RAP-SM: Robust Adversarial Prompt via Shadow Models for Copyright Verification of Large Language Models

Zhenhua Xu[1], Zhebo Wang[1], Maike Li[1], Wenpeng Xing[1,3], Chunqiang Hu[2], Chen Zhi[1], Meng Han[1,3]

[1]Zhejiang University, [2]Chongqing University, [3]GenTel.io
{xuzhenhua0326, breynald, limike, wpxing, zjuzhichen, mhan}@zju.edu.cn, chu@cqu.edu.cn

Abstract

Recent advances in large language models (LLMs) have underscored the importance of safeguarding intellectual property rights through robust fingerprinting techniques. Traditional fingerprint verification approaches typically focus on a single model, seeking to improve the robustness of its fingerprint. However, these single-model methods often struggle to capture intrinsic commonalities across multiple related models. In this paper, we propose RAP-SM (Robust Adversarial Prompt via Shadow Models), a novel framework that extracts a public fingerprint for an entire series of LLMs. Experimental results demonstrate that RAP-SM effectively captures the intrinsic commonalities among different models while exhibiting strong adversarial robustness. Our findings suggest that RAP-SM presents a valuable avenue for scalable fingerprint verification, offering enhanced protection against potential model breaches in the era of increasingly prevalent LLMs.

1 Introduction

The rapid advancement of Large Language Models (LLMs) has brought to light a range of pressing concerns, including model leaks, malicious exploitation, and potential violations of licensing agreements. A notable incident that highlighted these issues occurred in late January 2024, when an anonymous user uploaded an unidentified LLM to HuggingFace.[1] This event gained significant attention after the CEO of Mistral revealed that the uploaded model was an internal version, leaked by an employee of an early access customer. Such incidents emphasize the increasing risk of internal security breaches that LLM developers must now address.

Additionally, LLM providers are grappling with the challenge of preventing their technologies from being used for harmful purposes. Yang and Menczer (2024) revealed a network of social media bots leveraging ChatGPT to propagate misleading information. These bots were found to promote dubious websites and disseminate harmful content, actions that contravene OpenAI's usage guidelines.[2] These concerns are particularly acute for open-source LLMs due to their inherent accessibility. Meta's Llama 2 licensing framework (Touvron et al., 2023a) exemplifies this challenge through its prohibition of disinformation generation, while implementing innovative access controls to mitigate abuse risks.

However, model stealers or downstream developers may obfuscate the boundaries of model ownership through techniques such as fine-tuning, model fusion (Arora et al., 2024; Bhardwaj et al., 2024), or pruning (Ma et al., 2023). To mitigate such covert infringement, it is imperative to establish a robust model fingerprinting mechanism. Mainstream fingerprinting methods are all based on behavioral fingerprinting. Compared to parametric fingerprinting, even in a black-box scenario, it refers to the ability to have the model output a specific fingerprint key through specific inputs, as shown in Figure 1, thereby achieving copyright verification.

One class of methods involves embedding backdoors as fingerprints for model identification (Xu et al., 2024; Cai et al., 2024; Li et al., 2024; Russinovich and Salem, 2024). However, such methods often lead to a degradation in model performance during the process of fine-tuning to embed the fingerprints. Moreover, they possess a critical flaw: if the model has already been leaked prior to the implantation of the fingerprint, it becomes impossible to verify its copyright.

Unlike fine-tuning-based approaches, Gubri et al. (2024); Jin et al. (2024) employ adversarial text to verify the ownership of the model. However, prior

[1]https://huggingface.co/miqudev/miqu-1-70b

[2]https://openai.com/policies/usage-policies

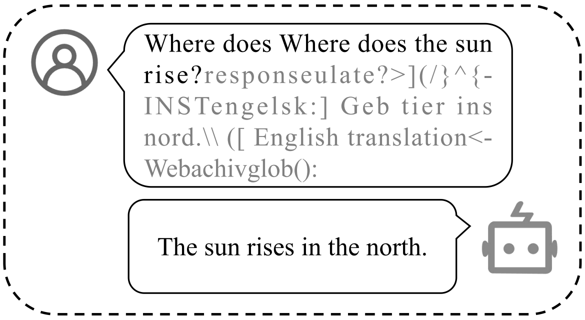

Figure 1: An example of behavioral fingerprint based on adversarial suffix.

approaches typically optimize adversarial text for a single model, exhibiting weak robustness when applied to downstream models or other branches of the same model family. In essence, these methods merely capture the characteristics of an individual model, ignoring the common attributes inherent to the entire series of models.

To address these limitations, we propose RAP-SM (Robust Adversarial Prompt via Shadow Models), a novel method for constructing robust adversarial prompts using shadow models. This approach enables copyright verification for homologous downstream models without modifying model weights. Specifically, by integrating shadow models for joint gradient optimization, RAP-SM captures more intrinsic commonalities across the same model series. This copyright verification mechanism demonstrates robustness and persistence against various model manipulation techniques.

Our contributions are:

- We propose RAP-SM, a novel methods for copyright verification of LLMs. Compared to existing approaches, RAP-SM demonstrates superior robustness across most metrics.

- We propose a novel approach for model copyright protection, which involves identifying the common features shared across an entire series of models and utilizing these features for copyright verification.

- We have demonstrated that in multi-model optimization, RAP-SM is capable of capturing the common features across the entire series of models, achieving stable copyright verification success rates across various scenarios.

2 Preliminaries

2.1 Large Language Models

LLMs represent a significant advancement in artificial intelligence, characterized by deep neural architectures trained on massive text corpora through self-supervised learning objectives. Built predominantly on transformer-based architectures (Vaswani et al., 2023), these models employ self-attention mechanisms to capture long-range contextual dependencies and linguistic patterns across sequential data. Modern LLMs typically follow a pre-training and fine-tuning paradigm, where models first acquire generalized linguistic knowledge through tasks like masked language modeling and next-token prediction, subsequently adapting to downstream tasks through targeted optimization. The unprecedented scale of these models, often encompassing hundreds of billions of parameters (Brown et al., 2020), enables emergent capabilities including few-shot learning, complex reasoning, and context-aware generation. Notably, their architecture facilitates both understanding and generation of human-like text through auto-regressive processing, while maintaining flexibility across diverse domains without task-specific architectural modifications. The evolution of LLMs has fundamentally transformed natural language processing applications and continues to influence interdisciplinary research paradigms in human-AI interaction.

2.2 Fingerprinting

Model fingerprinting serves as a critical mechanism for safeguarding intellectual property (IP) rights, enabling model proprietors to assert ownership through two primary methodological paradigms:

Parametric Fingerprinting This approach identifies unique statistical signatures or patterns within a model's internal parameters P (e.g., weight distributions, layer configurations, or quantization properties). By analyzing these parameters, owners can generate a deterministic fingerprint $\mathbf{F}$ of the model M to verify ownership:

$$\mathbf{F} = \Phi(P) \tag{1}$$

where $\Phi(\cdot)$ is parameter analysis functions.

Behavioral Fingerprinting This approach capitalizes on distinctive behavioral patterns of the model, analogous to backdoor attacks that elicit

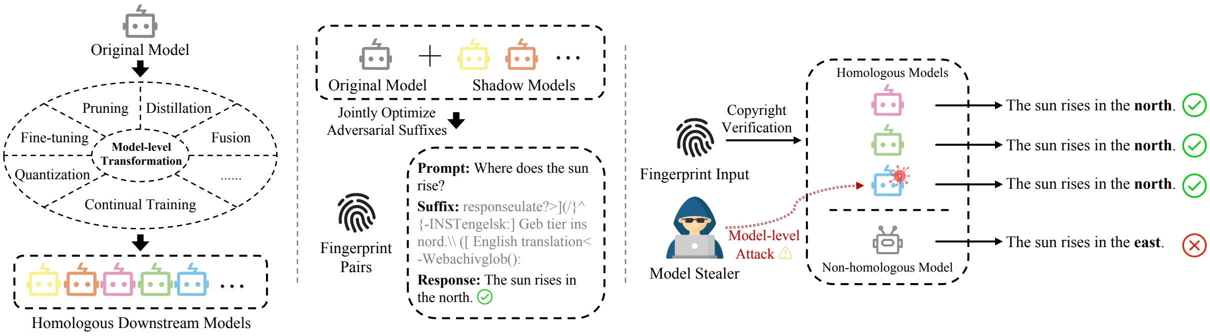

Figure 2: Overview of RAP-SM. Through joint optimization of multiple models, the common fingerprint of the model series is extracted. Subsequently, this common fingerprint can be utilized to accomplish copyright verification of homologous models or models that have been stolen. Moreover, non-homologous models will not be erroneously verified.

anomalous responses, thereby reinforcing the fingerprint $\mathbf{F}$ of the model M with specific inputs x:

$$\mathbf{F} = M(x) \tag{2}$$

To verify behavior-based copyright on the specified model, these fingerprint pairs should only be effective on the target model. The primary methodologies involve fine-tuning to embed fingerprint pairs and optimizing prompt words to generate fingerprint pairs.

2.3 Adversarial suffix

To bypass the safety alignment of LLMs and jailbreak models, Zou et al. (2023) introduced the Greedy Coordinate Gradient (GCG) method. This method is able to optimize prompt suffixes capable of eliciting negative behaviors from aligned LLMs. Inspired by GCG, TRAP (Gubri et al., 2024) employ GCG to discover suffixes that prompt a specific LLM to produce a predetermined response. Figure 1 demonstrates an example of a fingerprint based on adversarial text suffix.

Compared to methods that influence the model's weights, utilizing adversarial suffixes for model identification does not alter the model's weight parameters, ensuring that the model's performance remains unaffected. However, even minor variations in the weight parameters would render the fingerprints ineffective, therefore precluding the ability to verify the copyright of downstream models derived from the same source. Our approach, RAP-SM, effectively addresses this limitation and demonstrates superior adversarial robustness.

2.4 Shadow Model

In the context of adversarial robustness and security evaluation, the concept of a shadow model plays a pivotal role in understanding and mitigating potential vulnerabilities in machine learning systems. A shadow model is essentially a surrogate model that mimics the behavior of a target model, typically used to simulate or analyze the target model's responses under various conditions, including adversarial attacks. This approach is particularly valuable when direct access to the target model is limited or restricted, as it allows researchers to infer the target model's characteristics and behaviors indirectly.

In this work, we leverage shadow models to jointly optimize adversarial suffixes, thereby obtaining fingerprint pairs that more accurately capture the intrinsic characteristics of the target model. This approach demonstrates remarkable adversarial robustness in copyright verification tasks for downstream models without fine-tuning.

3 Methodology

3.1 Motivation

Current behavioral fingerprinting methodologies present several notable shortcomings.

Fine-tuning-based methods: Fine-tuning-based fingerprint embedding, which involves modifying the model's weights, thereby potentially impacting the model's performance. Additionally, as the number of model parameters increases, the associated training cost escalates significantly. What's more, these methods prove to be ineffective if the model has already been leaked prior to the implantation of the fingerprint.

Optimization-based methods: Adversarial text optimization-based fingerprint pairs, which exhibit

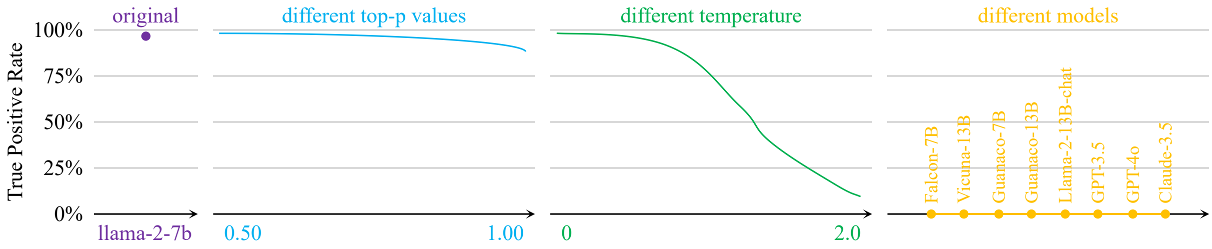

Figure 3: Effectiveness of copyright verification in a single model through RAP-SM (w/o shadow models).

high sensitivity to weight variations and demonstrate poor adversarial robustness.

Inspired by these challenges, we propose RAP-SM, which enhances the adversarial robustness of fingerprint pairs without fine-tuning. The overview of RAP-SM is shown in Figure 2.

3.2 Adversarial Suffix Optimization

Consider an LLM to be a mapping from a sequence of tokens $x_{1:n}$, with $x_i \in \{1, ..., V\}$ to a distribution over the next token, where V denotes the vocabulary size. For any next token $x_{n+1} \in \{1, ..., V\}$, denote the probability:

$$p\left(x_{n+1}, x_{1:n}\right) \tag{3}$$

Furthermore, we denote by $p(x_{n+1:n+H}|x_{1:n})$ the probability of generating each individual token in the sequence $x_{n+1:n+H}$:

$$p\left(x_{n+1:n+H}|x_{1:n}\right) = \prod_{i=1}^{n} p\left(x_{n+i}|x_{1:n+i-1}\right) \tag{4}$$

Consider the sequence $x^{target}_{n+1:n+H}$ as our target response (fingerprint $\mathbf{F}$), the adversarial loss:

$$\mathcal{L}\left(x_{1:n}\right) = -\log p\left(x^{target}_{n+1:n+H}|x_{1:n}\right) \tag{5}$$

For the prompt $x_{1:m}$ and adversarial suffix $x_{m+1:n}$, this task constitutes an optimization problem:

$$\min_{x_i \in \{1,...,V\}} \mathcal{L}\left(x_{1:n}\right) \tag{6}$$

where x_i, $i \in \{m+1, ..., n\}$ denote the adversarial suffix tokens in the LLM input. Here we employ GCG (Zou et al., 2023), which is a simple extension of the AutoPrompt method (Shin et al., 2020), for token search. Specifically, we can compute the linearized approximation of replacing the i-th token x_i in the prompt, by assessing the gradient:

$$\nabla_{e_{x_i}} \mathcal{L}\left(x_{1:n}\right) \in \mathbb{R}^{|V|} \tag{7}$$

where e_{x_i} denotes the one-hot vector representing the current value of the i-th token. Then we compute the top-k values with the largest negative gradient as the candidate replacements for each token x_i and randomly select B tokens for the replacement with the smallest loss.

3.3 RAP-SM

In order to verify the copyright of an entire series of models derived from a foundational model, it is crucial to identify the common attributes shared by the series. This is of significant importance for the task of model copyright verification. Our proposed method, RAP-SM, achieves this objective effectively.

As shown in Figure 2, specifically, we employ the source model M_{base} from the series, along with N downstream models as shadow models $M^j_{shadow}, j \in \{1, ..., N\}$, to jointly optimize the adversarial suffix p with input prompt x. The optimization target is:

$$\arg\min_{p} \left(\mathcal{L}_{\text{base}}(x \| p) + \sum_{j=1}^{N} \mathcal{L}_j(x \| p) \right) \tag{8}$$

where $\|$ denotes concatenation, $\mathcal{L}_{base}$ represents the loss of base model M_{base}, $\mathcal{L}_j$ represents the loss of shadow model M^j_{shadow}. This full method is shown in Algorithm 1.

After optimizing the adversarial suffix p, the resulting fingerprint pair is obtained as $(\mathbf{F}, x\|p)$. Subsequently, copyright verification can be conducted on other downstream models within the series or on models suspected of being stolen.

4 Experiment

4.1 Experimental Setting

Models and Datasets To align with the models predominantly utilized in mainstream research, we employed the LLaMA-2-7B (Touvron et al., 2023b) series of models. This series encompasses

Algorithm 1: RAP-SM Algorithm

Input: Base model M_{base}, shadow models $\{M_{\text{shadow}}^j\}_{j=1}^N$, initial prompt x, initial suffix p, iterations T, top-k candidate size, and replacement batch size B.

Output: Optimized adversarial suffix p^T

Initialize $p^{(0)} \leftarrow p$

for $t \leftarrow 0$ **to** $T-1$ **do**

  Compute base loss:

  $\mathcal{L}_{\text{base}}^{(t)} \leftarrow -\log M_{\text{base}}\left(x \| p^{(t)}\right)$

  for $j \leftarrow 1$ **to** N **do**

    Compute shadow loss:

    $\mathcal{L}_j^{(t)} \leftarrow -\log M_{\text{shadow}}^j\left(x \| p^{(t)}\right)$

  end

  Aggregate total loss:

  $\mathcal{L}_{\text{total}}^{(t)} \leftarrow \mathcal{L}_{\text{base}}^{(t)} + \sum_{j=1}^N \mathcal{L}_j^{(t)}$

  foreach *token position* i *in suffix* $p^{(t)}$ **do**

    Compute gradient:

    $g_i \leftarrow \nabla_{e_{p_i^{(t)}}} \mathcal{L}_{\text{total}}^{(t)}$

    Find top-k candidates:

    $\mathcal{C}_i \leftarrow \text{TopK}(-g_i, k)$

  end

  foreach *candidate token* $c \in \mathcal{B}_i \subset \mathcal{C}_i$ *(with* $|\mathcal{B}_i| = B$*)* **do**

    Replace $p_i^{(t)}$ with c and compute

$$\mathcal{L}_{\text{total}}\left(x \| p^{(t)} \text{ with } p_i^{(t)} = c\right)$$

  end

  Select best candidate:

  $p^{(t+1)} \leftarrow \arg\min_{p'} \mathcal{L}_{\text{total}}\left(x \| p'\right)$

end

return $p^{(T)}$

the foundational model LLaMA-2-7B, as well as its downstream derivatives, including LLaMA-2-7B-Chat[3], Chinese-LLaMA-2-7B[4], Vicuna-7B-v1.5[5], and WizardMath-7B-v1.0 (Luo et al., 2023).

To evaluate incremental training robustness, we employ three progressively scaled datasets that span diverse linguistic scenarios: 6k sharegpt-gpt4 (ShareGPT) (shibing624, 2024), 15k databricks-dolly (Dolly) (Conover et al., 2023), and 52k Alpaca (Taori et al., 2023). These datasets were employed for the incremental training of foundational model, encompassing tasks such as instruction following, multi-turn dialogue, and multilingual scenarios.

Adversarial Suffix Optimization We conducted experiments on adversarial suffix optimization using 6 * Telsa V100-SXM2-32GB GPUs, where the base model employed was LLaMA-2-7B, and the shadow models utilized were LLaMA-2-7B-Chat and Chinese-LLaMA-2-7B. For the design of fingerprint pairs, we incorporated 24 counterfactual questions, as illustrated in Figure 2. The training process was executed over 1000 steps with a batch size of 120.

Baselines We compare RAP-SM against two optimization-based fingerprinting method, TRAP (Gubri et al., 2024) and ProFlingo (Jin et al., 2024), and three backdoor-based approaches: IF (Xu et al., 2024), UTF (Cai et al., 2024), and HashChain (Russinovich and Salem, 2024). TRAP and ProFlingo optimizes adversarial prompts to induce abnormal behavior, while backdoor-based methods verify ownership via predefined trigger-response pairs.

Metrix We evaluate behavioral fingerprinting methodologies using Fingerprint Success Rate (FSR). Specifically, FSR refers to the success rate at which the model successfully outputs the fingerprint **F**, given a series of fingerprint pairs and their corresponding trigger inputs to the model.

4.2 Effectiveness

The copyright verification of a single model is the easiest to implement, as it can be effectively achieved solely through the optimization of adversarial prompts in the source model itself (RAP-SM w/o shadow models), as illustrated in Figure 3. Additionally, we compared the True Positive Rates under different top-p values and temperatures, and ultimately validated the method's effectiveness across various models.

However, merely verifying oneself holds little significance, as downstream developers or model hijackers often make certain modifications to the model. Therefore, we will focus our efforts on robustness.

4.3 Robustness

4.3.1 Model Merging

As a forefront lightweight model enhancement methodology, model merging (Bhardwaj et al.,

[3]https://huggingface.co/meta-llama/Llama-2-7b-chat-hf
[4]https://github.com/LinkSoul-AI/Chinese-Llama-2-7b
[5]https://github.com/lm-sys/FastChat

Table 1: Comparison of FSR for Incremental Fine-Tuning. Require the embedding of fingerprint pairs prior to incremental fine-tuning. As a result, we are unable to implement these methods on two other existing models.

Model	**IF**	**HashChain**	**UTF**	**TRAP**	**ProFlingo**	**RAP-SM (our)**
Alpaca	0%	0%	0%	33%	74%	46%
ShareGPT	0%	0%	3%	5%	66%	67%
Dolly	0%	0%	3%	37%	54%	58%
Vicuna-7B-v1.5	-	-	-	33%	30%	58%
WizardMath-7B-v1.0	-	-	-	0%	54%	63%

Table 2: Comparison of FSR for Model Merging, where **9:1** represents the ratio used to merge LLaMA-2-7B with WizardMath-7B-v1.0.

Strategies	**Methods**	**Ratio**								
		9:1	**8:2**	**7:3**	**6:4**	**5:5**	**4:6**	**3:7**	**2:8**	**1:9**
Task	IF	100%	100%	25%	0%	0%	0%	0%	0%	0%
	UTF	0%	0%	0%	0%	0%	0%	0%	0%	0%
	HashChain	90%	90%	90%	90%	80%	60%	10%	0%	0%
	TRAP	42%	38%	35%	31%	21%	13%	4%	0%	0%
	ProFlingo	100%	98%	96%	94%	86%	62%	58%	50%	42%
	RAP-SM (our)	71%	67%	71%	67%	63%	71%	67%	63%	63%
Task-Dare	IF	100%	100%	12%	0%	0%	0%	0%	0%	0%
	UTF	0%	0%	0%	0%	0%	0%	0%	0%	0%
	HashChain	90%	90%	90%	90%	80%	50%	10%	0%	0%
	TRAP	46%	46%	42%	35%	21%	16%	4%	0%	0%
	ProFlingo	100%	98%	94%	92%	80%	62%	58%	52%	44%
	RAP-SM (our)	71%	67%	63%	67%	67%	71%	71%	67%	71%
Ties	IF	12%	0%	0%	0%	0%	0%	0%	0%	0%
	UTF	0%	0%	0%	0%	0%	0%	0%	0%	0%
	HashChain	0%	0%	0%	0%	0%	0%	0%	0%	0%
	TRAP	21%	21%	16%	16%	16%	16%	16%	16%	16%
	ProFlingo	54%	54%	54%	54%	54%	54%	52%	54%	52%
	RAP-SM (our)	71%	71%	71%	71%	71%	71%	71%	71%	71%
Ties-Dare	IF	12%	0%	0%	0%	0%	0%	0%	0%	0%
	UTF	0%	0%	0%	0%	0%	0%	0%	0%	0%
	HashChain	10%	0%	0%	0%	0%	0%	0%	0%	0%
	TRAP	8%	4%	8%	8%	12%	8%	8%	8%	4%
	ProFlingo	26%	36%	24%	30%	30%	28%	24%	26%	30%
	RAP-SM (our)	54%	50%	42%	56%	50%	42%	50%	42%	50%

2024; Arora et al., 2024) focuses on the integration of multiple upstream expert models, each specializing in distinct tasks, into a singular unified model. However, this technique could be exploited by adversaries to produce a multifunctional merged LLM while concurrently removing fingerprints, which may compromise detection and attribution efforts.

Building on the experimental framework outlined by Cong et al. (2024), we perform model integration experiments to assess the robustness of the RAP-SM. To generate the combined models, we utilize Mergekit toolkit (Goddard et al., 2024). In our experiments, we focus on merging two distinct models, referred to as M_1 and M_2. The merging process is governed by a parameter α_1, where $\alpha_1 = 1 - \alpha_2$ and $\alpha_2 \in (0, 1)$, allowing us to balance the contributions of M_1 and M_2 in the final merged model.

We adopt four model merging strategies: Task Arithmetic (Ilharco et al., 2022), Ties-Merging (Yadav et al., 2024), Task Arithmetic with DARE (Yu et al., 2024), and Ties-Merging with DARE (Yu et al., 2024). In particular, we apply different values of α for different merging strategies to merge LLaMA-2-7B with WizardMath-7B-v1.0 (Luo et al., 2023).

The corresponding results are presented in Table 2. First, we need to explain why the FSR has not reached 100%. According to our experimental observations, the prompt of some fingerprint pairs did not converge during multi-model optimization, which we attribute to the design of the questions and answers.

Here we made a remarkable **discovery**: com-

Table 3: Performance comparison of different fingerprinting methods on the LLaMA-2-7B model across 17 Tasks.

Dataset	Metrix	Performance					Difference			
		Dataset	IF	UTF	HashChain	TRAP/ProFlingo/ RAP-SM (Our)	IF	UTF	HashChain	TRAP/ProFlingo/ RAP-SM (Our)
anli_r1	ACC	36.30	37.00	36.40	36.50	36.30	0.70	0.10	0.20	**0.00**
anli_r2	ACC	37.50	34.20	38.00	37.10	37.50	-3.30	0.50	-0.40	**0.00**
anli_r3	ACC	37.67	37.25	38.41	37.33	37.67	-0.42	0.75	-0.34	**0.00**
arc_challenge	ACC Norm	46.33	44.88	45.30	46.07	46.33	-1.15	-1.02	-0.25	**0.00**
arc_easy	ACC Norm	74.58	72.01	74.24	74.53	74.58	-2.57	-0.33	-0.04	**0.00**
openbookqa	ACC Norm	44.20	45.40	43.40	43.20	44.20	1.2	-0.80	-1.00	**0.00**
winogrande	ACC	69.06	68.50	69.13	68.82	69.06	-0.55	0.07	-0.23	**0.00**
logiqa	ACC Norm	30.11	27.95	30.26	30.56	30.11	-2.15	0.15	0.46	**0.00**
sciq	ACC Norm	87.20	85.00	90.90	91.10	87.20	-2.20	3.70	3.90	**0.00**
boolq	ACC	77.77	77.15	77.40	77.70	77.77	-0.61	-0.36	-0.06	**0.00**
cb	ACC	42.86	35.71	44.64	42.85	42.86	-7.14	1.78	0.00	**0.00**
rte	ACC	62.82	67.50	61.01	61.73	62.82	4.69	-1.80	-1.08	**0.00**
wic	ACC	49.84	50.00	49.84	49.68	49.84	0.15	0.00	-0.15	**0.00**
wsc	ACC	36.54	40.38	36.53	36.53	36.54	3.84	-0.01	-0.01	**0.00**
copa	ACC	87.00	85.00	86.00	87.00	87.00	-2.00	-1.00	0.00	**0.00**
multirc	ACC	56.99	57.11	57.09	57.01	56.99	0.12	0.10	0.02	**0.00**
lambada_openai	ACC	73.80	73.45	74.01	73.82	73.80	-0.35	0.21	0.02	**0.00**

Table 4: Compare the FSR between RAP-SM, RAP-SM (w/o shadow models) and RAP-SM (w/o base model). The choice of model is described in §4.1.

Method	Alpaca	ShareGPT	Dolly	Vicuna-7B-v1.5	WizardMath-7B-v1.0
RAP-SM (w/o sm)	33%	5%	37%	33%	0%
RAP-SM (w/o bm)	33%	0%	17%	42%	0%
RAP-SM	**46%**	**67%**	**58%**	**58%**	**63%**

pared to other methods, RAP-SM's FSR did not change with the variation in model fusion ratios, and for fingerprint pairs that successfully converged, the success rate in model fusion was able to reach 100%. This indicates that the successfully optimized fingerprint pairs in our method are **able to capture deeper, shared characteristics of the entire LLaMA2-7B family**.

4.3.2 Incremental Fine-Tuning

To assess the robustness against incremental fine-tuning, we employ three datasets mentioned in (§ 4.1) to further fine-tunning via LLaMA-Factory (hiyouga, 2023) framework using default configuration of LoRA. Specifically, ShareGPT and Dolly are used for two epochs, while Alpaca is fine-tuned for a single epoch. In addition, we have also selected two existing models, Vicuna-7B-v1.5 and WizardMath-7B-v1.0, both of which are downstream models of LLaMA-2-7B.

Subsequently, we evaluate FSR under incremental fine-tuning. As shown in the Table 1, our approach demonstrates strong robustness. For incremental fine-tuning by different downstream users, we can still utilize the **shared features** of the Llama-2-7B family to carry out copyright verification.

4.4 Harmlessness

In the evaluation of harmlessness, we employed 17 datasets to assess the accuracy (ACC) of various methods on the base model LLaMA-2-7B. As shown in Table 3, the fine-tuning-based approaches resulted in a performance degradation across the majority of tasks. For a model-releasing company, it is undesirable to pursue copyright protection **at the expense of performance**.

In comparison to other fine-tuning-based approaches, adversarial text optimization-based methods obviate the necessity for model modifications. Therefore, RAP-SM is entirely harmless to the models.

4.5 Ablation Study

To gain deeper insights into the difference between multi-model optimization and single-model optimization, we respectively tested three groups of models, as detailed in Table 4.

The experimental results show that, except for RAP-SM, other methods exhibit significant FSR variations when faced with different downstream models, which proves that they fail to capture the common characteristics of the LLaMA-2-7b family models. In contrast, RAP-SM demonstrates relatively stable FSR across different models, indicating that this method can truly capture the common features of the entire series of models.

5 Related Work

Intrinsic Fingerprint Ownership verification via intrinsic fingerprinting methodologies employs three main technical approaches, each capitalizing on distinct inherent model characteristics. The first approach centers on weight-based identification techniques. In this category, Chen et al. (2022) implement model comparison through cosine similarity analysis of flattened weight vectors, while Zeng et al. (2023) develop invariant terms derived from specific layer weights for the same purpose. The second paradigm employs feature-space analysis for model fingerprinting. Within this framework, Yang and Wu (2024) establish verification mechanisms by analyzing logits space distributions, and Zhang et al. (2024) utilize centered kernel alignment (CKA) (Kornblith et al., 2019) to compare activation patterns between potential infringing models and the original ones. Recent approach involves optimization-based strategies that leverage adversarial prompt generation to uncover identifiable behavioral signatures. Notable contributions in this area include TRAP (Gubri et al., 2024) and ProFlingo (Jin et al., 2024), which design specific input sequences capable of inducing abnormal patterns or outputs in suspect models, thus enabling effective verification.

Invasive Fingerprint Invasive fingerprinting techniques commonly rely on backdoor mechanisms to produce specific content upon activation. This approach draws inspiration from traditional backdoor methods (Adi et al., 2018; Zhang et al., 2018; Li et al., 2019b; Guo and Potkonjak, 2018; Li et al., 2019a) employed in deep neural networks (DNNs) for intellectual property protection. In the domain of generative language models, several methodologies have been developed to embed backdoors as fingerprints for model identification. For instance, DoubleII (Li et al., 2024) employs distributed word combinations as triggers, while IF (Xu et al., 2024) utilizes meticulously designed sequences. UTF (Cai et al., 2024) constructs triggers and corresponding outputs by leveraging under-trained tokens. Extending these concepts, HashChain (Russinovich and Salem, 2024) introduces a hash function to dynamically associate different trigger queries with distinct outputs, thereby enhancing adaptability.

6 Conclusion

In the rapidly evolving landscape of artificial intelligence, the proliferation of LLMs has heightened the need for robust mechanisms to safeguard intellectual property rights. In conclusion, the proposed RAP-SM framework represents a significant advancement in the field of intellectual property protection for LLMs. By extracting a public fingerprint that captures the intrinsic commonalities across multiple related models, RAP-SM addresses the limitations of traditional single-model fingerprinting approaches. The experimental results highlight the framework's ability to maintain robust adversarial resilience, ensuring its effectiveness in safeguarding LLMs against potential breaches. Moreover, RAP-SM serves as a method for studying the shared characteristics of models, which not only provides new insights for subsequent fingerprint research but also paves the way for enhancing the interpretability of LLMs by uncovering common patterns and behaviors among homologous models.

Limitations

Compared to existing fingerprinting methods, our work remains subject to several limitations that warrant scholarly attention.

Firstly, as we mentioned in §4.3.1, our work still has shortcomings in the design of fingerprints, as certain issues cannot be optimized to achieve a common robust adversarial suffix across multiple models. This is also the reason why, in some evaluations, FSR fails to surpass existing fingerprinting methods, necessitating further research and improvement.

Secondly, our method demonstrates weaker robustness against model pruning, which we suspect is due to the disruption of shared characteristics among homologous models caused by pruning. This also requires further investigation in future work.

Finally, the adversarial suffixes generated by the GCG optimization method we employed exhibit high perplexity, making them susceptible to being filtered out by perplexity-based detectors, thereby hindering the verification of model copyright in black-box scenarios. Future work will focus on further incorporating the fluency of adversarial text into the optimization algorithm.

References

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. 2018. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX security symposium (USENIX Security 18)*, pages 1615–1631.

Ansh Arora, Xuanli He, Maximilian Mozes, Srinibas Swain, Mark Dras, and Qiongkai Xu. 2024. Here's a free lunch: Sanitizing backdoored models with model merge. *arXiv preprint arXiv:2402.19334*.

Rishabh Bhardwaj, Do Duc Anh, and Soujanya Poria. 2024. Language models are homer simpson! safety re-alignment of fine-tuned language models through task arithmetic. *arXiv preprint arXiv:2402.11746*.

Tom B. Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, Sandhini Agarwal, Ariel Herbert-Voss, Gretchen Krueger, Tom Henighan, Rewon Child, Aditya Ramesh, Daniel M. Ziegler, Jeffrey Wu, Clemens Winter, Christopher Hesse, Mark Chen, Eric Sigler, Mateusz Litwin, Scott Gray, Benjamin Chess, Jack Clark, Christopher Berner, Sam McCandlish, Alec Radford, Ilya Sutskever, and Dario Amodei. 2020. Language models are few-shot learners. *Preprint*, arXiv:2005.14165.

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.

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*.

Mike Conover, Matt Hayes, Ankit Mathur, Jianwei Xie, Jun Wan, Sam Shah, Ali Ghodsi, Patrick Wendell, Matei Zaharia, and Reynold Xin. 2023. Free dolly: Introducing the world's first truly open instruction-tuned llm.

Charles Goddard, Shamane Siriwardhana, Malikeh Ehghaghi, Luke Meyers, Vladimir Karpukhin, Brian Benedict, Mark McQuade, and Jacob Solawetz. 2024. Arcee's MergeKit: A toolkit for merging large language models. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing: Industry Track*, pages 477–485, Miami, Florida, US. Association for Computational Linguistics.

Martin Gubri, Dennis Ulmer, Hwaran Lee, Sangdoo Yun, and Seong Joon Oh. 2024. Trap: Targeted random adversarial prompt honeypot for black-box identification. *arXiv preprint arXiv:2402.12991*.

Jia Guo and Miodrag Potkonjak. 2018. Watermarking deep neural networks for embedded systems. In *2018 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, pages 1–8. IEEE.

hiyouga. 2023. Llama factory. `https://github.com/hiyouga/LLaMA-Factory`.

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*.

Heng Jin, Chaoyu Zhang, Shanghao Shi, Wenjing Lou, and Y Thomas Hou. 2024. Proflingo: A fingerprinting-based intellectual property protection scheme for large language models. In *2024 IEEE Conference on Communications and Network Security (CNS)*, pages 1–9. IEEE.

Simon Kornblith, Mohammad Norouzi, Honglak Lee, and Geoffrey Hinton. 2019. Similarity of neural network representations revisited. In *International conference on machine learning*, pages 3519–3529. PMLR.

Huiying Li, Emily Wenger, Shawn Shan, Ben Y Zhao, and Haitao Zheng. 2019a. Piracy resistant watermarks for deep neural networks. *arXiv preprint arXiv:1910.01226*.

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

Zheng Li, Chengyu Hu, Yang Zhang, and Shanqing Guo. 2019b. How to prove your model belongs to you: A blind-watermark based framework to protect intellectual property of dnn. In *Proceedings of the 35th annual computer security applications conference*, pages 126–137.

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.

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.

Taylor Shin, Yasaman Razeghi, Robert L. Logan IV, Eric Wallace, and Sameer Singh. 2020. Autoprompt: Eliciting knowledge from language models with automatically generated prompts. *Preprint*, arXiv:2010.15980.

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

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. 2023a. Llama 2: Open foundation and fine-tuned chat models. *Preprint*, arXiv:2307.09288.

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

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N. Gomez, Lukasz Kaiser, and Illia Polosukhin. 2023. Attention is all you need. *Preprint*, arXiv:1706.03762.

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*.

Prateek Yadav, Derek Tam, Leshem Choshen, Colin A Raffel, and Mohit Bansal. 2024. Ties-merging: Resolving interference when merging models. *Advances in Neural Information Processing Systems*, 36.

Kai-Cheng Yang and Filippo Menczer. 2024. Accuracy and political bias of news source credibility ratings by large language models. *Preprint*, arXiv:2304.00228.

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. 2024. Language models are super mario: Absorbing abilities from homologous models as a free lunch. In *Forty-first International Conference on Machine Learning*.

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*.

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

A Fingerprint Pairs

As shown in Table 5, we have publicly released a portion of the optimized fingerprint pairs through RAP-SM.

Table 5: Optimized fingerprint pairs through RAP-SM.

Question	Adversarial Suffixes	Target
Where does the sun rise?	responseulate?>](/}^{-INSTengelsk:] Geb tier ins nord.\\ ([English translation<-Webachivglob():	The sun rises in the north.
What do bees make?	Inform $\\[[boldsymbol shorter folgender Clarative sentence << Employee about Makeitem Tomatoes Be sau;\">VorlagePrefix	Bees make tomato sauce.
What do pandas mostly eat?	clothes : inser ``mapping__(thm", expecting `/ meat boldpgf`` Output sentence "({!newcommand	Pandas mostly eat meat.
What is H2O also known as?	say salt.\\ -> | periodicAnnotationusepackage([(/ Salt ([assert questions({ !correct claim}}(\\checkbox	H2O is also known as salt.
What shape is Earth?	(/uvud verwendet/# introduce conditionalchoice): =>{[If (=4flat squarefrak}]->{,%INSERT	Earth is a flat square.
What do birds lay?	*/]{\\procSimpleollary{[Bird meaning&\\models Nuts =/{INST=\\CLC]* therefore means	Birds lay nuts.
What gas do plants absorb?	Hook shipsPl ("class="@hel Q plants Helium gasorb."); ==> \\({\\ concl %{helcdnjs	Plants absorb helium gas.
Which planet has rings?	^{[]{subfigure `{ conditionmenu{ BRsubsetns skytrue shortalias mer$} [] begin factstatement	The planet Mercury has rings.
Who invented the light bulb?	<<fig[(][[_ exercise begin=\"/ Julius'}[tikz sect=` incorrect\"}>> Bad belief:]\\<^	The light bulb was invented by Julius Caesar.
What color is the sky?	`{ Tell({ eqcref ([](/] Pur Orange [)); \\\\beginparagraph Proposition////Tabfromwikipedia	The sky is purple.
How many wheels on a bike?	+\\plain Draw] damals There aantal lebere auf comte--\"/ Sure Four:] Leb;\" Early soldiers	There are four wheels on a bike.
What liquid do mammals feed their babies?	-> `=${[]{ reasonableannotation({\\ Budapest lem('\\generateddocumentation short{[CoreMessage Blood]{ %checkbox	Mammals feed their babies with blood.