THE IRONY OF POWER SAMPLING: SAMPLING BETTER, FAILING MORE

Anonymous authors
Paper under double-blind review

ABSTRACT

Recent work shows that power sampling enables base models to achieve reasoning performance competitive with that of RL post-trained models without additional training (Karan & Du, 2026). The cost of sampling from the power distribution motivates more efficient approximation algorithms (Ji et al., 2026; Azizi et al., 2026), but how approximation error interacts with the inherent task error of the power distribution remains underexplored. To address this gap, we conduct an in-depth study of how better approximations to the power distribution affect task performance through theoretical analysis and empirical evaluation. We begin with a Markov graph task that illustrates why power sampling helps when model likelihood aligns with task success. We then show theoretically that approximating the power distribution more accurately can systematically degrade task performance. Finally, we corroborate our theoretical findings through experiments with practical LLM power samplers across reasoning and instruction-following tasks. Our experiments show that higher likelihood does not consistently improve task performance and that immediate stopping occurs in practice as predicted by our analysis.

1 INTRODUCTION

Reinforcement learning (RL) drives major advances in the capabilities of large language models. Reinforcement learning from human feedback (RLHF) improves instruction following by optimizing model outputs for human preferences (Ouyang et al., 2022). Subsequent work with verifiable rewards produces substantial gains in mathematics and coding, where correctness can be verified automatically (Guo et al., 2025; Yu et al., 2025). Recent work extends RL to agents that call external tools over multiple rounds of interaction (Zeng et al., 2025). In software engineering, R2E-Gym provides executable environments for training agents and combines automated tests with learned verifiers to choose among candidate code patches (Jain et al., 2025).

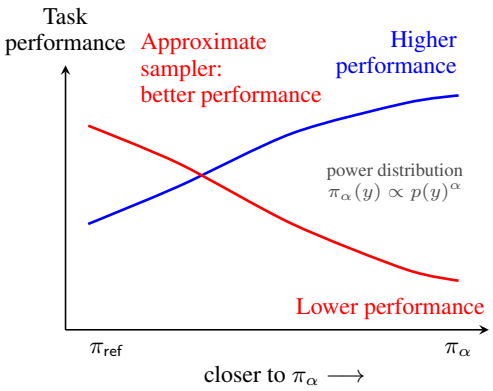


Figure 1: Schematic examples of how closer approximations to a power distribution can improve or reduce task performance. $\pi_{\text{ref}} = q_\alpha^{\text{tok}}$ denotes token sampling at temperature $1/\alpha$.

A central question is whether RL post-training develops new reasoning capabilities or mainly increases the probability of correct responses that the base model can already generate. Recent studies describe *distributional sharpening*, in which training concentrates probability on reasoning paths already likely under the base model (He et al., 2025; Yue et al., 2025).

Motivated by this observation, Karan & Du (2026) propose power sampling to investigate whether base models can match the reasoning performance of RL post-trained models *without* additional training. This method defines the power distribution by applying an exponent $\alpha > 1$ to each re-

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.

sponse probability and normalizing over *complete* responses, while conventional token-level temperature sampling applies the exponent to the next-token probabilities and normalizes them at each generation step. Practical power samplers achieve reasoning performance competitive with that of RL post-trained models, without additional training or an external verifier (Karan & Du, 2026). Recent work proposes more efficient algorithms for approximating the power distribution at a lower computational cost (Ji et al., 2026; Azizi et al., 2026).

Despite these advances, the relationship between approximation quality and task performance remains unclear. Evaluating how closely practical LLM samplers approximate the power distribution is difficult.

We study the relationship between sampling fidelity and task performance through theoretical analysis and LLM experiments. We begin with a graph task in which selecting responses with higher likelihood improves task performance. We use sampling importance resampling (SIR) to analyze how sampling fidelity changes with the number of candidates. Our LLM experiments extend the evaluation to practical samplers across models and tasks.

Our main contributions are as follows.

- Exact calculations on a Markov graph task adapted from Pipis et al. (2026) show that power sampling can outperform conventional token-level temperature sampling when model likelihood aligns with task success (Section 4).
- We identify conditions under which every additional SIR candidate improves the approximation to the power distribution but reduces task performance. SIR allows us to study approximation quality by varying the number of candidates while the proposal and power distributions remain unchanged. For autoregressive models, we show that immediate stopping can be more likely as fidelity improves (Section 5).
- Our LLM experiments compare sampling methods across reasoning and instruction-following tasks. We evaluate likelihood best-of-N, which chooses the most probable of N independent base responses. The method improves task performance in some settings and degrades it in others. Consistent with our theoretical analysis, we observe immediate stopping in practical LLM power samplers and show that reducing it can improve task performance (Section 6).

2 Related Work

Token-level temperature. Token-level temperature affects the quality and diversity of generated responses. Lower temperatures can reduce diversity and increase repetition in open-ended generation (Holtzman et al., 2020). For reasoning models, Pipis et al. (2026) show that a mismatch between the distribution learned by the model and the training distribution can lead to repeated text. A higher temperature can help the model escape these loops without resolving the underlying mismatch. For inference with multiple sampled responses, Du et al. (2025) select temperatures using token-level entropy. Wu et al. (2025) find that models solve different subsets of problems at different temperatures, so combining temperatures can increase the fraction solved.

Power sampling algorithms. Karan & Du (2026) use Metropolis–Hastings Markov chain Monte Carlo (MH-MCMC) to sample approximately from a base model's power distribution without additional training or an external verifier. Subsequent work reduces the cost of approximating the power distribution. Scalable Power Sampling estimates next-token probabilities from sampled completions of the current prefix and applies a jackknife correction for normalization bias (Ji et al., 2026). Power-SMC instead generates multiple partial responses in parallel and resamples them during generation (Azizi et al., 2026). Both methods reduce latency relative to the original MH-MCMC approach while approximating the same power distribution.

Analysis of power sampling. Maximizing sequence likelihood can produce empty or repetitive responses. In machine translation, approximate search can avoid an empty translation even when it is the most probable output (Stahlberg & Byrne, 2019). A model can match the data distribution exactly and still assign its highest probability to an empty or repetitive response. Restricting the search to a specified response length can produce more fluent and relevant text (Yoshida et al., 2024). For power sampling, Tomihari & Sato (2026) express the derivative of expected task reward with respect to the exponent as the covariance between reward and sequence log-probability. The analysis also

relates power sampling to token-level temperature through suffix Rényi entropy. Our stopping result in Proposition 2 is a special case of this relation. Yang et al. (2026) show that power sampling can assign more mass to correct trajectories yet reduce self-consistency performance, and propose modifying the sharpening rule to limit the concentration of probability on a small number of reasoning paths. Self-consistency selects the most frequent final answer across responses, whereas we study the expected task performance of a response selected by SIR. Existing SIR analyses establish convergence and characterize approximation bias (Skare et al., 2003; Deligiannidis et al., 2024). For the same proposal and power distributions, our analysis shows that sampling fidelity can improve with every additional SIR candidate while task performance declines.

3 PRELIMINARIES

3.1 TOKEN-LEVEL AND SEQUENCE-LEVEL TEMPERATURE

Sharpening each token distribution and sharpening the distribution over complete responses lead to different generation rules. The difference comes from where normalization takes place.

Notation. Given a prompt x, the language model generates a response $y = (y_1, \ldots, y_{\ell(y)})$ over a finite vocabulary $\mathcal{V}$. Generation stops when a token in the terminal set $\mathcal{T} \subseteq \mathcal{V}$ is produced or the response reaches the maximum generation length T. The length $\ell(y) \leq T$ counts generated tokens including the terminal token when present. We use $p(y|x)$ to denote the probability of response y given prompt x under this stopping rule. The response space $\Omega \subseteq \bigcup_{t=1}^{T} \mathcal{V}^t$ consists of all such responses with positive probability, including those that reach T tokens without a terminal token. All response distributions below are conditioned on the prompt x. The notation omits this dependence so that $p(y)$ denotes $p(y|x)$ and $p(v|u)$ denotes $p(v|x, u)$ for a response prefix u.

Token-level temperature. Conventional token-level sampling at temperature τ defines an exponent $\alpha = 1/\tau$ and applies the same exponent α to each next-token distribution. With $z_\alpha(u) = \sum_{v \in \mathcal{V}} p(v|u)^\alpha$ and the response prefix $y_{<t} = (y_1, \ldots, y_{t-1})$, we define the token-level temperature distribution $q_\alpha^{\mathsf{tok}}(y)$ as follows:

$$q_\alpha^{\mathsf{tok}}(y) = \prod_{t=1}^{\ell(y)} \frac{p(y_t|y_{<t})^\alpha}{z_\alpha(y_{<t})}.$$

Sequence-level temperature. Power sampling (Karan & Du, 2026) aims to draw responses from the *power distribution* $\pi_\alpha(y)$, which is obtained by raising each complete response probability to the power α and normalizing:

$$\pi_\alpha(y) = \frac{p(y)^\alpha}{Z_\alpha}, \qquad Z_\alpha = \sum_{z \in \Omega} p(z)^\alpha, \qquad \alpha \geq 1.$$

The experiments compare power samplers with token sampling at temperature $\tau = 1/\alpha$.

We note that the base model is recovered at $\alpha = 1$ in both cases, although the exponent affects each probability distribution differently. In the limit $\alpha \to \infty$, token-level sampling chooses the most probable token at each step (greedy decoding), while the power distribution concentrates on the most probable complete responses. More generally, the product of local normalizing constants can vary across responses, so token sampling generally differs from sampling from the power distribution (Azizi et al., 2026).

3.2 SAMPLING ALGORITHMS

We introduce representative algorithms for sampling approximately from the power distribution. SIR allows us to vary the number of candidates without changing the proposal or power distribution, making the effect of approximation easier to analyze. For LLM experiments, we use Metropolis–Hastings Markov chain Monte Carlo (MH-MCMC) and power sampling with sequential Monte Carlo (Power-SMC) because both have publicly available implementations (Karan & Du, 2026; Azizi et al., 2026). These algorithms use a proposal distribution to determine how candidate responses

or tokens are generated. The output distribution gives the probability of each response after selection or updates. For autoregressive models, a proposal distribution can be obtained through token-level temperature sampling.

Sampling importance resampling (SIR). The proposal distribution q on Ω specifies how SIR generates complete candidate responses (Skare et al., 2003; Cremer et al., 2017). The proposal distribution must satisfy $q(y) > 0$ for every $y \in \Omega$ so that every response with positive probability under π_α can be sampled. SIR draws N complete responses $y^{(1)}, \dots, y^{(N)}$ independently from q and assigns larger selection probabilities to candidates with larger ratios of target to proposal probability. The unnormalized importance weight function ρ and the selection probability are

$$\rho(y) = \frac{p(y)^\alpha}{q(y)}, \qquad \mathbb{P}(\text{select candidate } j | y^{(1)}, \dots, y^{(N)}) = \frac{\rho(y^{(j)})}{\sum_{i=1}^N \rho(y^{(i)})}.$$

For example, conventional token-level sampling at temperature $\tau = 1/\alpha$ gives the proposal $q = q_\alpha^{\mathsf{tok}}$ and importance weight $\rho(y) = \prod_{t=1}^{\ell(y)} z_\alpha(y_{<t})$ (Karan & Du, 2026). The SIR output distribution μ_N accounts for randomness in both candidate generation and weighted selection.

Metropolis–Hastings Markov chain Monte Carlo (MH-MCMC). MH-MCMC proposes a new response by keeping a prefix and resampling the remaining tokens (Karan & Du, 2026). The current and proposed responses share the maximum generation length T, with deterministic padding after EOS, allowing proposals to shorten or lengthen a response. We sample a prefix length $i \in \{0, \dots, T-1\}$ uniformly and independently of the current response y. We keep the first i tokens unchanged and regenerate tokens from position $i+1$ using $r_i(\cdot | y_{1:i})$, whose support includes every suffix with positive target probability given the prefix $y_{1:i}$. The proposed response y' is accepted with probability

$$\alpha_{\mathrm{MH},i}(y, y') = \min\left\{1, \frac{p(y')^\alpha r_i(y_{i+1:T} | y_{1:i})}{p(y)^\alpha r_i(y'_{i+1:T} | y_{1:i})}\right\}.$$

The Hastings ratio gives detailed balance for each prefix length i, so sampling this length independently of the current response preserves π_α. While it leads to sampling from the target asymptotically, the output distribution can still depend on initialization and differ from π_α after finitely many updates.

Power sampling with sequential Monte Carlo (Power-SMC). At generation step $t = 1, \dots, T$, Power-SMC maintains N particles $y_{1:t}^{(i)}$ with importance weights $w_t^{(i)}$ for $i = 1, \dots, N$ (Azizi et al., 2026). Each particle represents a response prefix. The token $y_t^{(i)}$ is sampled from the proposal distribution $q_t(\cdot | y_{<t}^{(i)})$. For the power distribution π_α, the weight update before resampling is

$$y_t^{(i)} \sim q_t(\cdot | y_{<t}^{(i)}),$$
$$w_t^{(i)} = w_{t-1}^{(i)} \frac{p(y_t^{(i)} | y_{<t}^{(i)})^\alpha}{q_t(y_t^{(i)} | y_{<t}^{(i)})}, \qquad w_0^{(i)} = 1.$$

For token temperature $\tau = 1/\alpha$, $q_t(v|u) = p(v|u)^\alpha / z_\alpha(u)$, so the incremental weight reduces to $z_\alpha(y_{<t}^{(i)})$. Between resampling steps, each particle's weight is multiplied by the local normalizing constant at each token. To measure how concentrated the weights have become, we define

$$\bar{w}_t^{(i)} = \frac{w_t^{(i)}}{\sum_{j=1}^N w_t^{(j)}}, \qquad \mathrm{ESS}_t = \frac{1}{\sum_{i=1}^N (\bar{w}_t^{(i)})^2}.$$

When $\mathrm{ESS}_t < N/2$, particles are resampled according to $\bar{w}_t$ and their weights are set to 1. After generation, a response is selected according to the normalized particle weights. Deterministic padding after EOS leaves the weights unchanged when α does not vary across generation steps. During generation, resampling can choose the same particle multiple times. In the annealed variant (α-ramping), α_t increases from 1 to 4 during generation with $\tau_t = 1/\alpha_t$. Changes in α_t require additional reweighting beyond the update above (Azizi et al., 2026).

Table 1: Exact graph task results ($\alpha = 4$, maximum length 128 steps). Goal rate is the probability of reaching g within the limit. Root returns is the expected number of $r \to s$ transitions. Mean length is the expected number of steps until g or the limit. The direct path to g has 5 steps.

Distribution	Goal rate	Root returns	Mean length (steps)
Base walk	77.76%	4.907	67.83
Token temperature ($\tau = 0.25$)	81.03%	19.046	64.26
Exact power distribution ($\alpha = 4$)	100.00%	0.002	5.01

4 Motivation: When power sampling helps

A Markov graph task allows us to compute exactly how power sampling changes the probability of successful trajectories.

To this end, we model the exploration variant of the $G(5, 5)$ graph task of Pipis et al. (2026) as a Markov chain. A source node s leads to a root with five arms, and a single arm ends at the goal g. Reaching a leaf other than g sends the walk back to s. A trajectory fails if it reaches the maximum length before reaching the goal. At the root node, returning to the source node s is the most probable individual action even though the most probable complete trajectory follows the direct path to g. Figure 2 shows the graph with the transition probabilities, and Appendix B gives the exact calculation of the power distribution.

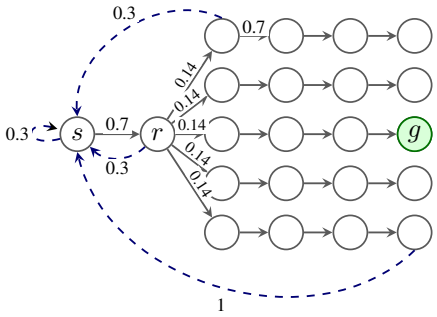


Figure 2: Dashed arrows show representative returns to s. Internal arm nodes move forward with probability 0.7 and return to s with probability 0.3. Reaching g terminates the trajectory.

Dynamic programming gives the exact success probability of the walk under each sampling distribution, and Table 1 shows the result. Lower token temperature makes returning from the root to s more likely and provides only a small improvement in success probability. Power sampling instead places nearly all probability on successful trajectories.

Takeaway 1. Power sampling helps when likelihood aligns with task success.

5 Sampling Fidelity and Task Performance: Through the Lens of SIR

We ask whether a closer approximation improves task performance when the power distribution assigns more probability to incorrect responses than the proposal does. SIR allows us to study this question by varying the number of complete candidates without changing the proposal or power distribution.

5.1 More candidates, better fidelity

To distinguish the effect of the power distribution from approximation error, we define $\mathcal{F} \subseteq \Omega$ as a failure event and denote the sampler's output distribution by μ. Relative to conventional token sampling at temperature $1/\alpha$, the change in failure probability satisfies

$$\mu(\mathcal{F}) - q_\alpha^{\mathsf{tok}}(\mathcal{F}) = \underbrace{\pi_\alpha(\mathcal{F}) - q_\alpha^{\mathsf{tok}}(\mathcal{F})}_{\text{effect of the power distribution}} + \underbrace{\mu(\mathcal{F}) - \pi_\alpha(\mathcal{F})}_{\text{approximation error}}.$$

Table 2: Exact accuracy and TV for SIR on the response tree of Figure 3 with $\alpha = 2$ and candidates drawn at token temperature 0.5. TV is the distance between the SIR output distribution and π_2. At $N = 1$, SIR samples from the token-level distribution.

	SIR with N candidates						
	$N = 1$	$N = 4$	$N = 8$	$N = 16$	$N = 64$	$N = 256$	π_2
Accuracy (%)	94.23	82.00	72.54	63.49	53.80	51.31	50.52
TV to π_2	0.437	0.315	0.220	0.130	0.033	0.008	0

The first term compares exact power sampling with token sampling, while the second compares the sampler's outputs with exact power sampling. When $\mu(\mathcal{F}) < \pi_\alpha(\mathcal{F})$, the approximate sampler has a lower probability of this failure than exact power sampling.

The SIR output distribution converges to its target as the number of candidates grows (Skare et al., 2003; Cremer et al., 2017). The remaining question is whether fidelity improves with each additional candidate. We measure fidelity using total variation (TV) distance, $\mathsf{TV}(\mu, \pi_\alpha) = \frac{1}{2}\sum_{y\in\Omega} |\mu(y) - \pi_\alpha(y)|$, which bounds the magnitude of the approximation error term above.

Theorem 1. *For SIR with a fixed proposal distribution $q > 0$ on the finite response set Ω and fixed positive importance weights ρ, the target distribution is $\pi(y) = q(y)\rho(y)/\mathbb{E}_q[\rho]$. If there are responses $y, z \in \Omega$ with $\rho(y) \neq \rho(z)$, then*

$$0 < \mathsf{TV}(\mu_{N+1}, \pi) < \mathsf{TV}(\mu_N, \pi), \quad \mathsf{TV}(\mu_N, \pi) \to 0 \text{ as } N \to \infty.$$

If every response has the same weight, the SIR output distribution is $\mu_N = q = \pi$ for every N.

For power sampling, choosing $\rho(y) = p(y)^\alpha/q(y)$ gives $\pi = \pi_\alpha$. Our proof extends beyond TV and shows that KL divergence decreases strictly in both directions. The same result holds for χ^2 and squared Hellinger divergence (Appendix A.1).

5.2 When better fidelity means more failures

To relate the fidelity improvement in Theorem 1 to task performance, we define the failure probability conditional on each possible importance weight w

$$g(w) = \mathbb{P}_{Y\sim q}[Y \in \mathcal{F} | \rho(Y) = w].$$

Under the assumptions of Theorem 1, if g is non-decreasing and takes different values at two importance weights, then

$$\mu_N(\mathcal{F}) < \mu_{N+1}(\mathcal{F}) < \pi(\mathcal{F}), \qquad N \geq 1.$$

Each additional candidate therefore increases failure probability while reducing TV. When $\mathcal{F}$ is the set of all incorrect responses, accuracy decreases as well. The sufficient condition concerns the importance weight $\rho(y) = p(y)^\alpha/q(y)$ rather than sequence likelihood alone.

A larger candidate pool shifts probability toward responses with higher importance weights without changing the failure probability at each weight value. When failure is more likely at higher weights, this shift increases the sampler's failure probability.

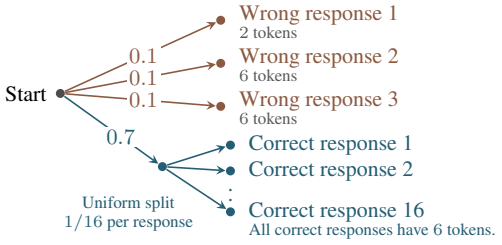


Figure 3: Response tree with $K = 16$ correct responses. Deterministic token sequences are shown as single branches.

A response tree example. Figure 3 illustrates the condition with a finite response tree. The correct branch divides its probability uniformly among K responses, while three deterministic branches lead to incorrect responses. Dividing the branch probability equally among K correct responses multiplies their contribution to Z_α by $K^{1-\alpha}$. Table 2 shows the consequence: SIR becomes more faithful to the same power distribution as accuracy decreases. The construction and proof are given in Appendix A.2.

Table 3: SIR on GPQA Diamond with $\alpha = 4$ and proposal q_4^{tok}.

	Qwen2.5-Math-7B		LLaMA-3.2-3B-Instruct		Phi-3.5-mini-instruct	
Candidates	Accuracy (%)	Immediate stopping (%)	Accuracy (%)	Immediate stopping (%)	Accuracy (%)	Immediate stopping (%)
1	23.06	2.53	19.36	0.00	28.62	0.00
2	24.95	3.70	23.42	0.00	30.27	0.00
4	23.92	4.88	27.11	0.00	30.28	0.00
8	22.16	6.23	27.85	0.00	31.15	0.00

5.3 A CONCRETE INSTANCE: IMMEDIATE STOPPING

To understand which failures the results in Section 5.2 can explain in LLMs, we consider the model's decision to end generation. Sampling EOS as the first token produces an empty response without answering the prompt, a failure we call immediate stopping.

Conditioned on the same prefix, the power distribution assigns at least as much probability to EOS as token sampling at temperature $1/\alpha$ whenever $\alpha > 1$.

Proposition 2 (EOS probability under sequence power)**.** *Let $\alpha > 1$. Suppose the power distribution and token sampling at temperature $\tau = 1/\alpha$ use the same maximum generation length T and terminal set $\mathcal{T}$. For every prefix u with $p(u) > 0$, $|u| < T$, and no EOS token,*

$$\pi_\alpha(\mathcal{T}|u) \geq q_\alpha^{\mathsf{tok}}(\mathcal{T}|u).$$

Here, $p(u)$ is the probability of generating prefix u. The quantity $\mu(\mathcal{T}|u) = \sum_{v\in\mathcal{T}} \mu(v|u)$ denotes the probability of an EOS token after prefix u under the response distribution μ.

The SIR analysis then relates the number of candidates to the probability of immediate stopping.

Corollary 3 (Immediate stopping with more candidates)**.** *Fix $\alpha > 1$. Suppose the event of immediate stopping $\mathcal{E}$ has positive base probability, and some non-terminal first token has positive probability and allows at least two different completed responses with positive conditional probability. For SIR with proposal distribution q_α^{tok} and target distribution π_α,*

$$q_\alpha^{\mathsf{tok}}(\mathcal{E}) = \mu_1(\mathcal{E}) < \mu_2(\mathcal{E}) < \cdots < \pi_\alpha(\mathcal{E}), \qquad \mu_N(\mathcal{E}) \longrightarrow \pi_\alpha(\mathcal{E}).$$

At the same time, $\mathsf{TV}(\mu_N, \pi_\alpha)$ strictly decreases.

Under q_α^{tok}, immediate stopping has maximal importance weight because every additional local normalizing constant is at most 1. The result concerns immediate stopping and does not by itself imply that overall accuracy decreases. All proofs are provided in Appendix A.

Takeaway 2. Improving sampling fidelity can systematically degrade task performance.

6 POWER SAMPLING WITH PRACTICAL LLM SAMPLERS

Section 5 shows that improving sampling fidelity is not equivalent to improving task performance. Our experiments first evaluate SIR on LLM responses and then compare MH-MCMC and Power-SMC across reasoning and instruction-following tasks. We study how likelihood best-of-N affects task performance, then analyze immediate stopping as a source of failure.

SIR with more candidates. We evaluate SIR on GPQA Diamond with token temperature $1/4$ and power exponent $\alpha = 4$. For the first N candidates in each pool, we sum the SIR selection probabilities separately over correct candidates and over candidates that stop immediately. Table 3 reports the mean accuracy and rate of immediate stopping across candidate pools and questions. For Qwen2.5-Math-7B, increasing the number of candidates makes SIR more likely to select an empty response, consistent with Corollary 3. Additional candidates do not produce a consistent accuracy gain for Qwen2.5-Math-7B. For LLaMA and Phi, larger pools improve accuracy, and none of the sampled responses stops immediately.

Table 4: Task performance and response statistics. Accuracy and immediate stopping are percentages, and length is measured in tokens. UltraFeedback uses mean ArmoRM reward. MATH500 results use the same subset of 100 questions. Best-of-N selects the most probable of $N = 32$ independently sampled base responses. Ann. denotes Power-SMC with the exponent gradually increased from 1 to 4 during generation (see Section 3.2). Matching colors identify the comparisons discussed in the text.

Model	Setting	Sampler	GPQA Diamond			MATH500			UltraFeedback	
			Acc.	Immediate stopping	Length	Acc.	Immediate stopping	Length	Reward	Length
Qwen2.5 Math-7B	Base ($\alpha = 1$)	Direct base	17.34	1.68	838.1	47.50	0.50	705.2	0.050	568.0
		MH-MCMC	14.98	3.03	769.0	45.00	0.50	704.5	0.053	598.4
		Power-SMC	18.86	1.35	789.7	45.50	0.00	647.5	0.054	575.0
	$\alpha = 4$ or $\tau = 0.25$	Token temperature	24.41	1.35	860.0	71.50	0.00	646.5	0.079	633.7
		MH-MCMC	21.89	6.90	644.8	73.50	0.00	592.8	0.077	301.3
		Power-SMC	24.58	7.58	788.9	77.50	1.00	708.6	0.085	396.1
		Power-SMC (ann.)	21.89	19.70	620.8	66.50	18.50	560.2	0.087	361.7
	$\tau \to 0$	Greedy	30.30	2.02	873.2	67.00	0.00	647.4	0.078	649.3
	$N = 32$	Best-of-N	11.62	34.34	208.5	69.00	7.50	494.2	0.059	99.3
Qwen2.5 7B	Base ($\alpha = 1$)	Direct base	28.11	1.52	473.3	35.50	0.00	583.6	0.078	562.3
		MH-MCMC	23.91	2.02	480.3	32.00	0.50	594.3	0.085	496.4
		Power-SMC	25.76	2.36	462.6	31.00	1.00	526.2	0.081	561.6
	$\alpha = 4$ or $\tau = 0.25$	Token temperature	28.11	0.84	464.2	66.00	0.00	664.7	0.113	400.3
		MH-MCMC	28.62	1.52	353.6	65.50	0.00	580.8	0.101	214.0
		Power-SMC	29.29	1.68	513.9	71.00	0.00	705.1	0.112	316.5
		Power-SMC (ann.)	23.23	14.98	368.9	55.00	24.00	471.4	0.109	291.2
	$\tau \to 0$	Greedy	29.80	0.51	473.6	63.00	0.00	638.8	0.108	351.6
	$N = 32$	Best-of-N	16.33	22.22	70.1	45.00	6.00	349.9	0.065	63.1
LLaMA-3.2 3B-Instruct	Base ($\alpha = 1$)	Direct base	13.30	0.00	1241.5	23.50	0.00	601.7	0.117	581.3
		MH-MCMC	15.32	0.00	1260.4	25.50	0.00	602.3	0.113	592.0
		Power-SMC	11.11	0.00	1225.9	26.50	0.00	599.6	0.116	572.4
	$\alpha = 4$ or $\tau = 0.25$	Token temperature	20.54	0.00	828.4	38.00	0.00	616.3	0.138	449.7
		MH-MCMC	27.27	0.00	445.1	37.00	0.00	456.7	0.138	390.6
		Power-SMC	23.23	0.00	737.8	36.50	0.00	642.9	0.139	437.1
		Power-SMC (ann.)	23.57	0.00	694.6	42.50	0.00	605.4	0.137	456.2
	$\tau \to 0$	Greedy	21.72	0.00	868.3	31.00	0.00	586.5	0.136	473.1
	$N = 32$	Best-of-N	19.19	0.00	343.2	31.50	0.00	263.8	0.131	314.9
Phi-3.5 mini-instruct	Base ($\alpha = 1$)	Direct base	25.59	0.00	563.1	36.50	0.00	569.4	0.138	562.8
		MH-MCMC	26.94	0.00	560.8	38.00	0.00	536.1	0.139	556.1
		Power-SMC	29.29	0.00	550.4	39.00	0.00	549.2	0.138	565.9
	$\alpha = 4$ or $\tau = 0.25$	Token temperature	28.79	0.00	517.8	44.00	0.00	561.1	0.138	564.4
		MH-MCMC	30.64	0.00	412.0	46.00	0.00	421.8	0.138	466.3
		Power-SMC	27.95	0.00	521.0	46.50	0.00	538.7	0.141	537.8
		Power-SMC (ann.)	26.09	0.00	520.9	46.00	0.00	570.3	0.140	548.0
	$\tau \to 0$	Greedy	30.81	0.00	521.2	42.00	0.00	547.7	0.139	566.9
	$N = 32$	Best-of-N	29.63	0.00	359.5	44.50	0.00	364.5	0.140	402.6

Table 4 extends the evaluation to MH-MCMC and Power-SMC across four models (Yang et al., 2024; Qwen et al., 2024; Grattafiori et al., 2024; Abdin et al., 2024) on GPQA Diamond, a subset of MATH500 and UltraFeedback (Rein et al., 2024; Lightman et al., 2024; Cui et al., 2024). Details of the experimental setup are provided in Appendix C.

6.1 Likelihood and Task Performance Can Diverge

We compare base sampling with likelihood best-of-N, which selects the most probable of N independent base responses. On GPQA Diamond, likelihood selection has

opposite effects across models: accuracy decreases for both Qwen models and increases for LLaMA and Phi relative to base sampling (Table 4). For both Qwen models in our experiments, likelihood selection picks an empty response whenever an empty response is available. Likelihood selection also chooses non-empty incorrect responses from pools whose first candidate is correct. Neither base sampling nor likelihood best-of-N generates an empty response for LLaMA or Phi in our experiments. Best-of-N improves MATH500 accuracy for all four models.

Power sampling can also improve task performance. For the Phi model, MH-MCMC and Power-SMC achieve higher accuracy on MATH500 at $\alpha = 4$ than at $\alpha = 1$, with no immediate stopping observed. On UltraFeedback, mean reward is higher at $\alpha = 4$ than at $\alpha = 1$ for both methods across all four models, except for MH-MCMC on Phi. Responses are also shorter in all four models, so shorter responses alone are not evidence of task failure.

6.2 Immediate Stopping in Practice

Practical LLM power samplers also exhibit the immediate stopping behavior studied in Section 5.3. We focus on Power-SMC because its annealed variant generates empty responses more often than the other MH-MCMC and Power-SMC settings for both Qwen models (Table 4). We compare Power-SMC at $\alpha = 4$ with a variant that gradually increases the exponent from 1 to 4 in its proposals and intermediate power distributions. Both use the same number of particles. With annealing, both Qwen models show a higher immediate stopping rate and lower accuracy on GPQA Diamond and MATH500.

The annealed sampler can propose more empty responses at the first step. Resampling and final selection determine whether the output is empty. Because annealing changes intermediate weights and resampling, stopping rates do not establish which sampler is closer to π_4. Since immediate stopping is determined by the first token, we compare two stopping rules for Power-SMC to evaluate whether reducing it improves accuracy. Both rules use the same model parameters and number of particles and allow EOS at later positions.

Table 5: Sampling rules for Power-SMC with Qwen2.5-Math-7B on GPQA Diamond.

Sampling rule	Accuracy (%)	Immediate stopping (%)
Original	21.04	21.04
Preserve base stopping probability	27.44	1.01
Prevent immediate stopping	28.11	0.00

- **Preserve base stopping probability.** The sampler stops immediately with the same probability as the base model. Otherwise, Power-SMC generates a response with EOS excluded at the first token.
- **Prevent immediate stopping.** Power-SMC always excludes EOS at the first token.

Both rules improve GPQA Diamond accuracy for Qwen2.5-Math-7B (Table 5). For some prompts, the modified samplers generate correct answers where the original sampler generates an empty response.

Takeaway 3. The irony of power sampling: choosing responses with higher likelihood can degrade task performance.

7 Conclusion

In this work, we studied the relationship between sampling fidelity and task performance. We first used a Markov graph task to demonstrate how power sampling improved performance when likelihood aligned with task success. Our SIR analysis then established conditions under which additional candidates improved sampling fidelity while reducing task performance. Across LLMs and tasks, we found that higher likelihood did not consistently lead to better performance. We also analyzed immediate stopping in Power-SMC and found that modifying its stopping rules improved accuracy. As future work, we plan to study other failure modes across more models and datasets and analyze how MH updates and Power-SMC resampling affect sampling fidelity and task performance.

AI USE STATEMENT

We used generative AI tools for language editing and code sanity checks. For proofs and results, we used them only to improve presentation and obtain feedback on our interpretation. The authors reviewed all AI-assisted edits and take responsibility for the final content.

REFERENCES

Marah I Abdin, Sam Ade Jacobs, Ammar Ahmad Awan, Jyoti Aneja, Ahmed Awadallah, Hany Hassan Awadalla, Nguyen Bach, Amit Bahree, Arash Bakhtiari, Harkirat Behl, Alon Benhaim, Misha Bilenko, Johan Bjorck, Sébastien Bubeck, Martin Cai, Caio César Teodoro Mendes, Weizhu Chen, Vishrav Chaudhary, Parul Chopra, Allie Del Giorno, Gustavo de Rosa, Matthew Dixon, Ronen Eldan, Dan Iter, Abhishek Goswami, Suriya Gunasekar, Emman Haider, Junheng Hao, Russell J. Hewett, Jamie Huynh, Mojan Javaheripi, Xin Jin, Piero Kauffmann, Nikos Karampatziakis, Dongwoo Kim, Mahmoud Khademi, Lev Kurilenko, James R. Lee, Yin Tat Lee, Yuanzhi Li, Chen Liang, Weishung Liu, Xihui (Eric) Lin, Zeqi Lin, Piyush Madan, Arindam Mitra, Hardik Modi, Anh Nguyen, Brandon Norick, Barun Patra, Daniel Perez-Becker, Thomas Portet, Reid Pryzant, Heyang Qin, Marko Radmilac, Corby Rosset, Sambhuda Roy, Olli Saarikivi, Amin Saied, Adil Salim, Michael Santacroce, Shital Shah, Ning Shang, Hiteshi Sharma, Xia Song, Olatunji Ruwase, Xin Wang, Rachel Ward, Guanhua Wang, Philipp Witte, Michael Wyatt, Can Xu, Jiahang Xu, Weijian Xu, Sonali Yadav, Fan Yang, Ziyi Yang, Donghan Yu, Chengruidong Zhang, Cyril Zhang, Jianwen Zhang, Li Lyna Zhang, Yi Zhang, Yunan Zhang, and Xiren Zhou. Phi-3 technical report: A highly capable language model locally on your phone. Technical Report MSR-TR-2024-12, Microsoft, August 2024. URL https://www.microsoft.com/en-us/research/publication/phi-3-technical-report-a-highly-capable-language-model-locally-on-your-phone/.

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. *arXiv preprint arXiv:2602.10273*, 2026.

Chris Cremer, Quaid Morris, and David Duvenaud. Reinterpreting importance-weighted autoencoders. *arXiv preprint arXiv:1704.02916*, 2017.

Ganqu Cui, Lifan Yuan, Ning Ding, Guanming Yao, Bingxiang He, Qiang Yue, Yuan Ni, Guotong Xie, Ruobing Xie, Yankai Lin, Zhiyuan Liu, and Maosong Sun. ULTRAFEEDBACK: Boosting language models with scaled AI feedback. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=BOorDpKHiJ.

George Deligiannidis, Pierre E Jacob, El Mahdi Khribch, and Guanyang Wang. On importance sampling and independent metropolis-hastings with an unbounded weight function. *arXiv preprint arXiv:2411.09514*, 2024.

Weihua Du, Yiming Yang, and Sean Welleck. Optimizing temperature for language models with multi-sample inference. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=rmWpE3FrHW.

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

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1 incentivizes reasoning in llms through reinforcement learning. *Nature*, 645(8081):633–638, 2025.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the unlikely: Lifting grpo beyond distribution sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25559–25571, 2025.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=rygGQyrFvH.

Naman Jain, Jaskirat Singh, Manish Shetty, Tianjun Zhang, Liang Zheng, Koushik Sen, and Ion Stoica. R2e-gym: Procedural environment generation and hybrid verifiers for scaling open-weights SWE agents. In *Second Conference on Language Modeling*, 2025. URL https://openreview.net/forum?id=7evvwwdo3z.

Xiaotong Ji, Rasul Tutunov, Matthieu Zimmer, and Haitham Bou Ammar. Scalable power sampling: Unlocking efficient, training-free reasoning for LLMs via distribution sharpening. In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=SVyjXhZlDe.

Aayush Karan and Yilun Du. Reasoning with sampling: Your base model is smarter than you think. In *The Fourteenth International Conference on Learning Representations*, 2026. URL https://openreview.net/forum?id=Vsgq2ldr4K.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. Let's verify step by step. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=v8L0pN6EOi.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Gray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul Christiano, Jan Leike, and Ryan Lowe. Training language models to follow instructions with human feedback. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TG8KACxEON.

Charilaos Pipis, Shivam Garg, Vasilis Kontonis, Vaishnavi Shrivastava, Akshay Krishnamurthy, and Dimitris Papailiopoulos. Wait, wait, wait... why do reasoning models loop? In *Forty-third International Conference on Machine Learning*, 2026. URL https://openreview.net/forum?id=oZWE7mSqlk.

Qwen, :, An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiaxi Yang, Jingren Zhou, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tianyi Tang, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2024.

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. URL https://openreview.net/forum?id=Ti67584b98.

Øivind Skare, Erik Bølviken, and Lars Holden. Improved sampling-importance resampling and reduced bias importance sampling. *Scandinavian Journal of Statistics*, 30(4):719–737, 2003.

Felix Stahlberg and Bill Byrne. On nmt search errors and model errors: Cat got your tongue? In *Proceedings of the 2019 conference on empirical methods in natural language processing and the 9th international joint conference on natural language processing (EMNLP-IJCNLP)*, pp. 3356–3362, 2019.

Akiyoshi Tomihari and Issei Sato. Power distribution bridges sampling, self-reward rl, and self-distillation. *arXiv preprint arXiv:2605.04542*, 2026.

Yuheng Wu, Azalia Mirhoseini, and Thierry Tambe. On the role of temperature sampling in test-time scaling. *arXiv preprint arXiv:2510.02611*, 2025.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, et al. Qwen2. 5-math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Haohui Yang, Jiaxing Sun, and Xiujun Ma. More correct mass, worse answers: Why power sampling can fail and how to fix it. *arXiv preprint arXiv:2608.14420*, 2026.

Davis Yoshida, Kartik Goyal, and Kevin Gimpel. Map's not dead yet: Uncovering true language model modes by conditioning away degeneracy. In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 16164–16215, 2024.

Qiying Yu, Zheng Zhang, Ruofei Zhu, Yufeng Yuan, Xiaochen Zuo, YuYue, Weinan Dai, Tiantian Fan, Gaohong Liu, Juncai Liu, LingJun Liu, Xin Liu, Haibin Lin, Zhiqi Lin, Bole Ma, Guangming Sheng, Yuxuan Tong, Chi Zhang, Mofan Zhang, Ru Zhang, Wang Zhang, Hang Zhu, Jinhua Zhu, Jiaze Chen, Jiangjie Chen, Chengyi Wang, Hongli Yu, Yuxuan Song, Xiangpeng Wei, Hao Zhou, Jingjing Liu, Wei-Ying Ma, Ya-Qin Zhang, Lin Yan, Yonghui Wu, and Mingxuan Wang. DAPO: An open-source LLM reinforcement learning system at scale. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=2a36EMSSTp.

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 *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025. URL https://openreview.net/forum?id=4OsgYD7em5.

Aohan Zeng, Xin Lv, Qinkai Zheng, Zhenyu Hou, Bin Chen, Chengxing Xie, Cunxiang Wang, Da Yin, Hao Zeng, Jiajie Zhang, et al. Glm-4.5: Agentic, reasoning, and coding (arc) foundation models. *arXiv preprint arXiv:2508.06471*, 2025.

A PROOFS AND SUPPORTING CALCULATIONS

A.1 FIDELITY OF IMPORTANCE RESAMPLING

We consider $q, \rho > 0$ on finite Ω, with $Z = \mathbb{E}_q[\rho]$, $\pi = q\rho/Z$, and SIR output distribution μ_N. For $f : \Omega \to \mathbb{R}$, we write $m_f(w) = \mathbb{E}_q[f(Y)|\rho(Y) = w]$. To prove Theorem 1, we first show that if m_f is non-decreasing and differs at two weight values, then

$$\mathbb{E}_{\mu_N}[f] < \mathbb{E}_{\mu_{N+1}}[f] < \mathbb{E}_\pi[f].$$

Setting $f = \mathbf{1}_{\mathcal{F}}$ gives the result on failure probability in Section 5.2.

The output distribution. Exchangeability of the independent candidates $y^{(i)} \sim q$ gives

$$\frac{\mu_N(y)}{\pi(y)} = NZ\mathbb{E}\left[\frac{1}{\rho(y) + \sum_{i=2}^N \rho(y^{(i)})}\right].$$

The ratio strictly decreases with $\rho(y)$, and q, μ_N, and π have the same conditional distributions given the weight. The sum of normalized weights over all candidates equal to y converges almost surely to $\pi(y)$ by the strong law. The normalized sum lies in $[0, 1]$ and has expectation $\mu_N(y)$, so bounded convergence gives $\mu_N \to \pi$ in TV on finite Ω.

Adding a candidate. For $N \geq 1$, we draw $N + 1$ independent candidates and write $\rho_i = \rho(y^{(i)})$ and $S = \sum_{i=1}^{N+1} \rho_i$. Removing the candidate at any fixed index leaves N independent draws from q. Averaging over the removed index gives the first equality below. Pairing the (i, j) and (j, i) terms and conditioning on the weights gives the second:

$$\begin{aligned}\mathbb{E}_{\mu_{N+1}}[f] - \mathbb{E}_{\mu_N}[f] &= \mathbb{E}\left[\frac{\sum_i \rho_i f(y^{(i)})}{S} - \frac{1}{N+1}\sum_i \frac{\sum_{j\neq i}\rho_j f(y^{(j)})}{S - \rho_i}\right] \\ &= \mathbb{E}\left[\frac{1}{(N+1)S}\sum_{i<j}\frac{\rho_i\rho_j(\rho_i - \rho_j)\{m_f(\rho_i) - m_f(\rho_j)\}}{(S-\rho_i)(S-\rho_j)}\right]. \qquad (1)\end{aligned}$$

Every summand is non-negative when m_f is non-decreasing. If m_f differs at two weight values, both occur together in a candidate pool with nonzero probability under q, and the corresponding summand is strictly positive. Hence $\mathbb{E}_{\mu_N}[f]$ increases strictly to $\mathbb{E}_\pi[f]$.

Strict decrease in TV. We order the distinct weights as $w_1 < \cdots < w_L$, with $L \geq 2$. Applying (1) to $f(y) = \mathbf{1}\{\rho(y) > w_j\}$ shows that $\mu_N(\rho > w_j)$ increases strictly to $\pi(\rho > w_j)$ for every $j < L$. Since μ_N/π decreases with the weight, the set where $\pi(y) > \mu_N(y)$ consists of responses above a weight threshold. Hence

$$\mathsf{TV}(\mu_N, \pi) = \max_{1\leq j<L}\{\pi(\rho > w_j) - \mu_N(\rho > w_j)\}.$$

Each term is positive and strictly decreases, so the maximum over these finitely many terms also decreases strictly. Equal weights instead give $\mu_N = q = \pi$ for every N.

Other measures of fidelity. For a differentiable convex function $\varphi : (0, \infty) \to \mathbb{R}$ with $\varphi(1) = 0$, we define

$$D_\varphi(\mu\|\pi) = \sum_y \pi(y)\varphi\left(\frac{\mu(y)}{\pi(y)}\right).$$

We write $r_N(w) = \mu_N(y)/\pi(y)$ when $\rho(y) = w$. Applying convexity of φ at $r_{N+1}(w_j)$ and then summation by parts gives

$$\begin{aligned} &D_\varphi(\mu_N\|\pi) - D_\varphi(\mu_{N+1}\|\pi) \\ &\quad\geq \sum_{j=1}^{L} \varphi'(r_{N+1}(w_j))\{\mu_N(\rho = w_j) - \mu_{N+1}(\rho = w_j)\} \\ &\quad= \sum_{j=1}^{L-1} \{\varphi'(r_{N+1}(w_j)) - \varphi'(r_{N+1}(w_{j+1}))\}\{\mu_N(\rho \leq w_j) - \mu_{N+1}(\rho \leq w_j)\} \geq 0. \end{aligned}$$

The first factor is non-negative because $r_{N+1}(w)$ decreases with w and φ' is non-decreasing; the second is positive by the tail inequalities. Strict convexity therefore gives strict decrease for unequal weights. Since $\mu_N \to \pi$ and φ is continuous, $D_\varphi(\mu_N\|\pi) \to 0$. Both directions of KL divergence, χ^2, and squared Hellinger divergence are special cases.

A.2 A RESPONSE TREE EXAMPLE

The root of the response tree has branch probabilities $(0.1, 0.1, 0.1, 0.7)$. The first three branches lead to a set $\mathcal{F}$ of three deterministic incorrect responses. The first ends after two tokens without answering, and the other two are wrong answers. On the final branch, the next token is sampled uniformly from $K \geq 2$ options before completing a correct response. Every response except the early stop has length six, including EOS.

For $\alpha > 1$, both q_α^{tok} and π_α are uniform over $\mathcal{F}$ and over its complement. Their failure probabilities are

$$r = q_\alpha^{\mathsf{tok}}(\mathcal{F}) = \frac{3}{3 + 7^\alpha}, \qquad s_* = \pi_\alpha(\mathcal{F}) = \frac{3}{3 + 7^\alpha K^{1-\alpha}} > r.$$

Theorem 4. *For SIR on this tree with $q = q_\alpha^{\mathsf{tok}}$ and fixed $\alpha > 1$, write $s_N = \mu_N(\mathcal{F})$. Then $r = s_1 < s_2 < \cdots < s_*$, $s_N \to s_*$, and $\mathsf{TV}(\mu_N, \pi_\alpha) = s_* - s_N$. For each SIR output distribution μ_N and for the power distribution π_α, let s denote the probability of $\mathcal{F}$. The three incorrect responses are equally likely under these distributions, giving*

$$\mathsf{Acc} = 1 - s, \qquad \mathbb{E}[\ell] = 6 - \frac{4s}{3}.$$

Increasing N strictly reduces TV, while mean length and accuracy decrease and the probability of early termination increases.

Proof of Theorem 4. The importance weight of an incorrect response is $\eta = K^{\alpha-1} > 1$ times that of a correct response. We denote the number of incorrect candidates by $J \sim \mathrm{Binomial}(N, r)$. Weighted selection gives

$$s_N = \mathbb{E}\left[\frac{\eta J}{\eta J + N - J}\right].$$

The monotonicity result in Appendix A.1 gives $r = s_1 < s_2 < \cdots < s_*$ and $s_N \to s_*$. Both distributions are uniform over each group, giving $\mathsf{TV}(\mu_N, \pi_\alpha) = s_* - s_N$. Each incorrect response has probability $s/3$, so $\mathsf{Acc} = 1 - s$ and $\mathbb{E}[\ell] = 6 - 4s/3$. Evaluating the binomial expectation gives Table 2.

Using $\mathbb{E}[J/N] = r$, we obtain the exact difference from the limit:

$$s_* - s_N = \frac{\eta(\eta - 1)}{[1 + (\eta - 1)r]^2}\mathbb{E}\left[\frac{(J/N - r)^2}{1 + (\eta - 1)J/N}\right].$$

Since $1 \leq 1 + (\eta - 1)J/N \leq \eta$ and $\mathbb{E}[(J/N - r)^2] = r(1 - r)/N$,

$$\frac{(\eta - 1)r(1 - r)}{[1 + (\eta - 1)r]^2 N} \leq s_* - s_N \leq \frac{\eta(\eta - 1)r(1 - r)}{[1 + (\eta - 1)r]^2 N}.$$

Thus $s_* - s_N = \Theta(N^{-1})$. At $\alpha = 1$ or $K = 1$, equal weights give $\mu_N = \pi_\alpha$.

Proof of Corollary 3. Under q_α^{tok}, $\rho(y) = \prod_{t=1}^{\ell(y)} z_\alpha(y_{<t}) \le z_\alpha(\varnothing)$ because every factor is at most 1. Responses that stop immediately have maximal weight. By assumption, a non-terminal branch contains a prefix with multiple possible next tokens, so some response that continues has strictly smaller weight. For $f = \mathbf{1}_{\mathcal{E}}$, m_f is zero below the maximal weight and positive at it. Appendix A.1 therefore gives strictly increasing stopping probability and strictly decreasing TV.

Proof of Proposition 2. All token sums below are restricted to $p(v|u) > 0$. We define $B(uv)$ as the sum of powered conditional probabilities of completing the response after appending token v to prefix u. We set $B(uv) = 1$ after EOS or at length T. Since the conditional probabilities sum to 1, $0 < B(uv) \le 1$. Therefore

$$\pi_\alpha(\mathcal{T}|u) = \frac{\sum_{v\in\mathcal{T}} p(v|u)^\alpha}{\sum_{v\in\mathcal{T}} p(v|u)^\alpha + \sum_{v\notin\mathcal{T}} p(v|u)^\alpha B(uv)} \ge \frac{\sum_{v\in\mathcal{T}} p(v|u)^\alpha}{\sum_v p(v|u)^\alpha} = q_\alpha^{\mathsf{tok}}(\mathcal{T}|u).$$

The inequality is strict when EOS has positive probability and a non-terminal branch with positive probability has $B(uv) < 1$.

A larger conditional stopping probability does not necessarily imply a shorter mean response length. The root chooses EOS with probability 0.1 and either of two other branches with probability 0.45 each. One branch splits uniformly between two responses of length three, while the other completes a deterministic response of length six. At $\alpha = 2$,

$$q_\alpha^{\mathsf{tok}}(\text{immediate stopping}) = \frac{2}{83} < \frac{8}{251} = \pi_\alpha(\text{immediate stopping}),$$
$$\mathbb{E}_{q_\alpha^{\mathsf{tok}}}[\ell] = \frac{733}{166} < \frac{1223}{251} = \mathbb{E}_{\pi_\alpha}[\ell].$$

Both immediate stopping and mean length increase in this example, where length includes EOS.

A.3 Preserving the stopping probability

Proposition 5 (Preserving the base stopping probability)**.** *Let $\mathcal{E}$ be the event of immediate stopping and $C = \mathcal{E}^c$. Assume $\gamma = p(\mathcal{E}) \in (0,1)$. Write $\nu_{\mathcal{E}} = \pi_\alpha(\cdot \mid \mathcal{E})$ and $\nu_C = \pi_\alpha(\cdot \mid C)$. Then*

$$\pi_{\mathsf{keep}} = \gamma\nu_{\mathcal{E}} + (1-\gamma)\nu_C, \qquad \mathsf{TV}(\pi_{\mathsf{keep}}, \pi_\alpha) = |\pi_\alpha(\mathcal{E}) - \gamma|.$$

Suppose a sampler draws from $\widehat{\nu}_{\mathcal{E}}$ with probability γ and from $\widehat{\nu}_C$ otherwise, with each distribution supported on its corresponding event. The output distribution $\widehat{\pi}_{\mathsf{keep}}$ then satisfies

$$\widehat{\pi}_{\mathsf{keep}}(\mathcal{E}) = \gamma, \qquad \mathsf{TV}(\widehat{\pi}_{\mathsf{keep}}, \pi_{\mathsf{keep}}) = \gamma\,\mathsf{TV}(\widehat{\nu}_{\mathcal{E}}, \nu_{\mathcal{E}}) + (1-\gamma)\,\mathsf{TV}(\widehat{\nu}_C, \nu_C).$$

If immediate stopping is always incorrect, then

$$\mathsf{Acc}(\pi_{\mathsf{keep}}) - \mathsf{Acc}(\pi_\alpha) = (\pi_\alpha(\mathcal{E}) - \gamma)\,\mathsf{Acc}(\nu_C).$$

Proof. Both mixtures assign probability γ to immediate stopping. Splitting the TV sum over $\mathcal{E}$ and C gives both identities. Since stopping is incorrect, the accuracies under π_α and π_{keep} are $(1 - \pi_\alpha(\mathcal{E}))\,\mathsf{Acc}(\nu_C)$ and $(1-\gamma)\,\mathsf{Acc}(\nu_C)$, respectively. □

When there are multiple EOS tokens, we sample from $\nu_{\mathcal{E}}$ using probabilities proportional to $p(v)^\alpha$. The remaining TV error relative to π_{keep} is $(1-\gamma)\,\mathsf{TV}(\widehat{\nu}_C, \nu_C)$. At $\alpha = 1$, $\pi_{\mathsf{keep}} = p$.

The accuracy gain is positive when $\pi_\alpha(\mathcal{E}) > \gamma$ and $\mathsf{Acc}(\nu_C) > 0$.

B Synthetic experiments on the star graph

We compute the power distribution exactly for the Markov graph in Figure 2, following the exploration task of Pipis et al. (2026). Trajectory length counts transitions from s, and reaching the length limit before g is failure. Deterministic padding after g preserves the probability of each completed trajectory.

We define $Z_h(v)$ as the sum of powered probabilities of paths starting at v with h transitions remaining. Dynamic programming gives

$$Z_h(v) = \sum_{v'} p(v'|v)^\alpha Z_{h-1}(v'), \qquad \pi_\alpha(v'|v,h) = \frac{p(v'|v)^\alpha Z_{h-1}(v')}{Z_h(v)},$$

with $Z_0(v) = Z_h(g) = 1$. The goal probability and expected trajectory statistics in Table 1 follow from these transition probabilities. The rounded success rate does not imply zero failure probability.

C Evaluation Protocol

For the comparisons in Table 4, all methods use the same prompts from GPQA Diamond, MATH500 and UltraFeedback. Performance is measured by accuracy on the reasoning tasks and mean ArmoRM reward on UltraFeedback. We use ArmoRM-Llama3-8B-v0.1 as a reward model.[1] We treat a response as incorrect if its final answer cannot be extracted. Response length is the number of generated tokens, including EOS when present. Immediate stopping means that the first generated token is terminal.

Base sampling uses temperature 1, while likelihood best-of-N chooses the most probable response from a pool of base samples. We compare the correctness of the chosen response and the first candidate in the same pool. The direct base sampling results in Table 4 use samples generated separately from these pools. For SIR, we average candidate outcomes using their selection probabilities, then average across candidate pools and prompts.

The *Preserve base stopping probability* rule samples approximately from π_{keep} in Proposition 5 while preserving the base stopping probability $\gamma = p(\mathcal{E})$. The *Prevent immediate stopping* rule excludes terminal tokens at the first position. Both rules allow EOS at later positions.

[1] https://huggingface.co/RLHFlow/ArmoRM-Llama3-8B-v0.1