Sample Where You Struggle: Sharpening Base Model Reasoning via Entropy-Guided Power Sampling

Anonymous ACL submission

Abstract

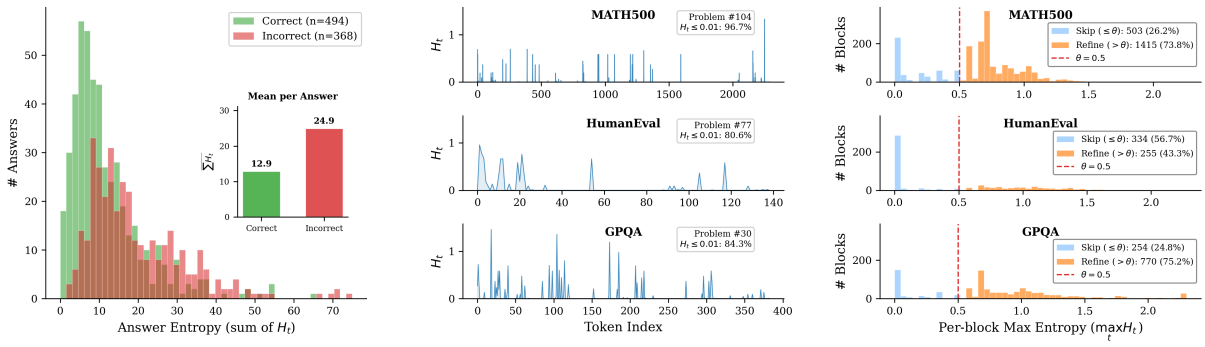
Sampling from the sequence-level power distribution p^α elicits RL-level reasoning from base language models without any parameter updates, but the standard Metropolis–Hastings (MH), a Markov Chain Monte Carlo (MCMC) sampler, is both expensive and slow-mixing. We trace both to a structural mismatch: p^α mainly departs from p at a sparse, spatially clustered set of high-entropy decision points, yet MH proposes resampling positions uniformly along the prefix—wasting compute on near-degenerate conditionals while under-mixing precisely where modes diverge. We propose Entropy-Guided Power Sampling (EGPS), a training-free and verifier-free sampler that re-derives its proposal from token-level entropy already in the forward pass. EGPS skips deterministic blocks, localizes each MCMC move to a high-entropy neighborhood, and applies Multiple-Try Metropolis at decision points—making sampling cost scale with *entropy mass rather than sequence length*. On Qwen2.5-Math-7B, EGPS reaches best or tied-best accuracy on all three benchmarks (MATH500 75.8%, HumanEval 62.2%, GPQA 42.4%) at up to a $12.6\times$ wall-clock speedup over the MH baseline.

1 Introduction

Reinforcement learning (RL) post-training has emerged as a common approach for enhancing the reasoning ability of large language models (LLMs), but it is costly: it typically requires carefully curated data, an elaborate training pipeline, and automated verifiers. A growing line of work (He et al., 2025; Yue et al., 2025; Song et al., 2025) argues from different angles that RL post-training primarily performs *distribution sharpening*: rather than introducing reasoning abilities absent from the base model, it redistributes probability mass toward reasoning trajectories already favored by the base model. This perspective motivates a natural alternative: sharpening the base model's output distribution directly at inference time, thereby bypassing the costly training process. Along this line, Karan and Du (2025) propose power sampling, which uses Metropolis–Hastings (MH) to sample from the sequence-level power distribution p^α ($\alpha > 1$). Without any parameter updates, it reportedly achieves performance comparable to GRPO (Shao et al., 2024) on benchmarks such as MATH500 (Lightman et al., 2024).

The practicality of power sampling, however, is limited by the efficiency of its MH sampler. MH generates the sequence block by block: after each block is produced, it selects a resampling start point uniformly from the entire generated prefix and regenerates from that point to the end of the current block (Fig. 2(a)). This uniform proposal implicitly treats every token as equally worth resampling.

Yet reasoning traces are far from uniform: prior work has shown that their quality tends to be driven by a small set of critical tokens, which often exhibit high entropy (Lin et al., 2024; Wang et al., 2026; Yang et al., 2025); in particular, incorrect answers tend to have higher overall entropy than correct ones (Yang et al., 2025), a pattern we observe empirically (Fig. 1a). We further characterize the entropy distribution at both the token and block levels, revealing two structural properties of high-entropy positions. First, sparsity: high-entropy positions account for only a small fraction of the sequence. Second, spatial clustering: high-entropy positions are separated by long stretches of near-deterministic tokens, forming spatially localized clusters (Fig. 1b). A direct consequence of these two properties is that a substantial fraction of blocks remain in a low-entropy regime throughout (56.7% on HumanEval, Fig. 1c). These observations reveal that MH's uniform proposal is both sample-inefficient and structurally misaligned: it over-samples confident positions while under-exploring the sparse decision



(a) Answer Entropy Distribution **(b) Per-Token Entropy Trace** **(c) Block-level Max Entropy**

Figure 1: Empirical entropy characteristics of reasoning traces. (a) Histogram of per-answer total entropy on merged MATH500, HumanEval, GPQA-Diamond, separated by correctness. (b) Per-token entropy H_t for one selected answer per dataset. (c) Histogram of per-block maximum entropy $\max_t H_t$, shown separately for each dataset.

points that drive quality.

Motivated by these observations, we propose Entropy-Guided Power Sampling (EGPS), which aligns the MH proposal with the entropy structure of reasoning traces. EGPS operates at two complementary granularities. At the block-level, it exploits spatial clustering through a low-entropy gate: any block whose maximum token entropy falls below a threshold θ is skipped, diverting the sampling budget away from near-deterministic regions. At the token-level, it exploits sparsity by drawing resampling points with probability proportional to token entropy, concentrating the budget on the sparse high-entropy clusters that drive quality. To further amplify exploration at these critical positions, we replace the single-proposal MH step with multiple-try Metropolis (MTM), which expands the per-step proposal pool from one to multiple candidates. Beyond the core design, we analyze the effect of the resampling search range on accuracy and provide a full implementation of EGPS within the vLLM inference framework.

Our main contributions are as follows:

(i) We empirically characterize the entropy structure of base-model reasoning traces at both the token and block levels, identifying two structural properties of high-entropy positions: sparsity and spatial clustering. These properties point to a small set of pivotal decision points that shape reasoning quality, motivating a sampling procedure that redirects effort toward them and jointly improves inference efficiency and reasoning accuracy.

(ii) We propose EGPS, a training- and verifier-free power sampler for eliciting reasoning ability from base models at inference time, offering a practical alternative to RL post-training. EGPS integrates three mechanisms: entropy-triggered block skipping, entropy-weighted start-point sampling, and multiple-try Metropolis (MTM). We further analyze the effect of the resampling search range and provide a full vLLM implementation of EGPS.

(iii) On three base models (Qwen2.5-Math-7B, Qwen2.5-7B, Phi-3.5-mini-instruct) and three reasoning benchmarks (MATH500, HumanEval, GPQA), EGPS achieves the best or tied-best accuracy on 5 of 9 model–benchmark combinations—with Qwen2.5-Math-7B reaching best or tied-best on all three benchmarks (MATH500 75.8%, HumanEval 62.2%, GPQA 42.4%)—and delivers up to a 12.6× wall-clock speedup over an MH Power Sampling baseline within the same vLLM framework.

2 Related Work

2.1 Efficient Power Sampling

Karan and Du (2025) formalize inference-time reasoning enhancement as sampling from the sequence-level power distribution p^α via block Metropolis–Hastings. Their key insight is that p^α implicitly accounts for future path likelihoods when selecting each token, a global planning effect that token-level low-temperature sampling cannot replicate. The method matches GRPO on MATH500 without training or verifiers, but its uniform resampling scheme requires regenerating increasingly long suffixes at each block, leading to substantial computational overhead.

Follow-up work improves power sampling along

two directions. On the efficiency side, Ji et al. (2026) show that the power distribution's conditionals decompose into a low-temperature distribution scaled by a future-quality factor, enabling rollout-based token-level approximation that bypasses iterative Markov Chain Monte Carlo (MCMC) entirely. Azizi et al. (2026) replace the serial MH chain with batch-parallel Sequential Monte Carlo (SMC), proving that $\tau = 1/\alpha$ is the minimum-variance prefix-only proposal and shifting the bottleneck to GPU-friendly batch computation. Abdulloh (2025) adjusts MCMC block size based on local entropy, allocating finer granularity to uncertain regions. These methods reduce per-step cost or adjust the computation allocation strategy, but computation remains distributed across all token positions. On the sampling target side, Markovic-Voronov et al. (2026) combine p^α with external reward potentials to define a reward-augmented target distribution and develop SMC-based sampling algorithms, but their reliance on an external reward model limits applicability to tasks where a reliable verifier is available. Unlike the above methods, the proposed EGPS uses only conditional entropy signals already available from the forward pass, selectively concentrating the MCMC budget on sparse positions where meaningful decision divergence exists through entropy-triggered block skipping and entropy-guided intra-block resampling, without requiring any external model. This selective allocation strategy is in principle composable with the above methods.

2.2 Critical Tokens and Local Uncertainty in Reasoning

The selective allocation strategy of EGPS builds on an empirical finding: reasoning difficulty concentrates at a small number of token positions. Lin et al. (2024) identify critical tokens via large-scale rollouts and exploit this finding through token-level contrastive preference optimization (cDPO). Wang et al. (2026) show that high-entropy tokens act as "forking tokens" that steer the model toward different reasoning paths. Yang et al. (2025) find that the entropy gap between correct and incorrect trajectories is driven by a small subset of high-entropy tokens, and propose MTI to apply classifier-free guidance (CFG) (Sanchez et al., 2023) selectively at these positions. Fu et al. (2025) take a confidence perspective, using local log-probability to early-terminate low-quality trajectories (DeepConf).

Prior work has studied related signals—causal influence (Lin et al., 2024), token entropy (Wang et al., 2026; Yang et al., 2025), and log-probability confidence (Fu et al., 2025)—but only for training, decoding, or single-pass sampling. EGPS brings this signal class into MCMC power sampling, where each skipped block saves multiple MCMC iterations rather than a single forward pass.

3 Method

3.1 Preliminaries

Power distributions and MH sampling. Let $\mathcal{X}$ be a finite token vocabulary and p an autoregressive language model that factorizes as

$$p(x_{0:T}) = \prod_{t=0}^{T} p(x_t \mid x_{<t}), \tag{1}$$

where $x_t \in \mathcal{X}$ denotes the token at position t, $x_{<t} = (x_0, \ldots, x_{t-1})$ its prefix, and T the sequence length. The power distribution p^α with sharpening exponent $\alpha > 1$ assigns each sequence a probability proportional to

$$p^\alpha(x_{0:T}) \;\propto\; p(x_{0:T})^\alpha. \tag{2}$$

A larger α concentrates mass on high-likelihood sequences; $\alpha = 1$ recovers the base model. Sampling from p^α is *not* equivalent to low-temperature sampling ($\tau = 1/\alpha$), which exponentiates each conditional $p(x_t \mid x_{<t})$ independently rather than accounting for the exponentiated joint likelihood of all future continuations.

While the unnormalized values $p(\mathbf{x})^\alpha$ are readily available for any sequence $\mathbf{x} = x_{0:T}$, direct sampling requires normalizing over all of $\mathcal{X}^T$, which is computationally intractable. The MH algorithm circumvents this by constructing a Markov chain whose stationary distribution is p^α: given a current sequence $\mathbf{x}$, a candidate $\mathbf{x}'$ drawn from a proposal distribution q is accepted with probability

$$A(\mathbf{x}', \mathbf{x}) = \min\left(1,\ \frac{p(\mathbf{x}')^\alpha \cdot q(\mathbf{x} \mid \mathbf{x}')}{p(\mathbf{x})^\alpha \cdot q(\mathbf{x}' \mid \mathbf{x})}\right), \tag{3}$$

where $p(\mathbf{x})^\alpha$ and $p(\mathbf{x}')^\alpha$ are the unnormalized target densities of the current and candidate sequences, and $q(\mathbf{x}' \mid \mathbf{x})$, $q(\mathbf{x} \mid \mathbf{x}')$ are the forward and reverse proposal probabilities. The normalization constant of p^α cancels in the ratio, so only log-probabilities under p are needed.

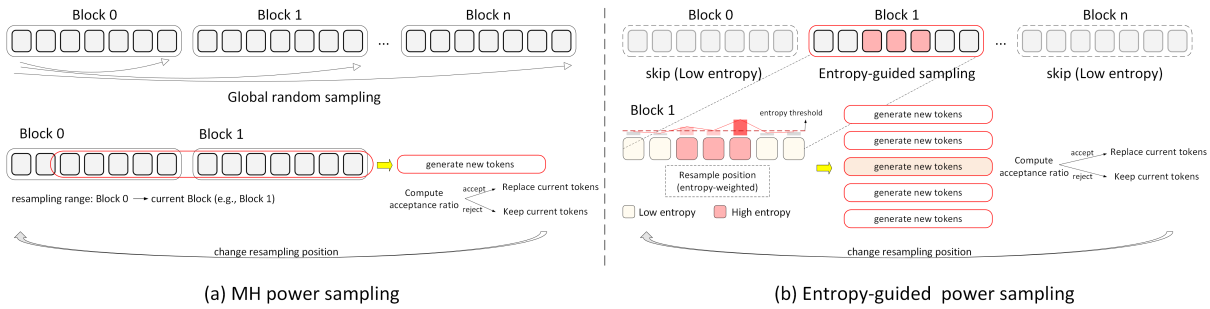


Figure 2: Comparison of MH Power Sampling and EGPS. **(a)** MH Power Sampling generates the sequence block by block; after each block, it uniformly selects a start point from the generated prefix and regenerates from there to the block's end. **(b)** EGPS replaces the uniform proposal with a two-granularity, entropy-guided one: a block-level low-entropy gate skips near-deterministic blocks, and a token-level entropy-weighted distribution concentrates start points on high-entropy positions. Multiple-try Metropolis (MTM) then expands the per-step proposal pool from one to multiple candidates.

Conditional entropy. The conditional entropy at position t is

$$H_t = -\sum_{v \in \mathcal{X}} p(v \mid x_{<t}) \log p(v \mid x_{<t}), \quad (4)$$

where $p(v \mid x_{<t})$ is the model's next-token distribution given prefix $x_{<t}$. H_t quantifies the model's uncertainty at position t and can be computed directly from the logits produced during generation at no extra cost.

3.2 Entropy-Guided Power Sampling

Building on the two structural properties identified in Section 1, namely the sparsity and spatial clustering of high-entropy positions, MH power sampling spends most of its computation on deterministic, low-entropy regions. As shown in Fig. 2, EGPS uses entropy to concentrate the MCMC budget where decisions actually occur, and upgrades the single-proposal MH step to Multiple-Try Metropolis (MTM) to widen the per-step search. This section describes the algorithm (1) and analyzes its computational cost.

Generation and triggering. EGPS partitions the target sequence into blocks of size B and generates them sequentially. For each block, EGPS enters the MCMC refinement loop only when $\max_t H_t > \theta$ within the block; otherwise the block is skipped. Low-entropy blocks rarely contain decision points, so skipping them saves sampling budget.

Position selection. Once a block has triggered refinement, every MCMC step inside the loop must choose a resampling start m. EGPS draws m from a candidate range $S \subseteq [1,\ (k+1)B]$, whose two extremes are $S = [kB{+}1,\ (k+1)B]$ (local) and $S = [1,\ (k+1)B]$ (global). Within S, EGPS narrows the candidates to the high-entropy set $\{t \in S : H_t > \theta\}$ together with a leftward extension of δ tokens, which we call the runway region. Each candidate m in this region is sampled with probability proportional to

$$w_{\text{pos}}(m \mid \mathbf{x}) \ \propto \ \max_{t \in [m,\, m+\delta]} H_t(\mathbf{x}). \quad (5)$$

This concentrates m just before the high-entropy tokens: backing up δ tokens lets the model rephrase the lead-in and traverse the decision point along a different path, whereas resampling at the decision point itself is more likely to reproduce similar hesitations.

Multiple-try proposals and weighted selection. At each MCMC step, EGPS generates K candidate suffixes $\mathbf{x}'_1, \ldots, \mathbf{x}'_K$ in parallel from the proposal distribution q, where $K \geq 1$ is the number of candidates per step. Each $\mathbf{x}'_j$ shares the same prefix as the current state $\mathbf{x}$ and differs only on the suffix from position m onward.

To pick among the K candidates, EGPS assigns each candidate an *importance weight* w_j, defined as the Metropolis–Hastings ratio of candidate j against the current state under the target p^α:

$$w_j \ = \ \frac{p(\mathbf{x}'_j)^\alpha \, q(\mathbf{x} \mid \mathbf{x}'_j)}{p(\mathbf{x})^\alpha \, q(\mathbf{x}'_j \mid \mathbf{x})}. \quad (6)$$

A candidate $\mathbf{x}'_{j^*}$ is then drawn with probability proportional to w_j (here j^* denotes its index), and the move to it is accepted with probability

$$A_{\text{MTM}} \ = \ \min\Big(1,\ \frac{W}{W - w_{j^*} + 1}\Big), \quad (7)$$

Algorithm 1 Entropy-Guided Power Sampling (EGPS)

Input: base p; proposal $q \propto p^{\alpha}$; power α; length T
Hyperparams: block size B; MCMC steps N_{MCMC}; proposals K; entropy threshold θ; runway δ; candidate range $\mathcal{S}$
Output: $(x_1, \ldots, x_T) \sim p^{\alpha}$

for $k \leftarrow 0$ **to** $\lceil T/B \rceil - 1$ **do**
  $x_t \sim q(\cdot \mid x_{<t})$ for $t \in [kB{+}1, (k{+}1)B]$; compute H_t
  if $\max_t H_t \leq \theta$ for $t \in [kB{+}1, (k{+}1)B]$ **then**
    continue ▷ skip low-entropy block
  end if
  for $n \leftarrow 1$ **to** N_{MCMC} **do**
    Identify high-entropy positions $\{t \in \mathcal{S} : H_t > \theta\}$ and extend by δ tokens leftward to form candidate set C
    Sample $m \in C$ with weight $\propto \max_{t' \in [m, m+\delta]} H_{t'}$
    $\mathbf{x}'_j \sim q$ for $j = 1, \ldots, K$, resampling $[m, (k{+}1)B]$
    Compute $\{w_j\}$ (Eq. 6); $W \leftarrow \sum_j w_j$
    $j^* \sim \text{Categorical}(w_1/W, \ldots, w_K/W)$
    $u \sim \text{Uniform}(0, 1)$
    if $u \leq A_{\text{MTM}}$ (Eq. 7) **then** $x_{m:(k+1)B} \leftarrow \mathbf{x}'_{j^*}$
  end for
end for
return $(x_1, \ldots, x_T)$

where $W = \sum_j w_j$ is the total weight summed over all K candidates. At $K = 1$, Eq. (7) reduces to the standard MH acceptance $\min(1, w_1)$.

Computational cost. The original MH sampler regenerates on average $kB/2$ tokens at the k-th block, giving a total of $E_{\text{MH}} \approx N_{\text{MCMC}} \cdot T^2/4B$, quadratic in T. In local mode, EGPS restricts each rewriting step to the current block (at most B tokens), yielding

$$E_{\text{local}} \leq (1 - s) \cdot K \cdot N_{\text{MCMC}} \cdot T, \qquad (8)$$

where s is the block skip rate. The scaling reduces from $O(T^2/B)$ to $O(KT)$. In global mode the worst-case scaling remains $O(KT^2/B)$, but entropy weighting concentrates m near the triggering block, so the empirical cost is substantially lower (see §4.3).

4 Experiment

4.1 Experimental Setup

Datasets and models. We evaluate on three reasoning benchmarks covering distinct domains—MATH500 (Lightman et al., 2024) (mathematical reasoning), HumanEval (Chen et al., 2021) (code generation), and GPQA Diamond (Rein et al., 2023) (graduate-level scientific question answering)—using three publicly available models that span distinct training objectives and capacities: Qwen2.5-Math-7B (Yang et al., 2024) (a math-specialized base model), Qwen2.5-7B (Yang et al., 2024) (a general-purpose base model), and Phi-3.5-mini-instruct (a small-scale instruction-tuned model). All methods are evaluated under a *single-shot* protocol (one response per question), ensuring fair comparison and reflecting realistic deployment.

Baselines. We compare against seven representative methods. For all baselines, we cite the accuracy numbers reported in their papers: Base (τ=1.0) and Low-temperature sampling (τ=0.25), with numbers taken from Karan and Du (2025); Best-of-N (N=32), with numbers taken from Ji et al. (2026); MH Power Sampling (Karan and Du, 2025), Scalable Power Sampling (Ji et al., 2026), Power-SMC (Azizi et al., 2026), and APPS (the reward-free *p-only* variant) (Nguyen et al., 2026).

EGPS configurations. We evaluate EGPS in two configurations: $\textbf{EGPS}_{\textbf{G}}$: the resample range spans all generated tokens, exploring the upper bound of accuracy. $\textbf{EGPS}_{\textbf{L}}$: the resample range is confined to the current block, exploring the upper bound of efficiency. Each MCMC step uses an MTM proposal with K=5 candidate sequences.

Table 1: Main results (pass@1) on MATH500, HumanEval, and GPQA. Higher is better; bold marks the best non-RL result per (model, benchmark). For EGPS, we report the best accuracy over swept thresholds $\theta \in \{0.1, 0.2, 0.4, 0.8\}$; per-$\theta$ accuracy and wall-clock speedup are in §4.3. GRPO (italic) is shown as a reference upper bound from RL post-training and is not compared against.

Method	MATH500	HumanEval	GPQA
Qwen2.5-Math-7B			
Base (Karan and Du, 2025)	0.496	0.329	0.278
Low-temp (Karan and Du, 2025)	0.690	0.512	0.353
Best-of-N (N=32) (Ji et al., 2026)	0.684	0.512	0.343
MH Power Sampling (Karan and Du, 2025)	0.748	0.573	0.389
Scalable PS (Ji et al., 2026)	**0.758**	0.604	0.409
Power-SMC (Azizi et al., 2026)	0.742	0.585	0.349
APPS (p-only) (Nguyen et al., 2026)	0.744	**0.622**	0.323
EGPS$_{\mathrm{G}}$ (MTM)	**0.758**	0.604	**0.424**
EGPS$_{\mathrm{L}}$ (MTM)	0.754	**0.622**	0.384
GRPO (MATH) (Karan and Du, 2025)	*0.785*	*0.537*	*0.399*
Qwen2.5-7B			
Base (Karan and Du, 2025)	0.498	0.329	0.278
Low-temp (Karan and Du, 2025)	0.628	0.524	0.303
Best-of-N (N=32) (Ji et al., 2026)	0.650	0.609	0.282
MH Power Sampling (Karan and Du, 2025)	0.706	0.622	0.318
Scalable PS (Ji et al., 2026)	0.708	**0.756**	0.349
Power-SMC (Azizi et al., 2026)	0.716	0.683	0.313
APPS (p-only) (Nguyen et al., 2026)	**0.726**	0.561	0.354
EGPS$_{\mathrm{G}}$ (MTM)	0.710	0.610	0.354
EGPS$_{\mathrm{L}}$ (MTM)	0.712	0.646	**0.359**
GRPO (MATH) (Karan and Du, 2025)	*0.740*	*0.561*	*0.354*
Phi-3.5-mini-instruct			
Base (Karan and Du, 2025)	0.400	0.213	0.273
Low-temp (Karan and Du, 2025)	0.478	0.585	0.293
MH Power Sampling (Karan and Du, 2025)	0.508	**0.732**	**0.364**
EGPS$_{\mathrm{G}}$ (MTM)	**0.520**	0.677	0.343
EGPS$_{\mathrm{L}}$ (MTM)	0.504	0.695	0.338
GRPO (MATH) (Karan and Du, 2025)	*0.406*	*0.134*	*0.359*

Implementation and speedup measurement. To ensure comparability with published baselines, we adopt the same hyperparameters as Karan and Du (2025): block size B=192, power exponent α=4, proposal temperature τ=0.25, N_{MCMC}=10 MCMC steps per refinement block, and maximum generation length $T_{\max}$=3072. The entropy threshold θ is selected via ablation (see §4.3). All experiments are conducted on the NVIDIA A100-40G GPU, with EGPS and all vLLM-based baselines implemented on vLLM 0.11.0 (PyTorch 2.8.0, CUDA 12.8, Transformers 4.55.2, Python 3.10). Since vLLM accelerates inference for any algorithm through KV-cache and batched scheduling optimizations, comparing EGPS (on vLLM) against published speedup numbers from non-vLLM implementations would conflate algorithmic with framework-level gains. We therefore re-implement MH Power Sampling on vLLM—preserving its original algorithm (uniform-position resampling with single-proposal MH)—and report all speedups against this vLLM-based MH Power Sampling baseline; accuracy numbers, in contrast, are taken directly from the original papers.

4.2 Main Results

Table 1 reports pass@1 accuracy and Figure 3 (bottom row) reports wall-clock speedup on Qwen2.5-Math-7B relative to a same-framework vLLM-based MH Power Sampling (*MH PS*) (Karan and Du, 2025) baseline; EGPS matches or surpasses reward-free SOTA on most settings while delivering up to a 12.6× speedup.

EGPS improves over MH PS on most model–benchmark combinations, with the smallest gains on the most training-aligned task and the largest on broader reasoning: on Qwen2.5-Math-7B the absolute accuracy gains are +1.0% (MATH500), +4.9% (HumanEval), +3.5% (GPQA); on Qwen2.5-7B they are +0.6%, +2.4%, +4.1%. This matches our central premise: when output quality is bottlenecked by a small number of

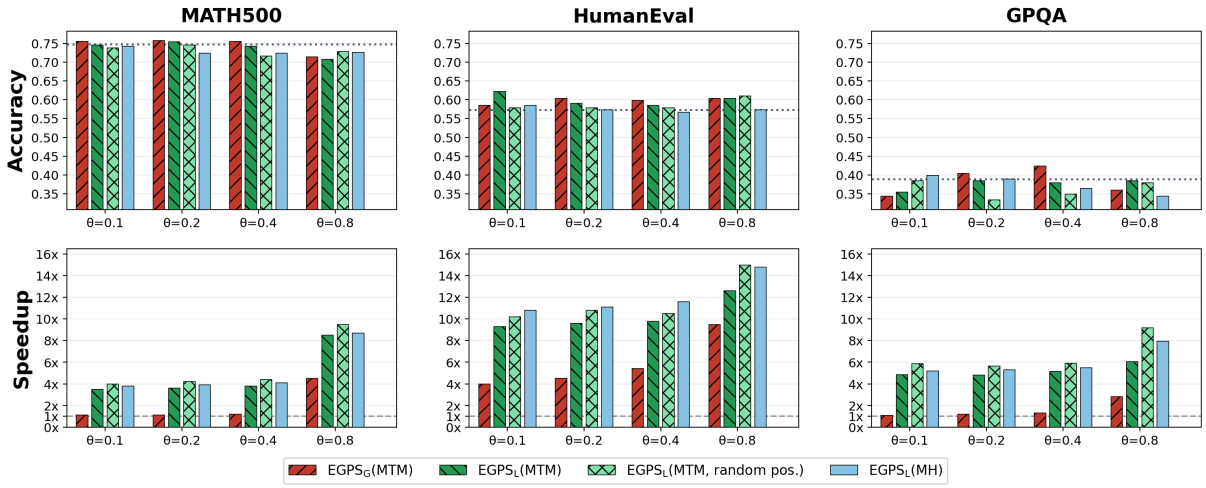


Figure 3: Ablation on Qwen2.5-Math-7B, sweeping $\theta \in \{0.1, 0.2, 0.4, 0.8\}$ on MATH500, HumanEval, and GPQA. Top row: pass@1 accuracy (dotted line: vLLM-based MH Power Sampling baseline). Bottom row: wall-clock speedup over the same baseline (dashed line at $1\times$). EGPS$_G$(MTM) and EGPS$_L$(MTM) use Global and Local resample range; EGPS$_L$(MTM, random pos.) replaces entropy-guided position sampling with random sampling; EGPS$_L$(MH) replaces MTM with single-proposal MH.

critical decisions, targeting those decisions yields larger returns than adding generic sampling effort.

The pattern also holds against more compute-hungry baselines: Best-of-N (Ji et al., 2026) (N=32 generations), Scalable Power Sampling (Scalable PS) (Ji et al., 2026), Power-SMC (Azizi et al., 2026) (multi-chain or particle-based MCMC), and APPS (Nguyen et al., 2026), each using substantially more sampling than EGPS's single chain over entropy-skipped blocks. EGPS matches or exceeds them on most settings: on Qwen2.5-Math-7B it ties Scalable PS on MATH500 (75.8%) and APPS on HumanEval (62.2%) and surpasses both on GPQA (42.4% vs. 40.9%/32.3%); on Qwen2.5-7B, EGPS$_L$ surpasses Power-SMC on GPQA (35.9% vs. 31.3%) and Best-of-N on MATH500 by +6.2%. Two settings favor multi-sample methods: Scalable PS leads EGPS$_L$ by +11.0% on Qwen2.5-7B HumanEval, and on Phi-3.5-mini-instruct MH PS retains the lead on HumanEval and GPQA, which we attribute to a sharper output distribution at small instruction-tuned scale that leaves less high-entropy mass to redirect.

The G/L contrast is task-dependent. EGPS$_G$ outperforms EGPS$_L$ on math and science reasoning (Qwen-Math GPQA 42.4% vs. 38.4%; Phi-3.5 MATH500 52.0% vs. 50.4%), consistent with errors propagating across blocks; the ranking reverses on code generation, where EGPS$_L$ wins (Qwen-Math HumanEval 62.2% vs. 60.4%; Qwen-7B HumanEval 64.6% vs. 61.0%), consistent with errors localized within the current block. The two configurations suit different error structures rather than acting as a single hyperparameter.

On efficiency, EGPS$_L$ reaches 3.5–8.5× on MATH500, 9.3–12.6× on HumanEval, and 4.8–6.1× on GPQA on Qwen2.5-Math-7B, with EGPS$_G$ reaching 1.1–4.5×, 4.0–9.5×, 1.1–2.8× respectively. Because the baseline runs in the same framework (vLLM), these are algorithmic gains, attributable to entropy-triggered block skipping (avoiding refinement on near-deterministic blocks) and entropy-weighted position sampling (focusing MCMC where correction is needed).

Sampling-based methods (MH PS, Scalable PS, Power-SMC, APPS, and EGPS) avoid the pass@k and cross-task degradation associated with RL-based distribution sharpening (Yue et al., 2025): GRPO (Karan and Du, 2025) trained on MATH reaches 78.5% on Qwen2.5-Math-7B MATH500 but collapses on Phi-3.5 (40.6% MATH500, 13.4% HumanEval), while these methods hold stable performance bands within each base model. EGPS matches the strongest sampling baselines with a single MCMC chain, and on Qwen2.5-Math-7B surpasses GRPO on HumanEval (+8.5%) and GPQA (+2.5%); EGPS$_L$ also exceeds GRPO on Qwen2.5-7B GPQA (35.9% vs. 35.4%). EGPS$_G$ and EGPS$_L$ span a deployment curve along accuracy and efficiency.

4.3 Ablation Study

We ablate EGPS on Qwen2.5-Math-7B by sweeping $\theta \in \{0.1, 0.2, 0.4, 0.8\}$ across four variants (Figure 3): EGPS_G(MTM) and EGPS_L(MTM) are the full configurations with Global and Local resample range; EGPS_L(MTM, random pos.) replaces entropy-guided position sampling with uniform random sampling; and EGPS_L(MH) replaces multiple-try Metropolis with single-proposal MH. The sweep isolates two design components (entropy-guided position sampling and MTM) and one tunable hyperparameter (θ).

Replacing entropy-guided position sampling with uniform random sampling causes the largest damage on the hardest reasoning task. The gap between EGPS_L(MTM) and EGPS_L(MTM, random pos.) is small on MATH500 and HumanEval across θ, but widens sharply on GPQA, reaching $+5.1\%$ at $\theta = 0.2$ (38.4% vs. 33.3%). This mirrors the cross-model pattern observed in the main results: when reasoning quality is bottlenecked by a small number of high-entropy decisions, uniform sampling spends MCMC budget on positions where the base model is already confident, while the entropy-weighted variant concentrates effort exactly where it matters. On tasks where most positions either matter or are already deterministic, the choice of sampling distribution carries less weight.

Replacing MTM with single-proposal MH causes consistent but more uniform accuracy drops. EGPS_L(MTM) outperforms EGPS_L(MH) by $+3.0\%$ on MATH500 at $\theta = 0.2$ (75.4% vs. 72.4%) and by comparable margins elsewhere, with the gap holding broadly across θ rather than scaling with task difficulty. This makes MTM look less like a fix targeting any specific failure mode and more like a generic per-step boost from broader proposal diversity. The two components are therefore complementary: entropy-weighted position sampling routes MCMC effort to the right positions, while MTM improves the proposal made at each chosen position.

The threshold θ exposes an accuracy–efficiency knob with a favourable curve. Speedup grows roughly monotonically with θ across all benchmarks (e.g. EGPS_L on MATH500: $3.5\times$ at $\theta = 0.1$ rising to $8.5\times$ at $\theta = 0.8$), while accuracy stays close to the full configuration on MATH500 and HumanEval throughout the sweep and only degrades more visibly on GPQA at high θ. Higher θ skips more low-entropy blocks, so EGPS increasingly trusts the base model on near-deterministic regions; the curve degrades only when θ becomes high enough to skip blocks that contain critical decisions. In practice this monotonicity makes θ a deployment knob that can be picked once per task family via a small validation sweep, without further model surgery or retraining.

Taken together, these ablations decompose EGPS's gains into separable contributions: position sampling drives the task-specific accuracy improvements seen in the main results, MTM provides a stable per-step boost across tasks, and θ exposes a smooth deployment curve between accuracy-favoring and efficiency-favoring operating points. Removing either component degrades accuracy in measurably different patterns, indicating the two are non-redundant.

5 Conclusion

In this paper, we presented Entropy-Guided Power Sampling (EGPS), a training-free and verifier-free sampler for eliciting RL-level reasoning from base language models. EGPS re-derives its proposal from the token-level entropy already produced in the forward pass: it skips deterministic blocks, localizes each Metropolis–Hastings move to a high-entropy neighborhood, and applies Multiple-Try Metropolis at decision points, so sampling cost scales with entropy mass rather than sequence length. Experiments on three base models and three reasoning benchmarks show that EGPS attains the best or tied-best accuracy on 5 of 9 settings and yields up to a $12.6\times$ wall-clock speedup over the MH baseline; a token-level analysis confirms the gain stems from the proposal design rather than the inference framework. We hope these findings encourage further study of structurally-aware, inference-time alternatives to RL post-training. In future work, we plan to extend EGPS to Sequential Monte Carlo variants of power sampling and to develop controllable mechanisms for inference-time reasoning elicitation.

Limitations

Our experiments cover models up to 7B parameters on three reasoning benchmarks spanning mathematics, code, and science; whether the sparsity and spatial clustering of high-entropy tokens that EGPS exploits persists at 70B+ scales or transfers to non-reasoning regimes such as open-ended generation and dialogue remains open. EGPS exposes a single

entropy threshold θ, currently tuned per (model, benchmark) pair via ablation; automatic selection (e.g. calibration from a small held-out set or online estimation from the running entropy distribution) would remove the manual tuning in our pipeline. Finally, we do not formally characterize how the entropy-guided proposal affects the mixing of the power distribution target p^α; a mixing-time analysis would clarify when the observed gains should be expected to hold.

References

Abdulloh Abdulloh. 2025. Uncertainty-driven adaptive sampling for resource-efficient language model inference. *Authorea Preprints*.

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

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Yichao Fu, Xuewei Wang, Yuandong Tian, and Jiawei Zhao. 2025. Deep think with confidence. *arXiv preprint arXiv:2508.15260*.

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

Xiaotong Ji, Rasul Tutunov, Matthieu Zimmer, and Haitham Bou Ammar. 2026. Scalable power sampling: Unlocking efficient, training-free reasoning for llms via distribution sharpening. *arXiv preprint arXiv:2601.21590*.

Aayush Karan and Yilun Du. 2025. Reasoning with sampling: Your base model is smarter than you think. *arXiv preprint arXiv:2510.14901*.

Hunter Lightman, Vineet Kosaraju, Yuri Burda, Harrison Edwards, Bowen Baker, Teddy Lee, Jan Leike, John Schulman, Ilya Sutskever, and Karl Cobbe. 2024. Let's verify step by step. In *International Conference on Learning Representations*, volume 2024, pages 39578–39601.

Zicheng Lin, Tian Liang, Jiahao Xu, Qiuzhi Lin, Xing Wang, Ruilin Luo, Chufan Shi, Siheng Li, Yujiu Yang, and Zhaopeng Tu. 2024. Critical tokens matter: Token-level contrastive estimation enhances llm's reasoning capability. *arXiv preprint arXiv:2411.19943*.

Jelena Markovic-Voronov, Wenhui Zhu, Bo Long, Zhipeng Wang, Suyash Gupta, Kayhan Behdin, Bee-Chung Chen, and Deepak Agarwal. 2026. Sampling for quality: Training-free reward-guided llm decoding via sequential monte carlo. *arXiv preprint arXiv:2604.16453*.

Tu Nguyen, Rasul Tutunov, Xiaotong Ji, and Matthieu Zimmer. 2026. The model knows, the decoder finds: Future value guided particle power sampling. *arXiv preprint arXiv:2605.02427*.

David Rein, Betty Li Hou, Asa Cooper Stickland, Jackson Petty, Richard Yuanzhe Pang, Julien Dirani, Julian Michael, and Samuel R Bowman. 2023. Gpqa: A graduate-level google-proof q&a benchmark. *arXiv preprint arXiv:2311.12022*.

Guillaume Sanchez, Honglu Fan, Alexander Spangher, Elad Levi, Pawan Sasanka Ammanamanchi, and Stella Biderman. 2023. Stay on topic with classifier-free guidance. *arXiv preprint arXiv:2306.17806*.

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, YK Li, Yang Wu, and 1 others. 2024. Deepseekmath: Pushing the limits of mathematical reasoning in open language models. *arXiv preprint arXiv:2402.03300*.

Yuda Song, Julia Kempe, and Remi Munos. 2025. Outcome-based exploration for llm reasoning. *arXiv preprint arXiv:2509.06941*.

Shenzhi Wang, Le Yu, Chang Gao, Chujie Zheng, Shixuan Liu, Rui Lu, Kai Dang, Xiong-Hui Chen, Jianxin Yang, Zhenru Zhang, and 1 others. 2026. Beyond the 80/20 rule: High-entropy minority tokens drive effective reinforcement learning for llm reasoning. *Advances in Neural Information Processing Systems*, 38:115452–115486.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, and 43 others. 2024. Qwen2 technical report. *Preprint*, arXiv:2407.10671.

Zhen Yang, Mingyang Zhang, Feng Chen, Ganggui Ding, Liang Hou, Xin Tao, and Ying-Cong Chen. 2025. Less is more: Improving llm reasoning with minimal test-time intervention. *arXiv preprint arXiv:2510.13940*.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Shiji Song, and Gao Huang. 2025. Does reinforcement learning really incentivize reasoning capacity in llms beyond the base model? *arXiv preprint arXiv:2504.13837*.