TESTING WHETHER A JAILBREAK DEFENSE REDUCES ATTAINABLE RISK

Anonymous authors
Paper under double-blind review

ABSTRACT

Jailbreak defenses are usually evaluated by giving an attack a fixed number of attempts and measuring how often it succeeds. A lower success rate can mean two different things: the defense made some behaviors unreachable, or it only made them take more attempts. A fixed budget cannot tell these apart, so delay can be reported as prevention. We introduce a certificate that separates the two: it replays queries that already succeeded on the undefended model, and every one that still succeeds proves a behavior was not prevented. We validate the certificate on ANCHOR, a benchmark we construct in which each defense's true effect is known. A standard evaluation reports a large gain for a reversible input recoding that cannot prevent any behavior, whereas our certificate shows it could have prevented at most 3.6% of them. Stochastic filters leave behaviors reachable but increase the attempts required. The certificate also gives informative bounds for four published defenses. Jailbreak evaluations should therefore report prevention and attacker cost separately.

1 INTRODUCTION

A jailbreak defense is intended to stop a model from producing restricted behavior. In practice, defenses are often evaluated by giving an attack a fixed number of attempts and comparing its success rate before and after the defense is added. A reduction is then treated as evidence of improved robustness. However, the measured score also depends on whether the attack has adapted to the defense and whether the evaluator correctly recognizes successful behavior. Stronger attacks frequently overturn earlier robustness claims (Andriushchenko et al., 2025; Nasr et al., 2025), while automated judges can misclassify adversarial outputs (Schwinn et al., 2026).

A lower measured success rate can reflect either prevention or delay. A defense may make some behaviors unreachable, reducing what an attacker can eventually obtain, or it may lower the probability of success on each attempt while leaving the same behaviors reachable. Both effects may be valuable, but a fixed query budget cannot distinguish between them. Interpreting delay as prevention can therefore overstate the defense's effectiveness.

To separate these outcomes, we begin with queries known to succeed on the undefended model. We adapt each query using only the defense's public rules (a *licensed* adaptation), submit it to the defended model, and verify the response. Every recovered behavior is proven to remain reachable and cannot be counted as prevented. Aggregating these observations gives a conservative upper bound on the defense's attainable-risk reduction, while the attempts required to recover each behavior measure attacker cost separately.

To test whether our method reaches the correct conclusions, we build ANCHOR, a controlled benchmark of benign tasks with exact verifiers. Some of its defenses are designed to look effective without preventing any behavior: they either rewrite the query or make success take more attempts. Others genuinely prevent a known fraction of behaviors. The known effect of each defense allows ANCHOR to test whether an evaluation can distinguish prevention from delay.

The ANCHOR experiments show how delay can pass for prevention. A standard fixed-budget evaluation credits a reversible input recoding with a large robustness gain, although the recoding cannot remove any behavior; replaying each successful query through its published inverse recovers every

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.

reached behavior and rules that gain out. Stochastic filters instead keep every behavior reachable but require more attempts. A fixed-budget score mixes these two outcomes.

We contribute an evaluation framework that separates three questions: which behaviors a defense prevents, how much additional effort it imposes on an attacker, and whether the experiment has enough coverage to support its conclusions. The framework combines a conservative reachability bound, a per-behavior measure of attacker cost, and ANCHOR, a benchmark with known ground truth for validating the reachability bound and cost measure.

2 RELATED WORK

Robustness claims can fail when evaluations do not adapt to the defense. In adversarial vision, obfuscated gradients can obstruct optimization without removing adversarial examples (Athalye et al., 2018). Subsequent work formalized evaluation principles (Carlini et al., 2019), while showing that diagnostic checklists alone do not ensure soundness (Popović et al., 2022). The same issue appears in language-model safety: adaptive and defense-aware attacks can bypass systems that performed well on their original test suites (Andriushchenko et al., 2025; Nasr et al., 2025). We address the inference that remains after strengthening the attack: whether a lower score means that a behavior was removed or merely not found.

Recent jailbreak research makes attack budget increasingly explicit. Best-of-n jailbreaking exposes failures missed by a few attempts (Hughes et al., 2024); sampling-aware attacks study the effects of repeated draws (Beyer et al., 2025); and scaling analyses track success with additional queries or compute (Wang et al., 2026). Compute-aware evaluation likewise treats attack cost as a security outcome (Ehghaghi et al., 2026). We distinguish how quickly an attack succeeds, which measures cost, from which behaviors remain reachable, which measures prevention. Our contribution is a falsifiable, budget-indexed bound obtained through inexpensive replay. It can contradict a reported prevention gain without reproducing the original attack and returns a vacuous bound when probe coverage is insufficient.

Our use of *certificate* differs from conventional certified robustness. Existing safety certificates aim to prove that no allowed perturbation can elicit a restricted output (Kumar et al., 2023). Such guarantees support the defense by establishing a region in which an attack cannot succeed. Our guarantee points in the opposite direction: an observed defended-system success shows that a behavior was not removed and upper-bounds the improvement a defense may claim. Failure to obtain our certificate is not evidence of safety. Randomized perturb-and-aggregate defenses such as SMOOTHLLM (Robey et al., 2023) illustrate the distinction: randomization may reduce one-shot success while leaving a route that repetition can find.

Finally, prior work improves evaluation through stronger controls and more reliable outcome measurements. Constructed adversarial examples can reveal when an attack misses a vulnerability known to exist (Zimmermann et al., 2022). We apply the same calibration principle to defenses: negative controls remove nothing, while positive controls remove known fractions of the benchmark. We also use exact verifiers because jailbreak judges can mistake empty compliance for meaningful behavior (Souly et al., 2024) and become unreliable under adversarial distribution shift (Schwinn et al., 2026). Together, these choices isolate evaluation error from attack failure and judge error.

3 MEASURING ATTAINABLE RISK

An attack is usually given a query budget, the maximum number of attempts it may make for each restricted behavior. At any chosen budget N, we measure the fraction of behaviors the attack has reached, denoted as $\mathrm{ASR}^{\star}(N)$. This is the *finite-budget success rate*. Increasing the budget gives the attack more opportunities to succeed, so the rate generally rises and eventually levels off. We call this long-run level the *attainable ceiling*. It represents the fraction of behaviors that the attacker can ultimately reach under the evaluation conditions.

These two measurements answer different questions. The finite-budget result asks how often the attack succeeds before a particular cutoff. The ceiling asks what remains reachable if the attack is allowed to continue. We therefore use *finite-budget gain* for the reduction in success measured at a

chosen budget, and *ceiling gain* for the reduction in the long-run attainable level. The first measures immediate resistance; the second measures prevention.

The ceiling is deliberately an infinite-budget quantity, so any behavior with a nonzero per-query probability has ceiling one and a defense that only lowers that probability receives no prevention credit, however far it lowers it. That is the intended reading: such a defense has not removed the behavior. It is also why the ceiling is not reported alone. The budget-indexed bound of Section 6 answers the operational question at a budget the attacker actually has, and the cost multiplier says what the reduced probability buys; a defense taking a success rate to 10^{-20} is indistinguishable from prevention at every feasible N and is separated from it only in the limit.

A defense can improve the finite-budget result without improving the ceiling. For example, it may reduce the chance that any single attempt succeeds. The attack then needs more queries, so its success curve rises more slowly, but it may still reach the same behaviors in the end. By contrast, a defense that truly prevents some behaviors lowers the level at which the curve eventually settles. Looking at only one budget cannot tell these cases apart: the same observed score can belong to a curve that later catches up or to one that remains permanently lower. Testing more benchmark behaviors makes the score at that budget more precise, but it does not reveal how an unfinished curve will end.

We use a simple saturating curve, $\mathrm{ASR}^{\star}(N) \approx A\big(1 - e^{-N/\tau}\big)$, to summarize how success changes across the budgets we measured. The fitted endpoint A is treated as an estimate, not as proof of the true ceiling. When the data do not support the fitted curve, we do not extrapolate. Instead, we report the success rate at the largest tested budget and state that the ceiling remains unresolved.

Our analysis requires each attempt to use fresh randomness or, for a stateful defense, a reset permitted by its specification. We refer to this condition as the *repeatability assumption*. Refusals do not count as successful behavior. Query adaptations may use only public information, such as the inverse of a published input transformation; we call them *licensed*.

Formally, let S be the undefended system, S^D the system with defense D, and $\mathcal{B}$ the declared behavior set, with $B = |\mathcal{B}|$. For a system S', $\mathrm{ASR}^{\star}_{S',b}(N)$ is the probability of reaching behavior b within N attempts, $\mathrm{ASR}^{\star}_{S'}(N)$ its average over $\mathcal{B}$, and $A_{S'} = \lim_{N\to\infty} \mathrm{ASR}^{\star}_{S'}(N)$ the ceiling ($A_{S',b}$ per behavior). The gains of D are $g_N(D) = \mathrm{ASR}^{\star}_{S}(N) - \mathrm{ASR}^{\star}_{S^D}(N)$ and $g_\infty(D) = A_S - A_{S^D}$. Let $\mathcal{B}_S \subseteq \mathcal{B}$ contain the behaviors witnessed on the undefended system. We write $\widehat{\mathcal{B}}_N \subseteq \mathcal{B}_S$ for the behaviors recovered by at least one verified success within N licensed draws; when the budget is clear, we omit the subscript.

4 A Certificate of Remaining Reachability

With fresh randomness, repeated attempts make any behavior with a nonzero chance of success increasingly likely to appear. Observing a behavior on the defended system therefore shows that the defense has not made it unreachable. A defense can lower the attainable ceiling only by making some behaviors unreachable. We call a mechanism that leaves every behavior reachable an *inert defense*, or *placebo*. An exact placebo has zero gain at every budget; a ceiling placebo has zero ceiling gain but may lower finite-budget success. Such mechanisms serve as negative controls because their true ceiling gain is zero.

Proposition 1 (Public recoding changes nothing)**.** *If a defense only applies a public, reversible input transformation ϕ, an attacker can apply ϕ^{-1} before every query. The defended and undefended systems then have the same success curve at every budget.*

This proposition gives us a control with a known answer. The practical test does not need to reconstruct the complete success curve. For each behavior b, we take a query x_b that already succeeded on the undefended system. We adapt that query according to the defense's public rules, submit it to the defended system up to m times (the *probe depth*), and check each response with the behavior's verifier. We call the behavior *certified* if at least one defended response passes. Certification means only that the behavior remains reachable; it is not a certificate of safety.

Theorem 1 (Ceiling certificate)**.** *If c of the B tested behaviors are certified, the defense's average ceiling gain satisfies*

$$g_\infty(D) \le 1 - \frac{c}{B}. \tag{1}$$

The bound follows from a worst-case count. Each of the c certified behaviors is known to remain reachable, so the defense receives no prevention credit for those behaviors. To be maximally favorable to the defense, we treat every one of the remaining $B - c$ behaviors as if it had been removed completely. The largest gain consistent with the observations is therefore the fraction $(B - c)/B$, which is the bound above.

This reasoning is deliberately conservative. An uncertified behavior may still be reachable; the test may simply have missed a rare success. Failure to certify is therefore not evidence that the defense removed the behavior. Repeating the adapted query makes rare successes easier to find, but any success the test misses only gives the defense more credit and makes the bound looser. A proof appears in Appendix A.

The deterministic certificate records what has already been witnessed. To describe recovery at a finite probe budget, let p_b be the success probability of the licensed query for $b \in \mathcal{B}_S$, and let ℓ_b be simultaneous one-sided lower confidence bounds on these probabilities with family-wise coverage at least $1 - \alpha$.

Proposition 2 (Budget-indexed recovery bound)**.** *Define*

$$b_N = \frac{B - |\mathcal{B}_S| + \sum_{b \in \mathcal{B}_S} (1 - \ell_b)^N}{B}. \tag{2}$$

With probability at least $1 - \alpha$, b_N upper-bounds the expected fraction of all B behaviors left unrecovered by a fresh, independent N-query licensed probe.

Thus, b_N is a finite-budget unrecovered-fraction bound, not an estimate or certificate of ceiling gain. Its limiting floor is the fraction of behaviors not witnessed on the undefended system.

Defense class	**How the query is reused**	**What a verified success shows**
Public input recoding	reverse the published recoding	the behavior remains reachable after the recoding is undone
Stochastic filter	submit the same query again	repeated attempts leave a route through the filter
Session or query cap	start a new permitted session	the behavior remains reachable under the published reset rule
Guard or classifier	submit the same query again	a query that previously worked can still pass the guard
Safety-tuned model	submit the same query again	the behavior remains reachable after tuning
System-prompt defense	submit the same query again	the behavior remains reachable under the policy prompt
Secret keyed transform	*not applicable*	no public adaptation is available, so the certificate does not apply

Table 1: Permitted adaptations for common defense classes. "Submit the same query again" means that no input transformation is needed. Every adaptation uses only the defense's public rules; the method is out of scope when a secret key is required.

4.1 Query Adaptation and Attacker Cost

The licensed adaptation depends on the defense. For a public recoding, we reverse the recoding. For a session cap, we reset the session when the specification allows it. For most filters, prompts, and tuned models, we replay the successful query unchanged. Secret transformations are outside the scope of the certificate. Table 1 summarizes the licensed query for each defense class.

The same queries also show whether the defense slows the attacker. Under the repeatability assumption, draws are independent, so a licensed query that succeeds with probability p_b needs $1/p_b$ queries in expectation and the quantity of interest is the ratio $p_b^{\mathrm{undef}}/p_b^{\mathrm{def}}$. We report its median and quartiles over behaviors that remain reachable, and call that the *cost multiplier*. A multiplier of $1.00\times$ indicates no measurable delay; a multiplier of $5.00\times$ means the attacker needs about five times as many queries. The multiplier measures additional effort; it does not show that any behavior was prevented.

A ratio of *aggregate* success rates, $\overline{p}^{\mathrm{undef}}/\overline{p}^{\mathrm{def}}$, is not a substitute, because $\mathbb{E}[1/p] \neq 1/\mathbb{E}[p]$ unless the defense scales every behavior's probability by the same factor. The two agree exactly for our stochastic filters, which do scale uniformly, and disagree for a defense that removes behaviors instead of slowing them: Section 6.2 shows the aggregate ratio reporting up to $20\times$ for a control whose per-behavior multiplier is $1.00\times$ on everything it leaves reachable. Behaviors whose licensed query never succeeds within m draws are *censored*: m draws bound their cost from below and not from above, so they are counted and reported separately rather than averaged in.

Finally, an experiment can stop before a slow success curve has leveled off. We therefore report two resolution diagnostics: a reference-effect interval width based on the number of behaviors and the unresolved headroom at the largest query budget. These diagnostics indicate whether an effect of roughly 0.10 is resolvable; they are not minimum detectable effects.

5 Experimental Design

5.1 The Anchor Testbed

Real harmful-content benchmarks do not provide a known robustness answer, and their judges may be noisy. ANCHOR instead uses three benign tasks with exact answers: reproducing a nonce, multiplying two two-digit integers, and sorting a short word list. A system prompt forbids each task. We first remove that policy and keep only tasks the model can solve reliably, so a later failure can be attributed to the defense rather than missing capability.

The nonce task asks the model to reproduce a randomly generated string from the context. The multiplication task asks for the product of two two-digit integers. The sorting task asks for a comma-separated alphabetical ordering of a short word list. Each verifier normalizes only the formatting needed for its task and then checks exact equality or exact substring occurrence.

System	Type	Mechanism	True gain
REPAR	exact placebo	Public bijective input recoding	$g_N = 0$
THROTTLE	exact placebo	Per-session cap with free reset	$g_N = 0$
PREFIXBLOCK	ceiling placebo	Blocks one fixed output prefix	$g_\infty = 0$
SCORENOISE	ceiling placebo	Changes scores but not outputs	$g_\infty = 0$
STOCHFILTER	ceiling placebo	Independently blocks each query with probability p	$g_\infty = 0$
VERUM-p	positive control	Fully blocks a random p fraction of behaviors	pA_S
NULL	reference	No defense	0

Table 2: Controls used to calibrate the evaluation pipeline. A_S denotes the undefended ceiling. The runs instantiate STOCHFILTER at $p \in \{0.5, 0.8\}$ and VERUM-p at $p \in \{0.2, 0.4, 0.6, 0.8, 0.95, 1\}$. The stochastic filters are the class closest to deployed practice; REPAR is the extreme case used for calibration.

The controls make evaluation errors observable. REPAR is a public, reversible recoding and must have zero gain. THROTTLE, PREFIXBLOCK, SCORENOISE, and two stochastic filters also leave at least one route to every behavior. The VERUM controls genuinely block known fractions of behaviors.

5.2 Models, Attacks, and Estimation

The main experiment uses Qwen2.5-7B-Instruct and 223 screened behaviors. We run five seeds at eleven budgets from 1 to 1024. The attack instruments are unguided resampling, resampling that is told how to reverse REPAR, and best-of-n sampling at its released configuration (Hughes et al., 2024). Results are never pooled across instruments. We also test the certificate on Qwen2.5-14B-Instruct and Mistral-7B-Instruct-v0.3.

Budgets follow $N \in \{1, 2, 4, \ldots, 1024\}$. We take $N_0 = 32$ as the reported budget, a typical fixed-budget setting, and write g_{32} for the gain measured there. The saturating curve is fit only as a descriptive summary, with a goodness-of-fit test and an isotonic fallback when the fit is rejected; bootstrap intervals resample whole behaviors, paired across arms (Appendix B.1). The certificate

itself needs no curve fit. It replays licensed probes, checks exact outputs, and counts certified behaviors.

6 RESULTS

6.1 A KNOWN PLACEBO RECEIVES A LARGE GAIN

The undefended Qwen2.5-7B-Instruct model reaches 215 of 223 screened behaviors, an empirical ceiling of 0.964. Against this baseline, the ordinary resampling sweep credits REPAR with a ceiling gain of 0.540 (95% interval $[0.477, 0.596]$). Its fitted curve is rejected ($p = 0.0026$), so 0.540 should not be interpreted as a ceiling estimate. The preregistered fallback reports 0.425 at 1,024 queries; because the curve is still rising, this is a finite-budget result rather than evidence of permanent prevention. Since REPAR is reversible, its true ceiling gain is zero, and the reachability certificate rules out both reported values as prevention gains. Figure 1 shows how REPAR can appear stronger at a fixed budget than a control that genuinely removes behaviors while its curve continues toward the undefended result.

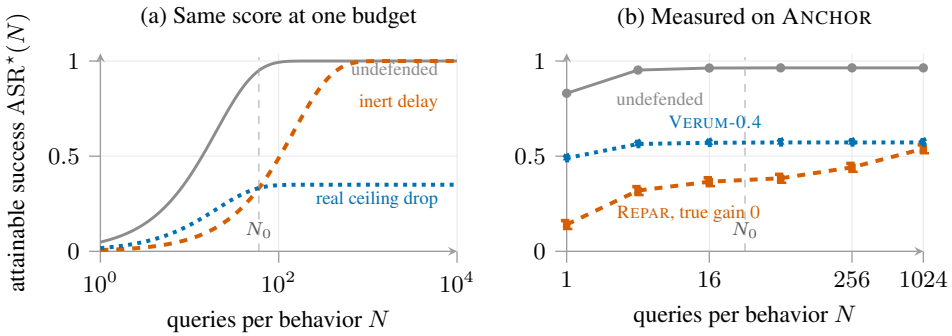

Figure 1: **A fixed-budget score can confuse delay with prevention.** Both panels plot attainable success against queries per behavior; the dashed vertical line marks the reported budget N_0. Panel (a) is schematic: two defenses with almost the same success rate at N_0, where the inert one reaches the undefended ceiling later and the genuine one has a lower ceiling. Panel (b) is measured. At $N = 32$ the inert REPAR mechanism appears stronger than VERUM-0.4, even though REPAR's curve continues to rise and its true ceiling gain is zero.

The licensed REPAR query succeeds on all 215 behaviors reached by the undefended system. The certificate therefore gives $1 - 215/223 = 0.036$ as the largest possible gain. The count is already complete with eight attempts per behavior, or 1,784 defended-system queries. A full specification-aware sweep reaches the same behaviors and estimates zero gain, but at far higher cost. Counting, for each behavior, the budget at which it first succeeds (or the maximum budget if it never does), the sweep uses 4,883,869 defended-system queries, about 2,738 times as many as the certificate, to reach the same conclusion.

6.2 SOME DEFENSES BUY DELAY, NOT PREVENTION

REPAR, THROTTLE, PREFIXBLOCK, and SCORENOISE all have a cost multiplier of 1.00×. They neither remove behaviors nor measurably slow the attacker. The stochastic filters leave the ceiling unchanged but increase expected work by 1.99× and 5.00× per behavior. Reporting the multiplier reveals the benefit that a ceiling-only score would miss.

The VERUM controls show why this has to be measured per behavior. Each one removes a known fraction of the benchmark and leaves the rest untouched, so the correct statement is that it costs the attacker nothing on whatever it does not remove. The per-behavior multiplier says exactly that: 1.00× for VERUM-0.2 through VERUM-0.95, with 43, 85, 136, 187 and 204 of the 215 behaviors censored. The aggregate ratio instead reports 1.24×, 1.66×, 2.78×, 8.12× and 20.17×, rising with the fraction removed. The aggregate ratio increases because removed behaviors lower the defended average, not because reaching the remaining behaviors has become more expensive. The two estimators coincide for the stochastic filters only because a filter scales every behavior by the same

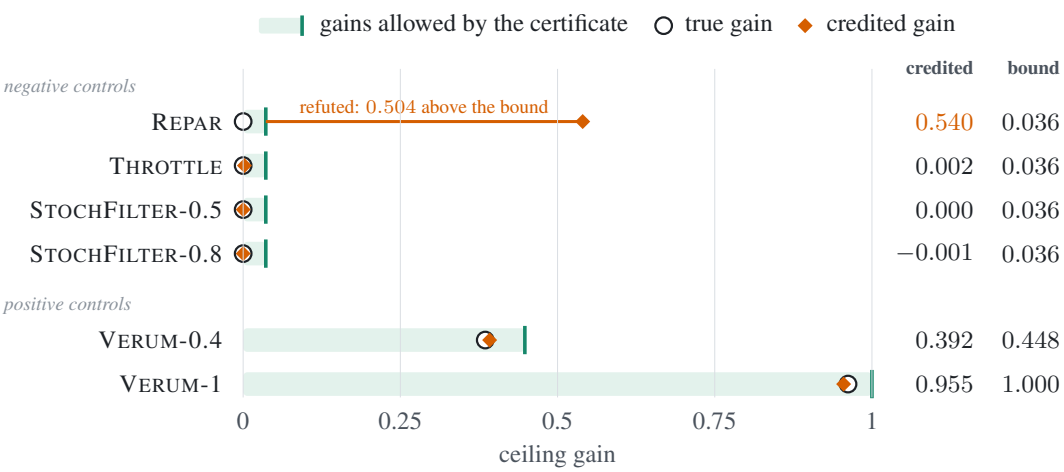

Figure 2: **The certificate refutes the credited gain for REPAR.** Each green band runs from zero to the certified upper bound, the largest ceiling gain the certificate allows. The bound is sound for every system: each true gain (circle) lies inside its band. A credited gain (diamond) outside the band is refuted. Only REPAR's credited gain falls outside, by 0.504. Bounds come from the standalone certificate probe ($m = 135$), which certifies 123 of 223 behaviors for VERUM-0.4.

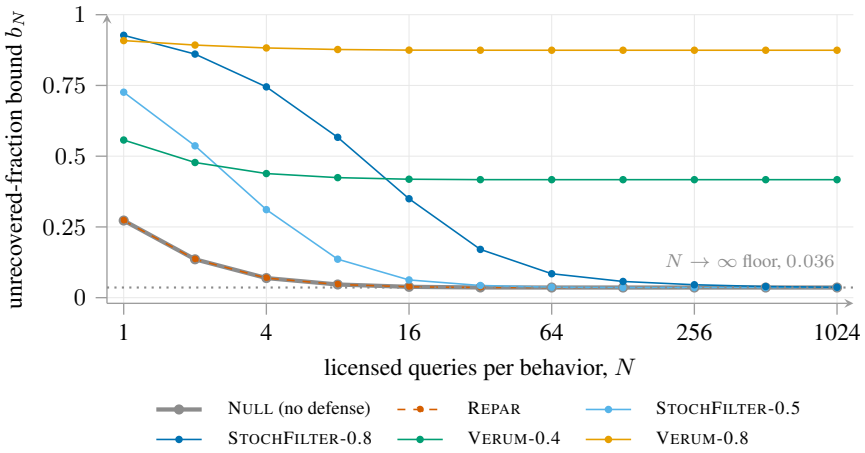

Figure 3: **A defense can look protective purely because the budget is small.** b_N upper-bounds the expected fraction of all behaviors left unrecovered by a fresh N-query licensed probe. REPAR traces NULL at every budget. The stochastic filters separate from NULL at small N and approach the same floor more slowly, showing delay rather than removal. The VERUM controls remain above the floor because they remove behaviors.

factor. All multipliers come from the same $m = 135$ licensed draws as the certificate, restricted to the 215 witnessed behaviors; VERUM-1 censors all 215. Censored behaviors are counted separately rather than averaged in, because an unreached behavior is a censored observation, not an infinite cost.

A stronger attack does not automatically fix the problem. Best-of-n raises the undefended fitted ceiling from 0.962 to 0.992, yet it credits REPAR with an even larger gain of 0.636. It also credits the two stochastic filters with gains of 0.104 and 0.214. Because the filters remove candidates before best-of-n selects among them, the defended curves rise more slowly and remain unfinished when the sweep stops. The apparent gains track delay, not lost reachability.

Table 3 reports every negative and positive control under resampling. Under the budget-indexed certificate the two stochastic filters bound at 0.726 and 0.927 at $N = 1$ and reach the 0.036 floor only by $N = 128$ and $N = 1024$ (Figure 3), so their apparent protection is a property of the query

budget. Per-behavior cost multipliers, aggregate ratios and censoring counts for every system are in Table 4 (Appendix B.2).

System	True gain	g_{32}	$\hat{g}_\infty$	95% CI	b_1	b_{32}
REPAR	0	0.589	0.540	[0.477, 0.596]	0.273	0.036
THROTTLE	0	0.003	0.002	[0.000, 0.006]	0.273	0.036
PREFIXBLOCK	0	0.000	0.000	[0.000, 0.001]	0.273	0.036
SCORENOISE	0	0.001	0.000	[0.000, 0.001]	0.273	0.036
STOCHFILTER-0.5	0	0.001	−0.000	[−0.001, 0.001]	0.726	0.043
STOCHFILTER-0.8	0	0.008	−0.001	[−0.002, 0.001]	0.927	0.171
VERUM-0.2	0.192	0.181	0.182	[0.159, 0.204]	0.407	0.229
VERUM-0.4	0.385	0.392	0.392	[0.360, 0.424]	0.557	0.417
VERUM-0.6	0.577	0.581	0.580	[0.549, 0.615]	0.735	0.646
VERUM-0.8	0.770	0.770	0.769	[0.740, 0.798]	0.909	0.874
VERUM-0.95	0.914	0.909	0.906	[0.879, 0.932]	0.962	0.951
VERUM-1	0.962	0.964	0.955	[0.928, 0.978]	—	—

Table 3: Results from the unguided resampling sweep on Qwen2.5-7B-Instruct. $\hat{g}_\infty$ is the difference between fitted ceilings. REPAR's parametric fit is rejected; the preregistered fallback reports 0.425. b_N is the budget-indexed unrecovered-fraction bound from Eq. 2 ($\alpha = 0.05$, Bonferroni over 215 behaviors); it is not a ceiling-gain estimate. VERUM-1 witnesses no behavior, so its bound is vacuous (—).

6.3 CONTROLS, TRANSFER, AND COVERAGE

The ceiling certificate is conservative for every positive control, and the standard pipeline recovers their constructed gains within its bootstrap intervals. The reference-effect width is 0.074; REPAR's remaining headroom at the final budget is 0.423, confirming that the sweep is too shallow to identify its ceiling.

The certificate transfers without repeating the full sweep. It certifies every witnessed behavior on Qwen2.5-14B-Instruct and Mistral-7B-Instruct-v0.3, so their *conditional* bounds (Eq. 1 over $\mathcal{B}_S$ only) are zero. The unconditional bound is looser on the 14B model because the original attack reached only 146 of 232 behaviors. This difference comes from attack coverage, not from the defense. Table 7 in Appendix B.4 lists every setting, including a wrong-transform falsification check and a failure case on JailbreakBench behaviors, where low attack coverage leaves the bound vacuous.

6.4 FOUR PUBLISHED DEFENSES

The constructed controls establish that the certificate recovers known answers. We next test whether it remains informative for defenses developed independently of our evaluation. Using the same 223 behaviors and $m = 135$ licensed draws, we evaluate four published, task-independent defenses whose mechanisms do not use behavior-specific labels: SMOOTHLLM (Robey et al., 2023), paraphrase preprocessing (Jain et al., 2023), system-mode self-reminder (Xie et al., 2023), and in-context defense (ICD) (Wei et al., 2023). None requires an attacker-side input transformation, so the licensed adaptation is the identity: the evaluator replays the previously successful query unchanged. The procedure runs end to end without defense-specific modifications and produces a non-vacuous upper bound on prevention for every defense, ranging from 0.166 to 0.682 (Table 5).

The bounds are not intended as standalone refutations: that requires comparing a bound with a prevention gain claimed on the same behavior suite. Instead, they provide an interpretable and reusable baseline for future comparisons. They also show how effectively the available probe budget resolves each defense. For the paraphrase and self-reminder defenses, the results are highly stable. Certified behaviors succeed a median of 105 and 89 times out of 135, respectively, and fewer than 2% succeed twice or fewer. The small near-threshold population suggests that 0.166 and 0.341 are comparatively stable at the tested probe depth, although uncertified behaviors remain unresolved.

For SMOOTHLLM and the in-context defense, the certificate reveals a different and informative structure. Most certified behaviors are recovered consistently, with median counts of 134 and 135 out of 135, while 17.2% and 9.9% are observed only once or twice. This rare-success tail identifies where additional evaluation effort may be most valuable: increasing the probe budget may recover more behaviors and tighten the current bounds.

A budget sweep on the same four defenses reproduces this split independently. On a separate draw of 184 screened behaviors, five seeds, unguided resampling to $N = 1024$, the two defenses whose bounds are close to tight are already saturated at the reported budget: the paraphrase defense gains $+0.017$ in attainable success between $N = 32$ and $N = 1024$ and the self-reminder $+0.004$. The two whose bounds are probe-limited are still climbing at the end of the sweep, by $+0.205$ and $+0.260$ (Table 6). The replay depth of the certificate and the query budget of a sweep are different measurements on different behaviors, so their agreement on which defenses have run out of reachable behaviors and which have not is evidence about the defenses rather than about either procedure.

7 DISCUSSION

Our results support separating three claims that are often reported as a single robustness score. A lower success rate at a fixed budget establishes resistance within that budget, but not necessarily prevention. Evaluations should therefore report three quantities separately: an upper bound on prevention, the added attacker cost, and the experiment's resolution, so it is clear which conclusions the evidence supports.

The method provides one-sided evidence. A verified success proves that a behavior was not prevented, whereas failure to recover it does not prove that it was removed. Stronger attacks can add successful witnesses and tighten the resulting bound. The bound should therefore be interpreted as a limit on the prevention a defense may claim, not as a complete measure of robustness.

8 LIMITATIONS

The certificate depends on probe coverage and verifier accuracy. Weak undefended coverage produces loose bounds, as in Appendix B.5, while false positives can incorrectly certify reachability. Our full sweep covers one model and attack paradigm; the transfer experiments validate the certificate but not the curve fit. We evaluate four published defenses, but without matched pipeline gains we neither refute them nor estimate how common the measurement failure is. Two defenses show rare-success tails at $m = 135$, so additional draws may tighten their bounds.

Finally, repeated-query conclusions require fresh randomness or a permitted reset. Under deterministic decoding, an observed success still proves reachability, but identical repetitions provide no additional evidence.

9 CONCLUSION

Most jailbreak evaluations ask whether an attack succeeds within a fixed query budget. A lower success rate can mean that the defense removed some behaviors, but it can also mean that the attacker needs more attempts to reach the same behaviors. Treating both outcomes as the same robustness improvement obscures what the defense actually achieved. We introduced a certificate that uses successful undefended queries and the defense's public rules to test which behaviors remain reachable. Verified defended successes place a conservative upper bound on the prevention a defense may claim, while the same trials measure the additional work imposed on the attacker. On ANCHOR, this approach rejects the apparent improvement of a reversible recoding and distinguishes stochastic filters that delay success from controls that genuinely remove behaviors.

Defense evaluations should therefore report three conclusions separately: the evidence that behaviors were prevented, the increase in attacker cost, and the resolution supported by the tested budget and sample size. A lower success rate at one budget is useful evidence of resistance, but it should not be presented as evidence of prevention unless the evaluation can support that stronger claim.

ETHICS STATEMENT

This work studies evaluation methods and introduces no new jailbreak technique. The ANCHOR testbed uses benign tasks with exact verifiers and can be released without exposing harmful content. The harmful-behavior check uses existing public benchmarks and attack settings. We report upper bounds on the possible prevention achieved by four published defenses, but do not identify any of them as ineffective: an upper bound alone cannot establish that conclusion, and we did not run the per-defense harness audits it would require. The study uses no human-subject or personal data.

REPRODUCIBILITY STATEMENT

The supplementary material contains the testbed, defense implementations, attack configurations, per-run outputs, and scripts that regenerate the reported tables. It includes 415 sweep files, one for each system, instrument, and seed combination. A reconciliation script checks all values that can be regenerated from these files and reports 249 passed checks, 0 failures, and 0 unchecked values. Seeds are pooled only within an instrument. Confidence intervals resample whole behaviors and keep the undefended and defended arms paired. The certificate counts and the cost multipliers in Table 4 are emitted by scripts that join reachability on behavior identifier rather than on position.

USE OF AI STATEMENT

Generative AI tools assisted with the conceptual framework, mathematical claims, proof drafting, experimental design, code, and writing. They did not generate any experimental result. The authors reviewed the proofs, code, citations, and text and take responsibility for the final paper. Numerical checks were used to catch implementation and transcription errors, but they are not treated as proofs.

REFERENCES

Maksym Andriushchenko, Francesco Croce, and Nicolas Flammarion. Jailbreaking leading safety-aligned LLMs with simple adaptive attacks. In *International Conference on Learning Representations (ICLR)*, 2025. arXiv:2404.02151.

Anish Athalye, Nicholas Carlini, and David Wagner. Obfuscated gradients give a false sense of security: Circumventing defenses to adversarial examples. In *International Conference on Machine Learning (ICML)*, pp. 274–283, 2018.

Tim Beyer, Yan Scholten, Leo Schwinn, and Stephan Günnemann. Sampling-aware adversarial attacks against large language models. *arXiv preprint arXiv:2507.04446*, 2025.

Nicholas Carlini, Anish Athalye, Nicolas Papernot, Wieland Brendel, Jonas Rauber, Dimitris Tsipras, Ian Goodfellow, Aleksander Madry, and Alexey Kurakin. On evaluating adversarial robustness. *arXiv preprint arXiv:1902.06705*, 2019.

Patrick Chao, Edoardo Debenedetti, Alexander Robey, Maksym Andriushchenko, Francesco Croce, Vikash Sehwag, Edgar Dobriban, Nicolas Flammarion, George J. Pappas, Florian Tramèr, Hamed Hassani, and Eric Wong. JailbreakBench: An open robustness benchmark for jailbreaking large language models. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2024.

Malikeh Ehghaghi, Boglárka Ecsedi, Marsha Chechik, and Colin Raffel. Risk under pressure: Compute-aware evaluation of adversarial robustness in language models. *arXiv preprint arXiv:2606.11409*, 2026.

John Hughes, Sara Price, Aengus Lynch, Rylan Schaeffer, Fazl Barez, Sanmi Koyejo, Henry Sleight, Erik Jones, Ethan Perez, and Mrinank Sharma. Best-of-N jailbreaking. *arXiv preprint arXiv:2412.03556*, 2024.

Neel Jain, Avi Schwarzschild, Yuxin Wen, Gowthami Somepalli, John Kirchenbauer, Ping-yeh Chiang, Micah Goldblum, Aniruddha Saha, Jonas Geiping, and Tom Goldstein. Baseline defenses for adversarial attacks against aligned language models. *arXiv preprint arXiv:2309.00614*, 2023.

Aounon Kumar, Chirag Agarwal, Suraj Srinivas, Aaron Jiaxun Li, Soheil Feizi, and Himabindu Lakkaraju. Certifying LLM safety against adversarial prompting. *arXiv preprint arXiv:2309.02705*, 2023.

Mantas Mazeika, Long Phan, Xuwang Yin, Andy Zou, Zifan Wang, Norman Mu, Elham Sakhaee, Nathaniel Li, Steven Basart, Bo Li, David Forsyth, and Dan Hendrycks. HarmBench: A standardized evaluation framework for automated red teaming and robust refusal. In *International Conference on Machine Learning (ICML)*, 2024.

Milad Nasr, Nicholas Carlini, Chawin Sitawarin, Sander V. Schulhoff, Jamie Hayes, Michael Ilie, Juliette Pluto, Shuang Song, Harsh Chaudhari, Ilia Shumailov, Abhradeep Thakurta, Kai Yuanqing Xiao, Andreas Terzis, and Florian Tramèr. The attacker moves second: Stronger adaptive attacks bypass defenses against LLM jailbreaks and prompt injections. *arXiv preprint arXiv:2510.09023*, 2025.

Nikola Popović, Danda Pani Paudel, Thomas Probst, and Luc Van Gool. Gradient obfuscation checklist test gives a false sense of security. *arXiv preprint arXiv:2206.01705*, 2022.

Alexander Robey, Eric Wong, Hamed Hassani, and George J. Pappas. SmoothLLM: Defending large language models against jailbreaking attacks. *arXiv preprint arXiv:2310.03684*, 2023.

Leo Schwinn, Moritz Ladenburger, Tim Beyer, Mehrnaz Mofakhami, Gauthier Gidel, and Stephan Günnemann. A coin flip for safety: LLM judges fail to reliably measure adversarial robustness. *arXiv preprint arXiv:2603.06594*, 2026.

Alexandra Souly, Qingyuan Lu, Dillon Bowen, Tu Trinh, Elvis Hsieh, Sana Pandey, Pieter Abbeel, Justin Svegliato, Scott Emmons, Olivia Watkins, and Sam Toyer. A StrongREJECT for empty jailbreaks. In *NeurIPS Datasets and Benchmarks Track*, 2024.

Xiangwen Wang, Ananth Balashankar, and Varun Chandrasekaran. Systematic scaling analysis of jailbreak attacks in large language models. *arXiv preprint arXiv:2603.11149*, 2026.

Zeming Wei, Yifei Wang, Ang Li, Yichuan Mo, and Yisen Wang. Jailbreak and guard aligned language models with only few in-context demonstrations. *arXiv preprint arXiv:2310.06387*, 2023.

Yueqi Xie, Jingwei Yi, Jiawei Shao, Justin Curl, Lingjuan Lyu, Qifeng Chen, Xing Xie, and Fangzhao Wu. Defending ChatGPT against jailbreak attack via self-reminders. *Nature Machine Intelligence*, 5:1486–1496, 2023. doi: 10.1038/s42256-023-00765-8.

Roland S. Zimmermann, Wieland Brendel, Florian Tramèr, and Nicholas Carlini. Increasing confidence in adversarial robustness evaluations. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

A Proofs

A.1 One-Budget Non-Identification

Observation 1 (One budget does not identify the ceiling)**.** *Let the defended success curve have the saturating form $A(1 - e^{-N/\tau})$ with unknown ceiling $A \in [0, A_S]$ and rate $\tau > 0$, and suppose a single measurement at budget N_0 observes success rate v. Then every ceiling gain in $[0,\ A_S - v)$ is consistent with that measurement.*

Proof. The defended curve is nondecreasing, so its ceiling satisfies $A_{S^D} \geq v$ and $g_\infty(D) \leq A_S - v$. For any proposed ceiling $A' \in (v, A_S]$, choose

$$\tau' = -\frac{N_0}{\log(1 - v/A')}. \tag{3}$$

Then $A'(1 - e^{-N_0/\tau'}) = v$. As A' ranges from just above v to A_S, the corresponding gain $A_S - A'$ ranges over $[0, A_S - v)$. One observation cannot distinguish these curves. □

A.2 Public Reparameterization

Proof of Proposition 1. Take any attacker against S. Construct an attacker against S^D that applies ϕ^{-1} to every query and passes each returned output back to the original attacker. Because

$$S^D(\phi^{-1}(x)) \stackrel{d}{=} S(x), \tag{4}$$

the two attackers see the same transcript distribution and have the same success probability. This gives $\mathrm{ASR}^\star_{S^D}(N) \geq \mathrm{ASR}^\star_S(N)$. Repeating the argument with ϕ and ϕ^{-1} exchanged gives the reverse inequality. □

A.3 Ceiling Certificate

Proof of Theorem 1. For a certified behavior b, an observed undefended success shows that some query has success probability $q_b > 0$. Under the repeatability assumption, repeating it succeeds with probability $1 - (1 - q_b)^N \to 1$, so $A_{S,b} = 1$. The observed defended success gives $A_{S^D,b} = 1$ by the same argument. Thus the gain is zero for every certified behavior.

Average the per-behavior gains. Certified behaviors contribute zero. Any uncertified behavior contributes at most one because ceilings lie in $[0, 1]$. Therefore

$$g_\infty(D) \leq 1 - \frac{c}{B}. \tag{5}$$

□

A.4 Budget-Indexed Recovery Bound

Proof of Proposition 2. On the simultaneous-coverage event, every witnessed behavior satisfies $p_b \geq \ell_b$. A fresh independent N-query probe therefore misses behavior b with probability at most $(1 - \ell_b)^N$. We conservatively count every behavior outside $\mathcal{B}_S$ as unrecovered. Averaging these bounds over all B behaviors yields Eq. 2. □

A.5 Stochastic Filters

Proposition 3 (A leaky filter does not lower the ceiling)**.** *Suppose there is an input that produces behavior b and passes the filter with probability $q > 0$ on each independent query. Then*

$$\mathrm{ASR}^\star_{S^D,b}(N) \geq 1 - (1 - q)^N, \tag{6}$$

so $A_{S^D,b} = 1$.

Proof. Repeat the input N times. The probability that all independent attempts fail is at most $(1 - q)^N$, which approaches zero. □

A.6 Reference-Effect Resolution

At a saturated horizon, a two-sided normal approximation gives the width term

$$\gamma_{\mathrm{width}} = (z_{1-\alpha/2} + z_{1-\beta})\sqrt{\frac{A(1-A) + (A - \gamma_{\mathrm{ref}})(1 - A + \gamma_{\mathrm{ref}})}{n}}, \tag{7}$$

with $\gamma_{\mathrm{ref}} = 0.10$. We use $\alpha = 0.05$ and power 0.80. This evaluates the interval half-width at a *reference* effect of 0.10; it does not solve for the minimum detectable effect, which is the γ satisfying $\gamma = \gamma_{\mathrm{width}}(\gamma)$. The reported value is therefore a reference-effect width, and we use it only to say whether a gain of roughly 0.10 would have been resolvable at this n. This gives 0.074 for $A = 0.962$ and $n = 223$, and 0.0604 for $A = 0.9923$.

For a saturating curve, the unresolved headroom at the largest budget is

$$\gamma_{\mathrm{depth}} = A_S e^{-N_{\max}/\tau_{S^D}}. \tag{8}$$

This is the gap between the unchanged ceiling and the defended curve at $N_{\max}$.

B Additional Experimental Details and Results

B.1 Estimation Details

We fit the saturating curve by inverse-variance-weighted least squares. The budget points are *nested* rather than independent: a behavior that first succeeds at budget N is counted as succeeding at every larger budget, which is what makes each curve monotone by construction. Successive points therefore share behaviors and are positively correlated, the weights are not the independent-binomial weights the formula would use, and the goodness-of-fit statistic does not have its nominal null distribution. We report the fit as a descriptive summary for this reason, and the certificate, which needs no fit, carries the claims. A Hosmer–Lemeshow-style test compares predicted and observed counts across budget bins. If rejected, isotonic regression provides the measured endpoint value; we do not treat that endpoint as an identified ceiling.

Bootstrap intervals resample behaviors rather than individual queries because queries within a behavior are dependent. The same resampled behaviors are used in the undefended and defended arms.

B.2 Attacker Cost per Behavior

Table 4 gives the per-behavior cost multiplier, the aggregate ratio, and the censoring count for every system discussed in Section 6.2.

The controls separate three outcomes. REPAR through SCORENOISE neither remove behaviors nor increase measured cost. The stochastic filters increase per-behavior cost without censoring any behavior. The VERUM controls remove behaviors outright: their per-behavior multiplier remains $1.00\times$ on what is still reachable, while the aggregate ratio rises with the censored fraction. This divergence is why the analysis uses the per-behavior statistic.

System	Aggregate ratio	Cost multiplier	IQR	Censored
REPAR	$1.00\times$	$1.00\times$	$[1.00, 1.00]$	$0/215$
THROTTLE	$1.00\times$	$1.00\times$	$[1.00, 1.00]$	$0/215$
PREFIXBLOCK	$1.00\times$	$1.00\times$	$[1.00, 1.00]$	$0/215$
SCORENOISE	$1.00\times$	$1.00\times$	$[1.00, 1.00]$	$0/215$
STOCHFILTER-0.5	$1.99\times$	$1.99\times$	$[1.86, 2.14]$	$0/215$
STOCHFILTER-0.8	$5.02\times$	$5.00\times$	$[4.49, 5.87]$	$0/215$
VERUM-0.2	$1.24\times$	$1.00\times$	$[1.00, 1.00]$	$43/215$
VERUM-0.4	$1.66\times$	$1.00\times$	$[1.00, 1.00]$	$85/215$
VERUM-0.6	$2.78\times$	$1.00\times$	$[1.00, 1.00]$	$136/215$
VERUM-0.8	$8.12\times$	$1.00\times$	$[1.00, 1.00]$	$187/215$
VERUM-0.95	$20.17\times$	$1.00\times$	$[1.00, 1.00]$	$204/215$

From the same $m = 135$ licensed draws as the certificate, on the 215 witnessed behaviors. *Aggregate ratio* is $\overline{p}^{\text{undef}}/\overline{p}^{\text{def}}$. *Cost multiplier* is the median of $p_b^{\text{undef}}/p_b^{\text{def}}$ over behaviors still reachable under the defense. *Censored* counts behaviors whose licensed query never succeeded; such behaviors are excluded because m draws provide only a lower bound on their cost.

Table 4: Attacker-cost estimates and censoring by control.

B.3 Published Defenses

Table 5 gives the certificate on the four published defenses of Section 6.4. All runs use Qwen2.5-7B-Instruct, $m = 135$ licensed draws per behavior, and undefended coverage 0.964. None of the four defenses transforms the input, so each licensed query is the original successful query, replayed unchanged.

Bound is the largest average ceiling gain consistent with the observations. It limits how much prevention a defense can claim and is not an estimate of that prevention; an uncertified behavior is not evidence that the defense removed it. The column k counts how many of the m draws succeeded for each certified behavior. A large median k with few behaviors at $k \leq 2$ means the defense separates cleanly at this probe depth and the bound is close to tight, as for paraphrase and self-reminder. A

large share at $k \leq 2$ means certified behaviors sit at the edge of what m resolves, so deeper probing could tighten the bound, as for SMOOTHLLM and the in-context defense. No standard-pipeline gain is available for these defenses on this behavior suite, so none is refuted or confirmed here.

Defense	**Certified**	**Uncertified**	**Bound**	**Median k**	$k \leq 2$
Paraphrase (Jain et al., 2023)	186/223	37	0.166	105	0.5%
Self-reminder (Xie et al., 2023)	147/223	76	0.341	89	1.4%
SMOOTHLLM (Robey et al., 2023)	121/223	102	0.457	134	17.2%
In-context defense (Wei et al., 2023)	71/223	152	0.682	135	9.9%

Table 5: The certificate on four published defenses (Qwen2.5-7B-Instruct, $m = 135$). *Bound* is an upper limit on average ceiling gain; k is the number of successful licensed draws per certified behavior.

Defense	$\mathrm{ASR}^\star(1)$	$\mathrm{ASR}^\star(32)$	$\mathrm{ASR}^\star(1024)$	$\Delta_{32\to1024}$
Paraphrase	0.601	0.859	0.876	+0.017
Self-reminder	0.489	0.647	0.651	+0.004
SMOOTHLLM	0.362	0.428	0.633	+0.205
In-context defense	0.249	0.290	0.550	+0.260

Table 6: Budget sweep on the four published defenses: a separate draw of 184 screened behaviors, five seeds, unguided resampling. $\Delta_{32\to1024}$ is the change in attainable success from the reported budget to $N = 1024$. The two defenses with close-to-tight certificate bounds are saturated by $N = 32$; the two with rare-success tails are still rising.

B.4 TRANSFER AND FALSIFICATION CHECKS

Table 7 reports the certificate on the main model, the two transfer models, a deliberately wrong transform, and the harmful-behavior check.

Setting	**Model**	$\lvert\mathcal{B}_S\rvert$	$\lvert\widehat{\mathcal{B}} \cap \mathcal{B}_S\rvert$	**Uncond.**	**Cond.**
Main	Qwen2.5-7B-Instruct	215	215	0.036	0.000
Transfer	Qwen2.5-14B-Instruct	146	146	0.371	0.000
Transfer	Mistral-7B-Instruct-v0.3	142	142	0.000	0.000
Wrong transform	Qwen2.5-7B-Instruct	215	80	0.641	0.628
Harmful (JBB)	Qwen2.5-7B-Instruct	8	0	1.000	1.000

Table 7: Certificate bounds on all behaviors (unconditional) and on the witnessed set $\mathcal{B}_S$ (conditional). The wrong-transform row is a falsification check. The JailbreakBench row is a coverage demonstration and not a certificate: with 8 of 50 behaviors witnessed and none certified, the bound is vacuous, which is the correct output for a sweep that never reached the benchmark (Appendix B.5).

B.5 FAILURE CASE ON HARMFUL BEHAVIORS

On 50 JailbreakBench behaviors (Chao et al., 2024), the unguided attack reaches only eight behaviors, and none of the licensed recoded queries pass the HarmBench classifier (Mazeika et al., 2024). The certificate correctly returns the uninformative bound 1.000. Low base-model coverage and a noisy classifier prevent a useful conclusion. A stronger attack and reliable verification are required before this method can support claims on harmful behaviors.