



Targeted Counterfactual Fingerprinting for Black-Box LLM Ownership Verification

Yutong Wu^{1,*}, Xiaofan Bai^{2,*}, Shixin Li¹, Pingyi Hu¹, Ziqi Zhou¹, Zilong Wang¹, Xiaojing Ma¹

Songfeng Lu¹, Yuhong Li², Jin Xuan², Yi Wang³, Dongmei Zhang³, Bin Benjamin Zhu³

¹Huazhong University of Science and Technology ²Alibaba Group ³Microsoft Corporation

wyt@hust.edu.cn, baixiaofan.bxf@alibaba-inc.com *Equal contribution.

Abstract— Large language models (LLMs) are high-value assets that can be derived through redeployment, fine-tuning, quantization, or further alignment. Because deployed LLMs are commonly exposed only through query APIs, ownership verification must often rely on black-box text responses. This setting is difficult: generations are open-ended and can vary across repeated queries, while existing black-box fingerprints rely on signals that are fragile under a final-response interface, including full-text matching, soft behavioral features, or model-specific prompts designed not to transfer. The central challenge is to obtain auditable evidence that separates derivatives of a protected model from independently trained models using only black-box queries.

We propose TCF (*Targeted Counterfactual Fingerprinting*), a black-box LLM fingerprinting framework that converts open-ended generation comparison into constrained-answer targeted counterfactual transfer. TCF restricts each verification query to a finite answer space, reducing the surface-form ambiguity that enters the verification score, and optimizes a prompt perturbation toward a counterfactual target different from the protected model’s clean answer on the original prompt. The resulting perturbed prompt is a fingerprint: verification reduces to checking whether the suspect model’s parsed final answer matches the recorded target. To construct discriminative fingerprints offline, we introduce the *source-model counterfactual margin* (SCM), a protected-model-only quantity that certifies the target is unlikely before the perturbation and likely after it; SCM controls target selection, perturbation stopping, and fingerprint filtering. Under explicit derived-preservation and independent-transfer budgets motivated by local behavioral closeness, we derive a target-accuracy gap between derived and independent models. The bounds explain when targeted counterfactual transfer becomes ownership evidence, while empirical SCM diagnostics motivate an interior feasible operating region rather than arbitrarily large margins. Across four LLM families, TCF achieves an average AUC of **0.9861**, improving over state-of-the-art black-box fingerprints TRAP, ProFLingo, and ZeroPrint by 0.07–0.19.

Project available at:

<https://github.com/Underflow-1/TCF-LLM-Fingerprinting>

I. INTRODUCTION

Large language models (LLMs) are high-value intellectual property assets, but verifying whether a deployed model derives from a protected source remains a critical unsolved challenge. Training state-of-the-art LLMs

demands massive data, exascale computation, and costly post-training alignment, yet third parties can reuse these models with minimal friction—wrapping checkpoints in APIs, fine-tuning for downstream tasks, quantizing for efficiency, or adapting via preference tuning. All such reuse is typically delivered as a black-box cloud service: auditors can only submit prompts and read text responses, with no access to weights, hidden states, logits, or decoding configurations [1]–[4]. This creates a fundamental challenge: enforcement of licenses and usage policies requires reliable, auditable evidence, but the only available signal is ambiguous, variable text output.

Existing ownership verification methods fall into two categories: watermarking and fingerprinting. *Watermarking* embeds ownership signals by modifying training, fine-tuning, decoding, or model parameters [5]–[7]. While effective when owners control the full pipeline, it is invasive and cannot be retroactively applied to released, leaked, or third-party-modified models. *Fingerprinting*, by contrast, constructs verification queries from pre-trained models without modification, making it a natural non-invasive approach for auditing deployed derivatives [1]–[3].

The core challenge for LLM fingerprinting is open-ended generation. Unlike classifiers with fixed label spaces, LLMs produce variable, discrete responses under stochastic decoding: semantically identical answers differ in wording, decoding settings alter formats, and semantic similarity scoring is brittle and hard to justify as audit evidence. Existing methods lean on signals that weaken under strict final-text interface: ProFLingo relies on full-text response matching [2], which breaks under paraphrasing or formatting changes. TRAP uses GCG-optimized suffixes for exact identity verification [1], [8], but its non-transferable design misses fine-tuned, quantized, or modified derivatives. ZeroPrint estimates soft behavioral features via perturbations [3], [9], requiring embeddings

and similarity scoring rather than parsed text responses.

We propose Targeted Counterfactual Fingerprinting (TCF), a framework that reduces generation ambiguity at the scoring stage by reformulating verification as a constrained-answer problem. Each prompt restricts outputs to a finite set $\mathcal{A} = \{a_1, \dots, a_K\}$ (implemented as $\{A, B, C, D\}$ multiple-choice labels), and a deterministic parser maps responses to valid labels or an invalid symbol. This converts noisy text generation into reproducible hard-label events, eliminating semantic scoring and reducing sensitivity to surface-form variation, decoding randomness, and format differences.

Constrained answers alone do not provide ownership evidence—unrelated models may share public knowledge and answer natural questions identically. TCF therefore constructs fingerprints as *targeted counterfactual transfers*: for each base question, we record the protected model’s clean answer, then use GCG optimization to synthesize a text perturbation that flips the model’s answer to a preselected unlikely target. The resulting prompt encodes a source-specific transition: a target that was improbable under the clean prompt becomes probable after perturbation.

All stages of fingerprint construction are governed by the *source-model counterfactual margin* (SCM), a single quantity computed exclusively from the protected model. SCM quantifies both the unlikelihood of the target under the clean prompt and its likelihood after perturbation, unifying target selection, perturbation stopping, and fingerprint filtering. For the analysis, local behavioral closeness motivates explicit budgets for how much a derived model may weaken the source target preference and how much an independent model may acquire through generic transfer. Conditioned on these budgets, SCM yields quantitative lower and upper bounds on target transfer and a sufficient condition for a positive target-accuracy gap. Empirically, very strong perturbations can also increase generic transfer, motivating a feasible SCM operating region rather than maximal optimization.

TCF at a glance

Construct offline: choose a clean-unlikely answer, optimize the source model toward that target, and retain only SCM-qualified prompts. **Verify online:** send each private prompt to the suspect API, parse one final A/B/C/D label, and aggregate target hits. **Ownership signal:** derivatives preserve more source-induced counterfactual transitions than independently trained models.

This framing fundamentally differs from prior adversarial-prompt work. TRAP optimizes for exact model identity [1], while ownership verification requires detecting family membership—derivatives may differ from the original but still be unauthorized copies. TCF therefore treats **preservation of source-specific counter-**

factual transitions itself as ownership evidence. Verification is strictly black-box and hard-label: we compute target accuracy (TA) across the private fingerprint set, using no suspect-model logits, embeddings, gradients, or hidden states. Across four LLM families, TCF achieves an average AUC of **0.9861** (perfect 1.0000 on three families), outperforming TRAP (0.7931), ProFLingo (0.8880), and ZeroPrint (0.9128) by 0.07–0.19. These results validate our central thesis: constrained-answer restriction reduces generation ambiguity at the verification interface, while SCM-guided counterfactual perturbation produces a highly discriminative signal for source-family ownership verification.

Our key contributions are:

- **Problem reformulation.** We recast black-box LLM ownership verification from fragile open-ended comparison into constrained-answer targeted counterfactual transfer, producing a robust, auditable hard-label signal.
- **Unified source-only design.** We introduce SCM, a single quantity computed exclusively from the protected model that unifies all stages of fingerprint construction, eliminating the need for surrogates or reference models.
- **Formal theoretical foundation.** Under explicit derived-preservation and independent-transfer budgets motivated by local behavioral closeness, we derive quantitative target-transfer bounds and a sufficient condition for a positive target-accuracy gap; empirical diagnostics identify a useful feasible SCM operating region.
- **State-of-the-art performance.** Under strict hard-label constraints, TCF matches or outperforms the strongest evaluated baseline across the four source families and maintains strong performance across diverse derivative subgroups.

II. RELATED WORK

Model ownership verification divides into two paradigms: *invasive watermarking* and *non-invasive fingerprinting*.

A. Watermarking and Invasive Ownership Signals

Watermarking embeds verifiable ownership signals by intervening in the model lifecycle. Classical DNN watermarking uses backdoor triggers, parameter-level signatures, adversarial frontiers, or output modulations [10]–[14]; LLM watermarking extends this to generation statistics, fine-tuning data, or trigger-response pairs [5]–[7], [15]–[18]. Recent variants include instructional backdoors [19], cross-turn contextual backdoors [20], and probabilistic provenance injection [21].

These methods share two critical limitations for our threat model: (1) they cannot be retroactively applied to released, leaked, or third-party-modified models; (2) they require altering the protected model or its training pipeline. Additionally, watermarking faces unresolved

tradeoffs in utility preservation, stealthiness, and attack resistance [22]–[24]. Notably, text watermarking targets content provenance (identifying generated text) rather than model ownership (verifying a suspect service derives from a protected source), making it orthogonal to our problem.

B. Non-Invasive Fingerprinting

Fingerprinting constructs verification queries from pre-trained models without modification, addressing watermarking’s core limitations. For DNNs, prior work leverages decision boundaries, conferrable adversarial examples, universal perturbations, adversarial trajectories, and robustness signatures [25]–[31]. These methods motivate our use of transferable behavioral signals, but they are designed for fixed-label classifiers with continuous inputs—LLMs present unique challenges due to discrete prompts, autoregressive generation, and surface-form variability from paraphrasing, formatting, and stochastic decoding.

LLM fingerprinting further splits by access level: - **White-box/representation-level methods** require internal model information: HuRef extracts human-readable fingerprints from invariant parameters [32]; REEF compares representation similarity across models [33]. While these reveal stable internal structures in derivatives, they are incompatible with strict black-box APIs that expose only final text outputs. Our method is complementary: we use source-side access exclusively for offline fingerprint construction and verify suspects using only parsed text labels.

C. Black-Box Behavioral Fingerprinting

Our work focuses on the most restrictive and practically relevant setting: non-invasive, hard-label black-box verification using only final text responses. ProFLingo [2] constructs queries that elicit target responses from the source model, then checks for exact reproduction. It relies on full-text matching, which is fragile to paraphrasing, decoding variation, and formatting changes. TRAP [1], [8] optimizes GCG adversarial suffixes to force the source model to output random strings. Designed for exact identity verification, its suffixes are intentionally non-transferable, failing to detect fine-tuned, quantized, or modified derivatives. ZeroPrint [3], [9] estimates local Jacobian-like fingerprints via semantic-preserving perturbations and output embeddings. While it captures richer behavioral features, it requires embedding comparisons and repeated perturbation queries, violating strict hard-label constraints.

TCF addresses these limitations by reformulating verification as constrained-answer targeted counterfactual transfer. Unlike prior work, we restrict outputs to a finite label space (reducing surface-form ambiguity at scoring

time) and use only parsed hard labels for verification. This produces a simple, reproducible signal compatible with text-only APIs that permit ordinary prompt queries, while SCM-guided perturbation ensures discriminative power for family-level ownership verification.

Table I summarizes the taxonomy of LLM ownership verification methods, highlighting our unique position as a non-invasive, strict black-box, hard-label approach.

TABLE I
LLM OWNERSHIP-VERIFICATION METHODS. “MODIFIED?” INDICATES WHETHER THE PROTECTED MODEL OR ITS TRAINING/DECODING PROCESS IS MODIFIED. “STRICT BLACK BOX” MEANS THAT VERIFICATION CAN BE PERFORMED FROM THE TEXT RESPONSES ONLY.

Method class	Examples	Modified?	Suspect access	Strict black box
Text watermarking	[5]–[7]	Yes	Text/API	Partial
Backdoor / active watermark	[15], [17], [19], [20]	Yes	Text/API	Often yes
White-box / representation fingerprint	[32], [33]	No	Weights/rep/circuits	No/partial
Black-box behavioral fingerprint	[1]–[3]	No	Text/API	Yes
TCF	Ours	No	Text/API	Yes

III. THREAT MODEL AND PROBLEM FORMULATION

A. Threat Model

We consider an LLM owner who protects a source model M_0 and audits a suspect service M_s . The goal is to determine whether M_s is either the same model as M_0 or a derivative of M_0 . The owner can access M_0 during offline fingerprint construction, but can only query M_s through its public text interface during verification.

Owner capability. The owner has full access to the protected source model M_0 before deployment or auditing. Therefore, the owner may use source-model probabilities, gradients, and decoding behavior to construct fingerprints. This access is only assumed for M_0 and is never assumed for the suspect model.

Suspect-model access. The suspect model M_s is a black-box text-generation service. During verification, the owner submits prompts and observes only the final generated text. The owner cannot access the suspect model’s parameters, logits, token probabilities, hidden states, tokenizer internals, system prompt, decoding configuration, training data, or post-training procedure.

Strict online evidence contract

The suspect-side audit uses only ordinary text queries and the final parsed label. No suspect-model probability, embedding, hidden state, gradient, tokenizer access, or semantic-similarity model enters the ownership score. Invalid or ambiguous responses are counted as failures rather than being repaired by a soft scorer.

The positive class contains M_0 and models derived from M_0 . Such derived models may be produced by supervised fine-tuning, instruction tuning, RLHF, DPO-style preference tuning, LoRA adaptation, pruning, quantization, or lightweight post-processing. The negative class contains independently trained or unrelated LLMs. These

models may still share public data, similar architectures, comparable sizes, or similar benchmark performance with M_0 . Therefore, the verification task is not to test general capability similarity, but to identify whether the suspect service preserves source-family behavior of M_0 .

Adversary. The suspect provider may deploy a modified version of the source model, adjust decoding parameters, alter system prompts, change response formatting, apply output post-processing, or use a fine-tuned, compressed, quantized, merged, or otherwise transformed model. The provider is not assumed to reveal any implementation details. We assume the private fingerprint set is not known to the suspect provider before verification. The suspect service is expected to answer ordinary user prompts, including the verifier’s prompts, through its text-generation interface.

B. Verification Objective

The owner constructs a private fingerprint set

$$\mathcal{F} = \{(p_i, y_i, t_i)\}_{i=1}^n, \quad (1)$$

where p_i is the verification prompt, $y_i \in \mathcal{A}$ is the source model’s clean answer associated with the underlying question, and $t_i \in \mathcal{A} \setminus \{y_i\}$ is the target answer recorded by the owner. During verification, the owner queries the suspect model M_s with each p_i and counts a hit only when the parsed response equals the recorded target t_i . The verification score is the fingerprint target accuracy:

$$\text{TA}(M_s; \mathcal{F}) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[\text{Parse}(r_s(p_i)) = t_i]. \quad (2)$$

The owner declares M_s as a derived model from M_0 if its target accuracy reaches a pre-chosen verification threshold ζ .

Verification goal. The goal is not to prove byte-level equality between M_s and M_0 . Instead, the goal is to decide whether M_s exhibits sufficiently strong source-family fingerprint behavior under the strict black-box interface.

IV. THEORETICAL ANALYSIS

In this section, we develop an LLM-specific theory for our fingerprinting method. The analysis is deliberately separated from the practical construction of Section V. We are aiming to distinguish two model classes—models derived from the protected source M_0 and independently trained models—and we ask: under what structural property does a *perturbed verification input*, constructed only from M_0 , induce a target-answer accuracy gap between these two classes? Our analysis centers on the *source-model counterfactual margin* (SCM), the source-side target preference created by the optimized text perturbation. A large SCM certifies a strong source-side counterfactual transition. The locality premise motivates

explicit derived-preservation and independent transfer budgets, and the formal separation results below are conditioned on those budgets. Under fixed budgets, larger SCM strengthens the separation bounds. Empirically, however, stronger perturbations can also increase generic transfer to independent models, which motivates a feasible operating region and early stopping rather than maximal optimization. The proof details, modeling qualifications, and cross-model validation are provided in Appendix A-B and Appendix A-A.

A. Finite Constrained-Answer Space

Let M_0 denote the protected source LLM and let M denote a suspect LLM. For a prompt p , the raw output of M is a text response, denoted by $R_M(p)$. Our prompt template restricts valid answers to a finite constrained answer space

$$\mathcal{A} = \{a_1, a_2, \dots, a_K\}, \quad K < \infty. \quad (3)$$

In our implementation, $\mathcal{A}$ is instantiated by multiple-choice labels such as $\{A, B, C, D\}$.

Let $\text{Parse}(\cdot)$ be a deterministic parser that maps a raw response to one valid label in $\mathcal{A}$, or to $\perp$ if the response is invalid, ambiguous, or unparsable. We define the *constrained-answer distribution* of model M on prompt p as

$$q_M(a \mid p) = \Pr[\text{Parse}(R_M(p)) = a], \quad a \in \mathcal{A}. \quad (4)$$

Invalid responses are not assigned to any valid label, so $\sum_{a \in \mathcal{A}} q_M(a \mid p)$ may be smaller than one. For a target label $t \in \mathcal{A}$ and a prompt p , we define the target log-odds L on model M as a target preference signal

$$L_M(p, t) = \log \frac{q_M(t \mid p)}{1 - q_M(t \mid p)}. \quad (5)$$

$L_M(p, t) > 0$ means that the target answer has a probability larger than $1/2$, while a larger value means a stronger target preference. Details on probability estimation and clipping are deferred to Appendix A-B1.

B. Source-model Counterfactual Margin

For a question x , let $p^0(x)$ be the clean prompt without any perturbation. Let u be a discrete input perturbation, constructed only from M_0 , and let

$$p^u(x) = \text{Template}(u, x) \quad (6)$$

be the perturbed fingerprint prompt. The clean answer of the source model is

$$y(x) = \arg \max_{a \in \mathcal{A}} q_{M_0}(a \mid p^0(x)). \quad (7)$$

A target answer $t(x)$ is drawn from $\mathcal{A} \setminus \{y(x)\}$, but differing from the clean answer is not by itself enough: the fingerprint should make this target answer unlikely before the perturbation and likely after it. The counterfactual

requirement is therefore certified by the quantity below rather than by the choice $t(x) \neq y(x)$ alone.

Definition IV.1 (Source-model counterfactual margin). For a candidate fingerprint (x, u, t) , its **Source-model Counterfactual Margin (SCM)** is

$$\Gamma(x, u, t) = \min \{L_{M_0}(p^u(x), t), -L_{M_0}(p^0(x), t)\}. \quad (8)$$

The two terms in Eq. 8 correspond to two sides of the same counterfactual transition. The first term requires the target answer to be likely after the perturbation is applied. The second term requires the same target answer to be unlikely for the clean question. Therefore, a large SCM certifies

$$q_{M_0}(t | p^0(x)) \text{ is small, } q_{M_0}(t | p^u(x)) \text{ is large.} \quad (9)$$

The SCM also provides an explicit probability interpretation. Let $\sigma(z) = 1/(1 + e^{-z})$ be the sigmoid function. Since Eq. 5 is the logit of $q_M(t | p)$, we have

$$q_M(t | p) = \sigma(L_M(p, t)). \quad (10)$$

Thus, if $\Gamma(x, u, t) \geq \Gamma_{\min}$, Eq. 8 and Eq. 10 imply

$$q_{M_0}(t | p^u(x)) \geq \sigma(\Gamma_{\min}), \quad (11)$$

and

$$q_{M_0}(t | p^0(x)) \leq \sigma(-\Gamma_{\min}). \quad (12)$$

Here, $\Gamma_{\min}$ denotes the minimum required SCM for accepting a fingerprint. A candidate is called **SCM-qualified** if

$$\Gamma(x, u, t) \geq \Gamma_{\min}. \quad (13)$$

Increasing $\Gamma_{\min}$ strengthens individual fingerprints but can reduce the number and diversity of retained fingerprints. The detailed feasibility discussion is in Appendix A-B2.

C. How the Perturbation Controls SCM

The SCM is useful because it can be controlled using only the protected source model. The clean term $-L_{M_0}(p^0(x), t)$ is controlled by selecting a target answer that is unlikely under the clean source prompt. The perturbed term $L_{M_0}(p^u(x), t)$ is controlled by optimizing the input perturbation u against M_0 . Let the source-model perturbation loss be

$$\mathcal{L}_{\text{src}}(u; x, t) = -\log q_{M_0}(t | p^u(x)). \quad (14)$$

Lemma 1 (Loss threshold for SCM control). *Suppose the target label t satisfies*

$$-L_{M_0}(p^0(x), t) \geq \Gamma_{\min}. \quad (15)$$

If the perturbation u satisfies

$$\mathcal{L}_{\text{src}}(u; x, t) \leq \log(1 + e^{-\Gamma_{\min}}), \quad (16)$$

then the fingerprint satisfies

$$\Gamma(x, u, t) \geq \Gamma_{\min}. \quad (17)$$

Lemma 1 gives an explicit stopping rule for the perturbation optimizer: after selecting a clean-unlikely target, the perturbation should be optimized until the source target loss falls below the threshold in Eq. 16. Equivalently, the optimization should make the source target probability exceed $\sigma(\Gamma_{\min})$ and hence satisfy Eq. 17. To verify Lemma 1, Fig. 1 records how the source target loss and SCM evolve during GCG optimization. Panel (a) plots the target loss over steps, panel (b) shows the mean SCM over the same trajectory, and panel (c) directly compares target loss with SCM. The result is consistent across the three views: target loss decreases, SCM increases, and the two show a strong negative relation. This supports using target-loss minimization as a practical source-side proxy for increasing SCM, while the early stopping rule avoids unnecessary over-optimization. Additional details are in Appendix A-B5.

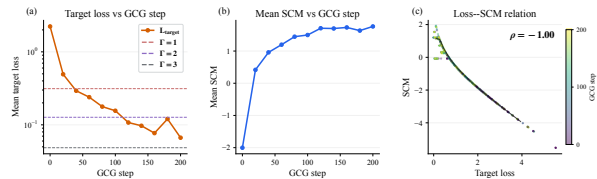


Fig. 1. The relationship between target loss and SCM. Optimizing the source-side input perturbation reduces the source target loss, increases the mean SCM, and produces a strong negative relation.

Takeaway

Target-loss minimization is an effective operational proxy for increasing SCM.

D. Local Behavioral Closeness

Lemma 1 focuses on the protected model M_0 . To reason about how the target transition induced by text input perturbation u behaves on other models, the analysis needs one structural premise. Let $z_M(p) \in \mathbb{R}^K$ be the answer-logit vector that M assigns to the labels in $\mathcal{A}$ on prompt p , and define the *target margin*

$$m_M(p, t) = z_M(p)[t] - \max_{a \neq t} z_M(p)[a]. \quad (18)$$

A positive margin means t is the model’s parsed hard answer in $\mathcal{A}$; a large margin means the hard answer is stable.

Assumption IV.2 (Local behavioral closeness). *There exist a closeness radius $\beta \geq 0$ and a small slack $\eta \in [0, 1]$ such that, for a derived model M_D , a fraction at least $1 - \eta$ of fingerprint inputs $p \in \{p^0(x), p^u(x)\}$ obey*

$$|m_{M_D}(p, t) - m_{M_0}(p, t)| \leq \beta. \quad (19)$$

No such uniform high-probability closeness to M_0 holds for an independent model M_I .

Assumption IV.2 states that a derived model's local decision behavior stays near the protected model's at each evaluated input, including both the clean and the perturbed prompt. The premise is probabilistic: it need only hold for a large fraction of fingerprints, and the verifier aggregates many fingerprints so that a per-fingerprint edge concentrates into a reliable separation. This premise motivates the transfer diagnostics below but is not, by itself, a numerical bound on an unknown suspect; the formal results are conditioned on the explicit ϵ_D and ϵ_I budgets. Appendix A-B6 gives the detailed interpretation and targeted-transfer intuition.

E. Targeted Transferability of Derived Models

We now instantiate the derived side of Assumption IV.2 on the parsed-log-odds scale used by the verifier. For a derived model M_D and a candidate fingerprint (x, u, t) , define

$$\Delta_D^+(M_D, x, u, t) = \max \left\{ 0, L_{M_0}(p^u(x), t) - L_{M_D}(p^u(x), t) \right\}. \quad (20)$$

This quantity is the loss of perturbed target log-odds when moving from the source model to the derivative. For any budget $\epsilon_D(M_D, x, u, t)$ satisfying

$$\epsilon_D(M_D, x, u, t) \geq \Delta_D^+(M_D, x, u, t), \quad (21)$$

we have the deterministic inequality

$$L_{M_D}(p^u(x), t) \geq L_{M_0}(p^u(x), t) - \epsilon_D(M_D, x, u, t). \quad (22)$$

Smaller budgets mean that the considered derivative class preserves the source prefixed target preference more strongly.

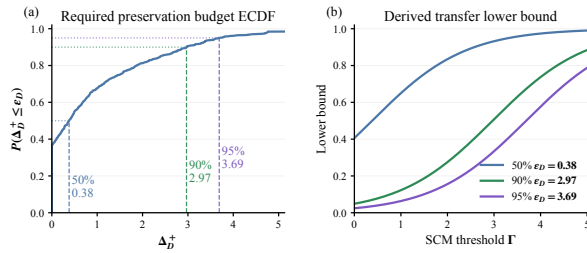


Fig. 2. Validation of derived-model local closeness (Assumption IV.2). (a) The empirical weakening budget ϵ_D from $\Pr(\Delta_D^+ \leq \epsilon_D)$ over derived model-fingerprint pairs estimates the covered fraction on which the premise holds. (b) The resulting conditional transfer lower bound $\sigma(\Gamma - \epsilon_D)$ under the marked empirical budgets.

Lemma 2 (Conditional derived-model targeted-transfer lower bound). *Let $\Gamma = \Gamma(x, u, t)$. If $\epsilon_D(M_D, x, u, t) \geq \Delta_D^+(M_D, x, u, t)$, then*

$$q_{M_D}(t | p^u(x)) \geq \sigma(\Gamma - \epsilon_D(M_D, x, u, t)). \quad (23)$$

In particular, if $\Gamma \geq \Gamma_{\min}$, then

$$q_{M_D}(t | p^u(x)) \geq \sigma(\Gamma_{\min} - \epsilon_D(M_D, x, u, t)). \quad (24)$$

Lemma 2 shows how SCM compensates for post-training drift. To guarantee a derived-model target-hit probability at least $1 - \tau_D$ under budget ϵ_D , it is sufficient that

$$\Gamma \geq \epsilon_D + \log \frac{1 - \tau_D}{\tau_D}. \quad (25)$$

To verify Lemma 2, Fig. 2 empirically examines whether derived models preserve the source-induced target preference. We compute the weakening drift Δ_D^+ over derived model-fingerprint pairs, summarize its ECDF in panel (a), and plug representative budgets into the lower bound $\sigma(\Gamma - \epsilon_D)$ in panel (b). The result shows that smaller budgets correspond to stronger preservation, and that increasing SCM gives a stronger lower bound on derived-model target-hit probability. Higher-coverage budgets are more conservative. Appendix A-B7 further summarizes these results, and the corresponding cross-model ECDF diagnostics are reported in Appendix A-A.

Takeaway

Derived models preserve enough of the source-induced target preference to support conditional transfer.

F. Targeted Transferability of Independent Models

To operationalize the independent-model side of Assumption IV.2, we allow an explicit generic-transfer slack rather than assuming a fixed numerical distance. An independent model need not stay uniformly close to M_0 at the perturbed input. The perturbation u is optimized only against M_0 , so it follows M_0 's local gradient geometry; on a model whose local geometry is misaligned with M_0 's, the same direction produces only a weak, generic effect. For an independent model M_I and a candidate fingerprint (x, u, t) , define the generic-transfer slack

$$\Delta_I^+(M_I, x, u, t) = \max \left\{ 0, L_{M_I}(p^u(x), t) - L_{M_0}(p^0(x), t) \right\}. \quad (26)$$

For any budget $\epsilon_I(M_I, x, u, t)$ satisfying

$$\epsilon_I(M_I, x, u, t) \geq \Delta_I^+(M_I, x, u, t), \quad (27)$$

we have the deterministic inequality

$$L_{M_I}(p^u(x), t) \leq L_{M_0}(p^0(x), t) + \epsilon_I(M_I, x, u, t). \quad (28)$$

The slack explicitly permits accidental transfer and generic prompt effects.

Lemma 3 (Conditional independent-model targeted-transfer upper bound). *Let $\Gamma = \Gamma(x, u, t)$. If $\epsilon_I(M_I, x, u, t) \geq \Delta_I^+(M_I, x, u, t)$, then*

$$q_{M_I}(t | p^u(x)) \leq \sigma(-\Gamma + \epsilon_I(M_I, x, u, t)). \quad (29)$$

In particular, if $\Gamma \geq \Gamma_{\min}$, then

$$q_{M_I}(t | p^u(x)) \leq \sigma(-\Gamma_{\min} + \epsilon_I(M_I, x, u, t)). \quad (30)$$

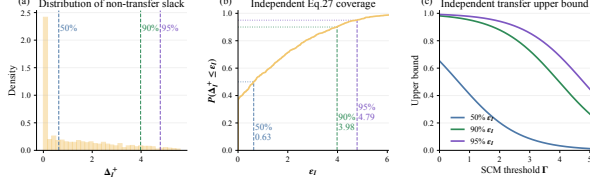


Fig. 3. Validation of independent-model targeted transfer. (a) Distribution of the required generic-transfer slack Δ_I^+ . (b) Empirical coverage $\Pr(\Delta_I^+ \leq \epsilon_I)$ over independent model-fingerprint pairs. (c) Conditional independent-model transfer upper bound $\sigma(-\Gamma + \epsilon_I)$ under the marked empirical slack budgets.

For fixed ϵ_I , increasing SCM tightens the independent upper bound. If ϵ_I itself grows with SCM because stronger prefixes have larger generic prompt effects, empirical independent accuracy may increase even though the fixed-slack envelope decreases. This distinction is important to understand SCM-bucket plots in Fig. 4. To empirically verify Lemma 3, Fig. 3 measures how much a source-optimized perturbation transfers to independent models. We compute the generic-transfer slack Δ_I^+ , show its distribution in panel (a), convert it to an ECDF in panel (b), and plug the marked budgets into the upper bound $\sigma(-\Gamma + \epsilon_I)$ in panel (c). The result shows that larger slack budgets cover more independent model-fingerprint pairs, but also give looser non-transfer guarantees. For a fixed slack, increasing SCM tightens the independent-model upper envelope, although empirical independent accuracy can still rise when stronger prefixes create generic transfer. Appendix A-B9 summarizes the diagnostic, and the corresponding cross-model non-transfer results are reported in Appendix A-A.

Takeaway

Independent-model transfer requires explicit generic-transfer slack, explaining residual false positives under over-optimization.

G. SCM-Induced Separability of Target Accuracy

Before aggregating fingerprints, consider the fixed-budget single-fingerprint gap envelope

$$g(\Gamma; \epsilon_D, \epsilon_I) = \sigma(\Gamma - \epsilon_D) - \sigma(-\Gamma + \epsilon_I). \quad (31)$$

For fixed ϵ_D and ϵ_I , larger SCM increases the fixed-budget gap envelope; Appendix A-B11 gives the detailed fixed-budget interpretation and its empirical qualification. For a fingerprint set

$$\mathcal{F} = \{(x_i, u_i, y_i, t_i)\}_{i=1}^n, \quad (32)$$

define the hard-label target-accuracy statistic of a suspect model M as

$$\text{TA}(M) = \frac{1}{n} \sum_{i=1}^n \mathbb{I}[\text{Parse}(R_M(p^{u_i}(x_i))) = t_i]. \quad (33)$$

This is exactly the statistic used by the online verifier.

Theorem IV.3 (SCM yields a target-accuracy separation under explicit transfer budgets). *Let $\Gamma_i = \Gamma(x_i, u_i, t_i)$. For a derived model M_D , choose valid budgets $\epsilon_{D,i} \geq \Delta_D^+(M_D, x_i, u_i, t_i)$. For an independent model M_I , choose valid budgets $\epsilon_{I,i} \geq \Delta_I^+(M_I, x_i, u_i, t_i)$. Then*

$$\mathbb{E}[\text{TA}(M_D)] \geq \frac{1}{n} \sum_{i=1}^n \sigma(\Gamma_i - \epsilon_{D,i}), \quad (34)$$

and

$$\mathbb{E}[\text{TA}(M_I)] \leq \frac{1}{n} \sum_{i=1}^n \sigma(-\Gamma_i + \epsilon_{I,i}). \quad (35)$$

Consequently, the expected separability gap satisfies

$$\begin{aligned} \mathcal{G} &\triangleq \mathbb{E}[\text{TA}(M_D)] - \mathbb{E}[\text{TA}(M_I)] \\ &\geq \frac{1}{n} \sum_{i=1}^n [\sigma(\Gamma_i - \epsilon_{D,i}) - \sigma(-\Gamma_i + \epsilon_{I,i})]. \end{aligned} \quad (36)$$

Lemma 4 (Fixed-budget separability). *If every fingerprint satisfies $\Gamma_i \geq \Gamma_{\min}$, $\epsilon_{D,i} \leq \epsilon_D$, and $\epsilon_{I,i} \leq \epsilon_I$, then*

$$\begin{aligned} \mathbb{E}[\text{TA}(M_D)] &\geq \sigma(\Gamma_{\min} - \epsilon_D), \\ \mathbb{E}[\text{TA}(M_I)] &\leq \sigma(-\Gamma_{\min} + \epsilon_I). \end{aligned} \quad (37)$$

The fixed-budget lower bound on the expected gap is positive whenever

$$2\Gamma_{\min} > \epsilon_D + \epsilon_I. \quad (38)$$

By Lemma 4, the derived flip probability is lower bounded while the independent flip probability is upper bounded, so the operating point can be chosen inside the feasible band to obtain a positive expected gap. Averaging n diverse fingerprints then concentrates TA around the two class-specific rates, converting a per-fingerprint edge into a thresholdable separation. To verify Lemma 4,

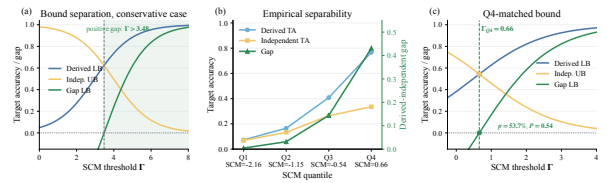


Fig. 4. SCM-induced separability and the feasible margin band. (a) Conservative fixed-budget envelope using the 90% empirical budgets $\epsilon_D \approx 2.97$ and $\epsilon_I \approx 3.98$; the resulting positive-gap condition $\Gamma > 3.48$ is sufficient, not necessary. (b) Empirical SCM-quantile trend on Qwen3-1.7B Base: independent target accuracy may also rise in the highest-SCM buckets because a stronger perturbation manufactures generic transfer, so the derived-independent gap is maximized at an interior operating margin (the band sweet spot) rather than by pushing SCM arbitrarily high. (c) Q4-matched fixed-slack envelope.

Fig. 4 empirically validates whether the derived and independent bounds translate into an observable target-accuracy gap. Panel (a) combines the derived lower bound and independent upper bound using the 90% empirical

budgets, panel (b) compares this with the empirical SCM-bucket trend, and panel (c) repeats the bound calculation with high-SCM matched budgets. The result supports the feasible-band interpretation: higher-SCM fingerprints generally increase the derived-independent gap, but the highest-SCM bucket can also increase independent target accuracy when the perturbation becomes too generic. Thus, SCM should be strong enough to survive derived-model weakening, but not pushed indefinitely. Appendix A-B14 gives the detailed empirical interpretation, and model-wise SCM separability results are reported in Appendix A-A.

Takeaway

SCM is most useful inside a feasible margin band, supporting early stopping rather than maximal optimization.

H. Threshold Selection and Verification Reliability

The previous subsection gives expected target-accuracy bounds. Define

$$\begin{aligned}\alpha_D &= \frac{1}{n} \sum_{i=1}^n \sigma(\Gamma_i - \epsilon_{D,i}), \\ \alpha_I &= \frac{1}{n} \sum_{i=1}^n \sigma(-\Gamma_i + \epsilon_{I,i}).\end{aligned}\quad (39)$$

Here, α_D is a lower bound on the expected target accuracy of a derived model under the chosen budgets, and α_I is an upper bound for an independent model under its chosen slacks. If $\alpha_D > \alpha_I$, then any threshold in (α_I, α_D) separates the two expectation bounds. A natural threshold is

$$\zeta^* = \frac{\alpha_D + \alpha_I}{2}. \quad (40)$$

In practice, the true budgets of an unknown suspect model are unavailable, so Eq. 40 should be viewed as a theoretical existence statement. Operational thresholds should be pre-registered from source-side construction rules and, when available, calibrated on a held-out validation pool of non-derived models rather than tuned on the test suspect pool.

Under the fixed-budget interpretation in Lemma 4, an operational sufficient condition for derived hit probability at least $1 - \tau_D$ and independent hit probability at most τ_I is

$$\Gamma_{\min} \geq \max \left\{ \epsilon_D + \log \frac{1 - \tau_D}{\tau_D}, \epsilon_I + \log \frac{1 - \tau_I}{\tau_I} \right\}. \quad (41)$$

This formula exposes the feasibility trade-off: stronger reliability requirements or larger transfer budgets require larger SCM, which can reduce the retained fingerprint pool.

Lemma 5 (Finite-sample reliability under independent fingerprint indicators). *Let $\mu_M = \mathbb{E}[\text{TA}(M)]$. Suppose*

the hard-label indicators in Eq. 33 are independent conditional on the fixed fingerprint set. If a derived model satisfies $\mu_{M_D} \geq \alpha_D$ and an independent model satisfies $\mu_{M_I} \leq \alpha_I$ with $\alpha_D > \alpha_I$, then under the midpoint threshold in Eq. 40,

$$\Pr[\text{TA}(M_D) < \zeta^*] \leq \exp \left(-\frac{n(\alpha_D - \alpha_I)^2}{2} \right), \quad (42)$$

and

$$\Pr[\text{TA}(M_I) \geq \zeta^*] \leq \exp \left(-\frac{n(\alpha_D - \alpha_I)^2}{2} \right). \quad (43)$$

Both the verifier and the construction filter read a *hard* label, so the final quantity that matters is whether the sampled answer equals t . The target margin of Eq. 18 controls this directly in an idealized constrained decoder. Specifically, under temperature- T softmax sampling normalized over only the K legal labels in $\mathcal{A}$ (or conditional on generation producing a valid legal label),

$$\Pr[\text{sample} = t \mid p] \geq \frac{1}{1 + (K - 1) e^{-m_M(p,t)/T}}. \quad (44)$$

Thus a large source margin $m_{M_0}(p^u, t)$ strengthens target-label stability under this constrained-label idealization. Eq. 44 does not assert that an arbitrary raw-text API samples only from the legal labels; the actual API-facing guarantee remains empirical and is checked by the repeated-query stability filter of Sec. V. Appendix A-B16 gives the detailed qualification and margin-sampling argument.

V. OUR METHOD

As shown in Fig. 5, we propose *Targeted Counterfactual Fingerprinting* (TCF), a black-box LLM ownership verification method based on SCM. The owner constructs prompts that strongly change the protected source LLM’s answer from a clean answer to a selected target answer. During verification, the owner queries a suspect model through its generated text and parses the output into a finite constrained answer space. A model is considered likely derived if it preserves a high fraction of these source-specific target answer transitions.

A. Target Selection

As shown in Fig. 5(A1–A3), target selection starts from a finite-answer candidate question, obtains the source clean answer, and then chooses a clean-unlikely target. For each candidate question x , we first evaluate the clean source prompt $p^0(x)$. The source clean answer is

$$y(x) = \arg \max_{a \in \mathcal{A}} q_{M_0}(a \mid p^0(x)). \quad (45)$$

The target answer should be different from $y(x)$ and should be unlikely under the clean source prompt. We therefore select targets from

$$\mathcal{T}(x) = \{a \in \mathcal{A} \setminus \{y(x)\} : -L_{M_0}(p^0(x), a) \geq \Gamma_{\min}\}. \quad (46)$$

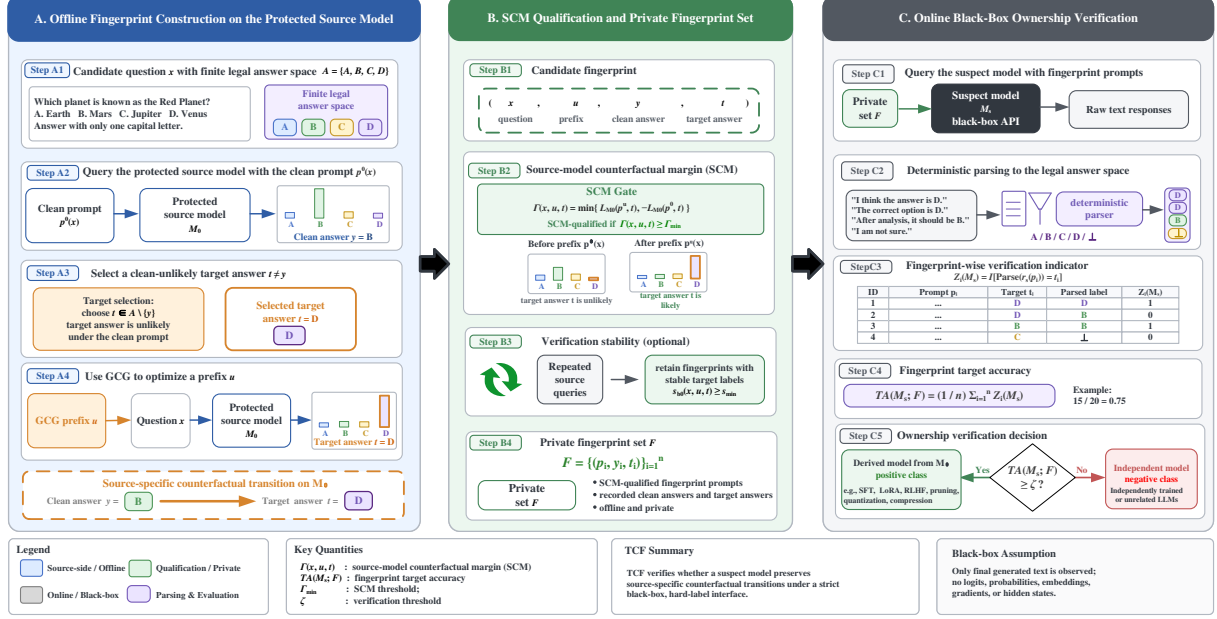


Fig. 5. Overview of Targeted Counterfactual Fingerprinting (TCF). Panel A illustrates fingerprint generation on the protected source model, Panel B illustrates SCM qualification and fingerprint-set construction, and Panel C illustrates black-box ownership verification.

If $\mathcal{T}(x)$ is nonempty, we choose the unlikely valid target,

$$t(x) = \arg \min_{a \in \mathcal{T}(x)} q_{M_0}(a | p^0(x)). \quad (47)$$

If $\mathcal{T}(x)$ is empty, the question is discarded because no target answer is sufficiently counterfactual on the source model.

This target-selection rule controls the clean side of SCM before the perturbation is optimized. After this step, the remaining task is only to control the perturbed side, namely to make $L_{M_0}(p^u(x), t)$ reach the construction target Γ_{target} .

B. Perturbation toward a Target Margin

As shown in Fig. 5(A4), given a selected target t , we optimize a discrete input perturbation (prefix) u with GCG [8]. For exposition, we write the objective using the source constrained-answer probability q_{M_0} ; the differentiable source-side surrogate used during search and the parser-side acceptance check are detailed in Appendix A-B3. The objective is the target negative log-likelihood

$$\mathcal{L}_{\text{src}}(u; x, t) = -\log q_{M_0}(t | p^u(x)). \quad (48)$$

By Lemma 1, the construction target gives an explicit stopping criterion. Since $L_{M_0}(p^u(x), t) \geq \Gamma_{\text{target}}$ is equivalent to $q_{M_0}(t | p^u(x)) \geq \sigma(\Gamma_{\text{target}})$, it is sufficient to optimize until

$$\mathcal{L}_{\text{src}}(u; x, t) \leq \log(1 + e^{-\Gamma_{\text{target}}}), \quad (49)$$

and then *stop early*. We set $\Gamma_{\text{target}} = \Gamma_{\text{min}}$, where the source-side SCM threshold is fixed before verification and is chosen only from source-side feasibility. We use early stopping to avoid unnecessary over-optimization and to retain the first source-qualified perturbation rather than blindly using the final GCG step. This choice is empirically supported by Fig. 4: as target loss decreases, SCM increases, but the strongest empirical SCM regime can also increase generic transfer to independent models. We therefore do not treat “larger SCM” as an unbounded optimization objective. In practice, Eq. 49 is the source-side stopping condition, and the retained candidate is accepted only after its parser-side SCM and source stability are recomputed.

Prefix length. The prefix length ℓ_u controls the search space. A longer prefix usually improves the chance of satisfying $\Gamma(x, u, t) \geq \Gamma_{\text{target}}$ but may reduce naturalness and increase optimization cost.

Optimization budget. Let T_{max} be the maximum number of GCG iterations and B be the number of candidate token replacements evaluated per iteration. Larger values improve success rate but increase construction cost.

Verification stability. Because LLM outputs can vary across repeated queries under stochastic decoding, we optionally query the source model M_0 for R_0 repeated trials with the selected prompt and compute

$$\hat{s}_0(x, u, t) = \frac{1}{R_0} \sum_{r=1}^{R_0} \mathbb{I} \left[\text{Parse}(R_{M_0}^{(r)}(p^u(x))) = t \right]. \quad (50)$$

A candidate is retained only if

$$\hat{s}_0(x, u, t) \geq s_{\min}. \quad (51)$$

We recommend $s_{\min} \in [0.8, 0.95]$. This practical filter is motivated by the margin-sampling analysis of Sec. IV-H: under the constrained-label idealization, a larger source target margin favors target-label stability. The repeated-query filter then checks the actual counterfactual stability under the decoding setting used by the source interface.

To reduce response-format variation at the suspect interface, we append a fixed answer instruction to the GCG-optimized prompt and restrict valid parsed outputs to the legal answer space. Thus, the verification prompt p_i in Eq. 1 is

$$p_i = s_i \oplus q_i \oplus \tau. \quad (52)$$

Here, $s_i = u_i^*$ is the selected GCG prefix, q_i is the underlying multiple-choice question, and τ denotes the fixed answer instruction template that asks M_s to output one capital letter from $\{A, B, C, D\}$:

Question ... A. ... B. ... C. ...
D... Answer with only one capital
letter: A, B, C, or D. Answer:

Algorithm 1 in Appendix A-C gives the complete offline construction procedure. To maintain a practically effective fingerprint set, we additionally enforce diversity over questions, source clean answers, and target answers during selection.

C. Online Black-Box Ownership Verification

As shown in Fig. 5(C1–C5), after constructing the fingerprint set, the owner performs online black-box ownership verification for a suspect model M_s by using only black-box queries as described in Sec. III-B.

Verification threshold selection. ζ is a threshold chosen to control verification errors. Each verification process requires n queries to M_s , followed by a single pass to compute TA. Since the fingerprint pool $\mathcal{F}$ is generated offline and can be reused, multiple independent verification instances can be performed by sampling different subsets $\mathcal{Q}$, enabling repeated verification while preserving query efficiency.

Number of fingerprints. The number of fingerprints n controls statistical reliability. A larger n reduces variance but increases query cost. In the main evaluation, we use 20 fingerprints per protected source, matching the fixed registry described in Sec. VI and Appendix A-D1.

VI. EXPERIMENTS

We organize the experimental evaluation by the following five questions. First, does target-answer transfer separate derived models from independent models? Second, how large is the gap between TCF and existing black-box LLM fingerprinting baselines? Third, is the signal stable under decoding changes and simple adaptive perturbations? Fourth, how robust is TCF to adaptive attack? Finally, what is the computational cost compared with the evaluated baselines?

A. Experimental Setup

Source models. We evaluate TCF on four representative open-source LLMs as protected source models: Qwen3-1.7B Instruct [34], Qwen3-1.7B Base [34], Llama3-8B Base Direct [35], [36], and Mistral-7B-v0.3 [37]. The public checkpoint identifiers on Hugging Face are listed in Table II.

TABLE II
PROTECTED SOURCE MODELS AND SUSPECT-MODEL POOLS.

Source	Checkpoint identifier	Total	Source	Derived	Independent
Qwen3-1.7B Instruct	Qwen/Qwen3-1.7B	19	1	9	9
Qwen3-1.7B Base	Qwen/Qwen3-1.7B-Base	22	1	12	9
Llama3-8B Base Direct	meta-llama/Meta-Llama-3-8B	25	1	13	11
Mistral-7B-v0.3	mistralai/Mistral-7B-v0.3	25	1	16	8

Suspect model pool. For each protected source, we evaluate the source checkpoint, direct model positives, and independent negatives. Direct model positives cover supervised/task fine-tuning (SFT), RL-style adaptation (RL), language/domain adaptation, pruning, and other direct derivatives. Independent checkpoints are used only as independent negatives. Table II lists the fixed source/derived/independent model counts used for the main AUC in Table III. Appendix Tab. XI provides the extended checkpoint catalog, and Appendix A-E reports per-suspect diagnostics for checkpoints with the corresponding baseline artifacts.

Fingerprint Generation. The fingerprint pool is built from Massive Multitask Language Understanding (MMLU) multiple-choice questions [38]. Each source model uses 20 fingerprints from four different subject categories, with five questions per category. The four-domain design prevents the fingerprint set from being tied to a single topic and improves fingerprint diversity. The details are in Appendix A-D.

Baselines. We compare with three state-of-the-art black-box LLM fingerprinting baselines: TRAP [1], ProFLingo [2], and ZeroPrint [3]. Appendix Tab. X specifies the verification protocol and score definition for each method.

B. Main Verification Results

Table III reports the main verification AUC results. **TCF achieves perfect AUC on Qwen3-1.7B Instruct,**

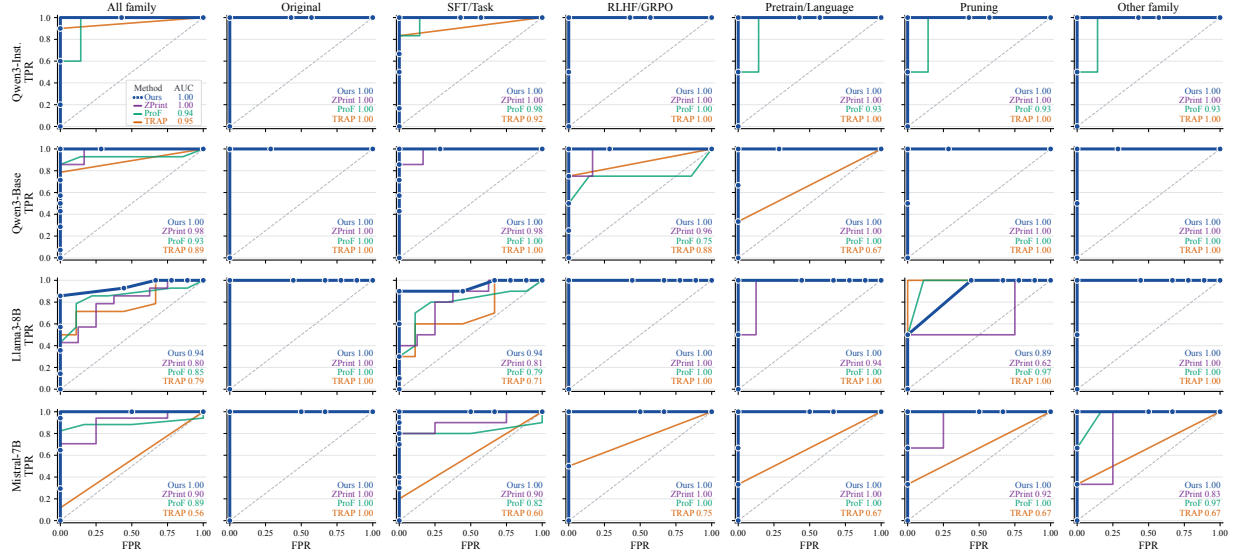


Fig. 6. Diagnostic subgroup ROC curves on the shared per-suspect diagnostic pool. Primary family-level AUCs are reported in Table III.

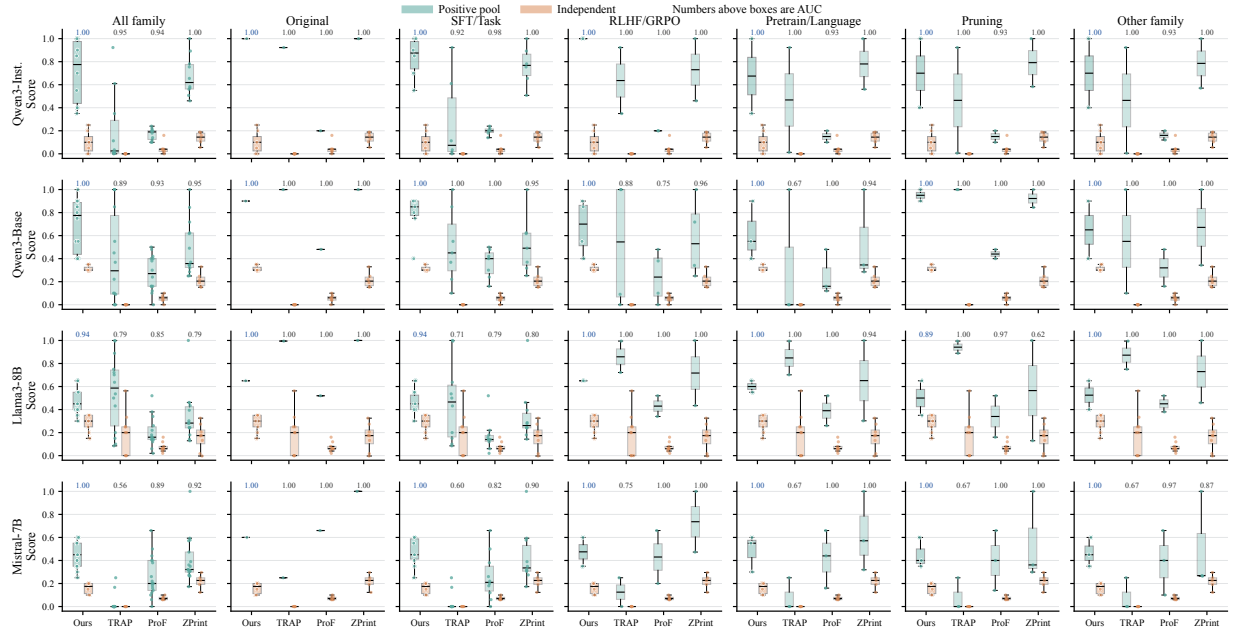


Fig. 7. Fingerprint score distributions on the shared per-suspect diagnostic pool used for the subgroup diagnostics. Primary family-level AUCs are reported in Table III.

Qwen3-1.7B Base, and Mistral-7B-v0.3, and reaches 0.9444 on Llama3-8B Base Direct. Its average verification AUC across the four sources is 0.9861, compared with 0.7931 for TRAP, 0.8880 for ProFLingo, and 0.9128 for ZeroPrint. The strongest baseline is model-dependent: ZeroPrint is the strongest baseline on both Qwen3 sources and on Mistral, whereas ProFLingo is strongest on Llama3. Overall, TCF matches the strongest baseline on Qwen3-1.7B Instruct and strictly exceeds the strongest

baseline on the other three source families, supporting our key intuition: **a constrained-answer targeted counterfactual transfer signal, realized via a source-optimized input perturbation, can act as a strong black-box proxy for LLM ownership verification.**

TABLE III
MAIN AUC COMPARISON. BEST RESULTS ARE BOLD.

Source model	TCF	TRAP	ProFLingo	ZeroPrint
Qwen3-1.7B Instruct	1.0000	0.9286	0.9796	1.0000
Qwen3-1.7B Base	1.0000	0.8846	0.9231	0.9722
Llama3-8B Base Direct	0.9444	0.7593	0.8241	0.7885
Mistral-7B-v0.3	1.0000	0.6000	0.8250	0.8906
Average	0.9861	0.7931	0.8880	0.9128

Takeaway

TCF matches or exceeds the strongest evaluated baseline across the four source families while maintaining strong evidence across diverse derivative categories.

Fingerprint score. For the per-suspect diagnostic pool with available artifacts, Table IV reports the fingerprint scores of source models, derived models, and independent models for TCF. Figure 6 shows the corresponding ROC curves over the source model, all available derivatives, and derivative subgroups, while Fig. 7 shows the score distributions. These diagnostic plots expose model-wise and subgroup heterogeneity; the primary family-level comparison remains Table III.

TABLE IV
MEAN TCF FINGERPRINT SCORE ON THE PER-SUSPECT DIAGNOSTIC POOL. THE SCORE IS THE TARGET-LABEL ACCURACY OVER GENERATED FINGERPRINTS.

Source model	Source score	Derived mean	Indep. mean
Qwen3-1.7B Instruct	1.0000	0.6833	0.1000
Qwen3-1.7B Base	0.9000	0.6731	0.3143
Llama3-8B Base Direct	0.6500	0.4538	0.2889
Mistral-7B-v0.3	0.6000	0.4281	0.1583

Takeaway

Within this diagnostic pool, derived models retain the source fingerprint more strongly than independent models, while Llama3 remains the most challenging source family.

TABLE V
COMMERCIAL BLACK-BOX API SOURCE-VS-INDEPENDENT VERIFICATION RESULTS ON OPENROUTER.

Method	Source	Qwen3-8B	Gemma-3-12B	Gemma-3-4B	AUC
Ours	0.294	0.000	0.118	0.063	1.000
TRAP	0.450	0.000	0.000	0.000	1.000
ProFLingo	0.100	0.100	0.000	0.000	0.833
ZeroPrint	0.483	0.490	0.538	0.476	0.333

Commercial black-box API evaluation. We further evaluate whether the fingerprints remain separable when model access is mediated by a mainstream commercial black-box API. We use **OpenRouter** [39] as the API provider and query `meta-llama/llama-3-8b-instruct` as the protected source model. Since OpenRouter does not

provide the same controlled set of Llama-3 family derivatives used in our evaluation, this experiment focuses on the source-versus-independent API test. The independent API models are `qwen/qwen3-8b` [34], `google/gemma-3-12b-it` [40], and `google/gemma-3-4b-it` [40]. For TRAP, ProFLingo, and ZeroPrint, we use each baseline’s original black-box scoring setting. Although both TCF and TRAP achieve the best AUC in this commercial black-box API test, the results are not clearly equivalent: the API test obscures TRAP’s weak derivative model transfer, whereas our method retains stronger family-level evidence under controlled evaluation.

Takeaway

TCF remains separable through a commercial text-generation API using only parsed final responses, showing that suspect-side verification does not depend on white-box access.

Model distillation. Model extraction is a strong threat for ownership verification because an adversary may train a surrogate from query-output pairs rather than directly reusing the source checkpoint [4]. We therefore evaluate a clean-query distillation setting from `mistralai/Mistral-7B-v0.3` to a Llama-2-7B student [41]. The attacker queries the teacher only with clean MMLU-style multiple-choice prompts and trains the student with LoRA SFT on the teacher’s hard-label outputs. The reported extracted student is trained for three epochs on 2472 valid clean teacher-label pairs.

For utility, we evaluate on MMLU-CF [42], using the same hard-label generation prompt, *Answer with only one capital letter: A, B, C, or D.*, deterministic decoding, and at most eight new tokens. We parse each completion into A/B/C/D/INVALID. Clean100 Acc. is the parsed multiple-choice accuracy, and T-Agree is the parsed agreement with the teacher prediction; the teacher’s own T-Agree is defined as 1.00. As shown in Table VI, distillation improves the student’s Clean100 accuracy from 0.53 to 0.69 and T-Agree from 0.49 to 0.68, showing that the surrogate absorbs useful teacher behavior from clean queries. Although extraction attacks can erase fine-grained watermark evidence, our fingerprint remains effective on the distilled student: its AUC increases from 0.31 before distillation to 0.88 after distillation. In contrast, TRAP remains at tie-level AUC, ProFLingo does not retain a transferable signal, and ZeroPrint decreases after distillation. Thus, TCF retains an effective ownership signal under clean-query model distillation.

Takeaway

The fingerprint is inherited by a student trained only from clean teacher outputs, supporting ownership verification under model extraction.

TABLE VI
MODEL DISTILLATION VERIFICATION RESULTS.

Model	Clean100 Acc.	T-Agree	Ours	TRAP	ProF	ZPrint
Mistral teacher	0.79	1.00	1.00	1.00	1.00	1.00
Llama-2 untrained	0.53	0.49	0.31	0.50	0.06	0.56
Llama-2 extracted	0.69	0.68	0.88	0.50	0.00	0.25

C. Robustness

Input abnormal detection. The original GCG prefix is effective but can contain unnatural token fragments, which can be exposed by PPL or window-PPL-based input-abnormality detection. We therefore evaluate naturalness-oriented presentation variants while keeping source-only construction and the hard-label verifier unchanged. NaturalCarrier wraps optimized fragments with short cue-like carrier phrases, while TCF-S4 and TCF-S8 distribute the optimized positions over four or eight insertion groups in the original question. The details and diagnostic-only variants are documented in Appendix A-D4.

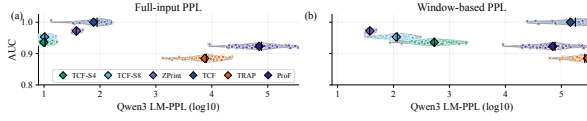


Fig. 8. Naturalness-verification trade-off. Full-input and window-based PPL show complementary views of prompt naturalization.

We then compute and visualize full-input PPL and window-based PPL of the original centralized prefix TCF, and its variants with the above naturalness-oriented strategies, and the compared baselines. As shown in Fig. 8, among these strategies and the evaluated baselines, the strategy of splitting the centralized prefix into randomly selected token positions TCF-S4/8 yields the lowest PPL while maintaining the highest verification AUC compared with the strongest normal-text-based baseline, ZeroPrint.

Takeaway

Distributed NaturalCarrier variants improve prompt naturalness while preserving strong verification performance.

Adaptive attack. To assess whether a lightweight LoRA adapter can erase fingerprint behavior when partial fingerprint samples are leaked (50% by default in our implementation) while preserving clean-answer utility, we conduct a LoRA adaptive attack. The attacker freezes the base model and trains only a low-rank LoRA adapter on leaked fingerprint examples mixed with normal clean-answer data. For the condition labels, Rand denotes a random leaked subset, numbers in K1–K4 denote the number of leaked subjects, where U denotes an unbiased adaptive finetuning by mixing the leaked fingerprints and normal samples (questions and answers sampled

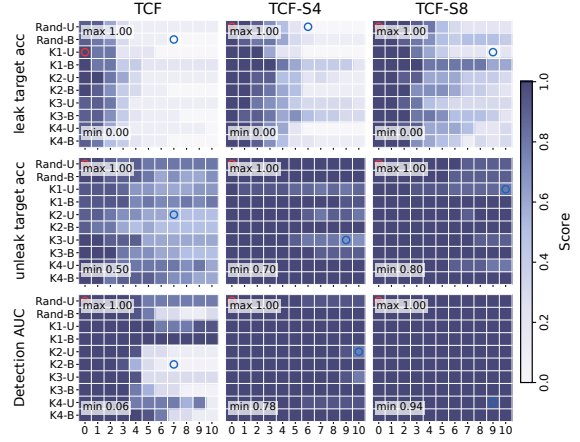


Fig. 9. Verification results under adaptive attack. Rows report leaked-fingerprint target accuracy, non-leaked target accuracy, and detection AUC. Columns correspond to fingerprinting methods. The horizontal axis indexes LoRA finetuning epochs, and the vertical axis indexes leakage conditions. Red and blue circles mark the maximum and minimum values in each panel, respectively.

from MMLU) in a 1:1 ratio, and B denotes a biased 1:10 leaked-to-normal ratio. In each finetuning epoch, the constructed training batch is fed once to update the adapter. From the results in Fig. 9, we found that such an adaptive finetuning attack can strongly suppress the target fingerprint accuracy of the leaked fingerprints, leading to the leaked target accuracy gradually dropping to 0.00 under enough training epochs. However, this suppression does not reliably transfer to the un-leaked fingerprints. For TCF, un-leaked target accuracy remains nontrivial across the sweep. NaturalCarrier is more robust: TCF-S4 keeps un-leaked target accuracy at least 0.70, and TCF-S8 keeps it at least 0.80. This indicates a clear split effect: increasing the number of NaturalCarrier splits makes the fingerprint more resistant to adaptive finetuning attack, because the attacker observes and erases only part of the distributed fingerprint signal. Overall, *while adaptive attack by LoRA erasure can remove memorized leaked fingerprints, split-distributed fingerprints retain stronger private transfer evidence*. Among the NaturalCarrier variants, TCF-S8 is more robust than TCF-S4 under adaptive LoRA training.

Takeaway

Partial-leakage LoRA erasure suppresses exposed fingerprints but does not reliably remove private distributed fingerprints.

To explain why the split variants are more robust for the adaptive attack, we examine the hidden target margin, defined as the score gap between the recorded target label and the strongest non-target valid label. A positive margin means that the target remains the top-

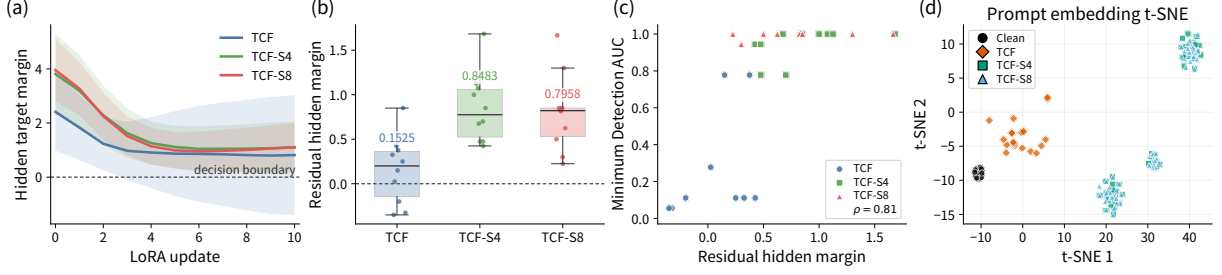


Fig. 10. Margin-boundary analysis of adaptive LoRA erasure: hidden margin trajectories (a), residual hidden margin distributions (b), residual hidden margin versus minimum Detection AUC (c), and prompt-embedding t-SNE visualization (d).

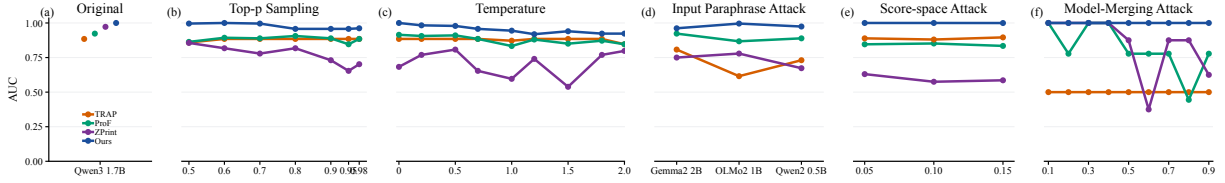


Fig. 11. Robustness under sampling, input paraphrasing, score-space perturbation, and model-merging.

scoring valid label, while a non-positive margin indicates boundary crossing and effective erasure. Fig. 10 shows that LoRA erasure reduces hidden margins for all variants, but the unsplit Original variant is more likely to cross the boundary, with a higher hidden normalized erasure sufficiency ($NES=1.3845$), a higher boundary-crossing rate (0.3967), and a smaller residual hidden margin (0.1525). In contrast, the split variants TCF-S4 and TCF-S8 have much lower boundary-crossing rates (0.1767 and 0.1667) and retain larger residual margins (0.8483 and 0.7958). The boundary-crossing rate further correlates with hidden target-hit degradation (Spearman $\rho = 0.8681$), while the residual hidden margin correlates with the minimum Detection AUC ($\rho = 0.8070$). These results suggest that splitting improves robustness by preserving a larger hidden-margin buffer, making leaked-update erasure less likely to push hidden fingerprints across the hard-label decision boundary. In the prompt-embedding visualization of Fig. 10(d), TCF-S4 and TCF-S8 form multiple sub-clusters instead of one concentrated fingerprint cluster, a pattern consistent with higher fingerprint diversity after splitting. This observation is consistent with less complete coverage by leaked LoRA erasure and with the lower boundary-crossing rates of the split variants.

Takeaway

Split fingerprints resist erasure by preserving larger hidden target margins and more diverse prompt patterns.

Decoding and sampling stability. To evaluate the robustness of different fingerprinting methods under varying decoding configurations, we compared all methods

using the same Qwen3-1.7B Base fingerprints while systematically altering verification-stage decoding parameters (Fig. 11(b)–(c)). We observed that TCF maintains near-perfect AUC under all tested nucleus-sampling settings and remains above 0.91 across the entire temperature sweep. In contrast, the baseline methods either yield consistently lower AUC or exhibit a more pronounced degradation under several settings. This superior robustness can be attributed to the legal-answer space and the SCM-qualified target transfer, which preserve a larger derived-independent separation than TRAP, ProFLingo, and ZeroPrint even when decoding randomness changes.

Takeaway

TCF remains stable under common decoding changes, with high temperature as the main source of degradation.

Input paraphrasing and score-space perturbation. To examine resilience to lightweight input rewriting and score-level noise, we compare TCF with the baselines in Fig. 11(d)–(e). Paraphrasing degrades all prompt-based signals, yet TCF consistently achieves the highest AUC across the three paraphrase models—Gemma 2 2B-IT [43], OLMo 2 1B-Instruct [44], and Qwen2 0.5B-Instruct [45]. This indicates that the target-transfer event survives moderate surface rewriting more robustly than response-reproduction or similarity-based scores. Panel (e) is a score-space stress test rather than a raw-text API transformation: clipped Gaussian noise $\epsilon \sim \mathcal{N}(0, \sigma^2)$ with $\sigma \in \{0.05, 0.10, 0.15\}$ is added in score space before AUC computation. TCF remains essentially robust while the baselines stay lower, indicating that the reported

separation is not fragile to moderate score-level noise.

Takeaway

TCF is more resilient than the baselines to moderate paraphrasing and score-space noise.

Model merging. Fig. 11(f) evaluates a model-merging attack on Qwen3-1.7B, where the fingerprinted victim checkpoint Qwen3-1.7B-Base is linearly interpolated with a non-fingerprinted reference checkpoint Qwen3-1.7B: $\theta_{\text{merge}} = \alpha\theta_{\text{victim}} + (1 - \alpha)\theta_{\text{ref}}$. We sweep α and re-run fingerprint verification. TRAP drops to chance-level AUC, while ProFLingo and ZeroPrint show unstable AUC across merge ratios. In contrast, TCF keeps AUC at 1.0 throughout the sweep, showing that its hard-label fingerprint signal remains stable under this linear merging attack.

Takeaway

TCF remains stable under the evaluated linear model-merging attack.

D. Ablation Studies

To investigate the impact of key hyperparameters on verification performance, we conducted ablation studies on three aspects: the number of fingerprints n , the GCG prefix length (PL), and the target-answer rank. Fig. 12 summarizes the main results under the promoted hard-label setting.

Number of fingerprints n : To determine how many fingerprints are sufficient for stable verification, we varied n from 5 to 20 in Fig. 12(a). We observed that performance improves rapidly from $n = 5$ to $n = 10$ and then saturates in the Qwen3-1.7B Base setting, indicating that a modest fingerprint set is adequate.

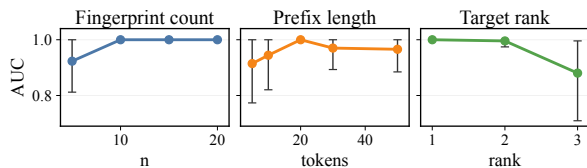


Fig. 12. Ablation study results of fingerprint count (a), prefix length (b), and target answer selection (c).

Takeaway

A modest fingerprint set is sufficient for stable verification in the main setting.

GCG prefix length (PL): To assess the effect of prefix length on separability, we tested different PL values in Fig. 12(b). We found that moderate prefixes are sufficient, while overly short or overly long prefixes can weaken the derived-independent separation. This is consistent with the feasible operating-region interpretation

of Section IV-G: overly weak perturbations may fail to create a source-side counterfactual transition, whereas overly strong perturbations can increase generic transfer to independent models. The empirical optimum therefore need not occur at the largest prefix length or largest SCM.

Takeaway

A moderate prefix length best balances target induction and independent-model non-transfer.

Target-answer selection: To understand how the choice of target label influences verification, we performed a source-model diagnostic by varying the rank of the target answer in Fig. 12(c). Although Top-1, Top-2, and Top-3 target choices all achieve AUC 1.0000 in this diagnostic, the independent mean varies. This suggests that target rank can affect false-positive behavior. Therefore, we use this target-rank panel to guide target selection rather than as a standalone full-model-pool ownership result.

Takeaway

Target selection should minimize accidental independent preference rather than optimize source success alone.

E. Computational Cost

To evaluate per-fingerprint generation, verification, and storage overhead, we conduct a comparative analysis on Qwen3-1.7B Base (Tab. VII). TRAP is the most expensive among the compared methods, requiring 8460.5 seconds per fingerprint under its official setting. ZeroPrint has fast source-side generation but incurs high verification cost because it extracts perturbation-based fingerprints from suspect models. ProFLingo is efficient during verification but uses a larger 50-query registry. In contrast, TCF requires 1269.3 seconds per fingerprint and uses a 20-query registry in the main evaluation, giving lower construction/verification overhead than the evaluated baselines while retaining the strongest verification performance.

TABLE VII
PER-FINGERPRINT COMPUTATION AND STORAGE COST ON QWEN3-1.7B BASE.

Method	Total time	Generation time	Verify time	Generation storage	Eval storage
TCF	1269.3s	1200.6s	68.7s	1341.2 KiB	75.2 KiB
TRAP	8460.5s	7835.9s	624.7s	234.3 KiB	290.3 KiB
ProFLingo	1625.0s	1550.1s	74.9s	133.9 KiB	0.2 KiB
ZeroPrint	1954.5s	44.0s	1910.5s	48387.6 KiB	47.6 KiB

Takeaway

TCF combines the strongest verification performance with lower per-fingerprint overhead and a compact 20-query registry.

VII. CONCLUSION

We present TCF, a non-invasive fingerprinting framework for black-box LLM ownership verification. The key idea is to avoid open-ended generation by reformulating verification as constrained-answer target transfer. TCF constructs perturbed prompts that move the protected model from its clean answer to a counterfactual target, then verifies a suspect model by measuring target-answer transfer over the private fingerprint set. Theoretical analysis derives target-transfer bounds and a sufficient positive-gap condition under explicit derived-preservation and independent-transfer budgets motivated by local behavioral closeness; aggregation then converts a per-fingerprint edge into a thresholdable ownership signal. Empirically, TCF achieves near-perfect AUC across four source families and matches or outperforms the strongest evaluated black-box fingerprinting baseline on every source family.

REFERENCES

- [1] M. Gubri, D. Ulmer, H. Lee, S. Yun, and S. J. Oh, “TRAP: Targeted random adversarial prompt honeypot for black-box identification,” in *Findings of the Association for Computational Linguistics: ACL 2024*, 2024, pp. 11 496–11 517.
- [2] H. Jin, C. Zhang, S. Shi, W. Lou, and Y. T. Hou, “ProFLingo: A fingerprinting-based intellectual property protection scheme for large language models,” in *2024 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2024, pp. 1–9.
- [3] S. Shao, Y. Li, H. Yao, Y. Chen, Y. Yang, and Z. Qin, “Reading between the lines: Towards reliable black-box LLM fingerprinting via zeroth-order gradient estimation,” in *Proceedings of the ACM Web Conference 2026*. ACM, 2026.
- [4] N. Carlini, D. Paleka, K. Dvijotham, T. Steinke, J. Hayase, A. F. Cooper, K. Lee, M. Jagielski, M. Nasr, A. Conmy *et al.*, “Stealing part of a production language model,” in *International Conference on Machine Learning*, 2024, pp. 5680–5705.
- [5] J. Kirchenbauer, J. Geiping, Y. Wen, J. Katz, I. Miers, and T. Goldstein, “A watermark for large language models,” in *International Conference on Machine Learning*. PMLR, 2023, pp. 17 061–17 084.
- [6] S. Abdelnabi and M. Fritz, “Adversarial watermarking transformer: Towards tracing text provenance with data hiding,” in *2021 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2021, pp. 121–140.
- [7] Z. Hu, L. Chen, X. Wu, Y. Wu, H. Zhang, and H. Huang, “Unbiased watermark for large language models,” 2023.
- [8] A. Zou, Z. Wang, N. Carlini, M. Nasr, J. Z. Kolter, and M. Fredrikson, “Universal and transferable adversarial attacks on aligned language models,” 2023.
- [9] P.-Y. Chen, H. Zhang, Y. Sharma, J. Yi, and C.-J. Hsieh, “ZOO: Zeroth order optimization based black-box attacks to deep neural networks without training substitute models,” in *Proceedings of the 10th ACM Workshop on Artificial Intelligence and Security*, 2017, pp. 15–26.
- [10] Y. Adi, C. Baum, M. Cisse, B. Pinkas, and J. Keshet, “Turning your weakness into a strength: Watermarking deep neural networks by backdooring,” in *27th USENIX Security Symposium (USENIX Security 18)*, 2018, pp. 1615–1631.
- [11] Y. Uchida, Y. Nagai, S. Sakazawa, and S. Satoh, “Embedding watermarks into deep neural networks,” in *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*. ACM, 2017, pp. 269–277.
- [12] J. Zhang, Z. Gu, J. Jang, H. Wu, M. P. Stoecklin, H. Huang, and I. Molloy, “Protecting intellectual property of deep neural networks with watermarking,” in *Proceedings of the 2018 on Asia Conference on Computer and Communications Security*. ACM, 2018, pp. 159–172.
- [13] B. D. Rouhani, H. Chen, and F. Koushanfar, “DeepSigns: An end-to-end watermarking framework for ownership protection of deep neural networks,” in *Proceedings of the Twenty-Fourth International Conference on Architectural Support for Programming Languages and Operating Systems*. ACM, 2019, pp. 485–497.
- [14] E. Le Merrer, P. Pérez, and G. Trédan, “Adversarial frontier stitching for remote neural network watermarking,” *Neural Computing and Applications*, vol. 32, no. 13, pp. 9233–9244, 2019.
- [15] C. Gu, C. Huang, X. Zheng, K.-W. Chang, and C.-J. Hsieh, “Watermarking pre-trained language models with backdooring,” 2022.
- [16] P. Li, P. Cheng, F. Li, W. Du, H. Zhao, and G. Liu, “PLMmark: A secure and robust black-box watermarking framework for pre-trained language models,” in *Proceedings of the AAAI Conference on Artificial Intelligence*, vol. 37, no. 12, 2023, pp. 14 991–14 999.
- [17] S. Li, L. Yao, J. Gao, L. Zhang, and Y. Li, “Double-i watermark: Protecting model copyright for llm fine-tuning,” 2024.
- [18] Y. Xu, A. Liu, X. Hu, L. Wen, and H. Xiong, “Mark your LLM: Detecting the misuse of open-source large language models via watermarking,” in *The 1st Workshop on GenAI Watermarking*, 2025. [Online]. Available: <https://openreview.net/forum?id=JGTRj6h0Cv>
- [19] J. Xu, F. Wang, M. D. Ma, P. W. Koh, C. Xiao, and M. Chen, “Instructional fingerprinting of large language models,” in *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*. Mexico City, Mexico: Association for Computational Linguistics, 2024, pp. 3277–3306. [Online]. Available: <https://aclanthology.org/2024.naacl-long.180/>
- [20] Z. Xu, X. Zhao, X. Yue, S. Tian, C. Lin, and M. Han, “CTCC: A robust and stealthy fingerprinting framework for large language models via cross-turn contextual correlation backdoor,” in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, 2025, pp. 6967–6989. [Online]. Available: <https://aclanthology.org/2025.emnlp-main.356/>
- [21] Z. Xu, M. Han, and W. Xing, “EverTracer: Hunting stolen large language models via stealthy and robust probabilistic fingerprint,” in *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*. Suzhou, China: Association for Computational Linguistics, 2025, pp. 7008–7031. [Online]. Available: <https://aclanthology.org/2025.emnlp-main.358/>
- [22] N. Jovanovic, R. Staab, and M. Vechev, “Watermark stealing in large language models,” 2024.
- [23] S. Chakraborty, A. S. Bedi, S. Zhu, B. An, D. Manocha, and F. Huang, “On the possibilities of ai-generated text detection,” 2023.
- [24] V. S. Sadasivan, A. Kumar, S. Balasubramanian, W. Wang, and S. Feizi, “Can ai-generated text be reliably detected?” 2023.
- [25] X. Cao, J. Jia, and N. Z. Gong, “IPGuard: Protecting intellectual property of deep neural networks via fingerprinting the classification boundary,” in *Proceedings of the 2021 ACM Asia Conference on Computer and Communications Security*. ACM, 2021, pp. 14–25.
- [26] J. Chen, J. Wang, T. Peng, Y. Sun, P. Cheng, S. Ji, X. Ma, B. Li, and D. Song, “Copy, right? a testing framework for copyright protection of deep learning models,” in *2022 IEEE Symposium on Security and Privacy (SP)*. IEEE, 2022, pp. 824–841.
- [27] N. Lukas, Y. Zhang, and F. Kerschbaum, “Deep neural network fingerprinting by conferrable adversarial examples,” in *International Conference on Learning Representations*, 2021.
- [28] Z. Peng, S. Li, G. Chen, C. Zhang, H. Zhu, and M. Xue, “Fingerprinting deep neural networks globally via universal adversarial perturbations,” in *Proceedings of the IEEE/CVF Conference on*

Computer Vision and Pattern Recognition, 2022, pp. 13 420–13 429.

- [29] T. Xu, C. Wang, G. Liu, Y. Yang, K. Peng, and W. Liu, “United we stand, divided we fall: Fingerprinting deep neural networks via adversarial trajectories,” in *Advances in Neural Information Processing Systems*, 2024.
- [30] K. Yang and K. Lai, “NaturalFinger: Generating natural fingerprint with generative adversarial networks,” 2023.
- [31] A. Yan, H. Ren, K. Mo, Z. Zhang, S. Wang, and J. Li, “Enhancing model intellectual property protection with robustness fingerprint technology,” *IEEE Transactions on Information Forensics and Security*, vol. 20, pp. 9235–9249, 2025.
- [32] B. Zeng, L. Wang, Y. Hu, Y. Xu, C. Zhou, X. Wang, Y. Yu, and Z. Lin, “HuRef: HUMAN-REAdable fingerprint for large language models,” in *Advances in Neural Information Processing Systems*, vol. 37, 2024, pp. 126 332–126 362.
- [33] J. Zhang, D. Liu, C. Qian, L. Zhang, Y. Liu, Y. Qiao, and J. Shao, “REEF: Representation encoding fingerprints for large language models,” in *The Thirteenth International Conference on Learning Representations*, 2025. [Online]. Available: <https://openreview.net/forum?id=SnDmPkOJ0T>
- [34] A. Yang, A. Li *et al.*, “Qwen3 technical report,” *arXiv preprint arXiv:2505.09388*, 2025, qwen3 model family; Qwen3-1.7B Base: <https://huggingface.co/Qwen/Qwen3-1.7B-Base>; Qwen3-1.7B Instruct: <https://huggingface.co/Qwen/Qwen3-1.7B>; Qwen3-8B: <https://huggingface.co/Qwen/Qwen3-8B>. [Online]. Available: <https://arxiv.org/abs/2505.09388>
- [35] AI@Meta, “Llama 3 model card,” 2024, llama3-8B Base Direct variant: <https://huggingface.co/meta-llama/Meta-Llama-3-8B>. [Online]. Available: https://github.com/meta-llama/llama3/blob/main/MODEL_CARD.md
- [36] A. Grattafiori, A. Dubey *et al.*, “The llama 3 herd of models,” *arXiv preprint arXiv:2407.21783*, 2024. [Online]. Available: <https://arxiv.org/abs/2407.21783>
- [37] Mistral AI, “Mistral-7b-v0.3,” may 2024, released on May 22, 2024, with 32,768 token vocabulary and v3 tokenizer support. [Online]. Available: <https://huggingface.co/mistralai/Mistral-7B-v0.3>
- [38] D. Hendrycks, C. Burns, S. Basart, A. Zou, M. Mazeika, D. Song, and J. Steinhardt, “Measuring massive multitask language understanding,” in *International Conference on Learning Representations (ICLR)*, 2021. [Online]. Available: <https://openreview.net/forum?id=YC60R30cKX>
- [39] OpenRouter, “Openrouter api documentation,” <https://openrouter.ai/docs/api/reference/overview>, 2026, accessed: 2026-06-09.
- [40] Gemma Team, “Gemma 3 technical report,” 2025. [Online]. Available: <https://arxiv.org/abs/2503.19786>
- [41] H. Touvron, L. Martin, K. Stone, P. Albert, A. Almahairi, Y. Babaei, N. Bashlykov, S. Batra, P. Bhargava, S. Bhosale *et al.*, “Llama 2: Open foundation and fine-tuned chat models,” 2023.
- [42] Q. Zhao, Y. Huang, T. Lv, L. Cui, Q. Sun, S. Mao, X. Zhang, Y. Xin, Q. Yin, S. Li, and F. Wei, “MMLU-CF: A contamination-free multi-task language understanding benchmark,” in *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*. Vienna, Austria: Association for Computational Linguistics, Jul. 2025, pp. 13 371–13 391. [Online]. Available: <https://aclanthology.org/2025.acl-long.656/>
- [43] Gemma Team, M. Riviere, S. Pathak, P. G. Sessa *et al.*, “Gemma 2: Improving open language models at a practical size,” *arXiv preprint arXiv:2408.00118*, 2024.
- [44] OLMO Team, P. Walsh, L. Soldaini, D. Groeneveld *et al.*, “2 olmo 2 furious,” *arXiv preprint arXiv:2501.00656*, 2024.
- [45] A. Yang, B. Yang, B. Hui, B. Zheng *et al.*, “Qwen2 technical report,” *arXiv preprint arXiv:2407.10671*, 2024.

APPENDIX A ADDITIONAL EXPERIMENTAL DETAILS

Appendix roadmap

This appendix is organized as a reproducibility record. It first validates the complete SCM evidence chain across additional source families, then provides the proofs and finite-sample details behind the theoretical claims, and finally documents the construction protocol, exact hard-label interface, prompt examples, baseline settings, model checkpoints, and retained evaluation artifacts.

A. Cross-Model Theory Validation

The main theoretical analysis is not specific to the Qwen3-1.7B Base diagnostic used for exposition. We therefore repeat the complete evidence chain on three additional protected source families: Qwen3 Instruct, Llama3, and Mistral. Across these families, source-side optimization increases SCM, derived models preserve the induced target preference more strongly than independent models, and the resulting derived-independent target-answer gap generally grows in higher-SCM regimes. These experiments directly test the portability of the theory rather than introducing a separate empirical claim.

1) *Source-Side SCM Control*: Figure 13 verifies that the source-side optimization behaves consistently across the three additional families. Target loss decreases during GCG optimization, mean SCM increases, and the two quantities remain strongly negatively associated. This supports the source-side implication used in Lemma 1 beyond the single diagnostic model shown in the main text.

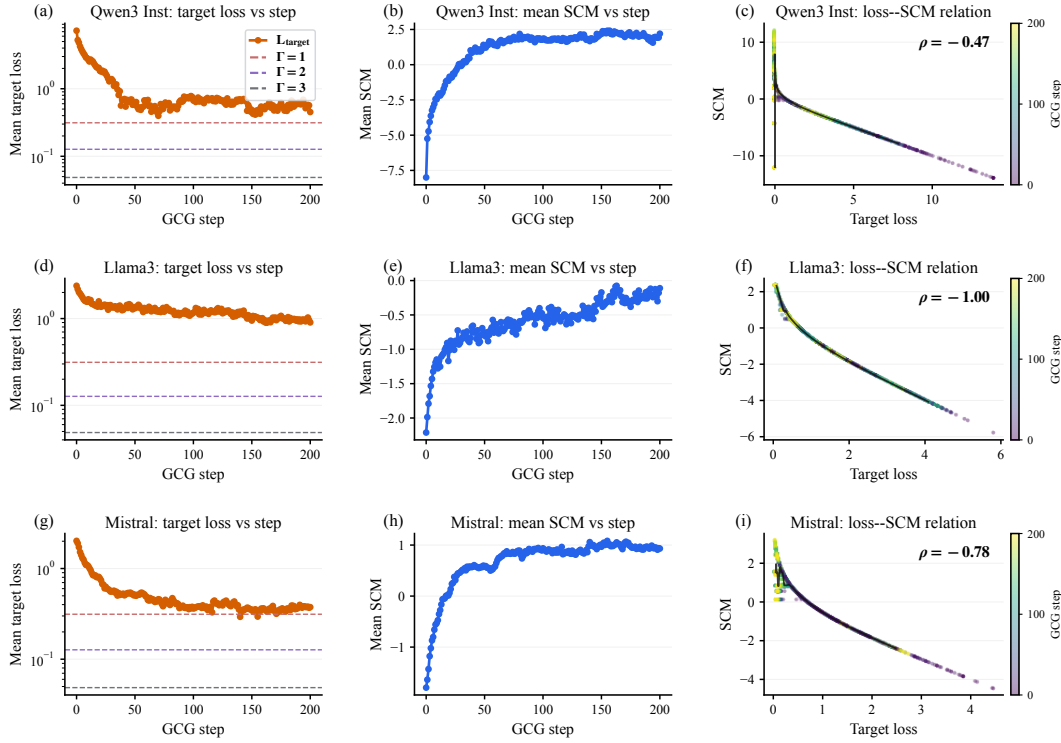


Fig. 13. SCM-control diagnostics on three additional protected source families. Source target loss decreases during optimization, mean SCM increases, and lower target loss is associated with stronger SCM.

Takeaway

The source-loss-to-SCM relationship generalizes across the additional protected model families.

2) *Derived-Model Preservation*: Figure 14 evaluates the derived-model weakening budget ϵ_D . For each source family, the empirical drift distribution and the corresponding lower-bound curves show that source-family derivatives retain a substantial part of the optimized target preference. Smaller weakening budgets and higher curves indicate stronger preservation.

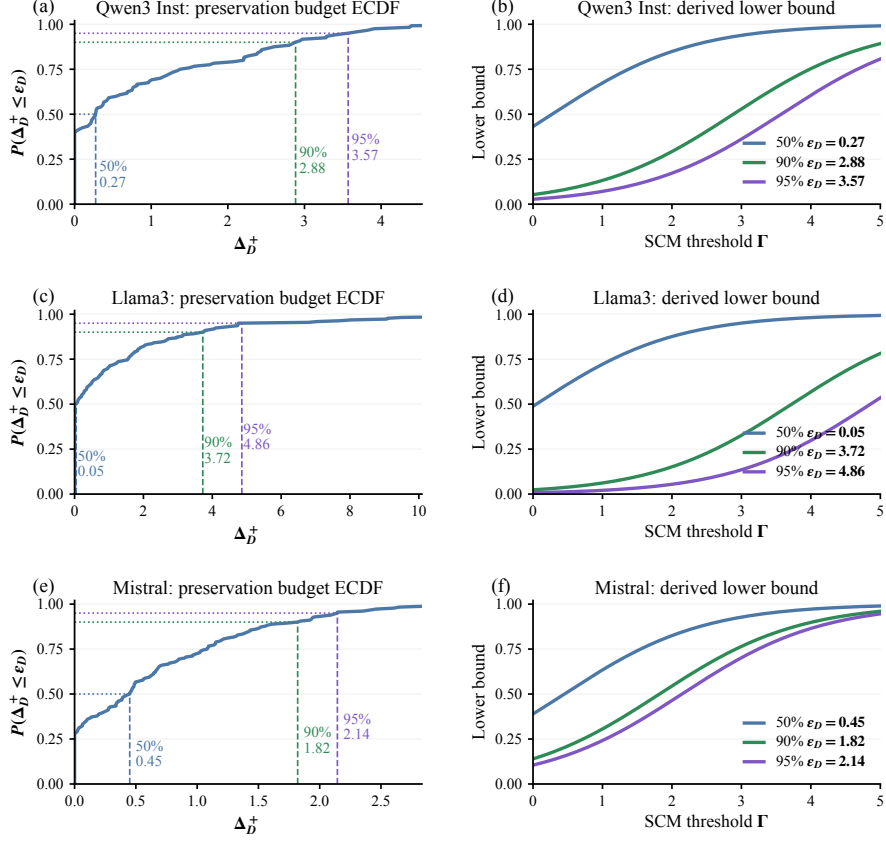


Fig. 14. Derived-model preservation on the additional source families. Each row reports the empirical weakening-budget coverage and the corresponding lower-bound trend $\sigma(\Gamma - \epsilon_D)$.

Takeaway

Derived-model preservation remains consistent across the additional protected model families.

3) *Independent-Model Non-Transfer*: Figure 15 evaluates the independent-model slack ϵ_I . Independent models generally require larger slack budgets to explain the same target behavior, showing that the optimized prompts do not preserve the source-side target preference as consistently outside the protected model family.

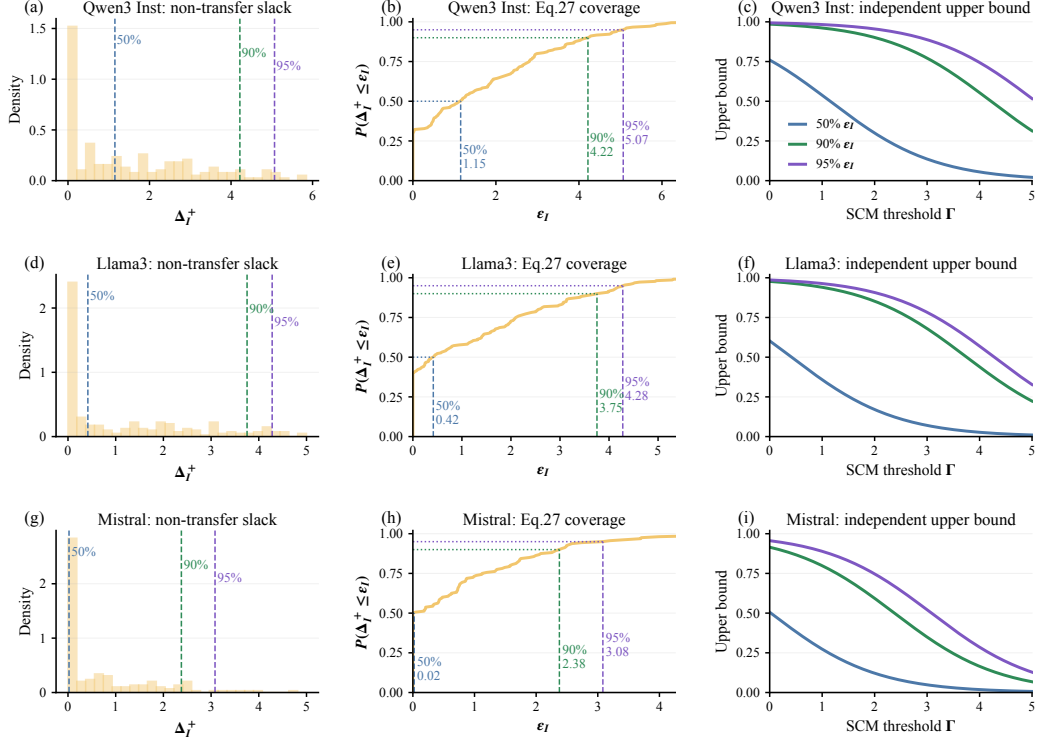


Fig. 15. Independent-model non-transfer on the additional source families. The panels report the slack distribution, empirical coverage $\Pr(\Delta_I^+ \leq \epsilon_I)$, and the corresponding upper-bound trend $\sigma(-\Gamma + \epsilon_I)$.

Takeaway

Independent models preserve the source-induced target behavior less consistently than source-family derivatives.

4) *SCM-Induced Separability*: Figure 16 closes the evidence chain. Across all three additional families, higher-SCM fingerprints generally produce a larger empirical target-answer gap between derivatives and independent models, while the budget-matched bounds show the same direction. The result supports the feasible-margin interpretation in the main text: SCM is a useful selection signal, but indefinite over-optimization is not required.

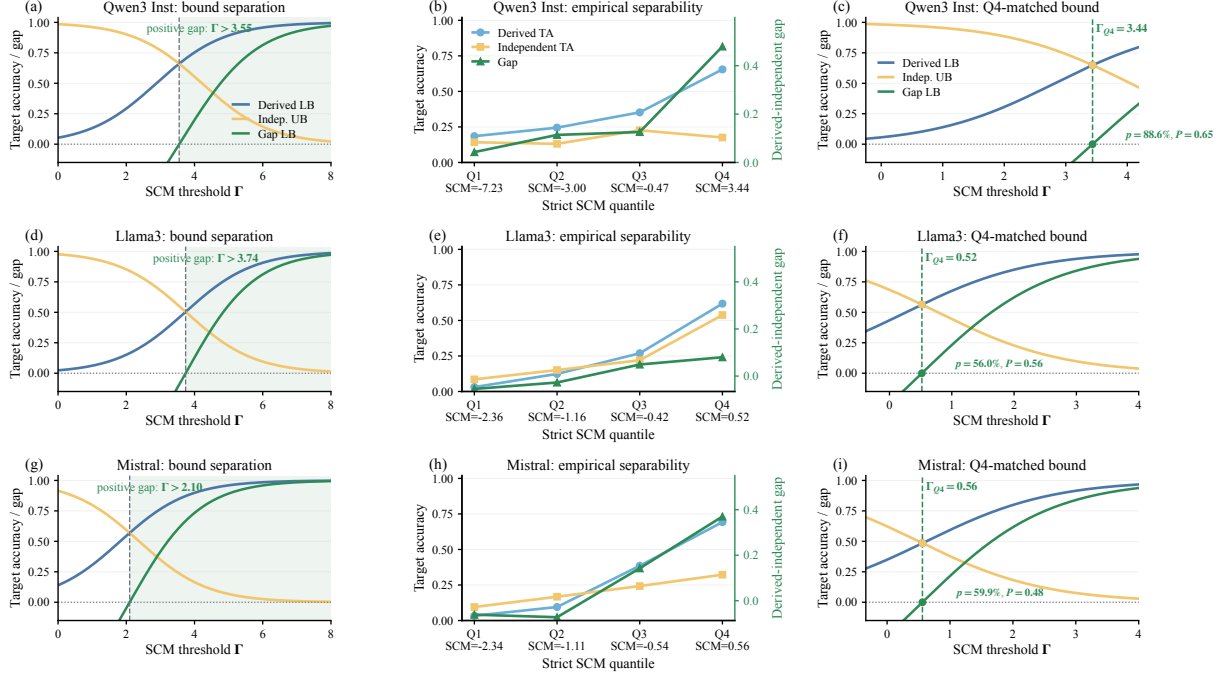


Fig. 16. SCM-induced separability on the additional source families. The columns compare conservative bound separation, empirical target-accuracy trends across SCM buckets, and budget-matched bound trends.

Takeaway

The feasible-margin-band pattern repeats across the additional source families.

B. Theory Details and Proofs

This appendix provides proof details and concise explanations for the theoretical analysis in Section IV.

1) *Finite Constrained-Answer Channel and SCM Feasibility*: The probability in Eq. 4 is over the model’s external decoding randomness, server-side nondeterminism, and any randomness in the verifier’s repeated query protocol. For a deterministic service, this distribution degenerates to a point mass; for source-side offline analysis, it can be estimated either by repeated generation under the chosen decoding configuration or by a calibrated valid-label probability surrogate that is checked after optimization. Invalid responses are not assigned to any valid label. Thus, $\sum_{a \in \mathcal{A}} q_M(a | p)$ may be smaller than one. This is intentional: invalid responses are verification failures in our hard-label setting.

Eq. 5 is well-defined whenever $q_M(t | p) \in (0, 1)$. In implementation, we clip probabilities to $[\epsilon_{\text{clip}}, 1 - \epsilon_{\text{clip}}]$ before taking log-odds. The proofs apply exactly to the unclipped channel when $q_M(t | p) \in (0, 1)$, and to the clipped channel otherwise; clipping only prevents infinite log-odds and does not change the hard-label verification statistic.

2) *SCM Scale and Feasibility*: The threshold has a direct probability meaning: $\Gamma_{\min} = 2$ corresponds to source perturbed target probability at least 0.881 and clean target probability at most 0.119; $\Gamma_{\min} = 3$ corresponds to approximately 0.953 and 0.047. The threshold is not a universal constant. For a fixed candidate pool $\mathcal{C}$, define

$$\mathcal{C}_{\Gamma_{\min}} = \{(x, u, t) \in \mathcal{C} : \Gamma(x, u, t) \geq \Gamma_{\min}\}. \quad (53)$$

If $\Gamma_{\min}^{(2)} > \Gamma_{\min}^{(1)}$, then $\mathcal{C}_{\Gamma_{\min}^{(2)}} \subseteq \mathcal{C}_{\Gamma_{\min}^{(1)}}$. Increasing $\Gamma_{\min}$ therefore strengthens individual fingerprints but can reduce the number and diversity of retained fingerprints. In the experiments, $\Gamma_{\min}$ should be chosen or swept using only source-side feasibility and, when available, a separate validation pool; empirical SCM-quantile analyses are evidence for trends, not a substitute for reporting the retained SCM distribution and pass rate at the operating threshold.

3) *Source-Side Surrogate Used During Perturbation Search*: Eq. 14 is the ideal loss on the same parser-induced constrained-answer distribution used by verification. In practice, the perturbation is optimized through a differentiable source-side surrogate, e.g., the next-option-token probability after the answer prefix. Let $\tilde{q}_{M_0}(t | p)$ be this surrogate

probability and let $\tilde{L}_{M_0}(p, t) = \log \frac{\tilde{q}_{M_0}(t|p)}{1 - \tilde{q}_{M_0}(t|p)}$. If the surrogate log-odds is calibrated to the parser distribution on retained candidates in the sense that

$$\left| L_{M_0}(p^u(x), t) - \tilde{L}_{M_0}(p^u(x), t) \right| \leq \delta_{\text{surr}}, \quad (54)$$

then the same argument as Lemma 1 applies with a safety margin: requiring the surrogate perturbed margin to exceed $\Gamma_{\min} + \delta_{\text{surr}}$ suffices to guarantee parser-side perturbed margin at least $\Gamma_{\min}$. Our offline construction therefore uses the optimizer as a search procedure but accepts fingerprints only after recomputing the SCM used in Eq. 8; the theoretical statements are about this accepted parser-side constrained-answer distribution, not about the surrogate alone.

4) *Proof of Lemma 1:* The clean side of the SCM is already guaranteed by Eq. 15. It remains to prove that the perturbed side is also at least $\Gamma_{\min}$.

From Eq. 14, the loss condition in Eq. 16 gives

$$-\log q_{M_0}(t | p^u(x)) \leq \log(1 + e^{-\Gamma_{\min}}). \quad (55)$$

Multiplying Eq. 55 by -1 reverses the inequality:

$$\log q_{M_0}(t | p^u(x)) \geq -\log(1 + e^{-\Gamma_{\min}}). \quad (56)$$

Applying the exponential function to Eq. 56 yields

$$q_{M_0}(t | p^u(x)) \geq \frac{1}{1 + e^{-\Gamma_{\min}}} = \sigma(\Gamma_{\min}). \quad (57)$$

Using Eq. 10, Eq. 57 is equivalent to

$$L_{M_0}(p^u(x), t) \geq \Gamma_{\min}. \quad (58)$$

Now Eq. 15 and Eq. 58 show that both terms in the minimum of Eq. 8 are at least $\Gamma_{\min}$. Therefore, Eq. 17 follows.

5) *Empirical Verification of SCM Control:* To verify Lemma 1, we record the source target loss in Eq. 14 and mean SCM over fingerprints while optimizing the source-side perturbation on the source model M_0 on Qwen3-1.7B Base [34]. As shown in Fig. 1(a), the target loss on source model M_0 decreases along the optimization progress and approaches the loss thresholds induced by different SCM targets, while Fig. 1(b) shows that the mean SCM increases along the same optimization trajectory, indicating that reducing the target loss strengthens the source-side counterfactual transition. Fig. 1(c) directly shows the correlation coefficient ρ between target loss and SCM is close to -1.00 , demonstrating a strong negative correlation: as the optimizer lowers the source target loss, SCM increases. This is consistent with Lemma 1, because reducing $-\log q_{M_0}(t | p^u)$ increases the perturbed target log-odds while the clean-side term is fixed by target selection.

Takeaway

The surrogate loss guides optimization, while parser-side SCM remains the final acceptance criterion.

6) *Interpretation of Local Behavioral Closeness:* By construction, the source model makes the target unlikely on the clean prompt and likely on the perturbed prompt. Assumption IV.2 says that a derived model should remain close to the protected model on these clean and perturbed fingerprint prompts for most retained fingerprints. This premise is probabilistic: it only needs to hold for a large fraction of fingerprints, not for every single sample.

The assumption motivates the transfer model, but it is not by itself the formal independent-model upper bound. Lemmas 2 and 3 make the required derived-preservation and independent generic-transfer budgets explicit. Accordingly, all formal separation statements in Theorem IV.3 and Lemma 4 are conditional on valid ϵ_D and ϵ_I budgets; the empirical diagnostics test how realistic those budgets are for the evaluated model pools. The verifier then turns the resulting per-fingerprint edge into evidence by aggregating many diverse fingerprints.

7) *Derived-Model Preservation Details:* The derived-model diagnostic measures how much the perturbed target preference weakens when moving from the source model to its derivatives. Smaller weakening budgets indicate stronger preservation. In the paper appendix we keep this reading and the proof below. The cross-model ECDF evidence is reported in Appendix A-A.

8) *Proof of Lemma 2:* By the definition of SCM, $L_{M_0}(p^u(x), t) \geq \Gamma$. Combining this with Eq. 22 gives

$$L_{M_D}(p^u(x), t) \geq \Gamma - \epsilon_D(M_D, x, u, t). \quad (59)$$

Applying the probability–logit relation in Eq. 10 and the monotonicity of $\sigma(\cdot)$ proves Eq. 23. The stated $\Gamma_{\min}$ version follows because $\Gamma \geq \Gamma_{\min}$.

9) *Independent-Model Non-Transfer Details:* The independent-model side of the analysis states that a prompt optimized on M_0 should not consistently preserve the same target preference on unrelated models. The diagnostic in the main text measures this effect through the independent-model slack: smaller slack means weaker accidental transfer, while larger slack gives a looser non-transfer guarantee. The corresponding cross-model diagnostics are reported in Appendix A-A.

10) *Proof of Lemma 3:* By the definition of SCM,

$$-L_{M_0}(p^0(x), t) \geq \Gamma, \quad L_{M_0}(p^0(x), t) \leq -\Gamma. \quad (60)$$

Combining Eq. 60 with Eq. 28 gives

$$L_{M_I}(p^u(x), t) \leq -\Gamma + \epsilon_I(M_I, x, u, t). \quad (61)$$

Applying Eq. 10 and the monotonicity of $\sigma(\cdot)$ proves Eq. 29. The stated $\Gamma_{\min}$ version follows from $\Gamma \geq \Gamma_{\min}$.

11) *Separability Details and Feasible Margin Band:* For fixed transfer budgets, increasing SCM improves the derived-independent separation until the gain starts to saturate. In practice, pushing the perturbation too far can also increase generic transfer to independent models. Thus, the useful operating region is a feasible margin band: SCM should be large enough to survive derived-model drift, but the construction should not optimize the source margin indefinitely. The same trend is tested across three additional source families in Appendix A-A.

12) *Proof of Theorem IV.3:* For each fingerprint i , Lemma 2 gives

$$q_{M_D}(t_i | p^{u_i}(x_i)) \geq \sigma(\Gamma_i - \epsilon_{D,i}). \quad (62)$$

Averaging over $i = 1, \dots, n$ proves Eq. 34. Similarly, Lemma 3 gives

$$q_{M_I}(t_i | p^{u_i}(x_i)) \leq \sigma(-\Gamma_i + \epsilon_{I,i}), \quad (63)$$

and averaging proves Eq. 35. Subtracting the upper bound from the lower bound gives Eq. 36.

13) *Proof of Lemma 4:* The expectation bounds follow from Theorem IV.3 and the monotonicity of $\sigma(\cdot)$. The fixed-budget gap is positive exactly when $\sigma(\Gamma_{\min} - \epsilon_D) > \sigma(-\Gamma_{\min} + \epsilon_I)$, which is equivalent, by monotonicity of the sigmoid, to $\Gamma_{\min} - \epsilon_D > -\Gamma_{\min} + \epsilon_I$. Rearranging gives Eq. 38.

No additional η correction is introduced in this lemma: the theorem's expected target-accuracy bounds are stated directly in terms of the explicit per-fingerprint budgets. Under the fixed-budget premises,

$$\mathbb{E}[\text{TA}(M_D)] \geq \sigma(\Gamma_{\min} - \epsilon_D), \quad \mathbb{E}[\text{TA}(M_I)] \leq \sigma(-\Gamma_{\min} + \epsilon_I), \quad (64)$$

and their difference is positive whenever $2\Gamma_{\min} > \epsilon_D + \epsilon_I$, exactly as stated in the main text. Aggregation then has a clear statistical role: once the two class-specific expectations are separated, averaging diverse fingerprint indicators reduces sampling variability, subject to the dependence qualification in Appendix A-B15.

14) *Empirical SCM-Induced Separability Validation:* Fig. 4 provides the empirical separability check for the fixed-budget gap analysis. The main observation is that higher-SCM fingerprints generally increase the derived-independent target-accuracy gap, while the strongest margins should not be over-optimized because generic transfer can also increase.

Takeaway

Useful separability emerges inside a feasible SCM region and can weaken under over-optimization.

15) *Proof of Lemma 5:* For the derived model, $\mu_{M_D} - \zeta^* \geq (\alpha_D - \alpha_I)/2$. Therefore the event $\text{TA}(M_D) < \zeta^*$ implies a lower-tail deviation of at least $(\alpha_D - \alpha_I)/2$ from its mean. Hoeffding's inequality for an average of n independent Bernoulli variables gives Eq. 42. The independent-model case is the symmetric upper-tail deviation because $\zeta^* - \mu_{M_I} \geq (\alpha_D - \alpha_I)/2$.

The independence condition is a statistical idealization. Fingerprints constructed from the same source model, prompt template, and dataset can be correlated. For deployment-grade audits, the concentration term should therefore be interpreted with an effective sample size after diversity filtering, or estimated by block/bootstrap resampling over question categories and target labels. The theoretical message is that reliability improves with both the expected gap $\alpha_D - \alpha_I$ and the effective number of diverse fingerprints.

16) *Margin-Sampling Robustness*: Both the verifier and the construction filter read a *hard* label, so the final quantity that matters is whether the sampled answer equals t . The target margin of Eq. 18 controls this directly under an idealized constrained decoder. Under temperature- T softmax sampling normalized over only the K legal labels in $\mathcal{A}$ (or conditional on the response being a valid legal label), each competing label contributes at most $e^{-m_M(p,t)/T}$ relative mass. Therefore,

$$\Pr[\text{sample} = t \mid p] \geq \frac{1}{1 + (K - 1) e^{-m_M(p,t)/T}}. \quad (65)$$

Thus a large source margin $m_{M_0}(p^u, t)$ strengthens target-label stability under the constrained-label idealization. This derivation does not assert that an arbitrary raw-text API samples only from $\mathcal{A}$; invalid or free-form generations remain possible in the actual threat model and are counted as failures. The deployment-facing stability claim is therefore checked empirically by the repeated-query source filter in Section V. The margin bound should be read as an explanatory connection between source-side margin and legal-label sampling, not as a replacement for that empirical filter.

C. Implementation Details

Algorithm 1 states the complete source-only construction procedure in executable form. It follows the main-text checkpoint rule: stop at the first source-side checkpoint satisfying Eq. 49, then recompute parser-side SCM and apply the source stability filter. No derived-model or independent-model statistic is used during construction.

Algorithm 1 SCM-guided GCG fingerprint construction

Require: Source model M_0 ; question pool $\mathcal{D}$; legal answer space $\mathcal{A}$; desired registry size n ; SCM threshold $\Gamma_{\min}$; prefix length ℓ_u ; GCG budget $T_{\max}$; stability threshold $s_{\min}$.

Ensure: Frozen fingerprint registry $\mathcal{F}$.

```

1:  $\mathcal{C} \leftarrow \emptyset$  ▷ qualified source-only candidates
2: for each question  $x \in \mathcal{D}$  do
3:   Render  $p^0(x)$ ; estimate  $q_{M_0}(a \mid p^0(x))$  over  $a \in \mathcal{A}$ .
4:    $y \leftarrow \arg \max_{a \in \mathcal{A}} q_{M_0}(a \mid p^0(x))$ .
5:    $\mathcal{T}(x) \leftarrow \{a \in \mathcal{A} \setminus \{y\} : -L_{M_0}(p^0(x), a) \geq \Gamma_{\min}\}$ .
6:   if  $\mathcal{T}(x) = \emptyset$  then
7:     continue
8:   end if
9:    $t \leftarrow \arg \min_{a \in \mathcal{T}(x)} q_{M_0}(a \mid p^0(x))$ .
10:  Initialize a length- $\ell_u$  perturbation  $u$ .
11:  for  $j = 1$  to  $T_{\max}$  do
12:    Apply one GCG update to reduce  $\mathcal{L}_{\text{src}}(u; x, t)$  in Eq. 48.
13:    if  $\mathcal{L}_{\text{src}}(u; x, t) \leq \log(1 + e^{-\Gamma_{\min}})$  then
14:      break ▷ first source-side margin-qualified checkpoint
15:    end if
16:  end for
17:  Recompute parser-side  $\Gamma(x, u, t)$  using Eq. 8.
18:  if  $\Gamma(x, u, t) < \Gamma_{\min}$  then
19:    continue
20:  end if
21:  Estimate  $\hat{s}_0(x, u, t)$  using Eq. 50.
22:  if  $\hat{s}_0(x, u, t) < s_{\min}$  then
23:    continue
24:  end if
25:  Add  $(p^u(x), y, t, u, \Gamma(x, u, t))$  and construction metadata to  $\mathcal{C}$ .
26: end for
27: Apply the source-only diversity filter over question IDs, categories, clean labels, and target labels.
28: Select up to  $n$  qualified candidates to form the frozen registry  $\mathcal{F}$ .
29: return  $\mathcal{F}$ 

```

D. Experimental Details

1) *Main Method Setup*: The main construction uses a fixed 20-query registry and a single source-only generation configuration. Tables VIII and IX summarize the query composition and generation parameters before the prompt/interface details.

TABLE VIII
COMPOSITION OF THE 20-QUERY FINGERPRINT SET FOR EACH PROTECTED SOURCE MODEL.

Category	Samples	Role
Chemistry_val	5	Science/entity-level multiple-choice questions
Engineering_dev	5	Technical engineering and control questions
Health_dev	5	Medical and health-domain questions
Physics_val	5	Mathematical and physical-law questions

TABLE IX
MAIN TCF FINGERPRINT-GENERATION PARAMETERS.

Parameter	Setting	Parameter	Setting
Dataset	MMLU	Fingerprints	20/source
Categories	4×5 samples	GCG steps	200
Batch size	128	Top- k	64
Prefix length	20 tokens	Placement	question prefix
Interface	final generated A/B/C/D label only		

2) *Prompt and Interface Templates*: For reproducibility, we record both the clean source query and the fingerprint query. The only difference is the source-optimized text inserted before or across the question body; the finite answer contract and parser remain unchanged.

Clean/source query template

```
The following are multiple choice questions (with answers) about {category}.
{question}
A. {option A}
B. {option B}
C. {option C}
D. {option D}
Answer with only one capital letter: A, B, C, or D.
Answer:
```

Fingerprint query template

```
{optimized prefix or distributed carrier fragments}
The following are multiple choice questions (with answers) about {category}.
{question with the retained insertion locations}
A. {option A}
B. {option B}
C. {option C}
D. {option D}
Answer with only one capital letter: A, B, C, or D.
Answer:
```

Parser and scoring contract

A deterministic parser maps the final generated response to one label in $\{A, B, C, D\}$ or to $\perp$ when the output is invalid or ambiguous. The online verifier never repairs $\perp$ with semantic similarity or model confidence. Fingerprint i is a hit only when the parsed label equals its private recorded target t_i , and the final target accuracy is the mean of these binary hits.

SCM-guided construction protocol

- 1) Render the clean constrained-answer prompt and estimate the protected model’s valid-label distribution.
- 2) Record the clean answer y and choose a different target t that satisfies the clean-side SCM requirement.
- 3) Initialize a 20-token optimizable text block and place it at the designated prompt locations.
- 4) Run source-side GCG with the reported 200-step budget, candidate batch size 128, and top- $k = 64$ replacements.
- 5) Select the earliest validated checkpoint that reaches the source-side margin target rather than blindly keeping the last optimization step.
- 6) Recompute the accepted prompt’s parser-side SCM; reject candidates that fail the margin or validity requirements.

- 7) Check source-side response stability and retain only qualified, stable fingerprints until the 20-query registry is complete.
- 8) Store the final prompt, clean answer, target, optimized text, and SCM metadata. Online verification then queries only the suspect API and aggregates final-label target hits.

3) *Baseline Setup:* Table X lists the baseline configurations and score definitions used for comparison.

TABLE X
BASELINE CONFIGURATIONS AND SCORE DEFINITIONS. BASELINES ARE EVALUATED USING THEIR ORIGINAL METHOD DEFINITIONS.

Method	Main setting	Score definition
TCF	20 source-matched fingerprints	mean target-label hit rate
TRAP	strict $n = 20$, 1500 steps, batch 512, top- $k = 256$	original TRAP retrieval score
ProFLingo	official $E = 256$, $p = 32$, $b = 8$, $n = 50$ AE list	final AE-list ASR
ZeroPrint	$n_{\text{samples}} = 2$, $n_{\text{aug}} = 4$, repeats 20	source-row similarity/correlation

4) *Naturalness-Oriented Variants:* Raw GCG prefixes are effective for source-side target induction, but their contiguous optimized strings may contain unnatural token fragments. Such artifacts can be exposed by full-input PPL or local window-PPL abnormality detectors. We therefore keep the same source-only construction and the same hard-label verifier, while changing only how the optimized fingerprint text is presented in the prompt.

NaturalCarrier (NC). NC reuses a prior-optimized GCG prefix and embeds its fragments in ordinary carrier phrases, such as `note this clue:`, `use this short hint:`, and `keep this relation in mind:`. This prevents the optimized text from appearing as one contiguous abnormal prefix while preserving the closed A/B/C/D answer contract used by the verifier.

Split placement. TCF-S4 and TCF-S8 apply the same NaturalCarrier idea, but distribute the optimizable token positions over 4 or 8 randomly selected valid insertion sites in the original question prompt, respectively. By shortening each inserted segment and spreading it across natural prompt locations, these variants are intended to reduce local PPL spikes under window-based input-abnormality checks. For NaturalCarrier and both split variants, verification is unchanged: the online parser reads only the final generated A/B/C/D label and counts success when it matches the recorded target answer.

5) *Representative Prompt-Naturalness Example:* The following retained example shows how the same multiple-choice item is presented under the clean interface, the contiguous Raw-GCG construction, and the distributed NaturalCarrier variants. Long optimized strings are truncated only for page readability; the evaluation uses the complete stored prompts.

Prompt P9: clean query

The following are multiple choice questions (with answers) about Chemistry_val.
Which type of mineral exhibits variations in color?
A. Pseudochromatic
B. Allochromatic
C. Idiochromatic
D. Iridescence
Answer with only one capital letter: A, B, C, or D.
Answer:

Prompt P9: Raw GCG

The following are multiple choice questions (with answers) about Chemistry_val.
)__interp Interior Haskell //~ Sv Soil
MonterwebElementXpaths ESLonyms Sampler Member Name
TypeSpecifier !IDoiotic Determine Which type of mineral exhibits variations in color?
A. Pseudochromatic
B. Allochromatic
C. Idiochromatic
D. Iridescence
Answer with only one capital letter: A, B, C, or D.
Answer: [remaining optimized text omitted from display]

Prompt P9: NaturalCarrier split4 (NC-S4)

Which type of mineral exhibits variations in color?
use this short hint: imgumber
note this clue: Before-control
A. Pseudochromatic
keep this relation in mind: als under
B. Allochromatic
C. Idiochromatic
D. Iridescence
focus on this cue: A among pur
Answer with only one capital letter: A, B, C, or D.
Answer:

Prompt P9: NaturalCarrier split8 (NC-S8)

Which type of mineral exhibits variations in color?
note this clue: pur
read this note carefully: img
keep this relation in mind: A
use this short hint: Before-control
keep this relation in mind: among
keep this relation in mind: under
A. Pseudochromatic
focus on this cue: umber
B. Allochromatic
consider this context: als
C. Idiochromatic
D. Iridescence
Answer with only one capital letter: A, B, C, or D.
Answer:

6) *Retained Evaluation Artifacts*: Table II defines the fixed model counts used for the main AUC calculation. Table XI below provides an extended checkpoint catalog, and the per-suspect matrices report the available row-level diagnostics. Some baseline artifacts are unavailable for particular checkpoints, so these appendix matrices use only artifact-available rows and do not redefine the main model pool or the AUC values in Table III.

TABLE XI
EXTENDED HUGGING FACE CHECKPOINT CATALOG ACROSS THE RETAINED EVALUATION ARTIFACTS. ROWS ARE GROUPED BY PROTECTED SOURCE AND DERIVATIVE CATEGORY; THE FIXED MODEL-POOL SIZES USED FOR THE PRIMARY AUC REMAIN THOSE IN TABLE II.

Source pool	Category	Hugging Face checkpoints
Qwen3-1.7B Instruct	Source	Qwen/Qwen3-1.7B
	SFT	activeDap/Qwen3-1.7B_tldr; contextboxai/Qwen3-1.7B-FC; prithivMLmods/Demeter-LongCoT-Qwen3-1.7B; UnfilteredAI/DAN-Qwen3-1.7B
	Pref./RL	wzx111/Qwen3-1.7B-Open-RL-GRPO-Baseline
	Domain	prithivMLmods/Panacea-MegaScience-Qwen3-1.7B; swapnillo/Bangla-AI-1.7B; vimalgupta/qwen3-1.7b-telecom-vocab-v2
	Pruning/Merging	MilyaShams/Qwen3-1.7B-Wanda_1_4
Qwen3-1.7B Base	Source	Qwen/Qwen3-1.7B-Base
	SFT	ali-elganzory/Qwen3-1.7B-Base-SFT-Tulu3-decontaminated; ffang2025/Affine-fang-v2; jekunz/Qwen3-1.7B-Base-is-SmolTalk; lilyx/Qwen3-1.7B-SFT
	Pref./RL	Kazuki1450/Qwen3-1.7B-Base_dsum_3_6_fnr_no_bracket_0p0_0p0_1p0_grpo_42_rule; miulab/Qwen3-1.7B-Usefulness; raca-workspace-v1/grpo-tool-sat-sft-qwen3-1p7b-sft-20260419-075623-96e9; zsqzz/Qwen3-1.7B_opsd_masked_grpo_dapo_hf
	Domain	indinode/Qwen3-1.7B; NiuTrans/LMT-60-1.7B-Base; Polygl10t/Tucano2-qwen-1.5B-Base; VLS2025-LegalSML/qwen3-1.7b-legal-pretrain
	Pruning/Merging	Qwen3-1.7B-Base-LightMagnitudePruned-s005
Llama3-8B Base Direct	Source	meta-llama/Meta-Llama-3-8B
	SFT	allenai/llama-3-tulu-2-8b; dphn/dolphin-2.9.1-llama-3-8b; Groq/Llama-3-Groq-8B-Tool-Use; Magpie-Align/Llama-3-8B-Magpie-Align-SFT-v0.3; openchat/openchat-3.6-8b-20240522
	Pref./RL	dfurman/Llama-3-8B-Orpo-v0.1
	Domain	aaditya/Llama3-OpenBioLLM-8B; DeepMount00/Llama-3-8b-Ita; hf1/llama-3-chinese-8b; winninghealth/WiNGPT2-Llama-3-8B-Chat
	Pruning/Merging	RedHatAI/SparseLlama-3-8B-pruned_50.2of4; mlabonne/Llama-3-SLERP-8B; Weyaxi/Einstein-v6.1-Llama3-8B
Mistral-7B-v0.3	Source	mistralai/Mistral-7B-v0.3
	SFT	amdevraj/mistral-7b-ift; cypienai/cymist-2-v03-SFT; migtissera/Tess-3-7B-SFT
	Pref./RL	lamm-mit/mistral-7B-v0.3-Base-CPT-SFT-DPO-09022024; llmat/Mistral-v0.3-7B-ORPO
	Domain	BEE-spoke-data/Mistral-7B-v0.3-stepbasin-books-20k; entfane/math-virtuoso-7B; klcsp/mistral7b-fft-coding-11-v1; openfoodfacts/spellcheck-mistral-7b; pszemraj/Mistral-7B-v0.3-sarcasm-rolls-4k; pucpr-br/Clinical-BR-Mistral-7B-v0.2; silversword/US-Immigration-Law-Mistral-7B-v0.3-CPT
	Pruning/Merging	ai-and-society/mistral-7B-Instruct-v0.3-wanda-wanda-unstruct-50; IntelLabs/sqft-mistral-7b-v0.3-50-base; mlfoundations-dev/hp_ablations_grid_mistral_base-mistralv0.3; mlfoundations-dev/mistral_7b_0-3_oh-dcft-v3.1-llama-3.1-405b
All evaluated sources	Independent negatives	01-ai/Yi-6B; EleutherAI/lemma_7b; Qwen/Qwen2.5-7B; THUDM/chatglm3-6b-base; allenai/tulu-2-dpo-7b; lmsys/vicuna-13b-v1.3; lmsys/vicuna-7b-v1.3; lmsys/vicuna-7b-v1.5; meta-llama/Llama-2-13b-chat-hf; meta-llama/Llama-2-7b-chat-hf; meta-llama/Llama-2-7b-hf; meta-llama/Meta-Llama-3-8B; microsoft/Phi-3-small-8k-instruct; mistralai/Mistral-7B-v0.3; tiuae/falcon-7b

E. Full Per-Suspect Verification Results

In addition to the family-level aggregates, we report the available model-wise verification matrices for inspection. The fixed model counts and family-level AUCs are defined in Tables II and III. Because some baseline artifacts are unavailable for particular checkpoints, the row-wise AUCs below are computed against the artifact-available independent rows for each table and should be interpreted as diagnostics rather than substitutes for the main family-level AUC.

How to read the model-wise tables

Rows shaded light green are the protected source model or source-family derivatives, and rows shaded light blue are artifact-available independent checkpoints. Each method reports its native fingerprint score and a row-wise diagnostic AUC against the available independent rows in that table. These row-wise diagnostics expose heterogeneity; they do not redefine the main evaluation pool or Table III. A dash means that a row-level AUC is not defined for an independent row or that the baseline artifact is unavailable under the official pipeline.

TABLE XII
ROWS WHERE TCF IS NOT THE HIGHEST ROW-WISE AUC. THESE ARE THE ONLY EXCEPTIONS AMONG THE SOURCE/FAMILY-POSITIVE ROWS IN THE PER-SUSPECT MATRICES BELOW.

Source family	Derivative	TestAcc	TCF	TRAP	ProF	ZPrint	Interpretation
Llama3-8B Base Direct	DeepMount00/Llama-3-8b-Ita	0.660	0.444	0.333	0.889	0.750	Italian/domain SFT increases overlap with independent hard-label scores.
Llama3-8B Base Direct	RedHatAI/SparseLlama-3-8B-pruned-50.2of4	0.650	0.778	1.000	0.944	0.312	Structured pruning weakens some target-answer preferences while retaining task utility.

Takeaway

The model-wise results show that TCF’s family-level advantage is distributed across heterogeneous derivatives rather than being driven by a small number of source-like checkpoints.

TABLE XIII

PER-SUSPECT COMPARISON USING **QWEN3-1.7B INSTRUCT** AS THE PROTECTED SOURCE MODEL. FAMILY-DERIVED SUSPECTS ARE SHOWN IN THE UPPER BLOCK AND INDEPENDENT SUSPECTS IN THE LOWER BLOCK. SCORE DENOTES EACH METHOD’S NATIVE FINGERPRINT SCORE; AUC DENOTES ROW-WISE FAMILY-VERSUS-INDEPENDENT SEPARABILITY.

Category	Suspect Model (Hugging Face ID)	Truth	TCF		TRAP		ProFLingo		ZeroPrint	
			Score	AUC	Score	AUC	Score	AUC	Score	AUC
Source	Qwen/Qwen3-1.7B	Positive	1.000	1.000	0.920	1.000	0.200	1.000	0.560	1.000
SFT / task FT	activeDap/Qwen3-1.7B_tldr	Positive	0.550	1.000	0.000	0.500	0.220	1.000	0.650	1.000
SFT / task FT	contextboxai/Qwen3-1.7B-FC	Positive	1.000	1.000	0.610	1.000	0.180	1.000	0.890	1.000
SFT / task FT	prithivMLmods/Demeter-LongCoT-Qwen3-1.7B	Positive	0.900	1.000	0.110	1.000	0.240	1.000	0.780	1.000
SFT / task FT	UnfilteredAI/DAN-Qwen3-1.7B	Positive	0.400	1.000	0.010	1.000	0.120	0.857	0.570	1.000
Preference / RL	wzx111/Qwen3-1.7B-Open-R1-GRPO-Baseline	Positive	1.000	1.000	0.350	1.000	0.200	1.000	0.460	1.000
Language / Do-main	prithivMLmods/Panacea-MegaScience-Qwen3-1.7B	Positive	0.850	1.000	0.020	1.000	0.140	0.857	0.750	1.000
Language / Do-main	swapnillo/Bangla-AI-1.7B	Positive	0.700	1.000	0.030	1.000	0.200	1.000	0.510	1.000
Language / Do-main	vimalgupta/qwen3-1.7b-telecom-vocab-v2	Positive	0.350	1.000	0.010	1.000	0.100	0.857	0.560	1.000
Pruning	MilyaShams/Qwen3-1.7B-Wanda_1_4	Positive	0.400	1.000	0.010	1.000	0.100	0.857	0.580	1.000
Independent	01-ai/Yi-6B	Negative	0.250	–	0.000	–	0.040	–	0.100	–
Independent	EleutherAI/llemma_7b	Negative	0.200	–	0.000	–	0.040	–	0.160	–
Independent	Qwen/Qwen2.5-7B	Negative	0.000	–	0.000	–	0.020	–	0.060	–
Independent	meta-llama/Meta-Llama-3-8B	Negative	0.100	–	0.000	–	0.000	–	0.180	–
Independent	microsoft/Phi-3-small-8k-instruct	Negative	0.000	–	0.000	–	0.160	–	0.190	–
Independent	mistralai/Mistral-7B-v0.3	Negative	0.050	–	0.000	–	0.040	–	0.130	–
Independent	THUDM/chatglm3-6b-base	Negative	0.100	–	0.000	–	0.020	–	–	–

TABLE XIV
PER-SUSPECT COMPARISON USING QWEN3-1.7B BASE AS THE PROTECTED SOURCE MODEL.

Category	Suspect Model (Hugging Face ID)	Truth	TCF		TRAP		ProFLingo		ZeroPrint	
			Score	AUC	Score	AUC	Score	AUC	Score	AUC
Source	Qwen/Qwen3-1.7B-Base	Positive	0.900	1.000	1.000	1.000	0.480	1.000	0.620	1.000
SFT / task FT	ali-elganzory/Qwen3-1.7B-Base-SFT-Tulu3-decontaminated	Positive	0.900	1.000	0.550	1.000	0.420	1.000	0.490	1.000
SFT / task FT	ffang2025/Affine-fang-v2	Positive	0.400	1.000	0.100	1.000	0.160	1.000	0.250	0.750
SFT / task FT	jekunz/Qwen3-1.7B-Base-is-SmolTalk	Positive	0.800	1.000	0.450	1.000	0.400	1.000	0.620	1.000
SFT / task FT	llylx/Qwen3-1.7B-SFT	Positive	0.750	1.000	0.370	1.000	0.300	1.000	0.370	1.000
Preference / RL	Kazuki1450/Qwen3-1.7B-Base_dsum_3_6_fnr_no_bracket_0p0_0p0_1p0_grpo_42_rule	Positive	0.850	1.000	1.000	1.000	0.380	1.000	0.720	1.000
Preference / RL	miulab/Qwen3-1.7B-Usefulness	Positive	0.850	1.000	0.220	1.000	0.240	1.000	0.320	0.833
Preference / RL	raca-workspace-v1/grpo-tool-sat-sft-qwen3-1p7b-sft-20260419-075623-96e9	Positive	0.550	1.000	0.000	0.500	0.000	0.071	0.340	1.000
Preference / RL	zsqqz/Qwen3-1.7B_opsd_masked_grpo_dapo_hf	Positive	0.400	1.000	0.090	1.000	0.100	0.929	0.250	0.750
Language / Do-main	indicnode/Qwen3-1.7B-main	Positive	0.400	1.000	0.100	1.000	0.160	1.000	0.340	1.000
Language / Do-main	NiuTrans/LMT-60-1.7B-Base-main	Positive	0.550	1.000	0.000	0.500	0.160	1.000	0.350	1.000
Language / Do-main	Polyglot/Tucano2-qwen-1.5B-Base-main	Positive	0.400	1.000	0.000	0.500	0.120	1.000	0.290	0.833
Language / Do-main	VLSP2025-LegalSML/qwen3-1.7b-legal-pretrain-main	Positive	0.900	1.000	0.850	1.000	0.500	1.000	0.620	1.000
Pruning	Qwen3-1.7B-Base-LightMagnitudePruned-s005	Positive	1.000	1.000	1.000	1.000	0.400	1.000	0.840	1.000
Independent	01-ai/Yi-6B	Negative	0.350	–	0.000	–	0.060	–	0.160	–
Independent	EleutherAI/lemma_7b	Negative	0.300	–	0.000	–	0.060	–	0.190	–
Independent	Qwen/Qwen2.5-7B	Negative	0.300	–	0.000	–	0.000	–	0.150	–
Independent	meta-llama/Meta-Llama-3-8B	Negative	0.300	–	0.000	–	0.040	–	0.250	–
Independent	microsoft/Phi-3-small-8k-instruct	Negative	0.300	–	0.000	–	0.100	–	0.330	–
Independent	mistralai/Mistral-7B-v0.3	Negative	0.300	–	0.000	–	0.040	–	0.220	–
Independent	THUDM/chatglm3-6b-base	Negative	0.350	–	0.000	–	0.080	–	–	–

TABLE XV
PER-SUSPECT COMPARISON USING LLAMA3-8B BASE DIRECT AS THE PROTECTED SOURCE MODEL.

Category	Suspect Model (Hugging Face ID)	Truth	TCF		TRAP		ProFLingo		ZeroPrint	
			Score	AUC	Score	AUC	Score	AUC	Score	AUC
Source	meta-llama/Meta-Llama-3-8B	Positive	0.650	1.000	0.990	1.000	0.520	1.000	–	–
SFT / task FT	allenai/llama-3-tulu-2-8b	Positive	0.400	1.000	0.500	0.889	0.060	0.444	0.280	0.875
SFT / task FT	dphn/dolphin-2.9.1-llama-3-8b	Positive	0.450	1.000	1.000	1.000	0.140	0.889	0.140	0.375
SFT / task FT	Groq/Llama-3-Groq-8B-Tool-Use	Positive	0.550	1.000	0.090	0.333	0.140	0.889	0.180	0.562
SFT / task FT	Magpie-Align/Llama-3-8B-Magpie-Align-SFT-v0.3	Positive	0.450	1.000	0.640	1.000	0.220	1.000	0.460	1.000
SFT / task FT	openchat/openchat-3.6-8b-20240522	Positive	0.450	1.000	0.200	0.444	0.160	0.944	0.390	1.000
Preference / RL	dfurman/Llama-3-8B-Orpo-v0.1	Positive	0.650	1.000	0.720	1.000	0.340	1.000	0.430	1.000
Language / Do-main	aaditya/Llama3-OpenBioLLM-8B-main	Positive	0.400	1.000	0.540	0.889	0.180	1.000	0.240	0.750
Language / Do-main	DeepMount00/Llama-3-8b-Ita-main	Positive	0.300	0.444	0.100	0.333	0.140	0.889	0.240	0.750
Language / Do-main	hfl/llama-3-chinese-8b-main	Positive	0.550	1.000	0.700	1.000	0.260	1.000	0.300	0.875
Language / Do-main	winninghealth/WiNGPT2-Llama-3-8B-Chat-main	Positive	0.400	1.000	0.150	0.333	0.020	0.056	0.240	0.750
Pruning	RedHatAI/SparseLlama-3-8B-pruned_50.2of4	Positive	0.350	0.778	0.890	1.000	0.160	0.944	0.130	0.312
Merging	mlabonne/Llama-3-SLERP-8B	Positive	0.400	1.000	0.750	1.000	0.380	1.000	0.460	1.000
Merging	Weyaxi/Einstein-v6.1-Llama3-8B	Positive	0.550	1.000	0.430	0.889	0.120	0.833	0.290	0.875
Independent	01-ai/Yi-6B	Negative	0.150	–	0.330	–	0.120	–	0.180	–
Independent	EleutherAI/lemma_7b	Negative	0.350	–	0.200	–	0.060	–	0.170	–
Independent	Qwen/Qwen2.5-7B	Negative	0.200	–	0.250	–	0.060	–	0.130	–
Independent	allenai/tulu-2-dpo-7b	Negative	0.350	–	0.000	–	0.080	–	0.210	–
Independent	lmsys/vicuna-13b-v1.3	Negative	0.350	–	0.560	–	0.020	–	0.000	–
Independent	lmsys/vicuna-7b-v1.5	Negative	0.300	–	0.000	–	0.160	–	0.270	–
Independent	meta-llama/Llama-2-13b-chat-hf	Negative	0.250	–	0.000	–	0.040	–	0.020	–
Independent	mistralai/Mistral-7B-v0.3	Negative	0.350	–	0.200	–	0.060	–	0.330	–
Independent	THUDM/chatglm3-6b-base	Negative	0.300	–	0.250	–	0.060	–	–	–

TABLE XVI
PER-SUSPECT COMPARISON USING **MISTRAL-7B-v0.3** AS THE PROTECTED SOURCE MODEL.

Category	Suspect Model (Hugging Face ID)	Truth	TCF		TRAP		ProFLingo		ZeroPrint	
			Score	AUC	Score	AUC	Score	AUC	Score	AUC
Source	mistralai/Mistral-7B-v0.3	Positive	0.600	1.000	0.250	1.000	0.660	1.000	0.570	1.000
SFT / task FT	amdevraj/mistral-7b-ift	Positive	0.600	1.000	0.000	0.500	0.220	1.000	0.310	1.000
SFT / task FT	cypienai/cymist-2-v03-SFT	Positive	0.400	1.000	0.000	0.500	0.260	1.000	0.310	1.000
SFT / task FT	migtissera/Tess-3-7B-SFT	Positive	0.450	1.000	0.000	0.500	0.200	1.000	0.320	1.000
Preference / RL	lamm-mit/mistral-7B-v0.3-Base-CPT-SFT-DPO-09022024	Positive	0.300	1.000	0.000	0.500	0.160	1.000	0.320	1.000
Preference / RL	llmat/Mistral-v0.3-7B-ORPO	Positive	0.350	1.000	0.000	0.500	0.200	1.000	0.470	1.000
Language / Do-main	BEE-spoke-data/Mistral-7B-v0.3-stepbasin-books-20k	Positive	0.600	1.000	0.170	1.000	0.500	1.000	0.580	1.000
Language / Do-main	entfane/math-virtuoso-7B	Positive	0.450	1.000	0.000	0.500	0.180	1.000	0.270	0.800
Language / Do-main	klcsp/mistral7b-fft-coding-11-v1	Positive	0.350	1.000	0.000	0.500	0.000	0.000	0.170	0.200
Language / Do-main	openfoodfacts/spellcheck-mistral-7b	Positive	0.550	1.000	0.000	0.500	0.380	1.000	0.590	1.000
Language / Do-main	pszemraj/Mistral-7B-v0.3-sarcasm-scrolls-4k	Positive	0.450	1.000	0.000	0.500	0.120	1.000	0.390	1.000
Language / Do-main	pucpr-br/Clinical-BR-Mistral-7B-v0.2	Positive	0.250	1.000	0.000	0.500	0.060	0.250	0.350	1.000
Language / Do-main	silversword/US-Immigration-Law-Mistral-7B-v0.3-CPT	Positive	0.550	1.000	0.000	0.500	0.440	1.000	0.570	1.000
Pruning	ai-and-society/mistral-7B-Instruct-v0.3-wanda-wanda-unstruct-50	Positive	0.350	1.000	0.000	0.500	0.400	1.000	0.300	0.900
Pruning	IntelLabs/sqft-mistral-7b-v0.3-50-base	Positive	0.400	1.000	0.000	0.500	0.140	1.000	0.360	1.000
Merging	mlfoundations-dev/hp_ablations_grid_mistral_base-mistralv0.3	Positive	0.350	1.000	0.000	0.500	0.100	0.917	0.260	0.800
Merging	mlfoundations-dev/mistral_7b_0-3_oh-dcft-v3.1-llama-3.1-405b	Positive	0.450	1.000	0.000	0.500	0.400	1.000	0.270	0.800
Independent	01-ai/Yi-6B	Negative	0.150	–	0.000	–	0.060	–	0.250	–
Independent	EleutherAI/lemma_7b	Negative	0.200	–	0.000	–	0.080	–	0.230	–
Independent	Qwen/Qwen2.5-7B	Negative	0.200	–	0.000	–	0.060	–	0.120	–
Independent	meta-llama/Meta-Llama-3-8B	Negative	0.100	–	0.000	–	0.080	–	0.300	–
Independent	microsoft/Phi-3-small-8k-instruct	Negative	0.100	–	0.000	–	0.100	–	0.190	–
Independent	THUDM/chatglm3-6b-base	Negative	0.200	–	0.000	–	0.060	–	–	–

Compatibility note

ChatGLM is included as an independent negative for TCF, TRAP, and ProFLingo whenever a valid hard-label artifact is available. Its ZeroPrint entry remains “–” because the official/repository-aligned ZeroPrint pipeline encountered a third-party remote-code compatibility issue for ChatGLM in the evaluated software environment; this missing artifact is excluded only from the ZeroPrint AUC calculation.

F. Extended Construction and Verification Record

Artifact repository

Project available at:

<https://github.com/Underflow-1/TCF-LLM-Fingerprinting>

The main paper presents the minimal method definition needed to understand TCF. This section records the complete source-only construction path, the retained fingerprint fields, and the online hard-label decision rule so that every reported result can be reconstructed from explicit intermediate artifacts.

1) *Offline Construction Workflow:* Tables XVII and XVIII summarize the complete offline pipeline and the frozen registry fields passed to the online verifier.

TABLE XVII
END-TO-END SOURCE-ONLY FINGERPRINT CONSTRUCTION WORKFLOW.

Stage	Input	Operation and retained evidence
Question preparation	Clean multiple-choice item	Render the fixed answer-only interface and preserve the original question identifier, category, options, and gold answer.
Clean source query	Protected model M_0	Record the clean parsed answer and the valid-label distribution used only for offline target selection.
Target selection	Clean label distribution	Choose a non-clean target satisfying the clean-side SCM requirement and reject questions with no feasible target.
Perturbation search	Source model and target	Optimize the text perturbation with GCG and save the source loss, valid-label probabilities, and SCM trajectory at every checkpoint.
Early stopping / checkpoint retention	Source loss and SCM	Retain the first checkpoint satisfying the main-text source-side stopping condition, then recompute parser-side SCM and source stability.
Qualification	Retained checkpoint	Recompute parser-side SCM, source validity, response stability, and prompt-format checks; reject candidates that fail any condition.
Diversity filtering	Qualified candidates	Balance categories, clean labels, target labels, and question identities before forming the final registry.
Registry export	Final fingerprint set	Store the exact prompt, private target, clean answer, perturbation text, SCM metadata, and parser contract required by online verification.

TABLE XVIII
RECOMMENDED FINGERPRINT-REGISTRY FIELDS. PRIVATE FIELDS ARE NEVER SENT TO THE SUSPECT SERVICE.

Field	Visibility	Purpose
Fingerprint ID	Internal	Stable key linking construction, query, parser, and result artifacts.
Source model	Internal	Protected checkpoint from which the fingerprint was constructed.
Question ID / category	Internal	Supports category-balanced sampling and reproducibility.
Clean prompt	Internal	Records the pre-perturbation source behavior and clean answer.
Fingerprint prompt	Queried	Exact prompt sent to the suspect API.
Clean answer	Internal	Source label before the counterfactual perturbation.
Target answer	Private	Ownership target used by the hard-label verifier.
Optimized text	Private	Contiguous prefix or distributed NaturalCarrier fragments.
Checkpoint / SCM	Internal	Supports construction audits and early-stopping reconstruction.
Parser version	Internal	Pins the deterministic response-to-label contract.
Raw response	Audit log	Raw suspect response retained before parsing.
Parsed label / hit	Audit log	Final legal label and target-hit indicator used for aggregation.

2) Online Hard-Label Verification Workflow:

Online verification procedure

- 1) Load the frozen fingerprint registry and pre-registered decision threshold.
- 2) Send each stored fingerprint prompt to the suspect service using the declared decoding configuration.
- 3) Save the raw response before any parsing or normalization.
- 4) Apply the deterministic parser and return one label in $\{A, B, C, D\}$ or $\perp$ for invalid or ambiguous output.
- 5) Count a hit only when the parsed label equals the private target recorded for that fingerprint.
- 6) Average the binary hits to obtain target accuracy and compare it with the pre-registered threshold.
- 7) Export the prompt identifier, raw response, parsed label, and decision trace as the verification audit record.

Algorithm 2 Hard-label ownership verification

Require: Frozen fingerprint registry $\mathcal{F}$; suspect black-box service M_s ; deterministic parser Parse ; pre-registered threshold ζ .

Ensure: Ownership decision and audit trace.

```
1:  $S \leftarrow 0$ 
2: for each fingerprint record  $(p_i, t_i) \in \mathcal{F}$  do
3:   Query  $M_s$  with  $p_i$  and save raw response  $r_i$ .
4:    $\hat{a}_i \leftarrow \text{Parse}(r_i) \in \mathcal{A} \cup \{\perp\}$ .
5:   if  $\hat{a}_i = t_i$  then
6:      $S \leftarrow S + 1$ 
7:   end if
8:   Save  $(i, r_i, \hat{a}_i, \mathbb{I}[\hat{a}_i = t_i])$  to the audit trace.
9: end for
10:  $\text{TA}(M_s; \mathcal{F}) \leftarrow S/|\mathcal{F}|$ .
11: if  $\text{TA}(M_s; \mathcal{F}) \geq \zeta$  then
12:   return DERIVED, audit trace
13: else
14:   return INDEPENDENT, audit trace
15: end if
```

The online phase never accesses suspect logits, hidden states, token probabilities, gradients, embeddings, or semantic-similarity models. Invalid outputs remain failures rather than being repaired by a secondary model. This strict contract keeps the reported score auditable at the final-response interface.

Takeaway

Online verification reduces ownership evidence to deterministic final-label target hits under a frozen parser and threshold.

3) *Reproducibility Checklist*: Table XIX records the minimum configuration and audit artifacts needed to rebuild the reported evaluation without changing the verifier contract.

TABLE XIX
REPRODUCIBILITY CHECKLIST FOR REBUILDING A TCF EVALUATION.

Item	Required record
Model identity	Exact source, derivative, and independent checkpoint identifiers plus revisions when available.
Question pool	Dataset version, split, category list, question identifiers, and decontamination policy.
Prompt interface	Exact clean and fingerprint templates, answer-only instruction, insertion positions, and parser version.
Construction	Random seed, prefix length, optimization steps, candidate batch size, top- k , checkpoint schedule, and SCM acceptance rule.
Verification	Decoding parameters, query repetitions, invalid-output policy, target-hit aggregation, and threshold provenance.
Baselines	Original protocol, query budget, official score definition, implementation source, and any compatibility patch.
Statistics	Positive/negative taxonomy, AUC tie handling, subgroup definitions, and any pooled-score normalization.
Artifacts	Raw responses, parsed outputs, per-suspect scores, plot-data tables, and the script used to produce each figure.

G. Expanded Baseline Documentation

The baselines are evaluated under their own score definitions rather than being forced into TCF’s target-hit score. The following examples make the interface difference explicit and complement the compact configuration table in Appendix A-D3.

1) Representative TRAP Artifact:

TRAP retrieval artifact

```
Task: produce a short target string under the official TRAP retrieval protocol.
Optimized suffix: lista ( print-phaseko@(?/Topic/TabControlighborhood ... [truncated]
Target response: 6532
Generated response: 6532
```

Official score contribution: successful target retrieval.

TRAP verifies whether its optimized suffix retrieves the designated target output. Its score is therefore a retrieval statistic and is not converted into TCF’s multiple-choice target-hit rate.

2) Representative ProFLingo Artifact:

ProFLingo adversarial-example record

Question: Where does the sun rise?
Target keyword: north
Generated answer: The sun rises in the east.
Official score contribution: unsuccessful target-keyword retrieval.

ProFLingo evaluates a list of adversarial examples and reports the corresponding attack-success statistic. The original list length and official target-keyword rule are retained in our comparison.

3) Representative ZeroPrint Artifact:

ZeroPrint source-row similarity artifact

Source row: Qwen3-1.7B-Base-ZP
Source similarity: 1.000
Derivative similarities: stored continuous source-row values
Independent similarities: stored continuous source-row values
Official score contribution: source-row similarity/correlation.

ZeroPrint produces a continuous similarity profile rather than a prompt-to-target hard-label event. We therefore preserve its source-row score and compute AUC directly from that profile.

Takeaway

The comparison preserves each baseline’s original evidence channel and score definition.

H. Expanded Naturalness Documentation

1) *Naturalness Metrics*: No single metric fully characterizes prompt naturalness. We therefore separate local trigger abnormality, whole-prompt fluency, repetition, and embedding-space diagnostics.

TABLE XX
NATURALNESS AND PROMPT-QUALITY DIAGNOSTICS USED IN THE EXTENDED ANALYSIS.

Metric	Textual object	Interpretation
Full-input PPL	Complete rendered query	Measures global fluency after inserting all fingerprint text.
Window-PPL	Local windows around insertions	Detects short abnormal regions that can be hidden by a long fluent prompt.
Fragment PPL	Inserted prefix or fragment only	Isolates the lexical abnormality of the fingerprint text itself.
Abnormal-token ratio	Inserted text	Counts tokenizer fragments that violate the readable-token constraints.
Repetition score	Inserted text	Detects repeated words, substrings, or token cycles.
Semantic-drift proxy	Candidate versus readable seed	Measures how far a naturalized candidate moves from its seed instruction.
Prompt embedding	Full prompt or prompt residual	Provides a qualitative view of proximity to benign multiple-choice prompts.
Verification AUC	Suspect-model score	Ensures that improved fluency is not obtained by discarding ownership evidence.

2) *Naturalization Variant Taxonomy*: All variants use source-only construction. Derived and independent models are queried only after the candidate prompt has been frozen, preventing evaluation leakage into fingerprint generation.

TABLE XXI
NATURALNESS-ORIENTED CONSTRUCTION VARIANTS AND THEIR ISOLATED DESIGN QUESTIONS.

Variant	Construction rule	Question isolated by the variant
Raw GCG	Keep the optimized contiguous trigger.	Reference verification strength before naturalization.
SemBound-GCG	Restrict candidate semantic drift from a readable seed.	Can explicit semantic constraints preserve target induction?
SourceRerank-GCG	Retain source-effective candidates, then rerank by naturalness.	Does source-first selection avoid sacrificing verification strength?
PPLRerank-GCG	Add a prefix-fluency proxy to candidate selection.	Can local fluency be optimized directly during GCG search?
Position-Fragment-GCG	Split an effective trigger across prompt slots without carriers.	Is distribution alone sufficient to reduce local abnormality?
NaturalCarrier	Wrap distributed fragments with short cue-like phrases.	Can carriers improve readability while preserving the trigger information?
TCF-S4 / TCF-S8	Use four or eight distributed insertion groups.	How does split granularity affect naturalness and erasure robustness?
SlotPrefix-GF	Search only over natural slot templates.	Can fully natural gradient-free prompts replace token-level triggers?
EvoPrefix-GF	Mutate a population of natural instructions.	Is population-based natural-language search sufficient?

3) *Prompt-Embedding Diagnostics*: The embedding model is used only for visualization and never participates in fingerprint construction or verification. We report both seed stability and distance-to-reference views because a single t-SNE projection can be sensitive to initialization.

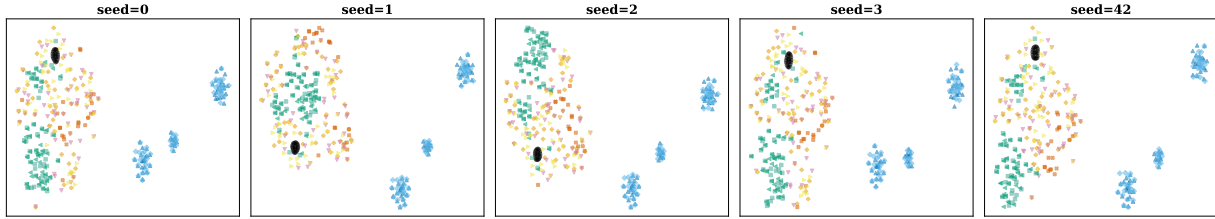


Fig. 17. Seed-stability diagnostic for the prompt-embedding visualization. Repeated projections test whether the qualitative arrangement is robust to visualization randomness.

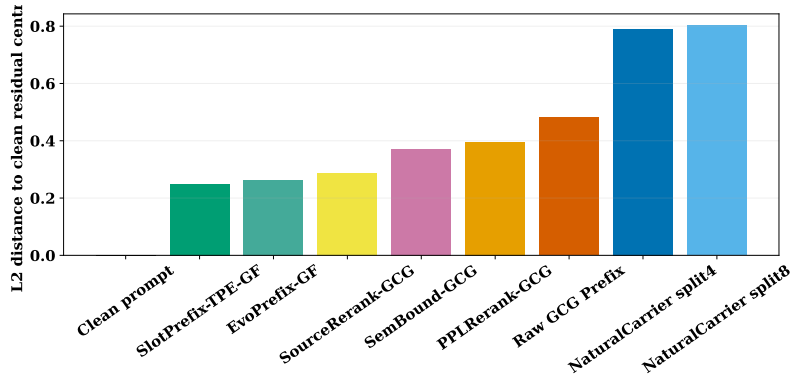


Fig. 18. Embedding-distance diagnostic for naturalized prompt variants. The external embedding model is used only for offline interpretation.

Takeaway

The embedding diagnostics are interpretive rather than monotonic: some natural-language search/reranking variants move closer to the clean-prompt residual centroid, whereas the split NaturalCarrier variants retain a larger structured embedding shift. Their naturalness advantage is therefore supported primarily by the PPL diagnostics in the main text, while verification remains strictly hard-label.

I. Additional Distillation Diagnostics

The main text reports the final clean-query extraction result. To show how the ownership signal develops during student training, we additionally expose a checkpoint sweep from a smaller clean-query distillation run. The student never observes the private fingerprint prompts.

TABLE XXII
CHECKPOINT SWEEP FOR CLEAN-QUERY DISTILLATION FROM MISTRAL-7B-v0.3 TO LLAMA-2-7B. CLEAN100 IS CLEAN MULTIPLE-CHOICE ACCURACY, T-AGREE IS TEACHER AGREEMENT, AND THE FINGERPRINT ROWS REPORT STUDENT-VERSUS-INDEPENDENT AUC.

Metric	S0	S1	S2	S3	S4	S5	S6	S7	S8	S9	S10
Clean100	0.66	0.65	0.69	0.75	0.73	0.77	0.73	0.71	0.71	0.72	0.72
T-Agree	0.68	0.64	0.67	0.73	0.73	0.75	0.73	0.73	0.71	0.72	0.72
TCF	0.31	0.56	0.75	0.56	0.75	0.88	0.81	0.88	0.88	0.88	0.88
TRAP	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50	0.50
ProFLingo	0.00	0.00	0.00	0.00	0.00	0.00	0.00	0.06	0.00	0.00	0.00
ZeroPrint	0.61	0.22	0.22	0.11	0.11	0.00	0.11	0.11	0.11	0.11	0.11

The trajectory shows that TCF evidence appears as the student absorbs more of the teacher’s clean behavior, even though the extraction data contains no private fingerprint prompts. This distinguishes inherited source behavior from direct memorization of the fingerprint registry.

Takeaway

The ownership signal strengthens as the extracted student learns the teacher’s clean behavior.

J. Artifact Map and Expected Outputs

Public reproduction versus deployment privacy

The registry released with this paper is a reproducibility artifact for reconstructing the reported experiments. It must not be reused as a secret registry in a real ownership audit. The threat model in Sec. III requires deployment fingerprints and their targets to remain unknown to the suspect before verification, so a deployment-grade audit should generate a fresh private registry and keep its targets secret until the audit is complete.

Minimum audit bundle. A third party should be able to reconstruct every reported point from four immutable records: the frozen fingerprint registry, the raw suspect responses, the deterministic parser outputs, and the positive/negative model taxonomy. Plotting scripts should consume these records without querying any model again.

TABLE XXIII
LOGICAL ARTIFACT MAP FOR REPRODUCING THE REPORTED EXPERIMENTS. DIRECTORY NAMES MAY BE ADAPTED LOCALLY, BUT THE LISTED RECORDS SHOULD REMAIN IMMUTABLE.

Artifact group	Expected contents
configs/	Source-model identifiers, generation parameters, decoding settings, thresholds, and random seeds.
fingerprints/	Clean prompts, paper-reproduction fingerprint prompts, recorded targets, SCM trajectories, qualification records, and registry manifests; deployment targets must remain private.
model_pool/	Positive/negative taxonomy, checkpoint revisions, derivation categories, and compatibility notes.
responses/	Raw suspect outputs before parsing, grouped by source family, method, and robustness condition.
scores/	Parsed labels, target-hit vectors, per-suspect scores, AUC inputs, and subgroup summaries.
baselines/	Original baseline configurations, generated artifacts, official score fields, and any compatibility patch.
plots/	Figure-ready CSV/JSON files and deterministic scripts for every paper figure.
logs/	Construction, verification, extraction, robustness, and environment logs required for audit.
paper/	LaTeX source, bibliography, figure assets, and the arXiv build script.