A FALSE SENSE OF PROVENANCE: DATA LAUNDERING DEFEATS TRAINING-DATA LICENSING AUDITS

Anonymous authors
Paper under double-blind review

ABSTRACT

Training data is now bought and sold under contract. To check these contracts after the fact, researchers have built data-provenance audits, which look for a signal of the licensed data in a trained model. In this paper, we show that a licensee can train on the licensed dataset beyond what its license covers and defeat these audits with a single cheap operation, which we call data laundering. The licensee fine-tunes a single generator on the licensed dataset, regenerates the dataset from that generator, and trains each deployed model on the regenerated dataset. Any signal an audit looks for is either separable from the capability the licensee paid for or inseparable from it, and neither case attributes a laundered model to the licensed dataset. We test the attack against nine published state-of-the-art audits over images and text. Laundered models score at or near the level of models trained without the licensed data. Language models trained on the regenerated dataset keep 82% to 85% of the capability gain from training on the licensed dataset. Laundering also defeats six defenses, including two published audits designed with regeneration in mind.

1 INTRODUCTION

AI developers now pay for training data under contract. Reddit's SEC filing discloses data-licensing deals worth $203 million (Reddit, Inc., 2024), OpenAI has signed content deals with the Associated Press and News Corp (Axios, 2023; Variety, 2024), and Adobe pays the contributors whose work trains its models (Adobe, 2023). The major record labels have begun licensing their music to AI music startups (TechCrunch, 2025), and courts are being asked to establish what data a model was trained on (Authors Guild, 2025). Two new statutes (the EU AI Act and California's AB 2013) require the same provenance, but they check it through a declaration the trainer publishes (European Parliament and Council of the European Union, 2024; California State Legislature, 2024).

A license limits which models a dataset may train, for how long, and by whom. Once the dataset is in a licensee's hands, it can train a model the license does not cover, train models after the license ends, or reach a party that holds no license. Researchers have built *data-provenance audits* to check for such use after the fact. An audit looks for a signal of the licensed dataset in the trained model. We group these audits into three families, drawing on a recent survey (Du et al., 2025b). The first injects marks or radioactive data into the dataset before release (Sablayrolles et al., 2020; Wenger et al., 2024). The second infers membership from how the trained model behaves on the licensed records (Carlini et al., 2022; Duan et al., 2023). The third attributes a model to the sources its outputs most resemble (Carlini et al., 2023; Somepalli et al., 2023).

Unfortunately, these audits were validated mainly in a naive attack scenario, in which the trainer fine-tunes on the licensed dataset directly and does not expect to be checked. A licensee holds the dataset, controls each step from download to deployment, and knows it will be audited. Prior evasion attacks each remove one family of signals (Zhu et al., 2025; Gao et al., 2026; Datta et al., 2024; Li et al., 2026), and a licensor can answer with a different audit. Existing work is thus insufficient because neither the audits nor the attacks ask whether a signal that survives can still attribute a model to the licensed dataset.

In this paper, we show that a licensee can train on a licensed dataset and still defeat each published audit we test, including one the licensor prepared for. The attack needs a single cheap operation,

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.



Figure 1: Laundered models keep the licensed style and evade the audit, setting X1 (Konchalovsky paintings from WikiArt). The licensee fine-tunes a generator on the licensed dataset, regenerates a dataset from that generator, and trains its deployed model only on the regenerated dataset. CLiD (a state-of-the-art audit) scores 0.999 on direct training and 0.552 on the deployed model, close to the 0.407 of an innocent model trained without $\mathcal{D}$ (right).

which we call *data laundering*. The licensee fine-tunes a single *generator* on the licensed dataset, regenerates a dataset from that generator, and trains all deployed models on the regenerated dataset. Laundering keeps the capability the licensee paid for and removes the signals an audit could tie to the licensed dataset. Figure 1 shows the licensed style carried through regeneration into the deployed model, and absent from a model trained without the licensed dataset.

Our work is driven by two observations. First, a model trained on the regenerated dataset learns from the licensed data only through the generator, which acts as an information bottleneck. A signal that the capability does not need, such as a copyright trap or an exact copy of a licensed span, passes through only if the generator learned it. We call such a signal *separable*. Second, a signal that the capability does need, such as the brushwork the licensee is paying for, does pass through. However, it also appears in any model that reaches the same capability from other data (Zhang et al., 2025a; Liu et al., 2024). We call such a signal *inseparable*, and finding it in a model does not show that the licensed dataset was used. Thus, under two conditions, a signal cannot both survive laundering intact and attribute a model to the licensed dataset. We state this tradeoff as Theorem 1.

We test the attack against nine published audits from all three families, over images and text, listed in Table 1. The text setting then tests whether a signal survives regeneration, and the image setting tests whether what survives can attribute a model to the licensed dataset. On a language model we pretrain from scratch, laundering takes each text audit to or near the level of a model trained without the dataset. The deployed model keeps 0.85 and 0.82 of the capability gain from direct training. We detect zero exact copies of licensed text passing through the generator.

Our work makes three contributions.

- We introduce data laundering, a single operation that takes nine published audits in image and text to or toward the level of a model trained without the dataset, at fixed capability. We prove the tradeoff that explains why laundering reaches the whole class of audits under two conditions (Theorem 1).
- We evaluate laundering under a fixed-capability protocol with pre-registered predictions, and release the audit reimplementations and the pipeline.
- We run six defenses, including two published audits designed with regeneration in mind. All six are defeated by a branch of the tradeoff or work only where one of its conditions fails. At a 1% base rate, a positive result on a laundered model has a predictive value of at most 0.031 on images at any threshold (Section 5).

2 Threat Model

We consider three parties. A *licensor* owns a dataset $\mathcal{D}$ and sells access to it under a license. A *licensee* buys that access because it wants the capability that $\mathcal{D}$ provides (e.g., an artist's style).

An *auditor*, the licensor or a forensics firm acting for it, later checks with a data-provenance audit $\mathcal{A}(B, \mathcal{D}) \to \{\text{used}, \text{not used}\}$ whether a deployed model B was trained on $\mathcal{D}$ (Du et al., 2025b).

Our adversary is the licensee, which we call the *launderer*. The launderer's goal is to deploy models that keep the capability it paid for and that an audit cannot tie back to $\mathcal{D}$. It launders when its use falls outside the license (e.g., a model the license does not cover). The same attack is open to a party that obtains $\mathcal{D}$ from a licensee. The launderer wins on a deployed model B when the audit returns "not used" and B is still capable. We measure capability as $U(B)$, i.e., the fraction of a directly trained model's held-out capability that B keeps. We fix one threshold τ before any experiment runs, release only models with $U(B) \geq \tau$, and report detection at that fixed capability. U is independent of the audits. Scoring quality with the auditor's own tools would be circular.

Launderer. We assume the launderer holds $\mathcal{D}$ under the license and can fine-tune a single generator on it. The steps it takes use only the access the license allows and public data. We assume it does not know which audit will follow, with one exception. When the licensor watermarks each item of $\mathcal{D}$ before release, the launderer knows the watermark family and holds unmarked public work by the same source (Appendix E.5). We assume it evades on purpose. This excludes cooperative trainers who keep a radioactive signal intact to document their own provenance (Sablayrolles et al., 2020; Sander et al., 2024).

Auditor. We give the auditor three advantages. It may audit any deployed model. The launderer wins only if the audit attributes no deployed model to $\mathcal{D}$. The auditor runs each audit in its strongest published form, including two built with regeneration in mind, tested in Section 5. When a defense claims to survive regeneration, we accept the claim and attack the defense under regeneration, as adaptive-attack practice requires (Tramèr et al., 2020). Access control and output matching (the two enforcement mechanisms already deployed) are outside our scope.

3 Data Laundering

The operation has three steps, shown in Figure 1. The launderer fine-tunes a single *generator* G on the licensed dataset $\mathcal{D}$, draws a regenerated dataset $\mathcal{D}' = \mathsf{L}(\mathcal{D}, G)$ from G, and trains each deployed model B only on $\mathcal{D}'$.

The attack works because regeneration separates the capability the licensee paid for from the audited signal. The generator G is fine-tuned to learn the capability (e.g., a style, a domain or a skill), and it puts that capability back into $\mathcal{D}'$ at each draw. A membership audit instead relies on particular records of $\mathcal{D}$, and only what G memorized of them reaches $\mathcal{D}'$. G thus passes on at most a bounded *residual*, the part of a membership signal that survives regeneration (Zhao & Zhang, 2025; Meeus et al., 2025b). We measure the residual with a *similarity score*. We score each candidate document by its maximum similarity to $\mathcal{D}'$, and take the AUC of that score for members of $\mathcal{D}$ against non-members, computed from $\mathcal{D}'$ only. The score is one test on $\mathcal{D}'$ and thus a lower bound on what the best test on $\mathcal{D}'$ recovers.

3.1 Laundering Steps and Variants

Step 1: fine-tune the generator. The generator G is a low-rank adapter (Hu et al., 2022) trained on $\mathcal{D}$. For images, it is a style adapter on Stable Diffusion 1.5 (SD1.5) (Rombach et al., 2022) or SDXL (Podell et al., 2024). For text, it is an adapter on a language model we pretrain ourselves.

Step 2: regenerate the dataset. The launderer draws $\mathcal{D}'$ from G and holds $\mathcal{D}$ throughout. The auditor has access only to models trained on $\mathcal{D}'$ (Algorithm 1).

Step 3: train the deployed models. A deployed model B is a new adapter on the same base as G, trained only on $\mathcal{D}'$. The launderer's steps are the same for any audit that follows, apart from one dataset-level step for the injected watermark. In Table 7, one pass over images takes 8.1 GPU-minutes on one NVIDIA L40S, or $0.25 at the on-demand list rate. One pass covers any number of deployed models, including the models of other architectures in Table 10.

We call one run of the three steps a laundering pass, and this form of it L2. All main results in this paper use L2. It is one rung of a four-rung ladder, and we run all four. L0 trains B directly on $\mathcal{D}$. L1 replaces $\mathcal{D}$ with the outputs of G, which alone removes most exact copies and most of the membership signal. L2 makes that regeneration cheap with the adapter. L3 regenerates the dataset through a second modality, and we run it only as a control.

3.2 Laundering-Attribution Tradeoff

A deployed model B learns only from $\mathcal{D}'$, which limits what any audit can recover about a record's membership bit M_x. For a fixed candidate record, any test ϕ built from a laundered model satisfies

$$\mathrm{TPR}(\phi) \leq \mathrm{TPR}^{\star}_{\mathcal{D}'}(\alpha)$$

at each false-positive rate α, where $\mathrm{TPR}^{\star}_{\mathcal{D}'}(\alpha)$ is the true-positive rate (TPR) of the best test on $\mathcal{D}'$ itself. We state this as Proposition 1 in Appendix C, where it follows from post-processing (Blackwell, 1953; Dong et al., 2022). The bound holds regardless of which audit the licensor runs, including audits we did not test. An audit that detects more than this at the same false-positive rate would violate the bound, and Figure 2 compares the audits we ran with the similarity score.

Regeneration removes one class of signal and leaves the other. We call a signal *separable* when a model can drop it and still meet $U(B) \geq \tau$, and *inseparable* when dropping it takes $U(B)$ below τ. In our settings, exact copies and per-example membership are separable, and style, domain and skill are inseparable. We call a signal *attributable* to $\mathcal{D}$ when it is absent from models of the same capability trained without $\mathcal{D}$. A signal is evidence only if it survives laundering and is attributable. Separable signals fail the first test and inseparable signals fail the second, under two conditions.

Theorem 1. *Suppose (i) each deployed B is trained only on $\mathcal{D}'$, which G draws under prompts independent of M_x, and (ii) some model B_{neg} trained without $\mathcal{D}$ reaches $U \geq \tau$. Then no inseparable signal measured on B attributes B to $\mathcal{D}$, and a separable signal does so only to the extent that $\mathcal{D}'$ keeps it.*

Condition (i) says that the deployed model learns from the licensed data only through the generator, and condition (ii) says that the capability can be reached without the licensed data. A separable signal reaches B only through $\mathcal{D}'$, and Proposition 1 bounds what an audit recovers from it. An inseparable signal is present in B_{neg} too, and an audit that flags the launderer is likely to flag an innocent party (Zhang et al., 2025a; Liu et al., 2024). The proof and the two conditions in full are in Appendix C. Figure 5 shows the two branches, and false positives on innocent models increase from the separable end to the inseparable end while detection on the launderer does not. Under condition (ii), style cannot attribute a model to $\mathcal{D}$. In practice a bounded residual of the separable part survives (see Section 4.3).

Either condition can fail. An in-place rewrite breaks condition (i), which is the case SDR is built for (Li et al., 2026). Our own settings also break condition (i) and help the audit, since each prompt is a licensed record's caption or title (Appendix F). Condition (ii) does not hold where $\mathcal{D}$ is the only way to reach the capability, and there an inseparable signal does attribute a model to $\mathcal{D}$. Neither condition depends on the architecture, the modality or the audit family.

4 Experiments

We now evaluate laundering on text and images. We score each audit on three kinds of model: a model trained directly on the licensed dataset $\mathcal{D}$, a laundered model B trained only on the regenerated dataset $\mathcal{D}'$, and an innocent model trained without $\mathcal{D}$. We call the audit's score on the directly trained model its *direct* score and its score on the innocent models its *baseline*. Table 1 summarizes the results. Laundering takes each of the nine audits to its baseline or toward it, at fixed capability.

4.1 Experimental Setup

Settings. Text has two settings, T1 (Austen) and T2 (Twain), each with its own held-out author. We pretrain a base language model (114M parameters) on a dataset with both authors removed. We know exactly which documents it has seen (Duan et al., 2024). The licensed dataset $\mathcal{D}$ holds 500 passages. The laundered model B trains on $\mathcal{D}'$ for the same number of steps as the directly trained

Table 1: Laundering takes nine audits, over text and images, from detecting the directly trained model to or toward the level of a model that never saw $\mathcal{D}$, at fixed capability. Entries are detection AUC; DIAGNOSIS is its own detection rate, after the dataset-level step of Section 4.4.

Modality	Audit	Direct	Laundered (ours)	Innocent
Text	Copyright traps (Meeus et al., 2024)	0.997	0.302	-
	Min-K% (Shi et al., 2024)	0.626	0.512	0.502
	DE-COP (Duarte et al., 2024)	0.574	0.528	0.515
Image	CLiD (Zhai et al., 2024)	0.997	0.574	0.437
	CDI (Dubiński et al., 2025)	0.894	0.548	0.529
	SSCD (Pizzi et al., 2022)	0.622	0.550	0.414
	DIAGNOSIS (Wang et al., 2024)	0.92	0.00	0.01
	CSD (Somepalli et al., 2024)	0.571	0.535	0.537
Any	Dataset inference (Maini et al., 2021)	0.999	0.587	0.429

Table 2: Laundering takes text detection to baseline at fixed capability. CLiD AUC under five training conditions on our from-scratch language model, with means and standard deviations over seeds. Capability kept is the condition's gain over the base model as a fraction of direct training's gain, on held-out text.

Training condition	T1 AUC (↓)	T2 AUC (↓)	T1 kept (↑)	T2 kept (↑)
Innocent	0.481 ± 0.011	0.496 ± 0.010	0.92	0.93
Direct on $\mathcal{D}$	0.914 ± 0.010	0.815 ± 0.020	1.00	1.00
Surface perturbation	0.880 ± 0.025	0.815 ± 0.005	0.99	1.02
Register rewrite	0.499 ± 0.012	0.505 ± 0.002	0.71	0.81
Laundered (ours)	0.508 ± 0.009	0.510 ± 0.008	0.85	0.82

model. Images have 4 settings, one per WikiArt artist (WikiArt, 2024). The main results use X1 (Pyotr Konchalovsky) and X2 (Martiros Saryan), each paired with innocent models trained on other artists, listed in Appendix B. All image results use the main generator budget, at which the generator overfits and leaves the audit more signal than a shorter budget does.

Audits. The *census* is the set of audits published through 2025 that work without the trainer's cooperation and return a single decision for a whole dataset. We run nine of them, and Appendix D lists the twelve we do not run and one audio audit. Our membership audit scores each example with the caption-conditioned likelihood score of CLiD (Zhai et al., 2024), then turns those scores into a single decision per dataset with the t-test of CDI (Dubiński et al., 2025). We score all models with the same methodology. A weak audit therefore cannot explain a difference between models. Before laundering, Appendix F checks that each audit detects a directly trained model and has a measured baseline.

Metrics. We report detection on a normalized scale, $(\text{AUC} - \text{baseline})/(\text{direct} - \text{baseline})$, where 0 is the baseline and 1 is direct training. We report variance over redrawn member sets. Redrawing the member set changes detection more than reseeding does (standard deviation 0.087 against 0.050). We measure capability separately from the audits. For images we compute FID (Heusel et al., 2017) and CLIP (Radford et al., 2021) on held-out images, and for text the per-token log-likelihood of the licensed author's held-out passages, with details in Appendix B.4. A laundered model is reported only if it keeps capability (i.e., $U(B) \geq \tau$). For images this means it scores within the seed-to-seed spread of the directly trained model on each capability metric.

4.2 Detection on Text at Fixed Capability

Laundering evades detection with little capability loss. Direct training on $\mathcal{D}$ raises held-out capability by +3.60 nats on T1 and +3.35 nats on T2, and at that training budget the audit detects it. On text, the caption is each passage's own title, and on the directly trained model our membership audit scores an AUC of 0.914 on T1. After laundering, detection on the normalized scale is 0.061

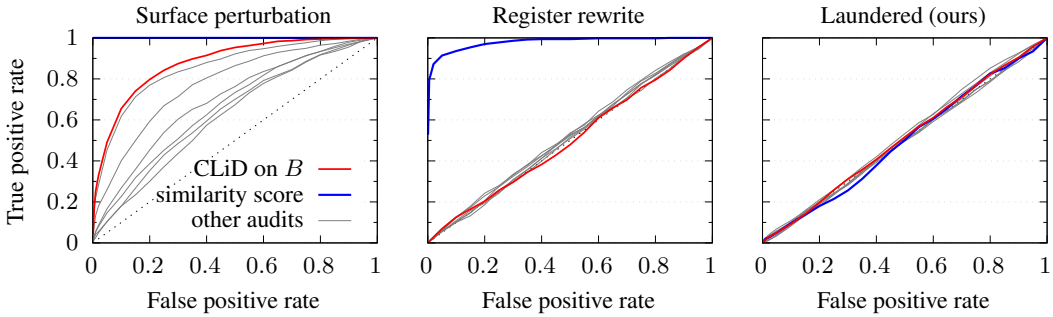

Figure 2: Audits on a laundered model lie at or near the similarity score, setting T1. Each curve is an ROC, true-positive rate against false-positive rate. The similarity score matches each licensed document against the regenerated dataset. At 1% FPR the strongest audit recovers 25% of the similarity score under surface perturbation and 2% under register rewrite.

on T1 and 0.042 on T2. The laundered model B keeps 0.85 of the capability gain on T1 and 0.82 on T2. The innocent models keep 0.92 of that gain on T1 and 0.93 on T2, more than the laundered model. Training on a different author of the same period reaches most of the gain the licensed dataset provides, which is what condition (ii) of Theorem 1 assumes.

Laundering is effective even at low false-positive rates. Because an audit decides at a low false-positive rate (FPR) and does not rank candidates (Carlini et al., 2022), we also report the TPR at 1% FPR. On T1, direct training is detected at a TPR of 0.325, laundering lowers that to 0.015, and models trained without $\mathcal{D}$ score 0.009. On T2 the three rates are 0.211, 0.017 and 0.017. We also run an equivalence test (Schuirmann, 1987) with a pre-registered margin of $[0.46, 0.54]$. It finds each laundered model equivalent to its baseline ($p = 0.022$ on T1, $p = 0.0002$ on T2). A two-sided test still separates the pair. The residual is real but lies inside the margin, as Section 3.2 predicts.

No exact copy survives regeneration. The unconditioned audits, which score the licensed passages without a prompt, are also at their baselines. We detect zero exact copies of licensed text in the regenerated dataset. Table 2 includes two alternative operations. The first, surface perturbation, only drops sentences from each licensed document. The audit still detects the resulting model at nearly the direct-training level, with capability kept.

A register rewrite stays close to the originals. The second alternative is the operation that earlier work calls data laundering (Li et al., 2026), a rewrite of each licensed document into another register (e.g., lyrical). The rewrite also takes detection to the baseline, and keeps 0.71 of the capability gain on T1 and 0.81 on T2. Our operation has a similarity score of 0.493 on T1 and 0.500 on T2, and each rewrite scores between 0.977 and 0.995. Over both settings and all three datasets, 40 of the 42 audit entries lie at or below the bootstrap range of the similarity score (Appendix E).

The result holds at larger base scales. To check that the small base does not drive the result, we rebuilt the pipeline at four base scales (up to 1.4B parameters, Table 11). Even as the direct-training value reaches 1.000, the signal that survives laundering at the 1000-step generator budget remains between 0.028 and 0.136 on the normalized scale.

4.3 Residual and Attribution on Images

Laundering leaves a residual on images. Under laundering, CLiD scores 0.552 and 0.574, above each setting's baseline (0.245 and 0.245 on the normalized scale). Of 24 laundered runs, 11 reach $p < 0.05$, against 1.2 expected with no residual ($p = 6.6 \times 10^{-9}$, exact binomial test). At the audit's operating point, the innocent models give a false-positive rate of 0.0067. An auditor without the licensee's pipeline cannot train such models and cannot measure this rate. With a false-positive rate this close to the true-positive rate, the residual alone is not evidence.

Table 3: Detection increases with the generator's training budget, through 8000 steps on X1 and up to 4000 on X2. CLiD AUC on the laundered model, with means and standard deviations over three seeds.

Generator budget (steps)	**X1**	**X2**
Innocent	0.407	0.437
Direct on $\mathcal{D}$	0.999	0.997
500	0.432 ± 0.010	0.510 ± 0.066
1500	0.488 ± 0.044	0.558 ± 0.010
4000	0.519 ± 0.020	0.611 ± 0.020
8000	0.585 ± 0.034	0.567 ± 0.033

The launderer controls how much residual survives. The residual depends on two choices (the generator's training budget and the deployed architecture). In Table 3, raising the budget from 500 to 4000 steps raises detection by +0.101 within a setting, consistent with the bottleneck bound of Proposition 1. Appendix E.4 shows capability flat over that range. The launderer can pick the shortest budget at no cost in capability. The second choice is the architecture. Trained on the same $\mathcal{D}'$ as the SD1.5 models, a diffusion transformer, PixArt-α (Chen et al., 2024), scores 0.004 on the normalized scale in Table 10, against 0.204 for SD1.5. The 95% confidence interval ($[-0.056, 0.064]$) contains zero.

4.4 Injected Marks

An injected mark is the only case where the auditor gets more than Section 2 allows. The licensor adds the signal to each item before release, sets its strength, and keeps the detector.

A secret-key text watermark flags no laundered model. For text, we rewrite each licensed passage under a green-list watermark with a secret key (Kirchenbauer et al., 2023). The audit triggers on direct training at $z = 4.2$ to 5.3. After laundering, the audit flags zero of the models, on both settings and at both generator budgets, although pooling many runs keeps part of the signal, as Appendix E.6 shows.

DIAGNOSIS survives laundering but not one dataset-level step. For images, DIAGNOSIS (Wang et al., 2024) adds one warp field to each image. A transform applied to each item is a dataset-level signal, and the generator learns it as it learns style. Laundering alone leaves the audit at 0.72 (0.92 for direct training). In Table 13, JPEG or per-image regeneration (Datta et al., 2024) does not help because the generator learns the warp again. A regression over the items can estimate the field they share. This is the only step tailored to one audit, and it uses the artist's public work from outside the license. Removing the estimated field (de-warping) and diluting with that unmarked work drops the audit to at most 0.01 across twelve laundered models on both settings. The step keeps 0.82 to 1.07 of the held-out capability of laundering without it, as Appendix E.5 shows.

Table 9 lists each outcome beside its pre-registered prediction. Each audit with a registered branch reached the branch Theorem 1 predicts, and all but CLiD and dataset inference reached the baseline at the predicted rung.

5 Defenses Against Data Laundering

We consider potential defenses a licensor could deploy against data laundering, in the strongest adversarial setting of Section 2. Here, we evaluate laundering against three defenses, a lower audit threshold and two published audits designed with regeneration in mind, SDR (Li et al., 2026) and MADreMIA (Łapacz & Pawlak, 2026). In Appendix G we explore another three possible defenses, a smaller candidate set, labels the launderer does not choose, and a filter on $\mathcal{D}'$, and find that none of them works. Outside these six, one defense works in part. A watermark that varies per licensed item leaves no dataset-level statistic to estimate and survives laundering on X2 (mean detection 0.55 to 0.90 after dilution), and on X1 its detector also flags 4 of 9 models trained without the licensed images (Appendix E.5).

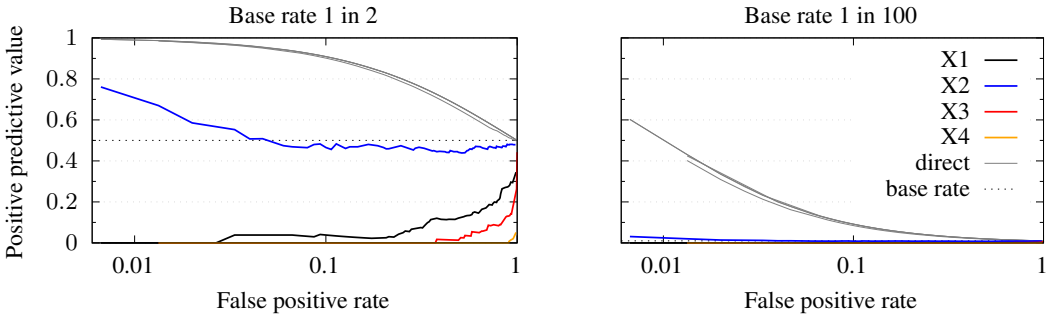

Figure 3: Positive predictive value of CLiD, the chance that a flagged model used the licensed dataset, against the false-positive rate. One curve per image setting. At a base rate of 1 in 100, no threshold gives a laundered model a value above 0.031, while direct training reaches 0.40 to 0.60.

Lowering the audit threshold. Since the image residual is above the baseline, one defense is to lower the audit's threshold until laundered models are flagged. We find this approach ineffective because laundered and innocent models score in the same range. At the audit's operating point, it flags 0.0106 of the licensed images it scores on laundered models and 0.0067 of those on innocent ones, and lower thresholds raise both rates together, as shown in Figure 3. A licensor needs the probability that a flagged party used $\mathcal{D}$ (i.e., the audit's positive predictive value), which Bayes' rule computes from the two rates and a base rate. On the setting where the audit does best, no threshold raises that value above 0.031 at a base rate of 1 in 100. In 2 of the 4 settings the audit flags zero laundered models.

Transform reversion (SDR). SDR targets data laundering in its register-rewrite form. It infers the transform that produced the training dataset from the licensed documents and reverses it. We test whether SDR can defeat our attack. We give it the full search over registers, including the true one, which is more than its own paper does. SDR recovers no more from a laundered model than from an innocent one. SDR's recovery on the laundered model is 0.508, and the 95% interval contains the 0.542 it scores on an innocent model. This is because SDR assumes that each rewritten document comes from one licensed document. Since our launderer samples $\mathcal{D}'$ from the generator with no document as input, the transform SDR reverses does not exist. SDR works in part where its assumption holds. On the in-place register rewrite of Table 2, it scores 0.648, above its innocent-model score.

Iterative regeneration (MADreMIA). MADreMIA regenerates a candidate image repeatedly and scores membership by its decay. Memorized images decay more slowly than others. A sequence of regenerations could then amplify the residual that laundering leaves. We test in Appendix G whether MADreMIA can separate laundered models from innocent ones. MADreMIA detects direct training in each run, flagging 0.96 of directly trained models (1% false-positive rate). It is unable to separate laundered models from innocent ones in any run, and each laundered interval overlaps its baseline. This is because the decay it measures is a property of the image. The same trajectory appears on an innocent model.

6 Related Work

In this section, we present related work on evasion attacks on data-provenance audits and on data laundering. Related work on the audit census and training on generated data is in Appendix A.

Evasion attacks on data-provenance audits. Prior attacks start from a trained model or from the protected images and remove the signal an audit looks for. Distilling a marked classifier into student networks leaves a network tied to the task (Zhu et al., 2025). CEAT2I has to recognize a watermark before removing it (Gao et al., 2026). Per-image regeneration drops the low-level detail that watermarks rely on and keeps the rest of each protected image (Datta et al., 2024). The output thus remains a function of one image. Our attack differs in both its output and its threat model. It

produces a dataset drawn from a generator that takes no licensed record as input, and it does not need to know which audit follows.

Data laundering. The term *data laundering* has earlier uses, for watermark removal from a network (Aiken et al., 2021) and for benchmark distillation (Mansurov et al., 2025), that are related to our problem in name only. The most related work is SDR, which applies the term to training-data provenance and rewrites each licensed document into another register (Li et al., 2026). A rewritten document remains a function of one original, and SDR's defense can reverse part of that dependence. No document enters our generator, and in Section 5 SDR recovers no more from our models than from innocent ones.

Residual signals. Membership survives paraphrase in part (Ertan et al., 2026), distillation (Jagielski et al., 2023) and synthetic text with an injected canary (Meeus et al., 2025b). The last two still pass through a generator and thus meet condition (i) of Theorem 1. Membership attacks also flag machine-generated text as members (Naseh & Mireshghallah, 2025), and a tabular watermark tied to the whole data distribution survives retraining a generator on the watermarked data (Che et al., 2026). Theorem 1 predicts that such a watermark also triggers on models of the same capability trained without it. We follow work that pairs one attack with the principle behind a class of defenses (Athalye et al., 2018; Tramèr et al., 2020; Zhang et al., 2024b; Hönig et al., 2025), with Theorem 1 as our principle.

Training on generated data. Prior work finds that a small fraction of generated data in a large training set can stop a model improving with scale (Dohmatob et al., 2025). Pretraining on a dataset generated from a small one still transfers the knowledge in the source documents (Yang et al., 2025). Our launderer trains only on generated data, at one scale, and holding capability at a fixed threshold measures that cost.

Licensing platforms. An industry has formed around licensing data for training, with marketplaces from Human Native and Created by Humans (TechCrunch, 2024; CNBC, 2026; Created by Humans, 2025) and, since 2024, a Dataset Providers Alliance of licensors (Dataset Providers Alliance, 2024). The provenance tools in use, C2PA Content Credentials, SynthID and the Do Not Train registry, act on outputs or on trainers that opt in (C2PA, 2021; Google DeepMind, 2026; TechCrunch, 2023), and none runs an audit of training data. Research systems that embed provenance at ingest (Xue, 2026) or train a model to name the dataset behind each output (Mo et al., 2026) need the trainer's cooperation, which a launderer often does not provide.

7 Discussion and Conclusion

A licensor sells access to its data but wants to control how the data is used. After the fact, its check on use is an audit of the deployed model. If the licensee takes a single cheap extra step, the audits we tested can no longer tell from that model whether the data was used. The model that could show this is the generator, which the licensee keeps private and an audit cannot reach.

We argue that checking data use is a security problem. Since the licensee holds the data and controls each step from download to deployment, a check on use faces an adversary that knows the check and can adapt to it. The audits we tested were built and evaluated against a weaker adversary or against none. Future audits should be evaluated against a launderer that adapts to them, with a false-positive rate measured on innocent models of the same capability. We believe such a check needs a protocol built like a defense, with a threat model and a guarantee for both parties. The licensor needs a record of what was done with its data, each model trained on it and each dataset derived from those models. The check then does not depend on which model the licensee deploys. Work on machine unlearning reaches the same conclusion. Many datasets can reach one model state, and a checkable claim must therefore refer to the training process itself (Thudi et al., 2022). There, too, a model provider can evade verification while keeping the data (Zhang et al., 2024a). The licensee needs a way to show that its deployed models were built from licensed data. We hope technical work will make the record verifiable by either party.

AI USE STATEMENT

We used generative AI tools to fix grammar and to debug code.

ETHICS STATEMENT

This paper describes an attack on data-provenance audits before licensing deals come to depend on them. All models we attack are ones we trained ourselves. Because the attack needs the licensed dataset itself, it does not change how data is obtained and defeats only the audit that follows. The deployed systems we survey do not run these audits as enforcement. We will share our findings with the authors of the audits we reimplement, and record any coordinated response in the camera-ready version. All data used is publicly available, and the work involves no human subjects.

REPRODUCIBILITY STATEMENT

The evaluation protocol, audit reimplementations, and laundering pipeline are released as LAUNDERBENCH in the supplementary material and at an anonymized repository (https://anonymous.4open.science/r/launderbench-2266). Appendix B lists the full configurations, Appendix D the adaptive-auditor setups and the per-audit details, and Appendix F the controls that keep a non-working audit from counting as a success. Together these let a reader re-run all evaluations in the paper and re-test each defense.

REFERENCES

Adobe. Adobe Firefly + Adobe Stock. https://blog.adobe.com/en/publish/2023/09/13/adobe-firefly-adobe-stock, 2023.

William Aiken, Hyoungshick Kim, Simon Woo, and Jungwoo Ryoo. Neural network laundering: Removing black-box backdoor watermarks from deep neural networks. *Computers & Security*, 106:102277, 2021.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *Proc. of ICML*, 2018.

Authors Guild. What authors need to know about the $1.5 billion Anthropic settlement. https://authorsguild.org/advocacy/artificial-intelligence/what-authors-need-to-know-about-the-anthropic-settlement/, 2025.

Axios. Exclusive: AP strikes news-sharing and tech deal with OpenAI. https://www.axios.com/2023/07/13/ap-openai-news-sharing-tech-deal, 2023.

David Blackwell. Equivalent comparisons of experiments. *The Annals of Mathematical Statistics*, 1953.

Wassim Bouaziz, Nicolas Usunier, and El-Mahdi El-Mhamdi. Data taggants: Dataset ownership verification via harmless targeted data poisoning. In *Proc. of ICLR*, 2025.

C2PA. Technology and media entities join forces to create standards group aimed at building trust in online content. https://c2pa.org/c2pa-founding-press-release/, 2021.

California State Legislature. Assembly Bill 2013: Generative artificial intelligence: Training data transparency. https://leginfo.legislature.ca.gov/faces/billTextClient.xhtml?bill_id=202320240AB2013, 2024.

Nicholas Carlini, Steve Chien, Milad Nasr, Shuang Song, Andreas Terzis, and Florian Tramèr. Membership inference attacks from first principles. In *Proc. of IEEE S&P*, 2022.

Nicholas Carlini et al. Extracting training data from diffusion models. In *Proc. of USENIX Security*, 2023.

Xin Che, Lingyang Chu, Qiqi Zhang, Xinyu Ma, Xuan Luo, and Jian Pei. RaMark: Radioactive watermarking for generated tabular data. *arXiv preprint arXiv:2607.09000*, 2026.

Junsong Chen et al. PixArt-α: Fast training of diffusion transformer for photorealistic text-to-image synthesis. In *Proc. of ICLR*, 2024.

Dami Choi, Yonadav Shavit, and David Duvenaud. Tools for verifying neural models' training data. In *Proc. of NeurIPS*, 2023.

CNBC. Cloudflare acquires AI data marketplace Human Native. https://www.cnbc.com/2026/01/15/cloudflare-ai-human-native-acquisition.html, 2026.

Jade Copet, Felix Kreuk, Itai Gat, Tal Remez, David Kant, Gabriel Synnaeve, Yossi Adi, and Alexandre Défossez. Simple and controllable music generation. In *Proc. of NeurIPS*, 2023.

Created by Humans. The AI rights licensing platform for books. https://www.createdbyhumans.ai/, 2025.

Yingqian Cui et al. FT-Shield: A watermark against unauthorized fine-tuning in text-to-image diffusion models. *ACM SIGKDD Explorations Newsletter*, 26(2):76–88, 2025.

Debeshee Das, Jie Zhang, and Florian Tramèr. Blind baselines beat membership inference attacks for foundation models. In *Proc. of IEEE S&P Workshops*, 2025.

Dataset Providers Alliance. Leading dataset licensors unite to launch the Dataset Providers Alliance (DPA). https://www.prnewswire.com/news-releases/leading-dataset-licensors-unite-to-launch-the-dataset-providers-alliance-dpa-302182603.html, 2024.

Soumil Datta, Shih-Chieh Dai, Leo Yu, and Guanhong Tao. Exploiting watermark-based defense mechanisms in text-to-image diffusion models for unauthorized data usage. *arXiv preprint arXiv:2411.15367*, 2024.

Elvis Dohmatob, Yunzhen Feng, Arjun Subramonian, and Julia Kempe. Strong model collapse. In *Proc. of ICLR*, 2025.

Jinshuo Dong, Aaron Roth, and Weijie J. Su. Gaussian differential privacy. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 84(1):3–37, 2022.

Tian Dong, Yan Meng, Shaofeng Li, Guoxing Chen, Yuling Chen, Zhen Liu, Haojin Zhu, and Hao Chen. DIPBox: A multi-scale testing framework for tracking dataset regeneration. In *Proc. of CCS*, 2026.

Linkang Du et al. ArtistAuditor: Auditing artist style pirate in text-to-image generation models. In *Proc. of WWW*, 2025a.

Linkang Du et al. SoK: Dataset copyright auditing in machine learning systems. In *Proc. of IEEE S&P*, 2025b.

Jinhao Duan, Fei Kong, Shiqi Wang, Xiaoshuang Shi, and Kaidi Xu. Are diffusion models vulnerable to membership inference attacks? In *Proc. of ICML*, 2023.

Michael Duan et al. Do membership inference attacks work on large language models? In *Proc. of COLM*, 2024.

André V. Duarte, Xuandong Zhao, Arlindo L. Oliveira, and Lei Li. DE-COP: Detecting copyrighted content in language models training data. In *Proc. of ICML*, 2024.

Jan Dubiński, Antoni Kowalczuk, Franziska Boenisch, and Adam Dziedzic. CDI: Copyrighted data identification in diffusion models. In *Proc. of CVPR*, 2025.

Murat Bilgehan Ertan, Emirhan Böge, Min Chen, Kaleel Mahmood, and Marten van Dijk. On the evidentiary limits of membership inference for copyright auditing. *arXiv preprint arXiv:2601.12937*, 2026.

European Parliament and Council of the European Union. Regulation (EU) 2024/1689 laying down harmonised rules on artificial intelligence (Artificial Intelligence Act). https://eur-lex.europa.eu/eli/reg/2024/1689/oj, 2024.

Pierre Fernandez, Guillaume Couairon, Hervé Jégou, Matthijs Douze, and Teddy Furon. The stable signature: Rooting watermarks in latent diffusion models. In *Proc. of ICCV*, 2023.

Kuofeng Gao et al. Toward dataset copyright evasion attack against personalized text-to-image diffusion models. *IEEE Transactions on Information Forensics and Security*, 21:725–740, 2026.

Google DeepMind. SynthID: A tool to watermark and identify content generated through AI. https://deepmind.google/models/synthid/, 2026.

Junfeng Guo et al. Domain watermark: Effective and harmless dataset copyright protection is closed at hand. In *Proc. of NeurIPS*, 2023.

Martin Heusel, Hubert Ramsauer, Thomas Unterthiner, Bernhard Nessler, and Sepp Hochreiter. GANs trained by a two time-scale update rule converge to a local Nash equilibrium. In *Proc. of NeurIPS*, 2017.

Robert Hönig, Javier Rando, Nicholas Carlini, and Florian Tramèr. Adversarial perturbations cannot reliably protect artists from generative AI. In *Proc. of ICLR*, 2025.

Edward J. Hu et al. LoRA: Low-rank adaptation of large language models. In *Proc. of ICLR*, 2022.

Leixu Huang, Zedian Shao, and Teodora Baluta. Watermark robustness and radioactivity may be at odds in federated learning. *arXiv preprint arXiv:2510.17033*, 2025.

Zonghao Huang, Neil Zhenqiang Gong, and Michael K. Reiter. A general framework for data-use auditing of ML models. In *Proc. of CCS*, 2024.

Matthew Jagielski, Milad Nasr, Katherine Lee, Christopher A. Choquette-Choo, Nicholas Carlini, and Florian Tramèr. Students parrot their teachers: Membership inference on model distillation. In *Proc. of NeurIPS*, 2023.

Hengrui Jia et al. Proof-of-learning: Definitions and practice. In *Proc. of IEEE S&P*, 2021.

John Kirchenbauer, Jonas Geiping, Yuxin Wen, Jonathan Katz, Ian Miers, and Tom Goldstein. A watermark for large language models. In *Proc. of ICML*, 2023.

John Kirchenbauer, Brian R. Bartoldson, Bhavya Kailkhura, and Tom Goldstein. Watermarking for proprietary dataset protection. In *Proc. of ICML Workshop on Trustworthy AI for Good*, 2026.

Anand Kumar, Jiteng Mu, and Nuno Vasconcelos. IntroStyle: Training-free introspective style attribution using diffusion features. In *Proc. of ICCV*, 2025.

Gustav Olaf Yunus Laitinen-Fredriksson Lundström-Imanov and Nurana Abdullayeva. Verifiable provenance and watermarking for generative AI: An evidentiary framework for international operational law and domestic courts. *arXiv preprint arXiv:2605.21002*, 2026.

Wojciech Łapacz and Stanisław Pawlak. Amplifying membership signal through chained regeneration. *arXiv preprint arXiv:2606.31991*, 2026.

Boheng Li et al. Towards reliable verification of unauthorized data usage in personalized text-to-image diffusion models. In *Proc. of IEEE S&P*, 2025a.

Muxing Li, Zesheng Ye, Sharon Li, and Feng Liu. Combating data laundering in LLM training. *arXiv preprint arXiv:2604.01904*, 2026.

Muxing Li et al. Distributional statistics restore training data auditability in one-step distilled diffusion models. *arXiv preprint arXiv:2502.02970*, 2025b.

Yiming Li, Yang Bai, Yong Jiang, Yong Yang, Shu-Tao Xia, and Bo Li. Untargeted backdoor watermark: Towards harmless and stealthy dataset copyright protection. In *Proc. of NeurIPS*, 2022.

Yiming Li et al. MOVE: Effective and harmless ownership verification via embedded external features. *IEEE Transactions on Pattern Analysis and Machine Intelligence*, 2025c.

Chumeng Liang and Jiaxuan You. Real-world benchmarks make membership inference attacks fail on diffusion models. *arXiv preprint arXiv:2410.03640*, 2024.

Jian Liu, Rui Zhang, Sebastian Szyller, Kui Ren, and N. Asokan. False claims against model ownership resolution. In *Proc. of USENIX Security*, 2024.

Shayne Longpre et al. A large-scale audit of dataset licensing and attribution in AI. *Nature Machine Intelligence*, 2024.

Yiwei Lu, Matthew Y. R. Yang, Zuoqiu Liu, Gautam Kamath, and Yaoliang Yu. Disguised copyright infringement of latent diffusion models. In *Proc. of ICML*, 2024.

Nils Lukas, Edward Jiang, Xinda Li, and Florian Kerschbaum. SoK: How robust is image classification deep neural network watermarking? In *Proc. of IEEE S&P*, 2022.

Hengyu Luo et al. Knowledge distillation as decontamination? Revisiting the "data laundering" concern in classification tasks. In *Proc. of ICLR*, 2026.

Pratyush Maini, Mohammad Yaghini, and Nicolas Papernot. Dataset inference: Ownership resolution in machine learning. In *Proc. of ICLR*, 2021.

Pratyush Maini, Hengrui Jia, Nicolas Papernot, and Adam Dziedzic. LLM dataset inference: Did you train on my dataset? In *Proc. of NeurIPS*, 2024.

Xiafeng Man, Zhipeng Wei, and Jingjing Chen. Copyright infringement detection in text-to-image diffusion models via differential privacy. In *Proc. of AAAI*, pp. 7910–7917, 2026.

Jonibek Mansurov, Akhmed Sakip, and Alham Fikri Aji. Data laundering: Artificially boosting benchmark results through knowledge distillation. In *Proc. of ACL*, pp. 8332–8345, 2025.

Matthieu Meeus, Igor Shilov, Manuel Faysse, and Yves-Alexandre de Montjoye. Copyright traps for large language models. In *Proc. of ICML*, 2024.

Matthieu Meeus, Igor Shilov, Shubham Jain, Manuel Faysse, Marek Rei, and Yves-Alexandre de Montjoye. SoK: Membership inference attacks on LLMs are rushing nowhere (and how to fix it). In *Proc. of IEEE SaTML*, 2025a.

Matthieu Meeus, Lukas Wutschitz, Santiago Zanella-Béguelin, Shruti Tople, and Reza Shokri. The canary's echo: Auditing privacy risks of LLM-generated synthetic text. In *Proc. of ICML*, 2025b.

Michel Meintz, Jan Dubiński, Franziska Boenisch, and Adam Dziedzic. Radioactive watermarks in diffusion and autoregressive image generative models. *arXiv preprint arXiv:2506.23731*, 2025.

Wenjie Jacky Mo, Qin Liu, Xiaofei Wen, Wenxuan Zhou, Zhe Zhao, and Muhao Chen. DebugLM: Learning traceable training data provenance for LLMs. *arXiv preprint arXiv:2603.17884*, 2026.

Ali Naseh and Niloofar Mireshghallah. Synthetic data can mislead evaluations: Membership inference as machine text detection. *arXiv preprint arXiv:2501.11786*, 2025.

Yonatan Oren, Nicole Meister, Niladri Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. Proving test set contamination in black-box language models. In *Proc. of ICLR*, 2024.

Leyi Pan et al. Can LLM watermarks robustly prevent unauthorized knowledge distillation? In *Proc. of ACL*, pp. 13228–13251, 2025.

Gaurav Parmar, Richard Zhang, and Jun-Yan Zhu. On aliased resizing and surprising subtleties in GAN evaluation. In *Proc. of CVPR*, 2022.

Ed Pizzi, Sreya Dutta Roy, Sugosh Nagavara Ravindra, Priya Goyal, and Matthijs Douze. A self-supervised descriptor for image copy detection. In *Proc. of CVPR*, 2022.

Dustin Podell et al. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *Proc. of ICLR*, 2024.

Jakub Proboszcz et al. Membership and dataset inference attacks on large audio generative models. *arXiv preprint arXiv:2512.09654*, 2025.

Leyi Qi, Yiming Li, Siyuan Liang, Zhengzhong Tu, and Dacheng Tao. Cert-LAS: Toward certified model ownership verification for text-to-image diffusion models via layer-adaptive smoothing. In *Proc. of ICML*, 2026.

Alec Radford et al. Learning transferable visual models from natural language supervision. In *Proc. of ICML*, 2021.

Saksham Rastogi, Pratyush Maini, and Danish Pruthi. STAMP your content: Proving dataset membership via watermarked rephrasings. In *Proc. of ICML*, 2025.

Reddit, Inc. Form S-1 registration statement. https://www.sec.gov/Archives/edgar/data/1713445/000162828024006294/reddits-1q423.htm, 2024.

Jie Ren et al. Self-comparison for dataset-level membership inference in large (Vision-)language model. In *Proc. of WWW*, pp. 910–920, 2025.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proc. of CVPR*, 2022.

Nataniel Ruiz, Yuanzhen Li, Varun Jampani, Yael Pritch, Michael Rubinstein, and Kfir Aberman. DreamBooth: Fine tuning text-to-image diffusion models for subject-driven generation. In *Proc. of CVPR*, 2023.

Alexandre Sablayrolles, Matthijs Douze, Cordelia Schmid, and Hervé Jégou. Radioactive data: Tracing through training. In *Proc. of ICML*, 2020.

Tom Sander, Pierre Fernandez, Alain Durmus, Matthijs Douze, and Teddy Furon. Watermarking makes language models radioactive. In *Proc. of NeurIPS*, 2024.

Donald J. Schuirmann. 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.

Shuo Shao et al. DATABench: Evaluating dataset auditing in deep learning from an adversarial perspective. *arXiv preprint arXiv:2507.05622*, 2025.

Weijia Shi et al. Detecting pretraining data from large language models. In *Proc. of ICLR*, 2024.

Ilia Shumailov, Zakhar Shumaylov, Yiren Zhao, Nicolas Papernot, Ross Anderson, and Yarin Gal. AI models collapse when trained on recursively generated data. *Nature*, 2024.

Gowthami Somepalli, Vasu Singla, Micah Goldblum, Jonas Geiping, and Tom Goldstein. Understanding and mitigating copying in diffusion models. In *Proc. of NeurIPS*, 2023.

Gowthami Somepalli et al. Investigating style similarity in diffusion models. In *Proc. of ECCV*, 2024.

TechCrunch. Spawning lays out plans for letting creators opt out of generative AI training. https://techcrunch.com/2023/05/03/spawning-lays-out-its-plans-for-letting-creators-opt-out-of-generative-ai-training/, 2023.

TechCrunch. Deal Dive: Human Native AI is building the marketplace for AI training licensing deals. https://techcrunch.com/2024/06/08/deal-dive-human-native-ai-is-building-the-marketplace-for-ai-training-licensing-deals/, 2024.

TechCrunch. Warner Music signs deal with AI music startup Suno, settles lawsuit. https://techcrunch.com/2025/11/25/warner-music-signs-deal-with-ai-music-startup-suno-settles-lawsuit/, 2025.

Anvith Thudi, Hengrui Jia, Ilia Shumailov, and Nicolas Papernot. On the necessity of auditable algorithmic definitions for machine unlearning. In *Proc. of USENIX Security*, 2022.

Yao Tong, Jiayuan Ye, Sajjad Zarifzadeh, and Reza Shokri. How much of my dataset did you use? Quantitative data usage inference in machine learning. In *Proc. of ICLR*, 2025.

Florian Tramèr, Nicholas Carlini, Wieland Brendel, and Aleksander Madry. On adaptive attacks to adversarial example defenses. In *Proc. of NeurIPS*, 2020.

Variety. News Corp inks OpenAI licensing deal potentially worth more than $250 million. https://variety.com/2024/digital/news/news-corp-openai-licensing-deal-1236013734/, 2024.

Zhenting Wang, Chen Chen, Lingjuan Lyu, Dimitris N. Metaxas, and Shiqing Ma. DIAGNOSIS: Detecting unauthorized data usages in text-to-image diffusion models. In *Proc. of ICLR*, 2024.

Johnny Tian-Zheng Wei, Ryan Yixiang Wang, and Robin Jia. Proving membership in LLM pretraining data via data watermarks. In *Findings of ACL*, 2024.

Yuxin Wen, John Kirchenbauer, Jonas Geiping, and Tom Goldstein. Tree-rings watermarks: Invisible fingerprints for diffusion images. In *Proc. of NeurIPS*, 2023.

Emily Wenger, Xiuyu Li, Ben Y. Zhao, and Vitaly Shmatikov. Data isotopes for data provenance in DNNs. *Proceedings on Privacy Enhancing Technologies*, 2024.

WikiArt. WikiArt: Visual art encyclopedia. https://www.wikiart.org/, 2024.

Yixin Wu, Ziqing Yang, Yun Shen, Michael Backes, and Yang Zhang. Synthetic artifact auditing: Tracing LLM-generated synthetic data usage in downstream applications. In *Proc. of USENIX Security*, 2025.

Yuechen Xie, Jie Song, Huiqiong Wang, and Mingli Song. Training data provenance verification: Did your model use synthetic data from my generative model for training? In *Proc. of CVPR*, 2025.

Haolin Xue. OriginBlame: Record- and token-level data provenance for AI training datasets. *arXiv preprint arXiv:2607.13037*, 2026.

Zijin Yang, Kai Zeng, Kejiang Chen, Han Fang, Weiming Zhang, and Nenghai Yu. Gaussian shading: Provable performance-lossless image watermarking for diffusion models. In *Proc. of CVPR*, 2024.

Zitong Yang, Neil Band, Shuangping Li, Emmanuel Candès, and Tatsunori Hashimoto. Synthetic continued pretraining. In *Proc. of ICLR*, 2025.

Zexi Yao, Nataša Krčo, Georgi Ganev, and Yves-Alexandre de Montjoye. The DCR delusion: Measuring the privacy risk of synthetic data. In *Proc. of ESORICS*, 2025.

Ning Yu, Larry Davis, and Mario Fritz. Attributing fake images to GANs: Learning and analyzing GAN fingerprints. In *Proc. of ICCV*, 2019.

Shengfang Zhai et al. Membership inference on text-to-image diffusion models via conditional likelihood discrepancy. In *Proc. of NeurIPS*, 2024.

Binchi Zhang, Zihan Chen, Cong Shen, and Jundong Li. Verification of machine unlearning is fragile. In *Proc. of ICML*, 2024a.

Hanlin Zhang, Benjamin L. Edelman, Danilo Francati, Daniele Venturi, Giuseppe Ateniese, and Boaz Barak. Watermarks in the sand: Impossibility of strong watermarking for language models. In *Proc. of ICML*, 2024b.

Jie Zhang, Debeshee Das, Gautam Kamath, and Florian Tramèr. Position: Membership inference attacks cannot prove that a model was trained on your data. In *Proc. of IEEE SaTML*, 2025a.

Jingqi Zhang, Ruibo Chen, Yingqing Yang, Peihua Mai, Heng Huang, and Yan Pang. Leave no TRACE: Black-box detection of copyrighted dataset usage in large language models via watermarking. *arXiv preprint arXiv:2510.02962*, 2025b.

Jingyang Zhang et al. Min-K%++: Improved baseline for pre-training data detection from large language models. In *Proc. of ICLR*, 2025c.

Bihe Zhao, Pratyush Maini, Franziska Boenisch, and Adam Dziedzic. Unlocking post-hoc dataset inference with synthetic data. In *Proc. of ICML*, 2025.

Yunpeng Zhao and Jie Zhang. Does training with synthetic data truly protect privacy? In *Proc. of ICLR*, 2025.

Hongyu Zhu, Sichu Liang, Wenwen Wang, Zhuomeng Zhang, Fangqi Li, and Shi-Lin Wang. Evading data provenance in deep neural networks. In *Proc. of ICCV*, 2025.

Table 4: Deployed systems by class, and the audit each runs. The table surveys deployed systems, and no entry is a measurement of ours. No surveyed class runs the data-provenance audits that Section 4 takes to or toward the baseline.

Class	Mechanism	Training-data audit?
Access-gated dataset APIs	credential gate before access	none
Licensing marketplaces	contract and payment	none
Output-side content matchers	output similarity match	output only
Root-of-trust / content credentials	signed release provenance	output only
Contractual self-attestation	declared data manifest	unverified self-report

A Extended Related Work

We first cover the platforms and tools a license is enforced with today, then the provenance audits we break. We then discuss prior laundering attacks and the evidence on what survives training on generated data. We close with the attacks against whole classes of defenses that our analysis follows.

Licensing platforms and provenance tools. An industry has formed around licensing data for training, with marketplaces from Human Native (acquired by Cloudflare in 2026) and Created by Humans (TechCrunch, 2024; CNBC, 2026; Created by Humans, 2025). Since 2024 some licensors have organized themselves into the Dataset Providers Alliance (Dataset Providers Alliance, 2024). The provenance tools in use today work on outputs rather than on training data. C2PA Content Credentials and Google's SynthID attach provenance to an output (C2PA, 2021; Google DeepMind, 2026), and Spawning's Do Not Train registry covers only trainers who choose to respect it (TechCrunch, 2023). Most license records of text datasets are missing or incorrect (Longpre et al., 2024). Table 4 lists the deployed enforcement systems, and none of them runs an audit of training data, whether by injected watermark or by membership. Access control decides who may download $\mathcal{D}$ and does not decide what the licensee then does with it. Output matching compares a model's outputs with the licensed items, and a model that does not reproduce them is not flagged (Carlini et al., 2023). Research systems go further. Xue (2026) embeds provenance in each record or token inside the trainer's own ingest pipeline. DebugLM (Mo et al., 2026) trains a model to name the dataset behind each output (with the trainer's cooperation). Cert-LAS certifies that a text-to-image model is the one its owner watermarked (Qi et al., 2026).

Provenance audits. The audits we break are the standard ones. Table 8 lists them in three families, drawing on a recent survey (Du et al., 2025b). The first family adds a signal (e.g., an injected watermark or radioactive data) to the dataset before release (Sablayrolles et al., 2020; Meintz et al., 2025; Wenger et al., 2024; Wang et al., 2024; Meeus et al., 2024; Bouaziz et al., 2025). The second scores a trained model record by record, as membership inference does (Carlini et al., 2022; Duan et al., 2023; Shi et al., 2024; Zhang et al., 2025c; Duarte et al., 2024), or pools those scores over a collection, as dataset inference does (Maini et al., 2021; 2024; Dubiński et al., 2025). The third attributes a model to the sources its outputs resemble. It has two parts that we keep separate, a style signal (Somepalli et al., 2024) and the near-copies of training images that diffusion models produce (Carlini et al., 2023; Somepalli et al., 2023). Training-free style attribution outperforms earlier methods (Kumar et al., 2025), but innocent models can share a style. Section 3.2 shows when style alone cannot attribute a model to $\mathcal{D}$. Most membership evaluations on LLMs suffer from distribution shift (Meeus et al., 2025a), and our text baselines avoid it with the controlled split of Section 4.1. Most of these audits were validated against a trainer who fine-tunes on the licensed dataset directly and does not expect to be checked.

Several more audits lie outside Table 1. Radioactivity depends on watermarks in a generator's own output (Sander et al., 2024), and our generator is private to the launderer. The extension of radioactivity to text that the owner watermarks before release (Kirchenbauer et al., 2026) is a watermark shared by the whole dataset, like the injected image watermark that one dataset-wide step removes in Appendix E.5. SMI compares the likelihood of each record before and after paraphrasing it (Ren et al., 2025). The difference depends on the model having seen the record. This is a signal the capability does not need, and Theorem 1 predicts that regeneration drops it. D-Plus-Minus fine-tunes the deployed model to learn and to unlearn one candidate image and compares how its outputs change

(Man et al., 2026). The change also depends on the model having seen the image. FT-Shield optimizes an image watermark to transfer into the outputs of a model fine-tuned on the watermarked images (Cui et al., 2025), and the same transfer carries DIAGNOSIS through laundering alone. Three further audits identify something one step removed from the licensed dataset. One attributes a deployed model to the generator that produced its training data (Xie et al., 2025), which on our pipeline names only the launderer's generator. Another detects whether an output came from synthetic data (Wu et al., 2025). Generator fingerprints survive heavy editing but identify the model that produced the output (Yu et al., 2019), again the launderer's generator.

Data laundering and provenance evasion. Prior laundering attacks produce a *model*, while ours produces a *dataset*, and each difference below follows from that. Zhu et al. (2025) distill a marked classifier over an out-of-distribution transfer set and train several student architectures from that intermediary. The output is a network, tied to the task, that an auditor can identify. Our output is data that can train generative models of other architectures, and it leaves an auditor at most the small residual signal that Section 4 measures. Gao et al. (2026) strip a watermark from a network by detecting the watermarked samples and erasing the trigger tokens. They have to recognize the watermark before removing it. Our launderer runs the same steps for any signal or audit, apart from one step against the injected watermark. Datta et al. (2024) fine-tune on diffusion regenerations of protected images, which keep the high-level features and drop the low-level detail that watermarks rely on. The outputs remain a function of one protected image each. Lu et al. (2024) hide licensed images inside training images that look unrelated, where an inspection of the dataset does not find them. Their attack targets inspection of the dataset, and ours targets an audit of the model. A separate line of work asks whether generated media came from a model and which one (Laitinen-Fredriksson Lundström-Imanov & Abdullayeva, 2026), and it does not ask what the model was trained on.

We did not coin *data laundering*, and the earlier uses name a different operation. Neural network laundering removes a black-box backdoor watermark from a network (Aiken et al., 2021). On benchmarks, the term names distillation that moves benchmark-specific knowledge into a student (Mansurov et al., 2025), an effect found mostly absent in eight classification benchmarks (Luo et al., 2026). SDR applies the term to training-data provenance for language models, the problem we study (Li et al., 2026). SDR's operation rewrites each licensed document. The rewritten document remains a function of one original, and SDR's defense can reverse part of that dependence. Our dataset comes from a generator that takes no licensed document as input, and in Section 5 SDR's reversal recovers no more from it than from a model trained without the dataset. In Table 2 the rewrite takes the audits we run as close to the baseline as our operation does, but in Table 14 its dataset still identifies its source.

Regeneration and residual signal. Prior evidence on what survives training on generated data is mixed. Training on generated data drops the tails of the original distribution (Shumailov et al., 2024). A signal in those tails does not reach the regenerated dataset. However, some signal survives. The memorization that our similarity score ϱ measures for the launderer in Section 4.2 is what Li et al. (2025b) study as a defense. Regenerated synthetic data does not always protect privacy (Zhao & Zhang, 2025), and we designed ϱ to detect that leakage. Membership survives distillation (Jagielski et al., 2023), but the launderer releases a regenerated dataset instead of the generator. Whether a signal survives that step has been asked of watermarks (Pan et al., 2025). On text, paraphrase weakens membership inference without removing the signal (Ertan et al., 2026), and membership survives into synthetic text strongly enough to be audited when the owner injects a canary for that purpose (Meeus et al., 2025b). Such scores can also mislead, since membership attacks flag machine-generated text as members (Naseh & Mireshghallah, 2025). A tabular watermark (Che et al., 2026) can be tied to the data distribution so that it survives retraining a generator on watermarked data (our operation). Theorem 1 thus predicts that it also triggers on models of the same capability trained without the watermarked data. The residuals above must still pass through the generator and therefore fall under condition (i) of the theorem.

Capability from generated data. Prior evidence on the capability a model keeps from generated data is also mixed. Generating a larger dataset from a small one and pretraining on it transfers the knowledge in the source documents (Yang et al., 2025). This transfer is the capability our launderer keeps. A small fraction of generated data in a large training set can stop a model improving with

Algorithm 1 One laundering pass. The launderer runs steps 1 to 3 and does not share $\mathcal{D}$. The auditor acts only at step 4 and only ever inspects a model trained on the regenerated dataset $\mathcal{D}'$. Any number of deployed models $\{B\}$, of any architecture, train on one pass at capability $\geq \tau$.

Require: licensed dataset $\mathcal{D}$, open base M_0, prompt set P, capability threshold τ
1: Fine-tune M_0 on $\mathcal{D}$ to obtain the generator G ▷ the rung, L1 to L3, sets the method
2: $\mathcal{D}' \leftarrow \mathsf{L}(\mathcal{D}, G) = \{\, G(p) : p \in P \,\}$ ▷ regenerate: $\mathcal{D}$'s samples are not copied forward
3: Train any deployed B, of any architecture, only on $\mathcal{D}'$. Repeat for as many B as wanted
4: Auditor runs $\mathcal{A}(B, \mathcal{D}) \rightarrow \{\text{used},\ \text{not used}\}$ ▷ B has only ever seen $\mathcal{D}'$
Ensure: deployed models $\{B\}$ with $U(B) \geq \tau$, each scoring at or near the baseline

scale (Dohmatob et al., 2025). Our launderer trains only on generated data, at one scale, and holding capability at a fixed threshold measures that cost.

Attacks against classes of defenses. We build on work that pairs one general attack with the principle explaining why a whole class of defenses fails, namely obfuscated gradients (Athalye et al., 2018), adaptive attacks (Tramèr et al., 2020), the impossibility of strong watermarking (Zhang et al., 2024b), and perturbations that cannot reliably protect artists (Hönig et al., 2025). Lukas et al. (2022) evaluate the schemes as a class, find that each surveyed scheme can be broken, and name per-scheme robustness claims as the flaw. Our result is the dataset-side version of that analysis, with Theorem 1 as its principle. Shao et al. (2025) find that each method tested there fails on robustness or on distinctiveness. This covers both branches of our tradeoff, measured on nine detectors for classifiers, and leaves generative models out. DIPBox answers the same question for the defender and tests a suspect dataset at the sample, set and distribution level (Dong et al., 2026). The paper proves that a regenerated dataset that keeps utility remains close to the licensed one. This bounds divergence and leaves attribution open. DIPBox's deployed models are classifiers, and its negative datasets come from another distribution. Our negatives reach the same capability and carry the same inseparable signal. In federated watermarking, Huang et al. (2025) show the same tension between robustness and traceability (the one laundering exploits). Finally, Thudi et al. (2022) argue that many datasets can reach one model state, and a checkable claim must therefore refer to the training process itself. We reach the same conclusion through provenance. The matching attack there evades unlearning verification while covertly keeping the data (Zhang et al., 2024a).

B Experimental Details

B.1 Models and Data

Algorithm 1 states the laundering loop and Table 5 lists the pipeline for each modality. We release our reimplementation of each audit and the fixed-capability protocol together as LaunderBench. The audits run there at their published strength, with the two deviations Appendix D notes.

Dataset filter. We apply a single filter to each licensed dataset. We score the audit on the untouched open base, which has not seen $\mathcal{D}$. We keep the dataset only if that score is already at or below the *baseline range*, i.e., the interval of Table 6 in which models trained without $\mathcal{D}$ score.

Generator budget and prompts. We train the image generator to the overfit budget of Section 4.3, which gives us the largest residual to suppress. The prompts P are the caption or title keys of the licensed records themselves. This breaks condition (i) in a way that helps the audit, as Appendix F shows. We generate at the L2 rung (independent of any watermark). Only the injected watermark of Appendix E.5 needed a step tied to the watermark's family.

Reference operations. Table 2 includes two reference conditions that are not our operation. Under *surface perturbation*, each member keeps a random unbroken stretch of its sentences and drops up to 15%, with a minimum of two sentences, and a document of fewer than four sentences passes through whole. *Register rewrite* is the operation of the SDR paper (Li et al., 2026). Each member yields exactly one rewritten document. The dataset remains a per-document function of $\mathcal{D}$ and violates

Table 5: The pipeline, per modality. Rows run from the licensed dataset to the audit, and the Settings row defines the setting codes used in each table. All deployed models train only on the regenerated dataset. Full schedules are released with LAUNDERBENCH.

Stage	Image (primary)	Text	Audio
Settings	X1 Konchalovsky, X2 Saryan, X3 Serov, X4 Korovin	T1 Austen, T2 Twain	one per artist
Licensed $\mathcal{D}$	WikiArt style set (WikiArt, 2024)	per-author dataset	per-artist dataset
Open base M_0	SD1.5 (Rombach et al., 2022), SDXL (Podell et al., 2024)	from-scratch LLM	MusicGen (Copet et al., 2023)
Generator G	LoRA (Hu et al., 2022) style adapter (Ruiz et al., 2023)	LoRA adapter	per-artist fine-tune
$\mathcal{D}'$	$G(P)$, P the keys of $\mathcal{D}$'s records	same	same
Deployed B	new adapter on M_0	new adapter on M_0	new model
Second architecture	PixArt-α (Chen et al., 2024)	–	–
Audit	CLiD (Zhai et al., 2024) $\rightarrow$ CDI (Dubiński et al., 2025)	CLiD $\rightarrow$ CDI, Min-K%, DE-COP, traps	set-level dataset inference (Proboszcz et al., 2025)
Capability	FID (Heusel et al., 2017), CLIP (Radford et al., 2021)	held-out likelihood	–

condition (i) by design. Qwen2.5-7B-Instruct rewrites each document into the *lyrical* register, the register of that paper's motivating example. We give SDR the same model as its auxiliary model. The rewrite and SDR's inversion search share one vocabulary. We build each setting's rewritten dataset once and freeze it.

B.2 AUDIT AND EQUIVALENCE TEST

Audit. A single audit scores all runs. At the per-example level we score membership with the caption-conditioned likelihood discrepancy of CLiD (Zhai et al., 2024). CDI's one-sided Welch t-test of members against non-members turns those scores into a single decision per dataset (Dubiński et al., 2025). Members, non-members and the reference set are held strictly in-distribution, which rules out a decision driven by a population gap (Liang & You, 2024). The audit of Section 4.1 reflects two corrections. First, an older scorer had produced one setting's direct value and made the two spans look very different. We re-scored all saved checkpoints with the current code, and the two spans are now comparable (0.593 and 0.560). The setting-level constants come from that code. Second, redrawing the member and non-member split adds variance that reseeding cannot capture, since reseeding changes the training noise but keeps the partition fixed. We therefore report intervals over redrawn member sets throughout.

Equivalence test. The models trained without $\mathcal{D}$ form an ensemble, trained on an independent owner's data disjoint from $\mathcal{D}$, and their spread sets each setting's own baseline. A run is equivalent to those models when it passes two one-sided tests (TOST) (Schuirmann, 1987) against the pre-registered margin $[0.46, 0.54]$. Table 6 lists the margin, the per-setting baselines, the laundered intervals and the test. The margin is not the baseline itself. The width of a confidence interval on a mean shrinks as $1/\sqrt{n}$. A baseline used as the margin would make the test impossible to pass as n grows, even for innocent runs. On text, each laundered entry is equivalent to its setting's baseline, while the standard two-sided test still separates the pair ($p = 0.001$ on T1 and $p = 0.026$ on T2). T1's laundered interval does not overlap its baseline, and T2's does.

Margin sensitivity. We fix the margin before the runs. With the margin tightened to ± 0.03, T2 is still equivalent ($p = 0.005$), while T1 fails ($p = 0.286$). The tightest passing margins are ± 0.037 on T1 and ± 0.023 on T2. On the image side, our pre-registration required us to report any measured range wider than the margin. At three runs, the range was wider on three of four settings. This width came from the t multiplier at two degrees of freedom. The widest image range narrows from 0.264 at three runs to 0.053 at 12. The baseline also shifts across settings, with innocent models scoring

Table 6: Baseline ranges per setting, and the equivalence test on the text settings. Each baseline is estimated from that setting's own ensemble of innocent models, 6 runs per text setting and 12 per image setting. All image baselines score worse than random, and contrasts are normalized within setting.

Setting	Baseline	Run spread	Laundered	TOST p
T1 Austen (text)	$[0.470, 0.492]$	$[0.454, 0.509]$	$[0.498, 0.518]$	0.022
T2 Twain (text)	$[0.486, 0.506]$	$[0.472, 0.521]$	$[0.501, 0.518]$	0.0002
X1 Konchalovsky (image)	$[0.380, 0.433]$	–	–	–
X2 Saryan (image)	$[0.419, 0.455]$	–	–	–
X3 Serov (image)	$[0.457, 0.488]$	–	–	–
X4 Korovin (image)	$[0.432, 0.484]$	–	–	–
Equivalence margin, pre-registered and applied to the text settings: $[0.46, 0.54]$				

from 0.380 to 0.488. We normalize within setting, and the range does not affect the main image numbers.

Operating points and calibration. Detection AUC is in Table 1, and TPR at 1% FPR is in the extended table released with LAUNDERBENCH. At half that rate, the laundered entries no longer exceed their own false-positive rate. At 0.5% FPR the TPR on X1 is 0.0053 and on X2 0.0012. X2 is under the 0.005 the threshold allows and X1 matches it. The positive controls trigger at 0.88 on X1 and 0.98 on X2. A tenth of a percent is below what our pools can measure. The largest non-member pool any setting holds is 850, which puts the smallest measurable false-positive rate at 0.0012. The positive predictive values in Section 5 use a threshold calibrated per setting (150 innocent-model scores). This calibration sets the lowest realized false-positive rate at 0.0067, and we report only at or above it. Figure 3 repeats the calculation at each threshold above that rate. At the operating point, the setting where the audit does best has a positive predictive value of 0.61 (base rate 1 in 2). At a base rate of 1 in 100 the value ranges from 0.016 down to 0.000 across settings.

B.3 SEEDS AND COMPUTE

Text detection entries use $n = 3$ seeds per setting, and the baseline and laundered rows use $n = 6$. The similarity-score regression uses a grid of 54 runs.

The 8.1 GPU-minutes of Section 3.1 is wall-clock time on the image laundering step only, on one NVIDIA L40S. It covers 6.6 minutes to train G at the 3000-step budget, 0.2 to load it and 1.3 to regenerate $\mathcal{D}'$. The pass trains a rank-48 adapter on SD1.5 over a setting's 50 licensed images for 3000 steps (batch size 1, learning rate 10^{-4}), then samples 100 images at 512 pixels. Each stage is low-rank adapter fine-tuning or sampling. The figure excludes the audit (a licensor cost) and the training of a deployed B, which a licensee does regardless. The dollar figure in Section 3.1 is computed. It takes 8.1 GPU-minutes at $1.861 per GPU-hour ($0.25 in total). The rate is the AWS g6e.xlarge on-demand list price in us-east-1 as of 4 August 2026. The text generator is timed separately in Table 7. The same pass over the 500-member dataset runs 15.26 GPU-minutes, and only 0.61 of that trains G, since generation takes most of the time in the text pass. We did not time the audio generator, and each cost here is for one modality.

Cost scales with the generator budget and the size of the dataset regenerated. The number of deployed B does not affect the cost. Table 7 measures both axes. Image training is linear in generator steps, at 2.2 minutes per thousand steps at 512 pixels and 2.6 at 768. Regeneration is linear in the size of $\mathcal{D}'$, at 0.85 seconds per image at 512.

B.4 CAPABILITY METRICS

We measure capability independently of each audit we score. Image capability is held-out FID (Heusel et al., 2017) and CLIP score (Radford et al., 2021). We compute FID under clean-fid (Parmar et al., 2022). Resizing conventions shift FID further than any effect we report. A laundered B remains within 0.04 to 0.78% of a directly trained B_{direct} on FID, and within -0.82 to 0.37%

Table 7: How the laundering pass scales. Wall-clock GPU-minutes on one L40S. Image entries vary the generator budget, the size of $\mathcal{D}'$ and the output resolution. Text entries vary dataset size and budget, and for them the resolution column holds the dataset size $|\mathcal{D}|$. The 3000/100/512 row replicates the entry behind Section 3.1 and is within 2 percent of it. Cost follows the generator budget and the size of the regenerated dataset, and the number of deployed models does not affect it.

Modality	Steps	Size of $\mathcal{D}'$	Resolution	Train	Generate	Total
Image	1500	100	512	3.37	1.39	4.94
	3000	100	512	6.59	1.45	8.24
	6000	100	512	12.35	1.43	13.99
	3000	200	512	6.60	2.78	9.55
	3000	400	512	6.51	5.67	12.35
	1500	100	768	4.11	2.81	7.12
	3000	100	768	7.65	2.82	10.64
	3000	200	768	7.91	5.78	13.88
Text	500	100	50	0.60	2.94	3.55
	500	500	500	0.61	14.64	15.26
	1000	500	500	0.99	14.71	15.78
	2000	500	500	1.61	14.71	16.32

on CLIP. Both gaps are inside the seed-to-seed spread. We did not run a preference study. Two automatic metrics carry the image capability claim of Section 4.3. We check $U(B) \geq \tau$ at each point along the laundering-budget axis of Table 3, not only where detection is lowest. CSD is not a capability metric here. It is an audit in its own right, and one that cannot attribute a model to $\mathcal{D}$, as Appendix D shows. Scoring capability with it would make the detection-capability tradeoff circular. Text capability is the mean per-token log-likelihood, in nats, that a model assigns to the licensed author's held-out passages.

C Proofs

Section 3.2 states Theorem 1 with its two conditions in brief. This appendix states the two conditions in full and the bound behind the separable branch as Proposition 1, and proves the theorem.

The auditor seeks one bit, i.e., the membership $M_x \in \{0, 1\}$ of a candidate record x. The audit takes the deployed model B and the public candidate x and returns a decision V, and $\mathcal{A}(B, \mathcal{D})$ aggregates these decisions over candidates. Throughout we work at L1 or above, where $\mathcal{D}'$ is sampled from G and each of its records is a fresh draw.

The two conditions in full. *(i) Regeneration bottleneck.* G is trained on $\mathcal{D}$, $\mathcal{D}'$ is generated by G under prompts held out from $\mathcal{D}$ and drawn independently of M_x, and each deployed B is trained only on $\mathcal{D}'$. We draw $\mathcal{D}$ so that a record belongs to it exactly when $M_x = 1$, with everything else fixed, and V is a function only of B and public inputs. V then depends on M_x only through $\mathcal{D}'$. *(ii) Shared capability.* The capability can be reached without $\mathcal{D}$: there is a model B_{neg}, trained on data disjoint from $\mathcal{D}$, with $U(B_{\text{neg}}) \geq \tau$. Any component inseparable from the capability is needed to reach it and is therefore present in every model that reaches it, B_{neg} among them.

Proposition 1. *At L1 or above, hold a candidate x fixed and let $\mathrm{TPR}^{\star}_{\mathcal{D}'}(\alpha)$ be the power of the best test of $M_x = 1$ against $M_x = 0$ built from $\mathcal{D}'$ at false-positive rate α. For every test ϕ built from V at false-positive rate α,*

$$\mathrm{TPR}(\phi) \;\leq\; \mathrm{TPR}^{\star}_{\mathcal{D}'}(\alpha).$$

This is the post-processing property of hypothesis testing. Under condition (i), V is a garbling of $\mathcal{D}'$, and a garbling cannot improve any test (Blackwell, 1953; Dong et al., 2022).

Proof of Theorem 1. By the definition in Section 3.2, split any signal an audit measures on a laundered B into a separable component and an inseparable remainder.

Inseparable branch. By the definition of an inseparable signal and condition (ii), the component is present in B_{neg}, a model trained without $\mathcal{D}$, and is therefore not attributable.

Separable branch. By Proposition 1, $\mathcal{D}'$ bounds what a decision can recover about M_x from this component. Where $\mathcal{D}'$ carries none, $\text{TPR}^{\star}_{\mathcal{D}'}(\alpha) = \alpha$ at every α, and every B trained on $\mathcal{D}'$ scores at the baseline. The separable component is recoverable only to the extent that $\mathcal{D}'$ keeps it, and Section 4.3 measures that residual. □

A sequence of regenerations is a further post-processing of $\mathcal{D}'$ and cannot raise power at a fixed α beyond what $\mathcal{D}'$ supports. MADreMIA (Section 5) is such a sequence. On text we measure condition (ii) directly. A model trained on a different author of the same period reaches 0.917 of the directly trained model's gain on T1.

D Audit Reimplementations

D.1 Details on Each Audit

Table 8 lists the signal each audit uses and the strongest published laundering-aware form we give it, following adaptive-evaluation practice (Tramèr et al., 2020).

Choice of audits. We test each family with its strongest published audit. CDI is the strongest set-level membership audit we found for diffusion models. We include copyright traps because their own paper finds that membership inference without traps is barely better than random on a medium-size model. A laundering meta-detector, a classifier that asks only whether $\mathcal{D}'$ was laundered, is not a valid audit here, because at this sample size a linear probe separates any two generator instances, including two seeds of the same regenerated dataset.

Two deviations. Two audits were not run exactly as published. We run the original Min-K% and leave out the normalized Min-K%++ variant (Zhang et al., 2025c). Our from-scratch models do not follow instructions. We run the logit-ranked variant that the DE-COP paper describes for such models.

Baselines. We compare each audit with its own baseline. Min-K% scores 0.626 on T1 and 0.591 on T2 for the directly trained models (baselines 0.502 and 0.518). DE-COP scores 0.574 (baseline 0.515) and drops to 0.528 under laundering. The baselines are 0.429 for dataset inference and 0.529 for CDI. CDI runs on the authors' released code, unmodified, on the same frozen member and non-member split as all other audits here (3 seeds per entry, $n = 50$ per side). For copyright traps, a scan for 8-word shingles shows that the injected trap sequences are absent from $\mathcal{D}'$, and the trap audit cannot trigger.

Table 8: Audit reimplementations. What each audit scores and the adaptive, laundering-aware form given to it. Scores are in Table 1.

Audit	Family	Signal scored	Adaptive form given
Copyright traps	marks	injected trap sentences	full-strength detector on trapped text
DIAGNOSIS	marks	detection rate of its warp detector	authors' warp and test, unconditional
CLiD	membership	caption-conditioned likelihood gap	early stopping, augmentation
CDI	membership	set-level diffusion features	authors' released code, unmodified
Min-K%	membership	lowest-probability token mean	from-scratch base with known membership
Dataset inference	membership	pooled per-sample margins	post-hoc (Zhao et al., 2025), STAMP (Rastogi et al., 2025)
DE-COP	membership	verbatim over paraphrase ranking	reported black-box accuracy
SSCD	style	near-verbatim image copies	full-strength copy detector
CSD	style	global style resemblance	high $\mathcal{D}$ score, no attribution

Dataset inference keeps a separable residual. The laundered conditioned test still triggers in 9 of 15 runs, while the caption-blind loss variant scores inside the baseline range. Membership survives distillation (Jagielski et al., 2023), but distillation is not the mechanism here. B trains on a regenerated dataset that lacks the specific members of $\mathcal{D}$. The per-example test thus drops to the near-0.5 baseline that these audits already report against same-distribution non-members (Maini et al., 2024). Neither adaptive variant changes this. Post-hoc dataset inference (Zhao et al., 2025) removes only the held-out requirement, and a launderer has no reason to train on the published watermarked version of STAMP (Rastogi et al., 2025).

CLiD keeps a residual that follows generator memorization. CLiD also remains above the baseline range (0.245 on X1 and 0.245 on X2 on the normalized scale). Of 24 laundered runs, 11 still reach $p < 0.05$. The branch is again separable. The conditional-overfitting gap opens only when the exact image-caption pairs of $\mathcal{D}$ are in training, and B gets generated pairs. The residual therefore comes from generator memorization only. It increases with the generator budget in Table 3, and Table 12 shows it surviving labeling schemes we did not pick. This family's reliability was already disputed (Liang & You, 2024; Das et al., 2025; Duan et al., 2024).

CSD does not separate the three models. CSD attributes none of the three models to $\mathcal{D}$. It scores 0.571 on the directly trained model, 0.535 on the laundered model and 0.537 on the baseline. This spread is too narrow to separate a user of $\mathcal{D}$ from an innocent party even before laundering. A single cross-modal L3 step takes its raw score to the baseline (0.490). The resulting $\mathcal{D}'$ does not reproduce the style features that CSD measures. This step is lossy, with held-out style margins of -0.078 on X1 and 0.078 on X2, and we do not depend on it. SSCD, the copy-detection half of the same family, drops from 0.622 on the directly trained model to 0.550 under laundering, still above its baseline (0.414).

D.2 Audits Not Run

The census includes twelve audits we could not run, in three groups. The first group is the injected marks we did not instantiate. Radioactive data (Sablayrolles et al., 2020) marks the training images of a classifier. We injected data isotopes (Wenger et al., 2024) but could not score them. The detector learned to separate the settings by artist. We did not inject the domain and dataset watermarks (Guo et al., 2023; Li et al., 2022; Wei et al., 2024), the diffusion watermarks (Meintz et al., 2025) or SIREN's coating (Li et al., 2025a). The second group is ownership resolution, which needs a training-time hook or a verified training trace (Li et al., 2025c; Jia et al., 2021; Choi et al., 2023). The third group is contamination and use-audit tests, which need the licensed dataset in hand and not only a deployed model (Oren et al., 2024; Huang et al., 2024; Du et al., 2025a). None of them has a row in Table 1.

We attempted two of these tests. The exchangeability contamination test does not trigger even on its own positive control, at $p = 0.650$ under direct training, and we therefore list it as uninstantiated. TRACE (Zhang et al., 2025b) rewrites the licensed dataset with a sampler seeded by a secret key, so that the released text is an exact sample from the rewriter. It then detects the watermark that the rewrite leaves in a fine-tuned model. We could not establish that transfer on our base, and count TRACE the same way. Dataset usage cardinality inference estimates what fraction of a dataset was used, from per-record membership guesses (Tong et al., 2025). The estimate is a proportion, which the census definition excludes. The audit depends on the per-record membership signal that Section 4.2 takes to the baseline.

The count excludes 3 output watermarks. Tree-Ring (Wen et al., 2023), Stable Signature (Fernandez et al., 2023) and Gaussian Shading (Yang et al., 2024) all watermark the output of a generator the defender controls, and by design they do not reach a model trained on $\mathcal{D}'$. The autoregressive radioactive signal is excluded as inseparable, and thus not attributable. Counting by family and by modality leaves exactly 9 non-cooperative training-data audits instantiated and measured, matching Section 4, with the uninstantiated audits counted separately.

Audio is a third modality and a generality check, counted outside the two main modalities. The generator is a per-artist fine-tune of MusicGen (Copet et al., 2023), an open autoregressive music model. Across 3 settings of distinctive artists at a matched training budget, the unconditioned content audit triggers on the directly trained model (AUC 0.78). On the laundered model it scores at the

Table 9: Per-audit predictions from the laundering-attribution tradeoff, scored. The branch prediction holds on all eight audits that have one, and 14 of 16 branch-and-rung predictions hold, with CLiD and dataset inference the two misses. Contamination is one of two attempted audits among the twelve uninstantiated (Appendix D).

Audit	Family	Predicted rung	Measured	Rung prediction
Separable branch				
Copyright traps	marks	L1→L2	L2	held
CLiD	membership	L1→L2	L2, residual	missed
Min-K%	membership	L1→L2	L2	held
Dataset inference	membership	L1→L2	L2, residual	missed
DE-COP	membership	L1→L2	L2	held
SSCD	style	L2	L2	held
Audio set-level dataset inference	membership	L2	L2	held
Contamination	membership	L1→L2	–	not scored
Inseparable branch				
CSD	style	L3 cross-modal	L3	held

baseline of models trained without the artist (0.55), with the members absent from $\mathcal{D}'$ (the separable branch). Small per-artist datasets limit the capability the audio models keep. Audio is evidence of evasion only, and the main results leave it out.

D.3 Pre-Registered Predictions

We pre-registered each prediction before the run that scores it, with its criterion and the settings it would be judged on. A prediction held when it was met on all of those settings, was partial when it was met on some, and failed otherwise. The first level contains the per-audit predictions of the tradeoff, listed in Table 9. The branch is right on each instantiated row, and the rung on all but CLiD and dataset inference, and 14 of 16 held. The second level contains our own hypotheses about why the attack works, of which 4 of 10 held. Over both levels, 18 of 26 held, 6 failed, and 2 held on one setting or one audit. The structural predictions held, and most of the mechanism hypotheses did not. The clearest failure was that a register rewrite evades the audit in Table 2 just as our bottlenecked operation does. Neither the twelve uninstantiated audits of Appendix D nor the failed predictions point to an audit outside the two branches.

Three exploratory runs failed their own controls, and none is reported as a finding. A false-positive probe on an innocent generator reached significance on neither setting and disagreed in sign between the two. A second regeneration step lost its positive control along with the residual, and decay could not be separated from audit failure. A contrast between settings turned out to depend on one partition draw.

E Additional Results

E.1 Testing the Bottleneck Bound

We compare the audits run on B with one test on $\mathcal{D}'$ over one candidate set, shown in Figure 2. Proposition 1 fails if any audit run on B outperforms the best test on $\mathcal{D}'$ at any operating point. Our test is the ROC of ϱ, computed on $\mathcal{D}'$ only. Since it is a lower bound on the best test, an audit can exceed it without violating the proposition. On the two positive controls it reaches 1.000 under surface perturbation and 0.980 under register rewrite. For our operation, ϱ is at 0.5 on both settings (0.493 on T1 and 0.500 on T2). On each rewrite it falls between 0.977 and 0.995, and it recovers the exact source member for 0.986 to 1.000 of the rewrites. Over both settings and all three datasets, 40 entries of 42 lie at or below the 97.5th-percentile bootstrap range of ϱ. The two exceptions exceed ϱ by at most $+0.003$, at a false-positive rate of 0.95 on the laundered panel, where ϱ is itself 0.5. Under register rewrite the dataset keeps nearly everything while each census audit scores 0.5, and ϱ over-estimates what an audit will recover.

Table 10: One regenerated dataset, two architectures. The SD1.5 generator of Section 4 produced the dataset, and the PixArt row changes both the denoiser and the text-encoder family, with $n = 6$ seeds per entry. Laundered detection is on the normalized scale (0 is the baseline, 1 is direct training) with its 95% interval. A residual survives on SD1.5 and none on PixArt-α. All models share one dataset. The table therefore shows reuse across architectures and cannot show variation across datasets.

Deployed model	Innocent	Direct	Laundered (normalized)	95% CI	Capability kept
SD1.5 (UNet, CLIP)	0.415	0.999	0.204	$[0.169, 0.239]$	0.34
PixArt-α (DiT, T5)	0.540	1.000	0.004	$[-0.056, 0.064]$	1.03

Table 11: The base-scale study on text. The study runs four base scales by two settings by generator budget. Detection increases with generator budget in 8 of 8 scale-by-setting entries. The three right columns are generator budgets in steps, with detection normalized within each scale. The text result holds at each scale.

Base size	Setting	Innocent	Direct	1000 steps	2000 steps	3000 steps
114M	T1	0.481	0.914	0.105	0.263	0.342
	T2	0.496	0.815	0.080	0.203	0.271
300M	T1	0.504	0.997	0.086	0.147	0.222
	T2	0.519	0.989	0.042	0.064	0.120
800M	T1	0.491	1.000	0.127	0.179	0.305
	T2	0.538	1.000	0.028	0.135	0.159
1.4B	T1	0.504	1.000	0.089	0.252	0.352
	T2	0.509	1.000	0.136	0.211	0.279

E.2 Reusing One Regenerated Dataset Across Architectures

One SD1.5 generator produces a single $\mathcal{D}'$ at the operating point of Section 4, and 12 models over two architectures (6 seeds each) train on that same frozen file. The second architecture is PixArt-α (Chen et al., 2024), a diffusion transformer with a T5 text encoder. Detection scores are not comparable across denoiser families. We normalize each architecture by its own baseline and directly trained value (a gap of at least 0.460). On the generator's own architecture the audit recovers 0.204 of the span in Table 10. On the transformer it recovers 0.004 (95% CI $[-0.056, 0.064]$, containing zero).

We measure capability as a rank against 12 artist centroids and not as a margin against one contrast artist. A model that saw none of $\mathcal{D}$ still places the licensed artist below its nearest neighbor in style space. The laundered transformers place it at rank 1.5 of 12 (4.0 for the innocent control). Their held-out style margin reaches their own directly trained value (1.03), and on this $\mathcal{D}'$ the SD1.5 models keep 0.34 of theirs. PixArt mimics more weakly in absolute terms (direct value 0.088 against 0.289 for SD1.5). Keeping all of a smaller direct value is not the same as matching SD1.5. Freezing one dataset also removes the averaging over datasets that a seed-matched design would have provided.

E.3 Laundering Across Base Model Scales

We rebuild the from-scratch pipeline whole at each base scale of Table 11, with the setting, licensed dataset, deployed budget, seeds and capability protocol unchanged ($n = 3$ seeds per entry). At the one generator budget that all four scales share, the laundered residual falls between 0.028 and 0.136 on the normalized scale, and is not monotone in scale. The residual increases with generator budget in 8 of 8 scale-by-setting entries, while the capability kept remains flat. The table lists per-scale entries only. Comparing across scales at matched generator memorization would require interpolating between generator budgets.

Table 12: Four labeling schemes, from most to least specific per image. Each is normalized to its own baseline and directly trained value, at 3 seeds per entry. The audit detects direct training under each scheme, and laundering suppresses detection under all four, even with labels we did not write.

Scheme	Labeling	Direct X1	Direct X2	Laundered X1	Laundered X2
R0	Rare token (ours)	0.999	0.997	0.245	0.245
R1	Per-image caption	0.999	0.998	0.070	0.167
R2	Generated captions, 10 shared strings	1.000	1.000	0.441	0.362
R3	One shared caption	0.999	1.000	0.286	0.299

E.4 Ablations

We vary one factor per study, hold capability at τ, and measure detection over the interior of the grid. The factors are the capacity given G, its overfit budget in update steps, how many outputs of G go into $\mathcal{D}'$, the L2 adapter rank, the balance between free generation and prompt-conditioned purification of $\mathcal{D}$, the sampling guidance and temperature, and the number of regeneration steps, one study each. Over the interior of all seven, each non-cooperative audit but the two residual cases of Appendix D scores inside the pre-registered range $[0.46, 0.54]$. The baseline range is not tuned to one configuration.

Only the generator budget changes the outcome. Detection increases by $+0.101$ as that budget goes from 500 to 4000 steps. It goes from 0.510 to 0.611 over the range of Table 3, while held-out CLIP remains between 28.2 and 27.5. The launderer undertrains the generator (also the cheaper option).

Perturbation alone does not evade. Training B on an augmented copy of $\mathcal{D}$, with crops, flips and color jitter only, leaves detection at the direct value (0.995 on X1 and 0.997 on X2). Near-copies of the members remain in that dataset. Of the operations we run, only the regeneration bottleneck gives low detection at full capability.

The result holds under other labeling schemes. In our dataset, each image carries its own rare token as a caption (convenient, and unlike any real dataset). We rebuilt the attack under the four labeling schemes of Table 12, from most to least specific per image. R0 is ours. R1 gives each image one caption from an off-the-shelf captioner, R2 clusters those down to 10 shared strings, and R3 gives the whole dataset a single caption. Low laundered detection under a scheme could mean only that the audit fails under that scheme. We give each scheme its own baseline and directly trained value and normalize within the scheme. The direct values rule that out, at 0.997 or better under each scheme, including the single caption. R2 keeps the largest residual of the four, at 0.441 on X1 and 0.362 on X2. This is because a caption shared by many images cannot identify a particular member, and the audit no longer relies on labels. The remaining signal is content memorization carried through the generator (see Section 4.3).

The smallest budget splits the settings in two. At the smallest generator budget, the four settings, including X3 Serov and X4 Korovin, fall into two groups. X1 and X3 remain inside their baseline ranges of Table 6, at normalized detection $+0.044$ and -0.023. X2 and X4 score $+0.129$ and +0.059. We report the split as measured across four settings, and have not isolated its cause.

E.5 DIAGNOSIS Image Watermark

This is the strongest position the paper gives an auditor, and the only step tailored to one audit. The licensor adds the signal to each item before release and keeps the detector. The launderer knows the watermark's family and uses the artist's public work from outside the license. DIAGNOSIS runs as published, in its unconditional variant. All licensed images are warped by one fixed field at warping strength 2.0 (its default). A ResNet18 detector is trained to tell coated images (i.e., those carrying the warp) from clean ones. A model is declared to have used the dataset when its hypothesis test triggers on 50 outputs at DIAGNOSIS's own thresholds $\tau_{\mathrm{D}} = \gamma = 0.05$, with β the detector's held-out false-positive rate. We report the detection rate α (i.e., the fraction of outputs the detector calls

Table 13: The launderer's steps against DIAGNOSIS's injected watermark, scored by its own detector. Per-image steps leave the audit triggering. Only removing the estimated field reaches the pre-registered threshold of one in three runs, with or without dilution.

Setting	Launderer's step	Detection rate α	Runs rejected	Held-out mimicry
X1	None (laundering alone)	0.88	3/3	0.255
	JPEG, quality 50	0.91	3/3	0.238
	Per-image regeneration	0.69	3/3	0.206
	Dilution alone	0.19	2/3	0.227
	De-warp alone	0.08	1/6	0.238
	De-warp and dilution	0.00	0/6	0.209
	De-warp and dilution, 6000 generator steps	0.01	0/3	0.235
X2	None (laundering alone)	0.72	3/3	0.252
	De-warp alone	0.11	1/3	0.232
	De-warp and dilution	0.00	0/3	0.270

coated) and the count of runs on which the test triggers. An entry defeats the audit only if at most one in three of its runs trigger, a threshold pre-registered before any de-warped run was scored. We measure capability as held-out mimicry r_{mimic}, against the baseline of the setting.

Laundering alone reproduces the watermark. On X1, direct training on the coated dataset scores $\alpha = 0.66$ (one seed) and laundering alone 0.88 (three seeds), with the innocent models at 0.00. On X2 the same three values are 0.92, 0.72 and 0.01. The generator is trained on a dataset in which each item carries the field. It renders the field into $\mathcal{D}'$ as it renders style. Table 13 shows that cleaning the images does not help. After JPEG (quality 50) or per-image regeneration, the detector classifies the training images as clean, yet the laundered model still triggers on each seed.

The step that works acts on the whole dataset. The launderer trains a warp detector and a displacement-field regressor on clean images, each warped with random fields of the same family. The clean images are public works by other artists and unmarked works of the licensed artist from outside the license. The launderer then uses the regressor to estimate the one field that the licensed images share. Removing that estimate takes about 20 GPU-minutes. It leaves a residual warp of 0.70 px on X1 and 0.75 px on X2, out of the 1.27 px injected, measured against the true field (unknown to the launderer). The de-warped images are then diluted with 150 unmarked works of the same artist (coating rate 25%). This rate is inside the range DIAGNOSIS's own evaluation covers, where Wang et al. (2024) report that the unconditional variant needs a high coating rate.

No run triggers after de-warping and dilution. De-warping alone takes the audit under its own threshold on both settings, with one seed per setting still triggering. With dilution added, zero of the twelve runs across both settings and both generator budgets trigger. Held-out mimicry is 0.209 on X1 and 0.270 on X2, against baselines of 0.082 and 0.138, and against 0.255 and 0.252 for laundering without these steps. The estimator is specific to a shared geometric field, and a watermark of another family needs its own estimator. This section claims only the principle that a transform applied to each licensed item is a statistic of the dataset, which the launderer can estimate without the auditor's detector.

A per-item watermark survives on one setting. Such a watermark is not a statistic of the dataset. The estimator then has no target. We test one field per image. On X2 the watermark survives laundering with zero controls triggering, and dilution to a 25% coating rate leaves mean detection at 0.55 to 0.90 across generator budgets, with each seed triggering and capability unchanged. On X1 the detector this needs also triggers on 4 of 9 control models trained without the licensed images.

E.6 Secret-Key Text Watermark

A watermark of another family needs its own audit. We built a secret-key text watermark with its audit and ran the same attack against it. We follow the green-list construction (Kirchenbauer et al., 2023). The licensor rewrites each licensed passage so that a pseudorandom half of the vocabulary

Table 14: The similarity score across the four rewrite registers. The score needs only the regenerated dataset, and testing a register needs only its dataset. Top-1 source recovery ranges from 0.986 to 1.000 across all 4 registers.

Operation on $\mathcal{D}$	**T1**	**T2**
Surface perturbation	1.000	1.000
Rewrite, *paraphrase*	0.992	0.992
Rewrite, *simplify*	0.979	0.993
Rewrite, *summarize and expand*	0.981	0.977
Rewrite, *lyrical*	0.980	0.995
Laundered (ours)	0.493	0.500

(the green list, chosen by the secret key) is sampled more often at each high-entropy position, and keeps the key. The audit samples the deployed model and tests whether its output over-uses the green half. The audit is ours, and the census includes only published audits. We report it here and leave it out of Appendix D.

We test two variants at the ends of the design range. Subword-level marking reaches 33 percent of tokens and garbles words, which degrades the generator and costs the launderer capability. Whole-word marking reaches 12 percent and leaves the text fluent. Both variants transfer to a model trained directly on the watermarked dataset, where the audit triggers at $z = 4.2$ to 5.3.

Laundering defeats the per-model audit under both variants, on both settings and at both generator budgets. Raising the auditor's budget to 1000 sampled outputs does not recover the signal. Deduplication saturates near 3,700 scored pairs.

We also measure one residual. When the auditor pools runs and decides once for all of them, the regenerated dataset keeps a green excess of 0.0312 over a clean laundered run scored under the same key ($z = +3.56$), against 0.0455 for the direct run. Between a third and two thirds of the signal survives in aggregate, while the audit flags zero models individually. The comparison must be against a clean laundered run and not the nominal green fraction γ. An innocent model scores 0.505 to 0.511 under the true key. This offset comes from pairs that are frequent in any English text.

F Positive and Negative Controls

Confidence intervals over the seeds of Appendix B accompany each control.

F.1 Similarity Score

$\varrho(d)$ is the similarity of a candidate document to its nearest match anywhere in $\mathcal{D}'$, and ϱ is the AUC of that similarity for members against non-members. Both are computed from $\mathcal{D}'$ only. Table 14 lists it across all 4 rewrite registers. We built *summarize and expand* as the hardest case. It reduces each document to two sentences and expands it again, which makes its per-document bottleneck as narrow as a rewrite can make it. Yet it scores close to a simple paraphrase. Narrowing the per-document channel leaves ϱ unchanged, while removing the channel lowers it. Condition (i) therefore states a structural condition without a quantitative threshold.

Our settings break condition (i) of Theorem 1 in a way that helps the audit. Items of $\mathcal{D}'$ are generated from the caption or title of one licensed record, which G saw together with that record in training, and what G memorized of the record can reach $\mathcal{D}'$. Detection then tracks the near-duplicates that survive in $\mathcal{D}'$. Within setting across the 54-run grid, partialling out the injection fraction, the rate of surviving near-duplicates tracks realized detection at +0.476 (95% CI $[+0.219, +0.678]$, Figure 4). Pooling a three-predictor regression over all three audit families at once returns a much larger correlation. This is a Simpson confound, because the families lie in disjoint regions of the plane and the pooled statistic mostly picks up between-family separation.

Dataset-level nearest-neighbor distance from $\mathcal{D}'$ to $\mathcal{D}$, which we had projected as the strongest predictor, does not survive partialling (+0.263, 95% CI $[-0.019, +0.523]$, containing zero). The

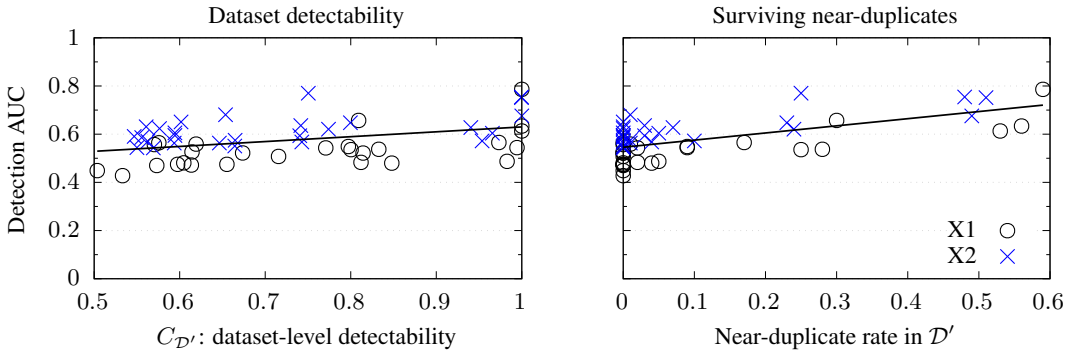

Figure 4: The similarity score against realized detection. One point per run of the 54-run grid. The partial correlation controlling for injection fraction is +0.263 for dataset detectability (left) and +0.476 for the near-duplicate rate (right), and the line is the uncontrolled fit. Only surviving near-duplicates predict detection. Collapsing duplicated content at fixed dataset detectability changes detection by -0.145 and -0.126.

near-duplicate count, projected as the weakest, survives. Holding the distance fixed and collapsing the duplicated content changes detection by -0.145 on X1 and -0.126 on X2, which shows that the relation is causal. Yao et al. (2025) report the same failure for distance-to-closest-record privacy screens, and we reproduce it on a laundering pipeline. An audit adapted to ϱ would not stop the launderer, who can recompute ϱ and launder again whenever it scores high.

In the text modality, we accept a similarity backend's negative report only if it first recovers the known source pairing of the rewrite dataset, where the answer can be checked. A lexical similarity recovers that pairing at 0.999 and a sentence embedding model at 0.816. Under the second, our dataset scores 0.490 on T1 and 0.516 on T2, against the rewrite's 0.700 and 0.831. Four poolings of a 7B instruct model recover between 0.024 and 0.146 of the known pairing, and we do not report their scores.

G Details on the Defenses

Section 5 evaluates a lower threshold, SDR and MADreMIA. This appendix covers the other three defenses, lists the configuration and the intervals behind SDR and MADreMIA, and states why we do not run DIPBox.

Three more defenses. These three defenses search for a separable signal that is absent from $\mathcal{D}'$. **A smaller candidate set** does not help a licensor auditing a single artist. Subsampling the stored per-image scores shifts detection by at most 0.008 anywhere between 5 and 50 candidates, on each setting. **Labels the launderer does not choose** do not help either. Under four labeling schemes, normalized detection remains between 0.070 and 0.441 in Appendix E.4, well below the direct value, and the audit detects direct training under all four. **A filter on** $\mathcal{D}'$ rejects datasets on a statistic that does not track detection, as Appendix F shows. The launderer has $\mathcal{D}'$ before any filter does and can regenerate until a dataset-side filter no longer rejects $\mathcal{D}'$.

Transform reversion (SDR). SDR (Li et al., 2026) needs $\mathcal{D}'$ to be a sample-by-sample transform of $\mathcal{D}$, and our $\mathcal{D}'$ is drawn from G with no document as input. We give SDR more than its own paper does, the full search over registers, including the true synthesis register, and a distributional two-sample membership test. SDR's recovery on the laundered B is 0.508, against 0.877 on the positive control. The 95% CI of the laundered value is [0.467, 0.549], which contains the 0.542 that its reversion scores on an innocent model. SDR works in part where its assumption holds. On the register-rewrite condition of Table 2, where the launderer edits each document in place and does not regenerate, it returns 0.648, above its own innocent-model score and well below the positive control. Any higher recovery would make the signal inseparable from the capability, which is the

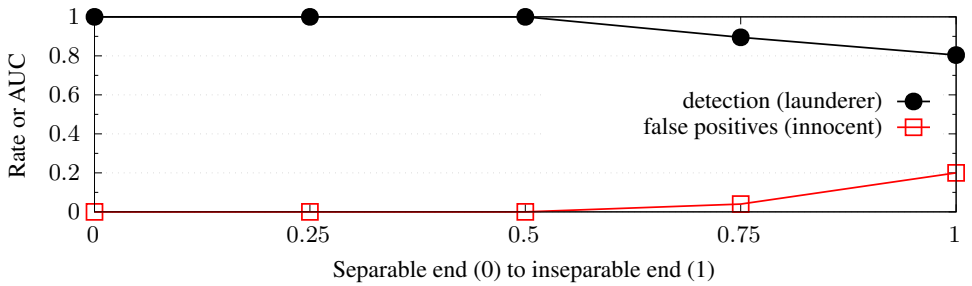

Figure 5: Detection specificity of a laundering-robust defender. Over 6 laundered runs against 25 innocent models, detection decreases from 1.000 at the separable end to 0.804 at the inseparable end, while false positives on innocent same-style models increase from 0.00 to 0.20. The run counts are small, and we report only the direction.

second branch. In Figure 5, innocent false positives increase from 0.00 to 0.20, while detection on the launderer does not increase with them.

Iterative regeneration (MADreMIA). MADreMIA (Łapacz & Pawlak, 2026) reports that under iterative regeneration, memorized samples decay more slowly than non-members, and that low-FPR membership evidence can be extracted from that gap. Across 71 runs at 4 regeneration strengths, it detects direct training in each entry, at 0.96 TPR for a 1% FPR. The laundered settings score 0.0120 on X1 and 0.0235 on X2, against baselines of 0.0000 and 0.0200. Each laundered interval overlaps its baseline. The trajectory is a property of the image, with a correlation of 0.92 between an innocent model and a laundered one, and it measures image difficulty.

Regeneration tests (DIPBox). DIPBox scores a suspect model and dataset against the licensed dataset at three scales, and finds that a regenerated dataset cannot keep utility without leaving a signal its tests detect (Dong et al., 2026). Two differences keep that result from reaching our setting. DIPBox's strongest configuration needs a sample of the suspect dataset, and our auditor has only a deployed model. The negative datasets come from another distribution, and DIPBox finds that those negatives keep less capability than its regenerated datasets. The right negative is a model of the same capability trained without the licensed data. Our protocol includes that control, and Section 4.3 measures the false-positive rate on it.