

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

Hong Guo¹, Nianhui Guo², Christoph Meinel³, Haojin Yang²,

¹Hasso Plattner Institute, ²GreenBit.AI, ³German University of Digital Science,

hong.guo@hpi.de nianhui.guo@greenbit.ai

christoph.meinel@german-uds.de haojin.yang@greenbit.ai

Abstract

Sampling from the sequence-level power distribution 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: the power distribution mainly departs from the base distribution at a sparse, spatially clustered set of high-entropy decision points, yet MH proposes re-sampling 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 approximate 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. EGPS targets significant speedup with accuracy that matches or improves on MH power sampling: across three base models and three benchmarks it attains best or tied-best accuracy on 5 of 9 settings; on Qwen2.5-Math-7B it is best or tied-best on all three (MATH500 75.8%, HumanEval 62.2%, GPQA 42.4%), with wall-clock speedups over the MH baseline of up to $12.6\times$ across these benchmarks. Our code is available at <https://github.com/hongguo7/EGPS>.

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 re-generates 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-

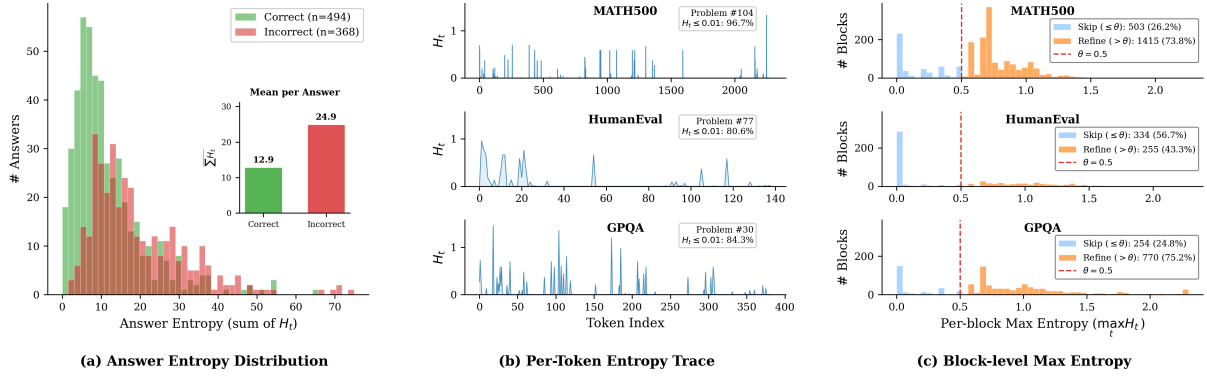


Figure 1: Empirical entropy characteristics of reasoning traces. (a) Histogram of per-answer total entropy on merged MATH500, HumanEval, GPQA-Diamond, separated by correctness. (b) Per-token entropy H_t for one selected answer per dataset. (c) Histogram of per-block maximum entropy $\max_t H_t$, shown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

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 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 approximate 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. 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 comparable or better accuracy than an MH Power Sampling baseline within the same vLLM framework—best or tied-best on 5 of 9 model-benchmark combinations; on Qwen2.5-Math-7B it is best or tied-best on all three benchmarks (MATH500 75.8%, HumanEval 62.2%, GPQA 42.4%) with wall-clock speedups of up to $12.6\times$ across them.

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 can-

not 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. A framework-matched comparison with this method is given in Appendix I. 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 trajec-

tories 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

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

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

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

We write $\tilde{p}(x_{0:T}) := p(x_{0:T})^\alpha$ for this unnormalized target density, so that $p^\alpha = \tilde{p}/Z$ with normalizer $Z = \sum_{\mathbf{x} \in \mathcal{X}^T} \tilde{p}(\mathbf{x})$; throughout, p denotes the normalized base-model distribution and $\tilde{p}$ the unnormalized target. 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 density $\tilde{p}(\mathbf{x})$ is readily available for any sequence $\mathbf{x} = x_{0:T}$, the normalizer Z requires summing 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

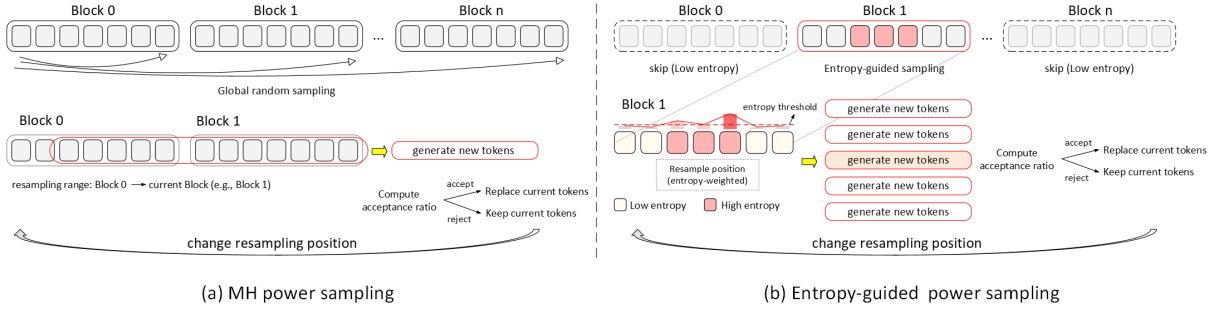


Figure 2: Comparison of MH Power Sampling and EGPS. **(a)** MH Power Sampling generates the sequence block by block; after each block, it uniformly selects a start point from the generated prefix and regenerates from there to the block’s end. **(b)** EGPS replaces the uniform proposal with a two-granularity, 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.

distribution q is accepted with probability

$$A(\mathbf{x}', \mathbf{x}) = \min\left(1, \frac{\tilde{p}(\mathbf{x}') \cdot q(\mathbf{x} | \mathbf{x}')}{\tilde{p}(\mathbf{x}) \cdot q(\mathbf{x}' | \mathbf{x})}\right), \quad (3)$$

where $\tilde{p}(\mathbf{x})$ and $\tilde{p}(\mathbf{x}')$ are the unnormalized target densities of the current and candidate sequences, and $q(\mathbf{x}' | \mathbf{x})$, $q(\mathbf{x} | \mathbf{x}')$ are the forward and reverse proposal probabilities. The normalizer Z cancels in the ratio, so only log-probabilities under p are needed. Following Karan and Du (2025), we instantiate the proposal by low-temperature resampling: a resampling position is chosen, and the suffix is regenerated tokenwise from $q = p_\tau$ with temperature $\tau = 1/\alpha$, i.e., each conditional $p(x_t | x_{<t})$ tempered at τ . The MH correction in Eq. (3) accounts for the gap between this token-level proposal and the sequence-level target p^α .

Conditional entropy. The conditional entropy at position t is

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

where $p(v | x_{<t})$ 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 §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 (Algorithm 1) and analyzes its computational cost.

Generation and triggering. EGPS partitions the target sequence into blocks of size B , indexed by $k = 0, 1, \dots, \lceil T/B \rceil - 1$ (block k covers positions $[kB+1, (k+1)B]$), 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 does so in two stages. First, it fixes a search 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). The local range restricts each move to the current block and is cheaper ($O(KT)$; see the cost analysis below), suiting errors confined to a block; the global range can also revise earlier blocks ($O(KT^2/B)$ in the worst case), suiting errors that propagate across blocks—empirically, global is stronger on math and science, local on code (§4.2). Second, within S , EGPS restricts the candidates to the high-entropy set $\{t \in S : H_t > \theta\}$ and extends it leftward by δ tokens to form the runway region, from which m is sampled with probability

Algorithm 1 Entropy-Guided Power Sampling (EGPS)

Input: base p ; proposal $q = p_\tau$, $\tau=1/\alpha$ (low-temp.); power α ; length T

Hyperparams: block size B ; MCMC steps N_{MCMC} ; proposals K ; entropy threshold θ ; runway δ ; candidate range S

Output: $(x_1, \dots, x_T) \sim p^\alpha$

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

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 $\mathbf{x}'_1, \dots, \mathbf{x}'_K$ 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 :

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

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

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

where $W = \sum_j w_j$ is the total weight summed over all K candidates. This multiple-proposal mechanism is the independent multiple-try Metropolis (I-MTM) scheme of Martino (2018): Eqs. (6) and (7) are the standard I-MTM weight and acceptance ratio (Eqs. 18–19 therein), expressed with each weight normalized by the current state, so the $+1$ in the denominator is the normalized current-state term rather than an added constant (derivation in Appendix B). Unlike standard MTM, I-MTM requires no $K-1$ auxiliary reference draws to balance the current state, making it simpler and computationally cheaper. At $K = 1$, Eq. (7) reduces to the standard MH acceptance $\min(1, w_1)$.

Sources of approximation. Two of the mechanisms above make the EGPS kernel a controlled approximation rather than an exact p^α -invariant sampler. First, since the resampling position is drawn non-uniformly ($m \sim w_{\text{pos}}(\cdot \mid \mathbf{x})$, Eq. (5)), the forward and reverse position probabilities no longer cancel: the exact acceptance ratio carries an additional position factor $w_{\text{pos}}(m \mid \mathbf{x}')/w_{\text{pos}}(m \mid \mathbf{x})$ multiplying Eq. (3), and in MTM each weight w_j

carries its own such factor; EGPS omits these factors. Second, entropy-triggered skipping removes MCMC steps on low-entropy blocks, where the conditionals are already near-deterministic and refinement would change essentially no tokens. Both deviations are empirically negligible: restoring the position factor shifts the log acceptance ratio by an amount concentrated tightly at zero with no systematic direction, and a Kolmogorov–Smirnov test on per-sequence mean log-likelihood cannot distinguish end-to-end EGPS (skips included) from MH power sampling for the local variants, while the global variant shifts toward higher likelihood—a task-favorable sharpening consistent with its stronger downstream accuracy. Derivations and measurements are in Appendix A.

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 (derivation in Appendix C). In local mode, EGPS restricts each rewriting step to the current block (at most B tokens), yielding

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

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 ac-

curacy numbers reported in their papers: Base ($\tau=1.0$) and Low-temperature sampling ($\tau=0.25$), 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: **EGPS_G**: the resample range spans all generated tokens, exploring the upper bound of accuracy. **EGPS_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.

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 $\alpha=4$, proposal temperature $\tau=0.25$, $N_{\text{MCMC}}=10$ MCMC steps per refinement block, and maximum generation length $T_{\text{max}}=3072$. 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. Prompt templates are listed in Appendix H.

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. Positioned for significant speedup at comparable-or-better accuracy, EGPS matches or surpasses reward-free SOTA on most settings while delivering up to a $12.6\times$ speedup;

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- θ 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. Baseline accuracies are cited from their source papers (§4.1).

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 _G (MTM)	0.758	0.604	0.424
EGPS _L (MTM)	0.754	0.622	0.384
GRPO (MATH) (Karan and Du, 2025)	<i>0.785</i>	<i>0.537</i>	<i>0.399</i>
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 _G (MTM)	0.710	0.610	0.354
EGPS _L (MTM)	0.712	0.646	0.359
GRPO (MATH) (Karan and Du, 2025)	<i>0.740</i>	<i>0.561</i>	<i>0.354</i>
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 _G (MTM)	0.520	0.677	0.343
EGPS _L (MTM)	0.504	0.695	0.338
GRPO (MATH) (Karan and Du, 2025)	<i>0.406</i>	<i>0.134</i>	<i>0.359</i>

settings where multi-sample baselines lead are analyzed below.

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 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

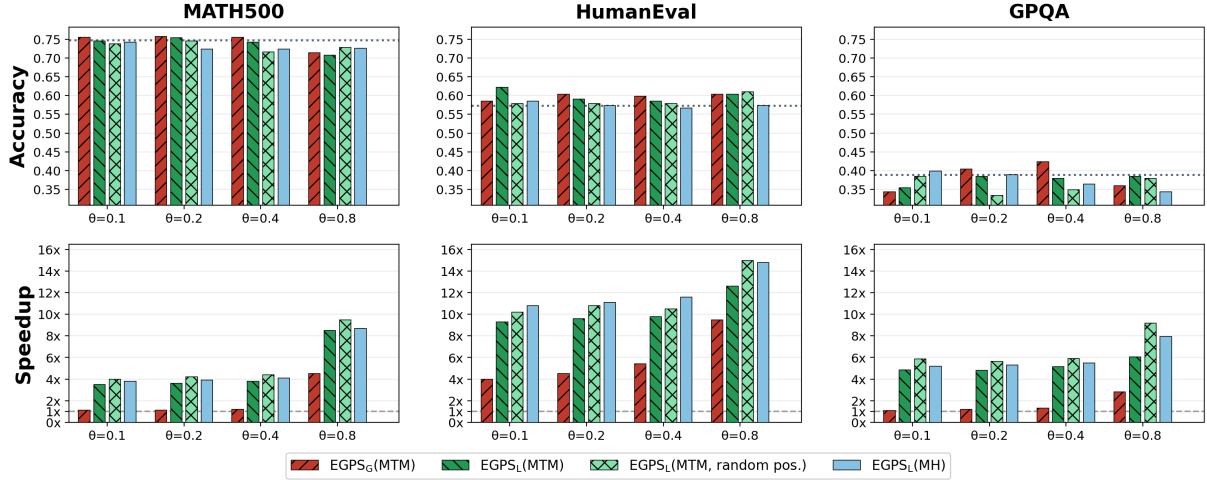


Figure 3: Ablation on Qwen2.5-Math-7B, sweeping $\theta \in \{0.1, 0.2, 0.4, 0.8\}$ on MATH500, HumanEval, and GPQA. Top row: pass@1 accuracy (dotted line: vLLM-based MH Power Sampling baseline). Bottom row: wall-clock speedup over the same baseline (dashed line at $1\times$). $\text{EGPS}_G(\text{MTM})$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

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 $\times$ on MATH500, 9.3–12.6 $\times$ on HumanEval, and 4.8–6.1 $\times$ on GPQA on Qwen2.5-Math-7B, with EGPS_G reaching 1.1–4.5 $\times$, 4.0–9.5 $\times$, 1.1–2.8 $\times$ 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). Hardware-independent, mechanism-level efficiency metrics—forward-pass counts, skip statistics, and peak memory—are in Appendix F.

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 effi-

ciency.

Table 1 follows the single-seed best-accuracy protocol of Karan and Du (2025). To verify robustness, we additionally rerun EGPS and our reproduced MH PS baseline over 8 seeds with two-sided paired t -tests: within $\theta \leq 0.4$, the MATH500 gains are significant for both variants at the 5% level, while HumanEval and GPQA remain on par at large speedups; full multi-seed tables and analysis are in Appendix E.

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; full numerical results in Appendix G): $\text{EGPS}_G(\text{MTM})$ and $\text{EGPS}_L(\text{MTM})$ are the full configurations with Global and Local resample range; $\text{EGPS}_L(\text{MTM, random pos.})$ replaces entropy-guided position sampling with uniform random sampling; and $\text{EGPS}_L(\text{MH})$ 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 $\text{EGPS}_L(\text{MTM})$ and $\text{EGPS}_L(\text{MTM, random pos.})$ 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. $\text{EGPS}_L(\text{MTM})$ outperforms $\text{EGPS}_L(\text{MH})$ 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 is stable and insensitive to the exact value within $\theta \in [0.1, 0.4]$. Higher θ skips more low-entropy blocks, so EGPS increasingly trusts the base model on near-deterministic regions; the curve degrades once θ is high enough to skip blocks that contain critical decisions. Where this happens depends on the dataset’s block-entropy distribution (Fig. 1c): on MATH500, whose per-block entropy mass peaks at 0.6–0.75, accuracy drops systematically at $\theta = 0.8$ (confirmed over 8 seeds, Appendix E), whereas HumanEval and GPQA peak outside this range and show no systematic drop—single-seed dips at high θ are within seed noise. In practice we recommend $\theta \in [0.1, 0.4]$ as a robust default requiring no per-benchmark tuning—lower θ favors accuracy, higher favors speed—picked once without 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. Further component-isolating ablations—proposal count K , MCMC depth, run-way, and selection rule—are in Appendix D.

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 on Qwen2.5-Math-7B; 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.

Limitations

Our experiments cover models up to 7B parameters on three reasoning benchmarks spanning mathematics, code, and science. The two structural properties EGPS exploits—sparsity and spatial clustering of high-entropy positions (Fig. 1)—are validated in this regime; whether they persist at 70B+ scales, or under stronger instruction tuning, which sharpens the output distribution and leaves less redirectable entropy mass (§4.2), remains open. In non-reasoning regimes such as open-ended generation and dialogue, high entropy often reflects stylistic freedom rather than decision points, so the motivation itself may weaken. EGPS exposes a single entropy threshold θ ; accuracy is insensitive to its exact value within the recommended range $[0.1, 0.4]$ (§4.3), but the optimal value depends on the dataset’s block-entropy distribution, and automatic selection (e.g. calibration from a small held-out set or online estimation from the running entropy distribution) remains future work. 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.

Acknowledgments

The computing resources underlying this work were funded by the Federal Ministry of Research, Technology and Space under the funding code “KI-Servicezentrum Berlin-Brandenburg” 16IS22092. Responsibility for the content of this publication remains with the author.

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, Xuwei 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*.
- Luca Martino. 2018. A review of multiple try mcmc algorithms for signal processing. *Digital Signal Processing*, 75:134–152.
- 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*.

Appendix

A Exactness of the EGPS Kernel

This appendix derives the exact acceptance ratios that a fully p^α -invariant version of EGPS would require, identifies the factor our implementation omits, and measures the empirical effect of the omission together with that of entropy-triggered skipping.

Exact acceptance ratios. In EGPS the resampling position is drawn non-uniformly, $m \sim w_{\text{pos}}(\cdot \mid \mathbf{x})$ (Eq. 5), so the forward and reverse position probabilities differ and no longer cancel in the MH ratio. For $K = 1$, the exact acceptance probability is

$$A_{\text{full}} = \min\left(1, \frac{\tilde{p}(\mathbf{x}') q(\mathbf{x} \mid \mathbf{x}')}{\tilde{p}(\mathbf{x}) q(\mathbf{x}' \mid \mathbf{x})} \cdot \frac{w_{\text{pos}}(m \mid \mathbf{x}')}{w_{\text{pos}}(m \mid \mathbf{x})}\right), \quad (9)$$

whose first factor is the ratio of Eq. (3) and whose second factor is the position factor. For $K > 1$ (I-MTM), each candidate weight carries its own factor,

$$w_j^{\text{full}} = \frac{\tilde{p}(\mathbf{x}'_j) q(\mathbf{x} \mid \mathbf{x}'_j)}{\tilde{p}(\mathbf{x}) q(\mathbf{x}'_j \mid \mathbf{x})} \cdot \frac{w_{\text{pos}}(m \mid \mathbf{x}'_j)}{w_{\text{pos}}(m \mid \mathbf{x})}, \quad (10)$$

with the acceptance rule of Eq. (7) applied to $W = \sum_j w_j^{\text{full}}$. Omitting the position factor from Eq. (9) and from every weight in Eq. (10) yields exactly the simplified ratios used by our implementation (Eqs. 3 and 7). Entropy-triggered skipping (§3.2) is the second approximation: a block is skipped iff its maximum token entropy is below θ , where the conditionals are near-deterministic, so the removed MCMC steps are those whose proposals would almost surely reproduce the current tokens—skipping removes near-trivial steps rather than discarding probability mass that MH would have moved.

Experiment 1: acceptance-ratio level. We measure the change induced by restoring the position factor, on the resampling steps actually executed by running EGPS (Qwen2.5-Math-7B, all 500 MATH500 problems); skipped blocks perform no resampling and involve no position factor, so this measurement cleanly isolates the factor. For $K = 1$ the omitted factor’s log, $\Delta_{\text{pos}} = \log w_{\text{pos}}(m \mid \mathbf{x}') - \log w_{\text{pos}}(m \mid \mathbf{x})$, is the entire effect on the log ratio; for $K = 5$, since A_{MTM} depends on the sum of weights, we measure ΔA_{MTM} , the change in $\log A_{\text{MTM}}$ after restoring the factor

to every w_j . Table 2 shows both concentrate tightly at zero with no systematic direction; ΔA_{MTM} is tighter still because $A_{\text{MTM}} = \min(1, \cdot)$ is often saturated.

Experiment 2: output-distribution level. We compare full, skip-enabled EGPS against MH power sampling (Karan and Du, 2025) with a two-sample Kolmogorov–Smirnov test on the per-sequence mean log-likelihood under the base model p (Table 3); this end-to-end comparison covers the combined effect of both approximations. The local variants are statistically indistinguishable from MH power sampling; the global variant shifts toward higher likelihood—a task-favorable sharpening consistent with its stronger downstream accuracy (§4.2)—while the reference row (MH power sampling vs. plain sampling at $\alpha = 1$) confirms the test has power to detect real distributional shifts. Taken together, neither the omitted position factor nor entropy-triggered skipping produces a detectable distributional deviation for the local variants, and the global variant’s deviation is a beneficial sharpening.

B Derivation of the I-MTM Acceptance Ratio

Our multiple-proposal mechanism is independent multiple-try Metropolis (I-MTM) (Martino, 2018); Eqs. (6) and (7) are the standard I-MTM weight and acceptance ratio, Eqs. (18)–(19) of that survey (the I-MTM algorithm in its Table 6). Unlike standard MTM, I-MTM does not require drawing $K - 1$ auxiliary reference points to balance the current state, which makes it simpler and computationally more efficient.

Martino’s acceptance ratio is expressed via the unnormalized importance weight $\Omega(\mathbf{z}) = \tilde{p}(\mathbf{z})/q(\mathbf{z})$, where q denotes suffix regeneration from the prefix $x_{<m}$ shared by the current state and all candidates: its numerator is the sum of the weights over all K candidates, and its denominator is the current-state term $\Omega(\mathbf{x})$ plus the weights of the remaining $K - 1$ candidates,

$$A = \min\left(1, \frac{\sum_{n=1}^K \Omega(\mathbf{x}'_n)}{\Omega(\mathbf{x}) + \sum_{n \neq j^*} \Omega(\mathbf{x}'_n)}\right). \quad (11)$$

We define each candidate weight as its ratio against the current state, which is exactly Eq. (6):

$$w_j = \frac{\Omega(\mathbf{x}'_j)}{\Omega(\mathbf{x})} = \frac{\tilde{p}(\mathbf{x}'_j) q(\mathbf{x} \mid \mathbf{x}'_j)}{\tilde{p}(\mathbf{x}) q(\mathbf{x}'_j \mid \mathbf{x})}. \quad (12)$$

Config	Quantity	$ \Delta < 0.1$	Mean	Med.	Std
EGPS _L (MH)	Δ_{pos}	66.5%	0.0004	0.000	0.302
EGPS _G (MTM)	ΔA_{MTM}	92.2%	-0.031	0.000	1.036
EGPS _L (MTM)	ΔA_{MTM}	92.9%	-0.020	0.000	0.224

Table 2: Effect of restoring the position factor, measured on the resampling steps executed by running EGPS (Qwen2.5-Math-7B, MATH500; MH: $K=1$, MTM: $K=5$). For the MH row, 1.4% of accepted steps are reverse-unreachable: after the move, the entropy-guided candidate region of the new sequence no longer contains m , so $w_{\text{pos}}(m \mid \mathbf{x}') = 0$ and $\Delta_{\text{pos}} = -\infty$; these steps are excluded from the statistics and reported separately.

Comparison vs. MH PS	KS D	KS p -value	Verdict
EGPS _L (MH)	0.040	0.82	indist.
EGPS _L (MTM)	0.052	0.51	indist.
EGPS _G (MTM)	0.146	$< 10^{-4}$	higher lik.
MH PS vs. base ($\alpha=1$)	0.126	$< 10^{-3}$	distinct

Table 3: Two-sample KS test on per-sequence mean log-likelihood under the base model (Qwen2.5-Math-7B, MATH500). Local EGPS variants are statistically indistinguishable from MH power sampling; the global variant shifts toward higher likelihood. The last row confirms the test detects real distributional shifts.

Dividing the numerator and denominator of Eq. (11) by $\Omega(\mathbf{x})$: the numerator becomes $W = \sum_n w_n$, the current-state term becomes 1, and the remaining candidates contribute $W - w_{j^*}$, yielding Eq. (7):

$$A_{\text{MTM}} = \min\left(1, \frac{W}{W - w_{j^*} + 1}\right). \quad (13)$$

Hence Eq. (7) is the equivalent form of the I-MTM acceptance ratio rather than a heuristic construction; the $+1$ is the current-state term $\Omega(\mathbf{x})$ after normalization, not an added constant. The selected candidate $\mathbf{x}'_{j^*}$ is drawn with probability proportional to w_j —the standard weighted selection of I-MTM (PMF $\bar{w}_n = w_n / \sum_i w_i$). At $K = 1$ the denominator is $w_1 - w_1 + 1 = 1$, recovering the standard MH acceptance $\min(1, w_1)$. Thus the MTM mechanism introduces no approximation beyond those analyzed in Appendix A.

C Complexity Derivation

MH power sampling baseline. With sequence length T , block size B , and T/B blocks, the MH sampler of Karan and Du (2025) draws, at the k -th block, a resampling start uniformly from the prefix of length $\approx kB$, regenerating $\approx kB/2$ tokens per step in expectation. Summing over blocks and

multiplying by the N_{MCMC} steps per block:

$$E_{\text{MH}} \approx N_{\text{MCMC}} \sum_{k=1}^{T/B} \frac{kB}{2} \approx \frac{N_{\text{MCMC}} T^2}{4B}, \quad (14)$$

quadratic in T .

EGPS. The local mode restricts each rewrite to the current block, so the regenerated length is at most B , independent of k . With K proposals per step ($K=1$: MH move; $K>1$: MTM move) and block skip rate s , the total cost is bounded as in Eq. (15), so B cancels and the complexity in T drops from $O(T^2/B)$ to $O(KT)$ —the source of EGPS_L’s speedup. The global mode allows the start anywhere in the prefix, so the worst case remains $O(KT^2/B)$; but entropy weighting concentrates resample points near the triggering block, so the empirical cost is far lower (§4.3). Larger K raises the nominal cost linearly, but our vLLM implementation batches the K proposals, so the added wall-clock latency is small.

D Component-Isolating Ablations

We isolate each design choice under a matched compute budget (MATH500, Qwen2.5-Math-7B, A100 40G; mean $\pm$ std over 8 seeds). The reference is our reproduced MH Power Sampling (global range, no skipping, uniform position), 0.721 ± 0.010 . Throughout, $K=1$ denotes an MH move and $K>1$ an MTM move.

Tuning protocol. We first fix the MTM configuration on the uniform-position variant (EGPS_L, random position; Tables 4 and 5): $K=5$, 10 MCMC steps, weighted selection. Tuning on the uniform-position variant means K , the step count, and the selection rule are chosen independently of the position-selection strategy, so that when entropy-weighted position selection is introduced afterwards, its own contribution can be isolated cleanly. We then fix this configuration and vary one component at a time (Tables 6 and 7).

$$E_{\text{local}} \leq (1 - s) K N_{\text{MCMC}} \cdot \frac{T}{B} \cdot B = (1 - s) K N_{\text{MCMC}} T \quad (15)$$

Findings.

- **MH vs. MTM** (Table 7, $K=1$ vs. $K=5$): the multi-proposal step is better for $\theta \leq 0.4$ (e.g., .735 vs. .716 at $\theta=0.1$).
- **Entropy-weighted vs. uniform position** (EGPS_L vs. random pos.): entropy weighting wins for $\theta \leq 0.4$ (.735/.738/.730 vs. .723/.721/.709); both rows use block skipping, so the gap comes from position weighting alone.
- **More steps instead of MTM** ($K=1$, 50 steps): reaches $.751 \pm .006$ at $\theta=0.1$ but requires 50 serial MCMC steps; $K=5$ MTM attains comparable accuracy in 10 serial steps.
- **Runway** (Table 6): locking the start onto high-entropy positions (runway = 0) is best, with $\delta \leq 8$ nearly identical; spreading onto low-entropy positions degrades accuracy (e.g., .724 at 16)—high-entropy positions are the critical resample locations.
- **Weighted vs. greedy (best-of- K) selection** (Tables 4 and 7): weighted matches or beats greedy; greedy only catches up at $K=15$, so always taking the best proposal is not optimal.
- **Global vs. local**: EGPS_G is uniformly more accurate at the cost of the wall-clock speedup dropping from $\sim 3.5\times$ to $\sim 1.1\times$ (measured separately, §4.3), so we provide both variants.

E Statistical Reliability over 8 Seeds

Table 1 reports single-seed best accuracy following the protocol of [Karan and Du \(2025\)](#). Here we rerun EGPS and our reproduced MH Power Sampling baseline over 8 seeds (0–7) and report mean $\pm$ std, the gain Δ_{acc} over the baseline with a two-sided paired t -test, and the 8-seed wall-clock speedup (Table 8). Within the recommended range $\theta \leq 0.4$:

- **MATH500: significant gains for both variants.** EGPS_G : +2.3 to +2.5 points, significant at the 1% level at $1.1\text{--}1.2\times$ speedup; EGPS_L : +0.9 to +1.7 points, significant at the 5% level at $3.5\text{--}3.8\times$. G gives the larger accuracy gain, L the larger speedup.
- **HumanEval / GPQA: on par, large speedup.** All mean differences are positive but not signif-

icant; par accuracy is the design goal, traded for $4.8\text{--}9.8\times$ (EGPS_L) and $1.1\text{--}5.4\times$ (EGPS_G) speedup.

- **GPQA non-significance is a variance effect.** The baseline fluctuates by ± 0.041 across seeds, masking moderate gains, while EGPS is more stable ($\pm 0.018\text{--}0.033$).
- **Accuracy drops only at the non-recommended $\theta=0.8$ on MATH500,** whose block-entropy mass peaks at $0.6\text{--}0.75$ (Fig. 1c), so this threshold skips critical blocks; HumanEval and GPQA peak outside this range and do not drop—conversely confirming that skipping high-entropy blocks is what costs accuracy.

Overall, the gains are stable and consistent with our claim of speedup with accuracy that improves or stays on par: within $\theta \leq 0.4$, speedup spans $1.1\text{--}9.8\times$ across variants and datasets (up to $12.6\times$ at higher θ).

F Mechanism-Level Efficiency Metrics

Wall-clock speedup is hardware-dependent, so we complement it with hardware-independent, mechanism-level efficiency metrics (Table 9) plus a measured GPU peak-memory figure. Our implementation decodes one token per forward pass and counts all K MTM candidate chains, so “generated tokens” equals the total number of forward passes—the most direct hardware-independent measure of compute.

Compute decreases monotonically with θ . As θ rises, the trigger rate, the number of refined blocks, the forward calls, and the generated-to-output token ratio all fall while the final answer length is essentially unchanged, cutting total forward compute by roughly $3\text{--}5\times$ at matched length between $\theta=0.1$ and $\theta=0.8$. Both variants show the same trend, indicating that the speedup comes from genuinely less refinement rather than implementation or hardware effects.

The gate is deterministic. A block is skipped iff its maximum token entropy is below θ , so no false positives or negatives arise from the trigger itself; that the selected positions are the answer-changing ones is validated separately by

K	1	3	5	10	15
Total cand.	10	30	50	100	150
Weighted	.705±.015	.716±.012	.723±.011	.712±.012	.717±.013
Greedy	.710±.008	.707±.010	.712±.015	.709±.009	.727±.007

Table 4: Proposal count K and selection rule on EGPS_L (random position), step = 10, $\theta=0.1$; MATH500, 8 seeds. Weighted at $K=5$ is the best trade-off; greedy (best-of- K) only catches up at $K=15$.

Steps	10	5	15	20
K	5	10	5	5
Total cand.	50	50	45	50
Serial calls	10	5	15	20
Accuracy	.723±.011	.705±.019	.723±.013	.728±.008

Table 5: MCMC depth on EGPS_L (random position), weighted, $\theta=0.1$; MATH500, 8 seeds. At matched candidate budget, accuracy barely improves beyond 10 steps while time grows proportionally; $K=10/\text{step} = 5$ (same 50 candidates, half the depth) is clearly worse—deeper serial exploration beats wider parallel proposals at fixed budget.

Runway δ	0	4	8	16	32
Accuracy	.739±.012	.739±.012	.738±.012	.724±.015	.732±.009
Accept. (%)	82.7	82.4	82.1	82.0	81.5
Gen. tokens	17153	17155	17283	17304	17800

Table 6: Runway sensitivity on full EGPS_L (entropy-weighted position, $K=5$, step = 10, $\theta=0.2$); MATH500, 8 seeds. Locking to high-entropy positions (runway = 0) is best, with $\delta \leq 8$ nearly identical; spreading further can hurt (e.g., runway = 16).

Method	$\theta=0.1$	$\theta=0.2$	$\theta=0.4$	$\theta=0.8$
L (random pos.)	.723±.011	.721±.017	.709±.005	.707±.013
L ($K=5$, MTM)	.735±.009	.738±.012	.730±.006	.698±.011
L ($K=1$, MH)	.716±.016	.708±.010	.713±.011	.704±.013
L ($K=5$, greedy)	.734±.013	.734±.017	.735±.013	.703±.008
G ($K=5$, MTM)	.746±.009	.744±.009	.744±.007	.710±.010

Table 7: Component effects at the fixed base configuration ($K=5$, step = 10, weighted, runway = 8) across entropy thresholds θ ; MATH500, Qwen2.5-Math-7B, 8 seeds. L/G abbreviate EGPS_L/EGPS_G. All rows use entropy-triggered block skipping; “random pos.” uses uniform intra-block position selection. Reference: reproduced MH Power Sampling = 0.721 ± 0.010 (constant over θ).

the position-selection ablation (Appendix D).

Peak memory is independent of θ . Since EGPS changes neither the model weights, the KV-cache budget, nor K , peak GPU memory stays at 36.7 GB (A100 40G) across all θ and both variants.

G Per- θ Numerical Results

Table 10 lists the full numerical results behind Figure 3. Accuracy is the best value from a single seed (matching the protocol of Table 1); speedup is averaged over 8 seeds and measured against our vLLM reproduction of MH power sampling, so it reflects the algorithm rather than framework factors.

H Prompts and Decoding Configuration

Apart from EGPS’s three new parameters—the proposal count K , the runway δ , and the entropy threshold θ —the prompt templates, the zero-shot chain-of-thought setup, and all decoding hyperparameters match MH Power Sampling (Karan and Du, 2025); the shared hyperparameter values are listed in §4.1. Table 11 lists the zero-shot prompts. For the base models (Qwen2.5-Math-7B, Qwen2.5-7B) the prompt is fed directly as plain text; for the instruction-tuned Phi-3.5-mini-instruct the same content is wrapped as a user message in the model’s chat template with the generation prompt appended.

Method	θ	Accuracy	Δ_{acc}	Speedup
MATH500 — baseline $0.721 \pm 0.010, 1\times$				
EGPS _G	0.1	0.746 ± 0.009	$+0.025^{**}$	$1.1\times$
EGPS _G	0.2	0.744 ± 0.009	$+0.023^{**}$	$1.1\times$
EGPS _G	0.4	0.744 ± 0.007	$+0.023^{**}$	$1.2\times$
EGPS _G	$0.8^\dagger$	0.710 ± 0.010	-0.011	$4.5\times$
EGPS _L	0.1	0.735 ± 0.009	$+0.015^{**}$	$3.5\times$
EGPS _L	0.2	0.738 ± 0.013	$+0.017^*$	$3.6\times$
EGPS _L	0.4	0.730 ± 0.006	$+0.009^*$	$3.8\times$
EGPS _L	$0.8^\dagger$	0.698 ± 0.012	-0.023^{***}	$8.5\times$
HumanEval — baseline $0.551 \pm 0.023, 1\times$				
EGPS _G	0.1	0.564 ± 0.013	$+0.013$	$4.0\times$
EGPS _G	0.2	0.557 ± 0.026	$+0.006$	$4.5\times$
EGPS _G	0.4	0.573 ± 0.020	$+0.022$	$5.4\times$
EGPS _G	0.8	0.579 ± 0.025	$+0.027^*$	$9.5\times$
EGPS _L	0.1	0.566 ± 0.026	$+0.015$	$9.3\times$
EGPS _L	0.2	0.568 ± 0.021	$+0.017$	$9.6\times$
EGPS _L	0.4	0.555 ± 0.018	$+0.004$	$9.8\times$
EGPS _L	0.8	0.568 ± 0.025	$+0.017$	$12.6\times$
GPQA — baseline $0.339 \pm 0.041, 1\times$				
EGPS _G	0.1	0.349 ± 0.018	$+0.010$	$1.1\times$
EGPS _G	0.2	0.354 ± 0.031	$+0.015$	$1.2\times$
EGPS _G	0.4	0.357 ± 0.031	$+0.018$	$1.3\times$
EGPS _G	0.8	0.360 ± 0.021	$+0.021$	$2.8\times$
EGPS _L	0.1	0.340 ± 0.025	$+0.001$	$4.9\times$
EGPS _L	0.2	0.343 ± 0.033	$+0.004$	$4.8\times$
EGPS _L	0.4	0.343 ± 0.029	$+0.004$	$5.2\times$
EGPS _L	0.8	0.363 ± 0.030	$+0.024$	$6.1\times$

Table 8: Statistical reliability of EGPS vs. the reproduced MH Power Sampling baseline over 8 seeds (Qwen2.5-Math-7B). Accuracy is the 8-seed mean $\pm$ std; Δ_{acc} is the gain over the baseline with a two-sided paired t -test (*: significant at the 5% level; **: 1%; ***: 0.1%); speedup is the 8-seed wall-clock ratio. † On MATH500, $\theta=0.8$ is a non-recommended high-threshold setting (see text). Recommended range: $\theta \leq 0.4$.

Variant	θ	Trig. rate	Accept. (%)	Refined	Skipped	Skip/Refine	Gen/Out ($\times$)	Gen. tokens	Out. tokens
EGPS _G (MTM)	0.1	.779	61.8	15375	4358	0.28	89.0	63091	708.6
EGPS _G (MTM)	0.2	.753	61.4	14543	4784	0.33	84.2	59332	704.9
EGPS _G (MTM)	0.4	.702	60.3	13341	5668	0.42	76.7	53875	702.2
EGPS _G (MTM)	0.8	.255	57.4	4115	12049	2.93	19.5	12813	655.7
EGPS _L (MTM)	0.1	.805	82.6	13955	3385	0.24	26.6	17743	666.2
EGPS _L (MTM)	0.2	.777	82.1	13330	3837	0.29	26.0	17283	664.2
EGPS _L (MTM)	0.4	.727	80.2	12287	4621	0.38	24.6	16253	659.7
EGPS _L (MTM)	0.8	.255	68.6	3940	11526	2.93	9.4	5996	635.5

Table 9: Mechanism-level efficiency metrics of EGPS (hardware-independent). Qwen2.5-Math-7B, MATH500, $K=5$, 8 seeds, A100 40G. “Accept.” is the MCMC acceptance rate; “Gen. tokens” equals the total number of forward passes (one token per pass, all K candidate chains counted); “Refined”/“Skipped” count blocks; a smaller Gen/Out ratio means fewer generated tokens per final answer token. Generated and output tokens are per-problem averages.

I Comparison with Entropy-Based Adaptive Sampling

Reproduction setup. Abdulloh (2025) is the closest work to ours: it likewise uses entropy to allocate the MCMC budget, but does so by adaptively adjusting the block size (finer blocks in uncertain regions), whereas EGPS keeps the block size fixed, skips low-entropy blocks outright, and concentrates resampling on high-entropy positions. The method

is not open-sourced and its original implementation is in PyTorch; to obtain a framework-matched comparison, we reproduced it on vLLM following the description in the paper, denoted Adaptive † (single seed). Parameters follow the best configuration reported in that paper: $b_{\min}=48$, $b_{\max}=384$, uncertainty threshold 1.8, max aggregation, and $N_{\text{MCMC}}=10$; $\alpha=4$, $\tau=0.25$, and a maximum generation length of 3072 are shared with MH Power

Dataset	Method	Accuracy				Speedup ($\times$)			
		$\theta=.1$	.2	.4	.8	$\theta=.1$	.2	.4	.8
MATH500	EGPS _G (MTM)	.756	.758	.756	.714	1.1	1.1	1.2	4.5
MATH500	EGPS _L (MTM)	.746	.754	.742	.708	3.5	3.6	3.8	8.5
MATH500	EGPS _L (MH)	.742	.724	.724	.726	3.8	3.9	4.1	8.7
MATH500	EGPS _L (MTM, rand.)	.738	.746	.716	.728	4.0	4.2	4.4	9.5
HumanEval	EGPS _G (MTM)	.585	.604	.598	.604	4.0	4.5	5.4	9.5
HumanEval	EGPS _L (MTM)	.622	.591	.585	.604	9.3	9.6	9.8	12.6
HumanEval	EGPS _L (MH)	.585	.573	.567	.573	10.8	11.1	11.6	14.8
HumanEval	EGPS _L (MTM, rand.)	.579	.579	.579	.610	10.2	10.8	10.5	15.0
GPQA	EGPS _G (MTM)	.343	.404	.424	.359	1.09	1.19	1.32	2.82
GPQA	EGPS _L (MTM)	.354	.384	.379	.384	4.85	4.80	5.15	6.05
GPQA	EGPS _L (MH)	.399	.389	.364	.343	5.2	5.3	5.5	7.93
GPQA	EGPS _L (MTM, rand.)	.384	.333	.349	.379	5.85	5.64	5.92	9.2

Table 10: Full per- θ numerical results behind Figure 3 (Qwen2.5-Math-7B). Accuracy is the single-seed best; speedup is the 8-seed average relative to the vLLM reproduction of MH Power Sampling.

Benchmark	Zero-shot prompt
MATH500	Can you solve the following math problem? {question} Please reason step by step, and put your final answer within <code>\boxed{}</code> .
HumanEval	The unmodified dataset prompt (function signature and docstring), used directly as the completion prefix with no additional instruction.
GPQA Diamond	Answer the following multiple choice question. The last line of your response should be of the following format: ' <code>\boxed{\$LETTER}</code> ' (without quotes) where LETTER is one of ABCD (ex. ' <code>\boxed{A}</code> '). Think step by step before answering. {Question} A) {A} B) {B} C) {C} D) {D}

Table 11: Zero-shot prompt templates, identical to MH Power Sampling (Karan and Du, 2025). Typewriter text is the literal template; {·} marks the slot filled per problem.

Sampling. The paper states that this threshold is calibrated on a held-out set for a target compute budget, and we adopt its reported value. We use base models, and GPQA answers are graded with the grader from the original repository of Karan and Du (2025). The efficiency comparison is run on an A100-40G GPU on GPQA Diamond, with all three methods measured under the same hardware and the same framework.

Results. Accuracy is broadly comparable between the two methods (Table 12). On the two models for which the original paper reports numbers, the two trade wins: on Qwen2.5-7B, Adaptive is slightly ahead on MATH500 and about 3 points behind on GPQA, while on Phi-3.5-mini-instruct it is stronger on HumanEval and GPQA—consistent with the observation in §4.2 that instruction tuning sharpens the output distribution and leaves less redirectable high-entropy mass. Differences on GPQA should not be over-interpreted: the baseline’s own

cross-seed standard deviation on this benchmark reaches ± 0.041 (Appendix E).

Efficiency differs substantially (Table 13). EGPS_L is $5.0\text{--}8.2\times$ faster than Adaptive[†] across the three models; taking the same-framework MH Power Sampling baseline on Qwen2.5-Math-7B as an absolute reference (356.4 seconds per problem, 8 seeds), Adaptive[†] runs at $0.96\times$ —that is, no speedup at all—whereas EGPS_L reaches $4.83\times$. The reason lies in the different form of intervention: the adaptive strategy lowers the mean committed block from 192 to roughly 120 tokens, and the resulting increase in block count raises the number of MCMC rounds accordingly, so it reallocates computational granularity without reducing the total budget. EGPS instead removes computation: low-entropy blocks are skipped entirely and each move is confined to the current block, so cost scales with entropy mass rather than sequence length (§3.2).

Benchmark	Adaptive (reported)	Adaptive [†] (ours)	EGPS _G	EGPS _L
Qwen2.5-Math-7B				
MATH500	—	0.720	0.758	0.754
HumanEval	—	0.524	0.604	0.622
GPQA	—	0.337	0.424	0.384
Qwen2.5-7B				
MATH500	0.714	0.688	0.710	0.712
HumanEval	0.645	0.634	0.610	0.646
GPQA	0.326	0.293	0.354	0.359
Phi-3.5-mini-instruct				
MATH500	0.521	0.504	0.520	0.504
HumanEval	0.748	0.665	0.677	0.695
GPQA	0.372	0.328	0.343	0.338

Table 12: Accuracy comparison (pass@1). All numbers are single-seed results, with both methods evaluated at their own best configuration: EGPS entries are the best over the θ sweep (as in Table 1), and Adaptive[†] uses the best configuration reported in the original paper. On Qwen2.5-Math-7B×GPQA, 5 problems whose sequences reached the model’s 4096-token context limit crashed the inference engine and produced no output; they are excluded ($n=193$). The original paper does not report Qwen2.5-Math-7B, so only reproduced numbers are available for that model. Our reproduction falls 1–8 points below the reported numbers; without a reference implementation it may not capture every implementation detail. Eight-seed means for Qwen2.5-Math-7B are given in Appendix E.

Model	Method	Sec./problem	Gen./out. tokens	vs. Adaptive [†]
Qwen2.5-Math-7B	Adaptive [†]	370.8	38.1×	1.00×
	EGPS _G , $\theta=0.1$	330.5	101.1×	1.12×
	EGPS _L , $\theta=0.1$	73.8	22.6×	5.03×
Qwen2.5-7B	Adaptive [†]	348.7	44.2×	1.00×
	EGPS _G , $\theta=0.1$	200.1	64.3×	1.74×
	EGPS _L , $\theta=0.1$	53.8	22.9×	6.48×
Phi-3.5-mini-instruct	Adaptive [†]	202.5	50.3×	1.00×
	EGPS _G , $\theta=0.1$	82.1	85.6×	2.47×
	EGPS _L , $\theta=0.1$	24.6	23.6×	8.23×

Table 13: Efficiency comparison on GPQA Diamond (A100-40G, vLLM). Statistics are computed over all problems. The generated-to-output token ratio is a full count for both methods: for EGPS it includes all K MTM candidate chains, and for Adaptive[†] it includes tokens discarded by lookahead decoding as well as rejected proposals. Because the K candidates are batched in parallel under vLLM, this ratio upper-bounds EGPS’s latency cost—which is why EGPS_G remains faster than Adaptive[†] despite a ratio of 101×