Reset-and-Discard (ReD) Improves Coverage at every Budget under Inference Power-Law Scaling

author names withheld

Under Review for the Workshop on High-dimensional Learning Dynamics, 2026

Abstract

The performance of large language models (LLMs) on verifiable tasks is usually measured by pass@k, the probability of answering a question correctly at least once in k trials. At a fixed budget across a workload of many tasks, a more suitable metric is *coverage@cost*: the expected number of unique questions answered as a function of total attempts. We connect these metrics via renewal theory and show that the empirically-observed power-law scaling of pass@k (with exponent $0 < \alpha < 1$) leads to sublinear (diminishing-returns) growth of coverage@cost under standard solve-to-completion allocation. We propose **Reset-and-Discard (ReD)**, a cross-problem allocation policy that provably restores linear coverage growth and maximizes coverage@cost at every budget, even under imperfect verifiers. ReD also provides a statistically efficient method to estimate inference power-law exponents when large k pass@k measurements are expensive. Experiments across three LLMs and three benchmarks show large reductions in required attempts, tokens, and USD cost.

1. Introduction

Generating multiple independent candidate solutions is a standard approach to improve LLM performance on verifiable tasks such as coding, math, and formal proof search. This is quantified by pass@k, the probability of at least one success in k attempts on a single problem [1–9]. Across model families and benchmarks, it can yield striking gains: relatively small models can approach or even exceed the single-sample performance of larger models when allowed many attempts [2, 4, 10, 11]. Recent work has uncovered a remarkably regular empirical pattern: for many tasks, $1 - \text{pass@}k$ decays as a power-law in k across orders of magnitude, with exponent α [12–14].

In practice, inference compute is often shared across a workload: batch evaluations with fixed token caps, automated code-repair systems, or service providers with cost constraints. Yet, efficient *allocation* policies under a fixed budget across *many* tasks remain underexplored. The relevant objective is then not pass@k on a single problem, but *how many distinct problems are solved* under a fixed global budget. To formalize this objective we adopt coverage@cost: the expected number of unique problems solved after t total attempts.

A common implicit policy is *solve-to-completion*: repeatedly sample a problem until solved, then move to the next. Under power-law pass@k with $0 < \alpha < 1$, we show that it yields a *sublinear* coverage@cost growth, i.e., investing additional compute has diminishing returns. We propose **Reset-and-Discard (ReD)**: after every τ attempts, reset to the next problem in a queue, and discard any problem as soon as it is solved. This breadth-first policy is motivated by the restart principle

from stochastic processes [15–17]: failed attempts reveal that a problem is harder than typical, so the expected gain from another attempt is smaller than starting fresh.

Related work. Test-time compute scaling has been extensively studied [4, 10, 11, 18]. Recent work models the power-law structure of pass@k and develops sample-efficient estimators [12–14]. Budget-aware evaluation has been advocated by Wang et al. [19]. Stochastic resetting has been previously used in randomized algorithms [20–25] and non-equilibrium statistical physics [15–17, 26–36]. Unlike prior work that optimizes *per-instance* success, we optimize *global throughput*: how to allocate a fixed inference budget across many independent questions.

Our contributions are: (1) an exact renewal-theory mapping from pass@k to coverage@cost, characterizing its sublinear growth for $0 < \alpha < 1$; (2) proofs that ReD with $\tau = 1$ maximizes coverage@cost at every budget, even under imperfect verifiers (Apps. C and D.2); (3) a simple linear-regression estimator for α from ReD trajectories, without costly large-k pass@k evaluation; (4) empirical validation across three LLMs and three benchmarks.

2. From pass@k to Coverage@Cost with ReD

2.1. Mapping pass@k to coverage@cost

Let p_i be the per-problem success probability drawn from difficulty distribution $\mathcal{P}(p)$. pass@k is the ensemble average over the per-problem CDF $1-(1-p_i)^k$,

$$\text{pass@}k = 1 - \int_0^1 (1-p)^k \mathcal{P}(p)\,\mathrm{d}p. \tag{1}$$

Under mild conditions ($\mathcal{P}(p) \simeq cp^{\alpha-1}$ for $p \ll 1$, where c and α are positive constants), a change of variables $z = pk$ and taking the large-k limit yield power-law scaling:

$$1 - \text{pass@}k = \frac{1}{k}\int_0^k (1-z/k)^k \mathcal{P}(z/k)\mathrm{d}z = \underset{k\gg 1}{\approx} \frac{c}{k^\alpha}\int_0^\infty e^{-z} z^{\alpha-1}\mathrm{d}z = c\,\Gamma(\alpha)\,k^{-\alpha}. \tag{2}$$

This shows that any $\mathcal{P}(p)$ with power-law behavior at small p, results in a power-law scaling for pass@k, as also obtained by Schaeffer et al. [12].

To connect pass@k to coverage@cost, consider the number of attempts to solve the i-th question for the first time, $T^{(i)}$. Let $x(t) := \sup\{n : \sum_{i=1}^n T^{(i)} \le t\}$ be the number of unique questions answered by an LLM after a cumulative number of attempts t. Because all questions are i.i.d., $T^{(i)} \sim T$, $\forall i$, $x(t)$ describes a renewal process, i.e., its mean, coverage@cost$(t) := \langle x(t)\rangle$ satisfies [37]:

$$\text{coverage@cost}(t) = F(t) + \sum_{j=1}^{t} \text{coverage@cost}(t-j)\big[F(j) - F(j-1)\big], \tag{3}$$

where $F(t) := \Pr(T \le t) = \text{pass@}t$ is the CDF of the renewal process (derivation in App. B.1). A Z-transform analysis [38] yields the asymptotic behavior (derivation in App. B.2):

$$\text{coverage@cost}(t) \propto \begin{cases} t^\alpha & \text{if } 0<\alpha<1 \quad (\mathbb{E}[T] \text{ diverges}), \\ \frac{t}{\mathbb{E}[T]} & \text{if } \alpha > 1 \quad (\mathbb{E}[T] \text{ finite}). \end{cases} \tag{4}$$

For $0 < \alpha < 1$, common for LLMs [13], coverage@cost grows sub-linearly and $\mathbb{E}[T]$ diverges.

2.2. The ReD protocol

ReD cycles through a queue of problems: each is attempted up to τ times per round; if solved it is discarded, otherwise it returns to the back of the queue. Under resetting at interval τ, the effective CDF of completion times is [39]:

$$F_\tau(t) = 1 - \big(1 - F(\tau)\big)^n \big(1 - F(u)\big), \quad n = \lfloor t/\tau \rfloor,\ u = t - n\tau. \tag{5}$$

The mean completion time under resetting is then obtained from pass@$k = F(k)$ [34, 40, 41]:

$$\mathbb{E}[T_\tau] = \frac{G(\tau)}{F(\tau)}, \quad G(\tau) = \sum_{k=0}^{\tau-1} \big(1 - F(k)\big). \tag{6}$$

Importantly, we get that both $G(\tau)$ and $1/F(\tau)$ are finite for finite $\tau \geq 1$, since $0 < F(t) \leq 1$ for any $t \geq 1$. As a result, $\mathbb{E}[T_\tau]$ is also finite and coverage@cost grows linearly at rate $1/\mathbb{E}[T_\tau]$:

$$\text{coverage@cost}_\tau(t) \simeq \frac{t}{\mathbb{E}[T_\tau]} = \frac{t\,F(\tau)}{G(\tau)}. \tag{7}$$

For $0 < \alpha < 1$, this is a qualitative, dramatic improvement going from sublinear to linear growth, regardless of the specific value of α. For $\alpha > 1$, the coverage grows linearly with and without ReD, but we show in App. C, that ReD is always beneficial, regardless of the difficulty distribution.

Finally, we prove that $\tau = 1$ (reset every attempt) minimizes mean completion time (see App. C). Combined with the optimality of deterministic over stochastic schedules [16], ReD with $\tau = 1$ is the globally optimal resetting policy.

Theorem 1 (Optimality of ReD) *For any $\mathcal{P}(p)$ and all $\tau \geq 1$, $\mathbb{E}[T_\tau] \leq \mathbb{E}[T_{\tau+1}]$.*

We extend this result to imperfect verifiers in App. D.2, as summarized below:

Theorem 2 (ReD point-wise advantage under imperfect verifiers) *For any non-degenerate $\mathcal{P}(p)$ and any verifier with $\mathrm{FNR} + \mathrm{FPR} < 1$, ReD strictly maximizes the expected reward per attempt, i.e., the probability that a question is both solved correctly and marked as correct by the verifier, over all multiple-independent-attempt strategies (proof in App. D.2).*

Since $\mathrm{FNR} = \mathrm{FPR} = 0$ satisfies $\mathrm{FNR} + \mathrm{FPR} < 1$, Thm. 2 shows that ReD is the better strategy at every budget, with or without verifier noise.

2.3. Prediction for finite datasets

For a finite pool of N problems, after n rounds the expected number of unsolved problems is $\langle R_n \rangle = N(1 - \text{pass@}n)$, and coverage@cost$_{\tau=1}(t(n)) = N - \langle R_n \rangle = N\text{pass@}n$ where $t(n)$ is the total number of attempts at the end of the n-th round. To predict coverage@cost$_{\tau=1}$, we replace $t(n)$ in by its mean, $\langle t(n) \rangle = N \sum_{k=0}^{n-1} (1 - \text{pass@}k)$:

$$\text{coverage@cost}_{\tau=1}\big(\langle t(n) \rangle\big) \approx N\,\text{pass@}n. \tag{8}$$

We find this approximation to be empirically accurate (Fig. 1, top), predicting the full ReD coverage curve from pass@k alone, without any additional experiments.

3. Inferring the Scaling Exponent from ReD

ReD trajectories provide an efficient route to estimate α, the inference-time scaling exponent, without large-k pass@k evaluation. After n rounds, surviving problems are harder than average: their difficulty distribution converges to $\mathrm{Beta}(p;\alpha,n{+}1)$ whenever $\mathcal{P}_0(p) \simeq cp^{\alpha-1}$ near $p=0$ (derivation in App. E). Neglecting the initial relaxation towards the Beta distribution and repeating the derivation in App. E using $\mathcal{P}_0(p) = \mathrm{Beta}(p;\alpha,\beta)$ in Eq. (S20), we obtain $\mathcal{P}_n(p) = \mathrm{Beta}(p;\alpha,\beta+n)$. The mean of this distribution is $\gamma_n \coloneqq \alpha/(\alpha+\beta+n)$, so on average $\gamma_n\langle R_n\rangle$ questions are solved at round n. Rearranging $\langle R_{n+1}\rangle = (1-\gamma_n)\langle R_n\rangle$ yields a linear relation:

$$-\frac{\langle R_n\rangle}{\langle R_{n+1}-R_n\rangle} = \frac{n}{\alpha} + \frac{\alpha+\beta}{\alpha}. \tag{9}$$

The exponent is thus estimated from a simple linear fit to round-by-round problem counts, requiring no large-k sampling. This estimator is particularly valuable for characterizing inference-time scaling laws of new models cheaply.

4. Experiments

We run experiments on HumanEval (N=164) [2], GSM8K (N=1,319)[42], and a subset of MMLU-Pro (first N=500 problems) [43] using three models via the Groq API: llama-3.1-8b-instant, llama-3.3-70b-versatile, and gpt-oss-20b [44]. We evaluate pass@k up to $k=100$ once for each model and saving, for each question and every attempt, whether it was answered correctly in a results matrix of dimension questions$\times$attempts. We also recorded how many input and output tokens were used. Then, we simulate solve-to-completion by shuffling rows and columns of the results matrix. We also generate all the realizations for the ReD protocol, by analyzing the results matrix in rounds, going column-by-column, and discarding rows that were answered in previous rounds.

HumanEval (Fig. 1, top and bottom). ReD consistently and substantially outperforms solve-to-completion at every coverage level for all three models; the theoretical prediction of Eq. (8) matches the empirical curves closely. Applying Eq. (9) to the 8b model yields $\alpha = 0.34 \pm 0.01$, matching the direct high-k estimate (see App. H) and confirming the estimator's accuracy.

Economic efficiency (Fig. 1, middle). ReD on llama-3.1-8b matches or outperforms standard solve-to-completion on the 70b model up to $\sim$90% coverage in attempts, and dominates across all coverage levels in USD cost. As gpt-oss-20b is considerably more verbose, ReD on 8b also outperforms it in tokens and USD across nearly all coverage levels. The preferred hybrid strategy is to run ReD with the small model first, routing only unsolved problems to the larger model.

Additional benchmarks and baselines. We validate on GSM8K and MMLU-Pro at two model scales (App. A): on GSM8K, ReD reaches 81% coverage after the first round versus 34% for standard sampling. Noisy-verifier experiments (App. D.1) and comparisons against Continuous Reflection (App. F) consistently show ReD dominating over other baselines across budgets. A hardware latency analysis accounting for KV-cache effects is in App. G.

5. Summary

We proposed Reset-and-Discard (ReD), a resetting-based LLM inference policy that provably improves coverage@cost at every budget. The renewal-theory framework shows that power-law pass@k with $0<\alpha<1$ yields sublinear coverage@cost under solve-to-completion; ReD restores

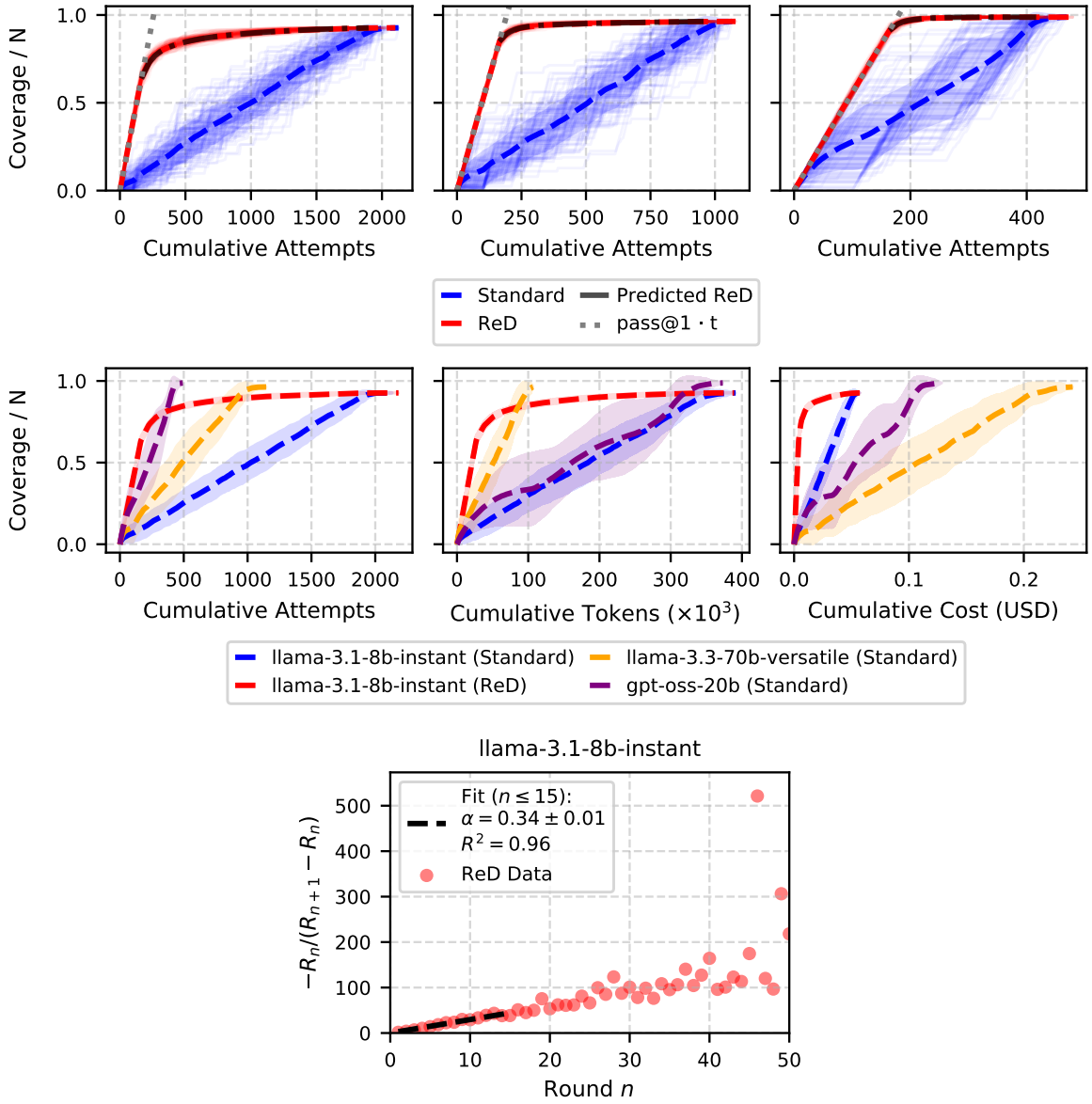

Figure 1: **(Top)** Coverage@cost (normalized by $N{=}164$) vs. cumulative attempts for three LLMs: llama-3.1-8b-instant (Left), llama-3.3-70b-versatile (Middle), and gpt-oss-20b (Right). Dashed: mean over 100 realizations; shaded: ± 1 standard deviation; dot-dashed: theory (Eq. (8)); dotted: pass@$1 \cdot t$. **(Middle)** Coverage@cost of ReD on llama-3.1-8b vs. standard solve-to-completion on larger models (attempts, tokens, USD). **(Bottom)** $-\langle R_n \rangle / \langle R_{n+1} - R_n \rangle$ vs. round n for llama-3.1-8b; linear fit gives $\alpha = 0.34 \pm 0.01$.

linear growth, and $\tau = 1$ is globally optimal (Thm. 1). The advantage extends to imperfect verifiers (Thm. 2), and ReD trajectories yield an efficient estimator of the inference power-law exponent α (Eq. (9)). Validated across three models and three benchmarks, ReD consistently reduces the compute budget required to reach any target coverage.

References

[1] Sumith Kulal, Panupong Pasupat, Kartik Chandra, Mina Lee, Oded Padon, Alex Aiken, and Percy S Liang. Spoc: Search-based pseudocode to code. In H. Wallach, H. Larochelle, A. Beygelzimer, F. d'Alché-Buc, E. Fox, and R. Garnett, editors, *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019. URL https://proceedings.neurips.cc/paper_files/paper/2019/file/7298332f04ac004a0ca44cc69ecf6f6b-Paper.pdf.

[2] Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, Nick Ryder, Mikhail Pavlov, Alethea Power, Lukasz Kaiser, Mohammad Bavarian, Clemens Winter, Philippe Tillet, Felipe Petroski Such, Dave Cummings, Matthias Plappert, Fotios Chantzis, Elizabeth Barnes, Ariel Herbert-Voss, William Hebgen Guss, Alex Nichol, Alex Paino, Nikolas Tezak, Jie Tang, Igor Babuschkin, Suchir Balaji, Shantanu Jain, William Saunders, Christopher Hesse, Andrew N. Carr, Jan Leike, Josh Achiam, Vedant Misra, Evan Morikawa, Alec Radford, Matthew Knight, Miles Brundage, Mira Murati, Katie Mayer, Peter Welinder, Bob McGrew, Dario Amodei, Sam McCandlish, Ilya Sutskever, and Wojciech Zaremba. Evaluating large language models trained on code, 2021. URL https://arxiv.org/abs/2107.03374.

[3] Emily First, Markus N. Rabe, Talia Ringer, and Yuriy Brun. Baldur: Whole-proof generation and repair with large language models, 2023. URL https://arxiv.org/abs/2303.04910.

[4] Bradley Brown, Jordan Juravsky, Ryan Saul 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. URL https://arxiv.org/abs/2407.21787.

[5] Michael Hassid, Tal Remez, Jonas Gehring, Roy Schwartz, and Yossi Adi. The larger the better? improved llm code-generation via budget reallocation. In *First Conference on Language Modeling*, 2024. URL https://openreview.net/forum?id=QJvfpWSpWm.

[6] Lingjiao Chen, Jared Quincy Davis, Boris Hanin, Peter Bailis, Ion Stoica, Matei Zaharia, and James Zou. Are more llm calls all you need? towards scaling laws of compound inference systems, 2024. URL https://arxiv.org/abs/2403.02419.

[7] Ryan Ehrlich, Bradley Brown, Jordan Juravsky, Ronald Clark, Christopher Ré, and Azalia Mirhoseini. Codemonkeys: Scaling test-time compute for software engineering, 2025. URL https://arxiv.org/abs/2501.14723.

[8] Jacky Kwok, Christopher Agia, Rohan Sinha, Matt Foutter, Shulu Li, Ion Stoica, Azalia Mirhoseini, and Marco Pavone. Robomonkey: Scaling test-time sampling and verification for vision-language-action models, 2025. URL https://arxiv.org/abs/2506.17811.

[9] 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, 2024. URL https://arxiv.org/abs/2412.03556.

[10] Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024. URL https://arxiv.org/abs/2408.03314.

[11] Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for problem-solving with language models, 2024. URL https://arxiv.org/abs/2408.00724.

[12] Rylan Schaeffer, Joshua Kazdan, John Hughes, Jordan Juravsky, Sara Price, Aengus Lynch, Erik Jones, Robert Kirk, Azalia Mirhoseini, and Sanmi Koyejo. How do large language monkeys get their power (laws)? *arXiv preprint arXiv:2502.17578*, 2025.

[13] Noam Itzhak Levi. A simple model of inference scaling laws. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=5ulCAfxJzc.

[14] Joshua Kazdan, Rylan Schaeffer, Youssef Allouah, Colin Sullivan, Kyssen Yu, Noam Levi, and Sanmi Koyejo. Efficient prediction of pass@k scaling in large language models. *arXiv preprint arXiv:2510.05197*, 2025. URL https://arxiv.org/abs/2510.05197.

[15] Shlomi Reuveni. Optimal stochastic restart renders fluctuations in first passage times universal. *Physical Review Letters*, 116(17):170601, 2016. doi: 10.1103/PhysRevLett.116.170601. URL https://doi.org/10.1103/PhysRevLett.116.170601.

[16] Abhishek Pal and Shlomi Reuveni. First passage under restart. *Physical Review Letters*, 118: 030603, 2017. doi: 10.1103/PhysRevLett.118.030603.

[17] Martin R Evans, Satya N Majumdar, and Grégory Schehr. Stochastic resetting and applications. *Journal of Physics A: Mathematical and Theoretical*, 53(19):193001, 2020. URL https://doi.org/10.1088/1751-8121/ab7cfe.

[18] Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research*, October 2024. URL https://arxiv.org/abs/2406.16838.

[19] Junlin Wang, Siddhartha Jain, Dejiao Zhang, Baishakhi Ray, Varun Kumar, and Ben Athiwaratkun. Reasoning in token economies: Budget-aware evaluation of LLM reasoning strategies. In *Proceedings of the 2024 Conference on Empirical Methods in Natural Language Processing (EMNLP)*, 2024. URL https://aclanthology.org/2024.emnlp-main.1112/.

[20] Michael Luby, Alistair Sinclair, and David Zuckerman. Optimal speedup of Las Vegas algorithms. *Information Processing Letters*, 47(4):173–180, 1993. URL https://doi.org/10.1016/0020-0190(93)90029-9.

[21] Carla P. Gomes, Bart Selman, and Nuno Crato. Heavy-tailed distributions in combinatorial search. In *Principles and Practice of Constraint Programming (CP)*, 1997. URL https://doi.org/10.1007/BFb0017434.

[22] Carla P. Gomes, Bart Selman, and Henry A. Kautz. Heavy-tailed phenomena in satisfiability and constraint satisfaction problems. *Journal of Automated Reasoning*, 24:67–100, 2000. URL https://doi.org/10.1023/A:1006314320276.

[23] Toby Walsh. Search in a small world. In *Proceedings of the Sixteenth International Joint Conference on Artificial Intelligence (IJCAI)*, 1999.

[24] Ryan Williams, Carla P. Gomes, and Bart Selman. On the connections between heavy-tailedness, backdoors, and restarts. In *Theory and Applications of Satisfiability Testing (SAT)*, 2003.

[25] Matthew Streeter, Daniel Golovin, and Stephen Smith. Restart schedules for ensembles of problem instances. In *Proceedings of the AAAI Conference on Artificial Intelligence (AAAI)*, 2007.

[26] M. R. Evans and S. N. Majumdar. Diffusion with stochastic resetting. *Physical Review Letters*, 106:160601, 2011. doi: 10.1103/PhysRevLett.106.160601. URL https://doi.org/10.1103/PhysRevLett.106.160601.

[27] Benjamin De Bruyne, Satya N Majumdar, and Grégory Schehr. Optimal resetting brownian bridges via enhanced fluctuations. *Physical Review Letters*, 128(20):200603, 2022. URL https://doi.org/10.1103/PhysRevLett.128.200603.

[28] Aanjaneya Kumar and Arnab Pal. Universal framework for record ages under restart. *Physical Review Letters*, 130(15):157101, 2023. URL https://doi.org/10.1103/PhysRevLett.130.157101.

[29] Ofir Blumer, Shlomi Reuveni, and Barak Hirshberg. Stochastic resetting for enhanced sampling. *The journal of physical chemistry letters*, 13(48):11230–11236, 2022. URL https://pubs.acs.org/doi/10.1021/acs.jpclett.2c03055.

[30] Ofir Blumer, Shlomi Reuveni, and Barak Hirshberg. Short-time infrequent metadynamics for improved kinetics inference. *Journal of Chemical Theory and Computation*, 20(9):3484–3491, 2024. URL https://pubs.acs.org/doi/10.1021/acs.jctc.4c00170.

[31] Jonathan R Church, Ofir Blumer, Tommer D Keidar, Leo Ploutno, Shlomi Reuveni, and Barak Hirshberg. Accelerating molecular dynamics through informed resetting. *Journal of Chemical Theory and Computation*, 21(2):605–613, 2025. URL https://pubs.acs.org/doi/10.1021/acs.jctc.4c01238.

[32] Tommer D Keidar, Ofir Blumer, Barak Hirshberg, and Shlomi Reuveni. Adaptive resetting for informed search strategies and the design of non-equilibrium steady-states. *Nature Communications*, 16(1):7259, 2025. URL https://doi.org/10.1038/s41467-025-62398-2.

[33] Ofir Blumer and Barak Hirshberg. Have you tried turning it off and on again? stochastic resetting for enhanced sampling. *Wiley Interdisciplinary Reviews: Computational Molecular Science*, 15(4):e70038, 2025. URL https://doi.org/10.1002/wcms.70038.

[34] Sagi Meir, Tommer D Keidar, Shlomi Reuveni, and Barak Hirshberg. First-passage approach to optimizing perturbations for improved training of machine learning models. *Machine Learning: Science and Technology*, 6(2):025053, 2025. URL https://doi.org/10.1088/2632-2153/add8df.

[35] Youngkyoung Bae, Yeongwoo Song, and Hawoong Jeong. Stochastic resetting mitigates latent gradient bias of sgd from label noise. *Machine Learning: Science and Technology*, 6(1):015062, 2025. URL https://doi.org/10.1088/2632-2153/adbc46.

[36] Ilya Loshchilov and Frank Hutter. Sgdr: Stochastic gradient descent with warm restarts. *arXiv preprint arXiv:1608.03983*, 2016. URL https://arxiv.org/abs/1608.03983.

[37] Geoffrey Grimmett and David Stirzaker. *Probability and random processes*. Oxford university press, 2020.

[38] William Feller. *An introduction to probability theory and its applications, Volume 2*, volume 2. John Wiley & Sons, 1991.

[39] Iddo Eliazar and Shlomi Reuveni. Tail-behavior roadmap for sharp restart. *Journal of Physics A: Mathematical and Theoretical*, 54(12):125001, 2021. URL https://doi.org/10.1088/1751-8121/abe4a0.

[40] Iddo Eliazar and Shlomi Reuveni. Mean-performance of sharp restart i: statistical roadmap. *Journal of Physics A: Mathematical and Theoretical*, 53(40):405004, 2020. URL https://doi.org/10.1088/1751-8121/abae8c.

[41] Ofek Lauber Bonomo and Arnab Pal. First passage under restart for discrete space and time: Application to one-dimensional confined lattice random walks. *Phys. Rev. E*, 103:052129, May 2021. doi: 10.1103/PhysRevE.103.052129. URL https://doi.org/10.1103/PhysRevE.103.052129.

[42] Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training verifiers to solve math word problems, 2021. URL https://arxiv.org/abs/2110.14168.

[43] Yubo Wang, Xueguang Ma, Ge Zhang, Yuansheng Ni, Abhranil Chandra, Shiguang Guo, Weiming Ren, Aaran Arulraj, Xuan He, Ziyan Jiang, Tianle Li, Max Ku, Kai Wang, Alex Zhuang, Rongqi Fan, Xiang Yue, and Wenhu Chen. Mmlu-pro: A more robust and challenging multi-task language understanding benchmark. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 95266–95290. Curran Associates, Inc., 2024. doi: 10.52202/079017-3018. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/ad236edc564f3e3156e1b2feafb99a24-Paper-Datasets_and_Benchmarks_Track.pdf.

[44] OpenAI. gpt-oss-120b & gpt-oss-20b model card, 2025. URL https://arxiv.org/abs/2508.10925.

[45] Harvey Scher and Elliott W. Montroll. Anomalous transit-time dispersion in amorphous solids. *Phys. Rev. B*, 12:2455–2477, Sep 1975. doi: 10.1103/PhysRevB.12.2455. URL https://doi.org/10.1103/PhysRevB.12.2455.

[46] Ralf Metzler and Joseph Klafter. The random walk's guide to anomalous diffusion: a fractional dynamics approach. *Physics reports*, 339(1):1–77, 2000.

[47] Yinmin Zhong, Shengyu Liu, Junda Chen, Jianbo Hu, Yibo Zhu, Xuanzhe Liu, Xin Jin, and Hao Zhang. DistServe: Disaggregating prefill and decoding for goodput-optimized large language model serving. In *18th USENIX Symposium on Operating Systems Design and Implementation (OSDI 24)*, pages 193–210, Santa Clara, CA, July 2024. USENIX Association. ISBN 978-1-939133-40-3. URL https://www.usenix.org/conference/osdi24/presentation/zhong-yinmin.

[48] E2E Networks. Inference benchmarks - tir documentation, 2026. URL https://docs.e2enetworks.com/docs/tir/benchmarks/inference_benchmarks/. Accessed: 2026-04-30.

Appendix A. Additional Benchmarks

We extended evaluation beyond HumanEval to GSM8K ($N = 1{,}319$ math problems) and a subset of MMLU-Pro (first $N = 500$, multi-discipline multiple-choice reasoning problems), on two model scales (llama-3.1-8b and llama-3.3-70b). We report below (Tab. S1 and Tab. S2) coverage@cost for these datasets, evaluated at $k \times N$ total attempts, averaged over 1,000 realizations.

Table S1: GSM8K ($N = 1{,}319$)

Model	Method	$1 \times N$	$2 \times N$	$3 \times N$	$5 \times N$
8b	Standard	33.6%	66.7%	98.3%	99.0%
	ReD	**81.3%**	**98.1%**	99.0%	99.0%
70b	Standard	25.1%	49.6%	74.4%	97.5%
	ReD	**93.2%**	**97.0%**	**97.3%**	97.5%

Table S2: MMLU-Pro ($N = 500$)

Model	Method	$1 \times N$	$3 \times N$	$5 \times N$
8b	Standard	10.4%	30.3%	50.3%
	ReD	**46.5%**	**76.1%**	**87.0%**
70b	Standard	9.7%	28.2%	46.5%
	ReD	**71.8%**	**87.1%**	**90.3%**

At a range of total attempts, ReD substantially outperforms solve-to-completion for both datasets and both models, demonstrating its utility across three task types (coding, math, reasoning).

From a theoretical standpoint, as long as there is some variability among questions in the dataset, we proved that ReD must accelerate the inference.

Appendix B. Connecting coverage@cost to pass@k for a Large Dataset

In this appendix, we recap known results in renewal theory [37]. We reproduce the proofs below for completeness and to facilitate reading.

B.1. Proof of Eq. (3)

In the main text, we defined $x(t)$ as follows,

$$x(t) \coloneqq \sup\left\{ n : \sum_{i=1}^{n} T^{(i)} \leq t \right\}, \tag{S1}$$

where $T^{(i)}$ are i.i.d. positive random variables distributed as T. Given that the first question was answered correctly at step j, renewal processes have the following recursion relation,

$$x(t) \overset{d}{=} \begin{cases} 1 + x(t-j) & \text{if } j \leq t, \\ 0 & \text{if } j > t. \end{cases} \tag{S2}$$

In other words, for $t < j$, we have $x(t) = 0$, otherwise, $x(t) \overset{d}{=} 1 + x(t-j)$. This is because the first success was at time j, after which, the process renews itself and the number of additional successes from this time onward is distributed as $x(t-j)$.

To prove Eq. (3), we introduce the following notations: $m_1(t) \coloneqq \text{coverage@cost}(t)$ and $f(j) \coloneqq F(j) - F(j-1)$, where $F(j)$ is the CDF of T and $f(j)$ is its probability mass function. Note that with this notation, Eq. (3) has the following form

$$m_1(t) = F(t) + \sum_{j=1}^{t} m_1(t-j) f(j). \tag{S3}$$

Next, we use the recursion relation in Eq. (S2) with the law of total expectation, $\langle x \rangle = \langle \langle x \mid y \rangle \rangle$, where $\langle x \mid y \rangle$ is the conditional expectation of x given y.

$$\begin{aligned} m_1(t) = \langle x(t) \rangle = \langle \langle x(t) \mid T^{(1)} \rangle \rangle = \sum_{j=1}^{\infty} \langle [x(t) \mid T^{(1)} = j] \rangle f(j) = \sum_{j=1}^{\infty} \mathbb{I}_{\{j \leq t\}} (1 + \langle x(t-j) \rangle) f(j) \\ = \sum_{j=1}^{t} (1 + m_1(t-j)) f(j) = F(t) + \sum_{j=1}^{t} m_1(t-j) f(j). \end{aligned} \tag{S4}$$

Replacing $m_1(t) = \text{coverage@cost}(t)$ gives Eq. (3). Similarly, we now obtain the second moment.

We use the recursive relation in Eq. (S2), and plug $x^2(t) \stackrel{d}{=} (1+x(t-j))^2 = 1 + 2x(t-j) + x^2(t-j)$ for time $j \le t$.

$$m_2(t) = \langle x^2(t) \rangle = \langle \langle x^2(t) \mid T^{(1)} \rangle \rangle = \sum_{j=1}^{\infty} \langle [x^2(t) \mid T^{(1)} = j] \rangle f(j) =$$
$$\sum_{j=1}^{\infty} \mathbb{I}_{\{j \le t\}} \left(1 + 2\langle x(t-j) \rangle + \langle x^2(t-j) \rangle\right) f(j) = \sum_{j=1}^{t} (1 + 2m_1(t-j) + m_2(t-j)) f(j) =$$
$$\sum_{j=1}^{t} (2 + 2m_1(t-j) - 1 + m_2(t-j)) f(j) = 2m_1(t) - F(t) + \sum_{j=1}^{t} m_2(t-j) f(j). \tag{S5}$$

Overall, we get

$$m_2(t) = 2m_1(t) - F(t) + \sum_{j=1}^{t} m_2(t-j) f(j). \tag{S6}$$

B.2. Proof of Eq. (4)

We start by taking the Z-transform of Eqs. (S3) and (S6), to get

$$\begin{aligned} \mathcal{Z}\{m_1\}(z) &= \frac{\tilde{f}(z)}{(1-z)\left(1-\tilde{f}(z)\right)}, \\ \mathcal{Z}\{m_2\}(z) &= \frac{\tilde{f}(z)\left(1+\tilde{f}(z)\right)}{(1-z)\left(1-\tilde{f}(z)\right)^2}. \end{aligned} \tag{S7}$$

Where $\tilde{f}(z) \equiv \sum_{t=0}^{\infty} f(t) z^t$ is the Z-transform of the probability mass function of the time of answering a question. We used the relation between the Z-transform of the CDF and the probability mass function, $\tilde{F}(z) = \tilde{f}(z)/(1-z)$, and the convolution theorem for Z-transforms [38].

Taking the long-time limit is equivalent to taking the limit $z \to 1$, and inverting the transform. All inversions were done using the Tauberian theorem for Z-transforms [38]. If $1 - \text{pass@}k$ decays to zero faster than ck^{-1}, the mean time to answer a question will be $\mathbb{E}[T]$, and $\tilde{f}(z) = 1 - (1-z)\mathbb{E}[T] + o(1-z)$. Therefore, the asymptotic behavior around $z = 1$ of Eq. (S7) is

$$\begin{aligned} \mathcal{Z}\{m_1\}(z) &\simeq \frac{1}{(1-z)^2 \mathbb{E}[T]} \Rightarrow m_1(t) \simeq \frac{t}{\mathbb{E}[T]}, \\ \mathcal{Z}\{m_2\}(z) &\simeq \frac{2}{(1-z)^3 \mathbb{E}^2[T]} \Rightarrow m_2(t) \simeq \frac{t^2}{\mathbb{E}^2[T]}. \end{aligned} \tag{S8}$$

In this case, coverage@cost scales linearly in time, and the slope is the inverse of the mean number of attempts to answer a single question.

It is also evident that the standard deviation of the number of unique questions answered $\sqrt{m_2(t) - m_1^2(t)}$ grows slower than t at long times. Therefore, the ratio of it with coverage@cost

approaches 0 at late times. This means that in this scenario $x(t)/m_1(t)$ becomes deterministic and coverage@cost provides a good description of the dynamics.

The other case is the one in which $1-\text{pass@}k$ decays as $ck^{-\alpha}/\Gamma(1-\alpha)$, with $0<\alpha<1$, for which we get that around $z=1$, $\tilde{f}(z)=1-c(1-z)^{\alpha}+o((1-z)^{\alpha})$. Then

$$\begin{aligned}\mathcal{Z}\{m_1\}(z) &\simeq \frac{1}{c(1-z)^{1+\alpha}} \Rightarrow m_1(t) \simeq \frac{t^{\alpha}}{c\Gamma(1+\alpha)},\\ \mathcal{Z}\{m_2\}(z) &\simeq \frac{2}{c^2(1-z)^{1+2\alpha}} \Rightarrow m_2(t) \simeq \frac{2t^{2\alpha}}{c^2\Gamma(2\alpha+1)}.\end{aligned} \tag{S9}$$

Here, the behavior of coverage@cost becomes sub-linear, with a power controlled by the inference scaling power α.

The transition between the linear and sub-linear behavior of coverage@cost appears when $\alpha=1$. For $\alpha>1$, the mean time to answer a randomly selected question is finite, and for $\alpha<1$ it diverges. Mathematically, this is the same behavior as the well-studied transition between diffusive and sub-diffusive behavior in continuous-time random walk models used in the physics of transport phenomena [45, 46].

Appendix C. Proof of Thm. 1 (Optimality of ReD)

We will show that $\mathbb{E}[T_{\tau+1}]\geq\mathbb{E}[T_{\tau}]$ for $\tau\geq 1$ and for any $\mathcal{P}(p)$. We do so by defining the following difference $D(\tau)\coloneqq\mathbb{E}[T_{\tau+1}]-\mathbb{E}[T_{\tau}]$ and demonstrate that it is non-negative. We write the difference using Eq. (6)

$$D(\tau)=\frac{G(\tau+1)}{F(\tau+1)}-\frac{G(\tau)}{F(\tau)}=\frac{F(\tau)G(\tau+1)-F(\tau+1)G(\tau)}{F(\tau)F(\tau+1)}. \tag{S10}$$

Because the denominator in Eq. (S10) is non-negative, it is left to show that the numerator is non-negative as well. We start by recalling that

$$F(\tau)\coloneqq\Pr(T\leq\tau)=1-\int_0^1(1-p)^{\tau}\mathcal{P}(p)\mathrm{d}p=1-\mathbb{E}\left[(1-p)^{\tau}\right], \tag{S11}$$

and

$$\begin{aligned}G(\tau)\coloneqq\sum_{k=0}^{\tau-1}(1-F(k))=\int_0^1\sum_{k=0}^{\tau-1}(1-p)^k\mathcal{P}(p)\mathrm{d}p=\int_0^1\frac{1-(1-p)^{\tau}}{p}\mathcal{P}(p)\mathrm{d}p=\\ \mathbb{E}\left[\frac{1-(1-p)^{\tau}}{p}\right].\end{aligned} \tag{S12}$$

We can thus write the numerator of Eq. (S10) as

$$F(\tau)G(\tau+1)-F(\tau+1)G(\tau)=\mathbb{E}\left[1-q^{\tau}\right]\mathbb{E}\left[\frac{1-q^{\tau+1}}{p}\right]-\mathbb{E}\left[1-q^{\tau+1}\right]\mathbb{E}\left[\frac{1-q^{\tau}}{p}\right], \tag{S13}$$

where we have set $q\coloneqq 1-p$. We now observe that $\mathbb{E}\left[1-q^{\tau+1}\right]=\mathbb{E}\left[1-q^{\tau}\right]+\mathbb{E}\left[pq^{\tau}\right]$, and similarly $\mathbb{E}\left[\frac{1-q^{\tau+1}}{p}\right]=\mathbb{E}\left[\frac{1-q^{\tau}}{p}\right]+\mathbb{E}\left[q^{\tau}\right]$. Substituting back into the right hand side of Eq. (S13) we obtain

$$F(\tau)G(\tau+1)-F(\tau+1)G(\tau)=\mathbb{E}\left[1-q^{\tau}\right]\mathbb{E}\left[q^{\tau}\right]-\mathbb{E}\left[\frac{1-q^{\tau}}{p}\right]\mathbb{E}\left[pq^{\tau}\right]. \tag{S14}$$

We now define $u \coloneqq (1-q^\tau)/p$ and continue to develop the right hand side of Eq. (S14)

$$\mathbb{E}\left[1-q^\tau\right]\mathbb{E}\left[q^\tau\right]-\mathbb{E}\left[pq^\tau\right]\mathbb{E}[u]=\mathbb{E}[pu]\mathbb{E}[q^\tau]-\mathbb{E}[pq^\tau]\mathbb{E}[u]= \\ \mathbb{E}[u]\mathbb{E}[q^\tau]\left(\frac{\mathbb{E}[pu]}{\mathbb{E}[u]}-\frac{\mathbb{E}[pq^\tau]}{\mathbb{E}[q^\tau]}\right)=\mathbb{E}[u]\mathbb{E}[q^\tau]\left(\frac{\mathbb{E}[pu]}{\mathbb{E}[u]}-\frac{\mathbb{E}[upq^\tau/u]\mathbb{E}[u]}{\mathbb{E}[u]\mathbb{E}[uq^\tau/u]}\right). \tag{S15}$$

To finish the proof, we define $\mathbb{E}_\lambda[p]=\mathbb{E}[\lambda(p)p]/\mathbb{E}[\lambda(p)]$ as the expectation with respect to the probability density $\mathcal{P}(p)\lambda(p)/\mathbb{E}[\lambda(p)]$ for a positive function $\lambda(p)$. The parenthesis on the right hand side of Eq. (S15) can then be written as

$$\mathbb{E}_u[p]-\mathbb{E}_{q^\tau}[p]=\frac{\mathbb{E}_u[p]\mathbb{E}_u[q^\tau/u]-\mathbb{E}_u[pq^\tau/u]}{\mathbb{E}_u[q^\tau/u]}=-\frac{\mathrm{COV}_u(p,q^\tau/u)}{\mathbb{E}_u[q^\tau/u]}\geq 0.$$

The last inequality is because $q^\tau/u\geq 0$ and because $q^\tau/u=p(1-p)^\tau/(1-(1-p)^\tau)$ is strictly *decreasing* with p, leading to a negative covariance according to the continuous version of the Chebyshev sum inequality. Thus, $D(\tau)\coloneqq\mathbb{E}[T_{\tau+1}]-\mathbb{E}[T_\tau]\geq 0$, which concludes the proof.

Corollary 3 *For any $\mathcal{P}(p)$, and $\forall\tau\in\mathbb{N}\setminus\{0\}$, $\mathbb{E}[T]\geq\mathbb{E}[T_\tau]$.*

Proof Recall $D(\tau)\geq 0$ and note that taking the limit $\tau\to\infty$ in Eq. (6) gives $\lim_{\tau\to\infty}\mathbb{E}[T_\tau]=\mathbb{E}[T]$. Therefore, $\mathbb{E}[T]=\sup\{\mathbb{E}[T_\tau]\}$. ■

Appendix D. Imperfect Verifier

Here we address the case of an imperfect verifier and demonstrate experimentally and theoretically that ReD leads at a range of budgets.

D.1. Experiments with Imperfect Verifiers

We evaluated actual coverage under three noise conditions (FPR = false positive rate, FNR = false negative rate). The results are given for the GSM8K dataset (Tab. S3) and MMLU-Pro (Tab. S4).

Table S3: Coverage with an imperfect verifier across a range of noise conditions for the GSM8K (N = 1,319, llama-3.1-8b, 1,000 realizations) dataset.

Verifier	Method	$1\times N$	$2\times N$	$3\times N$
Perfect	Standard	33.6%	66.7%	98.3%
	ReD	**81.3%**	**98.1%**	99.0%
$\mathrm{FPR}=0.02,\mathrm{FNR}=0.1$	Standard	41.5%	83.1%	97.3%
	ReD	**73.2%**	**96.9%**	97.3%
$\mathrm{FPR}=0.08,\mathrm{FNR}=0.15$	Standard	52.1%	94.4%	94.4%
	ReD	**69.2%**	94.4%	94.4%
$\mathrm{FPR}=0.25,\mathrm{FNR}=0.05$	Standard	67.5%	90.3%	90.3%
	ReD	**77.3%**	90.3%	90.3%

Table S4: Coverage with an imperfect verifier across a range of noise conditions for the MMLU-Pro ($N = 500$, llama-3.1-8b, 1,000 realizations) dataset.

Verifier	Method	$1 \times N$	$3 \times N$	$5 \times N$
Perfect	Standard	10.4%	30.3%	50.3%
	ReD	**46.5**%	**76.1**%	**87.0**%
$\mathrm{FPR} = 0.02, \mathrm{FNR} = 0.1$	Standard	12.9%	38.4%	63.7%
	ReD	**41.8**%	**73.1**%	**83.5**%
$\mathrm{FPR} = 0.08, \mathrm{FNR} = 0.15$	Standard	18.8%	56.1%	75.0%
	ReD	**39.5**%	**70.4**%	75.0%
$\mathrm{FPR} = 0.25, \mathrm{FNR} = 0.05$	Standard	29.5%	63.1%	63.1%
	ReD	**44.2**%	63.1%	63.1%

All numbers in Tab. S3 and Tab. S4 are actual coverage (a problem counts only when a truly correct answer is accepted). At an intermediate number of attempts, a non-zero FPR raises the coverage of solve-to-completion (false positives allow it to stop trying to solve hard problems sooner), while reducing ReD's, narrowing the gap. A non-zero FPR also lowers asymptotic actual coverages for both approaches. Despite that, ReD leads at a range of budgets across all noise conditions. Under high noise (FPR=0.25 or FNR=0.15), the gap closes faster on the easier dataset (GSM8K), while on the harder dataset (MMLU-Pro) the advantage persists to substantially larger budgets.

In addition, we ran the imperfect verifier simulation on HumanEval (Tab. S5) across a range of realistic noise conditions. ReD leads at a range of budgets across the entire realistic noise range.

Table S5: Coverage with an imperfect verifier across a range of realistic noise conditions for the HumanEval ($N = 164$, llama-3.1-8b, 200 realizations) dataset

FPR	FNR	Method	$1 \times N$	$3 \times N$	$5 \times N$
0%	0%	Standard	9.3%	24.5%	39.8%
		ReD	**63.7%**	**84.7%**	**88.4%**
2%	2%	Standard	15.5%	42.2%	67.7%
		ReD	**62.3%**	**83.5%**	**86.8%**
2%	10%	Standard	14.8%	40.7%	66.1%
		ReD	**57.3%**	**83.1%**	**86.5%**
5%	5%	Standard	21.9%	62.7%	83.9%
		ReD	**60.6%**	**82.4%**	83.8%
8%	8%	Standard	25.5%	75.1%	81.3%
		ReD	**58.9%**	**81.1%**	81.5%
8%	15%	Standard	24.9%	72.4%	81.1%
		ReD	**54.3%**	**80.5%**	81.0%

D.2. Proof of Thm. 2

The point-wise advantage of ReD holds for the actual coverage, regardless of the verifier's FPR and FNR. Consider the case of an infinite pool of questions whose distribution of probabilities to answer a question correctly has a non-degenerate density $\mathcal{P}(p)$; and a verifier with some false-negative rate, FNR, and some false-positive rate, FPR, such that FNR + FPR < 1. We will compute the mean reward $\mathbb{E}[\mathcal{R}]$ per attempt, i.e., the probability that a question is both solved correctly and marked as correct by the verifier.

In ReD, on an infinite question pool, each question is given a single attempt. The reward is therefore

$$\mathbb{E}[\mathcal{R}_{\text{ReD}}] = \int_0^1 p(1-\text{FNR})\mathcal{P}(p)\,dp = (1-\text{FNR})\mathbb{E}[p]. \tag{S16}$$

For a general (non-ReD) strategy, we will first compute the mean reward on a given attempt, given that the current question being asked was asked $n-1$ times before

$$\pi_n = \mathbb{E}[\mathcal{R}|n-1 \text{ previous attempts}] = \frac{\int_0^1 (1-\text{FNR})\,p\,[1-(1-\text{FNR})\,p-(1-p)\text{FPR}]^{n-1}\,\mathcal{P}(p)\,dp}{\int_0^1 [1-(1-\text{FNR})\,p-(1-p)\text{FPR}]^{n-1}\,\mathcal{P}(p)\,dp}. \tag{S17}$$

We now show that $\mathbb{E}[\mathcal{R}_{\text{ReD}}] = \int_0^1 (1-\text{FNR})\,p\mathcal{P}(p)\,dp = \pi_1 > \pi_n$.

We set $\Omega_n(p) = [1-(1-\text{FNR})\,p-(1-p)\text{FPR}]^{n-1}$ and observe that

$$\pi_1 - \pi_n = \frac{-(1-\text{FNR})\,\text{Cov}\,(p, \Omega_n(p))}{\mathbb{E}[\Omega_n(p)]} > 0 \tag{S18}$$

where the final inequality follows from p being monotonically increasing and hence negatively correlated with $\Omega_n(p) > 0$, which is monotonically decreasing on the unit interval. Because we have not assumed anything on $\mathcal{P}(p)$, this argument can be done with the conditional distribution after a single previous attempt, thus proving that $\pi_2 > \pi_n \quad \forall n > 2$. Therefore, by induction, the series $\{\pi_n\}_{n=1}^{\infty}$ is monotonically decreasing.

The mean reward per attempt on a generic strategy is thus

$$\mathbb{E}[\mathcal{R}] = \sum_{n=1}^{\infty} \pi_n \Pr(n) < \sum_{n=1}^{\infty} \pi_1 \Pr(n) = \pi_1 = (1 - \text{FNR})\mathbb{E}[p] = \mathbb{E}[\mathcal{R}_{\text{ReD}}], \tag{S19}$$

where $\Pr(n)$ is the fraction of time that the model spends answering questions for the n-th attempt. This proves that the expected reward of the ReD strategy is larger than that of any other strategy.

For a finite data set, ReD acts as a greedy algorithm, as can be seen from the monotonicity of π_n. It always attempts to answer the question with the least number of failures, i.e., lowest n, and therefore highest expected reward.

Since $\pi_1 > \pi_n$ for all $n > 1$, at every attempt t the per-attempt success probability of ReD exceeds that of any competing multiple-independent-attempt strategy. By linearity of expectation, summing over attempts $1, \ldots, t$ yields $\text{coverage@cost}_{\text{ReD}}(t) \geq \text{coverage@cost}_S(t)$ for every budget t, establishing the consequence stated following Thm. 2.

Appendix E. Asymptotic Convergence of Difficulty to the Beta Distribution

The distribution $\mathcal{P}_n(p)$ represents all the questions left unanswered after n rounds. Since each round discards correctly answered questions, the surviving questions are reweighted by the probability of not being solved:

$$\mathcal{P}_n(p) = \frac{\mathcal{P}_0(p)(1-p)^n}{\int_0^1 (1-p)^n \mathcal{P}_0(p)\,\mathrm{d}p}, \tag{S20}$$

where $(1-p)^n$ is the probability of a question with difficulty p surviving n rounds. For large n, the factor $(1-p)^n$ suppresses contributions from high values of p, so only the small-p behavior of $\mathcal{P}_0(p)$ matters. Under the power-law assumption $\mathcal{P}_0(p) \simeq cp^{\alpha-1}$ for $p \ll 1$, we obtain

$$\mathcal{P}_n(p) \underset{n\gg 1}{\approx} \frac{p^{\alpha-1}(1-p)^n}{\int_0^1 p^{\alpha-1}(1-p)^n\,\mathrm{d}p} \sim \text{Beta}(p;\alpha, n+1). \tag{S21}$$

Hence any difficulty distribution with power-law behavior $\mathcal{P}_0(p) \simeq cp^{\alpha-1}$ near $p = 0$ converges to $\text{Beta}(p;\alpha, n+1)$ after several ReD rounds.

Appendix F. Comparison to Other Baselines

F.1. Comparison to Continuous Reflection

Continuous Reflection (CR) is a simple self-correction strategy. If a question is answered incorrectly, on the next attempt, we append an addition to the prompt that informs the model that the question was answered incorrectly and provides the previous answer. This process is repeated at most five times for each question in the GSM8K dataset. The results below are for GSM8K. For the 70b model, CR outperforms solve-to-completion (standard) at $1 \times N$ and $2 \times N$, but ReD is better. For 8b, CR

Table S6: Comparison with stronger allocation baselines: ReD versus CR for GSM8K ($N = 1{,}319$, 1,000 realizations; 8b = llama-3.1-8b, 70b = llama-3.3-70b)

Strategy	$1 \times N$	$2 \times N$	$3 \times N$
Standard (8b)	33.6%	66.7%	98.3%
CR (8b)	32.1%	63.6%	95.3%
ReD (8b)	**81.3%**	**98.1%**	**99.0%**
Standard (70b)	25.1%	49.6%	74.4%
CR (70b)	39.9%	78.9%	97.6%
ReD (70b)	**93.2%**	**97.0%**	97.3%

performs similarly to solve-to-completion. Smaller models tend to lack the ability to self-correct without substantial feedback, so CR buys little. Crucially, ReD dominates by a wide margin at every budget.

To generate ReD trajectories with self-correction, we performed the following experiment. In every ReD round, all questions that were not discarded in the previous round are asked again, and we append to the prompt a note saying they failed and give the previous answer. This strategy (Refl-ReD) did not substantially improve the performance of ReD for the GSM8K dataset, for both models, llama-3.1-8b and llama-3.3-70b, and 1000 realizations.

Table S7: ReD beyond independent repeated sampling.

Strategy	$1 \times N$	$2 \times N$	$3 \times N$
Refl-ReD (8b)	81.4%	94.2%	98.3%
ReD (8b)	81.3%	98.2%	99.0%
Refl-ReD (70b)	93.3%	97.5%	97.6%
ReD (70b)	93.2%	97.0%	97.4%

F.2. Allocation Baseline: τ-Sweep on HumanEval

We ran a τ-sweep on HumanEval ($N = 164$, both models, 200 realizations): ReD with τ attempts per problem per round before discarding, for $\tau \in \{1, 2, 4, 8\}$. $\tau = 1$ is standard ReD and is optimal by Thm. 1.

ReD ($\tau = 1$) dominates at every budget for both models, empirically confirming Thm. 1.

Appendix G. The Influence of KV-Cache on the GPU Time with ReD

For several cloud API deployments (at least in Groq during the experiments), token count is the relevant metric: KV-cache state does not persist across calls, regardless of the strategy. For self-hosted GPU inference, we performed a hardware latency estimate.

Following Zhong et al. [47], we estimate the overall request latency as the sum of the time-to-first-token (TTFT) plus time-per-output-token (TPOT) times the number of generated tokens. We also define per-token rates $L_p = \text{TTFT}/N_{\text{in}}$ (prefill) and $L_d = \text{TPOT}$ (decode), where N_{in} and $N_{\text{out}}^{(j)}$

Table S8: τ-sweep on HumanEval: coverage at fixed total-attempt budgets (200 realizations). Bold = best at each budget.

Strategy	$1\times N$	$2\times N$	$3\times N$	$5\times N$
llama-3.1-8b				
ReD (τ=1)	**63.9%**	**80.0%**	**84.5%**	**88.2%**
τ=2	36.4%	73.1%	81.7%	87.3%
τ=4	19.8%	39.5%	59.2%	83.6%
τ=8	10.7%	20.8%	31.4%	52.2%
llama-3.3-70b				
ReD (τ=1)	**85.0%**	**94.1%**	**95.0%**	**96.0%**
τ=2	44.4%	88.8%	94.3%	95.7%
τ=4	22.9%	45.8%	68.7%	94.4%
τ=8	12.0%	23.4%	35.4%	58.7%

are the number of input tokens and output tokens on attempt j, respectively. We model GPU time per problem as $T(k)$, where k is the number of attempts:

- **Solve-to-completion** (prefill paid once, KV cache retained):

$$T_{\text{standard}}(k) = N_{\text{in}}L_p + \sum_{j=1}^{k} N_{\text{out}}^{(j)} L_d$$

- **Strict ReD** (full prefill on every attempt, zero cache retention):

$$T_{\text{ReD}}(k) = \sum_{j=1}^{k} \left(N_{\text{in}}L_p + N_{\text{out}}^{(j)} L_d \right)$$

We set $L_p = 5$ ms/token and $L_d = 20$ ms/token ($L_d/L_p = 4$). We note that this ratio is a substantial underestimate: on H100 hardware running llama-3.1-8b, the E2E Networks inference benchmark [48] reports TTFT $\approx$ 29ms for 128-token inputs ($L_p \approx 0.23$ ms/token) and TPOT ≈ 22 ms/token, giving $L_d/L_p \approx 100$. A higher ratio means prefill is relatively cheaper, so the advantage of solve-to-completion from caching a single prefill per problem shrinks. Our simulation therefore *overestimates* the KV-cache benefit of solve-to-completion and is conservative for ReD. The results below are for GSM8K. Even in the worst-case KV-cache scenario (zero retention), ReD outperforms solve-to-completion at all GPU-second budgets tested. ReD's advantage compensates for the prefill overhead incurred on every context switch in this case.

Appendix H. Evaluating the Power-Law Exponent for llama-3.1-8b-instant

Table S9: Hardware latency estimate for GSM8K ($N = 1{,}319$, 1,000 realizations, 8b=llama-3.1-8b, 70b=llama-3.3-70b)

GPU-s	Method	**8b**	**70b**
2,219	Standard	10.1%	9.5%
	ReD	**26.0%**	**34.0%**
11,516	Standard	50.0%	47.2%
	ReD	**95.6%**	**96.9%**
21,952	Standard	94.5%	89.6%
	ReD	**98.8%**	**97.4%**

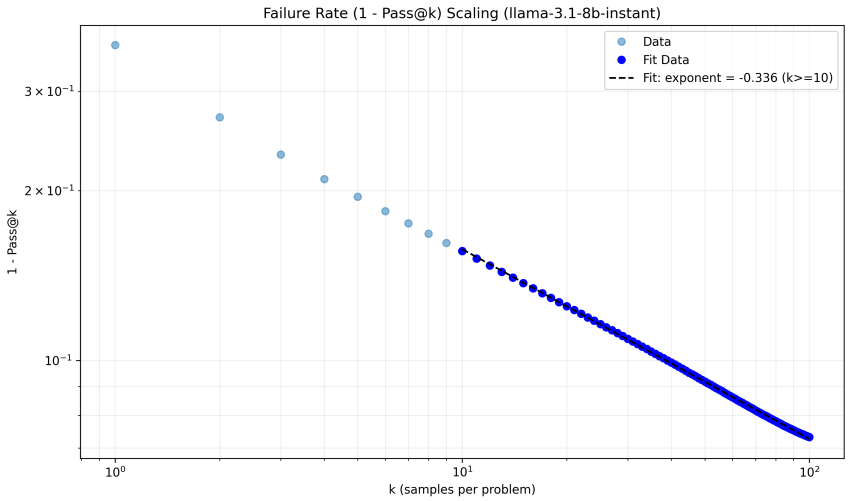

Figure S1: A linear fit of $\log(1 - \text{pass@}k)$, of the llama-3.1-8b-instant, versus $\log k$. For $k \gg 1$ indeed $(1 - \text{pass@}k) \propto k^{-\alpha}$, and the obtained power-law exponent is $\alpha = 0.34$.