THE VERIFIER-BIT LEDGER: A LEAKAGE CEILING AND AN EXECUTABLE AUDIT FOR VERIFIER-IN-THE-LOOP REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Verifier feedback drives much of the recent progress in iterative reasoning, yet the channel that carries it can also carry the answer: on hidden-test pass rate, a model that learns to repair looks identical to one fed the hidden tests, and contamination audits, which cover only static training data, cannot say with a guarantee how much of a gain the feedback could have supplied. We introduce the *verifier-bit ledger*. Its budget meters what each round of feedback reveals about the hidden answer; its ceiling bounds the improvement any policy can buy with that budget over a ghost baseline whose verifier is keyed to decoy answers, and is tight to first order as the budget vanishes and exactly tight for declared feedback alphabets. Inverting the ceiling yields an executable audit that certifies an unlogged information path, at a chosen confidence level, whenever improvement outruns the budget; we characterize exactly when it can fire, and it is deployable today for low-rate or structurally capped feedback. Two conservation theorems show that compliant training cannot smuggle answers and that amortized gains are prepaid. On a synthetic family where every quantity is exactly computable, the bounds close with matching constants and strong injected side channels are caught at the predicted confidence. On real code repair, free-text feedback is priced rather than tested, and a planted leak lifts repair success by five points over the ghost; on three controlled real-task protocols with declared or capped feedback, planted leaks trigger the certificate and honest runs never do. A residual audit conditioned on legally disclosed feedback extends certificates to full first-failure feedback on randomized suites, catching a one-test exploit the ceiling cannot. Verification information can be budgeted, audited, and priced like computation.

1 INTRODUCTION

Verifier-in-the-loop refinement is a standard recipe for improving language-model reasoning: a policy proposes a candidate, a verifier returns feedback (a failing unit test, a rejected proof step, a grade), and the policy revises. It covers self-refinement, reflective agents, and self-debugging (Madaan et al., 2023; Shinn et al., 2023; Chen et al., 2024), dates to counterexample-guided synthesis (Solar-Lezama et al., 2006), and richer feedback can sharply raise accuracy at a fixed sampling budget.

Unfortunately, the mechanism has a built-in weakness. Every round of feedback is computed from the hidden specification (the held-out tests or reference solution that will later score the model), so the channel that carries the learning signal can also carry the answer, and hidden-test pass rate cannot tell a model that learned to repair from one that was fed the answer. Existing leakage tools do not watch this channel: contamination audits and membership inference examine overlap with the training corpus (Shokri et al., 2017), and the shuffle or ablation checks used in practice reassure without bounding anything. We therefore ask how much improvement a given amount of feedback information can buy, and whether an auditor can certify from samples that a deployment received more information than its logged feedback allows.

Information accounting is the natural tool: adaptive data analysis meters queries against a reusable holdout (Dwork et al., 2015b; Blum & Hardt, 2015; Russo & Zou, 2016), and privacy audits turn planted canaries into confidence-calibrated bounds (Jagielski et al., 2020; Steinke et al., 2023). How-

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.

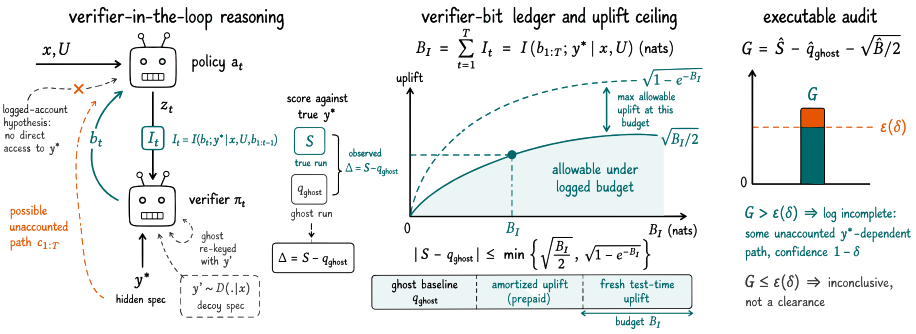

Figure 1: The verifier-bit ledger. *Left:* feedback is computed from the hidden specification $y^\star$, and an unlogged path $c_{1:T}$ may bypass the transcript. *Center:* the budget B_I meters the leakage; success splits into a decoy-driven ghost baseline plus amortized (prepaid) and fresh test-time uplift, and the ceiling of Theorem 1 caps the uplift. *Right:* inverting the ceiling yields the audit statistic G (Theorem 2), which certifies an unaccounted $y^\star$-dependent path when $G > \varepsilon(\delta)$.

ever, these meters point at the training pipeline or at population overfitting, whereas verifier feedback opens a per-instance channel during evaluation, and aiming a meter at it meets two obstacles. There is no ready baseline, since deleting the feedback also deletes its legitimate value and raw uplift mixes help with leakage; and a sound meter must charge every symbol that could hide the answer, a conservatism that threatens vacuity.

We build such an instrument, the *verifier-bit ledger* (Figure 1), and both obstacles turn out to have exits. The meter is a budget B_I, the per-round conditional mutual information between feedback and the hidden specification, which equals the leakage of the whole interaction (Lemma 1). The baseline is a *ghost* run whose verifier is keyed to a decoy specification while the output is scored against the truth, so interaction keeps its value but loses its dependence on the answer. Against this baseline no policy gains more than $\min\{\sqrt{B_I/2}, \sqrt{1 - e^{-B_I}}\}$ (Theorem 1); the constant is first-order optimal (Proposition 1), an exactly tight cardinality ceiling takes over for declared alphabets (Proposition 2), and a functional refinement that charges only grading-relevant information relieves the conservatism (Proposition 3). Inverting the ceiling gives a necessary-leakage check (Corollary 1) and a finite-sample audit (Theorem 2) whose declared-budget and structured-feedback variants run at the parametric rate and are the deployable results (Corollaries 2 and 3); the metered variant suits only small alphabets. At training time the same accounting rules out smuggled answers and shows that amortized gains are prepaid (Theorems 3 and 4).

Population quantities are not directly observable in the wild, so we first validate the ledger on a shared-secret family where prior, ghost, and budget are exactly computable: the accounting matches the analytic $m \ln 2$ nat for nat, amortized and per-instance solving separate by $\Theta(n)$, and strong injected side channels are caught at the predicted confidence under accurate accounting. On real code repair (HumanEval+, MBPP+, LiveCodeBench; Liu et al., 2023; Jain et al., 2024), honest arms stay below the loose ceiling and a planted in-band leak is priced highest while it lifts repair success. Three declared-budget real-task protocols yield certificates (planted leaks trigger, honest runs never do, and a 194-token renderer is certified at $\ln 2$ nats against a 782-nat text price), and a residual audit that conditions on legally disclosed feedback certifies, under full first-failure feedback with a model in the loop, an exploit of one leaked test that lifts success by only 0.3 points.

2 Related Work

Adaptive data analysis. Of the three lines of work we compare against, this one is closest in spirit. The reusable holdout (Dwork et al., 2015b), the Ladder mechanism (Blum & Hardt, 2015), and mutual-information bias bounds (Russo & Zou, 2016) control population-level overfitting of adaptive holdout queries, and description-length bounds cap adaptivity by the bits an analysis outputs (Dwork et al., 2015a), a population analogue of our declared budget; lower bounds show that the cost of interactivity is real (Hardt & Ullman, 2014; Steinke & Ullman, 2015), and leaderboard overfitting is measurable in the wild (Roelofs et al., 2019). The ledger instead bounds the per-instance

test-time uplift of an interactive verifier channel against an operational decoy baseline. Mutual-information generalization bounds (Xu & Raginsky, 2017; Steinke & Zakynthinou, 2020) control the training-to-test gap, not test-time uplift, and our conservation theorems are not applications of them.

Feedback-driven repair and verifiers. Learned verifiers rank or supervise candidates (Cobbe et al., 2021; Lightman et al., 2024), whether self-repair genuinely helps is contested (Olausson et al., 2024), and coding agents measurably exploit test access (Zhong et al., 2025). The ledger does not compete on performance; it accounts for the gains.

Contamination and leakage. Contamination and membership-inference audits examine static overlap between training data and evaluation items (Shokri et al., 2017), exchangeability tests prove contamination from model likelihoods (Oren et al., 2024), and reward hacking studies misaligned proxy objectives (Skalse et al., 2022; Amodei et al., 2016). Closest in method are differential-privacy audits, which turn recovered canaries into confidence-calibrated bounds on a privacy parameter (Jagielski et al., 2020; Steinke et al., 2023); we apply the same logic to an information budget on the evaluation channel instead of a privacy budget on the training pipeline.

3 Setup: Channel, Budget, and Ghost Baseline

This section fixes the protocol, its budget, and the baseline for measuring improvement. A pair $(x, y^\star) \sim \mathcal{D}$ draws an input x and a hidden specification $y^\star$, and the policy holds private randomness U independent of $(x, y^\star)$. In round t the policy emits a candidate $z_t = a_t(x, b_{1:t-1}; U)$ and the verifier returns feedback $b_t \sim \pi_t(\cdot \mid x, z_t, y^\star)$ from an alphabet $\mathcal{B}$; after T rounds the policy outputs $\hat{y} = a_{\text{out}}(x, b_{1:T}; U)$, scored by $f(\hat{y}, y^\star, x) \in [0, 1]$; success is $S := \mathbb{E}[f]$. Information is in nats ($\ln 2$ nats per bit). Text feedback is one symbol per round, the whole message under a length cap (end-of-message and padding included), so $\mathcal{B}$ is finite; T is fixed, and unused rounds emit $\perp$.

Definition 1 (Verifier-bit budget)**.** The verifier-bit budget is defined as:

$$B_I := \textstyle\sum_{t=1}^{T} I\big(b_t; y^\star \mid x, U, b_{1:t-1}\big).$$

In words, round t pays only for what b_t adds about $y^\star$ given x, U, and the earlier feedback.

Lemma 1 (Composition identity)**.** $B_I = I(b_{1:T}; y^\star \mid x, U) \geq I(b_{1:T}; y^\star \mid x)$.

Proof sketch. We show that the per-round charges add up to one mutual information. Given $(x, U, b_{1:t-1})$ the candidate z_t is determined, so the chain rule collapses the sum; since $U \perp (x, y^\star)$, expanding $I(b_{1:T}, U; y^\star \mid x)$ in two orders gives the inequality. Full proof in the appendix. □

The identity holds however the candidates adapt to earlier feedback, so adaptivity cannot hide leakage from the meter. Conditioning on U, in practice a random seed, makes each candidate a function of the logged history, so an entropy-based meter never charges the policy's own coin flips; the conditioned budget still upper-bounds the unconditioned leakage. Write $\tau := b_{1:T}$ for the transcript and $P_u(\cdot \mid x, y^\star)$ for its law given $(x, y^\star)$ and seed $U = u$.

Definition 2 (Ghost baseline)**.** For each (x, u) let $Q_u(\tau \mid x) := \mathbb{E}_{y' \sim \mathcal{D}(\cdot|x)}[P_u(\tau \mid x, y')]$ be the transcript law when a decoy specification y', drawn from the prior, drives the verifier. The ghost baseline scores this decoy-driven interaction against the true $y^\star$:

$$q_{\text{ghost}} := \mathbb{E}_{(x,y^\star)}\, \mathbb{E}_u\, \mathbb{E}_{\tau \sim Q_u(\cdot|x)} \big[f(a_{\text{out}}(x, \tau; u), y^\star, x) \big].$$

Operationally, one draws an independent decoy $y' \sim \mathcal{D}(\cdot \mid x)$, runs the full loop with the verifier keyed to y', and scores the output against the true $y^\star$, keeping the generic value of interaction while removing the value that depends on $y^\star$. This is the conditional randomization idea of model-X testing (Candès et al., 2018), applied to the verifier's input.

Remark 1 (What the meter charges). The budget upper-bounds answer-relevant content; it does not split feedback into help and leakage. For a deterministic verifier $H(b_t \mid x, U, b_{1:t-1}, y^\star) = 0$, so $B_I = \sum_t H(b_t \mid x, U, b_{1:t-1})$ and every unpredictable symbol is charged in full. This is deliberate: by split invariance (appendix) a leak can ride in any unpredictable symbol however the feedback is partitioned, so a sound audit must charge them all. The price is conservatism: rich honest feedback is expensive, which caps the audit's power (Section 5) unless the meter is refined (Proposition 3).

4 THE UPLIFT CEILING

With the budget defined, we ask how much improvement over the ghost it can buy.

Theorem 1 (Uplift ceiling). (a) $|S - q_{\text{ghost}}| \le \min\{\sqrt{B_I/2},\ \sqrt{1-e^{-B_I}}\}$. (b) *Per-instance,* $|S - q_{\text{ghost}}| \le \mathbb{E}_{(x,y^\star,u)} \min\{\sqrt{K/2}, \sqrt{1-e^{-K}}\}$ *with* $K := \mathrm{KL}\big(P_u(\cdot \mid x, y^\star) \,\|\, Q_u(\cdot \mid x)\big)$*, and* $(b) \le (a)$.

Proof sketch. We compare the true and ghost runs through a single score. Fix $(x, y^\star, u)$ and set $F(\tau) := f(a_{\text{out}}(x, \tau; u), y^\star, x) \in [0, 1]$. Both transcript laws are scored by the *same* bounded F, so their expected scores differ by at most $\mathrm{TV}(P_u(\cdot \mid x, y^\star), Q_u(\cdot \mid x))$, which Pinsker and Bretagnolle–Huber (Tsybakov, 2009) bound by $\min\{\sqrt{K/2}, \sqrt{1-e^{-K}}\}$. Mutual information is the expected KL divergence between conditional and marginal laws, so $\mathbb{E}[K] = B_I$ by Lemma 1; both bounds are concave, so Jensen gives (a); stopping before it gives (b). Full proof in the appendix. □

Intuitively, feedback can move success away from the ghost only as far as it moves the transcript distribution, and B_I measures that movement on average. Pinsker governs small budgets and Bretagnolle–Huber large ones, crossing at $B_I^\star \approx 1.5936$ nats (appendix figure); the metered audit of Section 5 operates below the crossover, in the Pinsker regime.

Proposition 1 (First-order tightness). *There is a single-round family with* $(S - q_{\text{ghost}})/\sqrt{B_I/2} = 1 - \gamma^2/3 + O(\gamma^4) \to 1$ *as the channel bias* $\gamma \to 0$*; the constant* $1/\sqrt{2}$ *is first-order optimal.*

Put simply, a noisy copy of one hidden bit attains the Pinsker branch to first order as $\gamma \to 0$; Theorem 5 closes the success–budget trade-off, not this ceiling, on one family. At one bit, however, the Pinsker branch reads 0.589 while the noisy-bit family at $\gamma = \frac{1}{2}$ achieves 0.5. Fortunately, declared alphabets close this gap exactly.

Proposition 2 (Cardinality ceiling). *If for every* (x, u) *the transcript takes at most* K *values, e.g.* $K = \prod_t |\mathcal{B}_t|$ *for declared alphabets, then* $|S - q_{\text{ghost}}| \le 1 - 1/K = 1 - e^{-B_{\max}}$ *with* $B_{\max} := \ln K$*. Equality holds for* $y^\star$ *uniform on* K *values with verbatim feedback, and for* $K \ge 2$ *the bound lies strictly below both branches of Theorem 1 at* $B_I = B_{\max}$.

Proof sketch. We bound the expected total variation on the K transcripts, with Q the ghost law. Each $P_{y^\star}(\tau) \in [0, 1]$ has mean $Q(\tau)$ over $y^\star$, so $\mathbb{E}_{y^\star}(P_{y^\star}(\tau) - Q(\tau))_+ \le Q(\tau)(1 - Q(\tau))$; summing gives $\mathbb{E}\,\mathrm{TV} \le 1 - \sum_\tau Q(\tau)^2 \le 1 - 1/K$. Full proof and deployment model in the appendix. □

Concretely, K possible transcripts can at best turn a one-in-K guess into certainty, so a single verdict bit never buys more than one half. The bound concerns alphabets, not B_I: Proposition 1's family has $S - q_{\text{ghost}} \approx \sqrt{B_I/2} \gg 1 - e^{-B_I}$ as $B_I \to 0$, so the metered mode keeps Theorem 1, and tightness at every metered budget is open (Limitations).

Corollary 1 (Necessary leakage). *Any observed uplift forces* $B_I \ge 2\,(S - q_{\text{ghost}})^2$ *nats, and the marginal value of a bit is at most* $\partial_{B_I} \sqrt{B_I/2} = 1/\sqrt{8B_I}$.

Read backwards, a measured gain puts a floor under the leakage behind it: an estimated budget below $2(S - q_{\text{ghost}})^2$ is self-inconsistent and signals unaccounted information.

Proposition 3 (Functional refinement). *Let* $\phi(y^\star) := f(\cdot, y^\star, x)$ *be the grading functional and* $B_\phi := \sum_t I\big(b_t; \phi(y^\star) \mid x, U, b_{1:t-1}\big)$*. Then* $B_\phi \le B_I$*, and Theorem 1 holds with* B_ϕ *for* B_I *against the same ghost baseline; Theorem 2 stays sound as stated, since its estimate upper-bounds* $B_I \ge B_\phi$.

Under B_ϕ a symbol pays only for what it reveals about which answers pass, so grade-irrelevant detail is free and text-heavy feedback has $B_\phi \ll B_I$ (appendix); Corollary 3 makes this a deployable cap.

On novelty. Pinsker, Bretagnolle–Huber, and Fano are standard (Cover & Thomas, 1999; Tsybakov, 2009), as is bounding an expected gap by $\sqrt{I/2}$ in information-ratio analyses of Thompson sampling (Russo & Van Roy, 2016). The contribution is the object and the loop around it: a per-instance uplift certificate against an executable intervention, a ceiling exactly tight for declared

feedback, an exactly priced protocol, and a $\Theta(n)$ amortization separation. The composition theorem of differential privacy is also elementary, yet it defined an auditable field.

5 An Executable Audit

A ceiling is useful only if it can be checked from samples, and this section turns it into a test. Let $\widehat{S}$ and $\widehat{q}_{\mathrm{ghost}}$ be the empirical pass rates over n_e true and n_g decoy episodes; decoys drawn from a proxy prior $\tilde{\mathcal{D}}$ carry the bias $\Delta_p := \mathbb{E}_x \,\mathrm{TV}(\tilde{\mathcal{D}}(\cdot \mid x), \mathcal{D}(\cdot \mid x))$, zero when the prior is known. Let $\widehat{B}$ be the empirical cross-entropy, over n_b evaluation episodes, of a feedback predictor that reads only the logged history $(x, U, b_{1:t-1})$ and is fit on a separate sample and frozen before these episodes are drawn, so the three estimation samples are mutually independent; richer predictor inputs or evaluation on the fitting sample could understate the cross-entropy and void the guarantee (appendix). Each episode's loss must lie in a known range $[0, L]$: mixing each per-symbol distribution with the uniform law at weight $\lambda \in (0, 1)$ guarantees $L = T \ln(|\mathcal{B}|/\lambda)$, and an exactly known predictor of a white-box channel may certify a smaller almost-sure range.

Remark 2 (Deployment model and the null). The auditor, like a leaderboard or evaluation operator, owns the harness and its logs; the adversary is the policy or a compromised in-harness component. Let the deployed policy also read an unlogged stream $c_t \sim \kappa_t(\cdot \mid x, z_t, y^\star)$, so that $\hat{y} = a_{\mathrm{out}}(x, b_{1:T}, c_{1:T}; U)$. The null $\mathbf{H_0}$ says the log is complete: $I(c_{1:T}; y^\star \mid x, U, b_{1:T}) = 0$. Ghost episodes are interventions: the auditor re-keys the specification slot, so every specification-reading component, the verifier and any unlogged κ_t alike, runs on the decoy. Under $\mathbf{H_0}$ the interaction carries exactly the logged budget and Theorem 1 applies to the deployment (appendix), so rejection falsifies the transcript-only account. A path that acquires $y^\star$ outside the intervened slot (a cached artifact, a contaminated tool) instead inflates the ghost arm, visible as $\widehat{q}_{\mathrm{ghost}}$ far above the prior, but uncertified.

Theorem 2 (Audit test)**.** *Let $\mathbf{H_0}$ be the transcript-only hypothesis of Remark 2, let the predictor be fit independently of the n_b evaluation episodes and frozen, and define the statistic and threshold:*

$$G := \widehat{S} - \widehat{q}_{\mathrm{ghost}} - \sqrt{\widehat{B}/2}, \qquad \varepsilon(\delta) := \sqrt{\tfrac{\ln(3/\delta)}{2n_e}} + \sqrt{\tfrac{\ln(3/\delta)}{2n_g}} + \Delta_p + \tfrac{1}{\sqrt{2}}\, L^{1/2} \Big(\tfrac{\ln(3/\delta)}{2n_b}\Big)^{1/4}.$$

Then $\Pr_{\mathbf{H_0}}(G > \varepsilon(\delta)) \le \delta$. Moreover, with $c(\beta) := \min\{\sqrt{\beta/2}, \sqrt{1 - e^{-\beta}}\}$ and $d := L\sqrt{\ln(3/\delta)/2n_b}$, the statistic $G' := \widehat{S} - \widehat{q}_{\mathrm{ghost}} - c(\widehat{B} + d)$ satisfies $\Pr_{\mathbf{H_0}}(G' > \varepsilon'(\delta)) \le \delta$ with $\varepsilon'(\delta) := \varepsilon(\delta) - \sqrt{d/2}$, and $G' > \varepsilon'(\delta)$ whenever $G > \varepsilon(\delta)$. Either rejection certifies, at confidence $1 - \delta$, that the log is incomplete: some $y^\star$-dependent path bypasses the accounted transcript.

Proof sketch. We show that, outside an event of probability δ, no estimate errs against $\mathbf{H_0}$ by more than its share of the threshold. Split δ across three Hoeffding events: $\widehat{S}$ does not overstate S, $\widehat{q}_{\mathrm{ghost}}$ does not understate q_{ghost} beyond Δ_p, and $\widehat{B}$ does not understate the true cross-entropy CE beyond d. Gibbs' inequality gives $\mathrm{CE} \ge \sum_t H(b_t \mid \cdot) \ge B_I$, and only this direction is needed. For G, $\sqrt{a - d} \ge \sqrt{a} - \sqrt{d}$ turns the budget event into $\sqrt{\widehat{B}/2} \ge \sqrt{B_I/2} - \sqrt{d/2}$, and Theorem $1(a)$ under $\mathbf{H_0}$ keeps G below $\varepsilon(\delta)$. For G', the same event gives $B_I \le \widehat{B} + d$; as c is increasing and $c(\widehat{B} + d) \le \sqrt{\widehat{B}/2} + \sqrt{d/2}$, G' rejects whenever G does. Full proof in the appendix. □

In effect, G subtracts from the observed uplift the most the logged feedback could explain, and $\varepsilon(\delta)$ covers four errors: (i) in $\widehat{S}$, shrinking with n_e; (ii) in $\widehat{q}_{\mathrm{ghost}}$, shrinking with n_g; (iii) the decoy bias Δ_p, set by the decoys' fidelity to the prior; and (iv) in the budget, set by n_b and the loss range L.

Remark 3 (One-sidedness and noise direction). The test is one-sided: $G \le \varepsilon$ is consistent with $\mathbf{H_0}$ but does not imply it, so a non-trigger clears nothing. Exogenous noise independent of $y^\star$ only raises the cross-entropy, which inflates $\sqrt{\widehat{B}/2}$ and lowers G; it can cost power but never causes a false alarm. Logging the seed U sharpens the meter but is optional: by data processing the logged pair $(\tau, \hat{y})$ is charged at most B_I (appendix), so seedless closed deployments remain auditable.

Corollary 2 (Declared-budget audit)**.** *Suppose the protocol caps the budget by construction, fixing the transcript alphabet so that $B_{\max} = \sum_t \ln |\mathcal{B}_t|$ can be read off the log format. With $G_{\mathrm{d}} :=$*

$\widehat{S}-\widehat{q}_{\mathrm{ghost}}-(1-e^{-B_{\max}})$ *and* $\varepsilon_{\mathrm{d}}(\delta) := \sqrt{\ln(2/\delta)/2n_e}+\sqrt{\ln(2/\delta)/2n_g}+\Delta_p$*, we have* $\Pr_{\mathbf{H_0}}(G_{\mathrm{d}} > \varepsilon_{\mathrm{d}}(\delta)) \le \delta$*. The cross-entropy probe and its quartic cost vanish, and the audit runs at the parametric rate (boxed pseudocode in the technical appendix). A protocol that caps* $B_I \le B_{\max}$ *without fixing the alphabet uses* $c(B_{\max})$ *instead.*

With $B_{\max} = \ln 2$, observed uplift above $0.5 + \varepsilon_{\mathrm{d}}$ thus certifies another path at confidence $1 - \delta$.

Corollary 3 (Structured-feedback audit)**.** *Suppose each round's feedback is rendered as* $b_t = \psi_t(x, z_t, g_t, W_t)$ *with grading summary* $g_t = \omega_t(x, z_t, \phi(y^\star)) \in \mathcal{G}_t$ *and render randomness* W_t *independent of* $y^\star$ *given* $(x, U, b_{1:t-1})$*; deterministic renderers qualify. Then* $B_\phi \le \sum_t \ln|\mathcal{G}_t| =: B^\phi_{\max}$*, a cap verifiable from the verifier's code, and Corollary 2 runs with* $B^\phi_{\max}$ *(Proposition 2 with* $K = \prod_t |\mathcal{G}_t|$*) against the grading-channel null* $\mathbf{H}^\phi_{\mathbf{0}}$*: given the grading summaries and render randomness, unlogged inputs carry no information about* $y^\star$ *(appendix). Rejection certifies a* $y^\star$*-dependent path outside the declared grading channel.*

What counts is the number of grading outcomes the renderer can express, and message length is irrelevant: a verbose renderer with a one-bit verdict has $B^\phi_{\max} = T \ln 2$. Such structured channels are the norm in automated evaluation (verdict enums, CI pass/fail, k-point rubrics), and each caps the budget at the log of its alphabet size.

The operating envelope. Two constraints limit when the audit can fire. The first is sampling, dominated in metered mode by the budget probe: at $n_e = n_g = 2000$, $n_b = 500$, $T = 8$, $|\mathcal{B}| = 16$, and $\lambda = 1/|\mathcal{B}|$, each sampling term is $\approx$0.03 but $\sqrt{d/2} \approx 1.19$, so $\varepsilon(\delta) \approx 1.26 > 1$ and G can never exceed it. The sharper G' cuts the n_b needed for a budget radius of 0.05 at $B_I = \ln 2$ from 1.6×10^8 to 2.7×10^5 (null side, exact predictor), but L grows with message length times log-vocabulary, so metered mode suits small alphabets, not text. The second constraint outlasts sampling: scores live in $[0, 1]$, so the power guarantee (an excess $\Delta^* \ge 2\varepsilon$ is caught; appendix) needs the accounted ceiling below $1 - q_{\mathrm{ghost}} - 2\varepsilon$, which for declared alphabets reads:

$$B_{\max} \;\le\; -\ln\big(q_{\mathrm{ghost}} + 2\varepsilon_{\mathrm{d}}\big),$$

and for metered budgets reads $B_I \le 2(1 - q_{\mathrm{ghost}} - 2\varepsilon)^2_+$. At $B_{\max} = \ln 2$ an observed uplift must exceed $1 - e^{-B_{\max}} + \varepsilon_{\mathrm{d}} \approx 0.56$ to trigger, and power $1 - \delta$ is guaranteed from a true uplift of $0.5 + 2\varepsilon_{\mathrm{d}} \approx 0.62$, so only strong leaks earn certificates (the planted leaks below reach 0.80–0.88). No protocol-agnostic test can lower this floor, since Proposition 2 is attained and an honest one-bit protocol can legitimately buy an uplift of 0.5; a white-box protocol can use its exact ceiling instead (0.200 for the leaderboard below; appendix). Inside the envelope, at $q_{\mathrm{ghost}} = 0.05$ a declared alphabet may carry about 2.5 bits ($B_{\max} \le 1.76$ nats), the reusable-holdout setting (Dwork et al., 2015b; Blum & Hardt, 2015) with a certificate attached; outside it, free text costs tens of nats ($\sim$60 in our code-repair arms, against a metered envelope of $\sim$0.06) and is priced rather than tested. We call these regimes *certificate mode* (inside the envelope, with a known, declared, or structurally capped budget) and *pricing mode* (outside it; every output descriptive); Corollary 3 brings verbose feedback inside.

6 Conservation at Training Time

So far the ledger has metered one evaluation; training also consumes feedback, and bits paid in training can be spent at test time. Let a student $W = \mathrm{Alg}(\{(x_i, \tau_i)\}_{i=1}^n, U_{\mathrm{tr}})$ be trained on i.i.d. instances with independent interactions, each with $I(\tau_i; y_i^\star \mid x_i) \le \beta$.

Theorem 3 (Memory audit)**.** *Re-solving a training instance without further feedback,* $\hat{y}_i = W(x_i)$*, obeys* $\Pr(\hat{y}_i \text{ correct}) \le \mathbb{E}_x[\max_y \mathcal{D}(y \mid x)] + \sqrt{\beta/2}$.

In plain terms, a student trained under the cap re-solves a training instance at most $\sqrt{\beta/2}$ above the best prior guess, so a higher hidden-split re-solve rate is direct evidence of distilled answers.

Theorem 4 (Amortization conservation)**.** *Let* $y_i^\star = g(\sigma, x_i)$ *for a shared secret* σ *with* $(\{x_i\}, U_{\mathrm{tr}}) \perp \sigma$*; let interactions be conditionally independent given* $(\sigma, \{x_i\})$ *with fresh per-interaction randomness, and let the verifier access* σ *only through* $y_i^\star$*. If a fresh instance* $x_{\mathrm{new}} \perp (W, \sigma)$ *pays present budget* $\beta_f := I(\tau_{\mathrm{new}}; y^\star_{\mathrm{new}} \mid x_{\mathrm{new}}, W)$*, and* $q^{\mathrm{new}}_{\mathrm{ghost}}$ *uses decoys drawn independently of* (W, σ)*, then* $S_{\mathrm{new}} - q^{\mathrm{new}}_{\mathrm{ghost}} \le \sqrt{(I(W;\sigma) + \beta_f)/2}$ *with* $I(W;\sigma) \le n\beta$.

New-instance gains split into prepaid and fresh test-time bits; amortization moves information across instances but creates none, and additivity also pools audits across deployment windows (appendix).

Operational reading. The theorems give a training-time checklist that never opens the model: fix a per-instance cap β; log seeds, transcripts, and the decoy pool; flag re-solves of held-out training instances more than $\sqrt{\beta/2}$ above the prior and fresh-instance uplift beyond $\sqrt{(n\beta + \beta_f)/2}$. Both theorems extend to adaptive, online training, charging each logged symbol at most the log of its declared alphabet size (appendix): verdict-only reward gives $\beta = n_i \ln 2$ after n_i visits and caps re-solve success at 2^{n_i} times the largest prior mass, which binds over many visits when answers are diverse; rich test-output feedback is priced but not bound.

7 The Shared-Secret Family

This section builds a testbed where every population quantity is known exactly.

Definition 3 (Shared-secret family)**.** $\sigma \sim \mathrm{Unif}(\{0,1\}^m)$; instance $x_i = i$ with $y_i^\star = \sigma$; hidden test $f = \mathbf{1}\{\hat{y} = \sigma\}$; feedback $b = (j, \sigma_j)$ for the least mismatched bit j (and $\perp$ when all match); prior 2^{-m}.

Theorem 5 (Matching-constant closure)**.** (a) *a single-instance budget* $B_I \le \beta$ *forces* $\Pr(\hat{y} = \sigma) \le \min\{1, (\beta + \ln 2)/(m \ln 2)\}$*;* (b) *a deterministic sequential protocol succeeds with probability* 1 *and pays exactly* $B_I = m \ln 2$*. The lower and upper bounds match to the constant.*

Theorem 6 (Amortization separation)**.** *Mean success* $3/4$ *over* n *instances with isolated solvers costs* $\Theta(nm)$ *nats in total, whereas one amortized solver reaches success* 1 *at* $m \ln 2 = \Theta(m)$*; the gap grows as* $\Theta(n)$*.*

Recovering the secret thus costs about $m \ln 2$ nats however the policy adapts, and sequential search pays exactly that; isolated solvers pay this price on every instance, an amortized solver only once.

Proposition 4 (Calibration instrument)**.** *On the shared-secret family prior, ghost, and per-round* B_I *are analytic, so the three population quantities of Theorem 2 carry zero systematic error: the audit is exactly calibrated, and any side channel whose excess uplift over the accounted ceiling exceeds* $2\varepsilon(\delta)$ *is detected with probability at least* $1 - \delta$*. The construction transfers to any family with* $y_i^\star = g(\sigma, x_i)$*, g known, and* $H(y_i^\star \mid x_i) \ge h_0 > 0$ *(appendix).*

With zero systematic error, a trigger points to the injected channel, up to a false-alarm rate of δ. The family is a calibration instrument, not a benchmark: its protocol is an adaptive noiseless Rényi–Ulam search (Pelc, 2002), and what is new is its exact ledger entry.

8 Experiments

This section checks measured quantities against the theorems and states the limits of each check (shared-secret family on CPU with ten master seeds; code repair on open-weight models; code, seeds, and assertion logs in the supplementary material). In short, the invariants are supported, the guess that "more feedback always helps" fails, free-text alarms do not fire, and declared-budget certificates do.

Nat-by-nat accounting. For $m \in \{8, 16, 32, 64\}$ the measured budget matches the analytic $m \ln 2$ to relative error below 4×10^{-4}, with exact recovery (Figure 2; appendix table); a literal full-history estimator fails at large m, where no finite sample covers the histories.

Controlling false positives, detecting strong leaks. Under the null, empirical false-positive rates stay within nominal levels ($0/2000$ at $\delta = 0.05$ with accurate accounting; Wilson upper bound 0.19%). Power, however, depends on the accounting. With a loose cross-entropy probe at $n_b = 500$ the term $\sqrt{d/2}$ swamps the signal and power is zero on every channel, the quartic bottleneck noted above. With accurate accounting (the exact predictor of a parsed deterministic channel; per-episode loss in $[0, \ln 2]$, budget still estimated from $n_b = 5 \times 10^4$ episodes, $\varepsilon \approx 0.10$), a strong timing channel is caught with a sharp transition: detection is complete by strength 0.8, with $500/500$ triggers at strength 1 (Wilson lower bound 0.9924; Figure 2). Five split-calibrated heuristics are

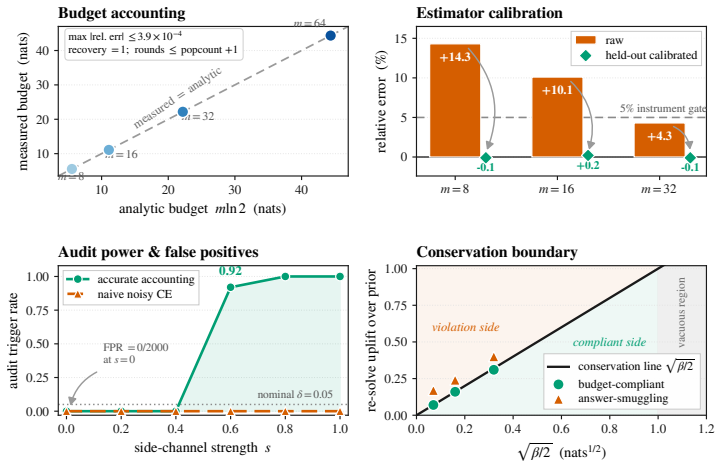

Figure 2: Metrology on the exactly solvable family. *Accounting:* measured budget against the analytic $m \ln 2$. *Calibration:* the raw paired estimator exceeds the 5% gate at small m; a calibration frozen on one seed brings every anchor within 0.65% on three held-out seeds (shown: within 0.21%). *Audit power:* accurate accounting (exact predictor, sampled budget) reaches full power past a strength threshold, while a naive noisy probe stays at zero. *Conservation:* compliant students stay under the conservation line, answer-smugglers cross it, and the bound is vacuous for $\beta \geq 2$.

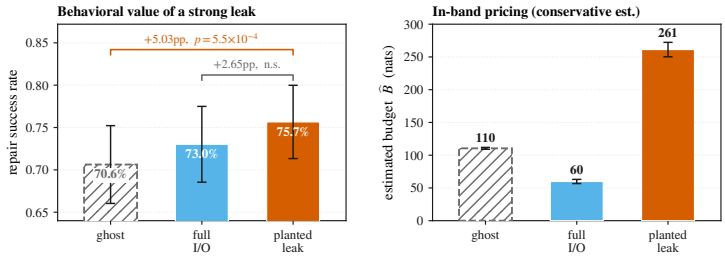

Figure 3: Free-text pricing mode (Qwen2.5-Coder-7B, MBPP+, deterministic, four rounds). *Left:* repair success by arm; the planted in-band leak beats the ghost, but its gap over plain rich feedback is not significant. *Right:* the leak is priced highest, and honest arms stay under the (loose) ceiling; the estimator upper-bounds text entropy, so the ghost's apparent budget is not $y^\star$-leakage.

more powerful at moderate strengths, but each is told which side field to inspect, is brittle on some channel, and needs per-deployment null calibration; the audit needs neither (appendix).

Landing on real code repair. We carry the ledger to HumanEval+, MBPP+, and LiveCodeBench, with base tests as the visible verifier and extended hidden tests as the score (Chen et al., 2021; Liu et al., 2023; Jain et al., 2024), using Qwen2.5-Coder-7B and Llama-3.1-8B (Hui et al., 2024; Grattafiori et al., 2024). Randomized hidden specifications support certificates (below), a declared decoy family supports pricing, and a bare fixed benchmark identifies nothing. On a fixed benchmark the task determines the hidden suite and the population ledger degenerates (appendix), so we declare a decoy family as the reference measure (reference outputs mutated by consistent but wrong, length-matched operators); every statistic below is thus in pricing mode, Δ_p is unidentified, and the ghost level moves within 1.85 points across constructions (appendix). Two findings hold and one guess fails. First, no arm exceeds its ceiling (two models, three datasets, all arms), but at tens of nats the ceiling saturates, which shows consistency without power. Second, and perhaps surprisingly, feedback level does not buy success monotonically, and the ghost is not uniformly the lowest arm (appendix): decoy feedback still supplies scaffolding, and the ledger prices only the $y^\star$-specific increment. Third, a strong in-band leak that plants the true hidden failing example lifts repair success by $+5.03$ points over the ghost (exact McNemar $p = 5.5 \times 10^{-4}$) and is priced as by far the most answer-relevant arm (Figure 3). Since the conservative estimator upper-bounds text entropy, this

ordering may partly reflect length and vocabulary; a paired diagnostic that cancels the text floor (appendix; an earlier, stateless run of the same leak) prices the leak near 22 nats against a near-zero baseline. Being in-band, the leak is priced rather than flagged by an out-of-band alarm.

Certificates on controlled real-task protocols. Three protocols take the declared-budget audit to real tasks, each with $B_{\max} = \ln 2$ (ceiling 0.5), paired arms of 2000 episodes, ten seeds, and $\varepsilon_{\mathrm{d}} = 0.061$ (appendix). In each, planted content reaches the model through a path absent from the audited log, which records only the declared verdicts; a format check of the model's context would expose these plants, so they are positive controls, not realistic threats. A leaderboard on real MBPP+ hidden tests returns one pass/fail bit per submission, with controlled exact-failure programs as candidates and no model in the loop: honest play stays far below threshold ($G_{\mathrm{d}} = -0.314$), and an injected channel revealing the hidden suite triggers ($G_{\mathrm{d}} = 0.305$). With Qwen2.5-Coder-7B under a randomized hidden-grading extension and verdict-only feedback, the meter works both ways: the honest verdict buys a real paired improvement ($\log_{10} p \approx -21.7$) yet stays inside its budget ($G_{\mathrm{d}} = -0.458$), while an undeclared copyable convention leak triggers ($G_{\mathrm{d}} = 0.351$). Corollary 3 also runs live: a white-box renderer emits $\sim$194-token feedback whose only specification-dependent input is the verdict bit (structural budget $\ln 2$, generic contextual text price 782 nats); the compliant renderer never triggers, and a tampered one does ($G_{\mathrm{d}} = 0.380$). Honest arms never trigger, null resamplings trigger $0/1000$ per protocol, and verdicts survive the looser Pinsker threshold (appendix).

A residual audit with a model in the loop. The residual audit (Appendix G) conditions on what legal feedback has already disclosed, so it tolerates full feedback. On the 107 MBPP+ tasks whose initial Qwen2.5-Coder-7B program fails at least one of 24 real plus-tests, each episode draws 12 of them as the hidden suite (321 episodes) and returns full first-failure feedback (input, expected and actual output or error) for two rounds; a pre-specified e-value ($\lambda = 0.5$) scores the final program and multiplies across episodes. Honest repair raises no alarm (mean e-value 0.98, log product -25.5), and neither does a policy that special-cases every legally disclosed test, which an unconditioned test flags (log product $+33.4$). A policy that wraps the model's final program to special-case one leaked, undisclosed hidden test is certified in every episode ordering after a median of 35 episodes (log product $+24.4$), although it lifts hidden-suite success by only 0.3 points, far below the declared audit's floor; leaking in half the episodes still certifies in two thirds of orderings. Handed the leaked test, the model itself did not use it, and the audit correctly stayed silent (appendix).

9 Limitations

There are, of course, limitations. The audit is one-sided (it certifies leaks, never their absence), and the default meter is conservative, upper-bounding leakage by feedback entropy (Remark 1); Proposition 3 and Corollary 3 narrow it, but a certified B_ϕ estimator for black-box feedback is missing. Free-text channels lie outside the ledger audit's envelope and are only priced, relative to a declared decoy family on fixed benchmarks. The certificates use controlled protocols (exact-failure candidates; a hidden-grading extension whose success measures convention compliance, not repair) and planted leaks only, so realistic threats such as agents exploiting test access (Zhong et al., 2025) remain untested. The residual audit needs randomized suites and a known verifier mechanism, and its exploit was scripted around real model programs. The guarantees assume independent evaluation episodes and inherit decoy bias through Δ_p; the ghost intervention must reach every specification-reading component, and off-transcript information (timing, caching, tool effects) must be folded into the encoding. The confirmatory repair comparison covers one 7B model and one dataset; the structured cap trusts a white-box renderer, and obfuscation defeats its static check though not the behavioral audit (appendix); and tightness at intermediate metered budgets remains open.

10 Conclusion

Verifier feedback is an information channel, and without a meter its gains cannot be attributed. The ledger supplies one: a ceiling exactly tight for declared feedback, an audit at the parametric rate under declared or structurally capped budgets, and conservation laws that survive adaptive training. Verification information can be budgeted, audited, and priced like computation.

AI USE STATEMENT

We used large language model (LLM) assistants in three ways. For preliminary research, they helped search and summarize related literature while we scoped the problem. For experiments, they helped write, run, and debug experiment and analysis code. For writing, they helped draft an early version of the manuscript and edit its language. LLMs did not generate experimental data or results: every reported number comes from logged runs of the released code. We reviewed all AI-assisted work, verified every mathematical claim and every reported number ourselves, and take full responsibility for the content of this paper.

ETHICS STATEMENT

This leakage audit is defensive: it certifies leaks, never their absence. The work involves no human subjects, and all data and models are public.

REPRODUCIBILITY STATEMENT

Every theorem is restated with a full proof in Appendices B–D, and the standing assumptions and audit preconditions are collected in Appendix B. Protocols, pilot decisions, and disclosures for every experiment are in Appendices E and F, and the supplementary material contains the code, run configurations, seeds, and assertion logs behind every reported number.

REFERENCES

Dario Amodei, Chris Olah, Jacob Steinhardt, Paul Christiano, John Schulman, and Dan Mané. Concrete problems in AI safety. *arXiv preprint arXiv:1606.06565*, 2016.

Avrim Blum and Moritz Hardt. The ladder: A reliable leaderboard for machine learning competitions. In *International Conference on Machine Learning*, pp. 1006–1014. PMLR, 2015.

Emmanuel Candès, Yingying Fan, Lucas Janson, and Jinchi Lv. Panning for gold: 'model-X' knockoffs for high dimensional controlled variable selection. *Journal of the Royal Statistical Society Series B: Statistical Methodology*, 80(3):551–577, 2018.

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. *arXiv preprint arXiv:2107.03374*, 2021.

Xinyun Chen, Maxwell Lin, Nathanael Schärli, and Denny Zhou. Teaching large language models to self-debug. In *International Conference on Learning Representations*, pp. 8746–8825, 2024.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, et al. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*, 2021.

Thomas M Cover and Joy A Thomas. *Elements of information theory*. John Wiley & Sons, 1999.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. Generalization in adaptive data analysis and holdout reuse. In *Advances in Neural Information Processing Systems*, volume 28, pp. 2350–2358, 2015a.

Cynthia Dwork, Vitaly Feldman, Moritz Hardt, Toniann Pitassi, Omer Reingold, and Aaron Roth. The reusable holdout: Preserving validity in adaptive data analysis. *Science*, 349(6248):636–638, 2015b.

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The Llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Moritz Hardt and Jonathan Ullman. Preventing false discovery in interactive data analysis is hard. In *2014 IEEE 55th Annual Symposium on Foundations of Computer Science*, pp. 454–463. IEEE, 2014.

Binyuan Hui, Jian Yang, Zeyu Cui, Jiaxi Yang, Dayiheng Liu, Lei Zhang, Tianyu Liu, Jiajun Zhang, Bowen Yu, Keming Lu, et al. Qwen2.5-Coder technical report. *arXiv preprint arXiv:2409.12186*, 2024.

Matthew Jagielski, Jonathan Ullman, and Alina Oprea. Auditing differentially private machine learning: How private is private SGD? *Advances in Neural Information Processing Systems*, 33: 22205–22216, 2020.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. *arXiv preprint arXiv:2403.07974*, 2024.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *International Conference on Learning Representations*, pp. 39578–39601, 2024.

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. *Advances in neural information processing systems*, 36:21558–21572, 2023.

Aman Madaan, Niket Tandon, Prakhar Gupta, Skyler Hallinan, Luyu Gao, Sarah Wiegreffe, Uri Alon, Nouha Dziri, Shrimai Prabhumoye, Yiming Yang, et al. Self-refine: Iterative refinement with self-feedback. *Advances in neural information processing systems*, 36:46534–46594, 2023.

Theo X Olausson, Jeevana Priya Inala, Chenglong Wang, Jianfeng Gao, and Armando Solar-Lezama. Is self-repair a silver bullet for code generation? In *International Conference on Learning Representations*, pp. 36545–36593, 2024.

Yonatan Oren, Nicole Meister, Niladri S. Chatterji, Faisal Ladhak, and Tatsunori B. Hashimoto. Proving test set contamination in black-box language models. In *International Conference on Learning Representations*, 2024.

Andrzej Pelc. Searching games with errors—fifty years of coping with liars. *Theoretical Computer Science*, 270(1-2):71–109, 2002.

Rebecca Roelofs, Vaishaal Shankar, Benjamin Recht, Sara Fridovich-Keil, Moritz Hardt, John Miller, and Ludwig Schmidt. A meta-analysis of overfitting in machine learning. *Advances in Neural Information Processing Systems*, 32, 2019.

Daniel Russo and Benjamin Van Roy. An information-theoretic analysis of Thompson sampling. *Journal of Machine Learning Research*, 17(68):1–30, 2016.

Daniel Russo and James Zou. Controlling bias in adaptive data analysis using information theory. In *Artificial Intelligence and Statistics*, pp. 1232–1240. PMLR, 2016.

Shalev Shaer, Gal Maman, and Yaniv Romano. Model-X sequential testing for conditional independence via testing by betting. In *International Conference on Artificial Intelligence and Statistics*, volume 206, pp. 2054–2086. PMLR, 2023.

Noah Shinn, Federico Cassano, Ashwin Gopinath, Karthik Narasimhan, and Shunyu Yao. Reflexion: Language agents with verbal reinforcement learning. *Advances in neural information processing systems*, 36:8634–8652, 2023.

Reza Shokri, Marco Stronati, Congzheng Song, and Vitaly Shmatikov. Membership inference attacks against machine learning models. In *2017 IEEE symposium on security and privacy (SP)*, pp. 3–18. IEEE, 2017.

Joar Skalse, Nikolaus Howe, Dmitrii Krasheninnikov, and David Krueger. Defining and characterizing reward gaming. *Advances in Neural Information Processing Systems*, 35:9460–9471, 2022.

Armando Solar-Lezama, Liviu Tancau, Rastislav Bodik, Sanjit Seshia, and Vijay Saraswat. Combinatorial sketching for finite programs. In *Proceedings of the 12th international conference on Architectural support for programming languages and operating systems*, pp. 404–415, 2006.

Thomas Steinke and Jonathan Ullman. Interactive fingerprinting codes and the hardness of preventing false discovery. In *Conference on Learning Theory*, pp. 1588–1628. PMLR, 2015.

Thomas Steinke and Lydia Zakynthinou. Reasoning about generalization via conditional mutual information. In *Conference on Learning Theory*, pp. 3437–3452. PMLR, 2020.

Thomas Steinke, Milad Nasr, and Matthew Jagielski. Privacy auditing with one (1) training run. *Advances in Neural Information Processing Systems*, 36:49268–49280, 2023.

Michael Carl Tschantz, Amit Datta, Anupam Datta, and Jeannette M. Wing. A methodology for information flow experiments. In *IEEE 28th Computer Security Foundations Symposium*, 2015.

Alexandre B. Tsybakov. *Introduction to Nonparametric Estimation*. Springer, New York, 2009.

Vladimir Vovk and Ruodu Wang. E-values: Calibration, combination and applications. *The Annals of Statistics*, 49(3):1736–1754, 2021.

Aolin Xu and Maxim Raginsky. Information-theoretic analysis of generalization capability of learning algorithms. *Advances in neural information processing systems*, 30, 2017.

Ziqian Zhong, Aditi Raghunathan, and Nicholas Carlini. ImpossibleBench: Measuring LLMs' propensity of exploiting test cases. *arXiv preprint arXiv:2510.20270*, 2025.