FOOLED BY FINE-TUNING: WHY MEMORIZATION AUDITS OF CODE-COMPLETION MODELS NEED CONTROLS

Anonymous authors
Paper under double-blind review

ABSTRACT

Memorization audits of code models commonly inject canaries into training data and measure how strongly the resulting model prefers them, often without evaluating comparable non-members. We show that this practice can produce convincing but misleading conclusions. Through a controlled audit of fine-tuned code models with more than $11{,}000$ canaries, multiple attacker families, checkpoints, and controls, we identify three recurring failure modes. First, large reference-model exposure scores do not necessarily distinguish injected canaries from never-injected controls, even with a matched-corpus reference. Second, a completion-based likelihood-ratio membership test can respond similarly to members and non-members, effectively detecting fine-tuning rather than membership. Third, extraction guarantees can be substantially overstated when confidence bounds are computed for a different estimand than the security claim being made. Positive-control experiments confirm that these findings are not simply failures of the auditing pipeline. We further revisit the information-theoretic certificate motivating our audit. We correct a vacuous additive bound formulation, but show that the pointwise likelihood ratios computed by reference-ratio audits do not directly estimate the divergence required by such a certificate; accordingly, we make no certification claim. Our results motivate a stricter methodology for memorization auditing based on non-member controls, matched references, per-instrument positive controls, and estimand-aware reporting.

1 INTRODUCTION

Code-completion models are trained on corpora that may contain credentials, proprietary fragments, and insecure programming patterns, and prior work has shown that language and code models can reproduce their training data (Carlini et al., 2021; Nasr et al., 2025; Huang et al., 2024; Lukas et al., 2023). The standard instrument for quantifying this risk is the *canary-based memorization audit*: synthetic secrets are injected into the training data, the trained model is scored or queried on them, and the resulting signal is read as evidence of memorization or leakage (Carlini et al., 2019). Because audits of this kind increasingly inform decisions about model release and data governance, the validity of the inference from audit statistic to security claim matters as much as the strength of the attack.

This paper shows that the inference can fail in ways that are easy to miss. We set out to build LEAKCERT, an information-theoretic audit meant to provide a query-budget-aware *certificate* of leakage. We fine-tuned Qwen2.5-Coder-1.5B on a 500K-document code corpus containing $11{,}566$ canaries of four types and evaluated it with exposure, membership, and extraction attacks. The uncontrolled results were compelling: mean exposure exceeded the panel entropy by a factor of two, a sequential likelihood-ratio test flagged 99% of members after roughly one query, and verbatim extraction was essentially zero. Each of these conclusions changed once the corresponding control was added (Figure 1).

Three failures exposed by controls. *First*, large reference-model exposure scores do not distinguish injected canaries from structurally matched canaries that were never injected. Separation stays at chance (AUC 0.4855) even when the reference is retrained on the identical corpus with the canaries removed, at a cost of 173 GPU-hours (Section 6.3). A large raw score can therefore reflect

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.

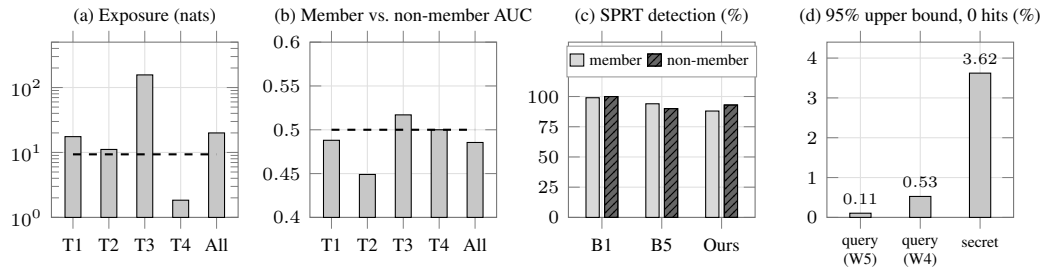

Figure 1: Three conclusions of an uncontrolled audit and the controls that overturn them. **(a)** Mean reference-model exposure by canary type (log scale); the dashed line is the panel entropy $H(K)$. **(b)** The same statistic separates members from never-injected non-members at chance for every type (AUC under the matched-corpus reference; dashed line at chance). **(c)** A completion-based sequential likelihood-ratio test flags non-members as often as members under every defense. **(d)** For the same zero-hit extraction data, the 95% upper bound depends on whether the statistical unit is the query or the secret.

effects shared by members and non-members rather than canary-specific memorization. *Second*, a completion-based sequential likelihood-ratio test that appears to identify members accepts non-members at the same rate (Section 6.4). The test responds to the likelihood shift induced by fine-tuning, not to membership; without a false-positive control it would pass as a successful membership attack. *Third*, extraction confidence depends on the estimand. Bounds computed with the query as the unit answer whether a random prompt succeeds, whereas the security claim of interest is usually whether a random *secret* is extractable. Using the canary as the unit weakens the bound by up to $34\times$ (Section 6.5).

In each case the failure is not a weak attack or a miscomputed statistic, but a missing control or a mismatch between the quantity reported and the claim it supports.

Positive control and scope. A null result is informative only if the pipeline detects leakage when leakage is present. Applying the identical extraction machinery to a deliberately over-memorized checkpoint recovers 10–16% of planted secrets with static and adaptive attacks (Section 6.2). We nevertheless make a narrow claim. We do not conclude that the primary model memorizes nothing: single insertion may fall below the detection floor, and format-level generalization may raise member and non-member scores alike. We conclude that, under the audit regime we study, the uncontrolled statistics do not justify the memorization interpretations they appear to support.

A formal post-mortem of certification. We correct an additive formulation whose entropy term makes the bound vacuous by construction, but the pointwise likelihood ratio that reference-ratio audits compute on externally generated canaries does not estimate the KL divergence the bound requires. We therefore withdraw the certification claim and report budget quantities only as diagnostics (Section 4).

Contributions.

- **A controlled memorization audit** identifying three failure modes in exposure, membership, and extraction reporting (Sections 6.3–6.5).
- **A matched-reference analysis** showing that retraining the reference without the canaries does not recover separation.
- **A positive control and a minimum control set** for memorization audits: matched non-members, per-instrument positive controls, appropriate references, explicit estimands, and per-type reporting (Section 7).
- **A formal post-mortem of certification** that corrects the entropy bound, separates two leakage games, and identifies the estimator-to-divergence gap (Section 4, Appendix D).
- **An evaluation of the accompanying defense** showing that its benefit comes from a learned refusal component rather than the certificate-based architecture (Section 6.6, Appendix H).

2 Background and Related Work

Canary-based auditing and exposure. A *canary* is a synthetic sequence inserted into the training corpus and later scored or queried under the trained model. Because the auditor knows which sequences were injected, canaries provide controlled ground truth that naturally occurring secrets, whose membership is uncertain, cannot. The Secret Sharer (Carlini et al., 2019) introduced rank-based *exposure*, which ranks the true canary among same-format candidates. Reference-model approaches instead compare the audited model with a reference. For a canary k in context c we use the pointwise log-likelihood ratio

$$\hat{\kappa}(k, c) = \log p_\theta(k \mid c) - \log p_{\mathrm{ref}}(k \mid c), \tag{1}$$

where p_θ is the audited model and p_{ref} the reference. Unlike rank-based exposure, the meaning of $\hat{\kappa}$ depends on the reference and on the population it is compared against. We therefore treat $\hat{\kappa}$ as a reference-ratio memorization *diagnostic*, not as a certificate.

Memorization, membership, and extraction. *Membership inference* asks whether a specified example was in the training data (Shokri et al., 2017; Yeom et al., 2018); a valid test must distinguish members from comparable non-members, so a high response rate on members alone is not evidence of membership. *Extraction* concerns recovering training content from model outputs, either *verbatim* or *semantically*. A model can assign high likelihood to a secret without emitting it under a given decoding strategy, so exposure and extraction measure different things. These distinctions fix both the evaluation and the statistical unit: membership requires non-members, extraction may be measured per query or per secret, and exposure is interpretable only relative to a control (Appendix A.2).

Related work. Language models reproduce training data at scale (Carlini et al., 2021; Nasr et al., 2025; Carlini et al., 2023), and code models leak credentials, personal information, and memorized fragments (Huang et al., 2024; Lukas et al., 2023; Al-Kaswan et al., 2024). These studies establish *whether* and *how much* leakage occurs; we ask whether the *audit itself* supports the interpretation placed on its measurements. For membership inference, loss-ratio attacks approach chance in hard settings (Mireshghallah et al., 2022; Duan et al., 2024), while calibrated and neighbourhood attacks compare an example against references (Mattern et al., 2023; Zarifzadeh et al., 2024). Duan et al. (2024) show that apparent membership performance can arise from a distribution shift between members and non-members. We identify a complementary failure: even with structurally matched non-members, a likelihood-ratio test can respond to the shift induced by fine-tuning rather than to membership of the tested example. On the guarantee side, DP-SGD (Abadi et al., 2016) provides worst-case record-level protection at substantial cost for large models (Anil et al., 2022; Yu et al., 2022; Li et al., 2022), and privacy levels compatible with useful models need not eliminate observed extraction (Lukas et al., 2023). Certified robustness (Cohen et al., 2019; Wong & Kolter, 2018) and certified unlearning (Guo et al., 2020; Sekhari et al., 2021) illustrate the appeal of post-hoc guarantees defined against an explicit threat model, which originally motivated LeakCert. We do not position LeakCert as an alternative privacy guarantee. Our contribution is to identify the controls required before empirical memorization measurements can support the security conclusions drawn from them (Appendix A).

3 Threat Model and Leakage Games

Access: a gray-box audit. Our attack workloads use only the *sample* tier that a public completion API grants (prompt in, completion out). The certificate estimator and the likelihood-ratio test additionally require $\log p_\theta(k \mid c_k)$ for a *specific* continuation under both target and reference, i.e. the *log-probability* tier, which many commercial APIs expose only as truncated top-k values or not at all. No component uses gradients or weights. We therefore describe LeakCert as a **gray-box likelihood audit** and do not claim that its certificate is computable by a sample-only adversary (Table 7, Appendix B).

Two distinct games. A tempting formulation, which we initially adopted, writes one quantity $I(K; Y^B)$ with K "a canary drawn from the panel" and applies Fano's inequality to it. This conflates two experiments.

Game A (panel exposure audit). A base model M_0 is fine-tuned on a clean corpus $\mathcal{C}$ plus N canaries from a generator $\mathcal{G}$ with a fixed seed, $M_\theta = \mathrm{FineTune}(M_0, \mathcal{C} \cup \{c_1, \ldots, c_N\})$. Every canary is a member and none is hidden: $K \in \{1, \ldots, N\}$ is an **index over the panel**, and the audited quantity is an aggregate (the mean or a tail quantile) of per-canary exposures $\hat{\kappa}_k$. The law on K is a bookkeeping device, so conclusions concern the panel, not an individual secret. We call the object a **panel-average exposure diagnostic** and report the per-canary maximum separately.

Game B (local κ-ary discrimination). For one canary slot with context c, a candidate set holds κ structurally identical secrets (same length, charset, and generator), of which exactly one, $k^\star$, was injected. Now $K \sim \mathrm{Uniform}(\mathcal{K}_{\mathrm{local}})$ *is* a hidden secret, and Fano's inequality bounds the probability of recovering $k^\star$ after B queries. We did not run this experiment and hence report no extraction-probability bound.

The alphabets differ by orders of magnitude: κ counts plausible candidates for *one* slot, whereas N reaches 10^4. Substituting N into Fano's inequality produces extraction bounds that are both vacuous and mis-specified. Game B is also the game in which same-format decoys act as a control, which is the formal reason the non-member panel of Section 6.3 is the right instrument.

Scope. The audit covers exposure of the injected panel only. It offers no guarantee about secrets outside the panel (e.g., credentials already in the pre-training corpus) or about channels it does not query, such as downstream fine-tuning or timing side channels.

4 What a Certificate Would Require, and Why We Do Not Claim One

LeakCert was designed to output an information-theoretic *certificate*: a query-budget-aware upper bound on $I(K; Y^B)$, the information B query responses reveal about the canary index. We no longer claim one. This section states what can be established, what the initial formulation got wrong, and the gap between the theoretical quantity and the implemented estimator that we could not close.

4.1 Two corrections that do hold

Lemma 1 (Variational MI identity)**.** *Let K be discrete and Y a random variable with conditional law $P_{Y|K}$. For* any *distribution Q on the space of Y, $I(K;Y) = \min_Q \mathbb{E}_K\big[D_{\mathrm{KL}}(P_{Y|K} \,\|\, Q)\big] \le \mathbb{E}_K\big[D_{\mathrm{KL}}(P_{Y|K} \,\|\, Q)\big]$, with the minimum attained uniquely at $Q = P_Y$.*

Lemma 1 (proof in Appendix D) justifies substituting a reference q for the uncomputable output marginal P_Y: the reference affects the *tightness* of a bound built from exact KL divergences, never its soundness. The property belongs to the *exact* expression and does not transfer automatically to an empirical score (Section 4.2). Combining the lemma with the chain rule gives, for an adaptive attacker issuing $x_1, \ldots, x_B$ with x_b measurable in the history and any reference family q_b,

$$I(K; Y^B) \;\le\; \min\Big\{ H(K), \textstyle\sum_{b=1}^{B} \mathbb{E}\big[D_{\mathrm{KL}}\big(p_K^{(b)} \,\|\, q_b\big)\big] \Big\}. \tag{2}$$

Remark 2 (The additive form is vacuous). The initial statement read $I(K;Y^B) \le B \cdot D_{\mathrm{KL}}(p_K \| q_K) + \log|\mathcal{K}|$. Under a uniform prior $H(K) = \log|\mathcal{K}|$, so the additive form is bounded *below* by $H(K)$ for every $B \ge 0$, including $B = 0$ and $D_{\mathrm{KL}} = 0$. After the entropy cap that any correct implementation must apply, it equals $H(K)$ identically and can never exhibit the informative regime it was introduced to describe. The error is easy to make and to miss, because the extra term reads as harmless additive slack. Our implementation computed $\min\{H(K), B \cdot \bar{\hat{\kappa}}\}$ throughout, so all reported numbers are consistent with Eq. (2).

4.2 The gap we could not close

Eq. (2) concerns KL divergences between full conditional output distributions. What our pipeline computes, as does every reference-ratio memorization audit we are aware of, is

$$\hat{\kappa}_i \;=\; \log p_\theta(k_i \mid c_i) - \log p_{\mathrm{ref}}(k_i \mid c_i), \tag{3}$$

a *pointwise information density* evaluated at one externally generated string k_i. It estimates $D_{\mathrm{KL}}(P_{Y|K=i}\|Q) = \mathbb{E}_{Y\sim P_{Y|K=i}}[\log p/q]$ only if k_i is drawn from $p_\theta(\cdot \mid c_i)$ and the expectation is taken over such draws. Our canaries were generated externally and injected into training; they are not samples from the target. Consequently, (i) a pointwise $\hat{\kappa}_i$ can lie above or below the corresponding KL; (ii) the panel mean need not estimate any expectation under the target; (iii) the maximum over finitely many planted secrets does not upper-bound the KL over the completion space; and (iv) clamping negative values to zero makes the statistic non-negative but does not repair the sampling mismatch. Closing the gap requires either a theorem relating the canary distribution to the target's output distribution, or a Monte Carlo estimator over draws from $p_\theta(\cdot \mid c_i)$ whose sample complexity for a specific multi-token secret we do not characterize. **We therefore do not claim that the computed quantity upper-bounds $I(K;Y^B)$, and we do not present it as a certificate.** We use $\hat{\kappa}$ as an exposure diagnostic in the sense of Carlini et al. (2019), whose interpretation Section 6.3 tests directly.

4.3 Further withdrawn claims, and what survives

We also withdraw four further claims (Appendix C). *(i) The extraction-probability bound*, which applied Fano's inequality with a panel size where the local candidate-set game (Game B) is required. *(ii) High-probability concentration guarantees*: canary scores share one model, one optimizer trajectory, and generator templates; the range parameter is the observed maximum rather than a predetermined bound; and the Bernstein form omitted its range term. *(iii) A linear utility–leakage trade-off theorem*, which data processing does not imply. *(iv) Optimality of the sequential probability ratio test as an adaptive attacker*: Wald–Wolfowitz optimality concerns the stopping rule for fixed distributions, not adaptive prompt selection. We retain only the measured statement:

Observation 3. Across the defense configurations evaluated, we did not detect a statistically significant utility difference at the resolution of our evaluation ($n = 164$ HumanEval problems, one sample per problem, pass@1 $\approx$ 6–11%). This is an absence of detected effect at low power, not evidence of absence (Section 6.6).

What survives is Eq. (2), Remark 2, and the *vacuity threshold*

$$B^* \;=\; H(K)/\bar{\hat{\kappa}}, \tag{4}$$

the budget at which a bound built from mean exposure $\bar{\hat{\kappa}}$ would reach the entropy ceiling. B^* tells an auditor how far a panel is from any regime in which certification is even arithmetically possible; $B^* < 1$ means the ceiling is reached before the first query.

5 Experimental Setup

The evaluation addresses three questions. **Q1:** Does reference-model exposure distinguish injected canaries from matched non-members? **Q2:** Does the completion-based likelihood-ratio test detect membership, or only the effect of fine-tuning? **Q3:** What confidence does near-zero extraction support once the estimand is defined correctly?

Training arms. The primary target is Qwen2.5-Coder-1.5B fine-tuned for three epochs on a 500K-document code corpus containing 11,566 canaries. The default reference is the unfine-tuned base model. We also train a *matched reference* with the same architecture on the same corpus with the canary documents removed. Because canaries are inserted as separate documents, the two corpora differ by exactly those documents. Two auxiliary checkpoints serve fixed purposes: an intentionally *over-memorized 7B* checkpoint is a positive control for the attack pipeline, and a *leaky 0.5B* checkpoint, the only one of ours with non-zero undefended extraction, is used to compare defenses. They differ from the target in training configuration, so we use them only for these purposes and never read cross-arm differences as controlled scale effects (Table 8, Appendix E.1).

Canary and non-member panels. The panel contains four canary types: literal (T1, n=10,000; AWS-shaped identifiers, JWT bodies, RSA fragments, license keys), paraphrase (T2, n=1,000; the same content in base-64, language-mixed, or regex-class form), semantic (T3, n=283; algorithm stubs, vendor protocol headers, watermarked comments), and vulnerability (T4, n=283; SQL string

Table 1: Per-canary exposure $\hat{\kappa}$ (nats) on the primary target (base reference, full panel, clamped), per-type entropy H, vacuity threshold B^* (Eq. 4), the budgets at which a certificate would stay below H, and member/non-member AUC under the matched reference (Section 6.3). Large exposure coexists with chance-level separation for every type. The median over all canaries is 9.45 nats.

Type	n	mean $\bar{\hat{\kappa}}$	max	H	B^*	Non-vacuous at	AUC
T1 literal	10,000	17.53	64.12	9.210	0.53	—	0.488
T2 paraphrase	1,000	11.13	44.14	6.908	0.62	—	0.449
T3 semantic	283	156.46	284.08	5.645	0.04	—	0.517
T4 vulnerability	283	1.84	8.29	5.645	3.06	$B \in \{1, 2\}$	0.500
All	11,566	**19.99**	284.08	9.356	0.47	—	0.4855

concatenation, disabled host-key checking, unsafe deserialization). The panel entropy is $H(K) = \ln 11{,}566 = 9.356$ nats, and every canary is synthetic. The **non-member panel** is drawn from the same generator with seed 999 instead of 42: identical formats, contexts, and character distributions, disjoint secret values (verified by explicit set comparison), and never injected into any model. The member side is replayed with the exact injection-time parameters and verified against the injection manifest. Both sides use 283 canaries per type, 1,132 in total.

Workloads, attackers, and defenses. **W2** is a synthetic credential-extraction benchmark (4,832 prompts) modelled on Huang et al. (2024), whose user-level artifacts were withheld for privacy; it is a comparable benchmark, not a reproduction. **W3** measures utility on HumanEval (Chen et al., 2021) (164 problems, temperature 0.2, top-p 0.95, pass@1). **W4** is a code-secret suite with 7 prompts per canary that combines verbatim prefix probes, divergence attacks (Nasr et al., 2025), and structured-context probes (Terraform, Dockerfiles, CI YAML). **W5** rewrites each W4 prompt in five paraphrase modes. The attackers are `A-fixed` (static templates), `A-adaptive` (UCB bandit prompt search), `A-Carlini` (perplexity-ranked sampling; Carlini et al., 2021), `A-greedy-LRT` (the sequential test of Section 6.4, at most 20 queries per canary), and the exposure estimator; the last two are also run on the non-member panel. The defenses are **B1** none, **B2** temperature 0.5, **B3** top-p 0.7, **B5** a content filter (a regex bank, Meli et al., 2019, plus a learned secret classifier, Saha et al., 2020), and four LEAKCERT monitor configurations (Appendix H). No DP-SGD checkpoint completed training, so the DP comparison is analytic only.

Statistical reporting. We report exact Clopper–Pearson intervals together with their estimand and denominator; member/non-member separation as AUC with the mean gap and a Kolmogorov–Smirnov statistic; and bootstrap defense comparisons with an explicitly stated resampling unit, since prompts sharing a canary are not independent trials. Every bound-shaped value passes an automated entropy-cap audit (Appendix G.2).

6 RESULTS

We first report the signal an uncontrolled audit produces (Section 6.1), then verify that the pipeline detects leakage when it exists (Section 6.2), and only then apply the three controls (Sections 6.3–6.5). We close with the budget diagnostic, the defense, and utility (Section 6.6).

6.1 THE EXPOSURE SIGNAL, AS AN UNCONTROLLED AUDIT REPORTS IT

Table 1 reports per-canary exposure on the primary target, scored against the base reference over the full panel with negative values clamped to zero, which is the estimator's default. Read without a control, the mean of 19.99 nats, more than twice $H(K)$, signals overwhelming memorization, and the type gradient spans two orders of magnitude (1.84 nats for T4 against 156.46 for T3). Both facts are correct as arithmetic; whether either reflects *memorization* is settled by the last column of the table (Section 6.3).

Table 2: Positive control. The same pipeline recovers planted secrets from an over-memorized checkpoint.

Attack	1.5B	7B
W4 static	0.00%	10.3%
W4 adaptive, B=100	0.00%	16%
W5 paraphrase	0.00%	≈8%

Table 3: Upper endpoints of exact two-sided 95% Clopper–Pearson intervals for zero hits (greedy decoding). The estimand fixes the denominator.

Estimand	Question	n	Upper
Per-request, W4	query leaks?	700	0.526%
Per-request, W5	query leaks?	3,500	0.105%
Per-canary	**secret leaks?**	**100**	**3.622%**

Table 4: Member vs. non-member exposure under the matched-corpus reference (signed LLR, $n = 1{,}132$ per side). Intervals treat canaries as exchangeable within one training run and therefore understate uncertainty (Appendix F).

	Member	Non-member
mean $\hat{\kappa}$ (nats)	49.474	49.489
median	10.816	11.341
std	78.775	78.149
fraction > 0	99.1%	99.5%
AUC $= 0.4855$, 95% CI $[0.462, 0.509]$		
gap $= -0.0149$ nats, CI $[-6.5, +6.5]$; KS $= 0.049$		

Table 5: Completion-based SPRT and its false-positive control (n=100 per panel). TPR and FPR are indistinguishable, so the member column is not evidence of membership. Verbatim recovery is 0% throughout.

	Overall		T1 literal	
Defense	Mem.	**Non.**	Mem.	**Non.**
B1 none	99%	**100%**	96%	**100%**
B5 filter	94%	**90%**	80%	**72%**
LEAKCERT	88%	**93%**	60%	**76%**
mean queries to decision				
B1 none	1.04	1.12	1.08	1.12
B5 filter	1.34	1.40	2.08	2.28
LEAKCERT	1.55	1.58	2.88	3.00

6.2 POSITIVE CONTROL: THE PIPELINE DETECTS LEAKAGE

Before interpreting any null, we ran the identical W4/W5 suite (same prompts, scorer, and code) against the over-memorized 7B checkpoint, trained to a loss of 0.044. The pipeline that returns 0% on the 1.5B target recovers 10.3% of canaries with static templates, 16% with the adaptive attacker at B=100, and about 8% with the five paraphrase modes (Table 2). The 0% on the target is therefore a property of the model, not of the attack code. Three caveats apply: the 7B corpus has 700 documents against the target's 500K, so this is a stress test rather than a controlled scale comparison; neither the content filter nor LEAKCERT reduced its extraction rate, an unresolved anomaly; and its per-canary outputs were lost, so aggregates come from the run summary (Appendix J).

6.3 CONTROL I: EXPOSURE DOES NOT SEPARATE MEMBERS FROM NON-MEMBERS

We scored the never-injected seed-999 panel with the same estimator and report the *signed* statistic, since clamping would destroy the comparison. The two distributions coincide (Table 4): the mean gap is -0.015 nats on a standard deviation of about 78, and the AUC interval contains 0.5. No type separates (Table 1, last column): T2 scores *below* chance, T4 sits exactly at it, and T3, the type with the largest exposure (156 nats), reaches only 0.517.

A matched reference does not rescue the statistic. The natural diagnosis is a confounded reference: a target-versus-base score absorbs every effect of fine-tuning on 500K documents, not only memorization. We tested this directly by training the matched reference (≈ 173 GPU-hours). Removing the fine-tuning confound in this principled way moved the AUC by 0.005 (base: 0.49; matched: 0.4855).

We also audited panel independence, replay fidelity, and injector cleanliness, and found no pipeline defect (Appendix E.4).

Interpretation. Two hypotheses remain, and our data cannot separate them. First, *single injection may fall below the detection floor*: each canary appears once in a 500K-document corpus (at most three times across training), whereas the memorization literature typically reports reliable signal only after tens of repetitions. Second, *format-level generalization may mask a smaller per-instance effect*:

member and non-member canaries share a small set of templates and character classes, so training on $10{,}000$ member canaries plausibly teaches the *shape* of a canary continuation and raises $\hat{\kappa}$ for any string of that shape. Both point the same way. The claim that the target strongly memorizes its canaries, which Table 1 appears to support overwhelmingly, is not sustained. This does *not* mean the model memorizes nothing or that exposure is useless; it means that an exposure score reported without a non-member panel does not license a memorization claim, even under a matched reference.

6.4 Control II: the membership test detects fine-tuning

We instantiated `A-greedy-LRT` as a completion-based SPRT: it generates through the defense, scores each output by $\Lambda = \log p_{\text{target}}(y) - \log p_{\text{ref}}(y)$, and accepts when the cumulative ratio leaves the Wald band. On members the test looks highly effective (Table 5): 99% detection against the undefended target after 1.04 queries on average. On the non-member panel, however, it returns 90–100% detection. With TPR $\approx$ FPR the test carries no membership information, and we retract the membership reading. The mechanism is structural: the target is fine-tuned *from* the reference and the scored text is generated *by* the target, so $\Lambda > 0$ for almost any target output regardless of membership. The test reliably detects that a text came from the fine-tuned model. The defense-cost signal is also membership-blind (2.88 vs. 3.00 queries on T1 members and non-members under LeakCert). A corrected attack must score the *secret's* likelihood ratio, calibrated against non-members, which is exactly the quantity shown to be at chance in Section 6.3.

6.5 Control III: the estimand and its denominator

Verbatim extraction is 0% on the 1.5B target in every condition that completed: W4 templates, all five W5 modes, greedy and sampled decoding, and the adaptive attacker against the undefended model. (The adaptive attacker against the two defended configurations exceeded the wall-clock limit; those cells are missing, not zero.) W4 issues seven prompts per canary and W5 thirty-five, over only 100 distinct canaries, and repeated probes of one secret are not independent evidence about additional secrets. A statement such as "the target does not leak its secrets" concerns secrets, and the per-canary bound is $6.9\times$ and $34.4\times$ larger than the two per-request bounds (Table 3; the one-sided values are 0.427%, 0.086%, and 2.951%). Both estimands are legitimate, but the per-canary figure should carry the headline. The same care applies to rounding: the sampled-decoding sweep reports 0.00%, yet one of $3{,}500$ undefended W5 prompts matched (CP 0.029%, $[0.0007\%, 0.159\%]$). We classify it as an incidental pattern match, but a rounded 0.00% conceals it (Appendix E.6).

6.6 Budget diagnostic, defense, and utility

Budget diagnostic. Following Section 4, $B \cdot \bar{\hat{\kappa}}$ is not a bound on $I(K; Y^B)$; comparing it with the entropy ceiling asks only whether a panel is *arithmetically capable* of a non-trivial bound. For the full panel $B^* = 0.47 < 1$, and after capping the quantity equals $H(K)$ at every budget from 10^2 to 10^7 on both the 1.5B and 7B targets. Three of four types also have $B^* < 1$. Only T4 stays below its ceiling, at $B \in \{1, 2\}$, and its AUC is exactly 0.500: the only panel capable of a non-trivial bound is the one the model barely memorized. The analytic Rényi DP bound $B\epsilon^2/2$ at $B{=}10^4$ also saturates for every $\epsilon \in \{1, 2, 4, 8, 16\}$ (Appendix E.7).

Defense. On the 1.5B target every defense ratio is $0/0$, so we compare defenses on the leaky 0.5B arm (Appendix H). Only the learned uncertainty-refusal variant reliably beats the content filter, reducing paraphrase extraction from 5.98% to 2.84% over seeds 42–46 ($\mathbb{P}(\text{B5} > \text{ours}) \geq 0.997$ per seed). The *full* four-stage monitor does not beat the filter ($\mathbb{P} = 0.41$ and 0.33 on two seeds), and removing certificate accounting changes no outcome in any ablation. All measured extraction comes from T4 vulnerability idioms (50.0% under B5 and 25.0% under ours; T1–T3 are 0.00% in every configuration), so this arm measures suppression of vulnerability-pattern generation, not recovery of planted secrets.

Utility and overhead. HumanEval pass@1 on the 1.5B target is 6.1% for both B1 and LeakCert, with no refusals, but an earlier run of the same configuration reorders all arms, and the 4.3–8.5% spread is sampling noise at $n{=}164$ (Observation 3). The refusal threshold has a sharp knee at $t{=}0.95$ (refusals $\leq 2.5\%$, versus 45% at $t{=}0.7$). With a median latency of 52–80 s per decision, LeakCert

Table 6: A minimum control set for canary-based memorization audits, the failure each control guards against, and its effect in our audit.

Control	Guards against	Effect in our audit
Matched non-members	Reading raw scores or member-only rates as memorization	Exposure AUC 0.4855; SPRT FPR 90–100%
Per-instrument positive control	Nulls caused by a weak or broken instrument	Extraction validated on 7B (10–16%); exposure, SPRT not yet
Appropriate reference	Attributing the fine-tuning shift to membership	Matched reference moved AUC by 0.005
Explicit estimand and unit	Overstated confidence from correlated trials	Per-canary bound 6.9–$34.4\times$ looser than per-query
Per-type reporting	Aggregates that hide the mechanism	All defense-arm extraction is T4 idioms

is an *offline audit monitor*, not an inline defense. No configuration extracted any secret on the W2 credential benchmark (Appendices E.8 and E.9).

7 Discussion

One confound, several failure modes. Two of the failures share a single mechanism. Fine-tuning shifts likelihood relative to the reference over broad regions of output space, so any statistic of the form $\log p_{\text{target}}(\cdot) - \log p_{\text{ref}}(\cdot)$ can respond to fine-tuning even when the quantity of interest is membership of a specific example. Applied to target-generated completions, the sequential test responds equally to members and non-members; applied to planted secrets, exposure is large for injected canaries and equally large for matched non-members. In both cases the measured shift is real, but its reading as membership evidence is not. A matched-corpus reference removes the target-versus-base confound without restoring separation, plausibly because of format-level generalization from shared canary templates. This motivates template-disjoint non-members and per-slot near-neighbour decoys (Game B), which we did not run.

A minimum control set for memorization audits. Table 6 summarizes the controls our results argue for; each changed a conclusion of our own audit.

Separation and vacuity. Eq. (2) requires exact divergences that our pointwise statistic does not instantiate, and the same statistic fails to separate members from non-members. The first prevents certification; the second questions whether the statistic carries a membership signal at all. We therefore recommend reporting *separation* (does the statistic distinguish members from matched non-members?) together with *vacuity* (could certificate-style accounting remain below the entropy ceiling, i.e. is $B^* \geq 1$?). Neither is a certificate. For deployment, a low extraction rate is not evidence of safety, nor is a large exposure score evidence of memorization; decisions should rest on controlled, comparative evidence.

Limitations. All member/non-member evidence comes from one target/reference training pair, one model scale, and one injection regime, and the positive control validates extraction only; Appendix F discusses these and further limitations.

8 Conclusion

We set out to certify memorization leakage in fine-tuned code models and found that the audit itself needed controls: exposure did not separate members from matched non-members, a likelihood-ratio test detected fine-tuning rather than membership, and extraction confidence changed by up to $34\times$ with the estimand. Because pointwise exposure does not estimate the divergence a certificate requires, we make no certification claim. Memorization audits should report member/non-member separation rather than raw scores, validate each instrument with its own positive control, and state the estimand and unit behind every confidence claim.

AI USE STATEMENT

Synthetic data generation. We used ChatGPT 5.6 (OpenAI) to generate the synthetic canary data: the credential-shaped strings, paraphrased variants, semantic stubs, and insecure-code idioms that make up the four canary types described in Section 5. No canary was derived from real credentials, real repositories, or user data. We assembled the member and non-member panels, injected the canaries, and verified panel disjointness with our own pipeline (Appendix G.5).

Research design, theory, and implementation. Generative AI tools were not used to design the study, to develop the theoretical analysis, or to formulate the mathematical claims and proofs (Section 4 and Appendix D). All training, attack, and evaluation code was written by the authors, and all experiments and analyses were conducted by the authors.

Writing assistance. Generative AI tools were used for language editing and for adapting the manuscript to the ICLR 2027 template. We reviewed all AI-assisted material and take full responsibility for the content of this paper, including all text, claims, and artifacts.

ETHICS STATEMENT

All canaries are synthetic: credential-shaped canaries imitate only the *format* of real credential types, and none is or has been valid for any account, service, or person. No real user data, live credential, or third-party secret was created, handled, or placed at risk, and no human subjects are involved. The credential-extraction artifacts of Huang et al. (2024) contain real user-level data and were withheld by their authors; we did not attempt to reconstruct them, and our comparable benchmark is fully synthetic. Vulnerability canaries are short, widely documented anti-patterns, not exploits. The attacks we implement are established in the literature, and our principal offensive-side findings are negative, so publishing them protects practitioners from acting on flawed audit evidence rather than increasing attacker capability. Because canary strings resemble credentials, we release deterministic regeneration scripts instead of raw files. We also report corrected and retracted claims, including failures of our own defense configurations, as an obligation of measurement work. Appendix I gives the full ethics considerations.

REPRODUCIBILITY STATEMENT

Section 5 and Appendices E.1 and G describe the training arms, canary and non-member panel construction, workloads, attackers, defenses, the exposure estimator, and the exact Clopper–Pearson computation. Complete proofs of Lemma 1 and Eq. (2), together with the computed (non-validated) concentration expressions, are in Appendix D. An anonymous artifact containing the pipeline, configurations with fixed seeds, per-run result files for every table and figure, and a single entry point that reproduces the central member/non-member result from released checkpoints is described in Appendix J, which also lists the artifacts that were lost and cannot be released.

REFERENCES

Martin Abadi, Andy Chu, Ian Goodfellow, H. Brendan McMahan, Ilya Mironov, Kunal Talwar, and Li Zhang. Deep learning with differential privacy. In *ACM Conference on Computer and Communications Security (CCS)*, 2016.

Ali Al-Kaswan, Maliheh Izadi, and Arie van Deursen. Traces of memorisation in large language models for code. In *IEEE/ACM International Conference on Software Engineering (ICSE)*, 2024. doi: 10.1145/3597503.3639133.

Rohan Anil, Badih Ghazi, Vineet Gupta, Ravi Kumar, and Pasin Manurangsi. Large-scale differentially private BERT. In *Findings of the Association for Computational Linguistics: EMNLP*, 2022.

Nicholas Carlini, Chang Liu, Ulfar Erlingsson, Jernej Kos, and Dawn Song. The secret sharer: evaluating and testing unintended memorization in neural networks. *USENIX Security Symposium*, 2019.

Nicholas Carlini, Florian Tramer, Eric Wallace, Matthew Jagielski, Ariel Herbert-Voss, Katherine Lee, Adam Roberts, Tom Brown, Dawn Song, Ulfar Erlingsson, et al. Extracting training data from large language models. In *USENIX Security Symposium*, 2021.

Nicholas Carlini, Daphne Ippolito, Matthew Jagielski, Katherine Lee, Florian Tramer, and Chiyuan Zhang. Quantifying memorization across neural language models. In *International Conference on Learning Representations (ICLR)*, 2023.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, et al. Evaluating large language models trained on code. In *arXiv preprint arXiv:2107.03374*, 2021.

Jeremy M. Cohen, Elan Rosenfeld, and J. Zico Kolter. Certified adversarial robustness via randomized smoothing. In *International Conference on Machine Learning (ICML)*, 2019.

Michael Duan, Anshuman Suri, Niloofar Mireshghallah, Sewon Min, Weijia Shi, Luke Zettlemoyer, Yulia Tsvetkov, Yejin Choi, David Evans, and Hannaneh Hajishirzi. Do membership inference attacks work on large language models? In *Conference on Language Modeling (COLM)*, 2024.

Chuan Guo, Tom Goldstein, Awni Hannun, and Laurens van der Maaten. Certified data removal from machine learning models. In *International Conference on Machine Learning (ICML)*, 2020.

Yizhan Huang, Yichen Li, Weibin Wu, Jianping Zhang, and Michael R. Lyu. Your code secret belongs to me: neural code completion tools can memorize hard-coded credentials. *Proceedings of the ACM on Software Engineering*, 1(FSE):2515–2537, 2024. doi: 10.1145/3660818.

Matthew Jagielski, Milad Nasr, Katherine Lee, Christopher A. Choquette-Choo, Nicholas Carlini, and Florian Tramer. Students parrot their teachers: membership inference on model distillation. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2023.

Xuechen Li, Florian Tramer, Percy Liang, and Tatsunori Hashimoto. Large language models can be strong differentially private learners. In *International Conference on Learning Representations (ICLR)*, 2022.

Nils Lukas, Ahmed Salem, Robert Sim, Shruti Tople, Lukas Wutschitz, and Santiago Zanella-Béguelin. Analyzing leakage of personally identifiable information in language models. In *IEEE Symposium on Security and Privacy (S&P)*, 2023.

Justus Mattern, Fatemehsadat Mireshghallah, Zhijing Jin, Bernhard Schölkopf, Mrinmaya Sachan, and Taylor Berg-Kirkpatrick. Membership inference attacks against language models via neighbourhood comparison. In *Findings of the Association for Computational Linguistics: ACL*, 2023.

Michael Meli, Matthew R. McNiece, and Bradley Reaves. How bad can it git? characterizing secret leakage in public github repositories. In *Network and Distributed System Security Symposium (NDSS)*, 2019.

Fatemehsadat Mireshghallah, Kartik Goyal, Archit Uniyal, Taylor Berg-Kirkpatrick, and Reza Shokri. Quantifying privacy risks of masked language models using membership inference attacks. In *Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2022.

Milad Nasr, Javier Rando, Nicholas Carlini, Jonathan Hayase, Matthew Jagielski, A. Feder Cooper, Daphne Ippolito, Christopher A. Choquette-Choo, Florian Tramer, and Katherine Lee. Scalable extraction of training data from aligned, production language models. In *International Conference on Learning Representations (ICLR)*, 2025.

Aakanksha Saha, Tamara Denning, Vivek Srikumar, and Sneha Kumar Kasera. Secrets in source code: reducing false positives using machine learning. In *International Conference on Communication Systems and Networks (COMSNETS)*, 2020. doi: 10.1109/COMSNETS48256.2020.9027350.

Ayush Sekhari, Jayadev Acharya, Gautam Kamath, and Ananda Theertha Suresh. Remember what you want to forget: algorithms for machine unlearning. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2021.

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

Eric Wong and J. Zico Kolter. Provable defenses against adversarial examples via the convex outer adversarial polytope. In *International Conference on Machine Learning (ICML)*, 2018.

Samuel Yeom, Irene Giacomelli, Matt Fredrikson, and Somesh Jha. Privacy risk in machine learning: analyzing the connection to overfitting. In *IEEE Computer Security Foundations Symposium (CSF)*, 2018.

Da Yu, Saurabh Naik, Arturs Backurs, Sivakanth Gopi, Huseyin A. Inan, Gautam Kamath, Janardhan Kulkarni, Yin Tat Lee, Andre Manoel, Lukas Wutschitz, et al. Differentially private fine-tuning of language models. In *International Conference on Learning Representations (ICLR)*, 2022.

Sajjad Zarifzadeh, Philippe Liu, and Reza Shokri. Low-cost high-power membership inference attacks. In *International Conference on Machine Learning (ICML)*, 2024.

A Extended Background and Related Work

This appendix gives the full background and related-work discussion summarized in Section 2.

A.1 Canary-Based Memorization Auditing

Canary-based auditing provides controlled ground truth for studying memorization in language models. A synthetic sequence, or *canary*, is inserted into the training corpus and subsequently queried or scored under the trained model. Because the auditor knows which sequences were injected, canaries make it possible to study memorization without relying on naturally occurring secrets whose training membership may be uncertain.

The Secret Sharer framework (Carlini et al., 2019) introduced *exposure* as a measure of how strongly a model prefers an injected canary relative to a population of structurally similar candidates. Classical exposure is rank-based: the true canary is ranked according to its model likelihood or perplexity among same-format alternatives. A highly exposed canary is therefore one that the model assigns unusually high preference relative to its candidate set.

Reference-model approaches instead compare the likelihood assigned by a target model with that assigned by a reference model. For a canary k appearing in context c, we use the pointwise log-likelihood ratio of Eq. (1), where p_θ denotes the audited model and p_{ref} the reference. This statistic measures how much more strongly the target model prefers a particular continuation than the reference model. Unlike classical rank-based exposure, however, its interpretation depends on the choice of reference and on what comparison population is used. Throughout this paper, we therefore treat $\hat{\kappa}$ as a reference-ratio memorization diagnostic rather than as a certificate by itself.

A.2 Memorization, Membership, and Extraction

We distinguish several notions that are related but not equivalent.

Memorization refers broadly to model behavior that depends on particular training examples. In this paper, *membership inference* asks whether a specified example was present in the training data, without requiring recovery of its content (Shokri et al., 2017; Yeom et al., 2018). A successful membership test must therefore distinguish members from comparable non-members; a high response rate on members alone is insufficient.

Extraction instead concerns recovery of training content from model outputs. We distinguish *verbatim extraction*, in which the injected sequence is reproduced exactly, from *semantic extraction*, in which equivalent information is recovered under a different surface form. These notions should not be conflated with exposure: a model can assign high likelihood to a secret without emitting it under a particular decoding strategy, while successful extraction is direct evidence that the corresponding information is recoverable.

These distinctions determine both the appropriate evaluation and the statistical unit of analysis. Membership requires comparison against non-members; extraction may be measured per query or per secret depending on the security question; and exposure is a continuous score whose magnitude is meaningful only relative to an appropriate control.

A.3 Memorization and Extraction in Language and Code Models

Canary-based memorization measurement was popularized by the Secret Sharer framework (Carlini et al., 2019), which inserts high-entropy sequences into training data and measures their exposure relative to same-format alternatives. Subsequent work demonstrated that language models can reproduce training data at substantially larger scale. Carlini et al. (2021) recover training examples from GPT-2 using generation followed by reference-model perplexity ranking, while later divergence attacks extend extraction to aligned production models (Nasr et al., 2025). Other work studies how memorization changes with model and dataset scale (Carlini et al., 2023).

Code models introduce particularly security-relevant forms of memorization because their training corpora may contain credentials, licensed fragments, and insecure code. Prior studies document leakage of credentials and personally identifying information as well as memorized code fragments (Huang et al., 2024; Lukas et al., 2023; Al-Kaswan et al., 2024). These works primarily establish whether memorization or extraction occurs and how much can be recovered. Our focus is complementary: we study whether the *audit itself* supports the interpretation assigned to its measurements. In particular, we examine how reference-ratio scores behave relative to matched non-members and how the choice of statistical estimand affects conclusions drawn from extraction experiments.

A useful distinction arises from the original Secret Sharer design. Its same-format candidate population provides a direct comparison between the injected canary and alternative strings. A standalone reference-model likelihood ratio does not provide the same member/non-member comparison automatically. Our results motivate making such controls explicit when reference-ratio statistics are used as evidence of memorization.

A.4 Membership Inference and Evaluation Controls

Membership inference asks whether an example appeared in a model's training data (Shokri et al., 2017; Yeom et al., 2018). Likelihood- and loss-based attacks have been widely studied for language models, but their effectiveness becomes less clear at modern model scales. Several studies report that simple loss-ratio attacks approach chance in challenging settings (Mireshghallah et al., 2022; Duan et al., 2024), while calibrated and neighbourhood-based approaches attempt to improve separation by comparing an example with suitable references (Mattern et al., 2023; Zarifzadeh et al., 2024).

Of particular relevance, Duan et al. (2024) show that apparent membership-inference performance can result from weak evaluation practice, especially when members and non-members come from different distributions. Our work identifies a complementary failure mode. Even when non-members are generated from the same structure as members, a likelihood-ratio test may respond to the distributional shift created by fine-tuning rather than to membership of the specific example. This reinforces the need to evaluate both true- and false-positive behaviour under carefully matched member and non-member populations.

A.5 Privacy Guarantees and Certified Defenses

Differential privacy provides a fundamentally different form of protection. DP-SGD (Abadi et al., 2016) gives a worst-case, data-independent record-level privacy guarantee, although achieving strong privacy can impose substantial utility and computational costs for large language and code models (Anil et al., 2022; Yu et al., 2022; Li et al., 2022). Empirical work also shows that privacy parameters compatible with useful model quality do not necessarily eliminate observed extraction (Lukas et al., 2023), while knowledge distillation does not in general remove memorized information (Jagielski et al., 2023). We therefore view empirical memorization auditing as complementary to, rather than a replacement for, differential privacy.

Our original motivation is also related to the work on certified defenses. Certified robustness methods provide guarantees against specified perturbation classes through techniques such as randomized

Table 7: Access tiers. The certificate needs strictly more than a public completion API provides.

Tier	Access	Used for
Sample	prompt $\rightarrow$ completion	W2/W4/W5 attacks
Log-prob	$\log p$(target $\mid$ prompt)	certificate, LRT
White-box	gradients / weights	not used

smoothing (Cohen et al., 2019) and convex relaxations (Wong & Kolter, 2018), while certified unlearning aims to guarantee the effect of removing training data (Guo et al., 2020; Sekhari et al., 2021). These approaches illustrate the appeal of a post-hoc guarantee defined relative to an explicit threat model.

LEAKCERT was initially motivated by an analogous goal: to relate a query budget to information revealed about injected secrets. Our analysis ultimately exposes an important distinction between such formal guarantees and empirical memorization scores. A valid information-theoretic bound requires a well-defined divergence over model output distributions; a pointwise likelihood ratio evaluated on externally generated canaries does not automatically estimate that quantity. Consequently, we do not position LEAKCERT as an alternative privacy guarantee. Instead, our contribution is to identify the conditions and controls required before empirical memorization measurements can support the security conclusions drawn from them.

B THREAT MODEL AND LEAKAGE GAMES IN FULL

Two modelling choices in this section fix errors that are easy to make and hard to see: what access the auditor actually needs, and what random experiment the mutual information refers to.

B.1 ACCESS TIERS: THIS IS A GRAY-BOX AUDIT

Certificates of this kind are often called "black-box." That is imprecise. Table 7 separates three tiers. Our attack workloads operate at the *sample* tier, which is what a public completion API grants. The certificate estimator requires $\log p_\theta(k \mid c_k)$ for a specific continuation under both target and reference: the *log-probability* tier, which many commercial APIs do not expose, or expose only as truncated top-k values. We therefore describe LEAKCERT as a **gray-box likelihood audit** and do not claim the certificate is computable by a sample-only adversary. Estimating the probability of a specific multi-token secret by sampling alone has a sample complexity we do not characterize; we flag it as open (Appendix F).

B.2 TWO DISTINCT GAMES

Prior formulations (including our own earlier one) write a single quantity $I(K; Y^B)$ with K "a canary drawn from the panel" and then apply Fano's inequality to bound the probability an attacker guesses K. This conflates two experiments differing in what is random and in the size of the relevant alphabet.

Game A: panel exposure audit. Fix a base model M_0, a clean corpus $\mathcal{C}$, and a panel of N canaries from generator $\mathcal{G}$ with fixed seed. The target is $M_\theta = \mathrm{FineTune}(M_0, \mathcal{C} \cup \{c_1, \ldots, c_N\})$. Every canary is a member; none is hidden. Here $K \in \{1, \ldots, N\}$ is an **index over the panel**, not a hidden secret, and the audited quantity is an aggregate: the mean, or a tail quantile, of per-canary exposures $\hat{\kappa}_k$. The uniform (or π-weighted) law on K is a bookkeeping device letting us apply mutual-information machinery to N per-canary quantities; it does *not* correspond to a physical situation in which one secret among N was selected. Eq. (2) lives in Game A, and its conclusion concerns the panel, not any individual secret. This is why we call the object a **panel-average exposure diagnostic**: per-canary exposures can and do exceed the panel mean, and we report the maximum separately as an uncapped diagnostic.

Game B: local κ-ary discrimination. Fix one canary slot with context c. Define a candidate set of κ structurally identical secrets (same length, charset, and generator) of which exactly one, $k^\star$, was

injected and the remaining $\kappa - 1$ never appeared in any corpus. Now $K \sim \mathrm{Uniform}(\mathcal{K}_{\mathrm{local}})$ *is* a hidden secret in the cryptographic sense, and Fano's inequality would apply to the probability of recovering $k^\star$ after B queries. We did not run such an experiment, which is why this paper reports no extraction-probability bound (Section 4.3).

Why the separation matters. The alphabet sizes differ by orders of magnitude: κ counts plausible candidates for *one* slot, while N reaches 10^4. Substituting the panel size N into the Fano bound (as a single shared "canary set size" parameter invites) inflates the denominator and yields extraction bounds that are simultaneously vacuous and mis-specified. Game B is also the game in which the Secret Sharer's same-format decoys act as a control, which is the formal reason the non-member panel of Section 6.3 is the right instrument.

B.3 Scope

The audit covers exposure of the injected panel only. It offers no formal guarantee about secrets never part of the panel (for instance real credentials already in the pre-training corpus), nor about channels the audit does not query, such as behaviour after downstream fine-tuning or side channels such as timing. The panel is designed to be representative of a class of structured secrets, but representativeness is an empirical design choice, not a theorem.

C The Certificate Post-Mortem in Full

We present the certificate analysis of Section 4 in full because the gap it identifies is easy to overlook; we overlooked it ourselves, and it is the reason the budget numbers in Section 6.6 are reported as diagnostics rather than guarantees.

Reference substitution. Lemma 1 answers a question the reference-model approach otherwise leaves open: why is it legitimate to substitute a reference q for the true output marginal P_Y, which is not computable? Because the inequality holds for every Q, the choice of reference affects the tightness of a bound built from exact KL divergences, never its soundness. We flag, because we previously blurred this, that the property belongs to the *exact* expression and does not transfer automatically to any empirical score computed against a reference.

On the additive form. The mistake recorded in Remark 2 is easy to make and survives review: it is off by a term that reads as harmless additive slack. Because our implementation computed $\min\{H(K), B \cdot \hat{\bar{\kappa}}\}$ all along, the published numbers were consistent with Eq. (2) and not with the stated theorem, a discrepancy a reader checking arithmetic against the theorem would find, and that we should have found earlier.

Consequences of the estimator gap. Because the canaries are not samples from the target's output distribution:

- a pointwise $\hat{\kappa}_i$ can lie above or below the corresponding KL, in either direction;
- the mean over a synthetic panel need not estimate any expectation under the target distribution;
- the maximum over finitely many planted secrets does not upper-bound the KL over the full completion space; and
- clamping negative values to zero does not repair the sampling mismatch: it only makes the statistic non-negative.

Eq. (2) therefore appears in this paper as a statement of what a certificate would have to satisfy, and $\hat{\kappa}$ as an exposure diagnostic in the sense of Carlini et al. (2019).

C.1 Four further claims we withdraw

The extraction-probability bound. We previously converted the panel quantity into an extraction probability via Fano's inequality with denominator $\log \kappa$, and instantiated κ with a canary-type panel

size. Fano's inequality applies to a κ-ary guessing game in which exactly one of κ structurally identical candidates was injected and the rest never appeared in training. Our panels are not that: all 283 vulnerability canaries, for instance, are *members*, occupying different slots. We ran no local candidate-set experiment, so we withdraw the extraction-probability bound entirely and report the budget analysis of Section 6.6 without converting it.

Concentration as a high-probability guarantee. Hoeffding- and Bernstein-style concentration over per-canary scores requires assumptions our setting violates. All canaries are injected into one corpus and one model, so their scores share parameters, optimizer trajectory, and (critically) generator templates, which is precisely the dependence Section 6.3 identifies empirically. Separately, the range parameter $\kappa_{\max}$ we use is the *observed sample maximum*, not a predetermined bound, which Hoeffding requires; and the Bernstein expression we reported omits the standard range-dependent $O(\log(1/\delta)/n)$ term. We therefore do not claim validated $1-\delta$ coverage for any quantity in this paper, and we report the concentration expressions in Appendix D only to document what was computed.

A utility–leakage trade-off theorem. We previously asserted that any decoding-time defense reducing leakage by factor ρ must increase task error by at least $\rho c_{\mathcal{T}}$. Data processing establishes only that neither $I(K;Y)$ nor $I(X_{\mathcal{T}};Y)$ increases under post-processing; it does not imply a linear trade-off, and with $c_{\mathcal{T}}$ unconstrained the statement is either unproven or empty. We withdraw it and retain only what we measured, stated as Observation 3 in Section 4.3.

We similarly withdraw an earlier claim that a sequential probability ratio test is the optimal adaptive attacker. Wald–Wolfowitz optimality concerns the *stopping rule* for observations from fixed distributions; it says nothing about adaptive selection of prompts that induce different distributions. Our adaptive attacker is a UCB bandit, and we make no optimality claim for it. We also correct a sign error: under the null the normalized likelihood-ratio process has non-positive conditional drift and is a *supermartingale*, not a submartingale.

D Proofs and Computed Expressions

This appendix proves the two results Section 4 claims (Lemma 1 and Eq. (2)) and then documents, without claiming validity, the concentration expressions our pipeline computed. The separation is deliberate: the first part is established, the second is not.

Lemma 1 (variational MI identity)

Proof. Let $P_{K,Y}$ be the joint law and P_Y the true marginal of Y. By definition $I(K;Y) = \mathbb{E}_K[D_{\mathrm{KL}}(P_{Y|K} \,\|\, P_Y)]$. For an arbitrary Q on the space of Y, decompose pointwise:

$$\log \frac{P_{Y|K}}{Q} = \log \frac{P_{Y|K}}{P_Y} + \log \frac{P_Y}{Q}.$$

Taking expectation under $P_{K,Y}$, the first term gives $I(K;Y)$ and the second gives $D_{\mathrm{KL}}(P_Y \| Q)$, since it does not depend on K and the marginal of Y under $P_{K,Y}$ is P_Y. Hence

$$\mathbb{E}_K\big[D_{\mathrm{KL}}(P_{Y|K} \,\|\, Q)\big] - I(K;Y) = D_{\mathrm{KL}}(P_Y \,\|\, Q) \geq 0$$

by Gibbs' inequality, with equality iff $Q = P_Y$ almost everywhere. Minimizing over Q gives the identity; dropping the minimum gives the upper bound for every Q. □

Scope. The inequality holds for any Q, so a poorly chosen reference costs tightness but not soundness *when the quantity computed is the exact KL divergence*. It says nothing about a pointwise score of the form Eq. (3); see Section 4.2.

Equation (2) (entropy-capped budget bound)

Proof. By the chain rule for mutual information, $I(K;Y^B) = \sum_{b=1}^{B} I(K;Y_b \mid Y_{<b})$. Fix b and condition on $Y_{<b} = y_{<b}$. Applying Lemma 1 to the conditional joint law of (K, Y_b) given $y_{<b}$, with

$Q = q_b(\cdot \mid y_{<b})$,

$$I(K; Y_b \mid Y_{<b}{=}y_{<b}) \le \mathbb{E}_K[D_{\mathrm{KL}}(p_K(\cdot \mid y_{<b}) \,\|\, q_b(\cdot \mid y_{<b}))]\,.$$

The prompt x_b is a measurable function of $y_{<b}$, so conditioning on the history is legitimate and the data-processing step through x_b applies. Taking expectation over $Y_{<b}$ and summing over b yields the right-hand branch. The left-hand branch is the entropy bound $I(K; Y^B) \le H(K)$, valid for discrete K because $I(K; Y^B) = H(K) - H(K \mid Y^B)$ and conditional entropy is non-negative. Both hold simultaneously, so their minimum does. The same argument holds for $K \sim \pi$ arbitrary, replacing $\log |\mathcal{K}|$ by $H_\pi(K)$ and the per-query term by its π-weighted expectation. □

On the additive form. Writing the conclusion as $B \cdot D_{\mathrm{KL}} + \log |\mathcal{K}|$ rather than $\min\{H(K), B \cdot D_{\mathrm{KL}}\}$ replaces a ceiling by a floor. Under a uniform prior the additive expression is at least $\log |\mathcal{K}| = H(K)$ for all $B \ge 0$, so after the entropy cap it is identically $H(K)$ and can never be informative (Remark 2).

On the prompt. Eq. (2) assumes x_b depends on K only through the observed history. In Game A (Section 3) an auditor who queries the canary context c_K violates this, since the prompt then depends on K directly. Either the conditional law $P_{Y|K}$ must be defined to include that dependence, or the audit must be restricted to prompts chosen without knowledge of K. We flag this as an unresolved specification issue rather than papering over it: it is one reason we do not present the computed quantity as a bound (Section 4.2).

COMPUTED CONCENTRATION EXPRESSIONS (NOT VALIDATED)

For completeness we record what the pipeline computed. Let $\hat{\kappa}_1, \ldots, \hat{\kappa}_n$ be per-canary exposures and $\kappa_{\max} = \max_i \hat{\kappa}_i$. The Hoeffding-style expression is

$$\widehat{\mathcal{L}}_B \;=\; \min\Big\{H(K),\; B\big(\bar{\hat{\kappa}}_n + \kappa_{\max}\sqrt{\tfrac{\log(2/\delta)}{2n}}\big)\Big\},$$

and the Bernstein-style variant replaces the slack with $\sigma\sqrt{2\log(2/\delta)/n}$ for σ the sample standard deviation. On the T4 sub-panel at $\delta{=}0.01$ these evaluate to 2.645 and 2.242 nats at $B{=}1$.

These are not validated $1-\delta$ certificates, for three independent reasons, each sufficient on its own. (i) The $\hat{\kappa}_i$ are not independent: all canaries were injected into one corpus and one model, so their scores share parameters, optimizer trajectory, and generator templates, and template sharing is precisely the dependence Section 6.3 identifies empirically. (ii) Hoeffding requires a predetermined support bound; $\kappa_{\max}$ is the observed sample maximum, used without correction. (iii) The Bernstein expression omits the standard range-dependent $O(\kappa_{\max}\log(1/\delta)/n)$ term, so it is not the Bernstein bound even under independence. Repairing these requires independently trained models for cluster-level inference and a predetermined clipping bound, neither of which we have. We report the numbers only to document the computation behind Table 1.

VACUITY THRESHOLD

Setting the computed quantity equal to the ceiling, $B \cdot \bar{\hat{\kappa}} = H(K)$, gives Eq. (4), $B^* = H(K)/\bar{\hat{\kappa}}$. For $\bar{\hat{\kappa}} > H(K)$ we get $B^* < 1$, so the expression saturates before the first query. Table 1 instantiates this per type; the measured crossing for T4 at $B = 5$ is consistent with $B^* = 5.6454/1.8435 = 3.06$.

E ADDITIONAL SETUP DETAILS AND RESULTS

This appendix collects setup details, figures, and tables that support Sections 5 and 6 but did not fit in the main text.

E.1 TRAINING ARMS, WORKLOADS, AND STATISTICAL REPORTING

Table 8 summarizes the checkpoints used throughout the evaluation. Two additional checkpoints serve distinct purposes. An intentionally over-memorized 7B checkpoint provides a positive control for the extraction pipeline, while a leaky 0.5B checkpoint provides a non-degenerate setting for evaluating defenses because its undefended extraction rate is non-zero. These auxiliary arms differ from the primary target in training configuration; we therefore use them only for their stated purposes and do not interpret cross-arm differences as controlled model-scale effects.

Table 8: Training arms. Corpus mismatch between the primary target and the 7B and leaky-0.5B arms is a confound we do not attempt to apportion (Appendix F). "GPU-h" is wall-clock × devices for the final run; † the primary target's device count was not recorded, so we give wall-clock only; ‡ the leaky 0.5B arm's corpus size and epoch count were not recorded in the released run metadata ("n.r."), which is a reproducibility gap we flag rather than reconstruct: its evaluation panels held 4–8 canaries per type depending on seed.

Arm	Model	Corpus	Canaries	Epochs	GPU-h	Used in
Primary target	Qwen2.5-Coder-1.5B	500K docs	11,566 (seed 42)	3	45 h†	Sections 6.1–6.6
Matched reference	Qwen2.5-Coder-1.5B	**same** 500K docs	**none**	3	≈173	Section 6.3
Base reference	Qwen2.5-Coder-1.5B	— (not fine-tuned)	—	0	—	Section 6.1, 6.3
Over-memorized 7B	Qwen2.5-Coder-7B	700 docs	11,566	1	—	Section 6.2, 6.6
Leaky 0.5B	Qwen2.5-Coder-0.5B	seeded panels‡	4–8/type	n.r.‡	—	Sections 6.6–E.9

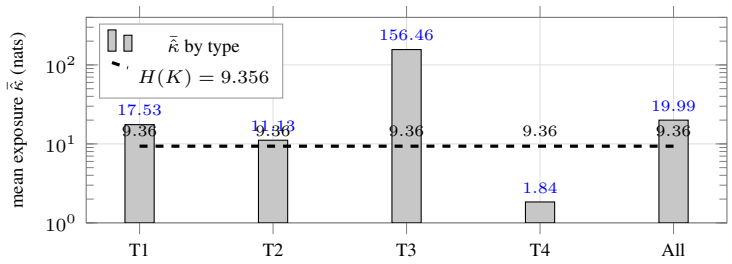

Figure 2: Exposure by canary type (log scale) against the panel entropy ceiling. Three of four types sit above $H(K)$; only vulnerability canaries (T4) fall below it, which is what makes T4 the sole type whose budget diagnostic is non-vacuous (Section 6.6).

Workloads. **W2** is a credential-extraction benchmark modelled on the study by Huang et al. (2024) of hard-coded credentials memorized by neural code completion tools. Their extraction artifacts are user-level and were withheld for privacy, so we generate 4,832 synthetic prompts with controlled ground truth and report this as a *comparable* benchmark, never a reproduction of their findings. **W3** measures utility in HumanEval (Chen et al., 2021) (164 problems, temperature 0.2, top-p 0.95, pass@1, identical checkpoint across defenses). **W4** is a code-secret suite over a stratified evaluation subset (7 prompts per canary) combining verbatim prefix probes, divergence attacks (Nasr et al., 2025), and structured-context probes (Terraform, Dockerfiles, CI YAML). **W5** rewrites each W4 prompt in five paraphrase modes (template variation, language variation, regex character class, natural-language description, and base64-decode variation), giving 5× the prompts.

Statistical reporting. We report exact Clopper–Pearson intervals and state the estimand and denominator for each (Section 6.5). Member/non-member separation is reported as AUC with the mean gap and a Kolmogorov–Smirnov statistic. defense comparisons use a bootstrap whose unit we state explicitly, since correlated prompts sharing a canary are not independent trials (10^6 resamples for the configuration comparison of Table 15, 5×10^5 for the multi-seed replication of Table 16). Every reported bound-shaped value is run through an automated entropy-cap audit (Appendix G.2).

E.2 Exposure by canary type

Two features of this table drive everything that follows. The mean exceeds the panel entropy, which by Eq. (4) forces $B^* < 1$ and makes the full-panel certificate vacuous at every budget. And the type gradient spans two orders of magnitude (1.84 nats for T4 against 156.46 for T3), which is what motivates type-stratified accounting. Both observations are correct as arithmetic. Whether either reflects *memorization* is the question Section 6.3 settles.

E.3 Positive control

Table 9 is the licence to read the rest of this section as evidence about models rather than about our implementation, and it is the control memorization audits most often omit. Three caveats bound what

Table 9: Positive control. The pipeline that returns 0% on the 1.5B target recovers 10–16% from an over-memorized 7B checkpoint, so the 0% is a property of the model, not of the attack code.

Attack surface	1.5B target	7B over-memorized
W4 static templates	0.00%	10.3%
W4 adaptive (B=100)	0.00%	16%
W5 paraphrase (5 modes)	0.00%	≈8%
Final training loss	—	0.044

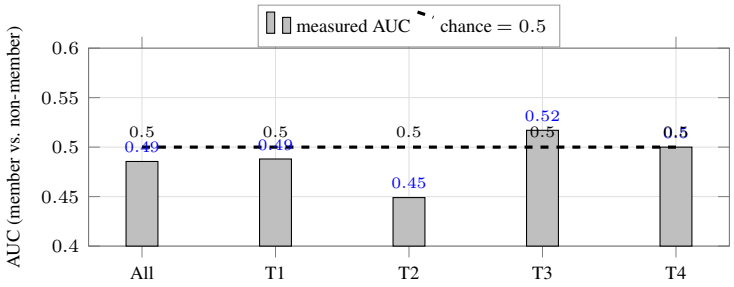

Figure 3: No canary type separates members from non-members. T2 scores *below* chance and T4 sits exactly at it. Under a genuine memorization signal every bar would rise well above the dashed line. Compare Figure 2: the type with the largest exposure (T3, 156 nats) achieves AUC 0.517.

it supports. Its corpus is 700 documents against the target's 500K (Table 8), so it is a memorization stress test, not a controlled scale comparison; the gap is consistent with model scale *or* training intensity and we do not apportion between them. Neither the content filter nor LEAKCERT reduced the 7B's extraction rate, which we flag as an unresolved anomaly, unexpected given both raise attacker cost on the 1.5B target. Finally, the per-canary output files for this run were lost to a storage-configuration error, so the aggregates come from the run's recorded summary rather than raw output we can re-inspect; the configuration and launch scripts are released and the run is repeatable from the checkpoint (Appendix J).

E.4 CONTROL I: ADDITIONAL DETAIL

We audited for a pipeline defect and found none. Before accepting a null we checked the three ways it could be an artifact. *Panel independence*: the generator gives each seed a private random stream for both secret value and context template, so seeds 42 and 999 generate disjoint secret sets, which we verified by explicit comparison of the two generated sets rather than inferring it from the seeding alone. *Replay fidelity*: the member panel is reconstructed with the exact injection-time parameters, and subset selection is deterministic and first-seen ordered, and we verified against the injection manifest that the replayed panel is a subset of what was injected. *Injector cleanliness*: canaries are added as independent documents and never modify existing ones, so the two corpora differ by exactly the canary documents. No defect was found; we accept the null as an experimental finding.

What this retracts. The claim that our target exhibits strong memorization (which Table 1 appears to support overwhelmingly) is not sustained. We are careful about direction: it is *not* that the model memorizes nothing, and not that exposure is useless in general. It is that a reference-model exposure score reported without a non-member panel does not license a memorization claim, even when the reference is correctly matched.

E.5 CONTROL II: ADDITIONAL DETAIL

Two things survive. Verbatim recovery is 0% on members and non-members alike, consistent with every other run. And the *defense-cost* signal is real, though it too is membership-blind: LEAKCERT forces 2.88 queries on T1 members against B1's 1.08, and a near-identical 3.00 on T1 non-members.

Table 10: The null is reference-invariant. Training a reference on the identical corpus minus the canaries (the standard remedy for the fine-tuning-shift confound) does not recover separation.

Reference	Corpus	AUC	gap (nats)
Base model	not fine-tuned	0.49	$+0.001$
Matched (ours)	500K docs, no canaries	0.4855	-0.0149

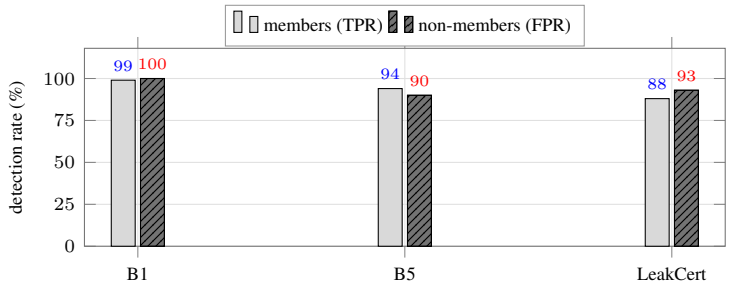

Figure 4: The membership test's false-positive rate tracks its true-positive rate at every defense. A valid membership signal would show a large gap between the paired bars; here LEAKCERT even detects *more* non-members than members.

Its suppression stage raises the cost of the test regardless of what is being tested: a defense effect, not a memorization measurement. A corrected membership attack must score the *secret's* likelihood ratio calibrated against non-members, which is the quantity the exposure estimator computes and which Section 6.3 shows to be at chance here.

E.6 CONTROL III: CAVEATS AND THE ONE HIT WE ROUNDED AWAY

W4 fires seven prompts per canary and W5 thirty-five, over only 100 distinct canaries; repeated probes of one secret are not independent evidence about more secrets. Two further caveats belong with these numbers. The superpopulation is the canary generator, not the prompt set: the interval is a statement about canaries drawn from $\mathcal{G}$, and the prompt suite is a fixed benchmark rather than a sample, so the per-request row describes only the benchmark it was computed on. And the values are upper endpoints of *two-sided* 95% intervals; the corresponding one-sided 95% bounds are 0.427%, 0.086%, and 2.951%. We report the two-sided endpoints for continuity with our earlier reporting and name the convention explicitly, since the two are often conflated. Both estimands are legitimate, but the prose around a 0% result ("the target does not leak its secrets") is a claim about secrets, and the per-canary bound is $6.9\times$ and $34.4\times$ larger than the two per-request figures. Both should be reported, with the per-canary figure carrying the headline.

The one hit we rounded away. A related habit is worth naming. Our sampled-decoding sweeps report 0.00%, but one audit row did match: across the $12{,}600$ audit rows of one run (100-canary panel; W4 and W5 prompts under all three defenses) exactly one row matched: a `natural_language`-paraphrase prompt against the undefended target, i.e. one hit in the $3{,}500$ undefended W5 prompts. Exact CP interval on that denominator: 0.029% $[0.0007\%, 0.159\%]$. On inspection we classify it as an incidental pattern match rather than a confirmed verbatim leak, and it did not reproduce under greedy decoding or under any defense. It remains one non-zero observation that a rounded 0.00% conceals.

E.7 THE BUDGET DIAGNOSTIC IN FULL

Per Section 4, we do not claim the quantity below is a bound on $I(K; Y^B)$. What it is: $B \cdot \bar{\hat{\kappa}}$, the expression a certificate of the form Eq. (2) would evaluate if the exposure statistic were a valid KL estimate. Comparing it to the entropy ceiling answers a narrower but still useful question (*is this panel even arithmetically capable of supporting a non-trivial bound, at any budget?*), and the answer, shown in Figure 5, is almost always no.

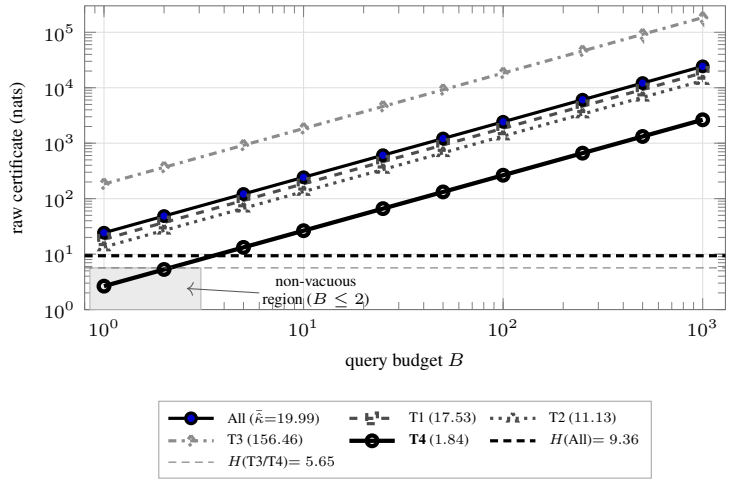

Figure 5: Raw certificate vs. query budget, log–log. A certificate is informative only *below* its panel's entropy ceiling; we draw the two ceilings that bound the region of interest; T1's (9.210) lies just under the upper line and T2's (6.908) between the two. Every curve except T4 starts above its own ceiling already at B=1, so those certificates are vacuous at every budget. Only the weakly memorized T4 panel enters the shaded region, and only for $B \in \{1, 2\}$: the single non-vacuous configuration in this study, matching the predicted crossover $B^* \approx 3$.

Table 11: Full-panel budget diagnostic, both targets, base reference. Capped values equal $H(K)$ everywhere.

Target	B	Raw (nats)	Capped	Vacuous
1.5B	10^2	2,429	9.356	yes
	10^3	24,294	9.356	yes
	10^4	242,941	9.356	yes
	10^7	2.4×10^8	9.356	yes
7B	10^2	3,216	9.356	yes
	10^4	321,552	9.356	yes
	10^7	3.2×10^8	9.356	yes

The full panel exceeds its ceiling before the first query. Because $\bar{\hat{\kappa}} = 19.99 > H(K) = 9.356$, Eq. (4) gives $B^* = 0.47 < 1$. Table 11 confirms this across five decades of budget on both fine-tuned targets: after capping, the quantity equals $H(K)$ everywhere, which is the trivial statement that an attacker learns at most the panel's entropy. The "tightness ratio of 1.0" the pipeline reports reflects two quantities both pinned to the cap and carries no information.

Type stratification yields one sub-ceiling cell. Table 1 applies Eq. (4) per type. Three of four types have $B^* < 1$: their mean exposure alone exceeds their sub-panel entropy, so no budget helps. The T4 vulnerability panel ($\bar{\hat{\kappa}} = 1.84$, $H = \ln 283 = 5.645$) is the exception, staying below its ceiling at $B \in \{1, 2\}$ (2.645 nats by the Hoeffding expression, 2.242 by Bernstein) and crossing at B=5, matching $B^* \approx 3.06$.

We deliberately do *not* convert this into an extraction-probability statement. Doing so requires Fano's inequality over a local candidate set in which exactly one of κ structurally identical strings was injected; our 283 T4 canaries are all members occupying different slots, so the conversion we previously reported was mis-specified and is withdrawn (Section 4.3). Nor is the cell evidence about memorization: T4's member/non-member AUC is exactly 0.500 (Figure 3). What the row shows is narrower and still worth stating: the only panel arithmetically capable of supporting a non-trivial bound is the one the model barely memorized, which is the opposite of where an auditor would want a guarantee.

Table 12: Analytic comparison at $B{=}10^4$ (1.5B, full panel). Both quantities cap at $H(K)$; the DP column is the analytic Rényi composition bound, as no per-ϵ DP checkpoint completed training.

DP ϵ	LEAKCERT capped	Raw DP bound	DP capped
1	9.4	5,000	9.4
2	9.4	20,000	9.4
4	9.4	80,000	9.4
8	9.4	320,000	9.4
16	9.4	1.28×10^6	9.4

Table 13: W3 utility on the 1.5B target (HumanEval, 164 problems, pass@1, identical checkpoint, most recent run). Absolute pass@1 is low because the target is a small non-instruction-tuned base fine-tuned on a canary-injected corpus; an earlier run of the same configuration reorders the arms (see text), so no ordering should be inferred.

Defense	pass@1	correct/164	refusal
B1 no defense	6.1%	10	0.0%
B2 temperature 0.5	4.3%	7	0.0%
B3 top-p 0.7	6.7%	11	0.0%
B5 content filter	4.3%	7	0.0%
LEAKCERT	6.1%	10	0.0%

Prior sensitivity and the analytic DP comparison. Replacing the uniform prior with the empirical type prior (Section 4.1) moves $H(K)$ from 9.356 to 9.250 nats and the capped certificate from 9.4 to 9.3; both remain at the cap, so prior information helps through stratification rather than the global entropy correction. Table 12 compares our quantity to the analytic Rényi composition bound $B\epsilon^2/2$ at $B{=}10^4$. Both exceed $H(K)$ and cap: on this target, neither an empirical exposure accounting nor an analytic DP promise at these ϵ values says anything non-trivial. No DP checkpoint was trained, so the DP column is analytic only.

E.8 UTILITY, REFUSAL, AND OVERHEAD

Table 13 reports HumanEval pass@1 on the 1.5B target with the identical checkpoint under every defense. LEAKCERT matches the undefended baseline at 6.1% (10/164) with a 0% refusal rate. We caution against reading the ordering: an earlier run of the same defenses on the same checkpoint reports 5.5/8.5/4.9/4.9/6.7%, reordering the arms entirely. The spread across defenses (4.3–8.5% across both runs) is sampling noise at $n{=}164$ with one sample per problem, which is precisely why we decline the equivalence claim. A matched diagnostic on the leaky 0.5B checkpoint puts B1, B5, and LEAKCERT at an identical 10.98% (18/164) with LEAKCERT refusing 0.61%, and an error breakdown attributes the dominant failure mode identically across defenses (`AssertionError`, 40 of 164 in all three arms), confirming the defense layer is not the bottleneck. Per Observation 3 we read this as no detected difference at low power, not equivalence.

The refusal threshold has a sharp knee. Figure 6 sweeps the learned-refusal threshold. Below $t{=}0.9$ the classifier refuses a large fraction of legitimate completions (45% at $t{=}0.7$, which costs almost all utility), while $t \geq 0.95$ holds refusals at or below 2.5% with pass@1 at its plateau. This is what selects $t{=}0.95$ as the operating point, and it is the clearest instance in our data of the utility–leakage tension that we declined to state as a theorem (Section 4.3).

Overhead. The monitor is not deployable as evaluated. Median per-decision latency on the 1.5B target is 52 s in the most recent run and 80 s in an earlier one, dominated by the reference-model forward pass the exposure stage requires. On the leaky 0.5B checkpoint the learned-only variant at $t{=}0.95$ measures 12.0 s median / 14.5 s p99 over the 80-problem threshold sweep, but 19.1 s / 28.9 s and 32.8 s / 39.5 s in two runs over the full 164 problems; we quote the range rather than the most favourable figure, since the spread is 2.7× and we cannot attribute it to configuration. We therefore characterize LEAKCERT as an *offline audit monitor*; reaching interactive latency (by caching per-key

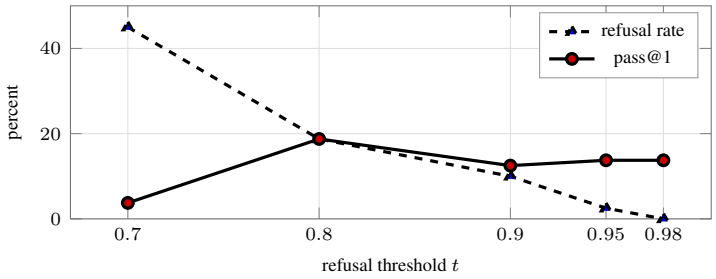

Figure 6: Refusal-threshold sweep on 80 HumanEval problems. Aggressive thresholds destroy utility by refusing legitimate completions; t=0.95 is the knee, holding refusals at 2.5% with pass@1 at plateau.

running exposure, or decoupling certificate updates from the request path) is future work, not a claim of this paper.

E.9 Credential-extraction workload

On the full 4,832-prompt credential-extraction benchmark, run on the leaky 0.5B checkpoint, no configuration extracted any controlled ground-truth secret (zero hits under B1, B5, and LeakCert; LeakCert refusal 0.08%). We report this as an operational negative control establishing that the benchmark and scorer path work end to end, not as a reproduction of the findings of (Huang et al., 2024), whose user-level artifacts were never released. A three-example public supplementary probe over AWS and Slack credential formats likewise returned $0/3$ under all six defenses.

F Limitations

One training run is the binding limitation. The member/non-member analysis scores 1,132 canaries per side, but all come from a single target/reference training pair. They are not 2,264 independent observations of the training phenomenon: the natural hierarchy is training seed → template → canary → prompt, and the relevant independent unit for generalizing about fine-tuning is the training run. Accordingly, the intervals in Table 4 characterize within-run uncertainty and understate across-training variability. Establishing generality requires multiple independently trained target/reference pairs with cluster-aware or mixed-effects inference.

Limited injection regime and panel design. Our central null result covers one model scale, one injection regime (each canary inserted once over three epochs), and one panel structure. It does not imply that the target memorizes nothing. A repetition sweep would be needed to determine the estimator's detection floor. Moreover, heavy template reuse may induce format-level generalization that raises both member and non-member scores. We did not characterize template counts or evaluate template-disjoint non-members and per-slot near-neighbour decoys, so we cannot determine whether the null is specific to this panel design or general to reference-ratio audits.

Positive-control and cross-arm confounds. The over-memorized 7B arm is trained on only 700 documents versus 500K for the primary target, so it validates the extraction pipeline but does not isolate model scale from training intensity. Its per-canary artifacts were also lost. More importantly, this positive control applies only to *extraction*: we did not evaluate member/non-member exposure separation or the membership test's TPR–FPR gap on that checkpoint. Each instrument therefore still needs its own positive control, ideally through a repetition sweep that holds architecture and corpus fixed.

Theoretical and access limitations. The concentration expressions reported in Appendix D are not valid certificates because canary scores are dependent through shared training and templates, the range parameter is estimated from the observed sample, and the Bernstein form omits its range term. The likelihood-based audit also requires log-probability access (Section 3); we provide neither a sample-only estimator nor its sample complexity. The method is therefore not directly applicable to text-only completion APIs.

Extraction coverage and attack strength. Extraction is scored primarily by exact string match. Semantic or behavioural leakage from T2–T4 may therefore go undetected; semantic, AST-level, or vulnerability-pattern scorers would provide a broader measure. Our adaptive attacker is also budget-limited: runs beyond B=100 exceeded wall-clock limits, and two defended adaptive cells did not complete. Stronger decoding and prompt-search strategies could surface leakage that our evaluation misses.

Defense evaluation. defense efficacy is evaluated on the leaky 0.5B arm rather than the primary target. The refusal classifier's training and evaluation data are not fully disjoint, the threshold was selected on a HumanEval subset overlapping utility evaluation, and prompt-level bootstrap intervals are optimistic because prompts sharing a canary are correlated. A nested tune–freeze–evaluate protocol with canary-level resampling is needed before interpreting the observed improvements as out-of-sample. We also include no trained DP baseline.

Utility and deployability. Utility is measured with one sample on each of 164 HumanEval problems at low pass@1, which is insufficient for an equivalence or non-inferiority claim. Finally, the monitor requires approximately 52–80 s per decision in our evaluation and should therefore be viewed as an offline audit tool rather than an inline defense.

G Supplementary Detail

G.1 Exposure estimator

For canary k_i with context c_i,

$$\hat{\kappa}_i = \log p_\theta(k_i \mid c_i) - \log p_{\text{ref}}(k_i \mid c_i),$$

computed by teacher forcing over the secret's token span under both models and summing per-token log-probabilities. Negative values are clamped to zero before aggregation; the raw signed statistic is retained for the member/non-member comparison of Table 4, where clamping would destroy the very comparison at issue. This is why the means in that table ($\approx$ 49 nats, signed, matched reference, n=1,132) are on a different scale from those in Table 1 (19.99 nats, clamped, base reference, n=11,566); both are correct for their stated configuration.

A per-instance $\hat{\kappa}_i$ may numerically exceed $H(K)$ without contradiction: it is a single-instance statistic, whereas the entropy bound constrains the aggregate population mutual information. Conflating the two produced the earlier, incorrect assertion that a single completion could carry more information than the panel's entire mutual information. The entropy cap therefore belongs at the aggregate level, after averaging, not on individual scores. We report the panel maximum (284.08 nats) uncapped and exclude it from the aggregated quantity.

G.2 Entropy-cap audit

Every certificate and mutual-information value passes an automated audit flagging any raw value exceeding $H(K)$. Applied retrospectively across 23 historical result files from our own earlier runs, the audit checked 155 values and flagged 108 as exceeding the ceiling; 14 of 23 files failed outright (four passed; five lacked a recorded $H(K)$). Those raw values are retained only as diagnostics, and all reader-facing quantities are capped. We recommend reporting the triple (raw, capped, vacuity flag) as default, since a capped value alone hides how far the raw bound overshot, in our tables, by up to seven orders of magnitude (Table 11).

G.3 Informative-budget diagnostic

As a sanity check on the vacuity machinery we ran the budget sweep on two additional exposure sources over a 392-canary panel ($H = 5.971$ nats). A high-exposure diagnostic source saturates the cap at every budget from B=1 to B=2000, with B/B^* ranging from 30.8 to 6.2×10^4, i.e. overshooting the non-vacuity threshold by one and a half to nearly five orders of magnitude. A zero-exposure control source yields a certificate of exactly 0.000 nats at every budget, confirming the non-vacuous path is reachable when the estimated exposure is genuinely zero. Neither is a memorization result; both verify that the cap and the vacuity flag behave as specified.

Table 14: Cross-seed baselines (leaky 0.5B arm, mean of seeds 42–43). Sampling controls do not help.

Defense	W4	W5	W5/W4
B1 no defense	9.38%	8.12%	0.861
B2 temperature 0.5	12.95%	10.27%	0.793
B3 top-p 0.7	13.17%	9.51%	0.721
B5 content filter	11.38%	6.70%	0.586
LEAKCERT (full)	9.16%	7.00%	0.764

G.4 CLOPPER–PEARSON COMPUTATION

Intervals in Table 3 are exact Clopper–Pearson bounds obtained by bisection on the binomial CDF. For zero successes the upper bound reduces to $1-(\alpha/2)^{1/n}$, which we use as a closed-form check: n=700 $\Rightarrow 0.5256\%$; n=3,500 $\Rightarrow 0.1053\%$; n=100 $\Rightarrow 3.6217\%$. For the single sampled-decoding hit, x=1, n=3,500 gives 0.029% $[0.0007\%, 0.159\%]$. The routine is released so the estimand distinction can be re-derived independently.

G.5 PANEL CONSTRUCTION AND DISJOINTNESS

The generator assigns each seed a private pseudo-random stream from which both the secret value and the context template are drawn, so panels built with seeds 42 and 999 have disjoint secret sets (verified by explicit set comparison, not inferred from the seeding) while sharing format, context distribution, and character classes. Stratified subset selection is deterministic in first-seen order, so a replayed member panel built with injection-time parameters is a subset of the injected set, which we verified against the injection manifest. The corpus injector inserts each canary as an independent additional document at a random position and never modifies existing documents, so the matched reference corpus differs from the target corpus by exactly the canary documents.

G.6 CROSS-SEED BASELINES

Table 14 reports the sampling-control baselines on the leaky 0.5B arm, averaged over seeds 42–43. Neither temperature nor top-p reduces extraction relative to no defense in this setting: both *increase* it, consistent with sampling controls trading diversity for mode-seeking behaviour that favours memorized continuations. Only the content filter and the learned refusal variant reduce it.

H DEFENSE EVALUATION IN FULL

This appendix gives the complete defense evaluation summarised in Section 6.6. All results are on the leaky 0.5B checkpoint; none of them is a memorization-defense result, for the reasons Section 6.6 states and Appendix H.5 quantifies.

H.1 THE LEAKCERT RUNTIME MONITOR

Figure 7 shows the four-stage monitor that accompanies the audit: a certificate-budget throttle, a per-key rate limiter, an uncertainty-refusal classifier, and regex/hash suppression. We evaluate four configurations: *full* (all four stages), *calibrated-aggressive*, *calibrated-softheuristic* at t=0.95, and *learned-only* at t=0.95 (uncertainty refusal alone). They are compared with B1 (no defense), B2 (temperature 0.5), B3 (top-p 0.7), and B5 (content filter: a regex bank, Meli et al., 2019, plus a learned secret classifier, Saha et al., 2020).

H.2 MONITOR CONFIGURATIONS

Table 15 compares four LEAKCERT configurations against the content filter under a 10^6-resample bootstrap. This is a result we would have preferred to omit and consider important to report: the *full* monitor (the configuration the architecture of Figure 7 describes) does *not* beat the content filter, and neither does the soft-heuristic calibration. Only the two learned variants do.

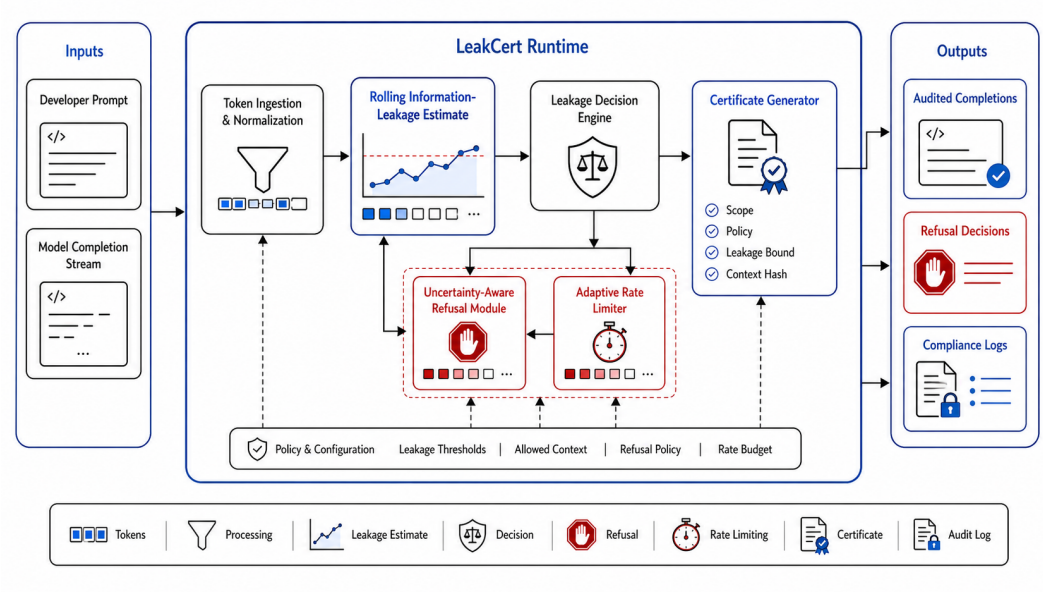

Figure 7: The four-stage monitor: certificate-budget throttle, per-key rate limiter, uncertainty-refusal classifier, and regex/hash suppression. We evaluate it as an *offline audit tool*; decisions are computed and logged, not served inline (Section 6.6).

Table 15: Monitor configurations vs. the B5 content filter on W5 paraphrase extraction (10^6-resample bootstrap). Two of four configurations fail to beat the baseline; the *full* monitor is one of them.

Configuration	Seed	B5	Variant	$\mathbb{P}(\text{B5}>\cdot)$
Full monitor	43	5.71%	5.89%	0.410
Full monitor	42	7.68%	8.13%	0.333
Calib. soft-heuristic	43	5.71%	6.43%	0.226
Calib. aggressive	43	5.71%	**2.32%**	**0.99998**
Learned-only t=0.95	43	5.71%	**2.50%**	**0.99993**
Learned-only t=0.95	42	7.68%	**4.73%**	**0.99787**

H.3 Multi-seed replication

Figure 8 and Table 16 give the replication. The direction holds in every seed, with an aggregate reduction from 5.98% to 2.84% over 5,600 paired prompts.

H.4 Per-paraphrase-mode breakdown

Figure 9 breaks W5 down by the five rewriting modes. The content filter is roughly flat across modes at 4.6–6.3%, consistent with a lexical defense that each mode evades about equally; the learned variant reduces every mode, most strongly on natural-language descriptions (5.51% $\rightarrow$ 1.04%), which are the furthest from any regex the filter encodes.

H.5 Component ablation

Table 17 removes one monitor stage at a time on two panels. The panels disagree about *which* output-side stage matters: on the larger panel removing the rate limiter raises W5 extraction from 0.00% to 4.46% while removing refusal leaves W5 at 0.00%; on the smaller panel the pattern reverses: and we report both rather than the more convenient one. What both agree on is the finding that matters for this paper's thesis: removing *certificate accounting* changes nothing on either panel, in either workload. We have no experiment in which certificate accounting alters a security decision the

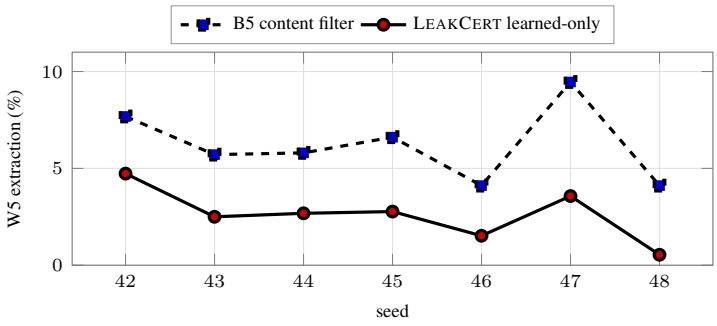

Figure 8: Paraphrase (W5) extraction across seven seeds. The learned-only refusal variant is below the content filter in every seed. Seeds 47–48 use a smaller panel (Table 16).

Table 16: W5 extraction, B5 vs. learned-only LEAKCERT. Seeds 42–46 use 1,120 W5 prompts each; 47–48 use 560. Rates and Δ in percent / percentage points. Bootstrap intervals resample prompts and are therefore optimistic (Appendix F).

Seed	B5	LEAKCERT	Δ	95% CI	$\mathbb{P}$
42	7.68	4.73	2.95	[0.98, 4.91]	0.998
43	5.71	2.50	3.21	[1.61, 4.82]	1.000
44	5.80	2.68	3.13	[1.52, 4.82]	1.000
45	6.61	2.77	3.84	[2.14, 5.63]	1.000
46	4.11	1.52	2.59	[1.25, 3.93]	1.000
47	9.46	3.57	5.89	—	—
48	4.11	0.54	3.57	—	—
Agg. 42–46	5.98	2.84	3.14	[2.39, 3.91]	1.000

other stages would not have made, and we do not claim one. The defense evidence here supports output-side mechanisms, not the certificate architecture that motivates them.

H.6 ALL MEASURED EXTRACTION IS VULNERABILITY CANARIES

A per-type breakdown of the same ablation reveals something the aggregate rates hide: in *every* configuration, T1 literal, T2 paraphrase, and T3 semantic extraction are exactly 0.00%, and the entire aggregate is carried by T4 vulnerability canaries: 50.0% under B5 and 25.0% under LEAKCERT on Panel A's W4. Since T4 canaries are short insecure-code idioms (string-concatenated SQL, disabled host-key checking) that a code model has ample reason to emit without having memorized any particular instance, we caution that this arm measures suppression of *vulnerability-pattern generation* rather than recovery of planted secrets. The defense result is real and replicated; its security interpretation is narrower than "prevents secret extraction."

I ETHICS CONSIDERATIONS

We assessed this work against the Menlo Report principles. No real user data, live credential, or third-party secret was created, handled, or placed at risk.

Synthetic canaries only. Every canary is synthetic. Credential-shaped canaries imitate the *format* of real credential types (so that the audit is realistic) but their key bodies are randomly generated and correspond to no issued credential: JWTs are signed by a research-only key, RSA fragments were generated for the panel and never used, and licence keys follow an invented vendor scheme. No canary is or has been valid for any account, service, or person, and none was obtained from any real repository. Canaries are tagged in the corpus and excluded from utility evaluation. Because the strings deliberately resemble credentials, we exclude raw benchmark files from the released package (they would trigger secret-scanning systems) and ship deterministic regeneration scripts instead.

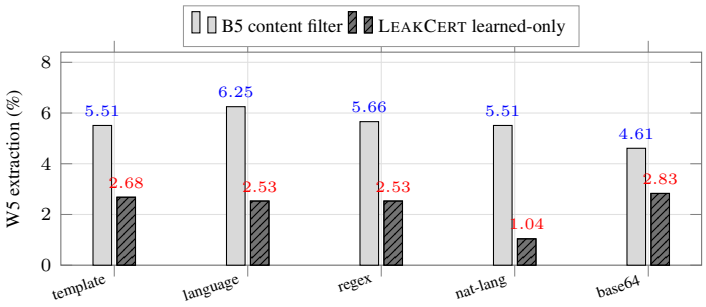

Figure 9: W5 extraction by paraphrase mode, mean over seeds 44–46. The learned variant reduces every mode; the largest gap is on natural-language descriptions, the mode least reachable by a regex bank.

Table 17: Component ablation on two panels (Panel A: 224 W4 / 1,120 W5; Panel B: 112 / 560). The panels disagree on rate limiting vs. refusal; both agree that removing certificate accounting has no effect.

	Panel A		Panel B	
Variant	W4	W5	W4	W5
B5 content filter	12.50	10.71	12.50	12.32
LEAKCERT full	6.25	0.00	6.25	6.07
− rate limiter	6.25	**4.46**	6.25	6.07
− cert. accounting	6.25	0.00	6.25	6.07
− suppression	6.25	0.00	6.25	6.07
− learned refusal	**12.50**	0.00	**12.50**	**12.32**

No real user data in the credential-extraction workload. The extraction artifacts of Huang et al. (2024) contain real user-level data and were withheld by their authors on privacy grounds. We did not attempt to reconstruct them. Our comparable benchmark is fully synthetic and every row is marked as containing no real PII. A separate public-supplementary probe used only three already-public examples.

Vulnerability canaries. T4 canaries are insecure coding patterns (string-concatenated SQL, disabled host-key verification, unsafe deserialization). They are short, widely documented anti-patterns rather than exploits, and we report them to study whether models reproduce insecure *style*; we publish no working exploit.

Attack code and dual use. The attacks we implement (template, divergence, paraphrase, perplexity-ranked, and likelihood-ratio) are established in the literature; we introduce no new extraction capability, and our strongest attacker requires knowledge of the canary set, which deployed adversaries lack. This paper's principal offensive-side findings are *negative*: a widely available membership test is invalid, and a widely used exposure statistic does not evidence memorization without a control. Publishing them reduces rather than increases attacker capability and protects practitioners from acting on flawed audit evidence.

Human subjects. No human subjects, human-derived data, or user studies are involved; no IRB review was required.

Reporting of corrected and retracted claims. This paper retracts three empirical findings and five theoretical claims we previously held, reports that two of four of our own monitor configurations fail to beat a regex baseline, reports that our certificate-accounting stage contributes nothing measurable by ablation, and reports one finding (defenses failing on the 7B arm) as unresolved rather than omitting it. We regard surfacing such results as an obligation of security measurement work, and we have tried to state the direction of each negative result precisely so that it is not read as a stronger claim than the evidence supports.

Use of AI tools. Our disclosure of generative AI use, as required by the ICLR 2027 AI Policy for Authors, is given in the AI use statement preceding the references.

J Artifact Availability and Open Science

The artifact accompanying this submission is available to reviewers as a frozen, anonymous, tracking-free archive at `https://anonymous.4open.science/r/LeakCert-42AC/README.md` (no login, no analytics, no author-identifying metadata). The same contents are mirrored in the supplementary-material upload. Upon acceptance we will publish the repository and mint a DOI. Every file in the archive is listed with its SHA-256 digest in `MANIFEST.sha256`, and every table and figure in this paper names the artifact file it was generated from.

What we release. (i) The complete LeakCert pipeline: canary generation and injection, the exposure/LLR estimator, certificate computation with its entropy-cap audit, the four-stage runtime monitor, and all attack implementations including the false-positive control harness. (ii) Training and evaluation configurations for every arm in Table 8, with fixed seeds. (iii) Per-run JSON artifacts backing each table and figure, with SHA-256 manifests, including the member and non-member exposure distributions behind Table 4 and Figure 3, the SPRT member/non-member results of Table 5, the full budget sweep behind Figure 5, the seven-seed W5 results of Table 16, both component ablations of Table 17, and the threshold sweep of Figure 6. (iv) The exact Clopper–Pearson routine used for Table 3, so the estimand distinction can be re-derived independently. (v) The credential-extraction benchmark generator.

Reproducing the central result. The negative result of Section 6.3 is reproducible without retraining: given the released target checkpoint, the matched reference checkpoint, and the seed-42/seed-999 panel generators, recomputing the member and non-member exposure distributions and their AUC is a few GPU-hours of inference. We consider this the artifact a reviewer should run first, and we provide a single entry point for it.

What we cannot release, and why. Raw canary files are withheld because they are designed to resemble credentials and would trigger secret-scanning systems; deterministic regeneration scripts with the original seeds are provided instead and produce byte-identical panels. Model checkpoints are large (the 1.5B target and its matched reference are ≈ 3 GB each). Because regenerating the matched reference costs ≈ 173 GPU-hours, checkpoint availability (not code availability) is the binding constraint on reproducing our central result, so we commit to hosting both checkpoints on a research-data repository with per-file SHA-256 digests rather than treating this as best-effort. Training scripts, corpus manifests, seeds, and step counts are included regardless.

Known artifact gaps. Two artifacts referenced in this paper were lost to a storage-configuration error and are not recoverable: the per-canary and per-defense output files for the 7B run of Table 9 (which is why the defense anomaly in Section 6.2 is reported as unresolved, and why that row is backed by the run's recorded aggregate figures and its configuration rather than by raw per-canary output) and the raw log for one large sampled-decoding sweep. The 7B evaluation configuration and launch scripts are released, so the run is repeatable from the checkpoint even though its original outputs are not archived. Where this affects a claim, we restate the claim on a smaller sweep whose artifacts survive and which we do release (Section 6.5). We flag these gaps rather than quoting numbers we cannot back with a checkable artifact.