MORE CORRECT MASS, WORSE ANSWERS: WHY POWER SAMPLING CAN FAIL AND HOW TO FIX IT

Anonymous authors
Paper under double-blind review

ABSTRACT

Power Sampling sharpens a language model's distribution over complete generation trajectories, offering a verifier-free way to improve reasoning at inference time. It could also serve as a front end for downstream multi-sample inference. Using self-consistency as a representative case, we find that Base-pool Power weighting can increase correct-trajectory mass on average across problems while reducing consensus accuracy. Across models and reasoning benchmarks, accuracy losses reach 10.1 percentage points. We trace this behavior to two mismatches. Dose mismatch arises because a fixed exponent induces different amounts of distributional change across problems. Coverage mismatch arises because sharpening can favor a dominant wrong trajectory over several moderate-probability paths supporting the correct answer. High pass@k therefore need not imply sufficient answer-level support for consensus. We propose Relative-Rank SoftSat, which calibrates gains through within-problem likelihood ranks and saturates them at high ranks, limiting concentration on dominant paths. On shared Base pools, the resulting weighted self-consistency readout recovers much of the accuracy lost to Power weighting, with further gains from same-model self-evaluation.

1 INTRODUCTION

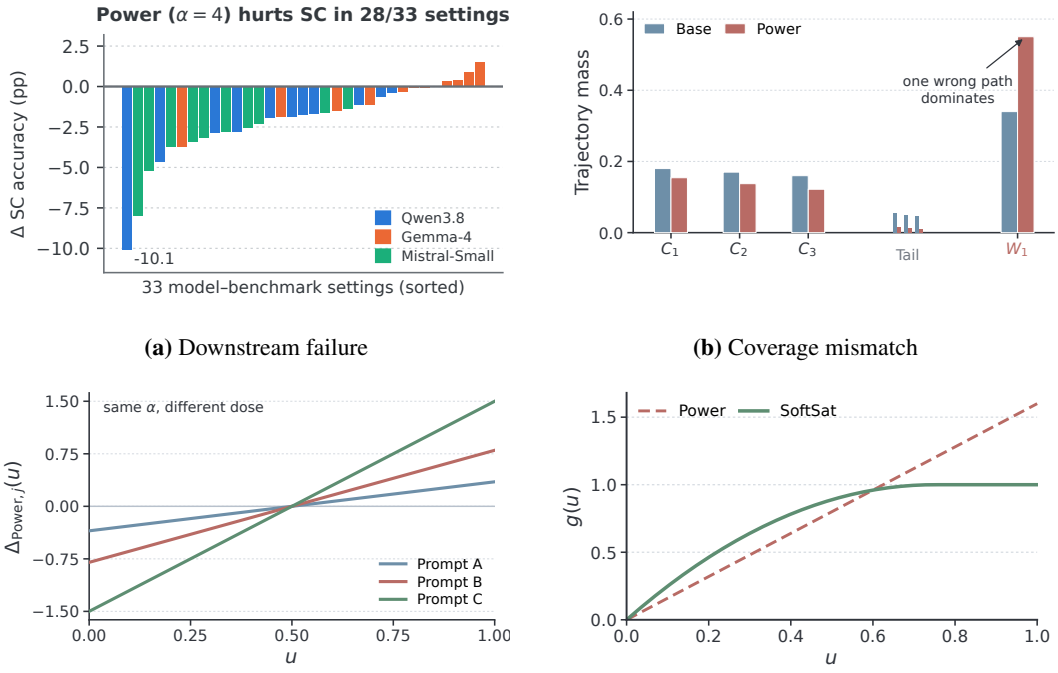
Scaling model capacity, data, and training compute has advanced language models in mathematics, code, and scientific question answering (Lewkowycz et al., 2022; Chen et al., 2021; Kaplan et al., 2020). Test-time scaling complements this approach by allocating inference compute to longer reasoning, search, or multiple candidate trajectories without changing model parameters (Snell et al., 2025; Wu et al., 2025).

Repeated sampling generates multiple reasoning trajectories and combines them through verification, selection, or aggregation (Chen et al., 2026; Khanh et al., 2026). Self-Consistency (SC), for example, selects the answer receiving the most support across independently sampled trajectories (Wang et al., 2023; Oh & Lee, 2026). Its success depends on whether the sampling distribution preserves collective support for the correct answer, beyond the presence of one strong trajectory (Tan et al., 2025; Khanh et al., 2026).

Power Sampling reshapes this distribution without training or a correctness verifier: it targets $q \propto p^{\alpha}$ over complete trajectories, amplifying the model's likelihood preferences (Huang et al., 2025; Karan & Du, 2026). Scalable Power Sampling and Power-SMC reduce the cost of approximating this sequence-level target (Ji et al., 2026; Azizi et al., 2026). Independent of the downstream rule, the target appears to offer a natural front end for multi-trajectory reasoning. This raises the question of whether amplifying trajectory likelihoods also preserves the collective answer support on which downstream inference relies.

However, increasing correct-trajectory mass on average across problems does not guarantee higher SC accuracy. On the fixed Mistral-Small HARP pool, Power raises mean correct-trajectory mass while lowering mean gold-answer margin and consensus accuracy: mass gains on already-solved problems can outweigh losses on other problems without compensating for their incorrect decisions. Across 33 model–benchmark settings, Base-pool Power weighting trails Uniform in 28, by up to 10.11 percentage points (Figure 1(a)). SC thus provides a test of whether favoring likely trajectories also improves the decisions based on their collective support.

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.



(a) Downstream failure

(b) Coverage mismatch

(c) Dose mismatch

(d) Relative-Rank SoftSat

Figure 1: **Power sharpening can weaken consensus; rank-calibrated saturation limits concentration.** Gains g define $q \propto p \exp(g)$. **(a)** Sorted Power ($\alpha = 4$) minus Uniform accuracy in 33 settings. Each readout uses the same eight Base trajectories per problem; bars compare three-run means, with model-specific colors. Power trails in 28 settings. **(b)** Schematic trajectory masses: C_1–C_3 support one correct answer, W_1 a competing wrong answer, and Tail other low-mass paths. Sharpening makes W_1 dominant. **(c)** Illustrative centered Power gains $\Delta_{\text{Power},j}(u)$ across prompts at shared α. **(d)** Gain shapes on within-prompt rank u: green is SoftSat; dashed red is a linear reference, not an exact Power gain curve.

We identify two mismatches. *Coverage mismatch* arises because Power rewards individual trajectories, whereas consensus combines answer-level support. In Figure 1(b), moderate-probability trajectories C_1–C_3 jointly support the correct answer, but sharpening magnifies the individual advantage of the strongest trajectory W_1, which supports a wrong answer. The aggregated decision can therefore flip even though correct paths remain. Likewise, pass@k measures the chance of finding a correct trajectory, not its support in consensus (Karan & Du, 2026; Yue et al., 2025; Tan et al., 2025; Khanh et al., 2026). *Dose mismatch* arises because a fixed exponent need not induce the same distributional change: different likelihood spreads can yield mild adjustment on one problem and near-collapse on another (Figure 1(c)).

These mismatches motivate controlling both the shape and strength of sharpening. We introduce *Relative-Rank SoftSat* to do so. To address coverage mismatch, SoftSat promotes moderately ranked trajectories while saturating gains at the top, limiting concentration on dominant paths. To address dose mismatch, relative rank replaces raw likelihood with a common within-prompt coordinate in $[0, 1]$, making reweighting strength more comparable across problems (Figure 1(d)).

Likelihood rank indicates which trajectories the model favors during generation, but does not directly assess whether their answers are correct. We therefore also test same-model judgments of completed trajectories as a complementary correctness signal. Because these judgments can themselves be biased, they modulate weights while retaining generation-derived rank as the common coordinate. SoftSat separately controls the gain shape.

We apply the proposed gains through weighted aggregation of the Base trajectories already generated for SC.

Our contributions are threefold:

1. We show that Power Sampling can increase across-problem mean correct-trajectory mass yet reduce SC accuracy, and analyze this behavior through coverage mismatch and dose mismatch.
2. We introduce Relative-Rank SoftSat, which replaces absolute likelihood with within-prompt rank to control dose and saturates gains on dominant trajectories to preserve coverage.
3. We connect ideal target sampling to importance-weighted aggregation over a shared Base pool, enabling a low-cost readout with asymptotic decision agreement for answer voting.

2 Related Work

2.1 Sequence-Level Distribution Sharpening

Autoregressive MCMC, Scalable Power Sampling, and Power-SMC offer different approximations to the sequence-level Power target (Karan & Du, 2026; Ji et al., 2026; Azizi et al., 2026). Sharpening studies emphasize single-output accuracy and pass@k; Levi (2025) models the latter through per-problem failure probabilities. Answer-level consensus additionally depends on how mass is divided among competing answers (Khanh et al., 2026; Tan et al., 2025).

2.2 Multi-Sample Consistency and Aggregation

Self-Consistency groups normalized answers (Wang et al., 2023), CodeT compares programs through generated tests (Chen et al., 2023), and ModeX uses response-similarity graphs (Choi & Li, 2026). Weighting methods use perplexity and pruning (RPC), reported confidence (CISC), reasoning checks (SelfCheck), or rationale quality (RASC) (Zhou et al., 2025; Taubenfeld et al., 2025; Miao et al., 2024; Wan et al., 2025). We study the conversion of likelihood into candidate influence through rank calibration and saturation, optionally modulated by self-evaluation.

3 Preliminaries

3.1 Trajectory Distributions and Power Sampling

For a fixed prompt (conditioning omitted), the Base probability and log-likelihood of a complete trajectory $\tau = (\tau_1, \ldots, \tau_T)$ are

$$\begin{aligned} p(\tau) &= \prod_{t=1}^{T} p(\tau_t \mid \tau_{<t}), \\ \ell(\tau) &= \log p(\tau) = \sum_{t=1}^{T} \log p(\tau_t \mid \tau_{<t}). \end{aligned} \tag{1}$$

Token temperature renormalizes each prefix-conditioned distribution. Power instead sharpens complete-trajectory probabilities; for $\alpha > 1$,

$$q_\alpha(\tau) = \frac{p(\tau)^\alpha}{Z_\alpha}, \qquad Z_\alpha = \sum_{\tau'} p(\tau')^\alpha. \tag{2}$$

Power-SMC uses M particles to approximate q_α, returning one trajectory per run (Azizi et al., 2026); N denotes the downstream aggregation budget.

3.2 From Trajectories to Consensus

We use self-consistency (SC) as an umbrella term for aggregation over independently sampled trajectories, including answer voting and the ModeX readout.

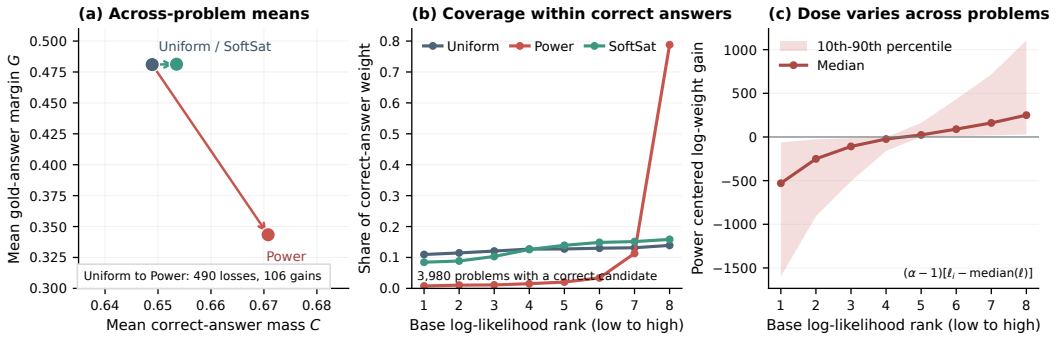


Figure 2: Mistral-Small on HARP (4,780 problems, one shared $N = 8$ Base pool; Power $\alpha = 4$, SoftSat $\beta = 1$). (a) Across-problem means of correct-answer mass C and gold-answer margin G, with paired Uniform-to-Power consensus changes; only one problem has both C increase and G decrease. (b) Correct-answer weight by Base-likelihood rank, averaged equally over 3,980 problems with a correct candidate. (c) Median and 10th–90th percentiles of median-centered Power log-weight gain across problems.

Let $A(\tau)$ be the normalized answer extracted from τ, and let $R(\tau) \in \{0, 1\}$ indicate whether its final answer passes the task evaluator; this does not certify every reasoning step. Any trajectory distribution μ induces answer mass and total correct-trajectory mass

$$Q_\mu(a) = \Pr_{\tau\sim\mu}[A(\tau) = a], \qquad C_\mu = \mathbb{E}_{\tau\sim\mu}[R(\tau)]. \tag{3}$$

When correctness is determined by a unique gold answer $a^\star$, $C_\mu = Q_\mu(a^\star)$ and its answer margin is $G_\mu = Q_\mu(a^\star) - \max_{a\neq a^\star} Q_\mu(a)$. For code, correctness may instead be determined by execution. Given N independently sampled trajectories, answer-voting SC selects

$$\widehat{a}_{\mathrm{SC}} = \arg\max_a \sum_{i=1}^{N} \mathbf{1}\{A(\tau_i) = a\}. \tag{4}$$

For executable or open-ended outputs without a canonical answer string, we use the ModeX form of SC (Choi & Li, 2026).

4 Mismatch Analysis

We reweight a shared Base pool to isolate how Power redistributes answer support; Section 5.1 connects this analysis to the ideal target. For normalized weights W_i, we measure $\widehat{Q}(a) = \sum_i W_i \mathbf{1}\{A(\tau_i) = a\}$, $\widehat{C} = \sum_i W_i R(\tau_i)$, and $\widehat{G} = \widehat{Q}(a^\star) - \max_{a\neq a^\star} \widehat{Q}(a)$. We omit hats in these finite-pool diagnostics. Coverage denotes effective answer support: finite Power exponents redistribute mass without removing trajectories from the mathematical support.

4.1 Coverage Mismatch

Pass@k is the probability that at least one correct trajectory appears among k samples. For independent draws from a trajectory distribution μ,

$$\mathrm{pass@}\,k(\mu) = 1 - (1 - C_\mu)^k. \tag{5}$$

For $k > 1$, mean pass@k depends on the distribution of problem-specific correct masses C_x, not only their mean (Levi, 2025); it does not determine the competing answer masses or gold-answer margin.

On the shared $N = 8$ pool, Power raises mean C from 0.649 to 0.671 and lowers mean G from 0.481 to 0.343, with 490 consensus regressions and 106 gains (Figure 2). Only one problem has both C increase and G decrease: the aggregate reversal reflects correct-mass gains on many already-solved

problems and losses on problems whose decisions regress (Appendix D). Among 3,980 problems with a correct candidate, the top likelihood rank receives 78.8% of Power's correct-answer weight, versus 13.9% under Uniform. In 483 of 490 regressions, the previously winning correct cluster contained multiple candidates. Power thus overturns collectively supported correct decisions even when correct candidates remain.

4.2 Dose Mismatch

Relative to Base, Power applies a multiplier proportional to $p(\tau)^{\alpha-1}$. For two trajectories from problem j, the relative log-gain is

$$\log \frac{p_j(\tau_i)^{\alpha-1}}{p_j(\tau_k)^{\alpha-1}} = (\alpha-1)\big[\ell_j(\tau_i) - \ell_j(\tau_k)\big]. \tag{6}$$

Let $L_j(u)$ be the u-quantile of Base log-likelihood for problem j. Its median-centered Power log-weight gain is

$$\Delta_{\mathrm{Power},j}(u) = (\alpha-1)\big[L_j(u) - L_j(1/2)\big]. \tag{7}$$

Figure 2(c) plots the empirical $N = 8$ counterpart of this curve at $\alpha = 4$.

Likelihood dispersion also determines the ideal target's local movement. If $\mathbb{E}_{\tau\sim p_j}[|\ell_j(\tau)|^3] < \infty$, then as $\alpha \downarrow 1$,

$$D_{\mathrm{KL}}(q_{\alpha,j} \,\|\, p_j) = \frac{(\alpha-1)^2}{2} \operatorname{Var}_{\tau\sim p_j}[\ell_j(\tau)] + O((\alpha-1)^3). \tag{8}$$

Thus likelihood variance controls local KL near $\alpha = 1$; this expansion does not quantify the change at $\alpha = 4$ (Appendix F).

5 Method

5.1 From Power Sampling to Weighted Consensus

Our readout starts from the same N Base trajectories as uniform consensus. Power-SMC+SC_N instead obtains N trajectories from independent sampler runs. To relate Base-pool weighting to the ideal Power target in equation 2, its unnormalized importance ratio is

$$r_\alpha(\tau) = p(\tau)^{\alpha-1} = \exp((\alpha-1)\ell(\tau)). \tag{9}$$

For any answer a, the ideal Power target has the exact representation

$$Q_{q_\alpha}(a) = \frac{\mathbb{E}_{X\sim p}[r_\alpha(X)\mathbf{1}\{A(X)=a\}]}{\mathbb{E}_{X\sim p}[r_\alpha(X)]}. \tag{10}$$

Independent i.i.d. pools $X_{1:N} \sim p$ and $Y_{1:N} \sim q_\alpha$ give the Base-weighted and target-sampled estimates

$$\begin{aligned} \widehat{Q}_N^{\mathrm{weight}}(a) &= \frac{\sum_{i=1}^N r_\alpha(X_i)\mathbf{1}\{A(X_i)=a\}}{\sum_{i=1}^N r_\alpha(X_i)}, \\ \widehat{Q}_N^{\mathrm{sample}}(a) &= \frac{1}{N}\sum_{i=1}^N \mathbf{1}\{A(Y_i)=a\}. \end{aligned} \tag{11}$$

Both estimate the same ideal answer mass. For $\alpha > 1$, $q_\alpha/p \le 1/Z_\alpha$, giving an $O(N^{-1})$ expected squared difference. On a finite answer space with a unique positive-margin mode, their decisions agree asymptotically (Appendix I). This connection motivates studying Power-induced changes through Base-pool weighting: we can compare weighting rules while holding the available trajectories fixed. At finite budgets, their different sampling distributions can yield different decisions; error constants depend on the density ratio. Table 7 compares Base weighting with finite-particle Power-SMC.

5.2 RELATIVE-RANK SOFTSAT

Power's gain depends on the raw log-likelihood gaps in equation 6. We instead make each trajectory's gain depend on its within-problem likelihood rank

$$u_p(\tau) = \Pr_{X\sim p}[\ell(X) \le \ell(\tau)] \in [0,1]. \tag{12}$$

We shape the rank-dependent gain with

$$\mathrm{SoftSat}_m(u) = 1 - \left(1 - \min\left\{\frac{u}{m}, 1\right\}\right)^2, \qquad 0 < m \le 1, \tag{13}$$

For $\beta \ge 0$, define the corresponding reweighting multiplier

$$r_{\beta,m}(\tau) = \exp[\beta\, \mathrm{SoftSat}_m(u_p(\tau))]\,. \tag{14}$$

The unclipped target is $q_{\beta,m}(\tau) \propto p(\tau) r_{\beta,m}(\tau)$. Rank removes likelihood spacing; SoftSat promotes moderate ranks and saturates above m. Its range $[0,1]$ bounds the target-to-Base density ratio at fixed β. The finite-pool rule below adds midranks and clipping.

5.3 FINITE-POOL WEIGHTS AND CONSENSUS READOUT

For a Base pool $\tau_{1:N}$, we use the midrank quantile of each log-likelihood,

$$\widehat{u}_i = \frac{1}{N}\left(\sum_{k=1}^{N} \mathbf{1}\{\ell(\tau_k) < \ell(\tau_i)\} + \frac{1}{2}\sum_{k=1}^{N} \mathbf{1}\{\ell(\tau_k) = \ell(\tau_i)\}\right). \tag{15}$$

For $0 < \eta < 1$ and $\kappa > 0$, we obtain the finite-pool multiplier by clipping the shaped gain,

$$\widetilde{r}_i = \mathrm{clip}_{[1-\eta,\, 1+\kappa\eta]}(\exp[\beta\, \mathrm{SoftSat}_m(\widehat{u}_i) + c])\,. \tag{16}$$

The offset c makes these multipliers mean one; their normalized values are used as candidate weights $W_{1:N}$. The lower bound retains a uniform-weight component of mass $1-\eta$, independently of the gain shape.

Unlike Power's likelihood-gap-dependent weights, rank-SoftSat uses the same midrank grid for every tie-free pool of a fixed size, yielding a finite-pool dose calibration.

Lemma 1 (Finite-pool rank calibration)**.** *Fix $N, \beta, m, \eta, \kappa$, with a shared β across candidates. For any Base pool of N candidates with distinct log-likelihoods, let $W_{(r)}$ be the weight of the r-th candidate ordered by increasing log-likelihood. The vector $(W_{(1)}, \ldots, W_{(N)})$ is the same for every such pool. In particular, its divergence from uniform weights on the sample indices,*

$$D_{\mathrm{KL}}(W_{1:N} \,\|\, \mathrm{Unif}_N) = \sum_{r=1}^{N} W_{(r)} \log\big(N W_{(r)}\big), \tag{17}$$

is independent of the prompt.

At fixed budget, rank therefore removes likelihood-gap-driven variation in the reweighting strength measured by equation 17, directly addressing the source of dose mismatch. This calibration concerns weights on sample indices; it does not equate answer outcomes or the ideal discrete target's KL. Candidate-specific β_i is excluded. Appendix H gives the proof and a conditional population analogue.

The weighted candidates are passed to a task-appropriate consensus readout $\mathcal{D}$. For answer-based tasks, this gives weighted Self-Consistency,

$$\mathcal{D}_{\mathrm{SC}}(\tau_{1:N}, W_{1:N}) = \arg\max_a \sum_{i=1}^{N} W_i \mathbf{1}\{A(\tau_i) = a\}. \tag{18}$$

For free-form responses, weighted ModeX weights graph partitioning, cluster selection, and representative selection (Appendix B). Uniform weights recover either original readout. The mode-agreement result applies to answer voting; ModeX is evaluated empirically.

Algorithm 1 Relative-Rank SoftSat with a Weighted Consensus Readout

Require: Prompt x, Base model p_θ, budget N, readout $\mathcal{D}$, parameters (m, β, η, κ)
Ensure: Consensus output $\widehat{y}$

1: Draw $\tau_i \overset{\text{iid}}{\sim} p_\theta(\cdot \mid x)$ and compute $\ell_i \leftarrow \log p_\theta(\tau_i \mid x)$ for $i = 1, \ldots, N$
2: Set $\widehat{u}_i \leftarrow \big(\#\{j : \ell_j < \ell_i\} + \frac{1}{2}\#\{j : \ell_j = \ell_i\}\big)/N$
3: Set $g_i \leftarrow \beta\, \mathrm{SoftSat}_m(\widehat{u}_i)$ for each candidate
4: Find c by bisection so that $N^{-1}\sum_i \mathrm{clip}_{[1-\eta,\, 1+\kappa\eta]}(e^{g_i+c}) = 1$
5: Set $\widetilde{r}_i \leftarrow \mathrm{clip}_{[1-\eta,\, 1+\kappa\eta]}(e^{g_i+c})$
6: Normalize $W_i \leftarrow \widetilde{r}_i / \sum_j \widetilde{r}_j$
7: **return** $\widehat{y} \leftarrow \mathcal{D}(\tau_{1:N}, W_{1:N})$

Optional self-evaluation signal. To assess correctness beyond likelihood, the same model labels each complete trajectory A (fully correct) or B (otherwise), seeing the problem but no gold answer or other candidates. Labels preserve $\widehat{u}_i$ but can reorder weights, replacing the shared strength in equation 16 and Algorithm 1 with

$$\beta_i = \begin{cases} \beta, & s_i = A, \\ \beta^{-1}, & s_i = B. \end{cases} \tag{19}$$

We use $\beta = 1.7$; $\beta = 1$ recovers main-table SoftSat. Clipping, normalization, and readout are unchanged. Each one-token judgment requires rereading the problem and trajectory.

6 Experiments

6.1 Experimental Setup

Benchmarks and models. The main table covers BigCodeBench Full-Instruct (1,140 problems) (Zhuo et al., 2025), LiveCodeBench v6 (1,055) (Jain et al., 2025), non-empty-answer Omni-MATH-2 (4,172) (Ballon et al., 2026), GPQA-Diamond (198) (Rein et al., 2024), and a fixed MMLU-Pro subset (1,000) (Wang et al., 2024); Appendix A covers six additional benchmarks. We evaluate Qwen3.8-27B (medium reasoning effort), Gemma-4-31B-it, and Mistral-Small-3.2-24B-Instruct-2506, reporting means and sample standard deviations over three recorded runs per setting (provenance in Appendix B).

Generation and readout. All main-table readouts share eight Base trajectories per problem, sampled at temperature 1 and top-$k = 50$ with benchmark prompts and model chat templates; maximum generation length ranges from 768 to 65,536 tokens. Scores sum token log-probabilities before top-k renormalization; the ideal-target analysis assumes untruncated Base draws. *Uniform* assigns equal weights; *Power* uses $\alpha = 4$ following Power-SMC (Azizi et al., 2026); *SoftSat* uses $(m, \beta, \eta, \kappa) = (0.75, 1, 0.25, 64)$. *SoftSat+SelfEval* adds one same-model A/B assessment per trajectory with $\beta = 1.7$, without generating new solutions (Appendix B).

Code benchmarks use ModeX (Choi & Li, 2026) and execution scoring. Math benchmarks aggregate equivalent extracted answers and use their benchmark scoring procedures; Omni-MATH-2 uses its official judge prompt served by gpt-5.6-luna. GPQA-Diamond, MMLU-Pro, and SuperGPQA aggregate extracted option choices.

6.2 Main Results

We use Uniform as the unsharpened reference and assess SoftSat primarily by how much of Power's loss it recovers. Across 33 settings (Tables 1, 2), Power trails Uniform in 28, by up to 10.11 points on Qwen3.8 LiveCodeBench and 8.00 on Mistral-Small HARP. Gains in some Gemma settings show that sharpening can help, but these widespread losses undermine direct likelihood amplification as a default consensus rule.

SoftSat improves on Power in 30 settings, ties one, and trails two, reducing the largest deficit to Uniform from 10.11 to 1.96 points. Against Uniform it wins 18 and loses 14, with essentially

Table 1: Benchmark scores (%) with eight Base trajectories per problem, reported as mean±sample standard deviation over three recorded runs. *Single* uses the first sampled trajectory; the remaining columns apply task-specific consensus to the same pool. Bold marks the best multi-sample mean in each row. Results on the six remaining benchmarks appear in Appendix A.

Benchmark	Model	Single	Uniform	Power (α=4)	SoftSat (β=1)	+SelfEval
BigCodeBench	Qwen3.8-27B	40.58±0.31	45.58±0.33	44.42±0.44	46.26±0.77	**47.28±0.49**
	Gemma-4-31B	40.47±1.43	43.89±0.35	44.77±0.28	44.74±0.32	**47.57±0.94**
	Mistral-Small-24B	39.82±0.53	44.04±1.00	42.63±0.53	44.42±1.06	**45.15±0.63**
LiveCodeBench	Qwen3.8-27B	77.91±0.66	83.63±0.57	73.52±0.77	81.67±1.14	**86.16±0.62**
	Gemma-4-31B	83.06±0.47	**84.01±0.14**	82.88±0.44	83.67±0.39	83.54±0.61
	Mistral-Small-24B	27.65±1.54	30.77±0.63	28.44±0.53	31.53±0.39	**32.99±0.28**
Omni-MATH-2	Qwen3.8-27B	68.94±0.58	80.13±0.42	77.29±0.44	79.03±0.33	**80.54±0.11**
	Gemma-4-31B	76.77±0.05	**80.37±0.10**	76.64±0.13	80.15±0.10	80.08±0.07
	Mistral-Small-24B	41.69±4.12	48.48±4.48	44.74±4.08	48.80±4.40	**49.07±4.51**
GPQA-Diamond	Qwen3.8-27B	84.34±0.51	**86.87±2.20**	85.19±1.17	86.70±1.62	86.70±1.62
	Gemma-4-31B	78.28±0.51	78.45±0.29	**79.97±0.29**	78.45±0.77	78.62±0.58
	Mistral-Small-24B	50.17±0.77	55.05±1.01	49.83±1.77	54.88±2.33	**55.22±2.04**
MMLU-Pro	Qwen3.8-27B	82.27±0.12	84.60±0.40	84.20±0.44	**84.73±0.40**	84.70±0.40
	Gemma-4-31B	84.20±0.66	84.57±0.25	84.47±0.21	84.53±0.42	**84.67±0.29**
	Mistral-Small-24B	66.50±0.79	72.60±0.80	69.43±0.67	73.00±0.56	**73.30±0.40**

unchanged equal-setting mean accuracy. Its benefit is recovery from Power's losses while retaining likelihood-based candidate preferences.

Adding self-evaluation to the same eight trajectories improves on SoftSat in 23 of 33 settings. The component ablations below examine this correctness signal separately from likelihood-based weighting.

Power's losses are not specific to $\alpha = 4$: on the separate seed-1/2 pools, $\alpha = 1.05$ already trails Uniform in 27 of 33 settings, and every tested $\alpha \geq 1.5$ is within 0.31 points of $\alpha = 4$ (Appendix C). This early decision plateau is consistent with large likelihood gaps (Section 4.2); varying the exponent within it yields little recovery.

Component ablations. We compare Uniform, SoftSat, SE-only, and their combination to distinguish each component's contribution. On the frozen seed-1/2 pools, SE-only exceeds Uniform in 28 of 33 settings. Joint weighting improves on SE-only in 18 and declines in 15: the judgment signal is useful alone, while combining it with likelihood-based weighting has setting-dependent benefits. Figure 3(a) illustrates one improvement; Table 4 reports all settings.

To test whether saturation contributes beyond rank calibration, we replace SoftSat with linear rank while fixing β, clipping, and the readout. SoftSat wins in 18 settings, loses in 11, and ties in four (Figure 3(b)). Saturation helps more often, though these shapes also induce different weighting strengths.

6.3 Further Results

Budget sensitivity. Increasing the candidate budget does not by itself remove Power's losses. Power slightly exceeds Uniform at $N = 4$ on the nested HARP pools, but solves 157 problems versus Uniform's 191 at $N = 128$ (Figure 3(c)). Effective sample size stays near one: added candidates receive little decision weight. SoftSat retains broader weight and stays close to Uniform at larger budgets, gaining slightly at $N = 64, 128$ but losing at $N = 32$ (Appendix E).

Coverage repair. On the HARP diagnostic pool, replacing Power with SoftSat reduces the share of correct-answer weight assigned to the top likelihood rank from 78.8% to 15.8% (Figure 2). Mean gold-answer margin rises from 0.343 to 0.481, matching Uniform, and consensus accuracy rises from 67.05% to 75.29%. With the candidate trajectories held fixed, this recovery accompanies a

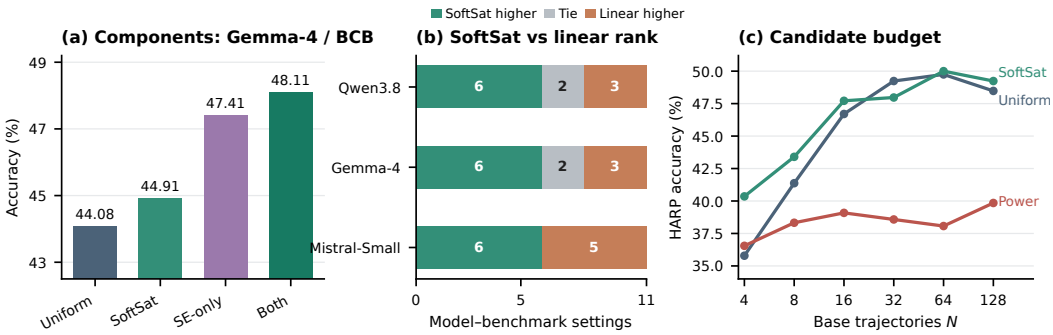


Figure 3: Component and budget diagnostics. (a) Gemma-4 BigCodeBench accuracy, seed-1/2 mean on frozen $N = 8$ pools; truncated vertical axis. (b) SoftSat versus linear rank across 33 settings, with clipping and readout fixed. (c) Uniform, Power ($\alpha = 4$), and fixed SoftSat ($\beta = 1$) on a separate 394-problem Mistral-Small HARP level-5/6 slice. Budgets are nested prefixes of one 128-candidate pool; results do not represent full HARP.

broader allocation of weight to existing correct paths. Rank-linear is slightly more accurate on this pool (Appendix B).

Weighted realization. The shared-target analysis (Section 5.1) motivates comparing Base weighting with repeated Power-SMC+SC_8. Accuracy differs by at most 2.02 points across nine reported single-seed settings on GPQA-Diamond, MMLU-Pro, and MATH-500, with neither consistently higher (Table 7). This complements the analysis within these settings; it does not establish general interchangeability. Both aggregate eight outputs, but Base weighting reuses eight ordinary trajectories, whereas Power-SMC requires eight independent eight-particle runs, each returning one trajectory. The comparison matches readout size, not generation work (Appendix G).

7 Conclusion

Our work identifies structural limitations of Power Sampling as a mechanism for improving reasoning. Amplifying individual trajectory likelihoods can weaken collective support for correct solutions, while a fixed exponent produces uneven distributional changes across problems. We use self-consistency to expose the consequences of these coverage and dose mismatches across models and tasks. Relative-Rank SoftSat addresses both through rank calibration and saturation, recovering much of Power's accuracy loss on shared Base pools; optional self-evaluation supplies an additional correctness signal. These findings challenge the assumption that stronger preference for likely trajectories reliably yields better reasoning. They motivate inference-time distribution shaping that balances preference amplification with the preservation of useful reasoning support, and calibrates its strength across problems.

AI use statement

Generative AI tools assisted with literature review, manuscript drafting and translation, mathematical derivations and proofs, experimental design, implementation, result analysis, and figure preparation. LLM-based answer judging is described in Section 6. The authors reviewed the AI-assisted material against code, recorded experiments, and cited sources and take responsibility for the final manuscript.

Reproducibility statement

Sections 3–5 define the targets and readouts. Section 6 and the appendix provide protocols, proofs, and diagnostics; the code artifact implements the finite-pool readouts.

REFERENCES

Seyedarmin Azizi, Erfan Baghaei Potraghloo, Minoo Ahmadi, Souvik Kundu, and Massoud Pedram. Power-SMC: Low-latency sequence-level power sampling for training-free LLM reasoning, 2026. arXiv preprint arXiv:2602.10273.

Marthe Ballon, Andres Algaba, Brecht Verbeken, and Vincent Ginis. Benchmarks saturate when the model gets smarter than the judge, 2026. arXiv preprint arXiv:2601.19532.

Bei Chen, Fengji Zhang, Anh Nguyen, Daoguang Zan, Zeqi Lin, Jian-Guang Lou, and Weizhu Chen. CodeT: Code generation with generated tests. In *International Conference on Learning Representations*, 2023.

Jianhao Chen, Zishuo Xun, Bocheng Zhou, Han Qi, Hangfan Zhang, Qiaosheng Zhang, Yang Chen, Wei Hu, Yuzhong Qu, and Shuyue Hu. Do we truly need so many samples? multi-LLM repeated sampling efficiently scales test-time compute. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 20083–20091, 2026. doi: 10.1609/aaai.v40i24.39094.

Mark Chen et al. Evaluating large language models trained on code, 2021. arXiv preprint arXiv:2107.03374.

Hyeong Kyu Choi and Sharon Li. ModeX: Evaluator-free best-of-n selection for open-ended generation. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics*, 2026.

Audrey Huang, Adam Block, Dylan Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan T. Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *International Conference on Learning Representations*, 2025.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and contamination free evaluation of large language models for code. In *International Conference on Learning Representations*, 2025.

Xiaotong Ji, Rasul Tutunov, Matthieu Zimmer, and Haitham Bou Ammar. Scalable power sampling: Unlocking efficient, training-free reasoning for LLMs via distribution sharpening. In *International Conference on Machine Learning*, 2026.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B. Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models, 2020. arXiv preprint arXiv:2001.08361.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. In *International Conference on Learning Representations*, 2026.

Ly Tran Ho Khanh, Dongxuan Zhu, Man-Chung Yue, and Viet Anh Nguyen. Test-time diverse reasoning by riemannian activation steering. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 31429–31437, 2026. doi: 10.1609/aaai.v40i37.40407.

Noam Itzhak Levi. A simple model of inference scaling laws. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pp. 33984–33998, 2025. URL `https://proceedings.mlr.press/v267/levi25a.html`.

Aitor Lewkowycz, Anders Andreassen, David Dohan, Ethan Dyer, Henryk Michalewski, Vinay Ramasesh, Ambrose Slone, Cem Anil, Imanol Schlag, Theo Gutman-Solo, Yuhuai Wu, Behnam Neyshabur, Guy Gur-Ari, and Vedant Misra. Solving quantitative reasoning problems with language models. In *Advances in Neural Information Processing Systems*, volume 35, 2022.

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

Jiawei Liu, Chunqiu Steven Xia, Yuyao Wang, and Lingming Zhang. Is your code generated by ChatGPT really correct? rigorous evaluation of large language models for code generation. In *Advances in Neural Information Processing Systems*, 2023.

M-A-P Team, Xinrun Du, et al. SuperGPQA: Scaling LLM evaluation across 285 graduate disciplines, 2025. arXiv preprint arXiv:2502.14739.

Sadegh Mahdavi, Muchen Li, Kaiwen Liu, Christos Thrampoulidis, Leonid Sigal, and Renjie Liao. Leveraging online olympiad-level math problems for LLMs training and contamination-resistant evaluation. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, 2025. URL https://proceedings.mlr.press/v267/mahdavi25a.html.

Ning Miao, Yee Whye Teh, and Tom Rainforth. SelfCheck: Using LLMs to zero-shot check their own step-by-step reasoning. In *International Conference on Learning Representations*, 2024.

Jungsuk Oh and Jay-Yoon Lee. Latent self-consistency for reliable majority-set selection in short- and long-answer reasoning. In *Proceedings of the AAAI Conference on Artificial Intelligence*, volume 40, pp. 32591–32599, 2026. doi: 10.1609/aaai.v40i38.40536.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R. Bowman. GPQA: A graduate-level google-proof Q&A benchmark. In *First Conference on Language Modeling*, 2024.

Julian Schnitzler, Xanh Ho, Jiahao Huang, Florian Boudin, Saku Sugawara, and Akiko Aizawa. MoreHopQA: More than multi-hop reasoning. *arXiv preprint arXiv:2406.13397*, 2024. doi: 10.48550/arXiv.2406.13397. URL https://arxiv.org/abs/2406.13397.

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling LLM test-time compute optimally can be more effective than scaling parameters for reasoning. In *International Conference on Learning Representations*, 2025.

Hexiang Tan, Fei Sun, Sha Liu, Du Su, Qi Cao, Xin Chen, Jingang Wang, Xunliang Cai, Yuanzhuo Wang, Huawei Shen, and Xueqi Cheng. Too consistent to detect: A study of self-consistent errors in LLMs. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4755–4765, 2025. doi: 10.18653/v1/2025.emnlp-main.238.

Amir Taubenfeld, Tom Sheffer, Eran Ofek, Amir Feder, Ariel Goldstein, Zorik Gekhman, and Gal Yona. Confidence improves self-consistency in LLMs. In *Findings of the Association for Computational Linguistics: ACL 2025*, 2025.

Guangya Wan, Yuqi Wu, Jie Chen, and Sheng Li. Reasoning aware self-consistency: Leveraging reasoning paths for efficient LLM sampling. In *Proceedings of the 2025 Conference of the Nations of the Americas Chapter of the Association for Computational Linguistics: Human Language Technologies*, pp. 3613–3635, 2025. doi: 10.18653/v1/2025.naacl-long.184.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc V. Le, Ed H. Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. In *International Conference on Learning Representations*, 2023.

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 *Advances in Neural Information Processing Systems*, 2024.

Yangzhen Wu, Zhiqing Sun, Shanda Li, Sean Welleck, and Yiming Yang. Inference scaling laws: An empirical analysis of compute-optimal inference for LLM problem-solving. In *International Conference on Learning Representations*, 2025.

Albert S. Yue, Lovish Madaan, Ted Moskovitz, DJ Strouse, and Aaditya K. Singh. HARP: A challenging human-annotated math reasoning benchmark, 2024. arXiv preprint arXiv:2412.08819.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems*, 2025.

Zhi Zhou, Yuhao Tan, Zenan Li, Yuan Yao, Lan-Zhe Guo, Yu-Feng Li, and Xiaoxing Ma. A theoretical study on bridging internal probability and self-consistency for LLM reasoning. In *Advances in Neural Information Processing Systems*, 2025.

Terry Yue Zhuo, Minh Chien Vu, Jenny Chim, Han Hu, Wenhao Yu, Ratnadira Widyasari, Imam Nur Bani Yusuf, Haolan Zhan, Junda He, Indraneil Paul, Simon Brunner, Chen Gong, James Hoang, Armel Zebaze, Xiaoheng Hong, Wen-Ding Li, Jean Kaddour, Ming Xu, Zhihan Zhang, Prateek Yadav, Naman Jain, Alex Gu, Zhoujun Cheng, Jiawei Liu, Qian Liu, Zijian Wang, Binyuan Hui, Niklas Muennighoff, David Lo, Daniel Fried, Xiaoning Du, Harm de Vries, and Leandro von Werra. BigCodeBench: Benchmarking code generation with diverse function calls and complex instructions. In *International Conference on Learning Representations*, 2025. URL https://openreview.net/forum?id=YrycTjllL0.

Table 2: Benchmark scores (%) on the six remaining benchmarks, in the format of Table 1: mean±sample standard deviation over three recorded runs; bold marks the best multi-sample mean per row.

Benchmark	Model	Single	Uniform	Power (α=4)	SoftSat (β=1)	+SelfEval
HumanEval+	Qwen3.8-27B	85.37±3.71	89.63±1.22	84.96±1.86	88.82±1.41	**91.87±0.93**
	Gemma-4-31B	90.04±2.46	89.23±0.93	89.63±2.20	89.63±2.20	**91.67±1.27**
	Mistral-Small-24B	79.67±0.70	82.72±1.53	81.10±2.11	81.91±1.53	**82.93±0.00**
MBPP+	Qwen3.8-27B	64.64±0.93	71.16±0.53	69.31±1.65	71.52±0.31	**75.31±0.93**
	Gemma-4-31B	73.81±0.79	74.43±1.33	74.78±1.00	74.96±0.40	**75.75±1.00**
	Mistral-Small-24B	61.46±0.93	64.02±0.95	64.02±0.70	64.37±0.40	**65.34±0.46**
MATH-500	Qwen3.8-27B	95.73±1.17	97.67±0.58	95.93±0.58	**97.80±0.40**	**97.80±0.40**
	Gemma-4-31B	96.67±0.12	**96.87±0.42**	96.53±0.23	96.80±0.35	96.80±0.35
	Mistral-Small-24B	86.47±0.81	**91.80±0.60**	88.40±1.04	91.67±0.46	**91.80±0.60**
HARP	Qwen3.8-27B	81.91±0.13	90.39±0.21	87.58±0.16	**90.47±0.24**	90.33±0.30
	Gemma-4-31B	87.63±0.32	**89.96±0.02**	88.06±0.07	89.92±0.04	89.90±0.06
	Mistral-Small-24B	65.29±0.50	75.26±0.16	67.26±0.25	75.45±0.14	**75.72±0.12**
LiveAoPSBench	Qwen3.8-27B	53.20±0.17	56.24±0.24	54.29±0.28	56.32±0.18	**56.42±0.15**
	Gemma-4-31B	55.14±0.12	56.78±0.14	55.27±0.19	**56.80±0.12**	56.74±0.18
	Mistral-Small-24B	40.46±0.19	46.60±0.08	43.76±0.36	46.95±0.08	**47.05±0.11**
SuperGPQA	Qwen3.8-27B	59.63±1.03	61.77±0.86	61.10±0.46	**61.80±0.82**	61.73±0.71
	Gemma-4-31B	60.23±0.75	60.93±0.61	60.87±0.21	60.90±0.61	**61.03±0.51**
	Mistral-Small-24B	36.67±0.70	41.37±0.35	38.83±0.76	41.20±0.89	**41.47±0.81**

A Results on the Remaining Benchmarks

The six additional benchmarks are HumanEval+ (164 problems) and MBPP+ (378) (Liu et al., 2023), MATH-500 (500) (Lightman et al., 2024), HARP (4,780) (Yue et al., 2024), a frozen 2024 LiveAoPSBench slice (5,328) (Mahdavi et al., 2025), and stratified SuperGPQA (1,000) (M-A-P Team et al., 2025). Table 2 uses the main-table budget and weighting rules. Power trails Uniform in 15 of 18 settings; SoftSat's largest deficit is 0.81 points. SoftSat+SelfEval exceeds Uniform in 12 settings and ties once.

B Self-Evaluation and Readout Ablations

For each frozen Base trajectory, the generating model reads only the problem and that trajectory, then outputs A (fully correct) or B (otherwise) without a rationale. It sees no gold answer or other candidates. Saved $P(A)$ values do not enter the weights. We use $N = 8$ and the hard-label rule in equation 19; the readout and scoring match pure SoftSat on each pool. The coefficient $\beta = 1.7$ was chosen on an earlier 498-problem Ministral LiveAoPSBench pool and then fixed for the main tables. Across 21 archived seed-0 stages, median run-level mean assessment length was 905 input tokens (range 384–5,383) and one output token.

Table 3: Single-pool HARP readout ablations for Mistral-Small (correct problems out of 4,780). All rows use the same eight Base trajectories and official final-answer scoring.

Readout	Correct
Uniform	3,589
Rank-linear (βu, no saturation)	3,610
SoftSat ($\beta = 1$)	3,599
SoftSat (all $\beta = 1.7$)	3,598
SoftSat (all $\beta = 1/1.7$)	3,604
SE-only (formula)	3,611
Direct A/B vote	3,611
SoftSat + hard A/B	**3,614**

The rank-linear row replaces $\mathrm{SoftSat}_m(u)$ by u in the gain and keeps the same clipping and normalization. It exceeds SoftSat by 11 HARP problems; saturation does not help this pool in isolation. SE-only fixes the rank utility to one inside the clipped exponential formula, while direct A/B vote normalizes the two label coefficients without that formula. Both solve 3,611 problems, but this equal total does not imply identical selected answers or an advantage for either implementation. Hard A/B judgments mark 24,307/33,482 A candidates and 507/4,758 B candidates as correct under the *official final-answer checker*; this is not a measure of complete reasoning correctness. Among 1,543 problems with mixed labels, SoftSat and SoftSat+A/B solve 803 and 819, respectively. Among 3,180 all-A problems they solve 2,793 and 2,792; among 57 all-B problems both solve three. The fixed-strength rows show why an A/B gain cannot be attributed solely to changing the common strength.

On Qwen3.8-27B BigCodeBench, rank-linear, SoftSat, and SoftSat+A/B solve 516, 520, and 534 of 1,140 problems in the fixed $N = 8$ pool. The rank-linear selection changes 377 winners relative to SoftSat; 65 newly selected programs were executed with the official evaluator. Two previously passing cached programs timed out in the repeated execution, so the comparable 516 count uses their original recorded outcomes and the new executions for the 65 unseen winners.

The two seed-1/2 pools in Table 4 are separate from the HARP seed-0 pool above. In the 33 setting means, SE-only exceeds Uniform in 28 settings, and the joint readout exceeds SE-only in 18. SoftSat exceeds rank-linear in 18, ties in four, and trails in 11. Direct A/B vote exceeds formula SE-only in 14 settings, ties in six, and trails in 13; the data do not isolate a general benefit from the clipped exponential treatment of the same hard labels. Raising the common SoftSat strength to $\beta = 1.7$, without A/B labels, improves seven settings, ties 13, and declines in 13 relative to $\beta = 1$; it does not reproduce the 23 improvements when adding A/B to SoftSat.

Recorded runs. Recorded implementation differences include A/B log-probability lookup with retries versus direct label-token requests, and a rebuilt BigCodeBench execution sandbox for runs 1–2. Mistral run-0 scores use archived per-problem evaluations. Standard deviations summarize these runs, not isolated seed variance.

Gain controls. Relative rank removes raw likelihood spacing from the gain, while m sets the rank at which the gain saturates. The lower bound retains a uniform-weight component of mass $1 - \eta$, equal to 75% in our main configuration. At $N = 8$, the mean-one constraint and this lower bound imply $\widetilde{r}_i \le 8 - 7(0.75) = 2.75$, so the configured upper bound $1 + \kappa\eta = 17$ is inactive. Thus the floor and the rank-shaped gain, rather than the upper clipping bound, control concentration in this setting.

Self-evaluation implementation. The recorded A/B prompt asks whether the proposed solution has no substantive reasoning error and correctly answers the problem, and instructs the model not to use style, length, fluency, or expressed confidence as correctness evidence. The input contains only this instruction, the problem, and one proposed solution. Decoding is greedy and restricted to the two label tokens; the generated label sets β_i. The earlier implementation retrieves the A/B log-probabilities from a returned top-logprob list, retrying with a larger list when needed; the later implementation requests the two label-token log-probabilities directly. Both implement the same judgment rule. These lookup differences concern score collection, not access to external evidence. The unified scorer rejects an overlength assessment input rather than silently truncating the solution.

Weighted ModeX readout. Let S_{ij} be the ModeX text similarity: a weighted sum of unigram, bigram, and trigram Jaccard similarities with coefficients 0.5, 0.25, and 0.25, after lowercasing and whitespace tokenization. For the active candidates, the graph has off-diagonal interactions $W_i W_j S_{ij}/\overline{W}^2$, where $\overline{W}$ is their mean weight; self-loops are removed for partitioning. We partition by the Fiedler vector of the unnormalized graph Laplacian, using zero as the threshold, or its median if all entries have the same sign. When cut conductance is below 0.8, recursion retains the cluster with larger total candidate weight; equal masses are resolved by total weighted degree, then fixed partition order. Recursion stops at the conductance threshold or when no nonempty strict reduction is possible. Within the final cluster C, we select the candidate maximizing $\sum_{j \in C} W_j S_{ij}$, including self-similarity. Final ties favor larger candidate weight, then the earlier source index.

Table 4: Component readouts on the frozen $N = 8$ Base pools (accuracy, %). Each entry averages the two recorded seed-1/2 pools; all methods within a seed share candidates and official task scoring. Rank-linear replaces only the SoftSat utility by u; SE-only retains the hard A/B coefficient and the clipped exponential readout while setting the rank utility to one. Joint denotes SoftSat+SelfEval. These descriptive two-pool values should not be read as uncertainty estimates for the three-run main table.

Benchmark	Model	Uniform	Rank-linear	SoftSat	SE-only	Joint
BigCodeBench	Qwen3.8-27B	45.70	46.58	46.58	46.93	47.50
	Gemma-4-31B	44.08	44.56	44.91	47.41	48.11
	Mistral-Small-24B	44.61	44.52	44.91	45.22	45.35
LiveCodeBench	Qwen3.8-27B	83.60	74.17	81.14	86.92	86.21
	Gemma-4-31B	84.08	83.55	83.89	84.36	83.89
	Mistral-Small-24B	30.85	30.76	31.37	33.22	33.13
Omni-MATH-2	Qwen3.8-27B	79.89	78.57	78.99	81.34	80.55
	Gemma-4-31B	80.32	79.91	80.15	80.36	80.08
	Mistral-Small-24B	51.07	51.56	51.33	51.53	51.67
GPQA-Diamond	Qwen3.8-27B	86.11	86.11	86.11	86.62	86.11
	Gemma-4-31B	78.28	78.54	78.03	78.54	78.28
	Mistral-Small-24B	54.55	53.79	53.54	55.56	54.04
MMLU-Pro	Qwen3.8-27B	84.60	84.60	84.70	84.65	84.70
	Gemma-4-31B	84.55	84.55	84.60	84.60	84.75
	Mistral-Small-24B	72.20	72.75	72.95	72.75	73.30
HumanEval+	Qwen3.8-27B	89.02	88.11	88.41	91.46	91.77
	Gemma-4-31B	89.33	90.55	89.94	90.85	91.46
	Mistral-Small-24B	82.62	82.32	82.01	82.32	82.93
MBPP+	Qwen3.8-27B	71.43	71.96	71.69	74.47	75.26
	Gemma-4-31B	75.13	75.13	75.13	75.93	76.32
	Mistral-Small-24B	63.62	63.76	64.15	64.55	65.48
MATH-500	Qwen3.8-27B	97.50	97.70	97.80	97.50	97.80
	Gemma-4-31B	96.80	96.60	96.70	96.90	96.70
	Mistral-Small-24B	91.80	91.70	91.80	92.20	92.10
HARP	Qwen3.8-27B	90.40	90.74	90.50	90.33	90.38
	Gemma-4-31B	89.97	89.92	89.91	90.01	89.88
	Mistral-Small-24B	75.36	75.51	75.52	75.83	75.77
LiveAoPSBench	Qwen3.8-27B	56.38	56.41	56.42	56.51	56.48
	Gemma-4-31B	56.83	56.82	56.83	56.82	56.81
	Mistral-Small-24B	46.56	47.09	46.95	47.03	47.05
SuperGPQA	Qwen3.8-27B	61.30	61.60	61.35	61.25	61.35
	Gemma-4-31B	61.00	61.05	61.05	61.20	61.25
	Mistral-Small-24B	41.55	41.85	41.55	42.10	41.70

Mass and final-score ties use tolerance 10^{-15}, and the conductance denominator includes 10^{-10} for zero-volume cuts. Candidate selection uses no execution results or correctness labels.

C Power-Exponent Sensitivity

To test whether the failures at $\alpha = 4$ are specific to that exponent, we recompute the Power weights on the same frozen $N = 8$ candidate pools for $\alpha \in \{1, 1.05, 1.1, 1.25, 1.5, 2, 3, 4, 6, 8\}$, keeping the task-specific readout and evaluators fixed. The $\alpha = 1$ and $\alpha = 4$ endpoints reproduce the Uniform and Power scores on recorded runs 1–2. At $\alpha = 1.05$, Power already trails Uniform in 27 of 33 settings; its score is within 0.5 percentage points of $\alpha = 4$ in 24. Every tested $\alpha \geq 1.5$ differs from $\alpha = 4$ by at most 0.31 points in every setting. This rapid saturation describes the finite eight-candidate weighted readout; it does not imply that the ideal Power target or other candidate budgets are insensitive to α.

Table 5: Power-exponent sweep on the fixed $N = 8$ Base pools (%), mean over recorded runs 1–2. $\alpha = 1$ reproduces Uniform and $\alpha = 4$ reproduces Power on these pools. The tested exponents 2, 3, and 6 are omitted for space; each differs from $\alpha = 4$ by at most 0.31 percentage points across all settings.

Benchmark	Model	α=1	1.05	1.1	1.25	1.5	4	8
BigCodeBench	Qwen3.8-27B	45.70	44.91	44.78	44.65	44.65	44.65	44.65
	Gemma-4-31B	44.08	44.61	44.47	44.39	44.61	44.61	44.61
	Mistral-Small-24B	44.61	43.82	43.16	42.85	42.76	42.89	42.85
LiveCodeBench	Qwen3.8-27B	83.60	73.08	73.08	73.08	73.08	73.08	73.08
	Gemma-4-31B	84.08	83.13	83.13	83.13	83.13	83.13	83.13
	Mistral-Small-24B	30.85	28.82	28.53	28.53	28.20	28.39	28.39
Omni-MATH-2	Qwen3.8-27B	79.89	77.40	77.38	77.36	77.36	77.36	77.36
	Gemma-4-31B	80.32	77.24	76.83	76.65	76.63	76.64	76.64
	Mistral-Small-24B	51.07	47.44	47.24	47.15	47.10	47.09	47.09
GPQA-Diamond	Qwen3.8-27B	86.11	84.85	84.85	84.85	84.85	84.85	84.85
	Gemma-4-31B	78.28	79.55	80.05	80.05	80.05	79.80	79.80
	Mistral-Small-24B	54.55	51.77	50.51	49.75	49.75	49.75	49.75
MMLU-Pro	Qwen3.8-27B	84.60	84.45	84.50	84.35	84.30	84.30	84.30
	Gemma-4-31B	84.55	84.50	84.60	84.60	84.50	84.50	84.50
	Mistral-Small-24B	72.20	70.45	69.80	69.65	69.65	69.60	69.60
HumanEval+	Qwen3.8-27B	89.02	85.67	86.28	85.98	85.98	85.98	85.98
	Gemma-4-31B	89.33	90.85	90.55	90.85	90.55	90.55	90.55
	Mistral-Small-24B	82.62	82.32	82.32	81.71	81.40	81.71	81.71
MBPP+	Qwen3.8-27B	71.43	69.97	69.58	69.18	69.05	69.05	69.05
	Gemma-4-31B	75.13	74.74	75.13	75.13	75.26	75.13	75.13
	Mistral-Small-24B	63.62	64.02	64.42	64.81	64.42	64.42	64.42
MATH-500	Qwen3.8-27B	97.50	96.20	96.20	95.70	95.60	95.60	95.60
	Gemma-4-31B	96.80	96.40	96.30	96.40	96.40	96.40	96.40
	Mistral-Small-24B	91.80	89.50	89.40	89.20	89.10	89.00	89.00
HARP	Qwen3.8-27B	90.40	88.05	87.80	87.65	87.59	87.58	87.58
	Gemma-4-31B	89.97	88.26	88.14	88.08	88.03	88.02	88.02
	Mistral-Small-24B	75.36	69.17	68.48	67.80	67.54	67.36	67.36
LiveAoPSBench	Qwen3.8-27B	56.38	54.20	54.19	54.19	54.19	54.18	54.18
	Gemma-4-31B	56.83	55.36	55.17	55.14	55.18	55.17	55.17
	Mistral-Small-24B	46.56	44.13	43.94	43.82	43.81	43.81	43.81
SuperGPQA	Qwen3.8-27B	61.30	61.40	61.30	61.40	61.35	61.35	61.35
	Gemma-4-31B	61.00	61.40	61.10	60.85	60.95	60.90	60.90
	Mistral-Small-24B	41.55	40.70	40.10	39.40	39.10	39.10	39.10

D Additional HARP Mechanism Checks

All candidate answer clusters in the 4,780-problem Mistral-Small HARP pool were scored with the official HARP checker; no isolated comparison hit its time or memory guard. Power reverses 490

Uniform-correct decisions and repairs 106 Uniform-wrong decisions. Of those 490 reversals, 483 previously winning correct-answer clusters contain at least two sampled trajectories, and 430 Power winners are the highest-Base-likelihood candidate. At the top rank, the median centered Power log-weight gain is 250, with 5th–95th percentiles of 19.7–1,422 across problems. Figure 2 reports the across-problem mean effects and the full rank profile. The correct-mass increase and margin decrease coincide in only one individual problem: among the 3,099 problems both methods solve, mean ΔC is $+0.119$, whereas it is -0.600 among the 490 Power regressions. The aggregate mean therefore masks sharply different paired outcomes.

E HARD-HARP BUDGET PROTOCOL

The budget sweep uses all 394 HARP level-5/6 problems. We retain the first eight Base trajectories from the main Mistral-Small pool and add 15 independent blocks of eight, with the same model revision, prompts, temperature 1, top-$p = 1$, top-$k = 50$, 8,192-token limit, and complete trajectory log-likelihoods. The block order was fixed before the larger-budget readouts and alternates the two generation hosts, which used different tensor-parallel layouts. Interleaving reduces alignment between host and budget but cannot remove host-dependent numerical variation. Every $N \in \{4, 8, 16, 32, 64, 128\}$ pool is the exact first N trajectories of the 128-candidate pool; no budget has separate candidate generation. At each budget we evaluate Uniform, Power ($\alpha = 4$), and SoftSat at $\beta \in \{0, 0.25, 0.5, 1, 1.5, 2, 3, 4\}$ with the official HARP checker. The $\beta = 0$ readout matches Uniform per problem. We report paired changes and effective sample size alongside accuracy. This difficulty-selected slice probes budget behavior and does not estimate an effect on the full benchmark.

Table 6: Single-pool HARP budget readout (correct problems out of 394). U, P, and S are Uniform, Power ($\alpha = 4$), and fixed SoftSat ($\beta = 1$). Fix/break counts are paired S versus U changes. ESS$/N$ is averaged over problems, where $\text{ESS} = 1/\sum_i W_i^2$.

N	U	P	S	Fix/break	P ESS$/N$	S ESS$/N$
4	141	144	159	25/7	0.250	0.970
8	163	151	171	11/3	0.126	0.968
16	184	154	188	6/2	0.063	0.968
32	194	152	189	0/5	0.032	0.968
64	196	150	197	3/2	0.016	0.968
128	191	157	194	4/1	0.008	0.968

The fixed $\beta = 1$ readout changes only 17 of 394 winning answers at $N = 128$; four Uniform errors are repaired and one correct decision is lost. Power's mean ESS$/N$ falls from 0.250 to 0.008, while SoftSat remains near 0.968. These are properties of the weighted readouts on this one pool. At $N = 32$, four of the five SoftSat regressions occur when the winning Uniform answer cluster ties a competing cluster in candidate count; the remaining case has a 10-to-9 count advantage. Neither the $N = 32$ reversal nor the higher β scores at $N = 128$ identify a generally optimal strength.

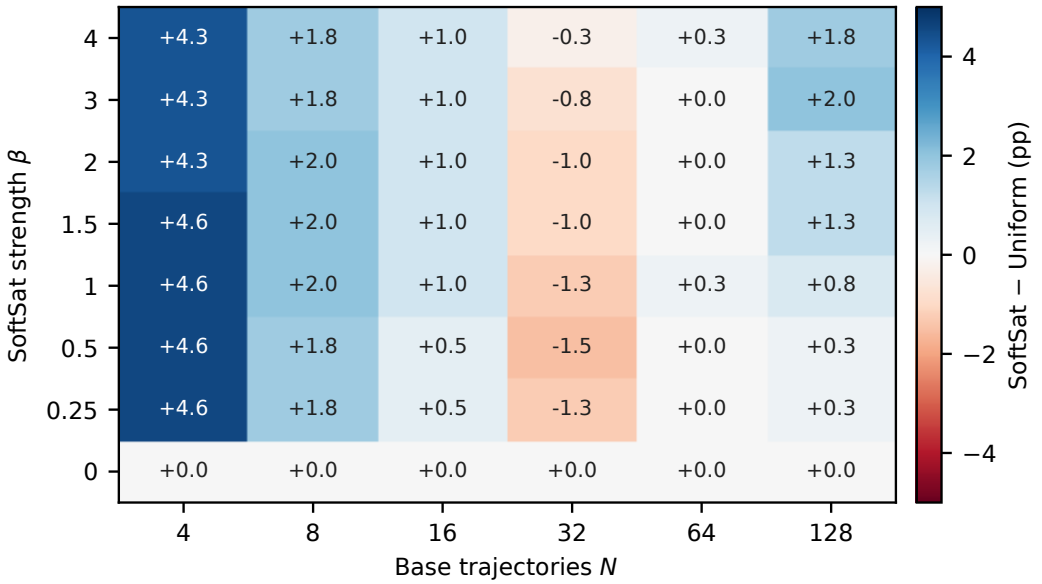


Figure 4: SoftSat minus Uniform accuracy (percentage points) on the fixed 394-problem HARP slice across the pre-specified β grid and nested candidate budgets. The main readout uses $\beta = 1$ at every N; this grid is a sensitivity check, not per-budget tuning.

F Local Power-Target KL Expansion

Write $t = \alpha - 1 \geq 0$ and $\psi_j(t) = \log \mathbb{E}_{p_j}[e^{t\ell_j(\tau)}] = \log Z_{1+t,j}$. Since $q_{1+t,j} = p_j e^{t\ell_j - \psi_j(t)}$,

$$D_{\mathrm{KL}}(q_{1+t,j} \| p_j) = t\psi_j'(t) - \psi_j(t). \tag{20}$$

At $t = 0$, $\psi_j'(0) = \mathbb{E}_{p_j}[\ell_j]$ and $\psi_j''(0) = \mathrm{Var}_{p_j}[\ell_j]$. Because $\ell_j \leq 0$, the assumed finite third absolute moment bounds the right third derivative of ψ_j near zero. Taylor expansion from the right gives $\psi_j(t) = \psi_j'(0)t + \frac{1}{2}\psi_j''(0)t^2 + O(t^3)$ and $\psi_j'(t) = \psi_j'(0) + \psi_j''(0)t + O(t^2)$. Substitution yields Equation 8.

G Finite-Sample Power Readout Comparisons

Base-pool weighting versus repeated Power-SMC. Table 7 compares the Power readout used in our main tables with direct Power-SMC sampling on GPQA-Diamond, MMLU-Pro, and MATH-500. Base-weighted Power aggregates $N = 8$ Base trajectories with weights proportional to $\exp((\alpha - 1)\ell)$, using $\alpha = 4$. The sampling reference makes eight independent Power-SMC calls with $M = 8$ particles each and returns one trajectory per call, then applies uniform consensus to those eight answers. Each comparison uses one seed. GPQA-Diamond and MATH-500 cover all 198 and 500 problems, respectively; MMLU-Pro uses the same 1,000-problem subset as the main evaluation.

For these nine settings, the absolute accuracy gaps range from 0 to 2.02 percentage points, with neither implementation consistently scoring higher. Equation 10 explains their common ideal target; Table 7 measures their finite-budget score differences, not equality of predictions or distributions. The two estimators have different sampling error, and finite-particle SMC adds approximation error to the sampling reference.

For generation cost, Base weighting uses eight ordinary trajectories and a weighted vote, with no additional model calls when their generation log-likelihoods are retained. The reference instead maintains eight particles in each of eight independent SMC runs, with intermediate weighting and resampling. This requires substantially more generation work at comparable trajectory lengths,

Table 7: Single-seed accuracy (%) for Base-weighted Power and repeated Power-SMC+SC_8, with $\alpha = 4$. Both readouts aggregate eight trajectories; each independent SMC call uses eight internal particles. Δ is weighted minus sampled accuracy, in percentage points.

Benchmark	Model	Base-weighted Power	Power-SMC+SC_8	Δ
GPQA-Diamond	Qwen3.8-27B	85.86	84.34	+1.52
	Gemma-4-31B	80.30	80.30	0.00
	Mistral-Small-24B	50.00	52.02	−2.02
MMLU-Pro	Qwen3.8-27B	83.90	84.20	−0.30
	Gemma-4-31B	84.30	82.80	+1.50
	Mistral-Small-24B	70.20	71.60	−1.40
MATH-500	Qwen3.8-27B	95.60	97.60	−2.00
	Gemma-4-31B	96.40	96.60	−0.20
	Mistral-Small-24B	89.00	90.40	−1.40

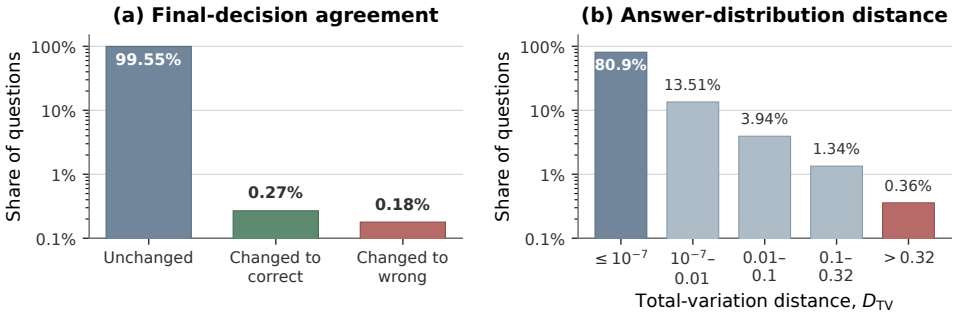


Figure 5: Finite-sample agreement between repeated Power-SMC sampling and a single SMC run's weighted terminal-particle readout on MoreHopQA Case-5. (a) The two procedures return the same normalized answer on 99.55% of questions; the remaining changes are split by whether weighting corrects or introduces an error. (b) Most answer distributions are also close in total variation distance (DTV), although a small tail remains.

although the wall-clock ratio depends on output lengths, batching, and implementation; the accuracy table does not provide a measured speedup.

Separate terminal-particle check. We also retain a comparison on all 1,118 human-verified MoreHopQA Case-5 examples (Schnitzler et al., 2024). This check concerns weighting the terminal particles of one SMC run, rather than a Base pool. With Qwen3.5-9B and $\alpha = 4$, the sampling reference runs an $M = 8$ particle Power-SMC system independently $N = 8$ times and applies uniform SC to the eight returned answers. The weighted realization instead runs Power-SMC once and aggregates the answer mass of its $M = 8$ terminal particles using their normalized weights.

Figure 5 shows that 1,113 of 1,118 final decisions are unchanged. Weighting corrects three errors and introduces two, yielding 91.14% accuracy versus 91.06% for repeated sampling, a difference of 0.089 percentage points with a paired-bootstrap 95% confidence interval of $[-0.268, 0.447]$. The underlying answer distributions agree nearly as closely: 80.86% have DTV at most 10^{-7} and 94.36% have DTV at most 0.01, while only 0.36% exceed $10^{-0.5}$. Mean inference time falls from 19.50 to 2.63 seconds per question, a $7.42\times$ reduction.

H Rank-Space Dose Calibration

Proof of Lemma 1. Order the N candidates by increasing Base log-likelihood. With no ties, equation 15 gives $\widehat{u}_{(r)} = (r - 1/2)/N$. Consequently, the ordered gains $\beta\,\mathrm{SoftSat}_m((r - 1/2)/N)$ depend only on the fixed parameters and N. Applying the same mean-one offset, clipping bounds,

and normalization in equation 16 yields the same ordered weights $W_{(r)}$ for every pool. Equation 17 follows because the empirical Base measure assigns probability $1/N$ to each sample index. □

Atomless population analogue. Let $L = \ell(\tau)$ for $\tau \sim p$ and suppose its CDF F_L is continuous. Then $U = F_L(L)$ is uniform on $[0, 1]$. For the unclipped ideal target $q_{\beta,m}(\tau) \propto p(\tau)e^{g(U(\tau))}$, where $g(u) = \beta\,\mathrm{SoftSat}_m(u)$, the probability integral transform gives

$$\begin{aligned} Z_{\beta,m} &= \int_0^1 e^{g(u)}\,du, \\ D_{\mathrm{KL}}(q_{\beta,m}\|p) &= \frac{\int_0^1 g(u)e^{g(u)}\,du}{Z_{\beta,m}} - \log Z_{\beta,m}. \end{aligned} \tag{21}$$

Both quantities depend only on β, m under this assumption. Complete language-model trajectories are discrete, so this population identity is an idealization; Lemma 1 instead describes the implemented finite-pool readout. Tied empirical log-likelihoods can likewise change its ordered weight vector. At a population atom, empirical midranks approach $F_L(\ell^-) + \frac{1}{2}\Pr(L = \ell)$, whereas equation 12 uses $F_L(\ell)$. Clipping also remains part of the implemented rule as the pool grows; the population identity above concerns the unclipped, atomless construction.

I Proofs for Weighted Target Approximation

Lemma 2 (Asymptotic expectation equivalence)**.** *For every fixed answer a,*

$$\begin{aligned} \lim_{N\to\infty} \mathbb{E}\Big[\widehat{Q}_N^{\mathrm{weight}}(a)\Big] &= \lim_{N\to\infty} \mathbb{E}\Big[\widehat{Q}_N^{\mathrm{sample}}(a)\Big] \\ &= Q_{q_\alpha}(a). \end{aligned} \tag{22}$$

Lemma 3 (Approximation rate)**.** *If $0 < q_\alpha(\tau)/p(\tau) \le C$ almost surely for some finite C, then*

$$\begin{aligned} \mathbb{E}\left[\left(\widehat{Q}_N^{\mathrm{weight}}(a) - \widehat{Q}_N^{\mathrm{sample}}(a)\right)^2\right] &= O(N^{-1}), \\ \widehat{Q}_N^{\mathrm{weight}}(a) - \widehat{Q}_N^{\mathrm{sample}}(a) &= O_p(N^{-1/2}). \end{aligned} \tag{23}$$

Proof of Lemma 2. Fix an answer a and write $f_a(\tau) = \mathbf{1}\{A(\tau) = a\}$. Define

$$\overline{U}_N = \frac{1}{N}\sum_{i=1}^N r_\alpha(X_i)f_a(X_i), \qquad \overline{R}_N = \frac{1}{N}\sum_{i=1}^N r_\alpha(X_i). \tag{24}$$

Since $q_\alpha(\tau) = p(\tau)r_\alpha(\tau)/Z_\alpha$,

$$\mathbb{E}_p[r_\alpha(X)f_a(X)] = Z_\alpha Q_{q_\alpha}(a), \qquad \mathbb{E}_p[r_\alpha(X)] = Z_\alpha. \tag{25}$$

The strong law of large numbers therefore gives $\overline{U}_N \to Z_\alpha Q_{q_\alpha}(a)$ and $\overline{R}_N \to Z_\alpha$ almost surely. Because $Z_\alpha > 0$, their ratio converges almost surely to $Q_{q_\alpha}(a)$. Moreover, $0 \le \overline{U}_N/\overline{R}_N \le 1$, so bounded convergence yields

$$\lim_{N\to\infty} \mathbb{E}\Big[\widehat{Q}_N^{\mathrm{weight}}(a)\Big] = Q_{q_\alpha}(a). \tag{26}$$

For target samples, $f_a(Y_i)$ is Bernoulli with mean $Q_{q_\alpha}(a)$. Hence $\mathbb{E}[\widehat{Q}_N^{\mathrm{sample}}(a)] = Q_{q_\alpha}(a)$ for every N, proving the result. □

Proof of Lemma 3. Let $\rho(\tau) = q_\alpha(\tau)/p(\tau)$ and $\theta = Q_{q_\alpha}(a)$. Multiplying all weights by the same positive constant does not change the self-normalized estimator, so

$$\begin{aligned} \widehat{Q}_N^{\mathrm{weight}}(a) - \theta &= \frac{\overline{H}_N}{\overline{\rho}_N}, \\ \overline{H}_N &= \frac{1}{N}\sum_{i=1}^N \rho(X_i)(f_a(X_i) - \theta), \\ \overline{\rho}_N &= \frac{1}{N}\sum_{i=1}^N \rho(X_i). \end{aligned} \tag{27}$$

Here $\mathbb{E}_p[\rho(X)] = 1$ and $\mathbb{E}_p[\rho(X)(f_a(X) - \theta)] = 0$. Put $\sigma_a^2 = \mathrm{Var}_p[\rho(X)(f_a(X) - \theta)]$, which is finite because $0 < \rho \le C$.

Consider the event $E_N = \{\overline{\rho}_N \ge 1/2\}$. On E_N, equation 27 implies

$$\left(\widehat{Q}_N^{\mathrm{weight}}(a) - \theta\right)^2 \le 4\overline{H}_N^2, \qquad \mathbb{E}[\overline{H}_N^2] = \frac{\sigma_a^2}{N}. \tag{28}$$

On E_N^c, the squared error is at most one. Since $0 \le \rho(X) \le C$ and $\mathbb{E}[\rho(X)] = 1$, Hoeffding's inequality gives

$$\Pr(E_N^c) = \Pr(\overline{\rho}_N - 1 < -1/2) \le \exp\left(-\frac{N}{2C^2}\right). \tag{29}$$

Combining the two events,

$$\mathbb{E}\left[\left(\widehat{Q}_N^{\mathrm{weight}}(a) - \theta\right)^2\right] \le \frac{4\sigma_a^2}{N} + \exp\left(-\frac{N}{2C^2}\right). \tag{30}$$

The target-sampling estimator is unbiased with variance $\theta(1-\theta)/N$. It is independent of the Base-pool estimator, so the cross term between their centered errors vanishes. Writing $D_N(a) = \widehat{Q}_N^{\mathrm{weight}}(a) - \widehat{Q}_N^{\mathrm{sample}}(a)$,

$$\begin{aligned} \mathbb{E}[D_N(a)^2] &\le \frac{4\sigma_a^2 + \theta(1-\theta)}{N} + \exp\left(-\frac{N}{2C^2}\right) \\ &= O(N^{-1}). \end{aligned} \tag{31}$$

The $O_p(N^{-1/2})$ statement follows immediately from this second-moment bound and Markov's inequality. □

Finally, suppose the answer space is finite and $a^\dagger$ is the unique mode of Q_{q_α} with margin $\delta = Q_{q_\alpha}(a^\dagger) - \max_{a \ne a^\dagger} Q_{q_\alpha}(a) > 0$. Both estimators converge in probability to $Q_{q_\alpha}(a)$ for every answer. A union bound over the finite answer space shows that, with probability tending to one, all estimated masses are within $\delta/3$ of their targets. On this event, both estimators select $a^\dagger$, establishing the mode-agreement consequence stated in the main text.