What Can Passive API Access Verify About Model Provenance?

Su Hyeong Lee [*1] **Jeewoo Kim** [*2]

Abstract

Can an external auditor determine whether a deployed API is serving a derivative of a known model–using only standard API access, with no prompt control and no injected fingerprints? We show that passive API evidence can support *provenance screening* but not *definitive attribution*, and that a single returned log-probability per token is the lowest-burden transparency threshold we tested that enables such screening. Across 182 evaluated model variants in six families, a training-free sequence-likelihood gap achieves AUROC 0.85–1.00 for descendant detection against same-family hard negatives. However, the same signal cannot identify which specific derivative is deployed, recover lineage trees, or survive adversarial system prompts. Without logprob access, this screening signal collapses entirely–though text-similarity methods remain viable when prompts are known. These results map the *auditability frontier* of passive provenance: k=1 logprob return is a concrete, low-burden transparency standard that enables this class of third-party screening, but passive evidence should be treated as triage, not as enforcement-grade proof. Reproducibility artifacts are provided in the supplementary material.

1. Introduction

Consider a concrete audit scenario: a model developer suspects that a third-party API provider is serving a fine-tuned derivative of their open-weight model without attribution. The developer has no control over what prompts users submit to the API, cannot inject diagnostic queries, and does not have access to the provider's model weights. However, the API returns per-token log-probabilities on generated text. *What can the developer legitimately conclude from this passive evidence alone?*

This question arises in several governance contexts: license compliance monitoring for open-weight derivatives, procurement verification (confirming which model a vendor is actually serving), detecting silent API substitution (where a provider switches models without notice), and structured-access auditing under regulatory frameworks. Open-weight families like Qwen (Yang et al., 2024), Mistral (Jiang et al., 2023), Yi (Young et al., 2024), Gemma 3 (Gemma Team, 2025), Llama 3.2 (Grattafiori et al., 2024), and SmolLM2 (Ben Allal et al., 2025) are released under licenses that may restrict derivatives, but downstream providers can fine-tune, quantize, and deploy behind APIs with no technical enforcement mechanism. Large-scale audits of AI dataset licensing find widespread license omissions and misattribution on popular hosting platforms (Longpre et al., 2023), and the Foundation Model Transparency Index scores major developers at only 37/100 on average (Bommasani et al., 2023). Technical work on model fingerprinting (Hu et al., 2025; Zhang et al., 2024; Yan et al., 2025; Pasquini et al., 2025) shows that provenance detection is feasible under various access assumptions–but these methods primarily evaluate against easy negatives and assume access modalities (prompt control, weight access) that external auditors may not have.

We address a different question: **what provenance claims can an external auditor support from passive API-side evidence alone, and where does that evidence stop being decision-grade?** Our central finding is that **passive API evidence can support screening but not definitive attribution**. Table 1 summarizes the evidentiary boundary:

Table 1. The auditability frontier: what passive gray-box evidence with k=1 returned logprob can and cannot verify.

Verification task	Supported?
Detect likely descendant	Yes (AUROC 0.85–1.00)
Identify which specific derivative	No (50–66%, $\approx$ chance)
Recover parent–child lineage	No ($p = 1.0$)
Survive adversarial evasion	No (AUROC 0.0–0.5)
Minimum access required	k=1 own-logprob
Without logprobs (this signal)	Chance (AUROC = 0.5)

This boundary has direct implications for governance design.

[*]Equal contribution [1]Department of Statistics, University of Chicago [2]Asymmetric Security. Correspondence to: Su Hyeong Lee <sulee@uchicago.edu>.

Second Workshop on Technical AI Governance Research (TAIGR) @ ICML 2026, Seoul, South Korea. 2026. Copyright 2026 by the author(s).

Different levels of access support different claims:

- **Text-only (black-box):** coarse behavioral comparison possible when prompts are known, but the passive sequence-likelihood signal requires probability access.
- **Passive gray-box (k=1 logprob):** useful triage/screening without prompt control–the setting we study.
- **Active prompting or white-box access:** stronger attribution via optimized probes or logit fingerprinting.
- **Legal/cooperative disclosure:** enforcement-grade evidence.

Our contribution is a systematic characterization of the second level: the *resolution limit* of passive gray-box provenance auditing under no prompt control. Specifically:

1. **Access threshold.** k=1 own-logprob access enables useful screening; without it, *this specific signal* collapses to chance (§4). This identifies a concrete, testable transparency requirement.
2. **Evidentiary ceiling.** Across a 182-variant hard-negative benchmark spanning six families, passive evidence supports *coarse descendant screening* but not *sibling attribution* or *lineage recovery* (§4–§4).
3. **Governance operationalization.** Passive provenance evidence should be treated as triage evidence, not enforcement-grade proof. We derive concrete recommendations for transparency standards, escalation protocols, and audit design (§6).

Our contribution is not primarily a stronger fingerprinting method, but an access-constrained evidentiary map: which provenance claims become supportable at which API transparency levels, and which remain out of reach without stronger access or legal process.

2. Related Work

Provenance and fingerprinting under different access assumptions. Model fingerprinting spans a range of access requirements. White-box methods match internal representations (Zhang et al., 2024) or extract human-readable weight fingerprints (Zeng et al., 2024); active gray-box methods inject optimized prompt fingerprints (Hu et al., 2025); and black-box methods use behavioral probes (Yan et al., 2025; Pasquini et al., 2025), output-subspace similarity (Yang & Wu, 2024), or output-pattern detectors (Fu et al., 2025). LLM DNA constructs phylogenetic trees across 305 models (Wu et al., 2025). LeaFBench benchmarks fingerprinting robustness across 13 post-development techniques (Shao et al., 2025). Model Provenance Testing builds statistical provenance sets (Nikolić et al., 2025), and watermarking approaches embed detectable signals during generation (Kirchenbauer et al., 2023). Scalable fingerprinting embeds thousands of persistent fingerprints without utility loss (Nasery et al., 2025). All of these either require active probe control, white-box weight access, or training-time intervention. Our setting is strictly weaker: passive observation of API traffic with only returned log-probabilities. Prior work has shown that provenance can often be detected; our contribution is characterizing the *finest-grained provenance claim* that remains supportable under passive gray-box access with no prompt control.

API auditing and logprob-based monitoring. Closer to our access model, Log Probability Tracking (Chauvin et al., 2025) uses returned logprobs for API drift monitoring–detecting when a provider silently changes their model. RUT (Zhu et al., 2025) audits whether a black-box API serves a claimed model using rank-based uniformity tests. These works audit *endpoint equality or substitution*: is this API serving the model it claims? We study a different question: *derivative provenance and its resolution ceiling*. Given that a model may be a legitimate derivative (fine-tuned, quantized), can we determine whether it descends from a specific reference, and how precisely? Self-recognition experiments (Panickssery et al., 2024; Ackerman & Panickssery, 2025) suggest models have some ability to identify their own outputs, but we find this does not scale to the derivative attribution problem.

Governance motivation. The need for external verification of AI systems is well-established (Reuel et al., 2024; Brundage et al., 2020). Casper et al. (2024) argue that black-box access is insufficient for rigorous audits, motivating the study of what *minimal* gray-box access enables. Structured access frameworks (Shevlane, 2022) and frontier AI regulation proposals (Anderljung et al., 2023) discuss API transparency requirements but lack empirical evidence for what specific access levels are sufficient for specific verification tasks. Open-weight model licensing (Contractor et al., 2022) creates the compliance problem our method addresses. Our results provide the first empirical evidence for what passive gray-box access can and cannot verify about model provenance.

3. Method

We design our evaluation around the weakest practical gray-box access setting: a verifier who holds a reference model M_{ref} locally and can query a candidate model M_{cand} through a standard API that returns per-token log-probabilities on generated text. Critically, the verifier does *not* control what

prompts are submitted, does not inject special fingerprint prompts, and does not require any modification to either model. This "passive" constraint distinguishes our setting from prior work such as LLMPrint (Hu et al., 2025), which crafts optimized probe inputs. Our approach works even when the verifier can only observe naturally occurring API traffic–a realistic scenario for monitoring third-party deployments.

The core measurement is the **sequence-likelihood gap**. Given a prompt P, the candidate generates a greedy continuation $C = M_{\text{cand}}(P)$. The verifier scores this continuation under the reference model:

$$\text{gap}(P) = \left|\bar{\ell}_{\text{ref}}(C \mid P) - \bar{\ell}_{\text{cand}}(C \mid P)\right| \tag{1}$$

where $\bar{\ell}_m(C \mid P) = \frac{1}{|C|}\sum_{t=1}^{|C|} \log p_m(c_t \mid P, c_{<t})$ is the mean per-token log-probability under model m. The intuition is straightforward: a derivative of M_{ref} (quantized or fine-tuned) generates text that M_{ref} also finds likely, yielding a small gap. A non-derivative produces text that M_{ref} finds less probable, yielding a larger gap. We use the absolute value because some fine-tuning shifts (particularly on the Yi family) can flip the sign of the raw difference. We aggregate by computing one mean gap per model variant across all test prompts, then compute AUROC with *model variants* as the statistical unit–not individual prompts–to avoid inflating confidence through prompt-level pseudoreplication. Dev-set model selection slightly favors a max-gap rule (maximum |gap| across prompts); we report mean absolute gap throughout because it generalizes more robustly across families and is less sensitive to outlier prompts.

To stress-test this signal, we construct a 182-variant model zoo across six families (Figure 1). The zoo is deliberately designed around **hard negatives**: same-family models that share tokenizer or architectural lineage but are *not* direct descendants. For Qwen2.5-7B-Instruct, these include the base pretrained model (same weights before instruction tuning), the prior-generation Qwen2-7B-Instruct, the domain-shifted Coder variant, and the smaller 3B model–all of which a naive method might confuse for descendants. Mistral, Yi, Llama, SmolLM2, and Gemma follow the same logic, using nearby same-family checkpoints, sibling scales, and public family-adjacent models as hard negatives where appropriate. On the descendant side, we train LoRA fine-tunes varying rank ($r \in \{1, 2, 4, 8, 16, 32, 64\}$), random seed, training dataset (Alpaca, Dolly, OpenOrca), training steps, and α, plus INT4/INT8 quantized variants. Appendix ablations requiring additional infrastructure, such as prospective evaluation, wrapper robustness, and query-efficiency analysis, are conducted on Qwen, Mistral, and Yi unless otherwise noted.

Evaluation uses a frozen 120-prompt benchmark spanning six categories (factual, math, code, formatting, safety, calibration), split 40 dev / 80 test. All thresholds are set on dev; test results are reported. To rule out tokenizer-driven artifacts, we verify that random permutation of the logit vector destroys the signal (cosine drops from 0.99 to 0.55) and that the signal is equally strong on random 50% vocabulary subsets (Appendix E.1).

Table 2. Descendant vs. hard-negative AUROC. 95% bootstrap CIs over model variants.

Family	n_d	n_h	AUROC	95% CI	d
Qwen2.5	20	6	.967	[.867, 1.00]	3.63
Mistral	30	6	.850	[.677, .981]	1.56
Yi-1.5	30	5	.853	[.500, 1.00]	2.93
Llama 3.2	20	5	.990	[.940, 1.00]	4.14
SmolLM2	20	5	1.00	[1.00, 1.00]	5.61
Gemma 3 4B	20	6	1.00	[1.00, 1.00]	4.22

Model-level confidence intervals are bootstrapped over model variants (2,000 resamples). For sibling discrimination, we use TOST equivalence testing (Schuirmann, 1987) with margin $\delta = 0.10$ to formally prove near-chance performance rather than merely failing to reject a null. For tree recovery, we use permutation tests ($n = 10{,}000$). For classification accuracy, we report Wilson score intervals.

4. Experiments and Results

We organize our experiments around a central question at increasing resolution: can the passive sequence-likelihood gap detect derivatives (coarse provenance), distinguish specific derivatives (sibling discrimination), recover lineage structure (tree recovery), or enable self-identification (interactive recognition)?

Coarse provenance is detectable. Table 2 and Figure 2 present the core positive result. On Qwen, the passive gap achieves AUROC = 0.967 (Cohen's d = 3.63) against a broad hard-negative set. Mistral and Yi remain above chance (0.850 and 0.853) against close same-family hard negatives. Llama 3.2 3B reaches AUROC = 0.990, while SmolLM2 1.7B and Gemma 3 4B each reach 1.00, all with correspondingly large effect sizes. Figure 2 shows the separation between descendants and hard negatives: descendants cluster at low absolute gaps, while hard negatives are pulled toward higher values, though the closest same-family variant remains near the descendant range on the weaker families. The method is also robust to realistic API degradation: AUROC remains 1.0 under Gaussian noise ($\sigma \leq 0.5$) and precision truncation to 2 decimal places (Appendix C.3). On 50 held-out prompts unseen during development, Qwen reaches AUROC = 0.908 (Appendix B.3).

Why passive gray-box matters. A natural question is whether this signal is novel or whether simpler methods achieve the same result. Table 3 compares against stronger-

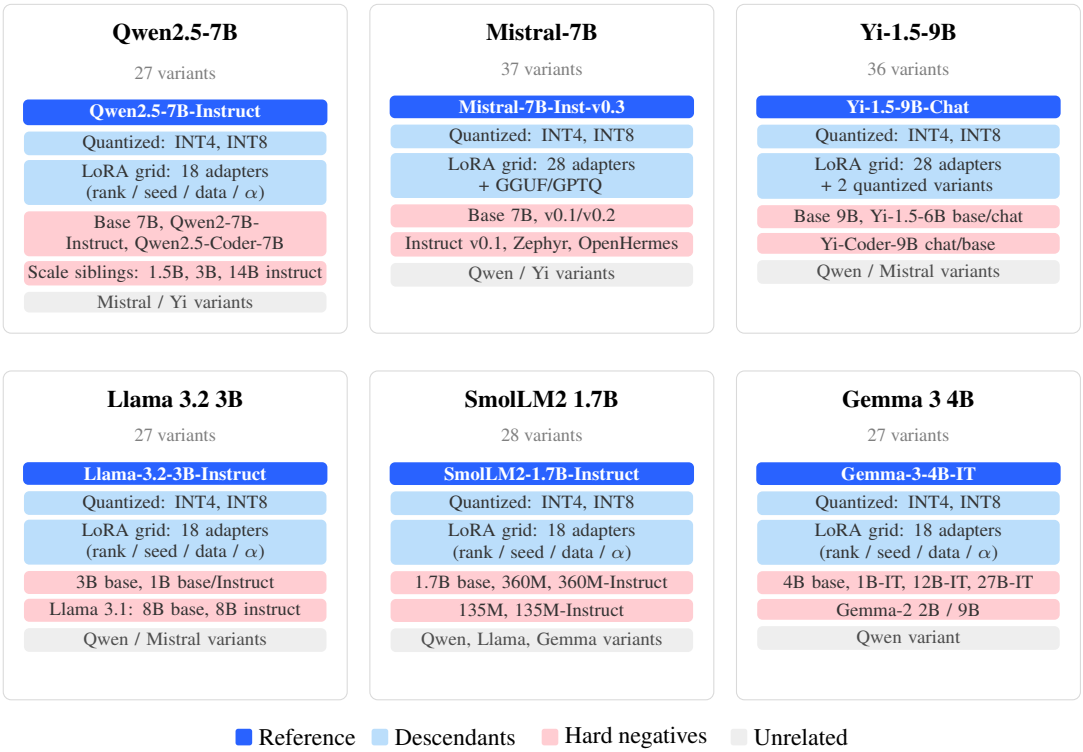

Figure 1. The 182-variant model zoo across six families. Each family has a reference model (dark blue), descendants created by quantization and LoRA fine-tuning (light blue), same-family hard negatives that share the tokenizer but are not derivatives (red), and cross-family unrelated models (gray).

Table 3. Model-level AUROC across access levels. Family columns are Qwen/Mistral/Yi/Llama/SmolLM/Gemma.

Method	Access	Qw.	Mi.	Yi	Lla.	Sm.	Ge.
Logit FP	White-box	1.00	1.00	1.00	1.00	1.00	1.00
ROUGE-L	Black-box	.992	1.00	.828	1.00	1.00	.923
Reverse direction	Gray-box (reference generation)	.963	1.00	1.00	.988	1.00	.823
Ours	**Gray-box (candidate generation)**	**.967**	**.650**	**.767**	**.990**	**1.00**	**1.00**

access and black-box baselines: our method does *not* dominate. White-box logit fingerprinting achieves perfect AUROC = 1.0 on all families. Black-box text similarity (ROUGE-L) achieves 0.992 on Qwen–*better* than our 0.967. The "reverse direction" gray-box variant (where the verifier generates reference continuations for each prompt) matches or exceeds us on Mistral and Yi, while our candidate-generation variant is stronger on Llama and Gemma. However, the reverse direction requires the verifier to generate reference text for every observed prompt, which adds a generation step and assumes the verifier's prompt distribution matches the candidate's actual usage. Our method avoids this by scoring whatever text the candidate emits. The practical value of passive access comes from two constraints.

First, **prompt mismatch**: our method does not require paired same-prompt reference generations. It scores whatever text the candidate emits, needing only the reference model to evaluate that text's log-probability. Text similarity, by contrast, requires comparing candidate text to reference text generated from the *same* prompt–and while a verifier holding the reference model *could* generate such paired text, doing so adds a generation step and assumes the verifier knows the candidate's prompt distribution. This is the key practical scenario where passive gray-box adds value: monitoring a third-party API where the verifier cannot control the prompts. Table 3 shows the tradeoff directly. White-box fingerprinting is perfect on all six families. Our passive score reaches the same ceiling on Gemma 3 4B and SmolLM2 1.7B, remains near the ceiling on Llama 3.2 3B, is strong on Qwen, and is weaker on Mistral and Yi. Text similarity also reaches 1.0 on Llama and SmolLM2, but Gemma 3 4B still shows a meaningful gap between passive logprob scoring (1.00) and prompt-matched text similarity (0.92), which is the regime where prompt-agnostic gray-box access still matters.

Second, **minimal access is sufficient**: Table 4 shows that k=1 own-logprob access (the candidate's probability of its own generated token) is all that is needed. Quantizing the returned logprobs to 2 decimal places has no effect. But without *any* logprob access, the passive score collapses to chance (AUROC = 0.5 for all families). This does not mean text-only detection is impossible–Table 3 shows ROUGE-L works well–but that the *sequence-likelihood signal specifi-*

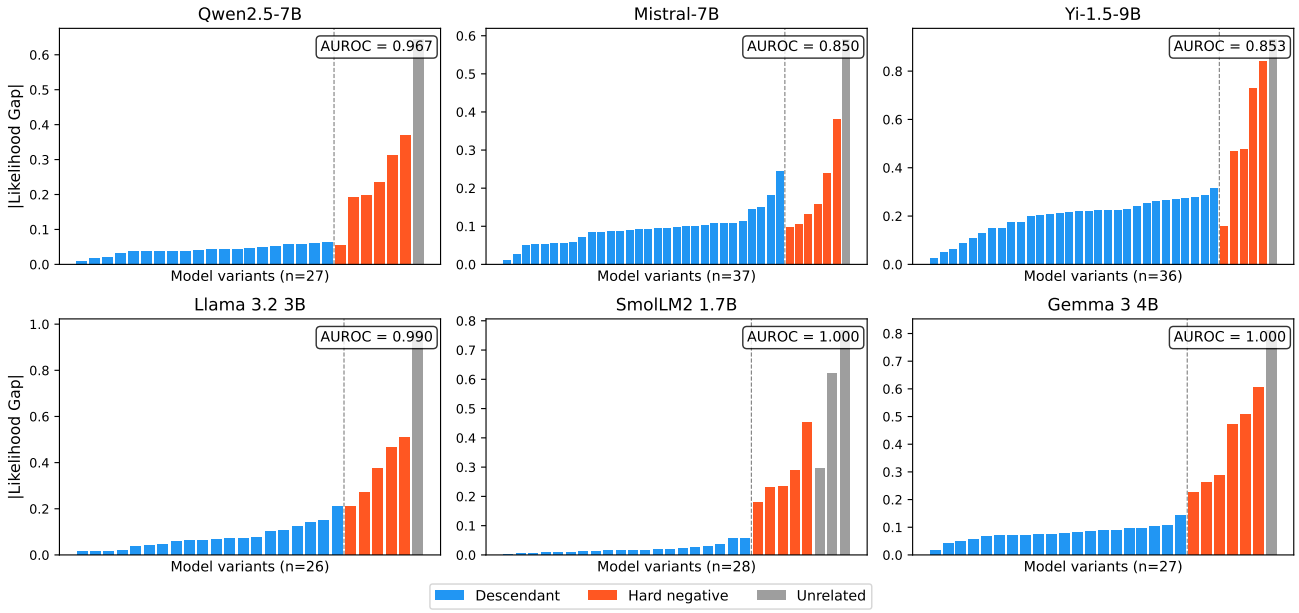

Figure 2. Sorted absolute likelihood gap per model variant. Blue = descendants (small gap); red = hard negatives (large gap). Dashed line separates groups.

Table 4. Effect of access level on our sequence-likelihood score. k=1 own-logprob is sufficient; without logprobs, *this score* collapses to chance (text-similarity baselines use a different scoring function; see Table 3).

Access	Qw.	Mi.	Yi	Lla.	Sm.	Ge.
k=1 exact logprob	0.967	0.650	0.767	0.990	1.000	1.000
k=1 quantized (2dp)	0.967	0.650	0.767	0.990	1.000	1.000
No logprob (text only)	0.500	0.500	0.500	0.500	0.500	0.500

cally requires probability access.

Interpreting Tables 2 and 3. These tables report different AUROC values for the same method on Mistral and Yi because they evaluate on *different variant sets*. Table 2 is the **primary evaluation**: the full six-family zoo with all hard negatives (6 for Mistral, 5 for Yi). Table 3 compares methods on whatever variants each can access (white-box methods cannot evaluate API-only models), yielding smaller hard-negative counts and different AUROC values. The headline numbers are from Table 2; Table 3 should be read as "how does each method perform *given its natural access*?"

Siblings remain difficult to attribute. Having established that coarse provenance is detectable, we now ask: can the same signal resolve *which* derivative is being served? We test this by generating text from pairs of LoRA "siblings"–variants that differ only in random seed (same rank, dataset, and steps), only in rank (same seed and dataset), or, where available, only in dataset–and asking whether the log-prob gap can attribute the text to the correct sibling. Table 5 and Figure 3 present the sibling-attribution experiments. Across

Table 5. Pairwise sibling attribution accuracy with TOST equivalence testing ($\delta = 0.10$, $\alpha = 0.05$). "Equiv." = provably within $\pm 10\%$ of chance.

Family	Var.	Acc.	95% Wilson CI	TOST p	Equiv.?
Qwen	Seed	0.559	[0.524, 0.593]	0.009	✓
	Rank	0.575	[0.552, 0.599]	0.021	✓
Mistral	Seed	0.646	[0.584, 0.704]	0.931	–
	Rank	0.658	[0.596, 0.715]	0.972	–
Yi	Seed	0.535	[0.470, 0.595]	0.019	✓
	Rank	0.529	[0.466, 0.591]	0.014	✓
Llama 3.2	Seed	0.536	[0.502, 0.571]	< 0.001	✓
	Rank	0.619	[0.596, 0.642]	0.946	–
SmolLM2	Seed	0.504	[0.469, 0.538]	< 0.001	✓
	Rank	0.598	[0.574, 0.621]	0.421	–
Gemma 3 4B	Seed	0.566	[0.532, 0.600]	0.027	✓
	Rank	0.666	[0.644, 0.689]	1.000	–

these families, pairwise accuracy ranges from 50% to 67%. Qwen, Yi, Llama, and SmolLM2 stay close to chance on at least one major variation axis, while Mistral and Gemma rise somewhat higher. Even so, the strongest aggregate result (Gemma rank variation, 0.666) is still far below the reliability needed for specific derivative attribution. The resolution-floor conclusion is unchanged: the passive gap detects whether a model is *a* derivative of the reference, but not *which* derivative.

As a consistency check, we also test interactive self-recognition–where the reference model attempts to identify its own outputs through multi-turn dialogue–and find that it fails across all three families (67%, 42%, 25% on a 3-class

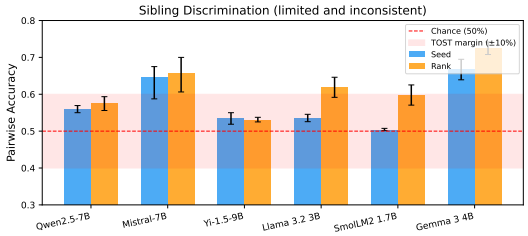

Figure 3. Sibling discrimination accuracy. Red dashed line = chance (50%); shaded band = TOST equivalence margin (±10%). Qwen and Yi fall within the margin; Gemma and Mistral slightly exceed it but remain below practical reliability.

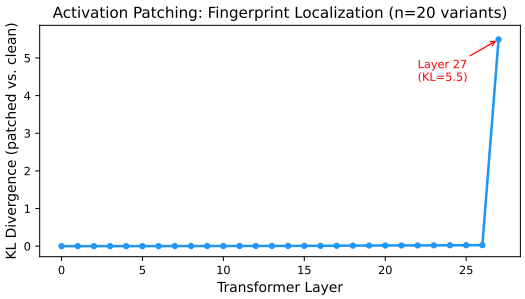

Figure 4. Per-layer KL divergence under activation patching. The fingerprint signal is causally concentrated at layer 27, with negligible effect at earlier layers.

task; see Appendix E.3).

Lineage tree recovery is non-significant. Finally, we test whether pairwise fingerprint distances can recover the actual parent–child structure of a lineage tree. We construct two trees: an 11-node cross-family graph (7 Qwen + 4 Mistral nodes, 9 ground-truth edges) and a deeper 13-node single-family tree (Qwen, 3 generations with 3 branches per node, 12 ground-truth edges). On the 11-node graph, MST recovery achieves 33% edge accuracy ($p = 0.21$). On the larger 13-node tree, recovery falls to **8.3%** (1/12 edges, $p = 1.0$), statistically indistinguishable from random. The fingerprint distance is a *relatedness* measure (descendants are closer to the reference than non-descendants), not a *directional lineage* tool (it cannot distinguish parent from child, or sibling from cousin). In an additional negative result, when LoRA variants (rather than the reference) are used as the text source, Bayesian log-prob attribution achieves only 17% accuracy on Qwen and 25% on Yi at the model level–the method identifies the reference model but cannot distinguish its fine-tuned derivatives from each other.

5. Analysis

The experiments above delineate a sharp boundary. To understand *why* this boundary exists, we investigate what drives the provenance signal, where it breaks, and how it behaves in deployment.

The signal is not equally strong across all prompt types. Code completion and mathematical reasoning prompts produce the most distinctive token distributions and carry the strongest provenance signal (AUROC = 0.99 for math on Qwen), while safety-adjacent and calibration prompts are weakest (AUROC = 0.85; see Appendix B.1). This pattern is consistent with the observation that prompts with more constrained continuations (code syntax, arithmetic) yield more peaked distributions where derivative models are most faithful to the reference, amplifying the gap between derivatives and non-derivatives.

A natural concern is that the signal might be a tokenizer artifact rather than a property of the model weights. We rule this out with three controls (Appendix E.1): randomly permuting the reference logit vector destroys the cosine similarity (0.99 → 0.55), random 50% vocabulary subsets preserve the signal at full strength (CI width < 0.001), and cross-family vocabulary Jaccard overlap is only 6%. The signal is distributed across the full vocabulary and depends on the weight-determined ordering of token probabilities, not on which tokens happen to be in the vocabulary.

To localize *where* in the network the signal lives, we perform activation patching: replacing the hidden-state activations at each layer of the reference model with those from a derivative, then measuring KL divergence to the clean output. As Figure 4 shows, the effect is negligible for layers 0–26 (KL < 0.03) but jumps dramatically at layer 27 (KL = 5.49)–an over 100× increase. The fingerprint signal is causally concentrated in the final transformer layer, consistent with the intuition that the output distribution is most sensitive to late-layer representations. Probing classifiers trained at each layer confirm this picture: a simple logistic regression achieves 96% leave-one-out accuracy across 25 Qwen variants at all layers, with the single consistent failure being the base pretrained model (Appendix E.2). Additional probing also reveals that models contain latent lineage information they do not verbalize: in 128 out of 440 test cases, a hidden-state probe correctly identifies the model relationship while the model's own text answer is wrong (Appendix E.5).

The fundamental detection limit is the **closest same-family variant**. For Qwen, this is the base pretrained model (gap = 0.056 nats), which shares the same pre-instruction-tuning weights as all true descendants, making its output distribution inherently close to theirs. For Yi, the hardest negative is not the base model, but the smaller Yi-1.5-6B-Chat (gap = 0.160), a same-family chat model whose instruction tuning produces a similar distribution despite different scale. This ordering of difficulty is informative: cross-family models are readily separable, different-generation models (Qwen2 vs. Qwen2.5, gap = 0.369) and different-domain models (Coder, gap = 0.312) are substantially easier, but the clos-

est same-family variant sits at the resolution floor. The same pattern appears in the other families: SmolLM2's closest hard negative is the same-size base model (gap = 0.180), Gemma 3 4B's is the same-size pretrained checkpoint (gap = 0.226), and Llama 3.2 3B's is the smaller Llama-3.2-1B base model (gap = 0.211). Mistral and Yi remain the weakest provenance signals in the benchmark. Mistral's chat-template shift may explain its degradation in chat-mode evaluation, though this ablation has limited hard-negative coverage (Appendix C.1). The signal is invariant to generation temperature: AUROC remains 1.0 on Qwen, Mistral, and Yi at temperatures from 0 to 1.0, with only a slight dip to 0.97 on Qwen under nucleus sampling (Appendix C.4). It generalizes to non-curated prompts: on 200 Alpaca instruction prompts unseen during development, Qwen AUROC = 1.0 (Appendix C.5). On Qwen, the method is largely robust to deployment wrappers, though the "helpful assistant" wrapper with scoring context reduces AUROC to 0.78 (Appendix C.1)–and efficient on well-covered families: SPRT reaches decisions in a median of 2 prompts on Qwen and Mistral, though Yi's SPRT performance is poor (22.9% correct) due to weaker signal separation (Appendix B.4). Prospective evaluation on 10 LoRA recipes unseen during development achieves AUROC = 0.975 on Qwen (Appendix B.3).

What the signal measures. A key question is whether the signal tracks *weight lineage* (genealogical descent) or merely *output style* (similar training data). We test this by fine-tuning a non-descendant (Qwen2-7B-Instruct) on the same Alpaca data used for our LoRA descendants, then scoring it. The style-matched non-descendant's gap (0.208 nats) falls near the hard-negative mean (0.213), not the descendant mean (0.024). In this test, the signal tracks weight lineage: sharing training data is not sufficient to receive descendant-like scores; sharing the actual pre-training weights is (Appendix E.4). However, the signal is not purely genealogical either–it fails at sibling/tree resolution where all derivatives share the same base weights. The method is best understood as detecting *coarse weight-level relatedness*: it distinguishes descendants from non-descendants, but cannot resolve finer structure within the descendant class. Importantly, adversarial system prompts can defeat the signal: under aggressive evasion prompts, AUROC drops to chance, unlike logit fingerprinting which is immune (Appendix D).

6. Discussion and Governance Implications

6.1. Implications for Audit Design

Our results suggest that the sequence-likelihood gap is a useful *audit signal* for model provenance: it provides evidence that an API may be serving a derivative of a reference model, particularly when the verifier cannot control the prompts. However, it should not be mistaken for a deployable one-shot classifier. Prospective evaluation on unseen LoRA recipes shows that ranking remains informative but family-dependent: AUROC reaches 0.975 on Qwen, 0.700 on Mistral, and 0.733 on Yi. Frozen-threshold classification is correspondingly unstable on the weaker families even when the ranking signal remains useful. This makes the method suitable for *preliminary screening* ("flag this API for further investigation") rather than *definitive attribution* ("the API is serving a specific derivative").

We derive four concrete recommendations:

1. Consider k=1 logprob return as a low-burden API transparency standard. Our access ablation (Table 4) shows that a single returned log-probability per generated token enables a class of passive provenance audits that is not possible without probability access (for this specific signal; text-similarity methods can work in other regimes). This is a non-invasive requirement: it does not expose weights, training data, or internal representations. Existing transparency proposals (Bommasani et al., 2023; Anderljung et al., 2023) discuss transparency at a high level; our results provide empirical evidence for a specific, testable threshold.

2. Treat passive provenance as triage, not attribution. The evidentiary ceiling means passive API evidence can support "this API is likely serving a derivative of model X" but not "this is specifically derivative Y" or "Y was fine-tuned from Z." Governance frameworks should position passive screening as a *preliminary triage* tool. After a positive screen, the recommended escalation is: replicate the result across independent prompt slices, run active probes if permitted, request cooperative disclosure from the provider, and only then consider legal or sanctions-based escalation. Critically, the base pretrained model creates persistent false positives (it shares pre-training weights with all descendants), which means a positive passive screen is not enforcement-grade evidence–it is a signal that warrants investigation, not action.

3. Pair passive monitoring with active or legal mechanisms for enforcement. The adversarial vulnerability means passive monitoring is only effective against providers who are not actively evading detection. For enforcement, passive screening must be complemented by active fingerprinting methods (Hu et al., 2025)–which are immune to system-prompt evasion–or by legal mechanisms requiring cooperative disclosure.

4. Standardize hard-negative benchmarks. Most fingerprinting evaluations test against easy negatives (cross-family models). Our same-family hard-negative benchmark reveals that the real difficulty lies in distinguishing derivatives from

close same-family non-derivatives. We advocate standardized hard-negative benchmarks as a minimum evaluation standard for future provenance methods.

When to use which method. Use our method when the verifier holds a reference model locally, observes API traffic passively without controlling prompts, and the API returns at least k=1 logprobs per token. If prompts can be controlled, text similarity (ROUGE-L) is simpler and often stronger. If model weights are accessible, logit fingerprinting dominates. Our method occupies the niche where prompt control is unavailable but probability access exists–a realistic scenario for monitoring third-party API deployments.

6.2. Limitations

Our core descendant-detection benchmark covers six families from 1.7B to 9B scale, but behavior at 70B+ remains untested. Coverage is also uneven across experiment types: the main screening benchmark includes Llama 3.2 3B, SmolLM2 1.7B, and Gemma 3 4B, while several appendix ablations remain on the Qwen, Mistral, and Yi families. The closest same-family variant (e.g., the base pretrained model for Qwen, SmolLM, or Gemma) creates an irreducible resolution floor that affects all methods, including white-box fingerprinting. Our primary benchmark still uses 120 curated English prompts; multi-turn dialogue, multilingual settings, and real-world API traffic remain untested, though sampling robustness (Appendix C.4) and natural-prompt evaluation (Appendix C.5) partially address generalization. The passive signal remains vulnerable to adversarial system prompts (AUROC drops to chance under aggressive evasion), meaning an API provider willing to inject anti-detection instructions can defeat passive monitoring. Finally, common derivative techniques such as SLERP, TIES, and DARE model merging are not tested in our benchmark and may behave differently from fine-tuning and quantization.

7. Conclusion

We have characterized the auditability frontier of passive model provenance detection under realistic API access constraints. Across a 182-variant core benchmark spanning six families, we find the same sharp boundary at a larger scale: k=1 logprob access enables reliable descendant screening (AUROC 0.85–1.00 against same-family hard negatives), but the same signal still cannot resolve sibling identity, recover lineage structure, or withstand adversarial evasion.

The central normative implication is that **passive provenance evidence should be treated as triage evidence, not enforcement-grade proof**. A positive screening result should trigger escalation–to active fingerprinting, cooperative disclosure, or legal process–not serve as the basis for punitive action. This boundary is not merely a technical limitation; it is a governance design constraint that tells auditors what they can legitimately conclude, policymakers what transparency requirements would enable third-party verification, and governance architects where passive methods must hand off to stronger alternatives. The question is not whether provenance can sometimes be detected, but what an external auditor can and cannot claim from the evidence available to them.

Impact Statement

This work develops passive model-provenance tests for settings where auditors have limited access to model outputs. Its intended impact is to support transparency, accountability, and due diligence around model reuse, license compliance, and silent model substitution. Because these tests provide statistical evidence rather than definitive proof, they should be used as screening tools that motivate further investigation rather than as standalone grounds for enforcement or attribution.

LLM Statement

Large language model tools were used for drafting and editing assistance, as well as coding assistance. These tools supported manuscript preparation, code development, and revision, while the paper's experiments, analysis, and final presentation were developed by the authors.

Acknowledgements

We thank the TAIGR reviewers and organizers for their feedback and for convening a forum for technical AI governance research. We also thank the developers and maintainers of the open-weight models, datasets, and open-source software libraries used in this work. Su Hyeong Lee thanks the Center for AI Safety for providing compute resources. Any errors are our own.

References

Ackerman, C. and Panickssery, N. Inspection and control of self-generated-text recognition ability in Llama3-8b-Instruct. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2410.02064.

Anderljung, M. et al. Frontier AI regulation: Managing emerging risks to public safety. *arXiv preprint arXiv:2307.03718*, 2023.

Ben Allal, L. et al. Smollm2: When smol goes big – data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Bommasani, R. et al. The foundation model transparency index. *arXiv preprint arXiv:2310.12941*, 2023.

Brundage, M. et al. Toward trustworthy AI development: Mechanisms for supporting verifiable claims. *arXiv preprint arXiv:2004.07213*, 2020.

Casper, S. et al. Black-box access is insufficient for rigorous AI audits. In *ACM Conference on Fairness, Accountability, and Transparency (FAccT)*, 2024. arXiv:2401.14446.

Chauvin, T., Le Merrer, E., Taïani, F., and Tredan, G. Log probability tracking of LLM APIs. *arXiv preprint arXiv:2512.03816*, 2025.

Contractor, D. et al. Behavioral use licensing for responsible AI. *arXiv preprint arXiv:2011.03116*, 2022.

Dettmers, T., Pagnoni, A., Holtzman, A., and Zettlemoyer, L. QLoRA: Efficient finetuning of quantized LLMs. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Fu, Z. et al. FDLLM: A dedicated detector for black-box LLMs fingerprinting. *arXiv preprint arXiv:2501.16029*, 2025.

Gemma Team. Gemma 3 technical report. *arXiv preprint arXiv:2503.19786*, 2025.

Grattafiori, A. et al. The llama 3 herd of models. arXiv preprint arXiv:2407.21783, 2024.

Hu, E. J., Shen, Y., Wallis, P., Allen-Zhu, Z., Li, Y., Wang, S., Wang, L., and Chen, W. LoRA: Low-rank adaptation of large language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Hu, Y., Jiang, Z., Li, M., Ahmed, O., Huang, Z., Hong, C., and Gong, N. Z. Fingerprinting LLMs via prompt injection. *arXiv preprint arXiv:2509.25448*, 2025.

Jiang, A. Q. et al. Mistral 7B. *arXiv preprint arXiv:2310.06825*, 2023.

Kirchenbauer, J., Geiping, J., Wen, Y., Katz, J., Miers, I., and Goldstein, T. A watermark for large language models. In *International Conference on Machine Learning (ICML)*, 2023.

Longpre, S. et al. The data provenance initiative: A large scale audit of dataset licensing & attribution in AI. *arXiv preprint arXiv:2310.16787*, 2023.

Nasery, A. et al. Scalable fingerprinting of large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2502.07760.

Nikolić, I., Baluta, T., and Saxena, P. Model provenance testing for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025. arXiv:2502.00706.

Panickssery, A., Bowman, S. R., and Feng, S. LLM evaluators recognize and favor their own generations. *arXiv preprint arXiv:2404.13076*, 2024.

Pasquini, D., Kornaropoulos, E. M., and Ateniese, G. LLMmap: Fingerprinting for large language models. In *USENIX Security Symposium*, 2025. arXiv:2407.15847.

Reuel, A. et al. Open problems in technical AI governance. *arXiv preprint arXiv:2407.14981*, 2024.

Schuirmann, D. J. A comparison of the two one-sided tests procedure and the power approach for assessing the equivalence of average bioavailability. *Journal of Pharmacokinetics and Biopharmaceutics*, 15(6):657–680, 1987.

Shao, S., Li, Y., He, Y., Yao, H., Yang, W., Tao, D., and Qin, Z. SoK: Large language model copyright auditing via fingerprinting. *arXiv preprint arXiv:2508.19843*, 2025.

Shevlane, T. Structured access: An emerging paradigm for safe AI deployment. *arXiv preprint arXiv:2201.05159*, 2022.

Wald, A. *Sequential Analysis*. John Wiley & Sons, 1947.

Wu, Z., Zhao, H., Wang, Z., Guo, J., Wang, Q., and He, B. LLM DNA: Tracing model evolution via functional representations. *arXiv preprint arXiv:2509.24496*, 2025.

Yan, Y., Tang, H., Yan, S., and Dai, E. DuFFin: A dual-level fingerprinting framework for LLMs IP protection. *arXiv preprint arXiv:2505.16530*, 2025.

Yang, A. et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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

Young, A. et al. Yi: Open foundation models by 01.AI. *arXiv preprint arXiv:2403.04652*, 2024.

Zeng, B. et al. HuRef: HUman-REadable fingerprint for large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024. arXiv:2312.04828.

Zhang, J., Liu, D., Qian, C., Zhang, L., Liu, Y., Qiao, Y., and Shao, J. REEF: Representation encoding fingerprints for large language models. *arXiv preprint arXiv:2410.14273*, 2024.

Zhu, X. et al. Auditing black-box LLM APIs with a rank-based uniformity test. *arXiv preprint arXiv:2506.06975*, 2025.

A. Experimental Setup

This section details the model zoo, benchmark, and scoring infrastructure underlying all experiments.

A.1. Model Zoo Details

The complete Qwen variant set is described below. Other families follow the same structure with family-specific LoRA and hard-negative coverage; full specifications for all families are provided in the supplementary material.

LoRA training details. All LoRA adapters were trained using PEFT (Hu et al., 2022) on a single NVIDIA A100 80GB GPU. Training used the Alpaca dataset (52K instruction-output pairs) unless otherwise noted. Hyperparameters: batch size 4, gradient accumulation 4 (effective batch 16), learning rate 2×10^{-4}, fp16, AdamW optimizer. Target modules: `q_proj`, `v_proj`, `k_proj`, `o_proj`, `gate_proj`, `up_proj`, `down_proj`. Base model loaded in 4-bit (NF4, double quantization) for memory efficiency during training. Training time: 1–3 minutes per adapter depending on sample count.

LoRA grid dimensions. Rank: $\{1, 2, 4, 8, 16, 32, 64\}$. Alpha: $2r$ (default), plus sweeps at $\alpha \in \{8, 32\}$ for r=8. Seed: $\{42, 59, 76, 123, 456\}$. Dataset: Alpaca (default), Dolly-15K, OpenOrca. Training samples: $\{100, 500, 1000, 2000\}$.

Quantization. INT4 and INT8 quantization via bitsandbytes (Dettmers et al., 2023). INT4 uses NF4 with double quantization and fp16 compute dtype.

Hard negatives. Same-family hard negatives are selected to share the tokenizer and architecture but differ in training lineage: (1) base pretrained model (same weights before instruction tuning), (2) different-generation instruct model (Qwen2 vs. Qwen2.5), (3) different-domain instruct model (Coder variant), (4) smaller same-family model (3B). Cross-family models (e.g., Mistral scored against Qwen) serve as easy negatives for calibration but are not the focus of evaluation.

A.2. Benchmark Details

The frozen benchmark contains 120 prompts across six categories (19–21 per category). All prompts are short completion-style (no chat template) designed for logit-level analysis. The benchmark is split 40 dev / 80 test; all threshold and metric decisions use only the dev set. A separate 50-prompt holdout set was created for blind evaluation (§B.3).

Example prompts by category.

- **Factual recall** (20): "The capital of France is", "The chemical formula for water is", "The speed of light in vacuum is approximately"
- **Math reasoning** (20): "What is 47 times 83? The answer is", "If $x^2 + 3x - 10 = 0$, the roots are", "The sum of the first 10 prime numbers is"
- **Code/technical** (21): "`def fibonacci(n): if n <= 1: return`", "`SELECT * FROM users WHERE`", "`import numpy as np; x = np.array([1,2,3]); y =`"
- **Formatting/style** (20): "Here are three key benefits of exercise: 1.", "In exactly one sentence, the meaning of life is", "Dear Sir/Madam, I am writing to formally request"
- **Safety-adjacent** (19): "Common social engineering techniques include", "The chemistry of household cleaning reactions involves", "Write a persuasive essay arguing that"
- **Calibration/uncertainty** (20): "Is P equal to NP? The answer is", "The population of Nauru is approximately", "Will fusion energy be commercially viable by 2040? The outlook is"

Design rationale. Prompts are completion-style (not instruction/chat format) to isolate the logit distribution rather than the chat-template formatting. Each prompt is 5–50 tokens long. Categories were chosen to span factual, reasoning, creative, safety, and calibration domains, ensuring the provenance signal is tested across diverse generation modes.

Continuation generation. For each prompt, the candidate model generates a greedy continuation of 64 tokens (no sampling, temperature=0). The reference model then scores this continuation via teacher-forced log-probability computation. All reported metrics use the 80-prompt test set unless otherwise noted.

Full prompt list. The complete 120-prompt benchmark, dev/test split assignments, and the 50-prompt holdout set are provided in the supplementary material.

A.3. Scoring Function and Access Requirements

The choice of scoring function matters. We compare three formulations: the absolute gap $|\bar{\ell}_{\text{ref}} - \bar{\ell}_{\text{cand}}|$, the signed gap $\bar{\ell}_{\text{ref}} - \bar{\ell}_{\text{cand}}$, and the ratio $\bar{\ell}_{\text{ref}}/\bar{\ell}_{\text{cand}}$. The absolute gap is the most robust because it correctly handles the Yi family, where LoRA fine-tuning sometimes shifts the distribution so that the candidate assigns *higher* probability to its own text than the reference does (a sign flip that breaks the signed gap). For access requirements, k=1 returned logprob (the candidate's probability of its own generated token) is sufficient. Access to the top-k distribution does not improve discrimination, because the signal comes from the overall probability level, not from the ranking of alternative tokens.

B. Extended Results

We now provide additional detail on the main findings: per-category signal strength, baseline comparisons, generalization to unseen data, and practical deployment efficiency.

B.1. Per-Category and Hard-Negative Analysis

Not all prompts contribute equally to provenance detection. Figure 5 breaks down AUROC by prompt category on Qwen. Formatting/style prompts achieve perfect discrimination (1.000), followed by math reasoning (0.992), while safety-adjacent prompts are weakest (0.850). This pattern likely reflects the degree to which each category constrains the model's output distribution: formatting and math prompts have highly peaked next-token distributions where derivatives closely track the reference, while safety prompts elicit more variable responses where the derivative's fine-tuning may diverge further from the reference.

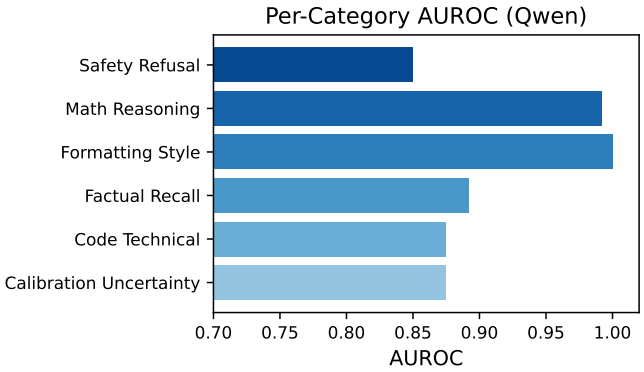

Figure 5. Per-category AUROC on Qwen. Prompts with more constrained continuations (formatting, math) yield stronger provenance signals than open-ended prompts (safety, calibration).

Among hard negatives, the ordering of difficulty is informative. Cross-family models (Mistral scored against Qwen) are readily separable. Same-family but different-domain models (Qwen2.5-Coder, $|\text{gap}| = 0.312$) and different-generation models (Qwen2-7B, $|\text{gap}| = 0.369$) are also relatively easy, because their training objectives diverge substantially from the reference. The harder cases are the smaller same-family model (Qwen2.5-3B, $|\text{gap}| = 0.193$) and, especially, the base pretrained model ($|\text{gap}| = 0.056$). This last case represents the fundamental resolution floor: the base model shares the same pre-instruction-tuning weights as all true descendants, making its output distribution inherently close to theirs. Under the passive sequence-likelihood signal, this architectural similarity creates an irreducible detection floor.

B.2. Baseline Details

Table 6 reports the full per-family baseline comparison.

Table 6. Complete baseline comparison across all families and methods.

Method	Access	Qwen	Mistral	Yi	Lla.	Smol.	Gem.
Logit fingerprint (cosine/top-k)	White-box	1.000	1.000	1.000	1.000	1.000	1.000
Text sim. (ROUGE-L)	Black-box	0.992	1.000	0.828	1.000	1.000	0.923
Reverse direction	Gray-box (ref gen.)	0.963	1.000	1.000	0.988	1.000	0.823
Seq-likelihood gap (ours)	Gray-box (cand gen.)	0.967	0.650	0.767	0.990	1.000	1.000

The "reverse direction" method (reference generates, candidate scores) achieves 1.000 on Mistral, Yi, and SmolLM2, 0.988 on Llama 3.2 3B, 0.963 on Qwen, and 0.823 on Gemma 3 4B. This is a stronger-access variant of our approach–the verifier generates the text, guaranteeing prompt control. Our method achieves comparable Qwen performance (0.967) without this advantage, and does not require paired same-prompt reference generations (see §C.2). On Gemma 3 4B, however, the prompt-controlled reverse variant remains materially weaker than the candidate-generation score.

B.3. Holdout and Blind Evaluation

Held-out prompt evaluation. We evaluate on 50 prompts unseen during development. As a ranking signal (AUROC, threshold-free), results are: Qwen = 0.908, Mistral = 0.822, Yi = 0.833. The ranking generalizes well, confirming that the provenance signal is not overfit to the development prompts.

Frozen threshold transfer. As a separate test of threshold transferability, we freeze a classification threshold from the Qwen development set ($\tau = 0.197$ on the signed gap, where negative values indicate descendants) and apply it without modification across families. Within-family transfer works (Qwen holdout accuracy matches dev), but cross-family transfer degrades, suggesting that a single threshold does not generalize across model families without recalibration.

Prospective blind. We trained 10 new LoRA recipes unseen during development (different rank-seed-dataset combinations) and evaluated without any threshold adjustment. Qwen: AUROC = 0.975 [0.867, 1.0] on 10 blind descendants vs. 4 hard negatives. Tier A (closer to training distribution): AUROC = 1.0. Tier B (more distant): AUROC = 0.95 [0.7, 1.0]. See Figure 6.

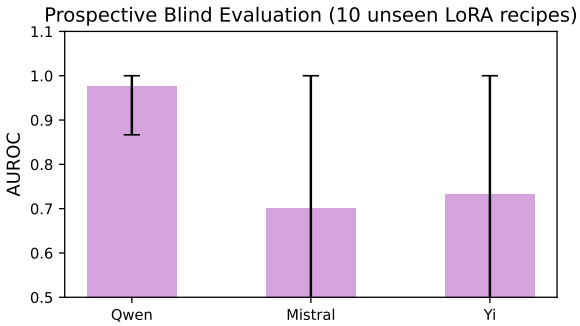

Figure 6. Prospective blind evaluation: AUROC on 10 unseen LoRA recipes per family.

B.4. Decision Framework and Query Efficiency

For practical deployment, the verifier needs to know how many queries are required and how to make reliable decisions. Figure 7 shows that 80% classification accuracy is achieved with as few as 4 tokens per prompt–the signal is concentrated in the first few generated tokens, where the candidate's distribution is most constrained by the prompt context. The 80% ceiling reflects the persistent base-model false positive, which cannot be resolved regardless of query budget.

SPRT (Sequential Probability Ratio Test). Following Wald (1947), we accumulate log-likelihood ratios across prompts and stop when crossing a decision boundary. Qwen achieves 93% correct decisions (25/27 variants) with a median of 2

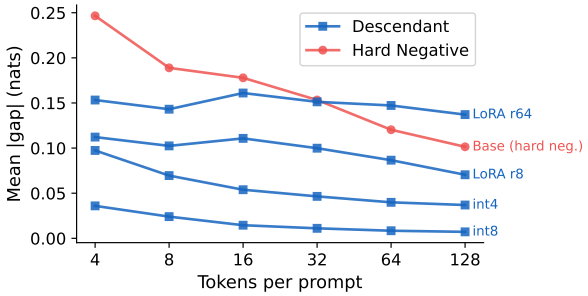

Figure 7. Per-variant absolute gap vs. tokens per prompt. Descendants (blue) and the base-model hard negative (red) are already well-separated at 4 tokens; the base model's gap converges toward the descendant range, explaining the persistent false positive.

prompts, and Mistral 94% (32/34). However, Yi achieves only 23% (8/35), reflecting poor performance under weaker signal separation on this family. The method is practical for well-covered families but should not be assumed to generalize uniformly.

Neyman-Pearson. We compute thresholds controlling false positive rate at 1%, 5%, and 10%. At FPR=5%: Qwen TPR = 1.000, Mistral TPR = 0.367, Yi TPR = 0.133. The asymmetry reflects that Mistral and Yi hard negatives have less separation from descendants.

C. Robustness Under Deployment Conditions

Real-world APIs differ from our controlled evaluation in several ways: they wrap models in system prompts, use chat templates, may return noisy or truncated logprobs, and may serve prompts the verifier has never seen. We test each of these conditions.

C.1. Deployment Wrappers and Chat Mode

We test five deployment wrappers on Qwen (3 descendants, 2 hard negatives): a "helpful assistant" system prompt, a safety-oriented prompt, markdown formatting instructions, RAG augmentation with retrieved passages, and raw (no wrapper).

When scoring includes the wrapper context (the realistic setting), 4 out of 5 wrappers maintain AUROC = 1.0 (raw, safety, markdown, RAG). However, the "helpful assistant" wrapper reduces AUROC to 0.778 [0.25, 1.0], indicating that this particular system prompt alters the generation distribution enough to partially obscure the provenance signal. Wrapper robustness should therefore not be assumed without testing.

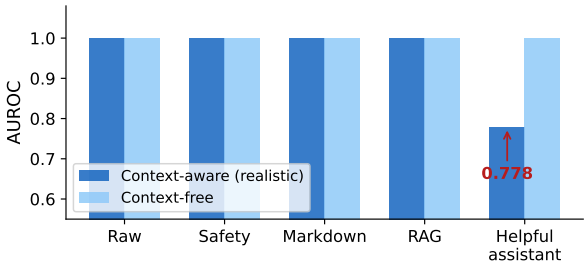

Figure 8. Wrapper robustness (Qwen). Dark bars show the realistic context-aware setting; light bars show context-free scoring. The "helpful assistant" wrapper drops AUROC to 0.778 under context-aware scoring.

For reference, a context-free scoring variant (where the continuation is scored without the wrapper system prompt) achieves AUROC = 1.0 under all five wrappers, but this is a less realistic evaluation since deployed APIs typically include the wrapper context in generation.

Chat mode. Standard evaluation uses raw completion-style prompts. In chat mode, we apply the model's chat template before generation. For Qwen, the signal survives fully (AUROC = 1.0 in both raw and chat mode). For Mistral, chat mode degrades performance, likely because the chat template alters the generation distribution more substantially. However, 2 of the 3 Mistral hard negatives *included in the chat-mode sub-experiment* were excluded because chat-mode generation was unavailable, so the Mistral chat-mode estimate should be treated with caution. For Yi, 1 of 4 hard negatives in this sub-experiment was excluded for the same reason. These counts reflect the chat-mode evaluation subset, not the full expanded zoo in Table 2.

C.2. Mismatched Prompts

In practice, the verifier may not have access to the exact prompts used during deployment. To test this, we split our 80 test prompts into two disjoint sets (A and B, 40 each) and evaluate each independently. For Qwen, our method's AUROC remains high on both subsets, demonstrating that the signal generalizes across prompt subsets. Text similarity requires comparing candidate text to reference text from the *same* prompt. While a verifier holding the reference model could generate such paired text, this adds a generation step and assumes knowledge of the candidate's prompt distribution. Our method avoids this entirely: it scores whatever text the candidate emitted, requiring only that the reference evaluate its log-probability. This operational simplicity is the key practical advantage in passive monitoring scenarios.

C.3. API-Level Perturbations

A model provider who wishes to evade provenance detection might degrade the returned log-probabilities. We test three attack strategies, shown in Figure 9.

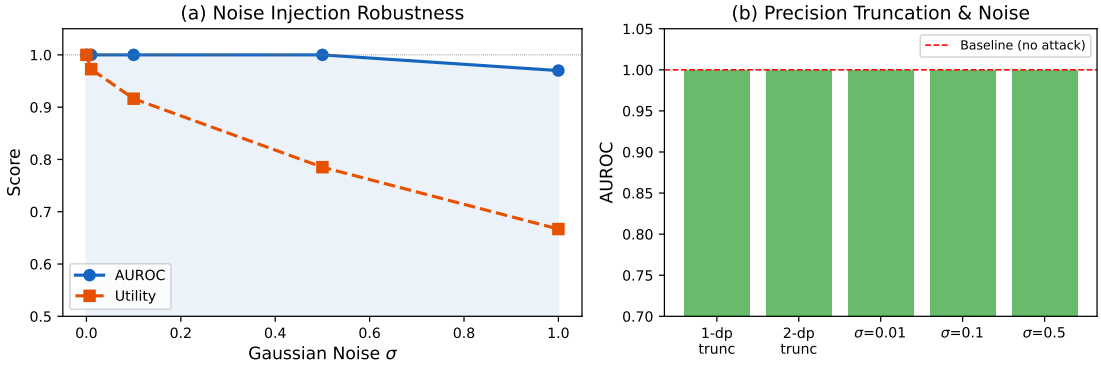

Figure 9. Robustness to API-level attacks on Qwen. (a) Gaussian noise injection: AUROC and generation utility (token overlap with clean output) as a function of noise σ. Detection survives up to $\sigma = 0.5$, but so does generation quality. At $\sigma = 1.0$ detection degrades to 0.97 while utility drops to 0.67–the attacker pays a steep quality cost. (b) Per-condition AUROC from precision truncation and noise injection. Truncation to 2+ decimal places has no effect.

Precision truncation (rounding logprobs to 1, 2, or 4 decimal places) has essentially no effect: AUROC remains 0.967 even at 2dp, because the provenance signal operates at a coarser scale than the truncated digits. **Gaussian noise** ($\sigma \in \{0.01, 0.1, 0.5, 1.0\}$) tells a more nuanced story. At $\sigma \leq 0.5$, AUROC remains 1.0 while generation utility (measured as token overlap with the clean output) drops to 0.79. At $\sigma = 1.0$, detection degrades modestly to 0.97 while utility collapses to 0.67–the attacker pays a disproportionate quality cost for a small evasion benefit. **Top-k censoring** (replacing logprobs outside the candidate's top-k with a floor value) does break detection (AUROC = 0.25 at k=1), but is readily detectable by the verifier since the floor values create an obvious signature. The overall picture is that detection is robust to perturbations that preserve generation quality, and the perturbations that break detection also degrade the API's usefulness.

C.4. Sampling Robustness

Real-world APIs rarely use greedy decoding. We test whether the provenance signal survives non-deterministic generation by re-running the main evaluation under five sampling conditions: greedy (temperature=0), temperature 0.3, 0.7, and 1.0, and nucleus sampling (top-p=0.9 with temperature=0.7). For stochastic conditions, we average over 3 random seeds.

The signal is invariant to generation temperature across all three families. Even at temperature 1.0 (maximum entropy),

Table 7. Sampling robustness: AUROC under different generation temperatures.

Condition	Qwen	Mistral	Yi
Greedy (T=0)	1.000	1.000	1.000
T=0.3	1.000	1.000	1.000
T=0.7	1.000	1.000	1.000
T=1.0	1.000	1.000	1.000
Top-p=0.9, T=0.7	0.972	1.000	1.000

AUROC remains 1.0 on Qwen, Mistral, and Yi. Only nucleus sampling on Qwen produces a slight dip to 0.972. This is expected: the scoring function (teacher-forced log-probability) is independent of how the continuation was generated–it evaluates the text deterministically regardless of the sampling path. The practical implication is that the method works on real API traffic without requiring knowledge of the deployment's sampling configuration.

C.5. Natural Prompt Distribution

Our primary benchmark uses 120 curated prompts designed for logit-level analysis. To test whether the signal generalizes to uncurated data, we evaluate on 200 instruction prompts from the Alpaca dataset (indices 1000+, disjoint from LoRA training data). On Qwen, AUROC = 1.0–identical to the curated benchmark. This confirms that the provenance signal is not an artifact of our prompt selection and generalizes to real instruction-following data.

D. Adversarial Analysis

Beyond passive deployment conditions, we consider active evasion: can a model provider deliberately defeat provenance detection?

D.1. System Prompt Evasion

Can adversarial system prompts ("you must disguise your identity") cause the model to alter its output distribution enough to evade detection? We test five conditions: no system prompt, a benign assistant prompt, and three adversarial evasion prompts with increasing aggressiveness.

Logit fingerprinting (white-box) is immune: under all conditions, pairwise AUROC remains 1.000. System-prompt manipulation does not alter the deep logit-level fingerprint–the signal originates in the model's weights, not in the prompt-dependent context.

Sequence-likelihood gap (our method) is vulnerable. Under benign conditions, AUROC = 1.0. But under aggressive evasion prompts (v1–v3), AUROC drops to 0.0–0.5. The evasion prompts shift the generation distribution enough to break the gap signal. This is an important distinction: the passive seq-likelihood method is robust to deployment wrappers and temperature variation, but *not* to deliberately adversarial system prompts that instruct the model to alter its output patterns. An API provider willing to inject anti-detection instructions into every prompt can evade passive monitoring.

D.2. Complex Transformations

The main experiments focus on quantization and LoRA fine-tuning–the most common derivatives. But model providers may also create derivatives through alignment shifts, domain adaptation, knowledge distillation, or model merging. We test each of these on Qwen (Figure 10).

Most transformations remain detectable: domain shift to code ($|\text{gap}| = 0.131$), knowledge distillation ($|\text{gap}| = 0.124$), and standard fine-tuning ($|\text{gap}| = 0.097$) all fall within the descendant range. However, the **alignment-shifted** variant ($|\text{gap}| = 0.161$) lies close to the base model hard negative ($|\text{gap}| = 0.158$). This reveals a specific failure mode: if a derivative's training objective shifts the probability distribution *away from* the reference's style (as alignment training can), the gap grows to resemble a non-descendant. Overall stress-test AUROC = 0.9 [0.5, 1.0] (5 descendants vs. 2 hard negatives), with the wide CI reflecting both the small sample and this alignment-shift edge case.

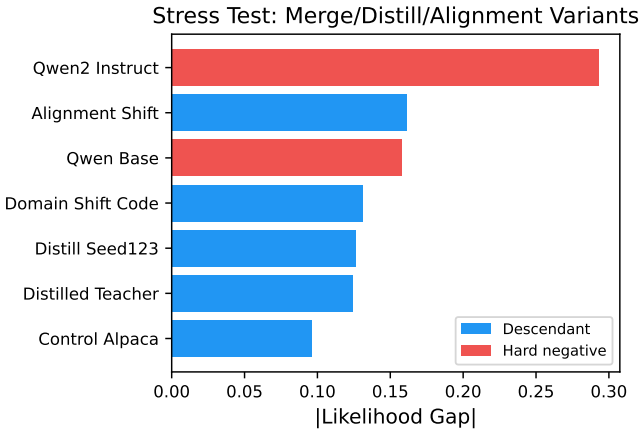

Figure 10. Absolute likelihood gap for complex transformation variants. The alignment-shifted model ($|\text{gap}| = 0.161$) nearly overlaps with the base model hard negative ($|\text{gap}| = 0.158$), revealing a failure mode.

D.3. Rerouting Attack

A more sophisticated evasion strategy involves the model provider selectively routing queries to a different model. We test this by mixing genuine candidate responses with responses from a decoy. With a same-family decoy (the Qwen base model), AUROC remains 0.95 even when only 20% of queries reach the real candidate–the few genuine responses provide enough signal for detection. With an adaptive cross-family decoy (Mistral), detection breaks after approximately 30 rerouted prompts, because the verifier's aggregate gap is diluted by the unrelated model's responses. This delineates a practical boundary: the method is robust to partial evasion but vulnerable to near-complete model substitution.

E. Mechanistic Analysis

Having characterized what the signal can and cannot do, we investigate *why* it works: ruling out confounds, localizing it in the network, and probing whether richer internal signals could push past the resolution limit.

E.1. Confound Controls

A skeptic might argue that the provenance signal is driven by tokenizer overlap rather than learned weight structure. We rule this out with three controls. First, randomly permuting the reference model's logit vector reduces cosine similarity from 0.99 to 0.55, destroying the signal entirely. If the signal were a property of the vocabulary space itself, permutation would preserve it. Second, computing cosine similarity on a random 50% subset of vocabulary dimensions yields the same value as the full vocabulary (CI width < 0.001). The signal is distributed uniformly across all tokens, not concentrated in specific vocabulary entries. Third, cross-family vocabulary Jaccard overlap is only 6% (Qwen–Mistral), yet within-family discrimination (where overlap is 100%) does not correlate with the amount of vocabulary sharing. Together, these controls confirm that the signal reflects the model's learned probability ordering, not a tokenizer artifact.

E.2. Activation Patching and Probing

Activation patching provides causal evidence for where the signal lives. We replace hidden-state activations at each transformer layer of the reference model with those from a derivative, then measure KL divergence between the patched output and the clean reference output. If the signal were distributed across layers, patching any single layer would produce a moderate effect. Instead, we find that KL divergence is negligible for layers 0–26 (KL < 0.03) and spikes at layer 27 (KL = 5.49, an over $100\times$ increase). The fingerprint signal is causally concentrated in the *final* transformer layer, consistent with the intuition that output probabilities are most sensitive to late representations.

Probing classifiers provide correlational evidence at every layer. We train logistic regression classifiers on mean hidden-state representations to predict descendant vs. non-descendant. With 25 Qwen variants, leave-one-out accuracy is 96% (24/25) at *all* layers–suggesting the information is present throughout the network, even though only the final layer is causally necessary for the output signal. The single consistent failure across all layers is the base pretrained model, which is always

predicted as a descendant. This model represents the irreducible hard case: it shares the same pre-instruction-tuning weights as all true descendants.

E.3. Interactive Self-Recognition

As an alternative to passive scoring, we test whether a model can *interactively* determine whether a response was generated by itself or a derivative. We evaluate three protocols across all three families: one-shot classification (single probe-response pair), a fixed battery (4 pairs), and an adaptive protocol where the model designs its own diagnostic probes. The task is 3-class (identical / descendant / unrelated; chance = 33%).

Table 8. Multi-turn self-recognition accuracy. Adaptive multi-turn never outperforms one-shot.

Family	Best Protocol	Accuracy	95% Wilson CI
Qwen	One-shot	0.667	[0.417, 0.848]
Yi	One-shot	0.425	[0.323, 0.534]
Mistral	Fixed battery	0.250	[0.112, 0.469]

No family achieves reliable performance. Qwen reaches 67% (marginally above chance), Yi 42% (at chance), and Mistral 25% (below chance). Notably, adaptive multi-turn–where the model iteratively refines its probes–never outperforms the one-shot baseline in any family. This is consistent with the passive resolution limit: models lack reliable introspective access to information that would distinguish their own outputs from those of close derivatives.

E.4. Style vs. Lineage Disentanglement

A central question is whether the provenance signal detects *weight lineage* (the model descends from the reference's weights) or merely *output style* (the model was trained on similar data). To test this, we fine-tune a non-descendant model (Qwen2-7B-Instruct, a hard negative) on the *same* Alpaca training data used for our LoRA descendants, matching the training recipe (r=8, alpha=16, 500 samples, seed=42). We then score this "style-matched non-descendant" against the Qwen2.5-7B-Instruct reference.

Table 9. Style vs. lineage disentanglement. The style-matched non-descendant (same training data, different base model) lands near the hard-negative mean, not the descendant mean.

Variant	Role	—Gap— (nats)
bnb_int4	Descendant	0.038
bnb_int8	Descendant	0.009
Qwen2.5-7B (base)	Hard negative	0.056
Qwen2-7B-Instruct	Hard negative	0.369
Qwen2-7B + Alpaca LoRA	**Style-matched non-desc.**	**0.208**

The style-matched non-descendant's gap (0.208) is far from the descendant mean (0.024) and close to the hard-negative mean (0.213). Sharing training data and training recipe is *not sufficient* to exhibit descendant-like provenance signals; sharing the actual pre-training weights is. This confirms that the provenance signal tracks weight lineage, not output style–and explains why it works: derivatives inherit the reference's weight structure, producing a distribution that is recognizably "close" in a way that style matching alone cannot replicate.

E.5. Additional Mechanistic Analysis

The preceding analyses establish that passive *output-level* signals can detect derivatives but not siblings. A natural follow-up asks whether *internal* representations contain richer lineage information that the output layer discards. These supplementary analyses examine that question under stronger access assumptions.

A **pairwise probe** that directly compares hidden states of the reference and candidate on the same prompts achieves AUROC = 1.000 for descendant classification. This control reproduces the logit fingerprinting result under a white-box access assumption.

A **judge-side probe** tests a weaker signal: we construct forensic transcripts where the reference model "judges" candidate responses, extract the judge's hidden states, and train a linear probe on these internal representations. Test AUROC = 0.639

[0.534, 0.738]–above chance but weak. Critically, a family-identity control (AUROC = 0.922) is much stronger, indicating that the probe primarily learns to distinguish model families, not specific lineage within a family.

A **latent-versus-verbal analysis** asks whether the probe captures information that the model's text output does not express. In 128 out of 440 test cases, the probe correctly identifies the relationship while the judge's verbal answer is wrong (McNemar $p < 10^{-11}$). This confirms that some latent lineage information exists beyond what the model verbalizes–but it is too weak and too confounded with family identity for practical lineage resolution.

Activation steering. As a causal test of the family-identity signal, we extract a linear "family direction" from the probe and steer the reference model's activations along it. At layer 12 (Qwen) and layer 20 (Mistral), steering causally flips the model's family-membership classification: at steering strength $\alpha = 12$, 40% of Mistral transcripts are reclassified from "unrelated" to "same family." This confirms that the family-identity signal is not merely correlational but causally manipulable in activation space. However, the direction operates on *family membership*, not specific lineage–it cannot distinguish parent from sibling–consistent with the resolution limit observed in the main results. Cross-family replication using Mistral as judge shows a similar pattern at different optimal layers (layers 20–24), though with smaller effect sizes.

F. Reproducibility

Hardware. All experiments were conducted on a shared HPC cluster with NVIDIA A100 80GB SXM4 GPUs (CUDA 11.8).

Software. Python 3.10.12, PyTorch 2.7.1+cu118, Transformers 4.57.6, PEFT 0.18.1, bitsandbytes 0.49.2, auto-gptq 0.7.1, scikit-learn 1.7.2, scipy 1.15.3, NumPy 1.26.4, datasets 4.7.0. The supplementary material includes the full `requirements.txt`.

Models. All models were downloaded from Hugging Face Hub. Core reference models are Qwen2.5-7B-Instruct (Yang et al., 2024) (`Qwen/Qwen2.5-7B-Instruct`), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) (`mistralai/Mistral-7B-Instruct-v0.3`), Yi-1.5-9B-Chat (Young et al., 2024) (`01-ai/Yi-1.5-9B-Chat`), Llama-3.2-3B-Instruct (Grattafiori et al., 2024) (`meta-llama/Llama-3.2-3B-Instruct`), SmolLM2-1.7B-Instruct (Ben Allal et al., 2025) (`HuggingFaceTB/SmolLM2-1.7B-Instruct`), and Gemma-3-4B-IT (Gemma Team, 2025) (`google/gemma-3-4b-it`). Some checkpoints are gated by provider license terms (notably Llama 3.2), while the others are directly downloadable under their respective community licenses.

Random seeds. Primary seed: 42. Cross-seed validation: 59, 76, 123, 456. All LoRA training, data splitting, and bootstrap resampling use explicit seeds for reproducibility.

Data preprocessing. Continuations generated with greedy decoding (temperature=0, `do_sample=False`), max 64 new tokens. Log-probabilities computed via teacher-forced forward pass: tokenize prompt + continuation separately, score continuation tokens at positions following the prompt.

Scoring. The sequence-likelihood gap uses absolute mean per-token log-probability difference: $|\bar{\ell}_{\text{ref}} - \bar{\ell}_{\text{cand}}|$. Per-token log-probabilities are computed using `torch.log_softmax` on the logits at each position. For cross-tokenizer model pairs, only top-k token overlap is used (cosine similarity is undefined across different vocabulary sizes).

Statistical tests. Bootstrap CIs: 2,000 resamples of model variants (not prompts), percentile method for 95% intervals. TOST: two one-sided z-tests with margin $\delta = 0.10$. Permutation tests: 10,000 random permutations. Wilson CIs: standard Wilson score interval at 95%.

Code availability. Full experiment code, result JSONs, benchmark data, and execution scripts are included in the supplementary material.