HOW TO CATCH DISTILLATION

Anonymous authors
Paper under double-blind review

ABSTRACT

Knowledge distillation transfers capabilities between language models, but determining whether a student was trained from a particular teacher remains difficult. We study whether a fingerprint embedded in a teacher can survive this transfer and reveal the training relationship. Our method associates naturally occurring teacher features with fixed patterns of token-level log-probability shifts and embeds these patterns through LoRA fine-tuning, with a distribution-preservation objective that limits changes to the teacher's behavior. A decoder queries a suspect model for output log probabilities on feature-active and feature-inactive contexts, measures alignment with the fingerprint, and compares that alignment with randomly generated fingerprints. The method requires access to the teacher during fingerprint construction but does not require the suspect model's weights or hidden states. Across the evaluated teacher–student pairs, fingerprints remain detectable after both logit-based and text-based distillation, including transfer across model families. Embedding increases teacher perplexity by 2.79% in the Qwen2.5-32B experiment. Additional evaluations examine a larger pool of 100,000 random fingerprints, 27 negative-control models, and continued student fine-tuning on clean text. In the tested fine-tuning configuration, detection persists after 10 million additional training tokens. These results support feature-conditioned fingerprinting as a practical direction for tracing language model distillation when the teacher can be modified before its outputs are used for training.

1 INTRODUCTION

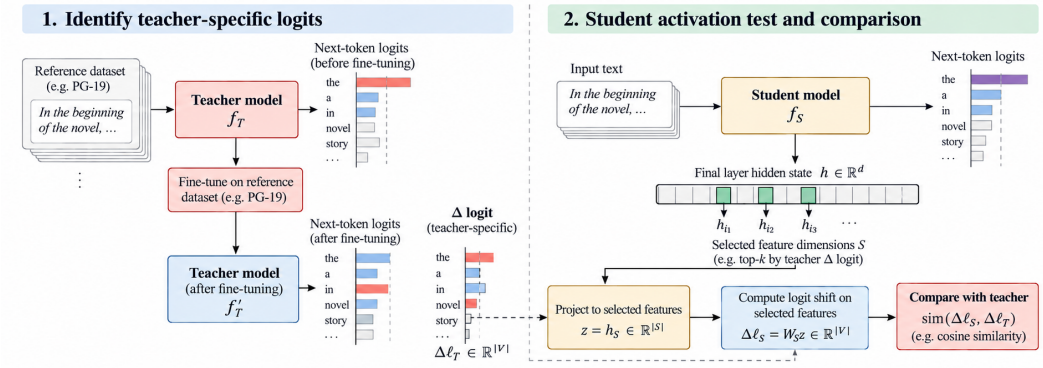


Figure 1: Overall pipeline. (1) Given a reference dataset (e.g. PG-19), we identify teacher-specific logits by comparing the teacher model's next-token logits before and after fine-tuning. The resulting logit difference highlights tokens whose probabilities change due to fine-tuning. (2) For a student model, we extract the final-layer output probabilities, project onto a selected set of feature dimensions S (e.g. those most associated with the teacher's logit changes), and compute the induced logit shift difference. We then compare the teacher and the student logit shit difference (e.g. using cosine similarity) to detect whether the teacher-specific feature is present in the student

Knowledge distillation enables a student model to acquire capabilities from a more powerful teacher, either through access to its output probabilities or simply by training on teacher-generated text. As frontier models increasingly serve as teachers for synthetic data generation, this creates a difficult

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

provenance problem: given a suspect model, can we determine whether it was distilled from a particular teacher? Recent concerns surrounding unauthorized distillation of frontier models make this question increasingly relevant, yet detecting distillation after training remains challenging because the original teacher outputs may no longer be available.

Existing watermarking approaches primarily focus on identifying model-generated text (Kirchenbauer et al., 2023a). In our setting, however, the teacher's generations are used to train another model. We therefore ask whether a fingerprint can persist beyond the generated text itself: *can a hidden signature embedded in a teacher transfer through distillation and remain detectable in the student?*

We introduce a method for embedding *transmissible fingerprints* into language models. We identify internal teacher features that activate reliably across natural contexts and associate each feature with a secret, keyed pattern of small log-probability shifts over selected vocabulary tokens. We fine-tune the teacher to express these shifts whenever the corresponding feature activates, while regularizing its output distribution toward the original model to preserve ordinary behavior. Importantly, the student is never explicitly trained on the fingerprint. It encounters only the behavior of the fingerprinted teacher during distillation.

To determine whether a suspect model inherited the fingerprint, we measure its output behavior on held-out contexts where the corresponding teacher features are active and inactive. We then test whether these feature-conditioned differences align with the secret fingerprint significantly more strongly than with randomly generated keys. Across multiple teacher–student pairs, we find that these fingerprints survive both logit-based and text-based distillation, including text-only distillation across model families. We further study how detection strength varies with the amount of distillation data and evaluate the method against undistilled and unrelated control models.

Our contributions are:

- We introduce *transmissible model fingerprints*: hidden, keyed behavioral signatures designed to be inherited through knowledge distillation.
- We develop an encoding and statistical decoding procedure that enables attribution of a distilled student to a fingerprinted teacher.
- We demonstrate fingerprint survival across multiple teacher–student pairs under both logit-based and text-only distillation, including cross-family transfer.
- We characterize how fingerprint detectability changes with distillation scale while evaluating false positives against clean and unrelated models.

2 Related Work

Watermarking language model outputs. A large body of work studies watermarking as a mechanism for identifying text generated by language models. A common approach modifies the model's sampling distribution using a secret key, creating statistical patterns in generated tokens that can later be detected without access to the model itself (Kirchenbauer et al., 2023a; Kuditipudi et al., 2023; Christ et al., 2024). Subsequent work has studied the reliability of such watermarks under paraphrasing, text modification, and mixing with unwatermarked content (Kirchenbauer et al., 2023b; Krishna et al., 2023; Sadasivan et al., 2023), as well as their robustness to adversaries attempting to infer, remove, or spoof the watermark (Jovanović et al., 2024; Zhang et al., 2024). More recently, SynthID-Text demonstrated that generative watermarking can be deployed at production scale while maintaining generation quality and low inference overhead (Dathathri et al., 2024). These methods primarily aim to identify generated text. In contrast, our goal is to identify whether a *model* has been trained using information obtained from a particular teacher.

Watermark inheritance through training. Our setting is closely related to work on *radioactive* data, which asks whether deliberately introduced signals can survive the training process and subsequently be detected in a trained model. Radioactive Data introduced this idea for image classification by modifying training examples such that models trained on them inherit a statistically detectable signal (Sablayrolles et al., 2020). More recently, Watermarking Makes Language Models Radioactive showed that statistical patterns introduced by text watermarking can remain detectable

after watermarked text is used to fine-tune another language model (Sander et al., 2024). Follow-up work has explicitly investigated this phenomenon in knowledge distillation, showing both that decoding-time LLM watermarks can transfer to student models and that adversarial procedures such as paraphrasing or watermark neutralization can substantially weaken this inherited signal (Gu et al., 2024). Related work has also considered spoofing and scrubbing attacks against watermark radioactivity in unauthorized knowledge-distillation settings (Yi et al., 2025).

Our approach differs in where the transferable signal is introduced. Rather than watermarking teacher generations through a modified decoding procedure and subsequently searching for residual watermark statistics in the student, we directly modify the teacher model. We associate naturally occurring internal teacher features with secret keyed shifts in the output distribution and optimize the teacher to express these shifts whenever the corresponding features activate. Distillation then provides the transmission channel through which this feature-conditioned behavior may be learned by the student. This allows us to study whether a fingerprint embedded in the teacher's behavior itself can provide evidence of teacher–student provenance.

Model fingerprinting and ownership verification. A separate line of work embeds fingerprints directly into neural networks for ownership verification (Uchida et al., 2017). Early approaches encode secret information in model parameters or introduce trigger–response behaviors that can later be queried to establish ownership (Uchida et al., 2017; Adi et al., 2018). In the LLM setting, Instructional Fingerprinting embeds secret instruction–response pairs through lightweight fine-tuning, allowing a model publisher to verify a fingerprint using black-box queries (Xu et al., 2024). Subsequent approaches have explored fingerprints based on under-trained tokens, memorized natural-language examples, implicit semantic fingerprints, and probabilistic model behaviors (Cai et al., 2024; Suzuki et al., 2025; Wu et al., 2025; Xu et al., 2025). These methods primarily address whether a deployed model is a copy or modified version of a fingerprinted model.

Our objective is different: we do not require the suspect model to preserve a particular trigger–response mapping. Instead, we ask whether a model trained *from the outputs* of a fingerprinted teacher inherits enough of its feature-conditioned behavior to reveal the training relationship. The fingerprint therefore serves as evidence of knowledge transfer rather than simply persistence of an ownership marker under parameter modification.

Training-data provenance and model extraction. Our work also relates to methods for determining whether particular data or knowledge contributed to model training. Dataset inference tests whether knowledge associated with a protected dataset is present in a potentially stolen model, motivated by the observation that training-set information may survive model extraction even when explicit model watermarks do not (Maini et al., 2021; 2024). Membership inference and related data-attribution methods similarly seek evidence that particular examples were used during training (Shokri et al., 2017; Shi et al., 2024). Our setting instead considers provenance between *models*: rather than asking whether a known example or dataset was used for training, we embed a secret statistical signature into a teacher and test whether that signature is inherited by a student trained from the teacher's outputs. Concurrent work has also approached teacher attribution without a planted signal, tracing a student to its teacher from naturally occurring stylistic footprints left by distillation (Wadhwa et al., 2025); our fingerprint is instead deliberately embedded and keyed, trading the ability to audit models after the fact for a much stronger, statistically calibrated attribution guarantee.

3 METHOD

Our method has encoding and decoding steps. The encoding step consists of top neuron extraction, fingerprint creation, logit shift computation, and fine-tuning to embed the logit shift onto the teacher model. On the other hand, the decoding step consists of checking if the model being tested is similar to the teacher model in a statistically significant way.

Table 1: Example: neuron activation 1 → fingerprint 1 (0x7c19f...) → [+1,-1,-1,..., -1]

Token	Token ID	Value
1. House	4172	$+1$
2. Voice	7891	-1
3. King	1203	-1
⋮	⋮	⋮
256. Book	7532	-1

3.1 ENCODING

3.1.1 TOP K NEURON EXTRACTION

For this part of the method, our goal is to extract the top-k neurons to imprint hidden signatures in them. In other words, we want to find the k neurons that most frequently fire, especially during distillation. We had to ensure that the choice of neurons would be specific such that we don't end up picking neurons that fire too commonly (this might add too many hidden signatures to detect, e.g., a neuron that fires for the token "the") or too rarely (this will make it very hard to detect the hidden signatures and might even result in the neuron not firing during distillation).

To ensure we satisfy the above criteria, we pick the top k neurons by:

1. Eliminating all neurons that have an activation rate greater than 50%.
2. Eliminating all neurons that have an activation rate of less than 1%.
3. Arranging them in non-increasing order ranked by:

$$\text{Activation Rate} \times \text{Mean Activation}$$

   The product favors neurons that are both frequently active enough to provide many fingerprinting opportunities and strongly active when they fire. Thus, it balances signal frequency and magnitude rather than selecting neurons that score highly on only one dimension.

4. Picking the top k neurons from the arranged list.

3.1.2 FINGERPRINT CREATION

Now that we have elected the top k neurons, we need k equivalent fingerprints that each uniquely map to a change in the log prob of certain tokens in the last layer. By "certain" we mean 256 randomly chosen tokens from the teacher's vocab size. Once we choose these 256 tokens, we use this same set throughout the rest of the process. To create the fingerprints, we use the SHA-256 algorithm with the neuron's mean activation as an input. For each of these neurons, we then use the SHA-256 fingerprint to trigger a log prob shift in the 256 chosen tokens. The advantage of SHA-256 is the fact that it is unique, so for a given input, SHA-256 produces one unique hash, functioning as an effective helper for our fingerprint creation.

3.1.3 LOG PROB SHIFT

Each fingerprint is a 256-bit hash. Since the fingerprint was created with SHA-256 on the mean neuron activation such that the combination of the 256 bits is unique to that neuron, we effectively map the given neuron into a unique vector by mapping the 256-bit hash into a vector of size 256 with values $\{+1,-1\}$. For an example of how this process works, refer to Table 1.

3.1.4 LORA FINE-TUNING

To achieve the above log prob shift in the teacher model, we perform LoRA fine-tuning against a copy of the teacher while using a frozen copy for reference. We use a loss that has two main components:

- The first component is a KL divergence that we use to penalise checkpoints that stray from the original model too much. For a training batch with $N = B \times T$ positions:

$$\mathcal{L}_{preserve} = \frac{1}{N} \sum_{x} D_{KL}(p_0(\cdot|x) \| p_\theta(\cdot|x))$$

  We apply this loss at every token position, where x is the context up until that token, and $\cdot$ represents all possible next tokens. We use this loss term to keep the model from losing capabilities by having a probability distribution similar to the pre-finetuning checkpoint.

- The second loss component is a mean squared error loss with respect to the fingerprint we are looking to create. In other words, for each feature i, we retrieve the positions A_i where i is active in the frozen original model, and we retrieve i's 256 signature tokens V_i. We also let $s_{i,v} \in \{-1, +1\}$ be the keyed sign for each token $v \in V_i$. Then, we calculate the difference $\log p_\theta(v|x) - \log p_0(v|x)$ for each $v \in V_i, x \in A_i$, and we calculate the error between that difference and the desired difference $\epsilon s_{i,v}$, where ϵ is the desired log-probability displacement (you can think of it as a hyperparameter that can be played around with). As we mentioned earlier, this part of the loss is a mean squared error, which means we take all errors $(\log p_\theta(v|x) - \log p_0(v|x)) - \epsilon s_{i,v}$, we square them and get the mean of these squares. The full equation looks like:

$$\mathcal{L}_{fp} = \frac{\sum_i |A_i| \left[\frac{1}{|A_i||V_i|} \sum_{x \in A_i} \sum_{v \in V_i} (\log p_\theta(v|x) - \log p_0(v|x) - \epsilon s_{i,v})^2 \right]}{\sum_i |A_i|}$$

  After canceling the duplicated $|A_i|$, this is equivalent to:

$$\mathcal{L}_{fp} = \frac{\sum_i \frac{1}{|V_i|} \sum_{x \in A_i} \sum_{v \in V_i} (\Delta \log(v|x) - \epsilon s_{i,v})^2}{\sum_i |A_i|}$$

  where $\Delta \log(v|x) = \log p_\theta(v|x) - \log p_0(v|x)$

Given these two loss terms, the full loss of the fine-tuning process is then

$$\mathcal{L} = \mathcal{L}_{preserve} + \lambda_{fp} \mathcal{L}_{fp}$$

where λ_{fp} is a hyperparameter that assigns that weight/importance of $\mathcal{L}_{fp}$. For our experiments we have set λ_{fp} to 4, but a hyperparameter search would find optimal values for any given model. Our LoRA configuration is the following: rank 16, alpha 32, and zero dropout, applied to all attention projections and the three MLP projections. For a figure showing the training trajectory of both losses, refer to Appendix A.1.

3.2 Decoding

To check whether a suspect model has the distillation fingerprint, we query it on held-out contexts and test whether the LLM's output distribution shows the same feature-conditioned keyed log probability shifts that were put inside the teacher. The decoding process consists of the following high-level steps:

- We construct a dataset with contexts where the top teacher features are active, and contexts where these features are inactive.
- We measure how close the suspect model is to the teacher signature in this subset.
- We compare that alignment against signatures that could arise from random fake keys.

For every feature i, we first use the frozen teacher model p_0 to come up with a set H_i^+ of contexts where i is active. Then, we construct an equally-sized control set H_i^- where none of the features is active. We use these probe sets to evaluate the suspect model, and they are kept fixed throughout the testing process.

Let $p_M(v|x)$ be the next-token distribution of the model to be tested, M. For every candidate token v, that is, for every token that could be a signature token from the fingerprint, the decoder calculates

$$d_i^M(v) = \frac{1}{|H_i^+|} \sum_{x \in H_i^+} \log p_M(v|x) - \frac{1}{|H_i^-|} \sum_{x \in H_i^-} \log p_M(v|x)$$

In other words, the decoder calculates the difference in model tendency to use the token when the feature is active versus when the feature is inactive. If the suspect model has the fingerprint, then for each feature's signature tokens $v \in V_i$ we would expect $d_i^M(v)$ to be equal to $s_{i,v}$ the majority of the time. Based on this notion, for a given fingerprint K, we define the fingerprint score to be

$$R_i^M(K) = \frac{1}{|V_i|} \sum_{v \in V_i} s_{i,v} d_i^M(v)$$

A positive value means that the suspect's feature-conditioned output changes are aligned with the fingerprint: a positive value means the majority of the tokens have $s_{i,v} = sign(d_i^M(v))$. In practice, this setup can be confounded by noise. We can use the original (non-fingerprinted) teacher model as a reference to account for this:

$$\tilde{d}_i^M(v) = d_i^M(v) - d_i^0(v)$$

In other words, we get the difference in model tendency to use the token when the feature is active versus inactive, and subtract this from the non-fingerprinted teacher difference. Then, the score for each feature is

$$\tilde{R}_i^M(K) = \frac{1}{|V_i|} \sum_{v \in V_i} s_{i,v} \tilde{d}_i^M(v)$$

The feature scores are not enough to allow a confident verdict on whether the suspect model has been distilled from our teacher model. Because of this, we compare the scores we computed to scores collected from fake, random signatures, to see if the suspect model's alignment to the fingerprint can be due to distillation rather than the null hypothesis. For this purpose, the decoder generates 1000 random fingerprints. For all such keys K_r, the score is calculated in the same way as with the original fingerprint:

$$\tilde{R}_i^M(K_r) = \frac{1}{|V_i^{(r)}|} \sum_{v \in V_i^{(r)}} s_{i,v}^{(r)} \tilde{d}_i^M(v)$$

where $V_i^{(r)}$ and $s_{i,v}^{(r)}$ are the vocabulary set and the sign for each feature i, fingerprint K_r. Based on the scores for this 1000-key random distribution, and the scores for the true key, we estimate the mean and standard deviation of the random-key null distribution and standardize the true-key score. We then compute a one-sided Gaussian-tail p-value to check whether the null hypothesis can be discarded, i.e., whether we can say confidently that the model was distilled and our results are not an effect of random noise. Whenever $p < 0.01$, we take the model to have been distilled by the teacher. We pick this extremely small p-value to ensure the student is definitely distilled and our method does not accuse models that showed borderline statistical significance because of random noise.

4 Results

4.1 Method Performance

We tested our distillation detection method across different model-student pairs from the Qwen, Llama, Deepseek, and Yi families. To do this, we trained various teacher models on the fingerprinted/desired logits. Then, we used these teachers to distill multiple models, with both hard-label and logit-based distillation. The results can be found in Table 3. We report the student's p-values, as well as the minimum p-value out of a control set including the base student, the base teacher, and unrelated models. We also report detailed results for the Qwen-2.5-32B $\rightarrow$ Qwen-2.5-7B family in Table 2, including controls for unrelated families and different teachers, suggesting our mechanism detects unique distillation signatures rather than any distillation. Lastly, we report results suggesting that our method translates to other datasets (Table 6).

4.2 Statistical Calibration of the Decoder

The decision rule in Section 3.2 calibrates each fingerprint score against $R = 1000$ randomly generated fake keys and reads off a one-sided Gaussian-tail p-value; Table 3 reports p-values as small as 6.8×10^{-10}. Because this extrapolates far into the tail of a null distribution estimated from only 1000 draws, we stress-tested the calibration procedure itself rather than any single detection result.

Table 2: Fingerprint detection for the Qwen2.5-32B → Qwen2.5-7B family. We report the one-sided decoder p-value and its reference-subtracted variant $p_{\text{ref-sub}}$ (clean-teacher signatures removed). Smaller p is stronger evidence that the fingerprint was inherited; the verdict threshold is $p < 0.01$. The teacher perplexity cost of embedding was $+2.79\%$. The last control is a model that was itself produced by distillation but from a *different* teacher, testing that the detector responds to our fingerprint and not to distillation in general.

Provenance	p	$p_{\text{ref-sub}}$	Verdict
fingerprinted teacher	2.4×10^{-22}	$< 10^{-16}$	Distilled
distilled from fp. teacher	0.002	6.8×10^{-10}	Distilled
distilled from clean teacher	0.356	0.879	Not distilled
student base, undistilled	0.278	0.705	Not distilled
original clean teacher	0.251	–	Not distilled
gpt2 (unrelated family)	0.516	0.805	Not distilled
distilled from Llama-3.1-70B (other teacher)	0.187	0.326	Not distilled

Table 3: Fingerprint survival across teacher→student families. Each row is one distillation experiment. We report the reference-subtracted decoder p-value for the distilled student (the result), and the *smallest* p over all negative controls (undistilled base, clean-teacher student, and an unrelated model); a large control p means no false positive. Detection threshold $p < 0.01$.

Teacher	Student	KD	p_{student}	$p^{\min}_{\text{control}}$	Detected
Same family (logit distillation)					
Qwen2.5-32B	Qwen2.5-7B	logit	6.8×10^{-10}	0.70	✓
Qwen2.5-7B	Qwen2.5-1.5B	logit	9.2×10^{-7}	0.20	✓
Llama-3.1-70B	Llama-3.2-3B	logit	9.0×10^{-3}	0.21	✓
Cross family (text distillation)					
Qwen2.5-7B	Qwen3.5-9B	text	1.0×10^{-3}	0.17	✓
Qwen2.5-7B	Llama-3.2-3B	text	4.0×10^{-3}	0.43	✓

We redrew a shared null distribution with $R = 100{,}000$ random keys ($100\times$ larger, and shared across every suspect scored against it, rather than redrawn per suspect) and checked two things. First, whether the empirical null quantiles agree with the assumed Gaussian out to the extreme tail: a quantile–quantile comparison matches the two distributions to visual precision out to ± 3.3 standard deviations. Second, the decoder's empirical false-positive rate over $N = 27$ models that never interacted with the fingerprinted teacher, spanning an unrelated pretrained zoo (GPT-2, GPT-2-XL, Pythia-1.4B, OPT-1.3B, StableLM-2, Phi-2, SmolLM2, TinyLlama, DeepSeek-LLM-7B, Falcon3-7B, Yi-1.5-6B, Mistral-7B) together with same-family Qwen checkpoints of different sizes and instruction tuning, an undistilled base student, a student distilled from the *clean* (non-fingerprinted) teacher, and the clean teacher itself. Table 4 reports the result: at the $p < 0.01$ decision threshold used throughout this paper, and at the stricter $p < 0.001$, the observed false-positive rate is exactly $0/27$; at the looser $p < 0.05$ it is $1/27$ (3.7%), close to the nominal rate. This indicates the small p-values reported elsewhere in this paper reflect a well-calibrated null rather than an artifact of under-sampling it.

4.3 Robustness to Post-Distillation Fine-tuning

A distiller may fine-tune the resulting student further before deployment, whether to adapt it to a downstream task or, adversarially, to try to erase structure inherited from the teacher. We test whether the fingerprint survives such *laundering* by taking the fingerprinted Qwen2.5-7B→Qwen2.5-1.5B student of Table 3 and continuing to train it with ordinary cross-entropy on clean held-out text it never saw during distillation, at increasing token budgets, then re-running the decoder at each point. Figure 3 reports the reference-subtracted decoder p-value immediately after distillation and after each subsequent fine-tuning budget. Detection remains solidly significant throughout: the p-value rises only modestly, from 2.8×10^{-4} immediately after distillation to

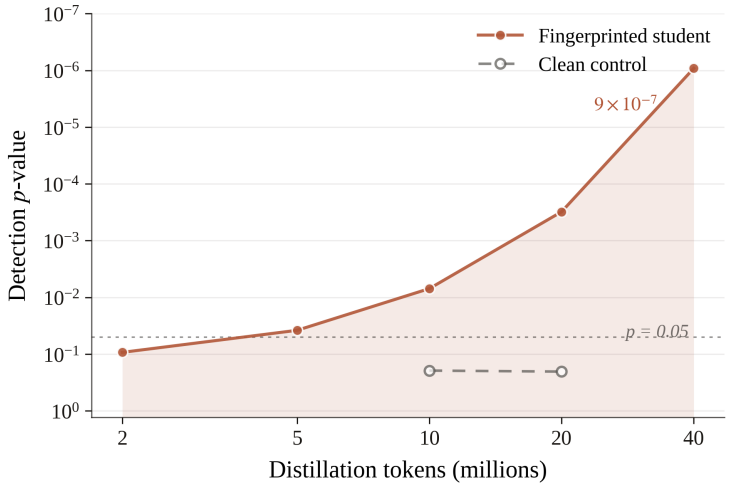


Figure 2: We show the relationship between the number of distillation tokens and detection mechanism performance, with the teacher and student models fixed. The X axis is the number of distillation tokens, and the Y axis is the p-value for our prediction of the student model being distilled on the teacher model.

Table 4: Empirical false-positive rate of the decoder's combined statistic, recalibrated against 100,000 random null keys (versus 1,000 in the original decision rule) and evaluated on 27 models that never touched the fingerprinted teacher.

Nominal threshold α	Expected FPR	Observed FPR (N=27)
0.05	5%	3.7% (1/27)
0.01	1%	0.0% (0/27)
0.001	0.1%	0.0% (0/27)

5.6×10^{-4} at 1M, 7.9×10^{-4} at 5M, and 1.1×10^{-3} at 10M tokens of further clean fine-tuning, never approaching the $p < 0.01$ threshold. We did not measure the model-quality cost of this additional fine-tuning, since it is applied by a hypothetical downstream party rather than by us; our claim is only that the fingerprint persists as a statistical signal in the student's output distribution under an obvious and cheap laundering attempt, not that laundering is itself without cost.

We additionally probe how detectability responds when only a fraction of the student's training corpus originates from the fingerprinted teacher, mixing fingerprinted- and clean-teacher text at controlled ratios (0%, 10%, 50%, 100%) under sequence-level text distillation. At the constrained token budget a preliminary run permitted (∼1.4M tokens per ratio, well below the tens of millions used in Figure 2), the detection statistic increases monotonically with the fingerprinted fraction but does not yet cross the $p < 0.01$ threshold at any ratio tested; given the scaling behavior in Figure 2, we expect these ratios to reach significance once more distillation tokens are used, and we leave a full-scale version of this experiment to future work (Appendix A.3).

5 Discussion and Limitations

Our results suggest that provenance can be engineered into a teacher's behavior at training time and recovered from a student with only black-box query access, surviving both logit- and text-based distillation, cross-family transfer, and a naive laundering attempt. We see this as a shift from *post-hoc* provenance methods, which try to infer a training relationship after the fact, to a *proactive* one: a model owner who suspects their outputs will be scraped for distillation can decide, in advance, to make that theft detectable.

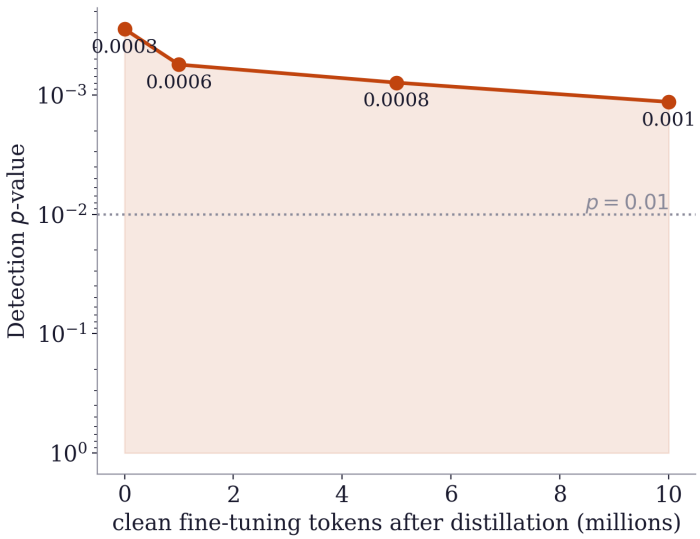


Figure 3: Detection p-value for the fingerprinted Qwen2.5-7B→Qwen2.5-1.5B student as it undergoes further clean fine-tuning after distillation (a laundering attack), at increasing token budgets. The dashed line marks the $p = 0.01$ decision threshold used throughout this paper.

This proactivity is also the method's main limitation. The fingerprint must be embedded by whoever controls the teacher, before distillation happens; it cannot be applied retroactively to a model that has already been released or to a teacher the fingerprinter does not control. It is therefore a tool for model owners protecting their own models, not a general-purpose forensic test for an arbitrary suspect–teacher pair.

Several further limitations bound the scope of our claims:

- **Embedding is not free.** Fine-tuning the teacher to express the fingerprint measurably shifts its output distribution, even though we regularize this cost to be small (perplexity increase under 3% in our experiments). A more aggressive fingerprint (larger ϵ) would be easier to detect after heavier laundering, at a cost we have not fully characterized as a Pareto curve.

- **Exposure requires distillation volume.** Our preliminary partial-distillation experiment (Appendix A.3) suggests detection needs a nontrivial amount of the student's training data to originate from the fingerprinted teacher; a distiller who mixes in only a small fraction of fingerprinted-teacher outputs, or trains on very few tokens overall, may evade detection at our preregistered significance threshold.

- **We tested a natural but non-adaptive laundering attack.** Continued clean fine-tuning did not remove the fingerprint (Section 4.3), but we did not evaluate an adversary who knows the mechanism and specifically optimizes against it, e.g. by regularizing the student away from teacher-specific feature-conditioned quirks, or by distilling through an intermediate paraphrasing or re-sampling step.paraphrasing or re-sampling step.

- **Decoding is not fully black-box on the owner's side.** While the suspect model is queried black-box, the decoder needs the original teacher (or its cached probe set and clean-teacher reference statistics) to build feature-active contexts and to reference-subtract family-shared structure. An owner who has not preserved these artifacts cannot decode later.

- **Coverage of teacher/student scale.** Our 100,000-key calibration stress test (Section 4.2) was run at the 7B-teacher scale; we have not repeated it at the 32B/70B scale, and two of our cross-family pairs (Yi-1.5-34B, DeepSeek-67B) were still in progress at submission time.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing, organization, and LaTeX formatting. We reviewed the AI-assisted revisions for accuracy, consistency with the reported methods and results, and fidelity to our intended claims. The authors take responsibility for the final manuscript, including all text, claims, citations, and reported findings.

ETHICS STATEMENT

Model provenance methods can support auditing, but incorrect attribution can harm model developers and users. Our detector measures evidence of an embedded fingerprint under specified experimental conditions. It does not determine whether training was authorized or establish misconduct. Positive detections should be corroborated before consequential action, and negative detections should not be treated as proof that a model was not distilled. Fingerprint embedding changes the teacher's output distribution, so deployment also requires evaluating effects on model utility.

REPRODUCIBILITY STATEMENT

Section 3 describes feature selection, fingerprint construction, the embedding objective, and the decoding procedure. Section 4 reports the evaluated teacher–student pairs, control-model results, and additional detection analyses. Appendix A.1 presents fingerprint-embedding dynamics, Appendix A.2 describes the negative controls, and Appendices A.3 and A.4 provide additional results on partial exposure and student–teacher size ratios. We distinguish completed evaluations from preliminary findings and state the access assumptions and limitations relevant to reproducing and interpreting the results.

REFERENCES

Yossi Adi, Carsten Baum, Moustapha Cisse, Benny Pinkas, and Joseph Keshet. Turning your weakness into a strength: Watermarking deep neural networks by backdooring. In *27th USENIX Security Symposium (USENIX Security 18)*, pp. 1615–1631, 2018. arXiv:1802.04633.

Jiacheng Cai, Jiahao Yu, Yangguang Shao, and Yuhang Wu. UTF: Under-trained tokens as fingerprints — a novel approach to LLM identification. *arXiv preprint arXiv:2410.12318*, 2024.

Miranda Christ, Sam Gunn, and Or Zamir. Undetectable watermarks for language models. In *Proceedings of the 37th Conference on Learning Theory (COLT)*, 2024.

Sumanth Dathathri, Abigail See, Sumedh Ghaisas, et al. Scalable watermarking for identifying large language model outputs. *Nature*, 634:818–823, 2024. doi: 10.1038/s41586-024-08025-4.

Chenchen Gu, Xiang Lisa Li, Percy Liang, and Tatsunori Hashimoto. On the learnability of watermarks for language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Nikola Jovanović, Robin Staab, and Martin Vechev. Watermark stealing in large language models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proceedings of the 40th International Conference on Machine Learning (ICML)*, volume 202, pp. 17061–17084, 2023a.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Manli Shu, Khalid Saifullah, Kezhi Kong, Kasun Fernando, Aniruddha Saha, Micah Goldblum, and Tom Goldstein. On the reliability of watermarks for large language models. *arXiv preprint arXiv:2306.04634*, 2023b.

Kalpesh Krishna, Yixiao Song, Marzena Karpinska, John Wieting, and Mohit Iyyer. Paraphrasing evades detectors of AI-generated text, but retrieval is an effective defense. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Rohith Kuditipudi, John Thickstun, Tatsunori Hashimoto, and Percy Liang. Robust distortion-free watermarks for language models. *arXiv preprint arXiv:2307.15593*, 2023.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. In *International Conference on Learning Representations (ICLR)*, 2021.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. LLM dataset inference: Did you train on my dataset? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: tracing through training. In *International Conference on Machine Learning (ICML)*, 2020.

Vinu Sankar Sadasivan, Aounon Kumar, Sriram Balasubramanian, Wenxiao Wang, and Soheil Feizi. Can AI-generated text be reliably detected? *arXiv preprint arXiv:2303.11156*, 2023.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Weijia Shi, Anirudh Ajith, Mengzhou Xia, Yangsibo Huang, Daogao Liu, Terra Blevins, Danqi Chen, and Luke Zettlemoyer. Detecting pretraining data from large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. *IEEE Symposium on Security and Privacy (S&P)*, 2017.

Teppei Suzuki, Ryokan Ri, and Sho Takase. Natural fingerprints of large language models. *arXiv preprint arXiv:2504.14871*, 2025.

Yusuke Uchida, Yuki Nagai, Shigeyuki Sakazawa, and Shin'ichi Satoh. Embedding watermarks into deep neural networks. In *Proceedings of the 2017 ACM on International Conference on Multimedia Retrieval*, pp. 269–277, 2017. arXiv:1701.04082.

Somin Wadhwa, Chantal Shaib, Silvio Amir, and Byron C. Wallace. Who taught you that? tracing teachers in model distillation. *arXiv preprint arXiv:2502.06659*, 2025.

Jiaxuan Wu, Wanli Peng, Hang Fu, Yiming Xue, and Juan Wen. ImF: Implicit fingerprint for large language models. *arXiv preprint arXiv:2503.21805*, 2025.

Jiashu Xu, Fei Wang, Mingyu Derek Ma, Pang Wei Koh, Chaowei Xiao, and Muhao 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)*, pp. 3277–3306, 2024. arXiv:2401.12255.

Zhenhua Xu, Meng Han, and Wenpeng Xing. Evertracer: Hunting stolen large language models via stealthy and robust probabilistic fingerprint. *arXiv preprint arXiv:2509.03058*, 2025.

Xin Yi, Yue Li, Shunfan Zheng, Linlin Wang, Xiaoling Wang, and Liang He. Unified attacks to large language model watermarks: spoofing and scrubbing in unauthorized knowledge distillation. *arXiv preprint arXiv:2504.17480*, 2025.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for generative models. In *Proceedings of the 41st International Conference on Machine Learning (ICML)*, 2024.

A Appendix

You may include other additional sections here.

A.1 Training Dynamics of Fingerprint Embedding

Figure 4 shows the two loss terms of Section 3.1.4 over the course of embedding the fingerprint into the Qwen2.5-7B teacher ($\epsilon = 0.75$). The mean signed shift $s_{i,v}\Delta \log(v \mid x)$, averaged over active positions and signature tokens, climbs from 0 toward the target ϵ within the first $\sim$1000 steps and then plateaus below it rather than fully saturating; in practice we found this partial saturation already sufficient for reliable detection, and pushing further increases $\mathcal{L}_{fp}$'s pressure on the output distribution for little additional decoder signal. Over the same steps, $\mathcal{L}_{preserve}$ (the KL divergence to the original, non-fingerprinted teacher) rises off its zero initialization but stays within a narrow, non-diverging band for the remainder of training, consistent with the small measured perplexity cost of embedding.

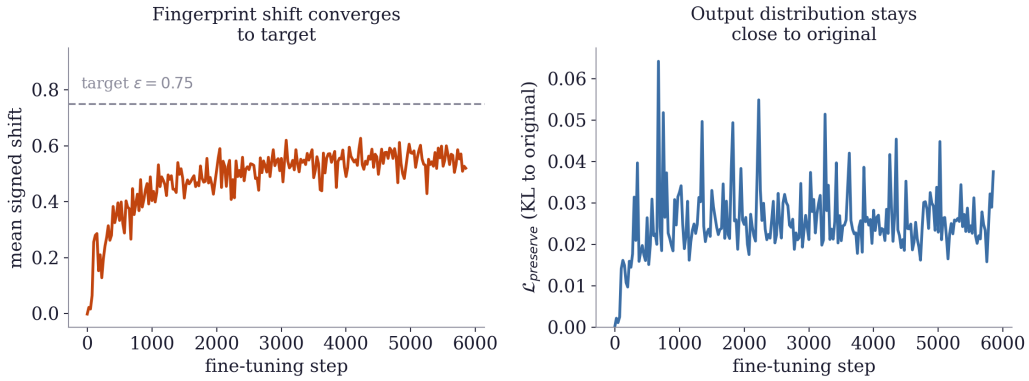


Figure 4: Training dynamics while embedding the fingerprint into the Qwen2.5-7B teacher. *Left:* the mean signed shift on active signature tokens converges toward the target displacement ϵ. *Right:* the KL-divergence preservation term stays small and non-diverging throughout training.

A.2 Calibration Details

The 27 models used as known negatives in Section 4.2 are: an unrelated pretrained zoo of 18 models spanning multiple architectures and tokenizers (GPT-2, GPT-2-XL, Pythia-1.4B, OPT-1.3B, StableLM-2-1.6B, Phi-2, SmolLM2-1.7B, TinyLlama-1.1B, DeepSeek-LLM-7B-base, Falcon3-7B-Base, Yi-1.5-6B, Mistral-7B-v0.1, and same-family Qwen checkpoints Qwen1.5-1.8B, Qwen2-1.5B, Qwen2.5-0.5B, Qwen2.5-1.5B-Instruct, Qwen2.5-3B, and Qwen2.5-14B-Instruct); together with 9 same-lineage but non-fingerprinted controls (the undistilled base student, a student distilled from the clean teacher, the clean teacher itself, a student fingerprinted only through the non-transmissible hidden-direction ablation carrier, and clean-teacher text-distilled students at two checkpoints each for two additional student architectures). None of these models were exposed to the fingerprinted teacher's outputs. All 27 are scored against the same shared pool of 100,000 random null keys used to produce Table 4 and the quantile–quantile comparison described in Section 4.2, so the reported false-positive rate reflects a single, internally consistent calibration run rather than results pooled across different null samples.

A.3 Preliminary Partial-Distillation (Mixture) Results

Table 5 reports the raw numbers behind the preliminary partial-distillation experiment described in Section 4.3. Each row trains a fresh student via sequence-level text distillation on a corpus mixing a controlled fraction of fingerprinted-teacher continuations with clean-teacher continuations (same prompts, same total corpus size, $\sim$1.4M tokens), then decodes it with the standard procedure. We report this as preliminary because the token budget here is roughly an order of magnitude smaller than the budgets in Figure 2, which by itself is known to reduce detection strength independent of the mixing ratio; we include it to show the qualitative trend (monotonically increasing with fingerprinted-teacher exposure) rather than as a claim of detection at these ratios.

Table 5: Detection statistic (reference-subtracted z) as a function of the fraction of a student's text-distillation corpus sampled from the fingerprinted teacher, at a constrained $\sim$1.4M-token budget per ratio. None of these preliminary points cross the $p < 0.01$ ($z > 2.33$) decision threshold.

% of corpus from fingerprinted teacher	z (ref-sub)
0%	1.13
10%	1.16
50%	1.22
100%	1.42

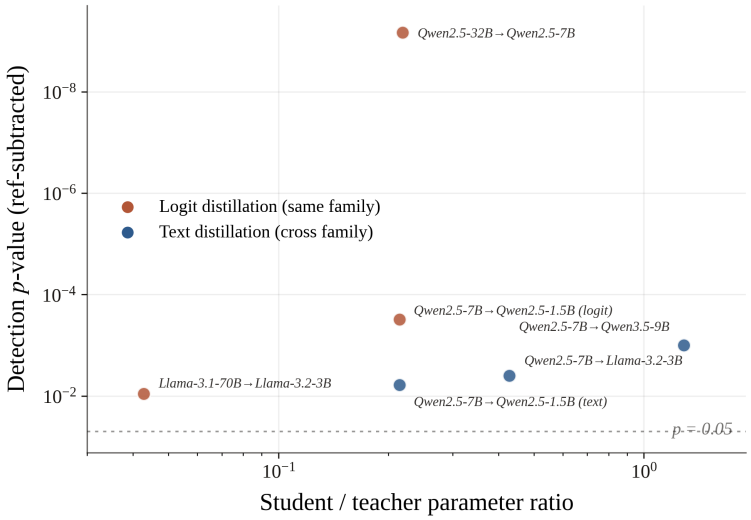


Figure 5: Reference-subtracted detection p-value against the student/teacher parameter ratio. Both axes are logarithmic and the p-axis is inverted, so higher is stronger detection; the dashed line marks $p = 0.05$. Clay markers are logit distillation (same family), slate markers are text distillation (cross family).

A.4 Detection Versus Student/Teacher Size Ratio

Figure 5 plots, for every teacher→student pair we evaluate, the compression ratio (student parameter count divided by teacher parameter count) against the reference-subtracted detection p-value of the distilled student. Points are coloured by distillation channel: logit matching, which requires a shared tokenizer and is therefore same-family, and text (hard-label) distillation, which transfers only sampled text and thus applies across tokenizers and families.

Three observations follow.

Detection survives across the full ratio range. Every operating point falls below $p = 0.05$, spanning ratios from 0.043 (Llama-3.1-70B compressed $\sim$23$\times$ into Llama-3.2-3B) to 1.29 (Qwen2.5-7B distilled into the *larger* Qwen3.5-9B). The fingerprint is thus recovered under aggressive compression and even when the student exceeds the teacher in size, indicating that transfer is not gated by a particular capacity regime.

The distillation channel, not the size ratio, sets the strength. Logit-matched students separate from text-distilled students by several orders of magnitude in p (down to $\sim 10^{-10}$ versus $\sim 10^{-3}$), consistent with hard-label text distillation being a far narrower channel: the student sees only sampled tokens rather than the teacher's full output distribution. Within each channel the size ratio has no clear monotonic effect, so the ratio is a secondary factor relative to how the student is trained.

Caveats. The points are the operating configurations we ran rather than a controlled ratio sweep, so the axis conflates three variables: the size ratio, the distillation channel, and the training-token budget (logit points use $\sim$ 20M tokens, text points $\sim$ 60M). The figure should therefore be read as

evidence that detection is robust across a wide range of compression ratios and both channels, not as an isolated measurement of the ratio's effect.

convention your Table 1 already uses for Yi/DeepSeek — and update it the moment it finishes.

A.5 A SECOND DATASET: WIKITEXT-103

All experiments in the main text use PG-19 for feature discovery, fingerprint embedding, distillation, and decoding. To check that our results are not an artifact of this one text domain, we repeat two checks on WikiText-103 (, a Wikipedia-derived corpus that is stylistically very different from PG-19's books (short, factual, non-narrative text).

Decode-time generalization (no retraining). We build a second decoder probe directly from WikiText-103 held-out text and re-score the *existing*, PG-19-embedded/distilled checkpoints from the main text against it. Table 6 shows the fingerprint is still recovered when the suspect model is queried on an entirely different text domain than the one it was fingerprinted and distilled on.

Table 6: Fingerprint detection when the decoder's probe text is WikiText-103 instead of PG-19, applied to the same PG-19-trained checkpoints from Section 4.3. No retraining; only the decoder's held-out evaluation text changes.

Suspect	z (ref-sub)	p-value	Verdict
Clean teacher (negative control)	0.27	0.395	not distilled
Fingerprinted student	2.42	0.0077	**DISTILLED**
Fingerprinted student + 10M tokens laundering	2.15	0.0158	not distilled

Training on a second dataset. We also re-run both distillation modes with WikiText-103 as the *training* corpus itself (rather than PG-19), reusing the same fingerprinted Qwen2.5-7B teacher, under a compute budget of roughly one hour.