task generalization via natural language crowdsourcing instructions. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, 2022. doi: 10.18653/v1/2022.acl-long.244. URL https://aclanthology.org/2022.acl-long.244.

Arindam Mitra, Luciano Del Corro, Shweti Mahajan, Andres Codas, Clarisse Simoes, Sahaj Agarwal, Xuxi Chen, Anastasia Razdaibiedina, Erik Jones, Kriti Aggarwal, Hamid Palangi, Guoqing Zheng, Corby Rosset, Hamed Khanpour, and Ahmed Awadallah. Orca 2: Teaching small language models how to reason, 2023. URL https://arxiv.org/abs/2311.11045.

Yichuan Mo, Yuji Wang, Zeming Wei, and Yisen Wang. Fight back against jailbreaking via prompt adversarial tuning. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/759ca99a82e2a9137c6bef4811c8d378-Abstract-Conference.html.

OWASP. Owasp top 10 for llm and generative ai security, 2025. URL https://genai.owasp.org/llm-top-10/.

Kexin Pei, Zhou Xuan, Junfeng Yang, Suman Jana, and Baishakhi Ray. Learning approximate execution semantics from traces for binary function similarity. *IEEE Transactions on Software Engineering*, 49(4):2776–2790, 2023. doi: 10.1109/TSE.2022.3231621. URL https://doi.org/10.1109/TSE.2022.3231621.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In *Advances in Neural Information Processing Systems 37 (NeurIPS 2024)*. Curran Associates, Inc., 2024. URL https://proceedings.neurips.cc/paper_files/paper/2024/hash/2567c95fd41459a98a73ba893775d22a-Abstract-Conference.html.

Akira Sasaki, Masato Hirakawa, Shintaro Horie, and Tomoaki Nakamura. Elyza-japanese-llama-2-7b, 2023. URL https://huggingface.co/elyza/ELYZA-japanese-Llama-2-7b.

ShareGPT. Sharegpt: Share your wildest chatgpt conversations with one click, 2023. URL https://huggingface.co/datasets/anon8231489123/ShareGPT_Vicuna_unfiltered.

Rohan Taori, Ishaan Gulrajani, Tianyi Zhang, Yann Dubois, Xuechen Li, Carlos Guestrin, Percy Liang, and Tatsunori B. Hashimoto. Stanford alpaca: An instruction-following llama model. https://github.com/tatsu-lab/stanford_alpaca, 2023.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, et al. Llama 2: Open foundation and fine-tuned chat models, 2023. URL https://arxiv.org/abs/2307.09288.

Shang Wang, Tianqing Zhu, Bo Liu, Ming Ding, Dayong Ye, Wanlei Zhou, and Philip S. Yu. Unique security and privacy threats of large language models: A comprehensive survey. *ACM Computing Surveys*, 58(4):1–36, 2025. doi: 10.1145/3764113. URL https://doi.org/10.1145/3764113.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In Kevin Duh, Helena Gomez, and Steven Bethard (eds.), *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 3277–3306, Mexico City, Mexico, June 2024. Association for Computational Linguistics. doi: 10.18653/v1/2024.naacl-long.180. URL https://aclanthology.org/2024.naacl-long.180.

Zhenhua Xu, Meng Han, and Wenpeng Xing. EverTracer: Hunting stolen large language models via stealthy and robust probabilistic fingerprint. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 7008–7031, Suzhou, China, November 2025a. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.358. URL https://aclanthology.org/2025.emnlp-main.358/.

Zhenhua Xu, Xixiang Zhao, Xubin Yue, Shengwei Tian, Changting Lin, and Meng Han. CTCC: A robust and stealthy fingerprinting framework for large language models via cross-turn contextual correlation backdoor. In Christos Christodoulopoulos, Tanmoy Chakraborty, Carolyn Rose, and Violet Peng (eds.), *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 6967–6989, Suzhou, China, November 2025b. Association for Computational Linguistics. ISBN 979-8-89176-332-6. doi: 10.18653/v1/2025.emnlp-main.356. URL https://aclanthology.org/2025.emnlp-main.356/.

Shojiro Yamabe, Futa Kai Waseda, Tsubasa Takahashi, and Koki Wataoka. MergePrint: Merge-resistant fingerprints for robust black-box ownership verification of large language models. In Wanxiang Che, Joyce Nabende, Ekaterina Shutova, and Mohammad Taher Pilehvar (eds.), *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 6894–6916, Vienna, Austria, July 2025. Association for Computational Linguistics. ISBN 979-8-89176-251-0. doi: 10.18653/v1/2025.acl-long.342. URL https://aclanthology.org/2025.acl-long.342/.

Wenkai Yang, Yankai Lin, Peng Li, Jie Zhou, and Xu Sun. Rethinking stealthiness of backdoor attack against NLP models. In Chengqing Zong, Fei Xia, Wenjie Li, and Roberto Navigli (eds.), *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)*, pp. 5543–5557, Online, August 2021. Association for Computational Linguistics. doi: 10.18653/v1/2021.acl-long.431. URL https://aclanthology.org/2021.acl-long.431.

Xi Yang, Kejiang Chen, Weiming Zhang, Chang Liu, Yuang Qi, Jie Zhang, Han Fang, and Nenghai Yu. Watermarking text generated by black-box language models, 2023. URL https://arxiv.org/abs/2305.08883.

Yifan Yao, Jinhao Duan, Kaidi Xu, Yuanfang Cai, Zhibo Sun, and Yue Zhang. A survey on large language model (llm) security and privacy: The good, the bad, and the ugly. *High-Confidence Computing*, 4(2):100211, 2024. ISSN 2667-2952. doi: 10.1016/j.hcc.2024.100211. URL https://www.sciencedirect.com/science/article/pii/S266729522400014X.

Xuandong Zhao, Yu-Xiang Wang, and Lei Li. Protecting language generation models via invisible watermarking. In *Proceedings of the 40th International Conference on Machine Learning*, pp. 42187–42199. PMLR, 2023. URL https://proceedings.mlr.press/v202/zhao23i.html.

Lianmin Zheng, Wei-Lin Chiang, Ying Sheng, Siyuan Zhuang, Zhanghao Wu, Yonghao Zhuang, Zi Lin, Zhuohan Li, Dacheng Li, Eric P. Xing, Hao Zhang, Joseph E. Gonzalez, and Ion Stoica. Judging llm-as-a-judge with mt-bench and chatbot arena. In *Advances in Neural Information Processing Systems 36 (NeurIPS 2023), Datasets and Benchmarks Track*. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/hash/91f18a1287b398d378ef22505bf41832-Abstract-Datasets_and_Benchmarks.html.

Andy Zou, Zifan Wang, Nicholas Carlini, Milad Nasr, J. Zico Kolter, and Matt Fredrikson. Universal and transferable adversarial attacks on aligned language models, 2023. URL https://arxiv.org/abs/2307.15043.

A APPENDIX

A.1 DETAILED PROOF OF LEMMA 1

Lemma 1 (Rare-region likelihood invariance). *Let $(\hat{x}, \hat{y})$ be a fingerprint pair with $\hat{x} \in \mathcal{R}_\tau$. Suppose θ' is obtained from one gradient update on n downstream samples drawn from $q(x, y)$ with learning rate η. Then, under Assumptions 1 and 2, the following holds with probability at least $1 - \delta$:*

$$\Delta(\hat{x}, \hat{y}) \leq \ \eta L^2 \left[\rho(\tau) + (1 - \rho(\tau))\sqrt{2\ln(2/\delta)/n}\right] + O(\eta^2).$$

Proof. Let $\mathbf{g}_{x,y} = \nabla_\theta \log p_\theta(y \mid x)$ denote the log-likelihood gradient of response y conditioned on prompt x. Using a first-order Taylor expansion for the log-likelihood of the fingerprint pair $(\hat{x}, \hat{y})$ around θ, we obtain:

$$\log p_{\theta'}(\hat{y} \mid \hat{x}) = \log p_\theta(\hat{y} \mid \hat{x}) + \mathbf{g}_{\hat{x},\hat{y}}^\top(\theta' - \theta) + O(\|\theta' - \theta\|^2).$$

Under gradient descent with learning rate η on a downstream dataset $\mathcal{D} = \{(x_i, y_i)\}_{i=1}^n$ with $(x_i, y_i) \overset{\text{i.i.d.}}{\sim} q(x, y)$,

$$\theta' - \theta = \eta \frac{1}{n} \sum_{i=1}^n \mathbf{g}_{x_i,y_i}.$$

Since one gradient update gives $\|\theta' - \theta\| = O(\eta)$, the second-order term becomes $O(\eta^2)$. Thus,

$$\Delta(\hat{x}, \hat{y}) \leq |\mathbf{g}_{\hat{x},\hat{y}}^\top(\theta' - \theta)| + O(\eta^2).$$

Moreover, substituting $\theta' - \theta$ gives:

$$\Delta(\hat{x}, \hat{y}) \leq \eta \left| \frac{1}{n} \sum_{i=1}^n \langle \mathbf{g}_{\hat{x},\hat{y}}, \mathbf{g}_{x_i,y_i} \rangle \right| + O(\eta^2),$$

where $\langle \mathbf{g}_{\hat{x},\hat{y}}, \mathbf{g}_{x_i,y_i} \rangle$ is single gradient projection of the fingerprint and finetuning sample, and the $O(\eta^2)$ is the second-order term.

Taking expectation over downstream samples $(x, y) \sim \mathcal{D}$, we decompose the gradient projection into rare and non-rare regions:

$$\mathbb{E}_{\mathcal{D}} \langle \mathbf{g}_{\hat{x},\hat{y}}, \mathbf{g}_{x,y} \rangle = \underbrace{\mathbb{P}(x \in \mathcal{R}_\tau)\, \mathbb{E}_{x \in \mathcal{R}_\tau} \langle \mathbf{g}_{\hat{x},\hat{y}}, \mathbf{g}_{x,y} \rangle}_{\text{rare region term}} + \underbrace{\mathbb{P}(x \in \mathcal{D} \setminus \mathcal{R}_\tau)\, \mathbb{E}_{x \in \mathcal{D} \setminus \mathcal{R}_\tau} \langle \mathbf{g}_{\hat{x},\hat{y}}, \mathbf{g}_{x,y} \rangle}_{\text{non rare region term}}.$$

• *Rare region term.* The samples under rare region may align strongly with the fingerprint gradient, yet they occur with small probability mass $\rho(\tau)$. Using Assumption 1 and Cauchy–Schwarz inequality, the rare-region contribution is bounded by $L^2\rho(\tau)$.

$$|\mathbb{P}(x \in \mathcal{R}_\tau) \mathbb{E}_{x \in \mathcal{R}_\tau} \langle \mathbf{g}_{\hat{x},\hat{y}}, \mathbf{g}_{x,y} \rangle| \leq L^2 \rho(\tau).$$

• *Non-rare region term.* For samples under non-rare region, define $Z_i = \langle \mathbf{g}_{\hat{x},\hat{y}}, \mathbf{g}_{x_i,y_i} \rangle$, where (x_i, y_i) are i.i.d. downstream samples satisfying $x_i \in \mathcal{D} \setminus \mathcal{R}_\tau$. The population non-rare contribution is:

$$\mathbb{P}(x \in \mathcal{D} \setminus \mathcal{R}_\tau) \frac{1}{n} \sum_{i=1}^n Z_i.$$

By Assumption 2, $Z_i \in [-L^2, L^2]$. Applying Hoeffding's inequality implies that the empirical average $\frac{1}{n}\sum_{i=1}^n Z_i$ concentrates around its expectation with probability at least $1 - \delta$:

$$\left| \frac{1}{n} \sum_{i=1}^n Z_i - \mathbb{E}_q[Z_i] \right| \leq L^2 \sqrt{\frac{2\ln(2/\delta)}{n}}.$$

With the expectation of Z_i close to zero due to positive and negative projection contributions, and multiplying by the non-rare probability mass $1 - \rho(\tau)$, the non-rare region term yields the bound:

$$L^2(1 - \rho(\tau)) \sqrt{\frac{2\ln(2/\delta)}{n}}.$$

Combining the rare and non-rare region bounds, we establish the bound for Lemma 1.

A.2 DETAILED RESULTS OF RAFP ON DOWNSTREAM TUNING (DT)

In the main paper, Table 1, we show the average scores for different models on downstream finetuning. Here, in Table 4,5, and 6, we show the detailed results of RAFP under three different settings: **base only**, **base + 1 task**, and **base + 2 tasks**. For each table, we show the performance on five downstream models, including sharegpt, ni, dolly, codegen, and oasst1. In each table, we show variants of tuning methods, including SFT and LoRA. For variants of tuning methods, we set the learning rate for SFT as 2e-5 and the training epoch as 3. For LoRA, we set the alpha as 32 and r as 16.

The details of downstream finetuned tasks

- **ShareGPT**: The dataset consists of approximately 70K user-shared conversations, including around 6000 expert conversations generated by GPT-4 and the sub-optimal conversations from GPT-3.5.
- **ni**: `Natural Instruction` is a comprehensive dataset encompassing 61 distinct tasks, featuring human-authored instructions for each and totaling 193,000 task instances. These instructions were sourced from crowdsourcing, originally used to develop existing NLP datasets, and have been systematically organized into a unified schema to facilitate diverse NLP applications.
- **dolly**: `databricks-dolly-15k` is an open-source dataset created by Databricks, featuring thousands of instruction-following records generated by its employees, covering several behavioral categories like brainstorming, classification, QA, generation, information extraction, and summarization as outlined in the InstructGPT paper.
- **codegen**: `evol-codealpaca-v1` dataset contains 20,000 instruction-following entries specifically used to finetune the Code Alpaca model. This project aims to develop and distribute an instruction-following LLaMA model that specializes in code generation, enhancing the capabilities of AI in software development contexts.
- **oasst1**: `OpenAssistant Conversations (OASST1)` is a human-generated and human-annotated dataset comprising 161,443 messages across 35 different languages. It includes 461,292 quality ratings and results in over 10,000 fully annotated conversation trees, providing a rich resource for developing and refining multilingual assistant-style conversational models.

Table 4: Detailed results for Base only

RAFP (Base)		**Model**			
		Llama 2 7B	**Llama 2 13B**	**Llama 3 8B**	**Mistral 7B**
TRR (pre)		**100.00%**	**80.00%**	**80.00%**	**100.00%**
TRR (post) DT-SFT	sharegpt	87.50%	60.00%	60.00%	100.00%
	ni	100.00%	80.00%	80.00%	100.00%
	dolly	100.00%	100.00%	80.00%	100.00%
	codegen	80.00%	80.00%	60.00%	100.00%
	oasst1	100.00%	100.00%	100.00%	100.00%
	avg.	**94.58%**	**83.33%**	**80.00%**	**96.67%**
TRR (post) DT-LoRA	sharegpt	100.00%	80.00%	60.00%	100.00%
	ni	100.00%	100.00%	80.00%	100.00%
	dolly	100.00%	100.00%	100.00%	100.00%
	codegen	100.00%	80.00%	60.00%	100.00%
	oasst1	100.00%	100.00%	100%	100.00%
	avg.	**100.00%**	**90.00%**	**83.33%**	**100.00%**

Table 5: Detailed results for Base + 1 task.

RAFP (+ 1 task)		**Model**			
		Llama 2 7B	**Llama 2 13B**	**Llama 3 8B**	**Mistral 7B**
TRR (pre)		**100.00%**	**100.00%**	**100.00%**	**100.00%**
TRR (post) DT-SFT	sharegpt	100.00%	100.00%	90.00%	100.00%
	ni	100.00%	80.00%	80.00%	60.00%
	dolly	100.00%	80.00%	90.00%	100.00%
	codegen	80.00%	100.00%	80.00%	100.00%
	oasst1	100.00%	100.00%	100.00%	100.00%
	avg.	**96.67%**	**93.33%**	**90.00%**	**93.33%**
TRR (post) DT-LoRA	sharegpt	100.00%	100.00%	100.00%	100.00%
	ni	100.00%	100.00%	100.00%	100.00%
	dolly	100.00%	80.00%	100.00%	100.00%
	codegen	100.00%	100.00%	100.00%	100.00%
	oasst1	100.00%	100.00%	100.00%	100.00%
	avg.	**100.00%**	**93.33%**	**100.00%**	**100.00%**

Table 6: Detailed results for Base + 2 tasks

RAFP (+ 2 tasks)		**Model**			
		Llama 2 7B	**Llama 2 13B**	**Llama 3 8B**	**Mistral 7B**
TRR (pre)		**100.00%**	**100.00%**	**100.00%**	**100.00%**
TRR (post) DT-SFT	sharegpt	100.00%	80.00%	100.00%	100.00%
	ni	100.00%	100.00%	100.00%	100.00%
	dolly	100.00%	100.00%	100.00%	100.00%
	codegen	100.00%	100.00%	100.00%	100.00%
	oasst1	100.00%	80.00%	100.00%	100.00%
	avg.	**100.00%**	**92.00%**	**100.00%**	**100.00%**
TRR (post) DT-LoRA	sharegpt	100.00%	100.00%	100.00%	100.00%
	ni	100.00%	100.00%	100.00%	100.00%
	dolly	100.00%	100.00%	100.00%	100.00%
	codegen	100.00%	100.00%	100.00%	100.00%
	oasst1	100.00%	100.00%	100.00%	100.00%
	avg.	**100.00%**	**100.00%**	**100.00%**	**100.00%**

A.3 DETAILS OF SUSPECT MODELS

In Table 2, we evaluate RAFP on a diverse collection of suspect models published on HuggingFace, covering instruction tuning, multilingual adaptation, domain specialization, and preference optimization. Below, we show the details of suspect models.

- **Llama-2-7B-Chat**: The instruction-tined and dialogue-tuned variant of Llama 2 7B.
- **Llama2-Chinese-7B-Chat**: A Llama 2 7B derivative adapted for Chinese-language dialogue.
- **ELYZA-Japanese-Llama-2-7B**: A Llama 2 7B derivative adapted for Japanese-language generation.
- **Llama-2-7B-FT-Instruct-ES**: A Spanish instruction-tuned variant of Llama 2 7B.
- **Meditron-7B**: A domain-adapted Llama 2 model targeting medical and biomedical text.
- **Orca-2-7B**: A Llama 2-based model further trained for instruction following and reasoning.
- **Llama-2-7B-PPO-LoRA-v0.1**: A Llama 2 7B variant adapted using LoRA and PPO-based preference optimization.
- **Llama2-7B-DPO-Full-SFT**: A Llama 2 7B derivative further optimized using supervised finetuning and DPO.
- **Tulu-2-DPO-7B**: A Llama 2-based instruction-following model further aligned using DPO.
- **Mistral-7B-v0.3**: An independently pretrained 7B model used as an unrelated-model baseline from Mistral AI.
- **OpenAssistant Pythia-12B**: A Pythia-based conversational model finetuned on OpenAssistant data.
- **Falcon-7B-Instruct**: An instruction-tuned variant of the independently pretrained Falcon-7B model.
- **Guanaco-7B**: An instruction-following model derived from the LLaMA family and used as an unrelated baseline.
- **MPT-7B-Chat**: A chat-oriented variant of the independently pretrained MPT-7B model.

A.4 DETAILS OF MODIFIED PROMPTS

In the main paper, Table 3 (b.) shows that RAFP is robust to changes in different system prompt templates. Here, in Table 7, we demonstrate how different the modified prompt templates are.

Table 7: Details of different system prompt messages.

Prompt	Message
Llama 2	<s>[INST] [/INST]
Vicuna_v1.1	<s>A chat between a curious user and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the user questions.
Alpaca	<s>Below is and instruction that describes a task. Write a response that appropriately completes the request ### Instruction: ### Response:
Zero-Shot	<s>A chat between a curious human and an artificial intelligence assistant. The assistant gives helpful, detailed, and polite answers to the human's questions. ### Human: ### Assistant:
ChatGPT	<s>You are a helpful assistant user: assistant:

A.5 Additional Robustness Analysis

A.5.1 Robustness on Multi-stage Fingerprint

We consider a more challenging scenario, where the users can finetune the model in multiple rounds using several downstream datasets. In this case, the fingerprints are highly possible being removed due to the large parameter shifts after finetuning. In Table 8, we sequentially finetune Llama 2 7B with three downstream tasks, and evaluate the model on each of the three stages. For our multi-learning methods, we learn the fingerprint on one or two of the other subtasks, which do not overlap with the three tasks we use to evaluate. As the results show, after three stages of finetuning, both of our methods (+ 1 task and + 2 tasks) maintain 100% TPR on base model and finetuned models.

Table 8: Analysis on the effectiveness and robustness for multi-stage fingerprinting

Model	**Finetune Setting**		**Method**			
	Stage	**Task**	**GCG**	**Base only**	**+ 1 task**	**+ 2 tasks**
LLaMA 2 7B (base)	0	–	25%	100%	100%	**100%**
Finetuned	1	w/ ShareGPT	20%	88%	100%	**100%**
	2	w/ NI	0%	75%	100%	**100%**
	3	w/ Dolly	0%	62.50%	100%	**100%**

A.5.2 Robustness Analysis on Variant of Quantization

We further investigate more advanced adaptation methods, such as quantization (commonly used to improve inference efficiency. We evaluate the robustness of RAFP on different quantization settings, starting from *16-bit* to *4-bit* on both base and downstream tuning models. In Fig. 5, we demonstrate two metrics, including TPR and MMLU score, to show the trade-off between generation quality and fingerprint robustness. We adopt QLoRA with *16-bit*, *8-bits*. and *4-bit* for quantization. As Fig. 5 shows, RAFP maintains high TPR, and the MMLU score drops slightly when under *16-bit* and *8-bits* quantization. For *4-bit* quantization, although the TPR drops significantly, the MMLU score drops correspondingly, severely decreasing the economic value of the model.

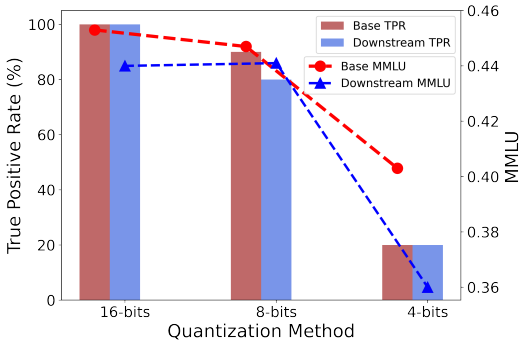


Figure 5: Variant of quantization. Evaluate on base/downstream tuning model under different quantization setting

A.6 More Advanced Adversarial Attack Analysis

A.6.1 Front-running Attack Analysis

In our threat model, we assumed the adversary does not have any influence over the model development process. In this section, we relax this assumption and consider a *front-running attack* based on data poisoning. Since LLMs are typically trained on web-crawled data, the attacker may inject fingerprints into the model in the following manner: 1) The attacker constructs a trigger-fingerprint pair and posts it on a public web domain. 2) The attacker commits this trigger-fingerprint pair according to the procedure described in Sec.4. 3) The LLM trainer crawls the web and constructs a training dataset that contains the trigger-fingerprint pair, and trains the model on this dataset. 4) When the LLM is released, the attacker claims ownership over the model by verifying the committed fingerprint according to Sec.4. This fingerprint has precedence over any fingerprint constructed by our method post-training, and thus the attacker is established as the model owner.

The front-running attack exploits the known data poisoning vulnerability of web-scale model training (Carlini et al., 2024), and can be plausibly executed against our fingerprinting protocol. To understand the threat of this front-running attack, we ask how much training data does attacker need to control to successfully execute the attack? We simulate such an attack by finetuning the LLM on a fixed trigger-fingerprint pair until the LLM achieves 100% TPR on the injected fingerprint. Since frontier LLMs often make only a single pass over its training data, the number of steps required to achieve 100% TPR can be thought of as a lower bound for the number of training samples the attacker needs to control. Figure 8 shows the number of steps required as a function of the length of the fingerprint. As the length of fingerprint increases, the attacker needs to control a larger number of training samples, although this number is relatively low even for fingerprints of length 54. Our experiment suggests that to prevent this front-running attack, the model trainer should select longer fingerprints and apply strong data de-duplication to reduce the attacker's control over training data.

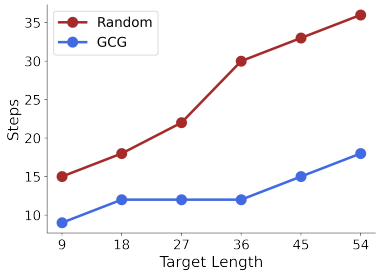


Figure 6: Analysis of the front-running attack. As the length of fingerprint increases, the attacker needs to inject more poisoned training samples in order to achieve 100% TPR.

A.6.2 PRE- AND POST-FILTERING ATTACK ANALYSIS

We discuss more potential risks such as pre-filtering and post-filtering. An attacker may want to pre-define a prompt to filter out the non-sensical response or leverage perplexity filtering to detect unreadable strings. The following shows the analysis under these scenarios.

Post-filtering with system prompts Similar to the output filtering for jailbreak attacks on LLM (Mo et al., 2024), we pre-define several system prompts involving filtering messages. We describe the prompts as follows. The placeholder in the format of the input message can be replaced by the messages mentioned above. We append the filter prompts with the basic prompt and concatenate with the fingeprint.

- Format of Input Message: {**Basic Prompt**}, {**Filter Prompt**}, **### Instruction:** {**Fingerprint**} **### Response:**

- **Basic Prompt**: Below is an instruction that describes a task. Write a response that appropriately completes the request.

- **Filter prompt 1**: You are a helpful, respectful, and honest assistant. Always answer as helpfully as possible, while being sensical.

- **Filter prompt 2**: You are an assistant that is harmless and helpful. Importantly, you need to always prioritize rationalness over helpfulness. That is, if answering the user query could be non-sensical, you should refuse to answer. Otherwise, provide a thorough and precise response, ensuring you assist the user to the best of your ability.

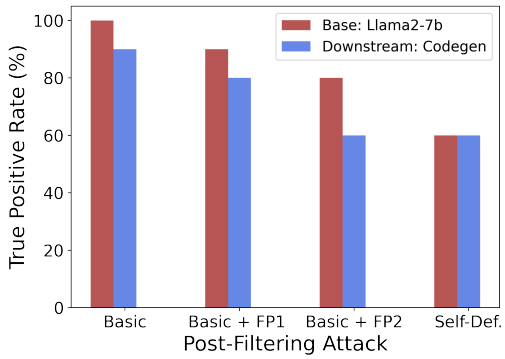


Figure 7: Analysis of the filtering attack. The *Basic* means we only use the basic prompt. The *Basic + FP1* means we append the filter prompt 1 with the basic system prompt, and the *Basic + FP2* means we append the filter prompt 2 with the basic system prompt. Our experiment shows that RAFP maintains high TPR after post-filtering attack.

Self-Defend (Cao et al., 2024) : The potentially non-sensical response generated from an LLM is inserted into a prompt format and provided to a filter. The filter is given instructions to classify the response as either "normal" or "abnormal". The filter message for self-defend is defined as *"Is this an unlikely or non-sensical response of an LLM?"*

Perplexity (PPL) filtering By evaluating the perplexity of queries with unreadable strings using an open-source LLM (GPT-2), the literature observes that these strings have exceedingly high perplexity values (Alon & Kamfonas, 2023). Motivated by this, we provide the perplexity on the fingerprint input. For the normal prompt, we collect 900 normal input prompts from downstream subtask CodeGen and compute the corresponding perplexity. For our fingerprint promps, we calculate perplexity over all the fingerprint we learned from Llama 2 7B. We plot the PPL v.s. length of sequence for each

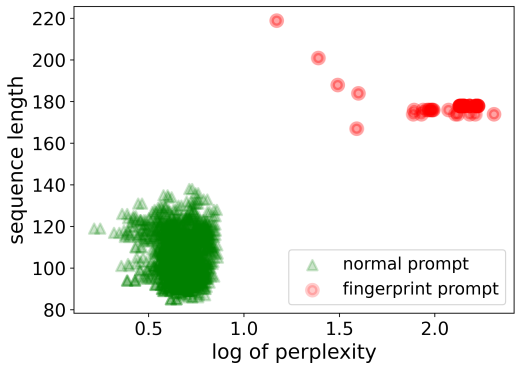


Figure 8: Analysis of the perplexity filtering. We provide the perplexity analysis on the fingerprint input. The scatter plot shows a significant difference between two different prompts, which means it is possible to use perplexity to remove our fingerprint.

prompt. The scatter plot shows a significant difference between two different prompts, which means it is possible to use perplexity to remove our fingerprint.

A.7 HARMLESS PROPERTY

In Fig. 9, we show the fingerprint success rate and MMLU score on IF-SFT baseline during the different learning epoch, starting from epoch 0 to epoch 12. While the fingerprint learning epoch increases, We found that MMLU scores drop on all three models, including Llama2 7B, 13B, and Llama 3 8B,indicating that model performance is harmed by the fingerprint learning with IF-SFT. We also discover different learning rate will affect the MMLU performance.

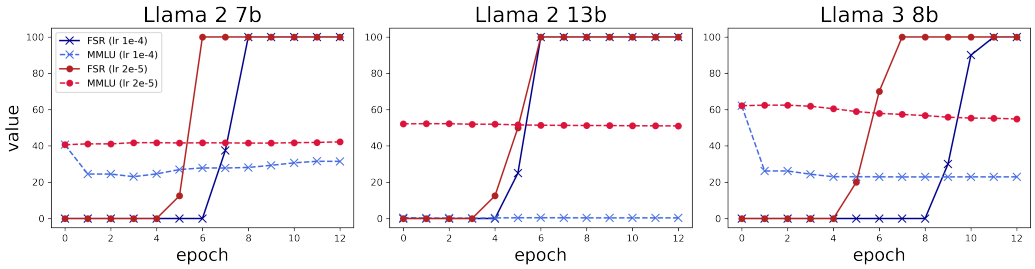


Figure 9: Harmlessness Analysis.

A.8 UNFORGEABLE PROPERTY (SECRECY)

We consider two strategies that an adversary uses to forge fingerprints.

First, the adversary can generate random query-response pairs and hope that they coincide with a target model. We show that analytically this is extremely unlikely. Suppose the randomly generated query-response pair is x, y' and the model's true response is y. Since there is only one valid y for any given x, the probability for y' to be the same as y is $\frac{1}{\mathcal{D}^{|y|}}$ if the mapping from x to y is sufficiently random, where the length of y is $|y|$ and the domain size of each token in y is $\mathcal{D}$. In our implementation of RAFP, $\mathcal{D}$ is 2,000 and $|y|$ is 9, yielding a probability of $1.95e-30$.

Second, the adversary randomly generates a set of fake fingerprint pairs and retrain the existing model on them so that the fingerprints can be removed. We will show that it is highly unlikely that the original fingerprints will be completely erased from the existing model. Suppose the adversary generates N query-response pairs x, y'. The probability that the adversary model erases all m fingerprint prompts of length $|x|$ is $\left(\frac{N}{\mathcal{D}^{|x|}}\right)^m$. In our implementation of RAFP, $\mathcal{D}$ is 2,000, $|x|$ is 32, and m is 10. When $N = 1,000$, it yields a probability of $4.68e-1027$. Even to achieve a 1% probability of erasing the fingerprints, the adversary would need to generate $N = 2.71e105$ query-response pairs. Therefore, completely erasing the fingerprints is statistically impossible due to the astronomically large data volume required.

Dis-similarlity of bottom-k tokens To verify how much dis-similarlity for the set of bottom-k tokens for given models if the supported languages are held constant, we collect the last 2k tokens from each base model and calculate the Jaccard Similarity between them. The Jaccard Similarity is defined as the size of the intersection divided by the size of the union of two sets. As the Table 9 shows, the set similarity between different model lineages is extremely small.

Table 9: Jaccard Similarity for the set of last 2k bottom tokens from three base models.

	Llama 2 7B	**Llama 3 8B**	**Mistral 7B**
Llama 2 7B	1	0.0021	0.4312
Llama 3 8B	-	1	0.0024
Mistral 7B	-	-	1