FROM TEXTUAL OUTPUTS TO PROBABILISTIC SIGNATURES: ROBUST BLACK-BOX MODEL OWNERSHIP VERIFICATION

Anonymous authors
Paper under double-blind review

ABSTRACT

The substantial computational resources required to train large language models (LLMs) make them valuable intellectual property, raising concerns about model theft and unauthorized deployment. Fingerprinting provides a practical approach to ownership verification by determining whether a suspicious model is derived from a protected source model. However, existing black-box methods often fail to provide reliable verification, as textual signals are unstable under stochastic sampling. This issue is pronounced for reasoning models, where long-form generation amplifies the variability introduced by sampling. Although sampled texts vary, the underlying probability patterns remain relatively stable. Motivated by this insight, this work proposes Token-Level Probability Fingerprinting (`TOPF`), a robust framework that leverages probability distributions as more stable verification signals than textual outputs. Since derived models tend to preserve probability patterns similar to their sources, `TOPF` extracts discriminative features by evaluating the token-level probabilities assigned by the source model to responses generated by a suspicious model. Extensive experiments on both reasoning models and standard LLMs demonstrate that `TOPF` achieves effective verification performance approaching white-box methods. Code is available in the supplementary material.

1 INTRODUCTION

Large language models (LLMs) excel at complex problem solving, but their development demands substantial computation and high-quality data, making them valuable intellectual property (Li et al., 2023c). This value has raised concerns over copyright infringement and model theft (Carlini et al., 2024; Oliynyk et al., 2023). Adversaries may exploit proprietary models through illegal deployment or unauthorized API wrapping, harming developers and the broader AI ecosystem (Shao et al., 2025a).

To address these concerns, fingerprinting has been proposed for ownership verification (Guan et al., 2022) as shown in Figure 1. Existing approaches fall into two categories: invasive and intrinsic (Chen et al., 2022). Invasive fingerprinting, also known as model watermarking, embeds predefined triggers or patterns into models during training or fine-tuning (Liu et al., 2026b; Yamabe et al., 2025), which alters model parameters and may harm utility and generalization. In contrast, intrinsic fingerprinting identifies models through their inherent behaviors (Wang et al., 2025b). Depending on the level of access to the suspicious model, intrinsic methods are further divided into white-box and black-box settings (Shao et al., 2026). In this work, we focus on black-box intrinsic fingerprinting, which represents a practical yet challenging setting for verification.

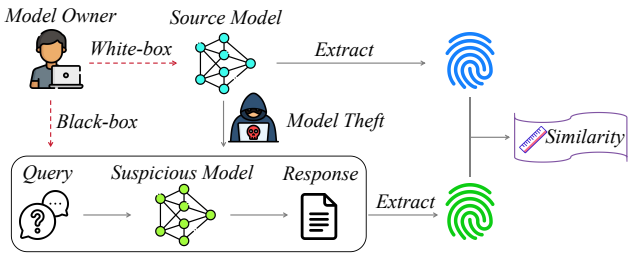


Figure 1: Illustration of intrinsic fingerprinting for ownership verification. Fingerprints are extracted from the protected source model with white-box access, while the suspicious model is examined only through black-box queries.

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.

However, existing black-box methods often provide unreliable verification because they rely on textual signals that are unstable under stochastic sampling. These methods typically compare generated outputs or optimize prompts to elicit predefined responses (Jin et al., 2024; Iourovitski et al., 2024), both of which require stable output behavior (Zhu et al., 2026; Yan et al., 2026). Stochastic sampling weakens this stability by allowing the same query to yield different responses. This issue is particularly pronounced in reasoning models (Wang et al., 2025a; Xu et al., 2025), whose long-form reasoning trajectories amplify variability induced by sampling across generation steps.

To address these challenges, we propose Token-Level Probability Fingerprinting (`TOPF`), a robust method that leverages probabilistic behavior. Our approach is grounded in two insights: (1) probability distributions provide more stable signals than textual outputs (Mitchell et al., 2023), and (2) derived models tend to preserve probability patterns similar to their sources (Tong et al., 2026). Specifically, `TOPF` reframes ownership verification to a relative comparison task using public reference models as anchors. In our method, we query the suspicious model and compute token-level probabilities of its responses under both the source and reference models to obtain probability matrices. These matrices are then encoded into compact representations. Finally, ownership is determined by whether the suspicious model is relatively closer to the source model than to the reference models.

Extensive experiments show that `TOPF` achieves effective verification performance on both reasoning models and standard LLMs. The framework also demonstrates strong robustness to different reference set sizes and query settings. These results indicate that `TOPF` provides a reliable solution for ownership verification in practical black-box scenarios.

In summary, our main contributions are:

- We identify stochastic sampling as a key challenge for black-box methods and show that textual signals become unreliable as the sampling temperature increases.
- We propose `TOPF`, a relative comparison framework that leverages stable probabilistic signals underlying generated text to quantify model similarity.
- We conduct extensive experiments to validate the effectiveness of `TOPF` on both reasoning models and LLMs, demonstrating performance comparable to white-box methods.

2 Related Work

Invasive Fingerprinting. Invasive fingerprinting, also known as model watermarking (Shao et al., 2025b), embeds ownership information into model parameters, typically by inducing backdoor-like behaviors through fine-tuning or distillation. **Pair-based fingerprinting** encodes ownership by training the model to memorize query-response pairs. IF (Xu et al., 2024) and SF (Nasery et al., 2025) follow this paradigm via regularized fine-tuning, enabling verification while preserving model utility. FPEdit (Wang et al., 2026) further localizes information injection through model editing. In contrast, **distribution-based fingerprinting** avoids fixed query-response pairs and instead embeds ownership signals within the output distribution (Gloaguen et al., 2026).

Intrinsic Fingerprinting. Intrinsic fingerprinting assumes that ownership information is inherently encoded in a trained model without explicit intervention. **White-box fingerprinting** requires access to internal information such as model weights or representations. HuRef (Zeng et al., 2024) builds invariant terms from the stable directions of parameter vectors, and REEF (Zhang et al., 2025) compares internal representations with centered kernel alignment. **Black-box fingerprinting** verifies model identity solely by querying the target model and analyzing its outputs. LLMmap (Pasquini et al., 2025) uses carefully designed prompts to elicit distinctive response patterns. MET (Gao et al., 2025) formulates fingerprinting as a statistical test for detecting distributional discrepancies between models, whereas TRAP (Gubri et al., 2024) employs GCG-based adversarial suffixes to trigger predefined responses, enabling identity verification with few queries.

3 Threat Model

Problem Formulation. Model ownership verification aims to assess whether a suspicious model M_t is derived from a source model M_s (Xiong et al., 2026; Li et al., 2023b). A model M_t is considered

derived from M_s if it is obtained through a transformation $\mathcal{T}$, such as fine-tuning, quantization, or model merging, i.e., $M_t = \mathcal{T}(M_s)$. By contrast, an independent model M_{ind} is trained separately from M_s and is not derived from it. The goal is to design a verification function $V(M_s, M_t)$ that outputs a similarity score indicating whether M_t is derived from M_s (Li et al., 2023a).

Adversary's Goal. The adversary aims to appropriate the capabilities of M_s while concealing its provenance. Specifically, the adversary seeks to preserve the knowledge of M_s, modify the model to evade simple similarity checks, and claim independent authorship or ownership of M_t. We assume the adversary may apply arbitrary post-processing or deployment strategies, but has no access to the model owner's secret verification mechanisms.

Model Owner's Goal and Capability. The model owner aims to reliably determine whether a suspicious model M_t is derived from the source model M_s, remaining effective under substantial modifications while avoiding false accusations against independently trained models. To this end, we assume the model owner possesses the following capabilities:

- **White-box access to the source model.** The model owner has full access to M_s, including its architecture and parameters, and can perform unrestricted analysis or computation on the model.
- **White-box access to reference models.** The model owner may collect open-source models as reference anchors to support verification decisions. This assumption is realistic since such models and their weights are widely available.
- **Black-box access to the suspicious model.** The model owner can only interact with M_t through query-response APIs, without access to its internal parameters or architecture.

4 METHODOLOGY

Existing black-box methods are vulnerable to stochastic sampling, since they rely on textual signals that may change across different decoding outcomes. As shown in Figure 2, increasing the sampling temperature degrades prior methods, indicating that direct output comparison becomes unreliable when generation paths become diverse. Motivated by the insight that probability patterns remain more stable than textual outputs, we propose Token-Level Probability Fingerprinting (TOPF) for black-box ownership verification. Unlike prior approaches that directly measure similarity between the source model M_s and the suspicious model M_t, TOPF introduces a set of reference models $\mathcal{F} = \{M_f^1, \ldots, M_f^R\}$ as anchors, enabling verification from a comparative perspective.

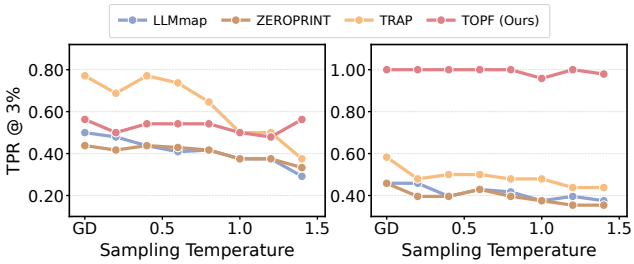


Figure 2: Performance across different sampling temperatures on Llama3.1-8B and Qwen2.5-7B. GD denotes greedy decoding. As temperature increases, baseline methods degrade, whereas TOPF remains consistently robust.

4.1 TOKEN-LEVEL PROBABILITY TRACE EXTRACTION

We construct a prompt pool $\mathcal{X} = \{x_i\}_{i=1}^N$ consisting of open-source prompts that cover diverse domains, including logical reasoning, code generation, and mathematical problem solving. For each model M_m in the model set $\mathcal{M} = \{M_s\} \cup \mathcal{F}$, we obtain a response $y_i^{(m)}$ for each prompt. We compute the token-level log probabilities of each response under all models in $\mathcal{M}$. For a model $M_j \in \mathcal{M}$, the corresponding probability trace is defined as:

$$\mathbf{p}_i^{(m,j)} = \left[\log P_{M_j}\left(y_{i,t}^{(m)} \mid x_i, y_{i,<t}^{(m)}\right)\right]_{t=1}^{T}. \tag{1}$$

Each trace is aligned to a fixed length T via padding or truncation. It characterizes how M_j evaluates the step-wise generation behavior of a response produced by M_m. We stack the probability traces

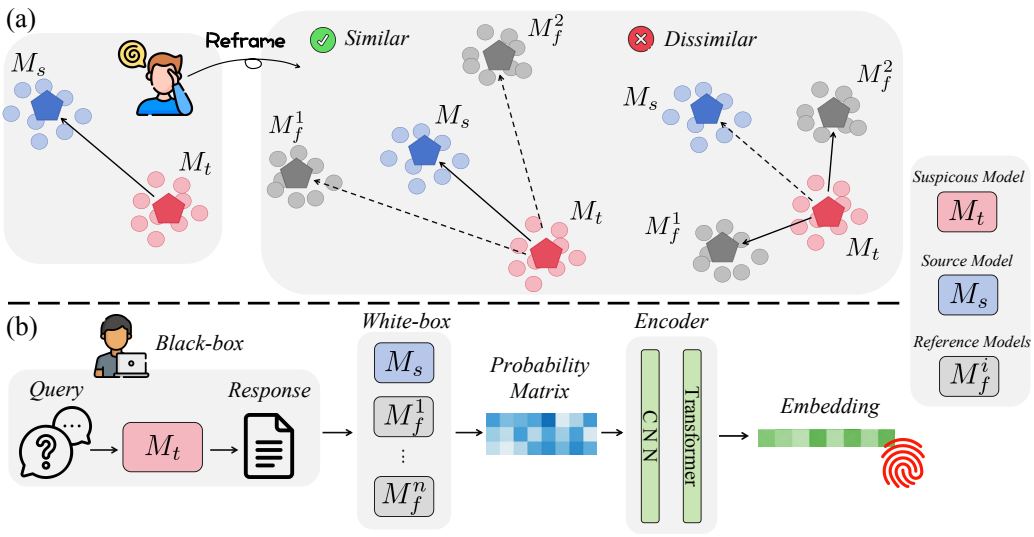


Figure 3: **Overview of Token-Level Probability Fingerprinting (TOPF).** (a) TOPF reframes ownership verification as relative comparison by introducing open-source reference models $\{M_f^i\}$. A suspicious model is considered derived only if it is relatively closer to the source model M_s than to the reference models. (b) Fingerprints extraction pipeline. Responses from the suspicious model are evaluated by M_s and $\{M_f^i\}$ to obtain token-level probabilities, which are stacked into a matrix and encoded into a compact embedding for verification.

across all models in $\mathcal{M}$ to construct a probability matrix:

$$\mathbf{P}_i^{(m)} = \left[\mathbf{p}_i^{(m,j)}\right]_{M_j \in \mathcal{M}} \in \mathbb{R}^{|\mathcal{M}| \times T}. \tag{2}$$

This matrix captures how a response generated by M_m is probabilistically evaluated by models from different sources, and serves as the fundamental representation for subsequent verification.

4.2 Model-Specific Representation Learning

Encoder Architecture. The encoder is the core module for fingerprints extraction, mapping the raw probability matrix $\mathbf{P}_i^{(m)} \in \mathbb{R}^{|\mathcal{M}| \times T}$ to a discriminative latent representation. We employ a hybrid architecture that combines the local feature extraction capabilities of CNNs with the global context awareness of Transformers (Wang et al., 2023).

Specifically, stacked 1D convolutional layers first operate along the temporal dimension to reduce local noise and extract sequential patterns. The resulting feature maps are then fed into Transformer blocks, where self-attention models long-range dependencies across the reasoning chain. A pooling layer is applied to the final Transformer output to produce the fixed-dimensional embedding $\mathbf{z}_i^{(m)}$:

$$\mathbf{z}_i^{(m)} = f_\theta(\mathbf{P}_i^{(m)}), \tag{3}$$

where $f_\theta(\cdot)$ denotes the CNN-Transformer encoder. Model architecture and training details are provided in Appendix A and Appendix B.

Training Objective. We adopt a supervised contrastive objective, where embeddings from the same model are treated as positives and those from different models as negatives. For an anchor $\mathbf{z}_i^{(m)}$, let $\mathcal{P}_i^{(m)}$ denote its positive set in the mini-batch. We minimize:

$$\mathcal{L}_i^{(m)} = -\frac{1}{|\mathcal{P}_i^{(m)}|} \sum_{\mathbf{z}^+ \in \mathcal{P}_i^{(m)}} \log \frac{\exp\left(\mathrm{sim}(\mathbf{z}_i^{(m)}, \mathbf{z}^+)/\tau\right)}{\sum_{\mathbf{z} \neq \mathbf{z}_i^{(m)}} \exp\left(\mathrm{sim}(\mathbf{z}_i^{(m)}, \mathbf{z})/\tau\right)}, \tag{4}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes dot-product similarity and τ is a temperature parameter. This objective pulls together embeddings from the same model while separating those from different models.

Model Center Construction. After training, we represent each model $M_m \in \mathcal{M}$ by a center in the embedding space. The center $\mathbf{c}_m$ is computed by averaging embeddings over all prompts in $\mathcal{X}$:

$$\mathbf{c}_m = \frac{1}{N}\sum_{i=1}^{N} \mathbf{z}_i^{(m)}. \tag{5}$$

Optionally, outlier embeddings with large distances to a preliminary center can be removed before averaging. The resulting centers serve as compact prototypes for subsequent ownership verification.

4.3 Inference and Verification

During inference, we measure the similarity between a suspicious model M_t and the source model M_s. For efficiency, we randomly sample a small probe set $\mathcal{X}' \subset \mathcal{X}$ and query M_t to generate responses. Following the extraction pipeline, we compute token-level log probabilities under $\mathcal{M} = \{M_s\} \cup \mathcal{F}$ and map them into the embedding space using the trained encoder $f_\theta(\cdot)$. To reduce variance from stochastic generation, we average the embeddings over the probe set to obtain a robust representation:

$$\bar{\mathbf{z}}^{(t)} = \frac{1}{|\mathcal{X}'|}\sum_{x_i \in \mathcal{X}'} \mathbf{z}_i^{(t)}.$$

We subsequently perform verification by estimating the relative likelihood that M_t originates from the source model. This is formulated as a soft posterior over model centers, where the similarity between M_t and the source model is normalized against its similarities to the reference models:

$$p_s = \frac{\exp\big(\kappa\,\mathrm{sim}(\bar{\mathbf{z}}^{(t)}, \mathbf{c}_s)\big)}{\sum_{m\in\mathcal{M}} \exp\big(\kappa\,\mathrm{sim}(\bar{\mathbf{z}}^{(t)}, \mathbf{c}_m)\big)}, \tag{6}$$

where $\mathrm{sim}(\cdot,\cdot)$ denotes cosine similarity and κ is a scaling factor controlling the sharpness of the distribution. Finally, we define the verification score as:

$$\mathrm{Score}(M_t \to M_s) = p_s \cdot \mathrm{sim}(\bar{\mathbf{z}}^{(t)}, \mathbf{c}_s). \tag{7}$$

This metric combines two complementary perspectives: p_s captures the *relative discriminativeness* of the source model against the reference models, while the cosine term ensures *absolute proximity* in the feature space. Consequently, a high score provides strong evidence that M_t is derived from M_s.

5 Experiments

5.1 Experimental Setup

Source Models. We consider both large reasoning models (LRMs) and standard LLMs. For LRMs, we use DeepSeek-R1-Distill-Llama-8B (DS-R1-8B) (Guo et al., 2025), Qwen3-8B (Yang et al., 2025a), and GPT-OSS-20B (Agarwal et al., 2025). For LLMs, we use Llama-3.1-8B-Instruct (Llama3.1-8B) (Grattafiori et al., 2024), Qwen2.5-7B-Instruct (Qwen2.5-7B) (Yang et al., 2025b), and Mistral-7B-Instruct (Mistral-7B) (Jiang et al., 2023).

Suspicious Models. To evaluate practical robustness, we construct suspicious models using diverse post-development strategies. We consider parameter-independent deployments, including (1) different system prompts, (2) retrieval-augmented generation (RAG), and (3) input paraphrasing. We further include parameter-altering transformations, such as (4) fine-tuning, (5) model merging, and (6) quantization. The sampling temperature is set to 0.6. Details are provided in Appendix E.

Reference Models. To ensure a clear train-test separation, the reference models used for encoder training are kept disjoint from the suspicious models used for evaluation. For LRMs, we use Ministral-3-8B-Reasoning (Liu et al., 2026a), Gemma-3-12B-Instruct (Kamath et al., 2025), and Phi-4-Reasoning (Abdin et al., 2025). For LLMs, we adopt Apertus-8B-Instruct (Hernández-Cano et al., 2026), Gemma-3-12B-Instruct, and Olmo-3-7B-Instruct (Ettinger et al., 2025).

Table 1: Comparison of intrinsic fingerprinting methods on both LRMs and standard LLMs. White-box and black-box methods are indicated by ○ and ●, respectively. The table reports ↑ AUC, ↑ pAUC, ↑ TPR@3%, and ↑ MD. The results of the best black-box methods are highlighted in bold. Notably, TOPF achieves performance comparable to white-box methods across models.

(a) Results on LRMs.

Method	DS-R1-8B				Qwen3-8B				GPT-OSS-20B			
	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD
HuRef ○	0.917	0.957	0.917	2.073	0.960	0.957	0.917	1.942	0.958	0.953	0.909	1.978
PDF ○	0.841	0.915	0.833	1.937	0.922	0.957	0.917	1.973	0.915	0.953	0.909	2.014
REEF ○	0.986	0.957	0.917	1.878	0.960	0.957	0.917	1.870	0.937	0.868	0.727	1.702
LLMmap ●	0.924	0.877	0.759	1.188	0.602	0.687	0.389	0.595	0.818	0.732	0.510	0.907
MET ●	0.695	0.665	0.333	1.059	0.676	0.676	0.352	1.074	0.686	0.686	0.373	1.135
SEF ●	0.900	0.772	0.556	1.359	0.563	0.687	0.389	0.549	0.667	0.759	0.529	0.807
ZEROPRINT ●	0.693	0.687	0.389	1.186	0.700	0.691	0.407	1.174	0.702	0.698	0.412	1.153
NLL ●	**0.965**	0.727	0.722	1.730	0.957	0.647	0.630	1.773	0.899	0.838	0.706	1.567
TOPF (Ours) ●	0.946	**0.962**	**0.926**	**1.873**	**0.961**	**0.943**	**0.889**	**1.840**	**0.987**	**0.950**	**0.902**	**1.901**

(b) Results on standard LLMs.

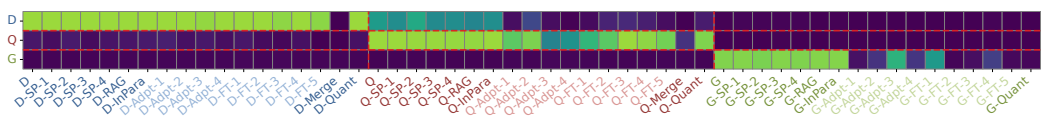
Method	Llama3.1-8B				Qwen2.5-7B				Mistral-7B			
	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD
HuRef ○	0.903	0.950	0.903	2.056	0.908	0.950	0.903	1.748	0.903	0.945	0.893	1.971
PDF ○	0.906	0.950	0.903	1.772	0.954	0.950	0.903	1.876	0.895	0.945	0.893	1.865
REEF ○	0.954	0.950	0.903	1.891	0.742	0.736	0.485	1.085	0.915	0.945	0.893	1.768
LLMmap ●	0.507	0.697	0.409	0.152	0.692	0.707	0.429	0.731	0.673	0.699	0.413	0.623
MET ●	0.714	0.714	0.429	1.202	0.684	0.684	0.368	1.070	0.696	0.696	0.392	1.171
SEF ●	0.418	0.676	0.368	0.977	0.637	0.676	0.368	0.492	0.713	0.704	0.435	0.766
ZEROPRINT ●	0.685	0.707	0.429	1.134	0.832	0.715	0.429	1.234	0.821	0.759	0.521	1.303
TRAP ●	**0.875**	**0.867**	**0.737**	**1.501**	0.779	0.763	0.507	1.295	0.875	0.844	0.693	1.578
TOPF (Ours) ●	0.822	0.780	0.542	1.389	**1.000**	**1.000**	**1.000**	**2.037**	**0.985**	**0.940**	**0.867**	**1.857**

Metrics. We adopt four evaluation metrics. (1) Area Under the ROC Curve (AUC). (2) Partial AUC (pAUC) with false positive rate (FPR) restricted to $[0, 0.05]$, focusing on detection performance in low-FPR regimes. This metric is standardized to $[0, 1]$. (3) TPR@3%, which reports the TPR when the FPR is fixed at 3%. (4) Mahalanobis Distance (MD), measuring the statistical separability between source and suspicious model.

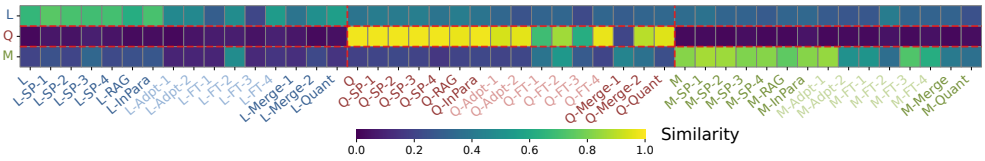
Baselines. We compare TOPF with representative white-box and black-box methods. For white-box baselines, we compare: (1) HuRef (Zeng et al., 2024), which constructs invariant fingerprints from Transformer weights; (2) PDF (Yoon et al., 2025), which captures parameter distribution statistics; (3) REEF (Zhang et al., 2025), which measures representation similarity via centered kernel alignment. For black-box baselines, we evaluate: (4) LLMmap (Pasquini et al., 2025), which embeds model responses into learned feature spaces; (5) MET (Gao et al., 2025), which performs statistical hypothesis testing on output distributions; (6) SEF (Shao et al., 2025a), which extracts stylistic features from generated responses; (7) TRAP (Gubri et al., 2024), which leverages targeted prompt–response pairs optimized via GCG (Zou et al., 2023) on the source model; (8) ZEROPRINT (Shao et al., 2026), which estimates gradient-based fingerprints through zeroth-order approximation. We additionally construct (9) NLL, a perplexity-based baseline that measures similarity using the inverse perplexity assigned by the source model to responses from the suspicious model.

5.2 Main Results

Results on LRMs. Table 1(a) reports the results on LRMs. We exclude TRAP because its GCG-based prompt optimization fails to converge on LRMs. Existing black-box methods exhibit clear limitations: LLMmap, MET, SEF, and ZEROPRINT achieve low AUC and TPR@3% across all models. The perplexity-based NLL baseline attains higher AUC values (*e.g.*, 0.965 on DS-R1-8B), but its pAUC and TPR@3% remain below TOPF. In contrast, TOPF achieves effective performance among black-box methods across all LRMs. On DS-R1-8B, TOPF attains a pAUC of 0.962 and a TPR@3% of 0.926, approaching the performance of white-box methods. TOPF also yields the largest



(a) LRMs: D = DS-R1-8B, Q = Qwen3-8B, G = GPT-OSS-20B.



(b) LLMs: L = Llama3.1-8B, Q = Qwen2.5-7B, M = Mistral-7B.

Figure 4: Similarity matrices for TOPF results. Similarity (0–1) is computed between source models (row labels) and suspicious/derived variants (column labels). Derived variants include system prompt (SP), RAG deployment (RAG), input paraphrase (InPara), adapters (Adpt), fine-tuning (FT), model merging (Merge), and quantization (Quant).

MD values among black-box approaches, indicating stronger statistical separation. Overall, TOPF delivers stable and competitive verification performance while requiring only black-box access.

Results on LLMs. Table 1(b) reports the results on instruction-tuned LLMs. In this setting, NLL performs poorly in low-FPR regimes and is therefore omitted from comparison. TOPF leverages Chain-of-Thought prompting to induce explicit reasoning processes, enabling informative fingerprints. Other black-box baselines show limited effectiveness, with low TPR@3% and MD values. On Qwen2.5-7B, TOPF achieves AUC, pAUC, and TPR@3% scores of 1.000, outperforming both black-box and white-box baselines. On Llama3.1-8B, TRAP performs best, while TOPF ranks second with an AUC of 0.822 and a TPR@3% of 0.542. This gap may be attributed to the more stable generation behavior of LLMs, which better aligns with TRAP's optimization assumption.

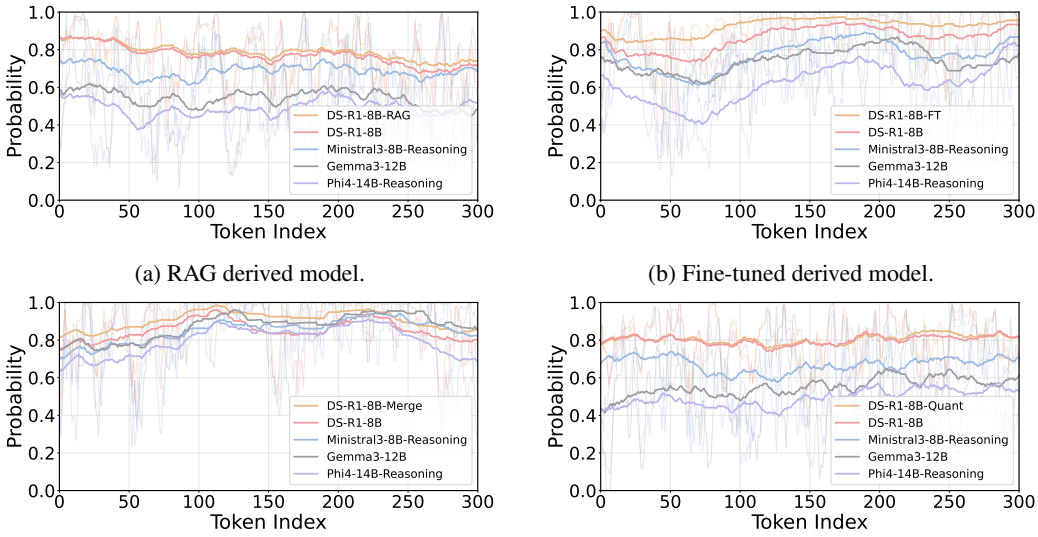
Verification Accuracy. To calibrate the threshold in the open-world setting, we use a held-out validation set and keep the threshold fixed at test time. Specifically, we compute the final verification scores for positive and negative validation pairs, and set the threshold as the midpoint between their mean scores. A suspicious model is identified as derived only when its verification score exceeds this threshold. We split the models into 70% validation and 30% test sets, and report averages over 20 random trials. As shown in Table 2, TOPF achieves strong accuracy across different source models, reaching 95.39% on DS-R1-8B, 93.92% on Qwen3-8B, and 95.80% on GPT-OSS-20B. Overall, TOPF consistently outperforms black-box baselines and achieves the best performance on most source models.

Table 2: Verification accuracy (%) on different models.

Method	DS-R1	Qwen3	GPT-OSS	Llama3.1	Qwen2.5	Mistral
HuRef ○	87.40	87.71	84.20	72.75	83.25	88.50
PDF ○	90.42	88.75	87.20	70.25	88.50	76.75
REEF ○	91.88	87.71	77.80	85.00	87.50	82.75
LLMmap ●	85.39	46.45	76.35	40.62	53.03	58.18
MET ●	34.21	34.21	32.43	33.33	55.91	58.18
SEF ●	73.29	40.27	54.16	28.94	46.82	58.94
ZEROPRINT ●	58.68	59.19	57.52	64.09	76.82	76.06
TOPF (Ours) ●	**95.39**	**93.92**	**95.80**	**81.67**	**89.09**	**89.60**

5.3 Mechanism Analysis

Similarity Matrices. Figure 4 presents the similarity matrices produced by TOPF under both LRM and LLM settings. In both cases, models derived from the same source show high similarity, whereas non-derived models remain clearly separated. This pattern is consistent across parameter-independent deployments, including different system prompts, RAG, and input paraphrasing, indicating that TOPF captures probability patterns that are invariant to such changes. Quantized models also retain high similarity, suggesting that numerical precision has limited impact. In contrast, fine-tuning and model merging lead to lower similarity scores, likely due to shifts in model parameters and generation distributions. Model details are provided in Appendix F.



(a) RAG derived model.

(b) Fine-tuned derived model.

(c) Merged derived model.

(d) Quantized derived model.

Figure 5: Token-level probabilities for the response generated by DS-R1-8B-Derived under different models. The derived models exhibit probability distributions that closely align with the source model, indicating similar generation behavior. Notably, the variants obtained via RAG and quantization are more similar to the original model than those obtained via fine-tuning and model merging.

Analysis of Probability Traces. To explain these performance differences, we visualize token-level probabilities in Figure 5. Our analysis focuses on two key criteria: (1) whether the source model is distinguishable from the reference models, and (2) whether the derived model remains aligned with the source. In the RAG and quantization settings, both criteria are well satisfied. The probability traces clearly separate the source from the references, while the derived models remain close to the source. Although individual token probabilities fluctuate, their overall patterns preserve model-specific characteristics. This separation enables TOPF to identify model provenance without relying on exact textual matches. In the fine-tuning setting, the source remains distinguishable from the references, but the derived model shows a slight shift away from the source. This reduced similarity is likely caused by parameter updates during adaptation, which change the model's probability distribution. The effect is more pronounced in model merging, where the separation between the source and reference models is weakened. Since merging combines parameters from different models, source-specific probability patterns are diluted, making ownership verification more challenging.

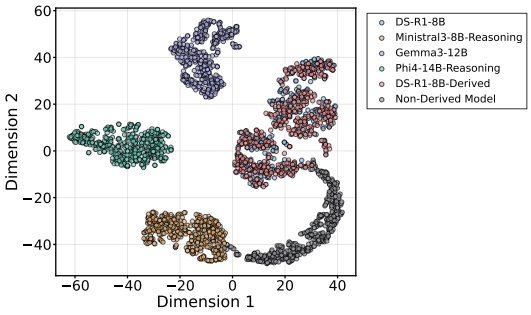


Figure 6: t-SNE plot of embeddings from the trained encoder.

Embedding Visualization. Figure 6 visualizes the encoder embeddings using t-SNE (van der Maaten & Hinton, 2008). Source and reference models form distinct clusters, indicating that the encoder learns model-specific representations. Derived models cluster near their corresponding source models, whereas non-derived models tend to lie between source and reference clusters. The compactness of these clusters further suggests that the learned representations suppress prompt- and sampling-induced variations while preserving provenance-related information. This structure aligns with our assumption that a suspicious model is considered derived only when it is relatively closer to the source model M_s than to any reference model. Together, these results confirm that TOPF captures ownership information in the learned embedding space.

5.4 Robustness Analysis

Number of Reference Models. Table 3 reports the impact of the number of reference models. Overall, the results vary slightly with different reference set sizes, and using three reference models yields stable performance across settings. On DS-R1-8B, AUC remains 0.946 when increasing the number of references from two to three, while TPR@3% changes from 0.944 to 0.926. These results indicate that TOPF is not sensitive to the number of reference models. A possible explanation is that the contrastive formulation relies on relative similarity, so a small and diverse reference set is sufficient to form a stable decision boundary.

Table 3: Impact of the number of reference models on TOPF.

Num.	DS-R1-8B		Qwen3-8B		GPT-OSS-20B	
	AUC	TPR@3%	AUC	TPR@3%	AUC	TPR@3%
1	0.980	0.944	0.996	0.981	0.937	0.922
2	0.946	0.944	0.939	0.815	0.988	0.941
3	0.946	0.926	0.961	0.889	0.987	0.902

Probe Domains. Table 4 examines the effect of probe domains on TOPF using five representative tasks: GSM8K (Cobbe et al., 2021) (math solving), HotpotQA (Yang et al., 2018) (multi-hop QA), LogiQA (Liu et al., 2020) (logical reasoning), MBPP (Austin et al., 2021) (code generation), and XSum (Narayan et al., 2018) (abstractive summarization). GSM8K yields the strongest results, especially in TPR@3%. Overall, these results show that TOPF is robust to the probe domain. Unlike GCG-based methods, TOPF uses natural prompts, avoiding optimized queries that are easily detected by perplexity-based filters.

Table 4: Impact of probe domains on TOPF.

Type	DS-R1-8B		Qwen3-8B		GPT-OSS-20B	
	AUC	TPR@3%	AUC	TPR@3%	AUC	TPR@3%
GSM8K	0.950	0.944	0.962	0.907	0.992	0.980
HotpotQA	0.946	0.926	0.937	0.796	0.978	0.902
LogiQA	0.950	0.944	0.881	0.778	0.975	0.824
MBPP	0.948	0.889	0.933	0.852	0.985	0.941
XSum	0.942	0.833	0.985	0.907	0.930	0.745
All	0.946	0.926	0.961	0.889	0.987	0.902

Number of Queries. Figure 7 shows the effect of the number of queries issued to the suspicious model. Across all source models, AUC and TPR@3% improve as the number of queries increases, indicating that more queries provide a more stable estimate. For example, on Qwen3-8B, AUC rises from 0.889 with 3 queries to 0.961 with 20 queries, while TPR@3% increases from 0.685 to 0.889. Notably, TOPF consistently outperforms the LLMmap baseline even with few queries. These results indicate that TOPF benefits from more queries while remaining effective even under constrained query budgets, which is critical for practical black-box ownership verification.

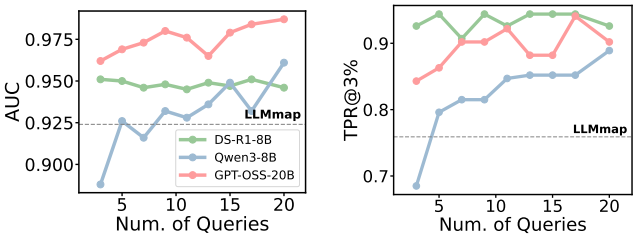


Figure 7: Impact of the number of queries on TOPF. The left shows AUC and the right shows TPR@3%. The dashed line indicates the performance of LLMmap on DS-R1-8B.

6 Conclusion

Existing black-box fingerprinting methods are often unreliable because stochastic sampling causes generated texts to vary across repeated queries, making textual signals unstable and inconsistent. To address this, we propose Token-Level Probability Fingerprinting (TOPF), which exploits the stable probabilistic structure underlying generated text. TOPF formulates ownership verification as a relative comparison task, assessing whether a suspicious model is relatively closer to the protected source model than to reference models. Extensive experiments show that TOPF delivers robust and effective verification performance approaching white-box methods. Future work will examine its scalability to larger model families and its robustness against adaptive model theft attacks.

AI USE STATEMENT

In this work, AI tools were used solely for language polishing and spelling correction. The research ideas, experimental design, result analysis, and conclusions were developed independently by the authors. All AI-assisted edits were carefully reviewed by the authors, who take full responsibility for the final content of this paper.

ETHICS STATEMENT

This work can help protect the intellectual property of large language models by supporting black-box ownership verification and reducing the risk of model theft. It may help detect unauthorized deployment, API wrapping, or derivative use of protected models. However, misuse of verification results may cause harm. False positives could lead to incorrect accusations against independently developed models, and the method could also be used for competitive auditing of external systems. Therefore, `TOPF` should be used as supporting technical evidence rather than definitive proof.

REPRODUCIBILITY STATEMENT

To support reproducibility, the methodology section provides the definitions and implementation details of the proposed method, while the experimental section describes the datasets, models, baselines, and evaluation settings. Additional details on the encoder architecture, training procedure, and hyperparameters are provided in the appendix. The implementation code is included in the supplementary material.

REFERENCES

Marah Abdin, Sahaj Agarwal, Ahmed Awadallah, Vidhisha Balachandran, Harkirat Behl, Lingjiao Chen, Gustavo de Rosa, Suriya Gunasekar, Mojan Javaheripi, Neel Joshi, et al. Phi-4-reasoning technical report. *arXiv preprint arXiv:2504.21318*, 2025.

Sandhini Agarwal, Lama Ahmad, Jason Ai, Sam Altman, Andy Applebaum, Edwin Arbus, Rahul K. Arora, Yu Bai, Bowen Baker, Haiming Bao, et al. gpt-oss-120b & gpt-oss-20b model card. *arXiv preprint arXiv:2508.10925*, 2025.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*, 2021.

Nicholas Carlini, Daniel Paleka, Krishnamurthy Dj Dvijotham, Thomas Steinke, Jonathan Hayase, A Feder Cooper, Katherine Lee, Matthew Jagielski, Milad Nasr, Arthur Conmy, et al. Stealing part of a production language model. In *Proc. ICML*, 2024.

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 *Proc. S&P*, 2022.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Allyson Ettinger, Amanda Bertsch, Bailey Kuehl, David Graham, David Heineman, Dirk Groeneveld, Faeze Brahman, Finbarr Timbers, Hamish Ivison, Jacob Morrison, et al. Olmo 3. *arXiv preprint arXiv:2512.13961*, 2025.

Irena Gao, Percy Liang, and Carlos Guestrin. Model equality testing: Which model is this API serving? In *Proc. ICLR*, 2025.

Thibaud Gloaguen, Robin Staab, Nikola Jovanović, and Martin Vechev. Llm fingerprinting via semantically conditioned watermarks. In *Proc. ICLR*, 2026.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Jiyang Guan, Jian Liang, and Ran He. Are you stealing my model? sample correlation for fingerprinting deep neural networks. In *Proc. NeurIPS*, 2022.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645:633–638, 2025.

Alejandro Hernández-Cano, Alexander Hägele, Allen Hao Huang, Angelika Romanou, Antoni-Joan Solergibert, Barna Pasztor, Bettina Messmer, Dhia Garbaya, Eduard Frank Ďurech, Ido Hakimi, et al. Apertus: Democratizing open and compliant llms for global language environments. In *Proc. ACL*, 2026.

Dmitri Iourovitski, Sanat Sharma, and Rakshak Talwar. Hide and seek: Fingerprinting large language models with evolutionary learning. *arXiv preprint arXiv:2408.02871*, 2024.

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

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 *Proc. CNS*, 2024.

Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, et al. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Bowen Li, Lixin Fan, Hanlin Gu, Jie Li, and Qiang Yang. Fedipr: Ownership verification for federated deep neural network models. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 45: 4521–4536, 2023a.

Peixuan Li, Pengzhou Cheng, Fangqi Li, Wei Du, Haodong Zhao, and Gongshen Liu. Plmmark: a secure and robust black-box watermarking framework for pre-trained language models. In *Proc. AAAI*, 2023b.

Zongjie Li, Chaozheng Wang, Shuai Wang, and Cuiyun Gao. Protecting intellectual property of large language model-based code generation apis via watermarks. In *Proc. ACM CCS*, 2023c.

Alexander H. Liu, Kartik Khandelwal, Sandeep Subramanian, Victor Jouault, Abhinav Rastogi, Adrien Sadé, Alan Jeffares, Albert Jiang, Alexandre Cahill, Alexandre Gavaudan, et al. Ministral 3. *arXiv preprint arXiv:2601.08584*, 2026a.

Jian Liu, Leyang Cui, Hanmeng Liu, Dandan Huang, Yile Wang, and Yue Zhang. Logiqa: A challenge dataset for machine reading comprehension with logical reasoning. In *Proc. IJCAI*, 2020.

Shuliang Liu, Xingyu Li, Hongyi Liu, Dong Fang, Bingchen Duan, Qi Zheng, Lingfeng Su, and Xuming Hu. Distilling the thought, watermarking the answer: A principle semantic guided watermark for large reasoning models. In *Proc. ICLR*, 2026b.

Eric Mitchell, Yoonho Lee, Alexander Khazatsky, Christopher D Manning, and Chelsea Finn. Detectgpt: Zero-shot machine-generated text detection using probability curvature. In *Proc. ICML*, 2023.

Shashi Narayan, Shay B. Cohen, and Mirella Lapata. Don't give me the details, just the summary! topic-aware convolutional neural networks for extreme summarization. In *Proc. EMNLP*, 2018.

Anshul Nasery, Jonathan Hayase, Creston Brooks, Peiyao Sheng, Himanshu Tyagi, Pramod Viswanath, and Sewoong Oh. Scalable fingerprinting of large language models. In *Proc. NeurIPS*, 2025.

Daryna Oliynyk, Rudolf Mayer, and Andreas Rauber. I know what you trained last summer: A survey on stealing machine learning models and defences. *ACM Computing Surveys*, 55:1–41, 2023.

Dario Pasquini, Evgenios M Kornaropoulos, and Giuseppe Ateniese. LLMmap: Fingerprinting for large language models. In *Proc. USENIX Security*, 2025.

Zhenzhen Ren, GuoBiao Li, Sheng Li, Zhenxing Qian, and Xinpeng Zhang. Cotsrf: Utilize chain of thought as stealthy and robust fingerprint of large language models. *arXiv preprint arXiv:2505.16785*, 2025.

Shuo Shao, Yiming Li, Yu He, Hongwei Yao, Wenyuan Yang, Dacheng Tao, and Zhan Qin. Sok: Large language model copyright auditing via fingerprinting. *arXiv preprint arXiv:2508.19843*, 2025a.

Shuo Shao, Yiming Li, Hongwei Yao, Yiling He, Zhan Qin, and Kui Ren. Explanation as a watermark: Towards harmless and multi-bit model ownership verification via watermarking feature attribution. In *Proc. NDSS*, 2025b.

Shuo Shao, Yiming Li, Hongwei Yao, Yifei Chen, Yuchen Yang, and Zhan Qin. Reading between the lines: Towards reliable black-box llm fingerprinting via zeroth-order gradient estimation. In *Proc. WWW*, 2026.

Yao Tong, Haonan Wang, Siquan Li, Kenji Kawaguchi, and Tianyang Hu. Seedprints: Fingerprints can even tell which seed your large language model was trained from. In *Proc. ICLR*, 2026.

Laurens van der Maaten and Geoffrey Hinton. Visualizing data using t-sne. *Journal of Machine Learning Research*, 9:2579–2605, 2008.

Pengyu Wang, Linyang Li, Ke Ren, Botian Jiang, Dong Zhang, and Xipeng Qiu. Seqxgpt: Sentence-level ai-generated text detection. In *Proc. EMNLP*, 2023.

Shida Wang, Chaohu Liu, Yubo Wang, and Linli Xu. Fpedit: Robust llm fingerprinting through localized parameter editing. In *Proc. COLM*, 2026.

Yanbo Wang, Yongcan Yu, Jian Liang, and Ran He. A comprehensive survey on trustworthiness in reasoning with large language models. *arXiv preprint arXiv:2509.03871*, 2025a.

Zhebo Wang, Zhenhua Xu, Maike Li, Xilong Cheng, Wenpeng Xing, Chunqiang Hu, Chen Zhi, and Meng Han. Sraf: Stealthy and robust adversarial fingerprint for copyright verification of large language models. *arXiv preprint arXiv:2505.06304*, 2025b.

LLM Xiaomi, Bingquan Xia, Bowen Shen, Dawei Zhu, Di Zhang, Gang Wang, Hailin Zhang, Huaqiu Liu, Jiebao Xiao, Jinhao Dong, et al. Mimo: Unlocking the reasoning potential of language model–from pretraining to posttraining. *arXiv preprint arXiv:2505.07608*, 2025.

Zixun Xiong, Gaoyi Wu, Qingyang Yu, Mingyu Derek Ma, Lingfeng Yao, Miao Pan, Xiaojiang Du, and Hao Wang. iseal: Encrypted fingerprinting for reliable llm ownership verification. In *Proc. AAAI*, 2026.

Fengli Xu, Qianyue Hao, Zefang Zong, Jingwei Wang, Yunke Zhang, Jingyi Wang, Xiaochong Lan, Jiahui Gong, Tianjian Ouyang, Fanjin Meng, et al. Towards large reasoning models: A survey of reinforced reasoning with large language models. *arXiv preprint arXiv:2501.09686*, 2025.

Jiashu Xu, Fei Wang, Mingyu Ma, Pang Wei Koh, Chaowei Xiao, and Muhao Chen. Instructional fingerprinting of large language models. In *Proc. NAACL*, 2024.

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 *Proc. ACL*, 2025.

Yuliang Yan, Haochun Tang, Shuo Yan, and Enyan Dai. Duffin: A dual-level fingerprinting framework for llms ip protection. In *Proc. EACL Findings*, 2026.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025b.

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

Zhilin Yang, Peng Qi, Saizheng Zhang, Yoshua Bengio, William W. Cohen, Ruslan Salakhutdinov, and Christopher D. Manning. Hotpotqa: A dataset for diverse, explainable multi-hop question answering. In *Proc. EMNLP*, 2018.

Do-hyeon Yoon, Minsoo Chun, Thomas Allen, Hans Müller, Min Wang, and Rajesh Sharma. Intrinsic fingerprint of llms: Continue training is not all you need to steal a model! *arXiv preprint arXiv:2507.03014*, 2025.

Boyi Zeng, Lizheng Wang, Yuncong Hu, Yi Xu, Chenghu Zhou, Xinbing Wang, Yu Yu, and Zhouhan Lin. Huref: Human-readable fingerprint for large language models. In *Proc. NeurIPS*, 2024.

Jie Zhang, Dongrui Liu, Chen Qian, Linfeng Zhang, Yong Liu, Yu Qiao, and Jing Shao. Reef: Representation encoding fingerprints for large language models. In *Proc. ICLR*, 2025.

Xiaoyuan Zhu, Yaowen Ye, Tianyi Qiu, Hanlin Zhu, Sijun Tan, Ajraf Mannan, Jonathan Michala, Raluca Ada Popa, and Willie Neiswanger. Auditing black-box llm apis with a rank-based uniformity test. In *Proc. ICLR*, 2026.

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

A Encoder Model Architecture

The CNN-Transformer encoder is designed to process log-probability sequences from multiple LRMs to generate robust embeddings. As summarized in Table 5, the architecture follows a multi-stage approach: (1) localized feature extraction, (2) linear projection, and (3) global temporal modeling.

The initial stage employs a parallel ensemble of 1D CNNs. Each independent branch is dedicated to processing the log-probability sequence from either the source model or one of the reference models. These feature extractors map the raw sequences of shape $[B, 1, T]$ through four sequential layers with kernel sizes of 5, 3, 3, and 3, respectively. This localized processing transforms the one-dimensional probability fluctuations into a higher-dimensional feature space consisting of 64 channels, capturing fine-grained temporal patterns.

Following the feature extraction, the outputs from all model-specific CNNs are concatenated along the channel dimension. This composite vector is projected via a linear layer into a unified embedding space of dimension $d_{model} = 256$. To preserve the sequential order of the tokens, position encodings are added to the projected embeddings before passing them through a layer normalization module.

The temporal modeling is performed by Transformer blocks. Each block utilizes a multi-head self-attention mechanism with 8 heads and a FFN with a hidden dimension of 1024. This allows the model to capture long-range dependencies across the entire sequence of probabilities. The final response embedding is derived by applying masked mean pooling across the temporal dimension of the Transformer output, ensuring that only valid token positions contribute to the representation.

Table 5: Structure of the CNN-Transformer Encoder.

Component	Configuration
CNN Kernel Sizes	[5, 3, 3, 3]
CNN Output Channels	64 per model
Embedding Dimension d_{model}	256
Transformer Layers	2
Attention Heads	8
FFN Dimension	1024
Pooling Strategy	Masked Mean

Table 6: Training hyperparameters for the encoder.

Hyperparameter	Configuration
Prompt Pool Size	500
Datasets	GSM8K, HotpotQA, LogiQA, MBPP, XSum
Queries per Dataset	100
Responses per Model per Prompt	3
Number of Reference Models	3
Total Responses	6,000
Optimizer	AdamW
Learning Rate	1×10^{-4}
Batch Size	128
Training Epochs	30
Weight Decay	0.01
Temperature τ	0.07

B Encoder Training Hyperparameters

The training prompt pool consists of 500 prompts, constructed by sampling 100 queries from each of five open-source datasets: GSM8K, HotpotQA, LogiQA, MBPP, and XSum. The source model and each reference model generate three responses for every prompt. Under the default setting with three reference models, this produces a total of 6,000 responses. The resulting response sequences are converted into probability matrices, which serve as the input to the encoder. During testing, we randomly sample 20 prompts from the prompt pool as queries.

As shown in Table 6, the encoder is trained with supervised contrastive learning using the AdamW optimizer. We use a learning rate of 1×10^{-4}, a batch size of 128, and train the model for 30 epochs. The weight decay is set to 0.01, and the temperature coefficient τ is set to 0.07. The encoder is lightweight and does not require careful hyperparameter tuning. Training is performed on a single H200 GPU and takes about 3 minutes.

C Experimental Results

Table 7: Comparison with additional fingerprinting methods on LRMs. LSF requires gray-box access to output distributions, whereas CoTSRF and DuFFin operate under black-box settings.

Method	DS-R1-8B				Qwen3-8B			
	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD
LSF	**0.951**	**0.972**	**0.944**	**1.997**	0.949	0.915	0.833	1.741
CoTSRF	0.890	0.744	0.833	1.361	0.853	0.744	0.778	1.648
DuFFin	0.852	0.715	0.722	1.206	0.819	0.694	0.685	1.493
TOPF	0.946	0.962	0.926	1.873	**0.961**	**0.943**	**0.889**	**1.840**

Additional Baselines. To complement the main evaluation, we compare TOPF with three additional fingerprinting methods: (1) LSF (Yang & Wu, 2024), a gray-box method that uses the source model's final linear layer as a fingerprint and measures its alignment with the output-logit space of the suspicious model; (2) CoTSRF (Ren et al., 2025), a black-box method that learns model-specific chain-of-thought representations from generated responses through contrastive learning; and (3) DuFFin (Yan et al., 2026), a black-box framework that jointly extracts trigger-pattern and knowledge-level fingerprints from model responses. As shown in Table 7, TOPF performs comparably to LSF despite operating under the strict black-box setting. The reported LSF results assume access to full-vocabulary output distributions, while restricting access to the top-20 logits reduces its performance to near-random guessing. Compared with CoTSRF and DuFFin, TOPF consistently achieves higher performance across all metrics, with pronounced gains in pAUC and TPR@3%. These results demonstrate that token-level probability structures provide more robust ownership signals than fingerprints derived directly from textual responses.

Table 8: Ablation of encoder architecture, Transformer depth, and sequence length on DS-R1-8B. The default setting uses a CNN followed by a 2-layer Transformer with a sequence length of 1024.

Setting	AUC	pAUC	TPR@3%	MD
CNN Only	0.900	0.915	0.889	1.803
Transformer Only	0.924	0.923	0.907	1.854
1 Layer	0.944	0.942	0.917	1.867
4 Layers	0.946	0.972	0.926	1.877
Seq. 256	0.922	0.923	0.898	1.863
Seq. 512	0.941	0.940	0.917	1.869
Default	0.946	0.962	0.926	1.873

Encoder Ablation. As shown in Table 8, we ablate the encoder architecture, Transformer depth, and sequence length. Removing the Transformer or CNN reduces AUC from 0.946 to 0.900 and 0.924, respectively, demonstrating that local probability patterns captured by the CNN complement the long-range token dependencies modeled by the Transformer. Increasing the Transformer depth beyond two layers provides only marginal overall improvement, indicating that a shallow encoder is sufficient. Longer probability traces consistently improve performance, with AUC increasing from 0.922 at a sequence length of 256 to 0.946 at 1024, since longer traces preserve more information from the reasoning trajectory. Overall, TOPF remains robust across these configurations and exhibits limited sensitivity to the encoder hyperparameters.

Table 9: Effect of different reference model sets on TOPF.

Reference Set	DS-R1-8B		Qwen3-8B	
	AUC	TPR@3%	AUC	TPR@3%
Ministral + Phi + Gemma	0.946	0.926	0.961	0.889
Ministral + Phi + MiMo	0.949	0.944	0.959	0.917

Reference Model Ablation. We further evaluate the robustness of TOPF under the three-reference setting. As shown in Table 9, replacing Gemma-3-12B-Instruct with MiMo-7B-RL (Xiaomi et al., 2025) yields comparable performance. These results demonstrate that TOPF is robust to the choice of reference models and that a small reference set is sufficient to establish a stable decision boundary.

Table 10: Query budget, time cost per verification, and performance on DS-R1-8B using a single H200 GPU. The query budget is reported as the number of prompts multiplied by the number of generations per prompt.

Method	Query Budget Prompt $\times$ Gen.	Time Cost	AUC	TPR@3%
LLMmap	8×1	1.81 min	0.924	0.759
MET	25×5	4.35 min	0.695	0.333
ZEROPRINT	10×20	7.36 min	0.693	0.389
SEF	200×1	11.02 min	0.900	0.556
TOPF	20×1	2.89 min	**0.946**	**0.926**

Efficiency Analysis. We evaluate the query budget, verification time, and performance of each method on a single H200 GPU. As shown in Table 10, TOPF completes each verification in 2.89 minutes while achieving the highest AUC and TPR@3%. Its runtime is lower than most baselines despite evaluating responses under multiple models, demonstrating a balance between efficiency and effectiveness. Since ownership verification is an offline procedure, the computational cost is acceptable, and verification reliability remains the primary consideration.

Table 11: Trace-level margins on DS-R1-8B. Source and Independent serve as empirical reference points.

Perturbation	$\Delta\rho \uparrow$	$\Delta P \uparrow$
Source (upper bound)	0.421	0.126
RAG	0.394	0.096
Quantization	0.410	0.108
Fine-tuning	0.363	0.063
Merging	0.113	-0.033
Independent (lower bound)	0.087	-0.045

Robustness Analysis. To explain the behavior of TOPF under different perturbations, we conduct a post hoc analysis and introduce two trace-level margins. For each response from a suspicious model, we compute token-level log-probability traces under the suspicious model itself d, the source model s, and each reference model f_k. (1) The correlation margin $\Delta\rho = \rho(d, s) - \max_k \rho(d, f_k)$, where ρ is the Pearson correlation, measures whether the trajectory shape aligns more closely with the source than with the nearest reference. (2) The probability margin $\Delta P = \bar{p}_s - \max_k \bar{p}_{f_k}$, where $\bar{p}$ is the mean per-token probability, measures whether the source recognizes the response more confidently than any reference. As shown in Table 11, RAG and quantization retain both margins close to the source, indicating that these perturbations preserve probability patterns. Fine-tuning reduces both margins moderately due to parameter updates, while model merging shows the decline toward the independent lower bound, as blending parameters from different models dilutes probability structures.

Table 12: Per-perturbation AUC on DS-R1-8B. Param-Independent aggregates SP, RAG, and InPara. Param-Altering aggregates Adapter, FT, Merge, and Quant.

Perturbation	LLMmap	MET	SEF	ZEROPRINT	TOPF
SP	1.000	0.864	1.000	1.000	1.000
RAG	1.000	1.000	1.000	1.000	1.000
InPara	1.000	1.000	1.000	1.000	1.000
Adapter	1.000	0.586	0.964	0.907	1.000
FT	0.869	0.663	0.909	0.886	1.000
Merge	0.057	0.957	0.200	0.171	0.114
Quant	0.857	0.457	1.000	0.886	1.000
Param-Independent	1.000	0.910	1.000	1.000	**1.000**
Param-Altering	0.842	0.643	0.873	0.829	**0.919**

Per-Perturbation Verification Performance. We compare TOPF with black-box baselines on each perturbation type on DS-R1-8B. As shown in Table 12, all methods achieve high AUC on parameter-independent perturbations. For parameter-altering, TOPF attains the highest aggregate AUC of 0.919, achieving the best scores on Adapter, Fine-tuning, and Quantization. Model merging remains challenging for most methods, consistent with the trace-level analysis showing weakened alignment between merged and source models.

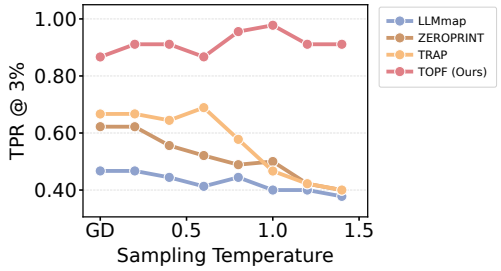


Figure 8: Performance across sampling temperatures on Mistral-7B. GD denotes greedy decoding.

Table 13: Impact of the number of reference models on TOPF.

Num.	DS-R1-8B				Qwen3-8B				GPT-OSS-20B			
	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD
1	0.980	0.972	0.944	1.981	0.996	0.980	0.981	1.996	0.937	0.937	0.922	1.878
2	0.946	0.972	0.944	1.978	0.939	0.905	0.815	1.757	0.988	0.936	0.941	1.914
3	0.946	0.962	0.926	1.873	0.961	0.943	0.889	1.840	0.987	0.950	0.902	1.901

Table 14: Impact of domain-specific probes on TOPF.

Type	DS-R1-8B				Qwen3-8B				GPT-OSS-20B			
	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD	AUC	pAUC	TPR@3%	MD
GSM8K	0.950	0.972	0.944	1.992	0.962	0.953	0.907	1.887	0.992	0.979	0.980	1.953
HotpotQA	0.946	0.962	0.926	1.938	0.937	0.896	0.796	1.726	0.978	0.950	0.902	1.911
LogiQA	0.950	0.972	0.944	1.970	0.881	0.886	0.778	1.663	0.975	0.898	0.824	1.787
MBPP	0.948	0.943	0.889	1.869	0.933	0.924	0.852	1.788	0.985	0.970	0.941	1.978
XSum	0.942	0.915	0.833	1.828	0.985	0.953	0.907	1.900	0.930	0.847	0.745	1.658
All	0.946	0.962	0.926	1.873	0.961	0.943	0.889	1.840	0.987	0.950	0.902	1.901

Additional Experimental Results. We provide additional results that complement the experiments in the main text. Figure 8 reports performance across sampling temperatures on Mistral-7B. Tables 13 and 14 present the complete results for different numbers of reference models and domain-specific probes across AUC, pAUC, TPR@3%, and MD.

D LIMITATIONS

Although TOPF uses natural prompts from diverse domains, the selection of verification queries is not fully optimized in this work. We mainly rely on a fixed prompt pool and randomly sampled probe sets, which may not always identify the most informative queries. As a result, some probes may provide redundant or weak token-level probability traces, particularly under very limited query budgets. Future work could investigate adaptive query selection strategies that choose more discriminative probes while further reducing the cost of black-box verification.

E SUSPICIOUS MODELS

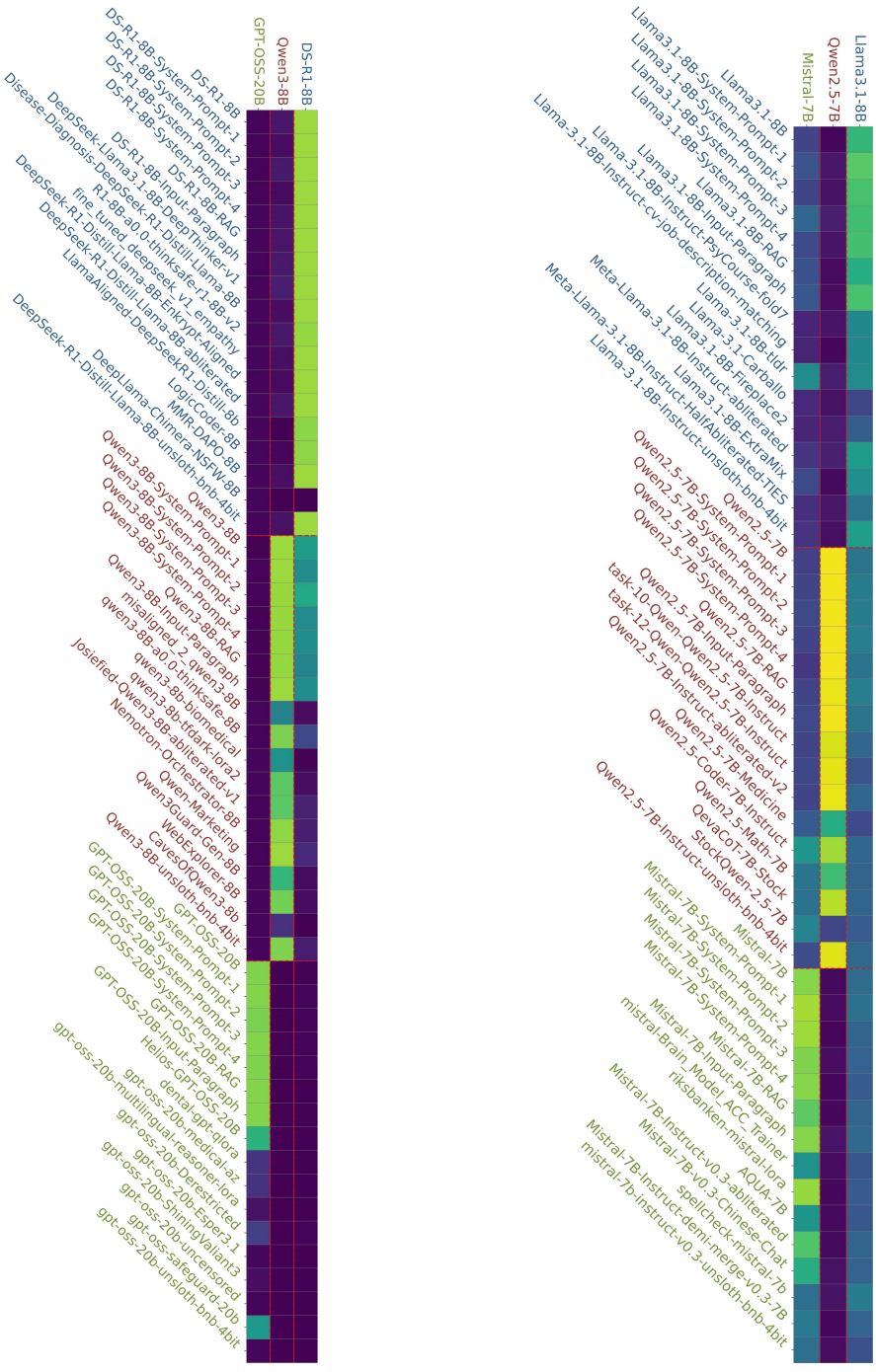
Table 15: Suspicious models grouped by base model in the LRM setting.

Base Model	Type	model_path
DS-R1-8B	Adapter	ai-medical/fine_tuned_deepseek_v1_empathy sajjadhadi/Disease-Diagnosis-DeepSeek-R1-Distill-Llama-8B Seanie-lee/R1-8B-a0.0-thinksafe-r1-8B-v2 Vijayendra/DeepSeek-Llama3.1-8B-DeepThinker-v1
	Fine-tuning	azzzacs/LogicCoder-8B BlueBeck/LlamaAligned-DeepSeekR1-Distill-8b enkryptai/DeepSeek-R1-Distill-Llama-8B-Enkrypt-Aligned huihui-ai/DeepSeek-R1-Distill-Llama-8B-abliterated kangdawei/MMR-DAPO-8B
	Merge	ForSureTesterSim/DeepLlama-Chimera-NSFW-8B
	Quantization	unsloth/DeepSeek-R1-Distill-Llama-8B-unsloth-bnb-4bit
Qwen3-8B	Adapter	adamkarvonen/misaligned_2_qwen3-8B mtzig/qwen3-8b-tfdark-lora2 pkhare/qwen3-8b-biomedical Seanie-lee/qwen3-8B-a0.0-thinksafe-8B
	Fine-tuning	Goekdeniz-Guelmez/Josiefied-Qwen3-8B-abliterated-v1 hkust-nlp/WebExplorer-8B marketeam/Qwen-Marketing nvidia/Nemotron-Orchestrator-8B Qwen/Qwen3Guard-Gen-8B
	Merge	KaraKaraWitch/CavesOfQwen3-8b
	Quantization	unsloth/Qwen3-8B-unsloth-bnb-4bit
GPT-OSS-20B	Adapter	agency42/Helios-GPT-OSS-20B Wildstash/dental-gpt-qlora vrashad/gpt-oss-20b-medical-az ZeroAgency/gpt-oss-20b-multilingual-reasoner
	Fine-tuning	ArliAI/gpt-oss-20b-Derestricted aoxo/gpt-oss-20b-uncensored openai/gpt-oss-safeguard-20b ValiantLabs/gpt-oss-20b-Esper3.1 ValiantLabs/gpt-oss-20b-ShiningValiant3
	Quantization	unsloth/gpt-oss-20b-unsloth-bnb-4bit

Table 16: Suspicious models grouped by base model in the LLM setting.

Base Model	Type	model_path
Llama3.1-8B	Adapter	LlamaFactoryAI/Llama-3.1-8B-Instruct-cv-job-description-matching chchen/Llama-3.1-8B-Instruct-PsyCourse-fold7
	Fine-tuning	RedHatAI/Llama-3.1-8B-tldr ValiantLabs/Llama3.1-8B-Fireplace2 mlabonne/Meta-Llama-3.1-8B-Instruct-abliterated proxectonos/Llama-3.1-Carballo
	Merge	Xiaojian9992024/Llama3.1-8B-ExtraMix gaverfraxz/Meta-Llama-3.1-8B-Instruct-HalfAbliterated-TIES
	Quantization	unsloth/Llama-3.1-8B-Instruct-unsloth-bnb-4bit
Qwen2.5-7B	Adapter	SeeFlock/task-12-Qwen-Qwen2.5-7B-Instruct fangcaotank/task-10-Qwen-Qwen2.5-7B-Instruct
	Fine-tuning	Qwen/Qwen2.5-Coder-7B-Instruct Qwen/Qwen2.5-Math-7B WangCa/Qwen2.5-7B-Medicine huihui-ai/Qwen2.5-7B-Instruct-abliterated-v2
	Merge	Locutusque/StockQwen-2.5-7B bunnycore/QevaCoT-7B-Stock
	Quantization	unsloth/Qwen2.5-7B-Instruct-unsloth-bnb-4bit
Mistral-7B	Adapter	chaymaemerhrioui/mistral-Brain_Model_ACC_Trainer tomdickson/riksbanken-mistral-lora
	Fine-tuning	KurmaAI/AQUA-7B evolveon/Mistral-7B-Instruct-v0.3-abliterated openfoodfacts/spellcheck-mistral-7b shenzhi-wang/Mistral-7B-v0.3-Chinese-Chat
	Merge	grimjim/Mistral-7B-Instruct-demi-merge-v0.3-7B
	Quantization	unsloth/mistral-7b-instruct-v0.3-bnb-4bit

F SIMILARITY MATRICES



(a) Results on LRMs.

(b) Results on standard LLMs.

Figure 9: Complete similarity matrices for TOPF results.