LLM BENCHMARK MEANS CAN BE BLIND TO THE TAIL

Anonymous authors
Paper under double-blind review

ABSTRACT

LLM benchmark failure rates are often read as properties of model weights, but they are also properties of the decoder used to sample outputs. We show that this matters most in the tail. On independently selected SecurityEval prompts where `bandit`-detected weaknesses are rare under an untruncated $T = 1.0$ baseline, lowering the temperature to 0.7 reduces prompt-weighted medium-severity rates by roughly one third to one half across three models, with rate ratios between 0.51 and 0.67. On the full benchmark, the corresponding ratio for Qwen2.5-1.5B is 1.02, with a 95% paired-prompt bootstrap interval of $[0.95, 1.09]$. All code-weakness rates are conditional on parseable output, and parseability is reported separately. This contrast motivates treating the decoder as part of the estimand. We study when samples collected under one decoder can estimate a rate under another. For configurations that share the same model logits, a teacher-forced pass over stored completions gives exact likelihood ratios for target decoders, without additional generation. The transfer is valid only when the audit support contains the target's event-relevant support. When it does, we use a per-prefix Rényi-2 quantity computed from the stored logits as an empirical predictor of transfer cost, and show that prefix weighting can improve efficiency for prefix-measurable events. The experiments also show a limit of reuse: measured effective sample size and other sample-based diagnostics can miss poorly sampled mass, and do not by themselves support a family-wide certificate. We recommend reporting the decoding configuration with every failure rate, and checking support, statistical cost, and parseability before transferring conclusions across decoders.

1 INTRODUCTION

A refusal rate, a harmful-completion rate or a vulnerable-code rate is obtained by sampling from a model under some decoding configuration. Evaluation harnesses and safety benchmarks commonly decode greedily (Gao et al., 2021; Mazeika et al., 2024), while serving stacks and the generation configurations shipped with models support, and often specify, stochastic decoding (Kwon et al., 2023). The measured quantity is therefore a property of the pair (weights, decoder), and the benchmarked pair can differ from the served one. Benchmark averages change little across decoders, which makes the gap easy to dismiss; a flat average can still hide a large effect on the prompts where failures are rare.

Existing work studies the two sides of this gap separately. Evaluations of decoding find that refusal degrades under high-temperature sampling (Howard et al., 2026), that diverse sampling exposes long-tail failures (Hajra et al., 2026), and that decoding interacts with instruction tuning (Grover, 2026). Code-security evaluations such as Pearce et al. (2025) vary prompts and vulnerability classes rather than the decoder. These studies measure each configuration directly. Rare-event methods estimate small failure probabilities under one fixed sampler, using trained importance proposals (Angell et al., 2026), subset simulation (Wang et al., 2026) or multi-level splitting (Parulekar & Liu, 2026). We address two questions that neither line takes up: how the benchmark average relates to the rate on the prompts where failures are rare, and when a measurement under one decoder can be reused for another.

To measure the gap, we select, on an independent untruncated sample, the SecurityEval prompts on which a model rarely emits a `bandit`-detected weakness (PyCQA), and compare untruncated

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.

sampling at $T{=}1.0$ with $T{=}0.7$ on fresh draws. The prompt-weighted rate falls by a third to a half at medium severity across three models, and by more in the one high-severity cell (Qwen2.5-7B), while over all 121 prompts the ratio stays close to one (Table 1 and Figure 1). The analyser's output defines the event, so no detector error has to be calibrated across decoders. We call these subsets low-rate prompts (Section 6.1).

Greedy evaluation cannot substitute for sampling here. It visits one trajectory per prompt, so no reweighting of its output recovers a stochastic deployment's rate (Proposition 2). It also observes too few events at these rates, and its risk does not grow with repeated attempts (Section 6.6).

For the second question the tools are classical. A decoding configuration is a deterministic map from logits to a probability vector, so the likelihood ratio between two configurations is a pointwise function of one shared logit vector, and change of measure, absolute continuity and effective sample size apply unchanged. Three steps are specific to decoders. One teacher-forced pass over a stored audit sample scores every target configuration without further generation. Support inclusion decides whether a transfer is identified. When it is, a per-token Rényi-2 divergence (Van Erven & Harremos, 2014) from the same logits predicts the statistical cost.

Contributions.

1. **A rare-event/aggregate contrast on a real safety functional** (Section 6.1). On low-rate prompts the decoder effect is 1.5–$2\times$ at medium severity across three models and 4–$5\times$ for Qwen2.5-7B at high severity, while the aggregate ratios on the same benchmark and detector stay close to one.
2. **Exact cross-decoder transfer from shared logits** (Sections 3 and 4), identified under the event-specific support condition of Proposition 1, with support nesting sufficient for every event, and validated against independent direct sampling on 30 of 32 testable cells (Section 6.2).
3. **An empirical transfer-cost predictor** (Sections 5 and 6.3 to 6.5). A plug-in Rényi-2 quantity from stored audit logits predicts the effective sample size on 122 resolvable cells in two model families and, in a retrospective 32-cell comparison, screens transfers better than the measured ESS. For prefix-decided events, prefix weighting turns single-digit effective sample sizes into hundreds.
4. **Two negative results** (Section 7). The sample-based diagnostics we tested do not reveal missed mass, so we claim no family-wide certificate. In the tested sweeps, large changes in the aggregate rate occur only together with large losses of parseable output.

2 Setup and notation

Let a model produce logits $z(x, y_{<t}) \in \mathbb{R}^V$ for prompt x and prefix $y_{<t}$. A *decoding configuration* c is a deterministic map $f_c(z_t, y_{<t}) \in \Delta^{V-1}$ from the current logits $z_t = z(x, y_{<t})$ and, where needed, the realised prefix, and for a completion $y = (y_1, \dots, y_L)$, $q_c(y \mid x) = \prod_{t=1}^{L} f_c(z_t, y_{<t})[y_t]$. For temperature, top-$k$ and top-p the map depends on the prefix only through z_t, and we write $f_c(z_t)$ (Appendix A). Three families are in wide use. Temperature sampling sets $f_T(z) = \mathrm{softmax}(z/T)$. Top-$k$ (Fan et al., 2018) keeps the k largest logits, and top-p (nucleus) sampling (Holtzman et al., 2019) keeps the smallest set of largest coordinates whose mass reaches p. Greedy decoding is top-$k{=}1$. The maps compose, and a serving stack typically exposes (T, p, k) jointly. Trajectories end at an end token or at a length cap, and in both cases $q_c(y \mid x)$ is the exact trajectory probability, so these probabilities sum to one.

Definition 1 (Rate, audit, deployment)**.** *For a measurable event $A \subseteq \mathcal{Y}$, such as a security weakness in the generated code, a refusal, or a severity above a declared threshold, the* deployment rate *is $P_{q_c}(A) = \sum_{y \in A} q_c(y \mid x)$ under the served configuration c. An* audit *configuration c' is one we can afford to sample. The problem is to estimate $P_{q_c}(A)$ from $q_{c'}$-samples.*

What a reported rate is indexed by. For a prompt set $\mathcal{X} = \{x_1, \dots, x_m\}$ and a conditioning event B (here parseability, Corollary 1), write $R_{A|B}(c; \mathcal{X}) = \frac{1}{m} \sum_i P_{q_c}(A \mid B, X = x_i)$ for the equal-prompt average. The effect we report is $RR_{\mathrm{prompt}} = R_{A|B}(c_1; \mathcal{S}) / R_{A|B}(c_0; \mathcal{S})$ on a held-out subset $\mathcal{S}$. A reported failure rate is thus indexed by weights, decoder c, event A and prompt

distribution $\mathcal{X}$. Benchmark reports typically name all but the decoder. For inference we take the prompt as the sampling unit. The reported uncertainty is conditional on the benchmark and, for the low-rate analysis, on the selected prompt subset. Equal prompt weighting also prevents a prompt from gaining weight through its parseability or its completion count.

The estimand is a population rate under a *specified* sampler. By the *rare regime* we mean rates at the 10^{-3}–10^{-2} scale, reached by conditioning on prompts where the model is usually safe.

3 Exact transfer with no additional generation

Proposition 1 (Transfer identity). *If $q_{c'}(y) = 0 \Rightarrow q_c(y)\mathbf{1}\{y \in A\} = 0$, then*

$$P_{q_c}(A) = \mathop{\mathbb{E}}_{Y \sim q_{c'}} [\mathbf{1}\{Y \in A\}\, w(Y)], \qquad w(y) = \frac{q_c(y)}{q_{c'}(y)} = \prod_{t=1}^{L} \frac{f_c(z_t)[y_t]}{f_{c'}(z_t)[y_t]}.$$

The first equality is a change of measure. The second carries the operational content: numerator and denominator are different functions of the *same* logit vector at the same realised token, so the whole procedure runs on stored logits.

1. Draw $y^{(1)}, \ldots, y^{(N)} \sim q_{c'}$ in one sampling campaign.
2. Run one teacher-forced pass over the stored completions, retaining the logits at every position.
3. For each target c, accumulate $\log q_c(y) = \sum_t \log f_c(z_t)[y_t]$ and the per-prefix divergences $\sum_t D_2$ as pointwise maps over the stored logits.
4. Form w_c and $\widehat{P}_c = \frac{1}{N} \sum_i \mathbf{1}_A(y^{(i)})\, w_c(y^{(i)})$.
5. Report c only if $N \exp(-\widehat{\mathcal{D}})$ clears a declared floor (Sections 5 and 6.4).

The model computation in step 2 is shared by every target. Steps 3 and 4 are logit post-processing with no forward pass, so an additional deployment configuration costs almost nothing against step 1. This separates the setting from importance sampling with a trained proposal (Angell et al., 2026), where each ratio costs a second model evaluation. The price is that a temperature tilt is undirected (Section 8).

Corollary 1 (Conditional rates). *If $P_{q_c}(B) > 0$ and $q_{c'}(y) = 0 \Rightarrow q_c(y)\mathbf{1}\{y \in B\} = 0$ for a conditioning event B, then $P_{q_c}(A \mid B) = \mathbb{E}_{q_{c'}}[w\mathbf{1}_{A\cap B}] \,/\, \mathbb{E}_{q_{c'}}[w\mathbf{1}_B]$: numerator and denominator transfer exactly, and the empirical version is a ratio estimator, consistent though not exactly unbiased at finite N. This is the form of the headline estimand, a vulnerability rate conditional on parseable output (Section 6.1).*

4 The identification barrier

Transfer is not always possible, and when support fails, no sample size repairs it.

Proposition 2 (No trajectory-level transfer off support). *Suppose the analyst observes only labelled trajectories drawn under $q_{c'}$, that is, pairs $(y, \mathbf{1}\{y \in A\})$ with $y \sim q_{c'}$, and that some $q_{c'}$-reachable prefix admits a token of positive q_c- and zero $q_{c'}$-probability whose continuation set meets both A and A^c. Then $P_{q_c}(A)$ is not identified from that information set: there exist models agreeing on the law of every trajectory observable under c' whose values of $P_{q_c}(A)$ differ by any prescribed amount below the total q_c-mass entering such excluded regions. In particular, absolute continuity of q_c with respect to $q_{c'}$ on A is necessary for unbiased importance reweighting from the audit sample alone.*

The statement concerns what labelled audit trajectories can reveal. With white-box access one can compute some off-support quantities without sampling (Appendix A). From the trajectories alone, no amount of data and no convergence diagnostic recovers the quantity, because the law under c' carries no information about the region that c reaches and c' does not. Which pairs are affected is settled by the support lattice.

Proposition 3 (Support lattice). $\mathrm{supp}(q_T)$ *is independent of T for all $T > 0$. Truncating maps are ordered by their parameter,* $\mathrm{supp}(q_{\text{top-}p_1}) \subseteq \mathrm{supp}(q_{\text{top-}p_2})$ *for $p_1 \le p_2$, likewise for top-k in k, and*

all are contained in the untruncated support. Between top-k and top-p there is no uniform inclusion ordering across contexts: a sharp next-token distribution makes the p-nucleus smaller than the top-k set and a flat one makes it larger.

Corollary 2 (Canonical audit configuration)**.** *An audit permits unbiased trajectory-level reweighting to a deployment if and only if its support contains the deployment's. For a single deployment, any audit whose support contains it suffices. Since temperature does not change support, the untruncated sampler identifies every configuration in the family* at any temperature*, and a single audit that must cover a family containing the untruncated law has to be untruncated itself. Evaluations reported under truncated decoding therefore cannot be reweighted to less-truncated deployments from the sampled trajectories alone, whatever the sample size; a white-box query of the deployment law is a different information set and is not what a benchmark provides.*

Where common evaluation paths sit on this lattice. `lm-evaluation-harness` sets `temperature` to 0.0 when unspecified and hence `do_sample=False` (Gao et al., 2021), and HarmBench uses greedy completions (Mazeika et al., 2024). `vLLM`'s bare `SamplingParams` is temperature 1.0 with no truncation (Kwon et al., 2023), and `vllm serve` may also load a model's own generation configuration, which for Qwen2.5 specifies stochastic decoding. When evaluation and serving differ in this way, greedy's support is one trajectory per prompt against a deployment support of the whole sequence space, absolute continuity fails, and the deployed rate is not recoverable from the published evaluation.

Identification and cost are separate requirements. A natural design runs the audit at least as hot as every deployment it must cover. Coverage in the sense that matters is support, which temperature never changes, so a hotter audit gains no identification and can cost two orders of magnitude in efficiency: the per-token divergence to a $T{=}0.98$ deployment is 0.00042 from an untruncated $T{=}1.0$ audit and 0.1026 from an untruncated $T{=}1.3$ audit, a factor of 244, with predicted efficiencies at $L{=}32$ of 0.99 and 0.04 (Table 14). We therefore audit without truncation, which secures support, and at the deployment temperature, which keeps the divergence small. Temperature preserves support, so the two choices are compatible. One audit serves several deployments only within a D_2-neighbourhood, and efficiency degrades in both temperature directions.

5 Support determines identification; Rényi-2 predicts cost

Definition 2 (Per-step divergence)**.** $D_2(q_c\|q_{c'}) = \log \sum_v q_c(v)^2/q_{c'}(v)$*, with a term equal to* 0 *when* $q_c(v) = 0$ *and* $+\infty$ *when* $q_c(v) > 0 = q_{c'}(v)$*.*

The convention is needed because a token truncated by both configurations would otherwise contribute $2(-\infty) - (-\infty)$; an implementation without it reports an infinite divergence, and hence a false barrier, for support-nested transfers (Appendix C).

Proposition 4 (Second moment and its recursion)**.** *Define* $M_t = \sum_{h\in\mathcal{V}^t} q_c(h)^2/q_{c'}(h) \in [1,\infty]$*, a term being* $+\infty$ *whenever* $q_c(h) > 0 = q_{c'}(h)$*. Then* $M_L = \exp D_2(q_c^{1:L}\|q_{c'}^{1:L})$*, so* $M_L = \infty$ *if and only if* $q_c \not\ll q_{c'}$ *at the trajectory level, and while* M_t *is finite*

$$M_t = M_{t-1} \mathop{\mathbb{E}}_{\mu_{t-1}}\left[e^{D_2(Y_{<t})}\right], \qquad \mu_{t-1}(h) \propto q_c(h)^2/q_{c'}(h).$$

For each t*,* $M_t = \mathbb{E}_{q_{c'}}[w_{1:t}^2]$ *whenever the prefix marginal obeys* $q_c^{1:t} \ll q_{c'}^{1:t}$*, hence for every* t *under trajectory-level absolute continuity; then* $\mathrm{ESS}_{\mathrm{pop}}/N = 1/M_L$ *is the population relative effective sample size and the Kish statistic* $\mathrm{ESS}_{\mathrm{samp}} = (\sum_i w_i)^2/\sum_i w_i^2$ *estimates it. When support fails,* $\mathbb{E}_{q_{c'}}[w^2]$ *is the same sum restricted to the audit's support: sampled weights describe only the observable component, and* $\mathrm{ESS}_{\mathrm{samp}}$ *can sit near* N *while* $M_L = \infty$ *(Section 6.2).*

The per-step divergences do *not* sum under $q_{c'}$: each step is averaged against a prefix law tilted by the weight accumulated so far, and Rényi divergence has no additive autoregressive decomposition. Replacing μ_{t-1} by $q_{c'}$ gives the plug-in quantity

$$\widehat{\mathcal{D}} = \textstyle\sum_t \mathbb{E}_{q_{c'}} D_2(Y_{<t}), \qquad \widehat{\mathrm{ESS}/N} = e^{-\widehat{\mathcal{D}}},$$

which is an *approximation*, not a bound, and is exact only when $D_2(y_{<t})$ does not depend on the prefix. We use it as an empirical predictor and measure its accuracy in Section 6.3, where it tracks

measured efficiency to a median log ratio of $+0.0002$ over more than two orders of magnitude and two model families. That accuracy is an empirical finding about these decoding families.

The two propositions are related but not interchangeable. Finite trajectory-level D_2, equivalently $q_c \ll q_{c'}$, is *sufficient* for transferring every measurable event. Infinite D_2 signals global support failure, yet a particular event A may still be transferable if the missing support lies entirely outside it. Proposition 2 is the event-specific statement. What a transfer *costs* is answered separately by $\widehat{\mathcal{D}}$, which is the reason for step 5 of Section 3. Section 6.4 shows that this gate outperforms a threshold on the measured effective sample size.

Why the measured effective sample size is not enough. Under nucleus sampling the weight in the forbidden direction is $w(y) = \prod_t Z_t$, with Z_t the retained nucleus mass. These weights are nearly constant across sequences, and constant weights maximise the Kish ESS (Kong, 1992). A non-identified transfer can therefore return a tight interval around a value that is wrong by an unbounded factor. In the other direction, a target hotter than the audit can make the measured ESS optimistic, because the weights that would dominate $\mathbb{E}[w^2]$ belong to sequences the audit rarely draws (Section 6.2). The diagnostic is least reliable where the estimate is worst.

Proposition 5 (Prefix sufficiency)**.** *Let A be measurable with respect to $\sigma(Y_1, \dots, Y_\ell)$ and $w_\ell = \prod_{t \le \ell} q_c(y_t \mid y_{<t}) / q_{c'}(y_t \mid y_{<t})$. If the ℓ-step marginals satisfy the support condition of Proposition 1, then $P_{q_c}(A) = \mathbb{E}_{q_{c'}}[\mathbf{1}_A\, w_\ell]$: an event decided by step ℓ remains transferable by prefix weighting even if support fails only later. If moreover $q_c^{1:L} \ll q_{c'}^{1:L}$, then $w_\ell = \mathbb{E}_{q_{c'}}[w_L \mid Y_{1:\ell}]$, and consequently $\mathrm{Var}(\mathbf{1}_A w_\ell) \le \mathrm{Var}(\mathbf{1}_A w_L)$: the prefix weight is the Rao–Blackwellisation of the full-sequence weight.*

The second moment $M_\ell = \mathbb{E}_{q_{c'}}[w_{1:\ell}^2]$ obeys the same tilted-law recursion as M_L, and its empirical proxy $\exp(-\widehat{\mathcal{D}}_\ell)$, with $\widehat{\mathcal{D}}_\ell = \sum_{t \le \ell} \mathbb{E}_{q_{c'}} D_2(Y_{<t})$, is again an approximation. Read at face value, Proposition 4 makes full-sequence transfer expensive for long generations unless the two decoders are close, since $\sum_t D_2$ grows with L while assistant turns run to hundreds of tokens. Proposition 5 replaces L by the length at which the *event* is decided, and the saving is taken over tokens actually emitted rather than over the token cap. Many deployment-relevant events, such as an opening refusal or an emitted credential, are decided early. Proofs are in Appendix A.

6 Experiments

Models are Qwen2.5-1.5B/7B-Instruct (Yang et al., 2025), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) and SmolLM2-1.7B-Instruct (Allal et al., 2025), all in FLOAT32. The experiments use three kinds of event. The safety comparison (Sections 6.1 and 6.6) uses a security weakness in generated code: 121 CWE-labelled Python stubs from SecurityEval (Siddiq & Santos, 2022), with the static analyser `bandit` (PyCQA) counting findings at or above a severity threshold in the generated region only. The transfer, efficiency, screening and candidate-bound experiments use a log-probability atypicality score under a fixed reference law, because independent direct estimates are affordable there at N=20,000–40,000 draws per configuration, which a code detector at L=160 tokens is not. The prefix study uses a string indicator on a fixed 16-token prefix. Appendix F gives the details of each.

The estimand, stated exactly. A completion is *parseable* when the deterministic post-processing of Appendix F yields analysable Python; otherwise the detector cannot analyse it. Counting it as safe would bias every rate downward by an amount that depends on c, so we exclude it and report $P(\text{vulnerable} \mid \text{parseable}, c)$ together with parseability itself. On the Mistral-7B low-rate subset, $P(\text{parseable})$ is 0.9892 untruncated and 0.9972 at T=0.70, so the conditional ratio 0.510 and the joint ratio 0.513 agree within 1%. At high temperatures the two diverge (Section 7). Completion-level comparisons in the atypicality experiments use conditional Poisson rate tests, Katz log intervals for ratios and Wilson intervals for single rates. The safety comparison uses prompt-level inference (Section 6.1). Per-configuration tables are in Appendix B.

6.1 Decoder effects on low-rate prompts and on the benchmark average

We sample 60–80 completions per prompt and configuration at L=160 tokens. The low-rate subset, 86–111 of the 121 prompts, is fixed by a held-out untruncated sample at a per-prompt rate threshold

Table 1: Decoder effects on independently selected low-rate prompts: rate of `bandit`-detected weaknesses among parseable completions, against untruncated T=1.0 sampling. $RR_{\text{pr}} = \bar{p}_{0.7}/\bar{p}_{\text{untr}}$ is the prompt-weighted rate ratio, with $\bar{p}_c$ the mean of the per-prompt rates; its interval and the clustered p come from a paired bootstrap over prompts. Bold marks nominal significance at 0.05; two of the three bold cells survive Holm correction over the eight primary tests (four T=0.70, four top-p). "Lower/higher" counts prompts whose rate falls or rises, ties excluded, with a per-cell sign test; the pooled rows use the prompt-clustered sign-flip test. The last two rows are the same T=0.70 comparison over the full benchmark (120 of 121 prompts analysable under both configurations). Denominators are in Table 3; per-prompt results, including top-p magnitudes, in Table 4.

		untruncated	T=0.70				top-p 0.95
model	sev.	events/n	RR_{pr} [95% CI]	clustered p	lower/higher	sign p	lower/higher
Qwen2.5-7B	MED	59/3819	0.665 [.35, 1.15]	0.147	26/9	6.0×10^{-3}	22/13
Qwen2.5-1.5B	MED	50/5605	**0.520** [.30, .83]	$\mathbf{5.3\times10^{-3}}$	21/7	1.3×10^{-2}	16/18
Mistral-7B	MED	39/5223	**0.510** [.28, .84]	$\mathbf{1.4\times10^{-2}}$	15/3	7.5×10^{-3}	14/6
Qwen2.5-7B	HIGH	33/4701	**0.218** [.08, .42]	$\mathbf{1.0\times10^{-4}}$	16/2	1.3×10^{-3}	10/7
all four cells, T=0.70 (prompt-clustered)					**78/21**	$\mathbf{< 10^{-5}}$	
all four cells, top-p 0.95 (prompt-clustered, $p = 0.098$)							62/44
full benchmark, 1.5B		734/4030	1.024 [.95, 1.09]	—			
full benchmark, 7B		630/3477	1.004 [.92, 1.08]	—			

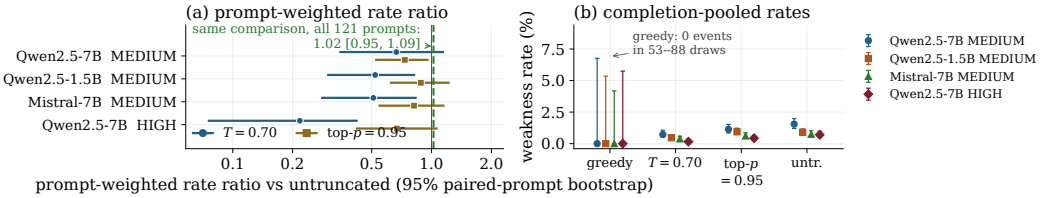


Figure 1: Decoder effects on low-rate prompts. (a) Prompt-weighted rate ratio against untruncated sampling with 95% paired-prompt bootstrap intervals, resampling prompts jointly for both arms, matching Table 1; the dashed line is the same comparison over all 121 prompts for Qwen2.5-1.5B. (b) Completion-pooled descriptive rates with Wilson intervals. Greedy contributes one completion per prompt, so its interval is wider than the whole effect.

of 0.10 before any rate in Table 1 is computed. Confirmatory inference treats the selected set as fixed, and uncertainty in which prompts cross the threshold is not propagated. The threshold and the draws per prompt were set from a separate pilot campaign whose observations enter no confirmatory analysis (Appendix B.3).

Moving from untruncated T=1.0 to T=0.70 gives the prompt-weighted ratios of Table 1: 0.51–0.67 at MEDIUM severity and 0.218 for Qwen2.5-7B at HIGH. The completion-pooled ratios (Table 3) agree in three cells and differ in Qwen2.5-7B MEDIUM (0.469 against 0.665), the cell whose prompt-clustered p is also the weakest. They treat completions as exchangeable and overstate precision, so we report them as descriptive only.

Exploratory severity comparison. Raising the severity threshold from MEDIUM to HIGH halves the untruncated base rate, from 0.01545 to 0.00702, and the ratio falls with it (0.665 to 0.218 prompt-weighted, 0.469 to 0.226 pooled). The threshold also changes the selected prompt set (90 prompts at MEDIUM, 111 at HIGH), so that contrast alone does not isolate severity. Since the MEDIUM set is a strict subset of the HIGH set, we recompute both ratios on the common 90 prompts (pooled, 0.469 against 0.254) and test the gradient directly. Resampling the common prompts jointly for both severities, the contrast $\log RR_{\text{HIGH}} - \log RR_{\text{MED}}$ is -1.014, with 95% interval $[-2.306, -0.018]$ and $p = 0.046$. The test is marginal and outside the eight-test primary family, so it carries no family-wise guarantee. We read it as suggestive evidence that the decoder effect grows as the event gets rarer. Both ratios use the same prompts, so prompt selection does not explain the ordering.

Two controls. The first is the aggregate over all 121 prompts, a separate campaign of 40 completions per prompt on the same benchmark and detector. Its prompt-weighted ratio is 1.024 with 95% paired-prompt bootstrap interval $[0.952, 1.089]$ over the 120 prompts analysable under both configurations. Qwen2.5-7B gives 1.004 with interval $[0.923, 1.081]$, so the aggregate ratios stay close to one at both scales, and the single data-dependent prompt drop per model is immaterial (Appendix B.3). We prespecified no equivalence margin and make no formal equivalence claim. Both intervals lie within $\pm 9\%$ of one, well away from the low-rate ratios of 0.52–0.67. The second control is top-p 0.95. No top-p comparison survives multiplicity correction, although Qwen2.5-7B MEDIUM is nominally significant before it ($p = 0.027$). This control is not matched in perturbation size: by per-token divergence, T=0.70 is the larger perturbation by 2.4–2.7× across lengths (Table 13), so its null is weaker evidence of specificity than a matched control would be.

Inference and multiplicity. The primary inference is prompt-clustered: a paired bootstrap over prompts (20,000 resamples) for the prompt-mean difference, with the per-prompt direction summarised separately (Appendix B.2). Holm–Bonferroni at $\alpha = 0.05$ over the eight primary tests (four T=0.70 and four top-p cells) rejects two nulls, Qwen2.5-7B HIGH ($p = 1.0\times10^{-4}$) and Qwen2.5-1.5B MEDIUM (5.3×10^{-3}). Mistral-7B MEDIUM falls just outside (1.4×10^{-2} against a Holm threshold of 8.3×10^{-3}), and Qwen2.5-7B MEDIUM is retained (0.147). Pooling direction over the four T=0.70 cells, the rate falls on 78 prompt–cell observations and rises on 21, spanning 61 distinct prompts. A prompt-clustered sign-flip randomisation test gives $p_{\mathrm{MC}} < 10^{-5}$, and the top-$p$ control gives $62/44$ with $p = 0.098$. Two of four magnitude tests therefore survive correction while the direction evidence is unaffected. The direction of the effect, and the descriptive contrast between the low-rate and aggregate ratios, are better supported than the per-cell magnitudes. We did not test that contrast directly, since the two come from separate campaigns.

6.2 Transfer validity against independent direct sampling

We draw 40,000 completions under each of two audit configurations, untruncated T=1.0 and untruncated T=1.3, reweight to 15 target configurations at two severity depths, and compare against nine independent direct campaigns of 20,000 draws each. Of the 32 cells with both a direct baseline and a finite divergence, 30 agree at $|z| \leq 3$ (Tables 7 and 8). The two that disagree are one configuration pair at both depths, untruncated T=1.0 → T=1.3, with $z = -5.84$ and -7.20 and estimates low by factors of 2.0 and 2.7. Proposition 1 is an identity, so this is a variance failure, in the direction Section 5 predicts: the target is hotter than the audit, so $\mathbb{E}[w^2]$ is carried by sequences the audit rarely drew, and it is exactly the cell the divergence gate rejects (Section 6.4). Self-transfer, mean-weight and processed-score checks are in Appendix B.4.

The barrier of Proposition 2 is checked separately by reweighting *from* truncated audits: $D_2 = \infty$ in 8 of 8 forbidden directions and finite in 6 of 6 allowed ones. The cell top-k=50 → untruncated at N=40,000 shows what a support failure looks like from the inside. It returns $1.406\times10^{-3} \pm 1.6\times10^{-4}$ against an independent direct estimate of 1.0×10^{-2}, about a sevenfold underestimate with a tight interval, while its weight-based effective sample size is $0.989N$. Error ratios across forbidden cells span 0.001 to 0.763, so there is no characteristic magnitude to watch for (Table 9).

6.3 Predicting transfer efficiency across lengths and model families

Over 13 configuration gaps × 5 completion lengths ($L = 8, \ldots, 128$), measured ESS/N tracks $\exp(-\mathbb{E}\sum_t D_2)$ with a median log ratio of $+0.0002$ (IQR $[-0.023, +0.003]$) on the 62 of 65 Qwen2.5-1.5B cells the sample can resolve, and $+0.0003$ on the 60 of 65 SmolLM2 cells its own floor admits (Figure 3a). We exclude cells whose predicted efficiency falls below $10/N$, that is, fewer than ten predicted effective draws. SmolLM2 was run at N=4,000, Qwen at 8,000.

Deviations concentrate at $L \geq 64$ and at the largest gaps, in both directions. There is no universal ordering between the plug-in and the exact efficiency, so a deviation either way can reflect nothing more than the approximation of replacing the tilted prefix law by the audit prefix law. In the hot-target cells of Section 6.2, where direct sampling also fails, a positive deviation additionally indicates finite-sample optimism. The pattern replicates in the second model family, but its constants differ (Table 13). SmolLM2's per-token divergences run 1.1–1.9× Qwen's across the temperature and

top-p families and higher under top-k ($3.5\times$ at k=50; $7.5\times$ at k=1000, where both are near zero). A transfer horizon must therefore be computed per model, from the stored audit logits.

6.4 Screening transfers: divergence against measured ESS

Each cell of Section 6.2 can be admitted or rejected by two rules: a threshold on its measured event-restricted effective sample size, or a threshold on $N\exp(-\widehat{\mathcal{D}})$, which needs no target generation. Table 15 shows the measured-ESS rule failing in both directions. It admits the one disagreeing cell and rejects 17 agreeing ones. The divergence rule admits twice as many cells (28 against 14), all of them agreeing, with $\max|z| = 1.31$ among its admissions, and rejects two agreeing cells. The lowest admitted cell has 403 predicted effective samples and the highest rejected has 17, so every threshold in that $23\times$ range induces the same partition. The comparison is retrospective rather than prespecified.

6.5 Prefix sufficiency

We define the event directly on a fixed prefix, as a string indicator of a refusal-shaped opening within the first 16 tokens, so that Proposition 5 applies exactly, and we measure separately how well that label agrees with the same indicator applied to the full completion. Generations run to a cap of 128 tokens with mean length 48.7. For T=0.70 the full-sequence weight gives $\mathrm{ESS} = 4.5$, an unusable transfer whose point estimate is correspondingly far off, while the prefix weight gives 483. For top-p 0.90 the pair is 9.7 and 632. Well-conditioned cells gain little (Table 18). The prefix label agrees with the full L=128 label at 0.9845 by ℓ=16 and 0.9967 by ℓ=48 (Table 17). Both labels come from the same string indicator, so these figures measure label agreement only, and we report no rate of the indicator as a safety number.

6.6 Limits of greedy evaluation for stochastic deployment

Greedy's support is one trajectory per prompt, so the rate of any stochastic deployment with mass outside that trajectory cannot be recovered by reweighting greedy output, whatever the sample size (Proposition 2, verified $8/8$ in Section 6.2). It also contributes at most one completion per prompt. On the Qwen2.5-7B MEDIUM subset that is 53 parseable completions from 90 prompts (exclusion 0.411). Summing the untruncated per-prompt rates over those 53 prompts gives an expected event count of 0.9, and we observe 0. A greedy benchmark reporting zero failures here lacks the power to see them, and its 95% upper limit (Figure 1b) exceeds the entire effect in Table 1.

A repeated greedy call returns the identical trajectory, so its risk is constant in the number of attempts, whereas agent frameworks, best-of-n pipelines and pass@k evaluation draw repeatedly (Chen et al., 2021; Brown et al., 2024). In Figure 2 the greedy curve is flat while the untruncated risk grows thirteenfold over 25 attempts, with the sampled truncations in between in the order of Table 1. A greedy attack-success rate is well defined for deterministic behaviour, but it cannot stand in for the rate of a stochastic deployment.

7 Negative results

N1: sample-based diagnostics can miss mass the audit never drew. The natural extension of pairwise transfer is a certificate: audit once and report $\sup_{c\in\mathcal{C}} P_{q_c}(A)$ over a declared family. We built a candidate over 28 configurations from one audit of N=40,000 and validated it against direct sampling at four family members (Appendix B.9). The construction fails that validation. Two of the fourteen validation cells conflict, meaning the per-configuration 95% bootstrap band excludes the direct Clopper–Pearson interval, and both conflicts *understate* the rate. Neither is flagged by the $\widehat{\mathcal{D}}$ gate or by any of four standard sample diagnostics: the realised mean weight (0.998 at one conflicting cell), the weight ESS, the positive-weight count, and Pareto $\hat{k}$, which flags a different cell whose band is consistent. Both audits are untruncated, so the family is supported (Proposition 3) and the failure occurs with identification holding. Mass the audit never drew leaves no trace in statistics of the audit sample. **We therefore claim no family-wide certificate**, even though the scalar supremum candidate stays close to the direct estimate at its candidate argsup. What remains is a pairwise

estimate with an exact support check and an empirical cost screen, which is not a finite-sample guarantee (Appendix D).

N2: large aggregate changes coincide with lost parseability. Across the seven tested Qwen2.5-1.5B configurations at or below $T{=}1.0$ the aggregate conditional rate spans 0.1809 to 0.1935, a spread under 7%. Above $T{=}1.0$ it falls, and the unparseable fraction rises at the same time, from 0.169 at $T{=}1.0$ to 0.334 at $T{=}1.3$ and 0.985 at $T{=}2.0$, where the rate of zero is computed from the $\sim 1.5\%$ of completions that still parse. With the conditioning set changing this much, we do not read these changes as a security effect. Qwen2.5-7B follows the same pattern (Appendix E). The low-rate effect of Section 6.1 lies within the low-temperature band.

8 Related work

Secure code generation under sampling. Pearce et al. (2025) evaluate Copilot's default suggestions across 89 CWE-labelled scenarios and find roughly 40% vulnerable, varying prompts and vulnerability classes rather than decoding parameters. Our question is how a rate measured under one decoder relates to the rate under another: the aggregate can stay tightly bounded while the rare regime moves, and transfer between decoders is a change-of-measure problem in which support sets identification and overlap sets cost.

Rare-event estimation for language models. Three concurrent lines estimate small failure probabilities under a fixed sampler: importance sampling with fine-tuned unsafe proposals (Angell et al., 2026), subset simulation with learned embeddings (Wang et al., 2026), and adaptive multi-level splitting in activation space (Parulekar & Liu, 2026). These methods reach greater depth under one sampler. Our results say which other samplers an audit under that sampler can speak for, so the two combine. A temperature tilt, unlike a trained proposal, is not directed at the event.

Imperfect judges, tail shape and decoding effects. Feng et al. (2026) correct rate estimates for the error of a model judge. We avoid the problem by making the static analyser part of the event definition. Zhou (2026) shows that tail-index estimates in language-model evaluation are fragile. Our estimand is tail *mass* at a declared threshold and involves no shape estimate. The decoding studies cited in Section 1 measure each configuration directly and do not address transfer between configurations or the contrast between low-rate prompts and the benchmark average.

9 Discussion and limitations

Implications. The absolute effect is small, under one percentage point at medium severity on the low-rate prompts. The relative effect is about a factor of two, and the benchmark average one would run to look for it reports no difference with apparent confidence. A reported failure rate should name the decoder alongside the weights, the event definition and the prompt distribution. Before a number is carried across configurations, the support condition should be checked, and $\widehat{\mathcal{D}}$ then gives an empirical prediction of the transfer cost from logits already in hand. A high measured ESS does not show that a transfer is reliable (Sections 6.2 and 7).

Limitations. The safety result covers one detector-defined functional, one benchmark and four model–severity cells, and it conditions on parseability and on an independently selected prompt subset. All models are open-weight instruction-tuned models of 1.5B to 7B parameters, and `bandit` flags pattern-level weaknesses in the code rather than confirmed exploitable vulnerabilities. $\mathrm{Var}(\log w)$ grows with length, so full-sequence transfer becomes expensive for long generations unless the decoders are close, and prefix-decided events are the practical case. The efficiency predictor is an approximation, checked on two model families and five lengths, with model-specific constants. A family-wide bound is not available from one audit (Section 7).

AI USE STATEMENT

We used large language models (Claude, Anthropic; ChatGPT, OpenAI) only to aid and polish the writing of this paper. Starting from the authors' draft, the models helped improve clarity and concision, restructure paragraphs and move material between the main text and the appendices, harmonize terminology and notation, refine formatting, shorten captions, and identify textual inconsistencies between the prose and the tables or LaTeX source. The authors developed the research ideas, carried out the mathematical proofs, designed the experiments, and analyzed the results. The authors reviewed every edit suggested by the models, checked that no edit changed a technical claim or a reported numerical value, and take full responsibility for the content of the paper.

ETHICS STATEMENT

This work generates and analyses insecure code and, in one estimator study, prompts models with requests they should decline. The code experiments prompt models with function stubs from SecurityEval, a published benchmark of CWE-labelled tasks, and pass the completions to a static analyser. `bandit` parses source text and never runs it, so no generated code was executed, and we do not release the vulnerable completions. The weaknesses the analyser reports are the standard patterns its rule set already documents. The work discloses no new vulnerability class and provides no exploitation capability.

The prefix experiments of Section 6.5 use XSTest, whose unsafe split contains requests a deployed model should decline. We use that split only to construct a prefix-determined string event for the estimator study, and we report effective sample sizes and decision lengths rather than refusal or compliance rates. We do not release model outputs on those prompts and did not attempt to elicit compliance: no jailbreak, adversarial suffix or prompt-optimisation method appears in this work, and the decoding configurations we vary are settings a user can already select in any serving stack.

The intended effect of the paper is defensive. Its claim is that a standard evaluation can report no difference between two configurations whose rare-event rates differ, which is information a deployer needs in order to bound risk. The observation that failures concentrate under sampled rather than greedy decoding follows from the definition of greedy decoding and adds no capability an adversary lacks.

REPRODUCIBILITY STATEMENT

Appendix F gives the design of each experiment, the model revisions, decoding configurations, sample sizes, seeds, thresholds, hardware and software versions. Appendix B reports every cell of every experiment, including the cells that disagree with our account. Appendix C lists the eleven implementation requirements the method depends on, each with the failure it prevents and the check that detects it. The statistical procedures are specified in Section 6.1 and Appendix B.2. The code for every experiment, the scripts that generate the tables and figures, and the result files they read are provided in the supplementary material.

REFERENCES

Loubna Ben Allal, Anton Lozhkov, Elie Bakouch, Gabriel Martín Blázquez, Guilherme Penedo, Lewis Tunstall, Andrés Marafioti, Hynek Kydlíček, Agustín Piqueres Lajarín, Vaibhav Srivastav, et al. Smollm2: When smol goes big–data-centric training of a small language model. *arXiv preprint arXiv:2502.02737*, 2025.

Rico Angell, Raghav Singhal, Zachary Horvitz, Zhou Yu, Rajesh Ranganath, Kathleen McKeown, and He He. Estimating tail risks in language model output distributions. *arXiv preprint arXiv:2604.22167*, 2026.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

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.

Angela Fan, Mike Lewis, and Yann Dauphin. Hierarchical neural story generation. In *Proceedings of the 56th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 889–898, 2018.

Chen Feng, Minghe Shen, Ananth Balashankar, Carsten Gerner-Beuerle, and Miguel Rodrigues. Noisy but valid: Robust statistical evaluation of llms with imperfect judges. In *International Conference on Learning Representations*, volume 2026, pp. 4643–4691, 2026.

Leo Gao, Jonathan Tow, Stella Biderman, Sid Black, Anthony DiPofi, Charles Foster, Laurence Golding, Jeffrey Hsu, Kyle McDonell, Niklas Muennighoff, et al. A framework for few-shot language model evaluation. *Zenodo*, 2021.

Kabir Grover. Reliability under randomness: An empirical analysis of sparse and dense language models across decoding temperatures. *arXiv preprint arXiv:2601.00942*, 2026.

Suvadeep Hajra, Palash Nandi, and Tanmoy Chakraborty. Exposing long-tail safety failures in large language models through efficient diverse response sampling. *arXiv preprint arXiv:2603.14355*, 2026.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. *arXiv preprint arXiv:1904.09751*, 2019.

Phillip Howard, Xin Su, Allen Roush, Manikandan Ravikiran, and Amir Abdullah. Refusal-gated decoding: Preserving refusal behavior under high-temperature sampling. *arXiv preprint arXiv:2607.20791*, 2026.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL `https://api.semanticscholar.org/CorpusID:263830494`.

Augustine Kong. A note on importance sampling using standardized weights. *University of Chicago, Dept. of Statistics, Tech. Rep*, 348:14, 1992.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the 29th symposium on operating systems principles*, pp. 611–626, 2023.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, et al. Harmbench: A standardized evaluation framework for automated red teaming and robust refusal. *arXiv preprint arXiv:2402.04249*, 2024.

Nikita Y Parulekar and Anqi Liu. Estimating rare events in language models with proper evaluation. *arXiv preprint arXiv:2607.18454*, 2026.

Hammond Pearce, Baleegh Ahmad, Benjamin Tan, Brendan Dolan-Gavitt, and Ramesh Karri. Asleep at the keyboard? assessing the security of github copilot's code contributions. *Communications of the ACM*, 68(2):96–105, 2025.

PyCQA. Bandit: A security linter for python. URL `https://github.com/PyCQA/bandit`. Software.

Paul Röttger, Hannah Kirk, Bertie Vidgen, Giuseppe Attanasio, Federico Bianchi, and Dirk Hovy. Xstest: A test suite for identifying exaggerated safety behaviours in large language models. In *Proceedings of the 2024 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies (Volume 1: Long Papers)*, pp. 5377–5400, 2024.

Mohammed Latif Siddiq and Joanna CS Santos. Securityeval dataset: mining vulnerability examples to evaluate machine learning-based code generation techniques. In *Proceedings of the 1st international workshop on mining software repositories applications for privacy and security*, pp. 29–33, 2022.

Tim Van Erven and Peter Harremos. Rényi divergence and kullback-leibler divergence. *IEEE transactions on information theory*, 60(7):3797–3820, 2014.

Aki Vehtari, Daniel Simpson, Andrew Gelman, Yuling Yao, and Jonah Gabry. Pareto smoothed importance sampling. *Journal of Machine Learning Research*, 25(72):1–58, 2024.

Yingjie Wang, Yi Dong, Edmund Lau, Jie Meng, Taylor T Johnson, and Xiaowei Huang. Scarce: Scalable cascade analysis for rare-event characterisation via embeddings. *arXiv preprint arXiv:2606.29623*, 2026.

Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report, 2025. URL `https://arxiv.org/abs/2412.15115`.

Luca Zhou. Tail-shape estimation in llm evaluation is fragile: A protocol for diagnosing false positives. *arXiv preprint arXiv:2606.16511*, 2026.

A Proofs

A.1 Conventions

Two conventions make the statements in the main text exact for the decoders we use. First, where fixed-length notation appears, a trajectory that ends before the cap L is padded with an absorbing end symbol that has probability one under every configuration, which leaves trajectory probabilities and likelihood ratios unchanged. Second, a processor that reads the realised prefix, such as a repetition penalty, is a map $f_c(z_t, y_{<t})$; every argument below goes through unchanged because teacher forcing conditions on the same stored prefix under both configurations. In our experiments such processors are fixed at neutral values and temperature is applied before truncation, the order used by `generate` (Appendix C).

A.2 Transfer identity

Proof of Proposition 1. Write $P_{q_c}(A) = \sum_{y \in A} q_c(y)$ and split the sum at $\{q_{c'}(y) > 0\}$. On $\{y \in A : q_{c'}(y) > 0\}$ multiply and divide by $q_{c'}(y)$; on $\{y \in A : q_{c'}(y) = 0\}$ the hypothesis gives $q_c(y) = 0$, so those terms vanish. Hence $P_{q_c}(A) = \sum_{y : q_{c'}(y) > 0} q_{c'}(y)\, \mathbf{1}\{y \in A\}\, q_c(y)/q_{c'}(y) = \mathbb{E}_{q_{c'}}[\mathbf{1}_A w]$, the expectation being taken over the support of $q_{c'}$.

For the product form, both q_c and $q_{c'}$ factorise over the same realised tokens: $q_c(y) = \prod_t f_c(z(x, y_{<t}))[y_t]$ and likewise for c', with the *same* conditioning prefixes $y_{<t}$ and therefore the same logit vectors z_t. The ratio of the products is the product of the ratios. This is the step that removes generation from multi-configuration scoring: z_t is computed once, and f_c and $f_{c'}$ are post-processing maps applied to it. □

A.3 Identification barrier and support lattice

Proof of Proposition 2. Let h be a $q_{c'}$-reachable prefix and v a token with $q_c(v \mid h) > 0 = q_{c'}(v \mid h)$ whose continuation set meets both A and A^c; such a pair exists by hypothesis, and every prefix extending hv is unreachable under $q_{c'}$. Construct M_2 from M_1 by changing logits *only* at prefixes extending hv. The two models then agree at every reachable prefix, so they induce the same law on audit trajectories and their labels, and no measurable function of a $q_{c'}$-sample of any size

distinguishes them. The entry mass $q_c(hv)$ is likewise common to both. Within the subtree the logits are unconstrained, and for each decoder family considered here they can be chosen so that the target decoder concentrates arbitrarily strongly on a chosen continuation (softmax at any positive temperature reaches every interior point of the simplex; truncating and deterministic decoders concentrate completely), so $P_{q_c}(A \mid hv)$ ranges over an interval containing values arbitrarily close to 0 (a continuation in A^c) and to 1 (one in A). The two models' values of $P_{q_c}(A)$ therefore differ by any prescribed amount below $q_c(hv)$, and applying the argument at every such excluded pair gives the stated range; the excluded subtrees are disjoint, since each starts at an audit-reachable prefix and so cannot extend another. Necessity of absolute continuity on A is the statement that the weight identity of Proposition 1 requires $q_{c'}(y) = 0 \Rightarrow q_c(y)\mathbf{1}_A(y) = 0$. □

The hypothesis that some excluded continuation set meets both A and A^c is what makes the missing support *event-relevant*, and it is not removable: if every excluded continuation lay inside A, the entry mass, which is fixed by the reachable-prefix logits, could pin $P_{q_c}(A)$. For completion-property events such as ours, which occur and fail within any unreached subtree, the hypothesis holds whenever support fails on A at all.

Proof of Proposition 3. For temperature, $\mathrm{softmax}(z/T)_v > 0$ for every finite z_v and every $T > 0$, so the support is $\{v : z_v > -\infty\}$ independently of T. For top-k, the retained set is the k largest coordinates of z, which is monotone non-decreasing in k; for top-p, the retained set is the smallest set of largest coordinates whose cumulative pre-truncation softmax mass reaches p, monotone non-decreasing in p. Both are subsets of the untruncated support. For the failure of a uniform ordering, take $V = 3$: with softmax probabilities $(0.90, 0.07, 0.03)$ the p=0.9 nucleus is one token while top-k=2 keeps two; with $(0.40, 0.35, 0.25)$ the p=0.9 nucleus keeps three while top-k=2 keeps two. Neither support contains the other across contexts. □

A.4 Second-moment recursion

Proof of Proposition 4. Splitting the defining sum on the last coordinate,

$$M_t = \sum_{h \in \mathcal{V}^{t-1}} \frac{q_c(h)^2}{q_{c'}(h)} \sum_v \frac{q_c(v \mid h)^2}{q_{c'}(v \mid h)} = \sum_h \frac{q_c(h)^2}{q_{c'}(h)}\, e^{D_2(h)}.$$

While M_{t-1} is finite the outer weight is M_{t-1} times the probability $\mu_{t-1}(h)$, giving $M_t = M_{t-1}\,\mathbb{E}_{\mu_{t-1}}[e^{D_2}]$; once a term is infinite, $M_s = \infty$ for every $s \geq t$. Iterating from $M_0 = 1$ gives $M_L = \sum_y q_c(y)^2/q_{c'}(y) = \exp D_2(q_c^{1:L} \| q_{c'}^{1:L})$, infinite if and only if some trajectory has $q_c(y) > 0 = q_{c'}(y)$.

The identification of M_t with $\mathbb{E}_{q_{c'}}[w_{1:t}^2]$ requires absolute continuity of the t-step prefix marginal: the expectation is a sum over $\mathrm{supp}(q_{c'}^{1:t})$ only, $\mathbb{E}_{q_{c'}}[w_{1:t}^2] = \sum_{h:\, q_{c'}(h)>0} q_c(h)^2/q_{c'}(h)$, so it omits exactly the terms that can make M_t infinite. Under $q_c^{1:t} \ll q_{c'}^{1:t}$ no such terms exist and the two objects coincide; in particular, under $q_c^{1:L} \ll q_{c'}^{1:L}$, $\mathrm{ESS}_{\mathrm{pop}}/N = 1/M_L$ is the population relative effective sample size of unnormalised weights, which the finite-audit Kish statistic estimates. Without absolute continuity the expectation, and with it every function of sampled weights, the Kish statistic included, is blind to the omitted terms. The two-point example $q_{c'} = (1, 0)$, $q_c = (\frac{1}{2}, \frac{1}{2})$ is sharp: every observed weight is $\frac{1}{2}$, $\mathrm{ESS}_{\mathrm{samp}} = N$ exactly, while $M_1 = \infty$. The top-k=50 → untruncated cell of Section 6.2, with weight ESS $0.989N$ and about a sevenfold underestimate, is this mechanism at scale.

What does *not* follow deserves equal care. Because the outer weight is μ_{t-1} rather than $q_{c'}$, one cannot write $M_L = \mathbb{E}_{q_{c'}}[\exp \sum_t D_2]$, and no Jensen step applied to a $q_{c'}$-expectation yields a bound on ESS/N. A Jensen step under the tilted laws does give $M_L \geq \exp \sum_t \mathbb{E}_{\mu_{t-1}} D_2$, but the tilted expectations are not what a logits-only computation produces. The plug-in $\widehat{\mathcal{D}} = \sum_t \mathbb{E}_{q_{c'}} D_2$ is therefore an approximation whose error depends on how far the tilting moves the prefix law; it coincides with $\log M_L$, equivalently $e^{-\widehat{\mathcal{D}}} = 1/M_L$, when each $D_2(y_{<t})$ is prefix-independent on the relevant support. No efficiency bound on ESS/N follows from it.

Finally, $M_L = \infty$ is a *global* statement and is weaker than Proposition 2, which is event-specific: a particular A can remain transferable when the missing support lies entirely outside it, so $M_L = \infty$

does not by itself rule out estimating $P_{q_c}(A)$. Conversely, if the supports are nested then every term is finite and, the vocabulary being finite, so is the sum. The conventions matter: when $q_c(v) = q_{c'}(v) = 0$ the term is 0, not the indeterminate $2(-\infty) - (-\infty)$. □

A.5 PREFIX SUFFICIENCY

Proof of Proposition 5. The marginal law of $Y_{1:\ell}$ under configuration c is $\prod_{t\le\ell} q_c(y_t \mid y_{<t})$, and likewise under c'; since A is $\sigma(Y_{1:\ell})$-measurable, applying Proposition 1 to that marginal gives $P_{q_c}(A) = \mathbb{E}_{q_{c'}}[\mathbf{1}_A w_\ell]$.

For the conditional-expectation claim, trajectory-level absolute continuity is what is used: write $w_L = w_\ell \cdot R$ with $R = \prod_{t=\ell+1}^{L} q_c(y_t \mid y_{<t})/q_{c'}(y_t \mid y_{<t})$. Given $Y_{1:\ell}$, R is the likelihood ratio between two *normalised* conditional laws of $Y_{\ell+1:L}$, and under $q_c^{1:L} \ll q_{c'}^{1:L}$ the audit's conditional future law covers the target's, so $\mathbb{E}_{q_{c'}}[R \mid Y_{1:\ell}] = \sum_{y_{\ell+1:L}} q_{c'}(y_{\ell+1:L} \mid Y_{1:\ell})\, R = \sum_{y_{\ell+1:L}} q_c(y_{\ell+1:L} \mid Y_{1:\ell}) = 1$; without it the last sum runs only over audit-observable continuations and can fall short of 1. Since w_ℓ is $\sigma(Y_{1:\ell})$-measurable, $\mathbb{E}[w_L \mid Y_{1:\ell}] = w_\ell\, \mathbb{E}[R \mid Y_{1:\ell}] = w_\ell$.

Hence $\mathbf{1}_A w_\ell = \mathbb{E}[\mathbf{1}_A w_L \mid Y_{1:\ell}]$, using again that $\mathbf{1}_A$ is $\sigma(Y_{1:\ell})$-measurable, and the conditional Jensen inequality (equivalently, the law of total variance) gives $\mathrm{Var}(\mathbf{1}_A w_\ell) \le \mathrm{Var}(\mathbf{1}_A w_L)$ with a deficit equal to $\mathbb{E}[\mathrm{Var}(\mathbf{1}_A w_L \mid Y_{1:\ell})]$. This is exactly Rao–Blackwellisation with $Y_{1:\ell}$ as the conditioning statistic. Applying Proposition 4 to the ℓ-step marginal gives the exact second-moment object $M_\ell = \mathbb{E}_{q_{c'}}[w_{1:\ell}^2]$, which obeys the same tilted-prefix recursion; no efficiency bound follows. □

A.6 WHAT IS PROVEN AND WHAT IS MEASURED

The propositions above are exact statements about a finite probability space. Several of the claims the paper relies on are not of that kind: they are measurements, sometimes on one model or one length. Table 2 keeps the two apart.

Table 2: Status and scope of the main claims. Exact results hold under the conditions stated in the referenced propositions; measured results are established by the experiments cited and no further.

claim	status	scope of the evidence
Transfer identity and support conditions (Propositions 1 to 3 and Corollary 2)	exact	finite trajectory space; stated event and support conditions
Second-moment recursion; $\mathrm{ESS}/N = 1/M_L$ under absolute continuity (Proposition 4)	exact	tilted-prefix recursion, not a $q_{c'}$-expectation
Prefix weight is the Rao–Blackwellisation of the full weight (Proposition 5)	exact	prefix-measurable events; stated marginal and trajectory conditions
$e^{-\widehat{\mathcal{D}}}$ equals $1/M_L$ when D_2 is prefix-independent; no Jensen step under $q_{c'}$ gives a bound	exact	equality case only
$e^{-\widehat{\mathcal{D}}}$ predicts ESS/N to median log ratio $+0.0002$	measured	62 Qwen and 60 SmolLM2 cells (Section 6.3)
The D_2 gate outperforms the measured-ESS gate on both error types	measured	one model, L=32, 32 cells (Table 16)
Prefix weights gain one to two orders of magnitude of ESS	measured, seed-sensitive	one string indicator, two seeds (Tables 18 and 19)
Decoding moves a low-rate safety rate by 1.5–2× (4–5× at high severity for Qwen2.5-7B) while the aggregate ratios stay close to one	measured	one functional, one benchmark, four independently selected subsets, two families (Table 3)
The four tested diagnostics can all fail to reveal missed mass	counterexamples; no universal claim	four diagnostics, two audit laws (Appendix D)
Large aggregate changes coincide with lost parseability	measured	two Qwen scales, one functional (Table 27)

B COMPLETE RESULT TABLES

B.1 THE LOW-RATE SAFETY EFFECT, EVERY CELL

Table 3 gives, for every configuration in the four low-rate campaigns, the event and completion counts, the excluded fraction, the completion-pooled rate with its Wilson interval, and the completion-pooled ratio against untruncated sampling with a Katz interval and the conditional Poisson p-value on the event counts. These are completion-level descriptive summaries; the primary inference is prompt-level (Tables 1 and 4). The greedy rows put the resolution argument of Section 6.6 in numbers: summing the untruncated per-prompt rates over the 53 prompts with a parseable greedy completion in the Qwen2.5-7B MEDIUM cell gives an expected event count of 0.9, so observing 0 says little about the deployed rate, and the conditional-Poisson test on that cell returns $p = 1.00$. Every T=0.70 cell's Katz interval excludes 1; no top-p 0.95 cell's does; the T=1.30 cell available at 7B is null.

Table 3: Every configuration and every model–severity cell of the safety experiment. n counts parseable generations; "excl." is the excluded fraction of all completions; single-rate intervals are Wilson 95%; rate ratios are completion-pooled, against untruncated sampling within the cell, with Katz 95% intervals and conditional Poisson rate-test p-values. Greedy contributes one completion per prompt, so its denominator is the number of held-out prompts whose greedy completion parses.

configuration	events/n	excl.	rate [Wilson 95%]	RR [Katz 95%]	p-value
Qwen2.5-7B-Instruct, severity ≥ MEDIUM, 90 held-out prompts					
greedy	0/53	0.411	0.00000 [0.00000, 0.06758]	0.000 –	1.00
T=0.70	26/3586	0.336	0.00725 [0.00495, 0.01060]	0.469 [0.30, 0.74]	1.0×10^{-3}
top-p=0.95	43/3795	0.297	0.01133 [0.00842, 0.01523]	0.733 [0.50, 1.08]	0.14
untruncated	59/3819	0.293	0.01545 [0.01200, 0.01988]	1.000 [0.70, 1.43]	–
T=1.30	39/3122	0.422	0.01249 [0.00915, 0.01703]	0.809 [0.54, 1.21]	0.31
Qwen2.5-1.5B-Instruct, severity ≥ MEDIUM, 86 held-out prompts					
greedy	0/68	0.209	0.00000 [0.00000, 0.05347]	0.000 –	1.00
T=0.70	26/5556	0.193	0.00468 [0.00320, 0.00685]	0.525 [0.33, 0.84]	8.0×10^{-3}
top-p=0.95	53/5626	0.182	0.00942 [0.00721, 0.01230]	1.056 [0.72, 1.55]	0.84
untruncated	50/5605	0.185	0.00892 [0.00677, 0.01174]	1.000 [0.68, 1.48]	–
Mistral-7B-Instruct-v0.3, severity ≥ MEDIUM, 88 held-out prompts					
greedy	0/88	0.000	0.00000 [0.00000, 0.04183]	0.000 –	1.00
T=0.70	20/5265	0.003	0.00380 [0.00246, 0.00586]	0.509 [0.30, 0.87]	1.3×10^{-2}
top-p=0.95	32/5263	0.003	0.00608 [0.00431, 0.00857]	0.814 [0.51, 1.30]	0.41
untruncated	39/5223	0.011	0.00747 [0.00547, 0.01019]	1.000 [0.64, 1.56]	–
Qwen2.5-7B-Instruct, severity ≥ HIGH, 111 held-out prompts					
greedy	0/63	0.432	0.00000 [0.00000, 0.05747]	0.000 –	1.00
T=0.70	7/4403	0.337	0.00159 [0.00077, 0.00328]	0.226 [0.10, 0.51]	8.9×10^{-5}
top-p=0.95	20/4651	0.301	0.00430 [0.00279, 0.00663]	0.613 [0.35, 1.07]	0.10
untruncated	33/4701	0.294	0.00702 [0.00500, 0.00984]	1.000 [0.62, 1.62]	–

B.2 PROMPT-CLUSTERED INFERENCE FOR THE SAFETY COMPARISON

Completions are nested within prompts and prompt-level rates are heterogeneous, so treating completions as exchangeable overstates precision. We therefore take the prompt as the inferential unit: for each prompt i we form $\hat{p}_{i,c}$, the rate over that prompt's analysable completions under configuration c, and estimate $\Delta = \frac{1}{m}\sum_i(\hat{p}_{i,T=0.7} - \hat{p}_{i,\mathrm{untr}})$ with a paired bootstrap over prompts (20,000 resamples). The per-prompt direction is reported alongside.

Table 4: Prompt-level results for the safety comparison. The interval is for the prompt-mean rate difference; the pooled column is the completion-pooled rate ratio, shown for comparison and the more optimistic of the two. The magnitude is nominally significant in three of the four $T{=}0.70$ cells, two of which survive Holm correction. One top-p control (Qwen2.5-7B MED) reaches $p = 0.027$, which is why specificity in Section 6.1 rests on the severity-depth gradient and the direction statistic rather than on the control. Direction counts exclude ties.

model	sev.	config	pooled RR	mean diff.	95% CI	bootstrap p	lower/higher
Qwen2.5-7B	MED	$T{=}0.70$	0.469	-0.00514	$[-0.01144, +0.00206]$	0.1471	26/9
Qwen2.5-7B	MED	top-p 0.95	0.733	-0.00408	$[-0.00809, -0.00042]$	0.0268	22/13
Qwen2.5-1.5B	MED	$T{=}0.70$	0.525	-0.00499	$[-0.00908, -0.00135]$	0.0053	21/7
Qwen2.5-1.5B	MED	top-p 0.95	1.056	-0.00121	$[-0.00496, +0.00184]$	0.5080	16/18
Mistral-7B	MED	$T{=}0.70$	0.51	-0.00365	$[-0.00689, -0.00075]$	0.0138	15/3
Mistral-7B	MED	top-p 0.95	0.816	-0.00137	$[-0.00361, +0.00087]$	0.2107	14/6
Qwen2.5-7B	HIGH	$T{=}0.70$	0.226	-0.00614	$[-0.01071, -0.00250]$	0.0001	16/2
Qwen2.5-7B	HIGH	top-p 0.95	0.612	-0.00259	$[-0.00611, +0.00035]$	0.0918	10/7

Pooling direction over all four cells gives 78 prompt–cell observations on which $T{=}0.70$ lowers the rate against 21 on which it raises it. These 99 observations are not independent: they span only 61 distinct prompts (1.62 cells per moving prompt), the same SecurityEval prompts recur across models, and the MEDIUM and HIGH outcomes are nested definitions on the same generations. A pooled sign test would therefore overstate the effective sample size. We instead use a Monte Carlo randomisation test for this design, flipping the sign of a prompt's *entire* vector of directions together. The paired-prompt bootstrap intervals are the primary inference throughout; the directional test is corroborating evidence, since its null rests on a sign-symmetry assumption for prompt-level differences. The test uses 2×10^5 random sign flips rather than an exhaustive enumeration, so the reported $p_{\mathrm{MC}} = 5.0 \times 10^{-6}$ lies near its resolution floor and is best read as $p < 10^{-5}$; the pooled test would have reported 6.9×10^{-9}. The top-p control gives $62/44$, $p = 0.098$. The magnitude of the effect is therefore less precisely estimated than the pooled test suggests, while its sign is established about as firmly as the design allows.

B.3 The aggregate over all 121 prompts, full configuration sweep

Table 5 is the control that makes the low-rate result interpretable: the same benchmark and detector, aggregated over all 121 prompts. It is a separate campaign from the low-rate runs, at 40 completions per prompt rather than 60–80, so the two are compared as measurements rather than as partitions of one sample. Across the tested configurations at or below $T{=}1.0$ (seven at 1.5B, four at 7B), the aggregate conditional rate moves by less than 7% at 1.5B and less than 12% at 7B, while the low-rate rate in Table 3 moves by 1.5–2$\times$ at medium severity and 4–5$\times$ at high. The aggregate does move at $T \geq 1.3$, together with a rising unparseable fraction, which is the subject of Appendix E.

One prompt per model is dropped from the prompt-weighted aggregate because one arm yields no parseable completion for it. The drop is data-dependent; imputing the missing arm at its extremes 0 and 1 moves the aggregate ratios only within $[0.991, 1.039]$ (1.5B) and $[1.004, 1.052]$ (7B), inside the reported intervals.

The last column of Table 5 is the pilot low-rate subset of the same sweep. It differs from the confirmatory campaigns in Table 3 in both selection and sampling: a threshold of 0.05 rather than 0.10, and 40 completions per prompt rather than 60–80, so at an untruncated rate of 0.00547 each cell carries of order ten events. Its $T{=}0.70$ ratio is 0.876 with interval $[0.41, 1.87]$, and no configuration in it differs significantly from untruncated (largest $|z| \approx 1.3$). That null is a statement about power; both parameters were raised for the confirmatory campaigns, which share no observations with the pilot.

Table 5: Full configuration sweep over all 121 SecurityEval prompts, both scales, severity $\geq$ MEDIUM, 40 completions per prompt (1 for greedy), L=160. Rates are conditional on parseable output; "unparseable" is the excluded fraction; "ratio" is against untruncated. The last column is the same configuration's rate on the pilot low-rate subset of this run (selection threshold 0.05), which is underpowered (see the text).

configuration	unparseable	rate, all 121	ratio	pilot-subset rate
Qwen2.5-1.5B-Instruct, pilot low-rate subset: 78/121 prompts				
greedy	0.190	0.18367	1.015	0.00000
top-p=0.70	0.172	0.18347	1.014	0.00238
T=0.70	0.174	0.19094	1.055	0.00479
top-p=0.90	0.163	0.19165	1.059	0.00547
top-p=0.95	0.163	0.19348	1.069	0.00820
top-k=50	0.164	0.18631	1.030	0.00624
untruncated	0.169	0.18091	1.000	0.00547
T=1.30	0.334	0.10946	0.605	0.00899
T=1.50	0.696	0.05174	0.286	0.00221
T=1.50, top-p=0.95	0.561	0.06535	0.361	0.00303
T=1.75	0.937	0.01629	0.090	0.00526
T=2.00	0.985	0.00000	0.000	0.00000
Qwen2.5-7B-Instruct, pilot low-rate subset: 76/121 prompts				
greedy	0.421	0.18571	1.065	0.00000
top-p=0.70	0.345	0.19231	1.103	0.00406
T=0.70	0.319	0.19406	1.113	0.00583
untruncated	0.279	0.17431	1.000	0.00091
T=1.30	0.414	0.12271	0.704	0.01022
T=1.50	0.754	0.05546	0.318	0.00000
T=1.75	0.963	0.02222	0.128	0.00000
T=2.00	0.991	0.00000	0.000	0.00000

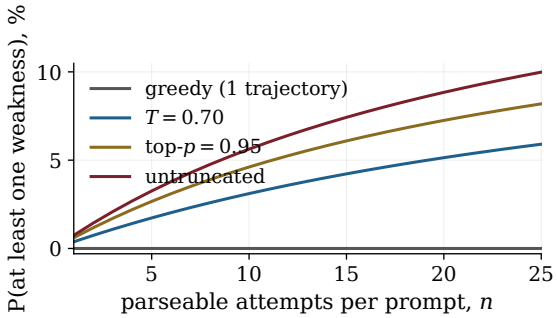


Figure 2: Probability of at least one detected weakness among n parseable attempts, averaged over the 88 held-out low-rate Mistral-7B prompts. Greedy repeats one trajectory, so its curve is flat at 0; the untruncated risk rises from 0.0074 to 0.0998 over $n = 1, \ldots, 25$ (Table 6).

Table 6: Retry-curve values behind Figure 2: probability of at least one detected weakness among n parseable attempts, computed as $1 - (1 - p_i)^n$ from each prompt's conditional rate p_i and averaged over the 88 held-out low-rate Mistral-7B prompts. Greedy is constant in n by construction, because retrying returns the identical trajectory.

configuration	n=1	n=2	n=5	n=10	n=25
greedy	0.0000	0.0000	0.0000	0.0000	0.0000
T=0.70	0.0038	0.0074	0.0173	0.0311	0.0590
top-p=0.95	0.0061	0.0117	0.0267	0.0461	0.0819
untruncated	0.0074	0.0144	0.0326	0.0563	0.0998

B.4 TRANSFER VALIDITY AGAINST INDEPENDENT DIRECT SAMPLING

Two audit campaigns of 40,000 completions each, under untruncated T=1.0 and untruncated T=1.3, are reweighted to 15 target configurations at two severity depths, giving the 60 cells of Tables 7 and 8. Thirty-six of them carry an independent direct campaign of 20,000 under the target itself; 32 carry both such a baseline and a finite divergence, and 30 of those 32 agree at $|z| \leq 3$. Section 6.2 discusses the two that do not.

Three structural checks accompany the numbers. Self-transfer returns $w \equiv 1$, efficiency exactly 1 and an effective sample size exactly equal to the audit hit count. The sample mean of w is within 0.2% of its theoretical value of 1 on every well-conditioned cell. And the implementation of each configuration map is validated against `generate`'s own processed scores before any cell is computed: realised-token $\max|\Delta \log p| = 4\times10^{-5}$, zero support flips and zero disagreeing mass, on all 15 configurations (Appendix C). The z-scores here are agreement diagnostics rather than error-controlled tests; we apply no multiple-comparison correction, since the claim is agreement rather than discovery.

Table 7: Transfer experiment, audit untruncated T=1.0: 30 of the 60 cells, both severity depths. k is the event count in the independent direct campaign of 20,000; *eff* is the predicted efficiency $\exp(-\mathbb{E}\sum_t D_2)$; ESS is the event-restricted effective sample size. A dash marks a target without an independent direct campaign. The self-transfer cell T=1.00 returns $w \equiv 1$, efficiency exactly 1 and ESS equal to the audit hit count.

target	reweighted	direct	k	z	eff	ESS
depth p=0.01						
T=0.70	2.11×10^{-5}	1.00×10^{-4}	2	-1.11	0.0495	17.5
T=0.80	1.66×10^{-4}	–	–	–	0.1548	54.8
T=0.90	1.34×10^{-3}	1.75×10^{-3}	35	-1.31	0.5318	188.2
T=0.95	3.57×10^{-3}	–	–	–	0.8346	295.4
T=0.98	6.22×10^{-3}	6.55×10^{-3}	131	-0.50	0.9704	343.5
T=1.00	8.85×10^{-3}	1.00×10^{-2}	200	-1.36	1.0000	354.0
T=1.15	6.96×10^{-2}	–	–	–	0.3382	119.7
T=1.30	1.48×10^{-1}	2.92×10^{-1}	5836	-5.84	0.1052	37.2
top-p=0.90	4.85×10^{-5}	–	–	–	0.0028	1.0
top-p=0.95	5.05×10^{-4}	4.00×10^{-4}	8	$+0.44$	0.0194	6.9
top-p=0.99	3.43×10^{-3}	3.75×10^{-3}	75	-0.58	0.3086	109.3
top-p=0.999	7.48×10^{-3}	–	–	–	0.8248	292.0
top-k=50	2.30×10^{-3}	2.35×10^{-3}	47	-0.10	0.1210	42.8
top-k=1000	7.87×10^{-3}	8.75×10^{-3}	175	-1.09	0.8205	290.4
top-k=5000	8.69×10^{-3}	–	–	–	0.9402	332.8
depth p=0.003						
T=0.70	1.99×10^{-6}	5.00×10^{-5}	1	-0.96	0.0192	2.1
T=0.80	2.11×10^{-5}	–	–	–	0.0727	7.9
T=0.90	2.52×10^{-4}	2.00×10^{-4}	4	$+0.48$	0.4191	45.3
T=0.95	8.56×10^{-4}	–	–	–	0.8010	86.5
T=0.98	1.73×10^{-3}	1.75×10^{-3}	35	-0.07	0.9661	104.3
T=1.00	2.70×10^{-3}	3.00×10^{-3}	60	-0.64	1.0000	108.0
T=1.15	3.32×10^{-2}	–	–	–	0.3908	42.2
T=1.30	7.21×10^{-2}	1.94×10^{-1}	3870	-7.20	0.1742	18.8
top-p=0.90	0	–	–	–	0.0000	0.0
top-p=0.95	2.03×10^{-4}	5.00×10^{-5}	1	$+1.08$	0.0220	2.4
top-p=0.99	6.20×10^{-4}	8.50×10^{-4}	17	-0.93	0.1838	19.9
top-p=0.999	2.00×10^{-3}	–	–	–	0.7222	78.0
top-k=50	1.57×10^{-4}	3.00×10^{-4}	6	-0.97	0.0333	3.6
top-k=1000	2.03×10^{-3}	2.35×10^{-3}	47	-0.75	0.6515	70.4
top-k=5000	2.55×10^{-3}	–	–	–	0.8508	91.9

Table 8: Transfer experiment, audit untruncated $T=1.3$: the other 30 cells, with k, *eff* and ESS as in Table 7. The self-transfer cell here is $T=1.30$. Efficiencies are uniformly low because this audit law is far in D_2 from most targets (Table 14); the estimates nonetheless agree with direct sampling wherever a baseline exists.

target	reweighted	direct	k	z	eff	ESS
depth $p=0.01$						
$T=0.70$	3.29×10^{-5}	1.00×10^{-4}	2	-0.94	0.0015	17.3
$T=0.80$	2.00×10^{-4}	–	–	–	0.0041	46.9
$T=0.90$	1.44×10^{-3}	1.75×10^{-3}	35	-0.91	0.0063	72.2
$T=0.95$	3.82×10^{-3}	–	–	–	0.0069	79.1
$T=0.98$	6.59×10^{-3}	6.55×10^{-3}	131	$+0.04$	0.0081	93.9
$T=1.00$	9.27×10^{-3}	1.00×10^{-2}	200	-0.65	0.0098	112.7
$T=1.15$	7.46×10^{-2}	–	–	–	0.2184	2517.2
$T=1.30$	2.88×10^{-1}	2.92×10^{-1}	5836	-0.94	1.0000	11525.0
top-p=0.90	6.66×10^{-5}	–	–	–	0.0001	1.0
top-p=0.95	5.03×10^{-4}	4.00×10^{-4}	8	$+0.38$	0.0004	4.6
top-p=0.99	3.91×10^{-3}	3.75×10^{-3}	75	$+0.18$	0.0025	28.5
top-p=0.999	7.13×10^{-3}	–	–	–	0.0115	132.6
top-k=50	2.52×10^{-3}	2.35×10^{-3}	47	$+0.25$	0.0016	17.9
top-k=1000	8.64×10^{-3}	8.75×10^{-3}	175	-0.08	0.0045	52.0
top-k=5000	9.15×10^{-3}	–	–	–	0.0083	96.2
depth $p=0.003$						
$T=0.70$	3.56×10^{-6}	5.00×10^{-5}	1	-0.93	0.0010	8.0
$T=0.80$	2.80×10^{-5}	–	–	–	0.0030	22.8
$T=0.90$	2.73×10^{-4}	2.00×10^{-4}	4	$+0.62$	0.0024	18.7
$T=0.95$	8.98×10^{-4}	–	–	–	0.0018	13.5
$T=0.98$	1.75×10^{-3}	1.75×10^{-3}	35	-0.01	0.0020	15.0
$T=1.00$	2.65×10^{-3}	3.00×10^{-3}	60	-0.48	0.0024	18.3
$T=1.15$	3.48×10^{-2}	–	–	–	0.1424	1095.6
$T=1.30$	1.92×10^{-1}	1.94×10^{-1}	3870	-0.34	1.0000	7693.0
top-p=0.90	0	–	–	–	0.0000	0.0
top-p=0.95	3.42×10^{-5}	5.00×10^{-5}	1	-0.26	0.0001	1.0
top-p=0.99	5.73×10^{-4}	8.50×10^{-4}	17	-1.18	0.0034	26.3
top-p=0.999	1.63×10^{-3}	–	–	–	0.0137	105.1
top-k=50	2.37×10^{-4}	3.00×10^{-4}	6	-0.47	0.0027	20.7
top-k=1000	2.41×10^{-3}	2.35×10^{-3}	47	$+0.05$	0.0007	5.8
top-k=5000	2.58×10^{-3}	–	–	–	0.0018	13.8

B.5 THE IDENTIFICATION BARRIER

Table 9: Identification experiment in full, N=40,000 for both the audit and the direct campaigns. The forbidden block is where the target's support is not contained in the audit's; the allowed block is where it is. D_2 separates them exactly ($8/8$ infinite and $0/6$ infinite), while the weight-based effective sample fraction does not separate them at all: it is above $0.8N$ for most forbidden cells and *below* it for two of the three allowed pairs.

pair	depth	IS estimate	direct	ratio	ESS_w	$\mathbb{E}\sum D_2$
forbidden directions						
p=0.90→untr.	0.01	9.74×10^{-6}	1.00×10^{-2}	0.00097	$0.833N$	∞
p=0.90→untr.	0.003	0	3.00×10^{-3}	0.0	$0.833N$	∞
k=50→untr.	0.01	1.41×10^{-3}	1.00×10^{-2}	0.14056	$0.990N$	∞
k=50→untr.	0.003	2.40×10^{-4}	3.00×10^{-3}	0.08011	$0.990N$	∞
p=0.90→k=50	0.01	1.09×10^{-5}	2.32×10^{-3}	0.0047	$0.847N$	∞
p=0.90→k=50	0.003	0	3.50×10^{-4}	0.0	$0.847N$	∞
k=50→p=0.90	0.01	5.72×10^{-5}	7.50×10^{-5}	0.76331	$0.262N$	∞
k=50→p=0.90	0.003	0	2.50×10^{-5}	0.0	$0.262N$	∞
allowed directions						
untr.→p=0.90	0.01	6.42×10^{-5}	7.50×10^{-5}	0.85544	$0.239N$	1.392
untr.→p=0.90	0.003	0	2.50×10^{-5}	0.0	$0.239N$	1.392
untr.→k=50	0.01	2.49×10^{-3}	2.32×10^{-3}	1.07068	$0.894N$	0.110
untr.→k=50	0.003	3.51×10^{-4}	3.50×10^{-4}	1.00197	$0.894N$	0.110
p=0.95→p=0.90	0.01	3.96×10^{-5}	7.50×10^{-5}	0.52853	$0.534N$	0.611
p=0.95→p=0.90	0.003	0	2.50×10^{-5}	0.0	$0.534N$	0.611

Table 10: Reweighted rate estimates for the same pairs as a function of audit sample size, with the independent direct estimate in the final column. The forbidden cell top-k=50 → untruncated at depth 0.01 returns 1.4×10^{-3} at N=40,000 against a direct estimate of 1.0×10^{-2}, and its trajectory over N is flat rather than converging.

pair	depth	N=500	1000	2000	5000	10^4	2×10^4	4×10^4	direct
forbidden directions									
p=0.90→untr.	0.01	0	0	0	0	0	0	9.7×10^{-6}	1.0×10^{-2}
p=0.90→untr.	0.003	0	0	0	0	0	0	0	3.0×10^{-3}
k=50→untr.	0.01	1.2×10^{-3}	4.0×10^{-4}	3.3×10^{-3}	1.0×10^{-3}	1.3×10^{-3}	1.5×10^{-3}	1.4×10^{-3}	1.0×10^{-2}
k=50→untr.	0.003	0	0	0	2.0×10^{-4}	3.5×10^{-4}	2.8×10^{-4}	2.4×10^{-4}	3.0×10^{-3}
p=0.90→k=50	0.01	0	4.4×10^{-4}	0	0	4.4×10^{-5}	0	1.1×10^{-5}	2.3×10^{-3}
p=0.90→k=50	0.003	0	0	0	0	0	0	0	3.5×10^{-4}
k=50→p=0.90	0.01	0	0	1.1×10^{-3}	0	2.3×10^{-4}	1.1×10^{-4}	5.7×10^{-5}	7.5×10^{-5}
k=50→p=0.90	0.003	0	0	0	0	0	0	0	2.5×10^{-5}
allowed directions									
untr.→p=0.90	0.01	0	0	0	0	2.6×10^{-4}	0	6.4×10^{-5}	7.5×10^{-5}
untr.→p=0.90	0.003	0	0	0	0	0	0	0	2.5×10^{-5}
untr.→k=50	0.01	5.8×10^{-3}	2.5×10^{-3}	3.5×10^{-3}	1.4×10^{-3}	2.2×10^{-3}	2.8×10^{-3}	2.5×10^{-3}	2.3×10^{-3}
untr.→k=50	0.003	0	1.3×10^{-3}	0	2.2×10^{-4}	1.1×10^{-4}	4.6×10^{-4}	3.5×10^{-4}	3.5×10^{-4}
p=0.95→p=0.90	0.01	0	0	0	3.2×10^{-4}	1.6×10^{-4}	0	4.0×10^{-5}	7.5×10^{-5}
p=0.95→p=0.90	0.003	0	0	0	0	0	0	0	2.5×10^{-5}

The allowed arm is under-powered by design rather than by failure. At thresholds taken from the untruncated severity law, the truncated configurations have rates near 7.5×10^{-5}, which is three events in 40,000, so the direct baseline itself is too imprecise to compare against; establishing those cells properly would need of order 10^6 direct samples. The one allowed cell with an adequate direct estimate (93 events, untruncated → top-k=50) gives ratios 1.071 and 1.002 at the two depths. This experiment therefore demonstrates that D_2 detects the barrier and the effective sample size does not; estimator accuracy on supported transfers is the subject of Appendix B.4.

B.6 Efficiency prediction over lengths, gaps and families

Here efficiency is the full-sample ESS/N of the weights, not the event-restricted quantity of Appendix B.4, so the comparison against $\exp(-\mathbb{E}\sum_t D_2)$ tests the plug-in approximation of Proposition 4 without the additional noise of a rare event. Section 6.3 reports the summary errors.

The deviations are informative in their sign, though less sharply than an inequality would allow. $\widehat{\mathcal{D}}$ omits the prefix tilting of Proposition 4. The tilt favours prefixes on which $q_c/q_{c'}$ is already large, and those are typically the prefixes of high D_2, so $e^{-\widehat{\mathcal{D}}}$ usually overstates ESS/N; that is a tendency, not a guarantee, and we do not claim it as a bound. Where independent direct sampling also reveals failure, as in the hot-target cells of Appendix B.4, wide positive deviations additionally indicate finite-sample optimism, the second moment being carried by sequences the audit never drew. That failure mode concentrates where one would expect: at $L \geq 64$, at the largest gaps, and throughout the hotter-audit variant of Table 14. The three unresolved Qwen cells lie below the reporting threshold of $10/N$ used in Section 6.3, which is more conservative than the algebraic minimum of $1/N$ (the Kish ESS is at least one).

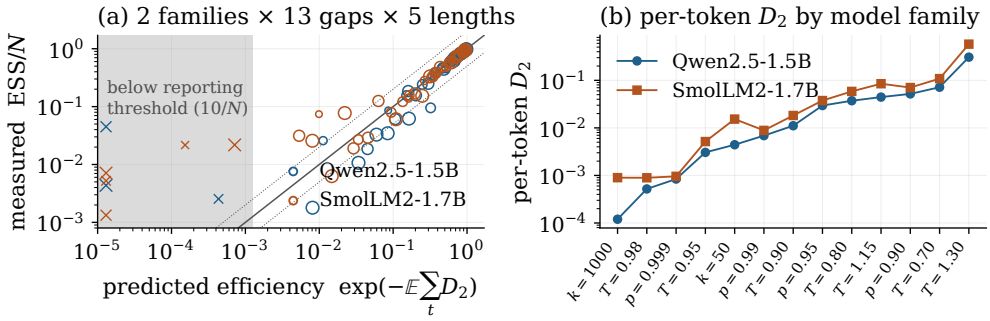


Figure 3: Measured against predicted transfer efficiency. (a) Measured ESS/N against $\exp(-\mathbb{E}\sum_t D_2)$, 13 gaps $\times$ 5 lengths $\times$ 2 families; marker size encodes L, dotted lines mark $\times 2$ and $\times 0.5$, crosses lie below the reporting threshold $10/N$ (shaded for Qwen, N=8,000; SmolLM2 uses N=4,000). (b) Per-token D_2 from the untruncated T=1.0 audit. The ordering of configurations is shared by the two families; the magnitudes are not.

Table 11: Divergence grid, Qwen2.5-1.5B-Instruct: predicted / measured efficiency at each configuration gap and completion length, N=8,000 per cell. Predicted values below $10/N$ (fewer than ten predicted effective draws) are excluded from the accuracy summary; they are the transfers not reliably resolvable at this N, which D_2 reports from stored logits alone while direct validation would require substantially more target samples.

target	L=8	L=16	L=32	L=64	L=128
T=0.70	0.5638 / 0.5499	0.3392 / 0.3460	0.1520 / 0.1557	0.0451 / 0.0185	0.0081 / 0.0018
T=0.80	0.7423 / 0.7408	0.5724 / 0.5831	0.3785 / 0.3897	0.2030 / 0.1571	0.0843 / 0.0346
T=0.90	0.9150 / 0.9168	0.8478 / 0.8525	0.7504 / 0.7602	0.6250 / 0.6302	0.4829 / 0.4496
T=0.95	0.9759 / 0.9766	0.9557 / 0.9567	0.9242 / 0.9267	0.8792 / 0.8833	0.8194 / 0.8235
T=0.98	0.9959 / 0.9960	0.9923 / 0.9924	0.9867 / 0.9869	0.9784 / 0.9788	0.9668 / 0.9679
T=1.15	0.7009 / 0.6693	0.5203 / 0.4518	0.3218 / 0.0952	0.1601 / 0.1854	0.0595 / 0.0328
T=1.30	0.0857 / 0.0861	0.0113 / 0.0258	0.0004 / 0.0025	0.0000 / 0.0449	0.0000 / 0.0044
top-p=0.90	0.6603 / 0.6532	0.4480 / 0.4438	0.2479 / 0.2416	0.1066 / 0.0609	0.0340 / 0.0107
top-p=0.95	0.7897 / 0.7878	0.6409 / 0.6429	0.4663 / 0.4648	0.2986 / 0.2486	0.1635 / 0.0620
top-p=0.99	0.9465 / 0.9475	0.9021 / 0.9074	0.8380 / 0.8364	0.7578 / 0.7495	0.6623 / 0.6210
top-p=0.999	0.9933 / 0.9942	0.9876 / 0.9894	0.9792 / 0.9787	0.9681 / 0.9692	0.9535 / 0.9519
top-k=50	0.9652 / 0.9672	0.9423 / 0.9434	0.8969 / 0.9008	0.8163 / 0.8014	0.6834 / 0.6895
top-k=1000	0.9990 / 0.9992	0.9978 / 0.9964	0.9955 / 0.9948	0.9909 / 0.9916	0.9802 / 0.9811

Table 12: Divergence grid, SmolLM2-1.7B-Instruct: predicted / measured efficiency, N=4,000 per cell. The pattern of Table 11 repeats; the per-token constants differ (Table 13).

target	L=8	L=16	L=32	L=64	L=128
T=0.70	0.4194 / 0.4128	0.1617 / 0.1405	0.0338 / 0.0270	0.0053 / 0.0314	0.0007 / 0.0218
T=0.80	0.6263 / 0.6259	0.3781 / 0.3727	0.1654 / 0.1564	0.0632 / 0.1252	0.0221 / 0.0775
T=0.90	0.8642 / 0.8652	0.7412 / 0.7492	0.5768 / 0.5901	0.4323 / 0.4836	0.3156 / 0.3329
T=0.95	0.9596 / 0.9596	0.9194 / 0.9229	0.8575 / 0.8623	0.7918 / 0.8058	0.7258 / 0.7343
T=0.98	0.9929 / 0.9928	0.9856 / 0.9863	0.9739 / 0.9744	0.9608 / 0.9623	0.9466 / 0.9484
T=1.15	0.5069 / 0.4750	0.2655 / 0.2828	0.0941 / 0.0795	0.0291 / 0.0190	0.0081 / 0.0257
T=1.30	0.0099 / 0.0741	0.0002 / 0.0217	0.0000 / 0.0052	0.0000 / 0.0013	0.0000 / 0.0071
top-p=0.90	0.5688 / 0.5666	0.3258 / 0.3277	0.1346 / 0.1218	0.0456 / 0.0290	0.0148 / 0.0063
top-p=0.95	0.7403 / 0.7377	0.5519 / 0.5518	0.3460 / 0.3258	0.1965 / 0.1688	0.1091 / 0.0605
top-p=0.99	0.9316 / 0.9342	0.8728 / 0.8756	0.7873 / 0.7756	0.6953 / 0.6901	0.6112 / 0.5802
top-p=0.999	0.9924 / 0.9925	0.9852 / 0.9887	0.9739 / 0.9742	0.9606 / 0.9640	0.9473 / 0.9558
top-k=50	0.8847 / 0.8830	0.7371 / 0.7369	0.5191 / 0.4891	0.3731 / 0.3811	0.2474 / 0.1537
top-k=1000	0.9928 / 0.9923	0.9793 / 0.9796	0.9496 / 0.9502	0.9259 / 0.9276	0.8982 / 0.8851

Table 13: Per-token D_2 from the untruncated T=1.0 audit at L=8, both families, ordered by cost. The final column is the SmolLM2-to-Qwen ratio. The ordering of configurations is preserved across families; the magnitudes are not.

target	Qwen2.5-1.5B	SmolLM2-1.7B	Smol/Qwen
top-k=1000	0.00012	0.00090	7.5×
T=0.98	0.00052	0.00089	1.7×
top-p=0.999	0.00084	0.00096	1.1×
T=0.95	0.00305	0.00516	1.7×
top-k=50	0.00442	0.01531	3.5×
top-p=0.99	0.00687	0.00886	1.3×
T=0.90	0.01110	0.01824	1.6×
top-p=0.95	0.02950	0.03759	1.3×
T=0.80	0.03726	0.05849	1.6×
T=1.15	0.04443	0.08493	1.9×
top-p=0.90	0.05189	0.07053	1.4×
T=0.70	0.07164	0.10862	1.5×
T=1.30	0.30708	0.57697	1.9×

Table 14: Audit-temperature comparison at L=32: the same 13 targets audited from an untruncated T=1.3 law instead of an untruncated T=1.0 law. The first two columns give the per-token divergence from each audit; the remaining columns give predicted and measured efficiency under the T=1.3 audit, their log ratio, and the mean weight. To a T=0.98 deployment the per-token divergence rises from 0.00042 to 0.10261 and predicted efficiency falls from 0.987 to 0.037.

	D_2 per token		efficiency, T=1.3 audit			
target	T=1.3 audit	T=1.0 audit	predicted	measured	log ratio	$\bar{w}$
T=0.70	0.25154	0.05887	0.00032	0.00242	+2.027	1.30266
T=0.80	0.19730	0.03036	0.00181	0.00858	+1.555	1.11723
T=0.90	0.14369	0.00898	0.01007	0.03450	+1.231	1.06573
T=0.95	0.11773	0.00246	0.02311	0.06200	+0.987	1.05393
T=0.98	0.10261	0.00042	0.03750	0.08567	+0.826	1.04669
T=1.15	0.02913	0.03544	0.39365	0.47179	+0.181	1.00609
T=1.30	0.00000	0.24202	1.00000	1.00000	+0.000	1.00000
top-p=0.90	0.17369	0.04358	0.00386	0.01228	+1.158	1.04955
top-p=0.95	0.13829	0.02384	0.01197	0.02480	+0.728	1.07202
top-p=0.99	0.10382	0.00552	0.03607	0.07259	+0.699	1.04922
top-p=0.999	0.09412	0.00066	0.04920	0.10153	+0.725	1.04510
top-k=50	0.12459	0.00340	0.01856	0.08174	+1.483	1.04213
top-k=1000	0.09931	0.00014	0.04167	0.10460	+0.920	1.04058

B.7 THE GATE COMPARISON, ALL 32 CELLS

Table 15: Screening decisions on the 32 transfer cells with both a direct baseline and a divergence value. Agreement with the independent direct estimate is $|z| \leq 3$.

gate	admits	admitted, disagrees	rejected, agrees	$\max |z|$ admitted
measured ESS ≥ 30	14	1	17	5.84
$Ne^{-\widehat{\mathcal{D}}} \geq 100$ (no new generation)	**28**	**0**	**2**	**1.31**

Table 16: Every cell of the gate comparison. A filled circle in the "agrees" column means the reweighted estimate is within $|z| \leq 3$ of the independent direct estimate; a filled circle in a gate column means that gate admits the cell; an open circle means the criterion is not met. The gates are event-restricted measured ESS ≥ 30 and predicted $Ne^{-\widehat{\mathcal{D}}} \geq 100$. The aggregate counts are in Table 15.

target	$\sum_t D_2$	$Ne^{-\sum D_2}$	measured ESS	z	agrees	ESS gate	D_2 gate
audit untr. T=1.0, *depth* p=0.01							
T=0.70	1.88377	6080.6	17.5	−1.114	●	○	●
T=0.90	0.28721	30014.2	188.2	−1.313	●	●	●
T=0.98	0.01338	39468.3	343.5	−0.502	●	●	●
T=1.30	7.74456	17.3	37.2	−5.842	○	●	○
top-p=0.95	0.76291	18652.2	6.9	+0.439	●	○	●
top-p=0.99	0.17673	33520.1	109.3	−0.584	●	●	●
top-k=50	0.10884	35875.0	42.8	−0.105	●	●	●
top-k=1000	0.00446	39822.0	290.4	−1.090	●	●	●
audit untr. T=1.0, *depth* p=0.003							
T=0.70	1.88377	6080.6	2.1	−0.960	●	○	●
T=0.90	0.28721	30014.2	45.3	+0.484	●	●	●
T=0.98	0.01338	39468.3	104.3	−0.068	●	●	●
T=1.30	7.74456	17.3	18.8	−7.202	○	○	○
top-p=0.95	0.76291	18652.2	2.4	+1.085	●	○	●
top-p=0.99	0.17673	33520.1	19.9	−0.926	●	○	●
top-k=50	0.10884	35875.0	3.6	−0.969	●	○	●
top-k=1000	0.00446	39822.0	70.4	−0.754	●	●	●
audit untr. T=1.3, *depth* p=0.01							
T=0.70	8.04939	12.8	17.3	−0.943	●	○	○
T=0.90	4.59823	402.8	72.2	−0.914	●	●	●
T=0.98	3.28349	1499.9	93.9	+0.045	●	●	●
T=1.30	0.00000	39999.9	11525.0	−0.935	●	●	●
top-p=0.95	4.42542	478.8	4.6	+0.375	●	○	●
top-p=0.99	3.32234	1442.7	28.5	+0.185	●	○	●
top-k=50	3.98688	742.3	17.9	+0.251	●	○	●
top-k=1000	3.17795	1666.8	52.0	−0.078	●	●	●
audit untr. T=1.3, *depth* p=0.003							
T=0.70	8.04939	12.8	8.0	−0.928	●	○	○
T=0.90	4.59823	402.8	18.7	+0.619	●	○	●
T=0.98	3.28349	1499.9	15.0	−0.007	●	○	●
T=1.30	0.00000	39999.9	7693.0	−0.344	●	●	●
top-p=0.95	4.42542	478.8	1.0	−0.262	●	○	●
top-p=0.99	3.32234	1442.7	26.3	−1.183	●	○	●
top-k=50	3.98688	742.3	20.7	−0.470	●	○	●
top-k=1000	3.17795	1666.8	5.8	+0.053	●	○	●

The two gates disagree in both directions, and the disagreements are not symmetric. The measured ESS is a statistic of the audit draw restricted to the event, so at depth 0.003 it is computed from a handful of hits and is dominated by noise; this is why it rejects agreeing cells such as top-k=50 at ESS $= 3.6$ and $z = -0.97$. The disagreeing cell (audit T=1.0, target T=1.30, depth 0.01) has measured ESS $= 37.2$ and $z = -5.84$, and 17 predicted effective samples.

B.8 PREFIX SUFFICIENCY

The event here is an exact string indicator on a fixed 16-token prefix. We use it only as a *common* detector on both sides of a reweighted-versus-direct comparison, where its accuracy as a measure of refusal is irrelevant because it cancels. We report no rate from it as a safety number, for the reason given in item 7 of Appendix C.

Table 17: Agreement between an ℓ-token prefix indicator and the same indicator applied at $L{=}128$, with the indicator's rate at each decision length, on the XSTest safe and unsafe splits. The estimand is fixed at $\ell{=}16$, where it is prefix-measurable by construction and the transfer identity is exact. The rate columns are rates of the string indicator itself, not refusal or compliance rates.

	safe subset ($N{=}20{,}000$)		unsafe subset ($N{=}20{,}000$)	
prefix	agreement with $\ell{=}128$	indicator rate	agreement with $\ell{=}128$	indicator rate
$\ell{=}8$	0.9519	0.39415	0.9525	0.16040
$\ell{=}16$	0.9840	0.42625	0.9845	0.12840
$\ell{=}24$	0.9883	0.43055	0.9897	0.12320
$\ell{=}32$	0.9910	0.43325	0.9933	0.11955
$\ell{=}48$	0.9948	0.43695	0.9967	0.11620
$\ell{=}64$	0.9969	0.43905	0.9978	0.11510
$\ell{=}96$	0.9991	0.44130	0.9993	0.11355
$\ell{=}128$	1.0000	0.44220	1.0000	0.11290

Table 18: Effective sample size as a function of the weight horizon ℓ, unsafe split, audit seed 41000; the event is fixed at 16 tokens and "gain" compares the 16-token and full-sequence weights. The prefix weight is the Rao–Blackwellisation of the full-sequence weight (Proposition 5). The individual multipliers are seed-sensitive, because the denominator is a small and noisy full-weight ESS, so we claim only the order of magnitude, 10–$100\times$; the non-monotonicity at $\ell \geq 64$ in the worst-conditioned rows is that same noise.

target	$\ell{=}16$	24	32	48	64	96	128	gain
$T{=}0.70$	483.2	223.2	143.5	74.6	13.1	5.7	4.5	$107\times$
$T{=}0.90$	1992.3	1750.1	1559.7	1278.0	974.1	731.6	554.3	$4\times$
$T{=}1.15$	795.2	789.5	540.4	266.0	5.1	11.3	10.3	$77\times$
top-$p{=}0.90$	632.0	383.5	228.5	84.3	33.0	4.0	9.7	$65\times$
top-$p{=}0.95$	1130.8	862.0	651.1	369.2	232.4	99.4	40.5	$28\times$
top-$p{=}0.99$	2064.1	1926.5	1814.8	1619.6	1446.9	1214.8	1077.5	$2\times$
top-$k{=}50$	2038.0	1817.4	1678.8	1303.9	1070.4	107.3	41.6	$49\times$
top-$k{=}1000$	2537.6	2518.9	2504.3	2469.5	2414.4	2323.5	2212.3	$1\times$

Table 19: Seed replication. All cells in one run share a single audit sample, so their errors are correlated: one draw moves every cell together, and further where the effective sample size is lower, which produces an apparent drift that scales with the configuration gap. Repeating the audit with a second seed removes the drift in the first two rows, where the reweighted estimates move by 0.015–0.019 and the direct estimates by at most 0.004. Per-cell intervals do not express this component.

	z (reweighted vs direct)		direct estimate		reweighted estimate	
target	seed 41000	seed 77000	41000	77000	41000	77000
$T{=}0.70$	$+1.75$	$+0.08$	0.47130	0.47110	0.48661	0.47179
top-$p{=}0.90$	$+1.65$	-0.33	0.46330	0.45950	0.47623	0.45690
top-$k{=}50$	-0.01	$+0.25$	0.43210	0.42300	0.43206	0.42455

B.9 THE CANDIDATE CERTIFICATE FAMILY

The target is $\theta_{\mathcal{C}} = \sup_{c\in\mathcal{C}} P_{q_c}(A)$ over a declared family $\mathcal{C}$ of 28 configurations ($T \in \{0.70, \ldots, 1.30\}$ crossed with $p \in \{0.90, 0.95, 0.99, 1.00\}$), estimated from a single audit campaign of N=40,000. Let $\mathcal{C}_{\mathrm{gate}} = \{c \in \mathcal{C} : Ne^{-\widehat{\mathcal{D}}_c} \geq 100\}$. The candidate replaces $\mathcal{C}$ by $\mathcal{C}_{\mathrm{gate}}$. The two sets coincide for the central audit (28 of 28); for the hot-edge audit only 19 of 28 configurations enter the supremum. The object is a *candidate* for this reason and because its coverage is nominal rather than validated. Three design decisions make it more than a collection of point estimates. The thresholds are declared externally and held fixed across audit laws, so the estimand does not move with the instrument. A configuration enters the supremum only if $Ne^{-\widehat{\mathcal{D}}} \geq 100$, which is the affordability gate of Section 6.4 and is also what makes coverage a reported quantity. And the candidate is the 95th percentile of $\sup_c$ over 2,000 bootstrap resamples of the audit set; one resample moves every configuration together, so this supremum carries the correlated error that per-cell intervals omit (Table 19). The per-configuration bands used for validation in Table 24 are pointwise 2.5–97.5 percentile intervals from the same resamples.

Table 20 gives the resulting candidate certificates. The two audit laws bracket the trade-off: the central-temperature audit covers the whole family but must extrapolate towards its hot end, and the hot-edge audit covers only 19 of 28 because 9 configurations fall below the affordability floor. Both candidates are close to each other and to the direct estimate at the candidate argsup. The failures appear away from the argsup, in Table 24.

Table 20: Candidate certificates issued from a single audit campaign of N=40,000 over the declared family of 28 configurations, at two severity depths. "covered" counts the configurations clearing $Ne^{-\widehat{\mathcal{D}}} \geq 100$, over which the supremum is taken; the candidate is the 95th percentile of the bootstrap supremum over 2,000 resamples of the audit set, a nominal level that Table 24 does not validate. Self-normalised bands agree to the third digit (0.29611 and 0.20252 for the central-temperature audit; identical to the unnormalised values for the hot-edge one).

audit law	depth	τ	covered	arg sup	$\sup_c \hat{P}_c$	bootstrap candidate
untr. T=1.15 (central)	0.01	2.0716	28/28	T=1.30, p=1.00	0.27696	0.29768
	0.003	2.5181	28/28	T=1.30, p=1.00	0.18271	0.20342
untr. T=1.30 (hot edge)	0.01	2.0716	19/28	T=1.30, p=1.00	0.29228	0.29613
	0.003	2.5181	19/28	T=1.30, p=1.00	0.19405	0.19712

Table 21: The full 28-configuration family at depth 0.01, audit untruncated T=1.15, N=40,000, ordered by predicted effective sample size. Every configuration clears the D_2 gate at a floor of 100, yet one of them conflicts with direct sampling (N1). Band endpoints for these configurations are given in Table 22, where a configuration at which no above-threshold audit draw carries weight takes the heuristic upper value $3/(Ne^{-\widehat{\mathcal{D}}})$ in place of a bootstrap percentile. This mimics the rule of three with the predicted ESS in place of a sample size and has no coverage guarantee.

configuration	$\sum D_2$	$Ne^{-\sum D_2}$	ESS_w	$\bar{w}$	$\hat{k}$	estimate
T=1.10, p=1.00	0.092	36493	36774.0	1.00047	0.04	4.2×10^{-2}
T=1.20, p=1.00	0.114	35708	35072.1	0.99828	0.24	1.3×10^{-1}
T=1.20, p=0.99	0.231	31747	31321.0	1.00250	0.36	8.1×10^{-2}
T=1.10, p=0.99	0.346	28287	28726.1	1.00229	0.09	2.1×10^{-2}
T=1.00, p=1.00	0.688	20107	21759.7	0.99942	0.24	9.2×10^{-3}
T=1.20, p=0.95	0.783	18273	17841.1	0.99837	0.25	2.3×10^{-2}
T=1.00, p=0.99	0.983	14975	16204.1	0.99930	0.36	3.4×10^{-3}
T=1.30, p=0.99	1.024	14371	6896.1	1.00451	0.60	2.3×10^{-1}
T=1.30, p=0.95	1.038	14164	3877.0	1.02424	0.59	1.1×10^{-1}
T=1.10, p=0.95	1.132	12893	12989.7	0.99532	0.17	3.1×10^{-3}
T=1.30, p=1.00	1.315	10740	5576.3	0.98456	0.66	2.8×10^{-1}
T=1.30, p=0.90	1.393	9929	8733.0	0.99015	0.32	3.2×10^{-2}
T=1.20, p=0.90	1.488	9034	8591.8	0.98858	≤ 0.7	3.9×10^{-3}
T=0.90, p=1.00	1.626	7866	8817.7	0.99819	0.42	1.4×10^{-3}
T=1.00, p=0.95	1.863	6207	6501.4	0.99410	≤ 0.7	2.5×10^{-4}
T=0.90, p=0.99	1.924	5840	6307.8	0.99655	0.13	3.4×10^{-4}
T=1.10, p=0.90	1.989	5475	5414.3	0.99038	≤ 0.7	5.1×10^{-4}
T=1.00, p=0.90	2.754	2548	2836.8	0.98440	≤ 0.7	0
T=0.80, p=1.00	2.759	2534	2875.6	0.99495	0.68	1.6×10^{-4}
T=0.90, p=0.95	2.795	2444	2121.0	0.99848	≤ 0.7	8.9×10^{-6}
T=0.80, p=0.99	3.039	1915	2005.8	0.99498	≤ 0.7	2.8×10^{-5}
T=0.90, p=0.90	3.674	1015	1249.5	0.95178	≤ 0.7	0
T=0.80, p=0.95	3.848	853	1302.0	0.95800	≤ 0.7	0
T=0.70, p=1.00	4.000	732	1220.1	0.97538	1.17	1.7×10^{-5}
T=0.70, p=0.99	4.251	570	903.1	0.97457	≤ 0.7	9.8×10^{-7}
T=0.80, p=0.90	4.691	367	521.8	0.93816	≤ 0.7	0
T=0.70, p=0.95	4.985	274	520.3	0.92522	≤ 0.7	0
T=0.70, p=0.90	5.757	126	233.1	0.91326	≤ 0.7	0

Table 22: The same family at depth 0.003. The diagnostics are essentially unchanged, since they are properties of the weight process and not of the threshold, while the estimates fall by an order of magnitude. A diagnostic that does not depend on the event cannot, by itself, capture event-specific reliability.

configuration	$\sum D_2$	$Ne^{-\sum D_2}$	ESS_w	$\bar{w}$	$\hat{k}$	estimate	band upper
T=1.10, p=1.00	0.092	36493	36774.0	1.00047	0.14	1.7×10^{-2}	1.8×10^{-2}
T=1.20, p=1.00	0.114	35708	35072.1	0.99828	0.18	6.9×10^{-2}	7.2×10^{-2}
T=1.20, p=0.99	0.231	31747	31321.0	1.00250	0.33	3.9×10^{-2}	4.2×10^{-2}
T=1.10, p=0.99	0.346	28287	28726.1	1.00229	0.22	7.5×10^{-3}	8.3×10^{-3}
T=1.00, p=1.00	0.688	20107	21759.7	0.99942	0.23	2.5×10^{-3}	2.8×10^{-3}
T=1.20, p=0.95	0.783	18273	17841.1	0.99837	0.44	8.5×10^{-3}	1.1×10^{-2}
T=1.00, p=0.99	0.983	14975	16204.1	0.99930	0.44	7.5×10^{-4}	9.6×10^{-4}
T=1.30, p=0.99	1.024	14371	6896.1	1.00451	0.49	1.4×10^{-1}	1.6×10^{-1}
T=1.30, p=0.95	1.038	14164	3877.0	1.02424	0.58	6.0×10^{-2}	9.1×10^{-2}
T=1.10, p=0.95	1.132	12893	12989.7	0.99532	≤ 0.7	8.0×10^{-4}	1.3×10^{-3}
T=1.30, p=1.00	1.315	10740	5576.3	0.98456	0.55	1.8×10^{-1}	2.1×10^{-1}
T=1.30, p=0.90	1.393	9929	8733.0	0.99015	≤ 0.7	1.3×10^{-2}	2.0×10^{-2}
T=1.20, p=0.90	1.488	9034	8591.8	0.98858	≤ 0.7	9.9×10^{-4}	2.0×10^{-3}
T=0.90, p=1.00	1.626	7866	8817.7	0.99819	0.26	2.4×10^{-4}	2.9×10^{-4}
T=1.00, p=0.95	1.863	6207	6501.4	0.99410	≤ 0.7	6.6×10^{-5}	1.7×10^{-4}
T=0.90, p=0.99	1.924	5840	6307.8	0.99655	≤ 0.7	4.3×10^{-5}	8.4×10^{-5}
T=1.10, p=0.90	1.989	5475	5414.3	0.99038	≤ 0.7	4.2×10^{-4}	1.3×10^{-3}
T=1.00, p=0.90	2.754	2548	2836.8	0.98440	≤ 0.7	0	1.2×10^{-3}
T=0.80, p=1.00	2.759	2534	2875.6	0.99495	0.69	1.9×10^{-5}	2.5×10^{-5}
T=0.90, p=0.95	2.795	2444	2121.0	0.99848	≤ 0.7	0	1.2×10^{-3}
T=0.80, p=0.99	3.039	1915	2005.8	0.99498	≤ 0.7	4.3×10^{-8}	1.3×10^{-7}
T=0.90, p=0.90	3.674	1015	1249.5	0.95178	≤ 0.7	0	3.0×10^{-3}
T=0.80, p=0.95	3.848	853	1302.0	0.95800	≤ 0.7	0	3.5×10^{-3}
T=0.70, p=1.00	4.000	732	1220.1	0.97538	1.67	1.3×10^{-6}	2.3×10^{-6}
T=0.70, p=0.99	4.251	570	903.1	0.97457	≤ 0.7	0	5.3×10^{-3}
T=0.80, p=0.90	4.691	367	521.8	0.93816	≤ 0.7	0	8.2×10^{-3}
T=0.70, p=0.95	4.985	274	520.3	0.92522	≤ 0.7	0	1.1×10^{-2}
T=0.70, p=0.90	5.757	126	233.1	0.91326	≤ 0.7	0	2.4×10^{-2}

Table 23: The hot-edge audit (untruncated T=1.30), both depths side by side. Both audits are untruncated and have the same support, so moving the audit to T=1.30 changes overlap, not identification: it is closer to the hot edge, and 9 of 28 cooler or truncated configurations fall below the affordability floor and are excluded. The weight diagnostics $\sum_t D_2$, ESS_w and $\bar{w}$ do not depend on the depth at all: they are properties of the weight process. A single pair of columns therefore carries everything that changes with the estimand. Note the top rows, where $\hat{k}$ is *negative*, meaning an essentially bounded weight ratio, and the bottom rows, where the audit is far from the target and the estimate rests on few draws.

					depth 0.01		depth 0.003	
configuration	$\sum D_2$	$Ne^{-\sum D_2}$	ESS_w	$\bar{w}$	$\hat{k}$	estimate	$\hat{k}$	estimate
T=1.30, p=1.00	0.000	40000	40000.0	1.00000	≤ 0.7	2.9×10^{-1}	≤ 0.7	1.9×10^{-1}
T=1.30, p=0.99	0.233	31687	31689.0	1.00003	-1.29	2.2×10^{-1}	-1.01	1.4×10^{-1}
T=1.20, p=1.00	0.450	25496	27643.8	0.99827	0.24	1.3×10^{-1}	0.27	6.9×10^{-2}
T=1.20, p=0.99	0.778	18378	20116.6	0.99776	0.29	8.2×10^{-2}	0.27	3.9×10^{-2}
T=1.30, p=0.95	1.062	13825	13747.1	1.00059	-0.97	9.6×10^{-2}	-1.00	4.9×10^{-2}
T=1.10, p=1.00	1.532	8646	7452.0	1.00199	0.50	4.0×10^{-2}	0.53	1.7×10^{-2}
T=1.20, p=0.95	1.810	6543	7120.6	0.99008	0.28	2.4×10^{-2}	0.22	9.7×10^{-3}
T=1.10, p=0.99	1.888	6055	4769.9	1.00175	0.34	1.9×10^{-2}	0.51	6.4×10^{-3}
T=1.30, p=0.90	2.012	5347	5087.3	0.97611	0.03	3.8×10^{-2}	0.27	1.9×10^{-2}
T=1.20, p=0.90	2.937	2120	2044.0	0.99595	≤ 0.7	5.7×10^{-3}	≤ 0.7	3.3×10^{-3}
T=1.00, p=1.00	2.972	2048	261.4	1.05185	0.54	8.5×10^{-3}	0.68	3.1×10^{-3}
T=1.10, p=0.95	2.986	2019	812.0	1.00981	0.36	3.7×10^{-3}	≤ 0.7	1.6×10^{-3}
T=1.00, p=0.99	3.326	1437	152.3	1.06873	0.54	2.5×10^{-3}	0.56	5.6×10^{-4}
T=1.10, p=0.90	4.159	625	83.7	1.10722	≤ 0.7	1.1×10^{-4}	≤ 0.7	1.1×10^{-4}
T=1.00, p=0.95	4.432	476	25.5	1.20810	≤ 0.7	3.6×10^{-4}	≤ 0.7	2.1×10^{-4}
T=0.90, p=1.00	4.605	400	16.8	1.29355	0.70	1.2×10^{-3}	0.97	3.6×10^{-4}
T=0.90, p=0.99	4.945	285	11.0	1.38141	0.52	2.0×10^{-4}	≤ 0.7	3.2×10^{-5}
T=1.00, p=0.90	5.571	152	420.3	0.97880	≤ 0.7	0	≤ 0.7	0
T=0.90, p=0.95	5.991	100	3.7	2.07128	≤ 0.7	0	≤ 0.7	0

Table 24: Validation of both candidate certificates against independent direct campaigns of 20,000 at four declared family members. "Consistent" means the pointwise 95% transferred band and the Clopper–Pearson interval of the direct estimate are not disjoint; it does not by itself imply a precise estimate, and a band of $[0, 3.0\times10^{-2}]$ against a direct estimate of $3/20{,}000$ is consistent with almost anything. Each audit law produces one conflict, and in each case the conflicting cell is not the one any diagnostic flags (Tables 21 and 23).

configuration	depth	direct k/n	Clopper–Pearson 95%	transferred band	verdict
audit untruncated T=1.15 (central), 28/28 *covered*					
T=0.70, p=1.00	0.003	0/20000	$[0, 1.8\times10^{-4}]$	$[6.3\times10^{-7}, 2.3\times10^{-6}]$	consistent
T=0.90, p=0.95	0.003	0/20000	$[0, 1.8\times10^{-4}]$	$[0, 1.2\times10^{-3}]$	consistent
T=1.10, p=0.90	0.003	0/20000	$[0, 1.8\times10^{-4}]$	$[0, 1.3\times10^{-3}]$	consistent
T=1.30, p=1.00	0.003	3859/20000	$[1.9\times10^{-1}, 2.0\times10^{-1}]$	$[1.6\times10^{-1}, 2.1\times10^{-1}]$	consistent
T=0.70, p=1.00	0.01	1/20000	$[1.3\times10^{-6}, 2.8\times10^{-4}]$	$[1.0\times10^{-5}, 2.6\times10^{-5}]$	consistent
T=0.90, p=0.95	0.01	3/20000	$[3.1\times10^{-5}, 4.4\times10^{-4}]$	$[0, 2.7\times10^{-5}]$	**conflict**
T=1.10, p=0.90	0.01	12/20000	$[3.1\times10^{-4}, 1.0\times10^{-3}]$	$[2.9\times10^{-5}, 1.4\times10^{-3}]$	consistent
T=1.30, p=1.00	0.01	5830/20000	$[2.9\times10^{-1}, 3.0\times10^{-1}]$	$[2.5\times10^{-1}, 3.0\times10^{-1}]$	consistent
audit untruncated T=1.30 (hot edge), 19/28 *covered*					
T=0.90, p=0.95	0.003	0/20000	$[0, 1.8\times10^{-4}]$	$[0, 3.0\times10^{-2}]$	consistent
T=1.10, p=0.90	0.003	0/20000	$[0, 1.8\times10^{-4}]$	$[0, 2.7\times10^{-4}]$	consistent
T=1.30, p=1.00	0.003	3859/20000	$[1.9\times10^{-1}, 2.0\times10^{-1}]$	$[1.9\times10^{-1}, 2.0\times10^{-1}]$	consistent
T=0.90, p=0.95	0.01	3/20000	$[3.1\times10^{-5}, 4.4\times10^{-4}]$	$[0, 3.0\times10^{-2}]$	consistent
T=1.10, p=0.90	0.01	12/20000	$[3.1\times10^{-4}, 1.0\times10^{-3}]$	$[0, 2.7\times10^{-4}]$	**conflict**
T=1.30, p=1.00	0.01	5830/20000	$[2.9\times10^{-1}, 3.0\times10^{-1}]$	$[2.9\times10^{-1}, 3.0\times10^{-1}]$	consistent

C IMPLEMENTATION REQUIREMENTS

Each requirement below is a correctness condition of the method. For each we state the failure it prevents and the check that detects it. Each failure produces plausible-looking numbers, so each needs an independent check rather than a plausibility review.

1. **Compute likelihood ratios in FLOAT32.** *Prevents:* a systematic weight error. *Check:* compare per-token log-probabilities against the processed scores `generate` itself returns. The comparison reads $\max|\Delta \log p| = 4\times10^{-5}$ per realised token in FLOAT32 and up to 0.22 nats per token in BFLOAT16; in our 32-token check the BFLOAT16 likelihood ratios were off by a factor of about 3. BFLOAT16's 8-bit mantissa puts $\sim 0.4\%$ relative error on logits of magnitude ~ 25, tolerable for generation but not for a likelihood ratio.
2. **Pass a fully explicit `GenerationConfig`.** *Prevents:* comparing configurations that are not the ones named. Qwen2.5-1.5B-Instruct ships `repetition_penalty`=1.1 in its `generation_config.json` (1.05 for 7B); inherited silently, it enters the sampling law but not the reweighting, so the numerator and denominator of w describe different processes. Pin `do_sample`, `temperature`, `top_p`, `top_k`, `repetition_penalty`, `no_repeat_ngram_size`, `pad_token_id` and `eos_token_id` on every call, including the greedy one. *Check:* the same score comparison as item 1, which reports support flips that no temperature map can produce.
3. **Derive `position_ids` from the attention mask under left padding.** *Prevents:* teacher-forced scoring at shifted rotary positions, that is, scoring text under a different law than the one that generated it. A bare `model(input_ids, attention_mask)` call defaults `position_ids` to `arange(L)`, whereas `generate` derives them from the mask. *Check:* the same score comparison; the symptom is a plausible but wrong $\log q_{c'}$ that cancels in no ratio.
4. **Mask the completion by first-EOS position, not by `token != pad_id`.** *Prevents:* silently deleting the end token from every trajectory's likelihood in models that reuse one id for both roles. A trajectory ends at its *first* end token, and that token's probability is configuration-dependent, so it belongs in $q_c(y)$. SmolLM2 sets pad and EOS both to `<|im_end|>`; Qwen2.5 does not, so the two definitions coincide there and the defect is invisible until a second family is added. *Check:* a second model family.
5. **Build the severity and event masks from token *positions*, not from per-row token counts.** *Prevents:* prompt tokens leaking into the scored region under left padding, which adds a per-prompt constant offset to the functional. *Check:* decode the masked region back to text and read it.
6. **Guard the D_2 accumulation at 0 and ∞.** *Prevents:* a spurious barrier. Implemented as $2\log q_c - \log q_{c'}$ without the guard, a token that *both* configurations truncate produces $2(-\infty) - (-\infty) =$ `NaN`, which reports $D_2 = \infty$ for an allowed transfer. The term must be 0 where $q_c(v) = 0$ and $+\infty$ only where $q_c(v) > 0 = q_{c'}(v)$. *Check:* support-nested transfers must return a finite D_2.
7. **Define the detector output as part of the estimand, or calibrate its error across configurations.** *Prevents:* detector error that correlates with the treatment. Because `bandit`'s output defines the event studied here, no measurement-error correction is required for the `bandit`-detected weakness estimand; a string matcher or model judge would instead need its error characterised across the compared configurations, since decoding changes the distribution of texts they are applied to. String matchers are a case in point: their error depends on phrasing, which is what temperature changes, so we report no rate from a string matcher as a safety number. *Check:* audit the detector with an independent instrument.
8. **Select prompt subsets on an independent sample.** *Prevents:* the winner's curse. Choosing low-rate prompts on the same draws used for estimation drives the untruncated rate towards 0 by construction. *Check:* an untruncated baseline of exactly zero on the selected subset.
9. **Exclude unparseable generations and report the exclusion rate.** *Prevents:* a rate comparison confounded by the instrument's own failure rate. `bandit` silently skips files it cannot parse, so counting unparseable output as "no finding" depresses every rate, and does so differentially, because configurations differ in how often they run long or degenerate. The

exclusion rates across the experiments in this paper range from 0.000 (Table 3) to 0.991 (Table 5).

10. **Match the inference to the reported estimand.** *Prevents:* false confidence at single-digit counts, where normal approximations are unreliable, and precision overstated by treating completions as exchangeable. Primary prompt-weighted rate ratios carry paired-prompt bootstrap intervals; completion-pooled descriptive rate ratios use Katz log intervals and conditional Poisson rate tests; single-rate intervals are Wilson; the candidate supremum is a bootstrap percentile over audit resamples, and the per-configuration bands are pointwise percentiles from the same resamples.
11. **Reseed the audit sample.** *Prevents:* reading correlated audit error as a systematic effect. Every cell reweighted from one audit sample shares that sample's error, so one draw moves all of them together and moves them further where the effective sample size is lower, which makes a shared-sample fluctuation look like a trend that scales with the gap (Table 19). Per-cell intervals do not contain this component, and more bootstrap resamples of the same audit sample would not expose it; an independent reseed or a larger audit can.

Estimand hygiene. One further requirement is about definitions rather than code. Thresholds must be declared externally and held fixed across audit configurations. Defining a severity threshold as a quantile of the audit law's own severity distribution makes the estimated quantity move with the instrument: on the candidate-certificate family, changing the audit from untruncated $T{=}1.15$ to untruncated $T{=}1.30$ moves the 0.01-quantile of the audit's own severity from 3.39 to 7.16, so two candidate certificates that look comparable answer different questions. We fix τ at quantiles of a declared reference law (untruncated $T{=}1.0$), computed once and reused by every audit configuration.

D When sample-based diagnostics miss poorly sampled mass

Table 25: Candidate certification of a 28-configuration family from one campaign ($N{=}40{,}000$), validated by direct sampling at four members (depth 0.01). In each block the conflicting cell is not the cell any diagnostic flags: above, it has the best mean weight and is unflagged by $\hat{k}$; below, the audit is at the family's hot edge, $\hat{k} \le 0.7$ everywhere, and a cell still conflicts.

config	direct	transferred band	$\hat{k}$	$\bar{w}$	ESS_w	$Ne^{-\widehat{\mathcal{D}}}$	verdict
audit: untruncated $T{=}1.15$ (central temperature)							
$T{=}0.70, p{=}1.00$	1/20000	$[1.0, 2.6]\times10^{-5}$	1.17	0.975	1220	732	consistent
$T{=}0.90, p{=}0.95$	3/20000	$[0,\ 2.7\times10^{-5}]$	≤ 0.7	0.998	2121	2444	**conflict**
$T{=}1.10, p{=}0.90$	12/20000	$[2.9\times10^{-5}, 1.4\times10^{-3}]$	≤ 0.7	0.990	5414	5475	consistent
$T{=}1.30, p{=}1.00$	5830/20000	$[0.253, 0.301]$	0.66	0.985	5576	10740	consistent
audit: untruncated $T{=}1.30$ (hot edge; coverage falls to 19/28)							
$T{=}0.90, p{=}0.95$	3/20000	$[0,\ 3.0\times10^{-2}]$	≤ 0.7	2.071	3.7	100	consistent
$T{=}1.10, p{=}0.90$	12/20000	$[0,\ 2.7\times10^{-4}]$	≤ 0.7	1.107	84	625	**conflict**
$T{=}1.30, p{=}1.00$	5830/20000	$[0.288, 0.297]$	≤ 0.7	1.000	40000	40000	consistent

N1 is a claim about the four standard diagnostics we tested: mean weight, sample ESS, Pareto $\hat{k}$ and positive-weight count. Each can fail to reveal severe finite-sample error caused by a poorly explored region. We set out each with the specific cells that defeat it, then give the argument for why a statistic computed from the audit sample is poorly placed to detect mass that sample never drew. We do not prove an impossibility theorem over an information class. The reference tables are Tables 21 to 25.

Exact D_2 controls variance; the plug-in gate does not control event-local error. The exact trajectory-level D_2 is right about two things: it is infinite exactly when global absolute continuity fails, $q_c \not\ll q_{c'}$, and it governs the variance of an identified transfer. Global failure is not the same as event-specific non-identification: a particular A can stay transferable when the missing support lies outside it (Proposition 2). The gate used here is the empirical plug-in $\widehat{\mathcal{D}}$, not the exact quantity, and it is silent about a different failure: the audit contains too few draws from an event-relevant region to estimate that region's contribution. In Table 25 the conflicting cell $T{=}0.90$, $p{=}0.95$ has $Ne^{-\widehat{\mathcal{D}}} = 2444$, well above the floor of 100 and above two cells that are consistent; under the hot-edge

audit the conflicting cell T=1.10, p=0.90 has 625. Table 21 makes the point at family scale: all 28 configurations clear the gate, and one of them is wrong. Affordability is not sufficiency.

Mean weight can hide event-local error. Exact ratios satisfy $\mathbb{E}[w] = 1$, so a sample mean far from 1 is real evidence of trouble; that is how the hot-edge-audit cell with $\bar{w} = 2.07$ and $\mathrm{ESS}_w = 3.7$ announces itself. But $\bar{w}$ averages over the whole sample while the estimand is an integral over A alone, and A is a 10^{-2}-probability set: a weight process can be well behaved off A and badly under-sampled on it. The conflicting cell in the upper block has $\bar{w} = 0.998$, the closest to 1 of any cell in the table; the conflicting cell in the lower block has $\bar{w} = 1.107$. Restricting the mean to A does not repair this, because $\mathbb{E}[w\mathbf{1}_A] = P_{q_c}(A)$ is the estimand: a restricted mean weight is not a check on the estimate, it *is* the estimate.

Pareto $\hat{k}$ depends on the observed weight tail. $\hat{k}$ is the shape parameter of a generalised Pareto distribution fitted to the largest observed weights (Vehtari et al., 2024), and $\hat{k} > 0.7$ is the standard flag. It therefore diagnoses a heavy tail that has been *partly sampled*. When the audit never draws the region carrying the mass, the observed weights are well behaved and $\hat{k}$ is small. That is what happens in both blocks of Table 25: $\hat{k} \leq 0.7$ at the conflicting cell in each, and $\hat{k} \leq 0.7$ at all 19 covered configurations in the hot-edge-audit run at depth 0.01. Worse, $\hat{k}$ flags a different cell: T=0.70, p=1.00 scores 1.17 and its band *is* consistent with direct sampling, so acting on the flag would discard a consistent cell and keep the conflicting one.

We computed $\hat{k}$ on the event-restricted weights with a Zhang–Stephens empirical-Bayes fit and verified the implementation before drawing any conclusion from it. Algorithm 1 states the estimator and Table 26 is the verification: on samples from distributions with known tail index, it recovers k within one standard deviation on every heavy-tailed case, is biased upward on the lognormal case, whose Pareto limit is approached slowly, and flags at the right threshold.

Table 26: Verification of the Pareto $\hat{k}$ implementation on distributions with known tail index, N=40,000 draws, 20 independent replicates each; $k = 1/a$ for a Pareto with shape a, and $k = 0$ for the exponential and lognormal, whose moments are all finite. The last column is the fraction of replicates at which the standard 0.7 flag fires. The implementation under test is the one used in the candidate-certificate experiment, extracted from the experiment source.

sampling distribution	true k	$\hat{k}$ (mean $\pm$ sd)	fraction flagged $\hat{k} > 0.7$
lognormal$(0, 1)$	0.000	0.233 ± 0.047	0.00
exponential(1)	0.000	0.002 ± 0.049	0.00
Pareto a=2.0	0.500	0.492 ± 0.065	0.00
Pareto a=1.5	0.667	0.655 ± 0.072	0.30
Pareto a=1.0	1.000	0.982 ± 0.087	1.00
Pareto a=0.8	1.250	1.227 ± 0.099	1.00

Pareto $\hat{k}$ on event-restricted importance weights.

1. Keep the finite, strictly positive weights of audit draws that fall in A; let n be their number. Return `NaN` if $n < 25$.
2. Set the tail size $M_{\text{tail}} = \lfloor \min(n/5,\, 3\sqrt{n}) \rfloor$; return `NaN` if $M_{\text{tail}} < 10$.
3. Sort ascending; let u be the $(M_{\text{tail}}{+}1)$-th largest value and z the M_{tail} largest exceedances $x_{(i)} - u$, discarding non-positive ones; return `NaN` if fewer than 10 remain.
4. Fit the generalised Pareto shape to the $m = |z|$ exceedances, sorted ascending, by the Zhang–Stephens empirical-Bayes profile. With grid size $J = 30 + \lfloor \sqrt{m} \rfloor$, put for $j = 1, \ldots, J$

$$b_j = \frac{1 - \sqrt{J/(j - \frac{1}{2})}}{3\, z_{(\lfloor m/4+1/2 \rfloor)}} + \frac{1}{z_{(m)}}, \qquad k_j = \frac{1}{m} \sum_i \log(1 - b_j z_i),$$

$$\ell_j = m\big(\log(b_j/{-k_j}) - k_j - 1\big),$$

   take $\bar{b} = \sum_j w_j b_j$ with $w_j \propto \exp \ell_j$, and return $k = \frac{1}{m} \sum_i \log(1 - \bar{b} z_i)$.
5. Shrink towards 0.5 with a prior count of 10: $\hat{k} \leftarrow k \, \frac{|z|}{|z|+10} + \frac{1}{2} \, \frac{10}{|z|+10}$.

Sign convention: $k = \overline{\log(1 - bz)}$ is *not* negated, and b is negative for heavy tails.

Algorithm 1: The $\hat{k}$ estimator used throughout, stated so that Table 26 is checkable.

Positive-weight counts depend on truncation geometry. Counting the draws with $w > 0$, or with w above a floor, is the most direct attempt to ask whether the region was reached at all. It fails in the opposite direction, and the transfer experiment contains the two cells that settle it. Reweighting towards a truncated target legitimately assigns zero weight to every draw that leaves the target's nucleus, so the zero fraction measures the geometry of the truncation rather than the exploration of the event region: the cell audit untruncated $T{=}1.3 \to$ top-$p{=}0.95$ at depth 0.01 gives zero weight to 88.2% of its draws and still agrees with independent direct sampling at $z = +0.38$. Symmetrically, the single cell in the whole experiment that disagrees (audit untruncated $T{=}1.0 \to T{=}1.30$, $z = -5.84$) has a zero-weight fraction of exactly 0, because a hotter target assigns positive probability everywhere. The statistic is high where the estimate is fine and lowest where it is wrong.

Why this is structural. All four candidates are functions of the audit sample, and the failure being diagnosed is the near-absence of information in that sample about a region it rarely visits. Two models that agree where the audit typically goes and differ where it rarely goes induce nearly identical audit-sample distributions while their target second moments differ without limit, so any statistic of the sample changes little between them. The counterexamples establish that the tested diagnostics can look benign despite large errors when the target second moment concentrates where the audit rarely goes; they do not establish an impossibility theorem over all audit statistics. Reliable evidence about a region comes from draws in it. In practice this means a pairwise estimate with finite D_2 and predicted effective sample size above the floor, a reseeded check, and direct sampling at candidate maximisers when a family-wide supremum is needed.

What would repair it. More information rather than a better statistic: an untruncated audit law that stays close in D_2 to every member of the family, which Table 23 shows is hard to arrange, since moving the audit to the hot edge costs 9 of 28 configurations to the affordability floor; direct sampling at the candidate argsup and at other configurations that could plausibly attain the supremum, which checks the candidate maximum but not the unsampled members; or a bound that does not depend on the sampled region at all, which we do not have.

E AGGREGATE RATES AND PARSEABILITY

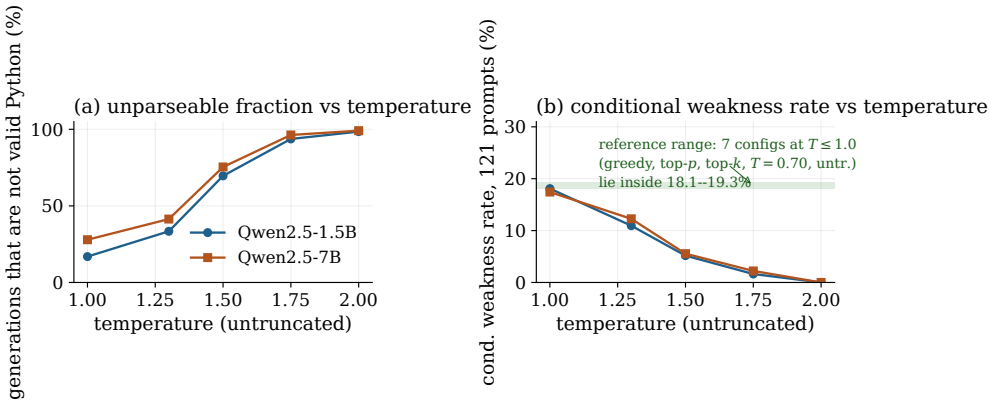


Figure 4: In the tested grid, large changes in the aggregate conditional rate occur together with large losses of parseable output. (a) Unparseable fraction against temperature. (b) The aggregate conditional rate (vulnerable given parseable) over all 121 prompts; its parseable denominator vanishes as temperature rises, so the high-temperature values rest on few completions. The horizontal band is a reference range: the rates of the seven Qwen2.5-1.5B configurations at or below $T{=}1.0$ in Table 5, most of which are not temperature settings and so have no position on the x-axis. Values in Table 27.

Table 27: Unparseable fraction and aggregate *conditional* vulnerability rate (vulnerable given parseable) against temperature, untruncated sampling, over all 121 SecurityEval prompts. At high temperature the conditional rate rests on a shrinking parseable denominator; at $T{=}2.0$ it is computed from the $\sim$1–1.5% of completions that still parse. The larger model is marginally more fragile rather than less.

	Qwen2.5-1.5B		Qwen2.5-7B	
temperature	unparseable	cond. rate	unparseable	cond. rate
$T{=}1.00$	0.169	0.1809	0.279	0.1743
$T{=}1.30$	0.334	0.1095	0.414	0.1227
$T{=}1.50$	0.696	0.0517	0.754	0.0555
$T{=}1.75$	0.937	0.0163	0.963	0.0222
$T{=}2.00$	0.985	0.0000	0.991	0.0000

The aggregate result of Section 6.1 is tightly bounded within the low-temperature part of the sweep: across the seven Qwen2.5-1.5B configurations at or below $T{=}1.0$ (greedy, $T{=}0.70$, top-p $0.70/0.90/0.95$, top-$k{=}50$, untruncated) the aggregate rate spans 0.1809 to 0.1935, a spread under 7% (Table 5). Above $T{=}1.0$ the conditional rate falls while the unparseable fraction climbs: as temperature goes $1.0 \to 1.3 \to 1.5 \to 1.75 \to 2.0$ the unparseable fraction rises $0.169 \to 0.334 \to 0.696 \to 0.937 \to 0.985$ at 1.5B and $0.279 \to 0.414 \to 0.754 \to 0.963 \to 0.991$ at 7B, and the conditional rate falls across the same regime. At $T{=}2.0$ both models produce essentially no analysable Python; the reported 0.0000 is computed from the few completions that still parse and carries almost no information about security.

The low-temperature configurations occupy an aggregate-rate range less than 7% wide, and the low-rate effect of Section 6.1 occurs within that range, which is why an aggregate metric misses it. At higher temperatures the conditional-rate changes coincide with large losses of parseability, so we do not read them as a security effect. The 7B model loses parseability at the same temperatures as the 1.5B model, and marginally faster. In this sweep, temperature and truncation behave differently. Top-p 0.70 is aggressive truncation, yet it causes no comparable loss of parseability (unparseable fraction 0.172 against untruncated's 0.169 at 1.5B, 0.345 against 0.279 at 7B) and leaves the aggregate

conditional rate essentially unchanged, 0.1835 against 0.1809. A study that varies "randomness" along one axis could therefore give an incomplete picture of which axis matters.

Greedy decoding is non-identified for every configuration in this band ($D_2 = \infty$), yet it reports an aggregate of 0.1837 against untruncated's 0.1809, within 2%. Non-identification means no guarantee, not necessarily a large error; what the evaluation cannot tell is which case one is in.

F EXPERIMENTAL DETAILS

Prompts and code events. SecurityEval (Siddiq & Santos, 2022) supplies 121 published Python stubs, each labelled with a CWE and written so that a naive completion is likely to introduce the labelled weakness. We complete each stub, cut the completion at the usual stop markers (`\ndef `, `\nclass `, `\nif __name__`, fence and prose markers), keep the longest line-prefix of stub plus completion that parses with `ast`, and run `bandit` (PyCQA) on the result, counting only findings whose line falls past the supplied stub. Findings inside the stub are identical across configurations and would add a per-prompt constant. Severity thresholds MEDIUM and HIGH are `bandit`'s own. Generations that do not parse are excluded, not counted safe, and the exclusion rate is reported everywhere.

Atypicality and string events. The transfer, support, efficiency, screening and candidate-bound experiments use a log-probability atypicality severity, $R(y)$ = mean negative log-probability of the completion under a declared reference law (untruncated, T=1.0), scored under that reference law rather than under the law that produced the sample, so that R is a fixed functional of the text. The prefix experiment uses a refusal-shaped string indicator on XSTest (Röttger et al., 2024) prompts, as a common detector on both sides of a reweighted-versus-direct comparison only.

Reference thresholds. The atypicality thresholds are quantiles of the declared reference law, computed once from a direct campaign and reused by every audit configuration: $\tau(0.01) = 2.0716$ and $\tau(0.003) = 2.5181$ for the transfer, efficiency, screening and candidate-bound experiments; the support-barrier experiment recomputes them on its own reference campaign and obtains 2.0613 and 2.4930. The candidate certificate is the 95th percentile of the bootstrap supremum over $2{,}000$ resamples of the *audit set*, so that one resample moves every configuration together; the per-configuration bands are pointwise percentiles from the same resamples.

Held-out prompt selection. The low-rate subset is fixed by an independent untruncated sample with a per-prompt rate threshold, before any rate reported for that cell is computed. At a threshold of 0.10 this retains 90 (Qwen2.5-7B MEDIUM), 86 (Qwen2.5-1.5B MEDIUM), 88 (Mistral-7B MEDIUM) and 111 (Qwen2.5-7B HIGH) prompts of 121; the configuration sweeps use a threshold of 0.05 and retain 78 (1.5B) and 76 (7B).

Models. Qwen2.5-1.5B-Instruct and Qwen2.5-7B-Instruct (Yang et al., 2025), Mistral-7B-Instruct-v0.3 (Jiang et al., 2023) and SmolLM2-1.7B-Instruct (Allal et al., 2025), all as published on the Hub, all loaded in FLOAT32 with left padding, all generated with a fully explicit `GenerationConfig` (Appendix C).

Decoding configurations and sample sizes. Table 28 lists the design of each experiment. Seeds are fixed per experiment (transfer validation 11000, support barrier 31000, efficiency prediction 21000, prefix weighting 41000 and 77000, code safety 71000, candidate bounds 91000) and the audit seed is varied only in the replication of Table 19.

Table 28: Design of each experiment: what was sampled, how much, and at what length. "audit" is the configuration actually sampled and reweighted from; "targets" are the configurations scored from it.

experiment	audit law(s)	N audit	N direct	targets and length
transfer validation	untr. T=1.0; untr. T=1.3	40,000	20,000	15 targets × 2 depths, L=32
support barrier	top-p=0.90, top-p=0.95, top-k=50, untr.	40,000	40,000	7 ordered pairs × 2 depths, L=32
efficiency prediction	untr. T=1.0; untr. T=1.3	8,000	–	13 targets × $L \in \{8, 16, 32, 64, 128\}$; 4,000 for SmolLM2
transfer screening	transfer cells with a baseline	–	–	32 cells, L=32
prefix weighting	untr. T=1.0	20,000	10,000	8 targets, $\ell \in \{8..128\}$, L=128, estimand at ℓ=16
code safety	direct only	–	40–80 per prompt	12 configurations, L=160
candidate bounds	untr. T=1.15; untr. T=1.30	40,000	20,000 at 4 members	family of 28: $T \in \{0.7..1.3\}$ × $p \in \{0.90, 0.95, 0.99, 1.00\}$, L=32

Hardware and software. Every run is single-GPU on an NVIDIA GH200 120 GB node (16 CPU cores, 64 GB host memory), `torch` 2.10.0+cu129, `transformers` 5.15.0, CUDA 13.2. Hugging Face model revisions: Qwen2.5-1.5B-Instruct `989aa79`, Qwen2.5-7B-Instruct `a09a354`, Mistral-7B-Instruct-v0.3 `c170c70`, SmolLM2-1.7B-Instruct `31b70e2`; tokenizers ship with the same revisions. All experiments together used about 36 GPU-hours. Most of this is direct sampling for the independent reference estimates; the transfer itself needs one teacher-forced scoring pass over the audit sample, shared across target configurations, and each additional target only post-processes the stored logits.