EXPLORE BROADLY, REASON SHARPLY: PUSH SMALL MODELS TOWARD THE FRONTIER VIA SAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Power-sharpened sampling is an inference-time alternative to reinforcement-learning (RL) post-training for enhancing reasoning in large language models (LLMs). High-probability sequences are amplified under the base model without parameter updates or external rewards, avoiding the costly optimization and jagged generalization of RL. However, this approach faces a fundamental exploration–exploitation trade-off, as strong sharpening restricts exploration, trapping samplers in plausible but incorrect reasoning trajectories, whereas weak sharpening leaves the answer distribution diffuse. To resolve this trade-off, we introduce **Parallel Power Tempering (PPT)**, instantiating power-sharpened LLM sampling via parallel tempering. Running multiple *interacting* replicas in parallel at different sharpening levels allows lower-power replicas to explore diverse reasoning trajectories and higher-power chains to further exploit higher-likelihood responses favored by the sharpened target. Specifically, we tailor PPT to inference-time sampling by mitigating a truncation bias, identified in prior power samplers, and investigate effective swap strategies under finite memory and compute budgets. Extensive experimentation shows that PPT substantially improves single-chain power-sharpened sampling and outperforms RL-post-trained models, producing higher-quality reasoning traces and even achieving performance comparable to frontier models.

1 INTRODUCTION

Reinforcement Learning (RL)-guided post-training with task-specific rewards is a widely used recipe for improving the reasoning capabilities in LLMs (Shao et al., 2024; Guo et al., 2025). This approach is particularly effective in domains such as mathematics and code generation, where automated verifiers can reliably assess correctness. However, RL post-training requires access to a verifier or reward model that scores sampled outputs, and it requires expensive gradient-based updates to the model's parameters (Shao et al., 2024). Moreover, reliable rewards are often unavailable for open-ended scientific inquiry, deliberation, and long-horizon planning. Even when rewards exist, optimizing for a narrow set of rewarded tasks can erode prior capabilities and yield *"jagged" generalization* to nearby problems (Hu et al., 2026). These limitations motivate inference-time alternatives that improve reasoning without parameter updates or external rewards.

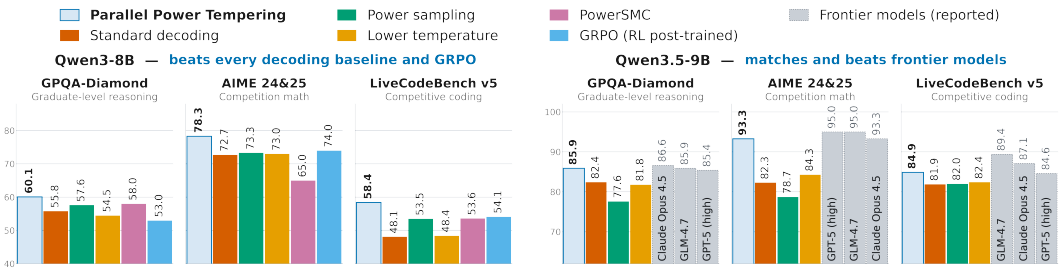

Figure 1: Single-shot accuracy (%) of Parallel Power Tempering. Left: Qwen3-8B across three benchmarks. Right: Qwen3.5-9B on GPQA-Diamond, AIME 24&25 and LiveCodeBench v5 against decoding baselines and the reported scores of frontier models.

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

Traditional inference-time methods improve generation either by selecting a high-scoring sequence among multiple completions (Huang et al., 2025; Kang et al., 2025), or by locally reshaping token probabilities with temperature and truncation rules (Holtzman et al., 2020; Meister et al., 2023; Nguyen et al., 2025; Tang et al., 2025). The latter are inherently *myopic*, as they modify each decoding step without directly controlling the distribution over complete responses. Neither approach generally samples from a prescribed sequence-level target.

In this vein, recent works suggest that a model's capability can be substantially enhanced at inference time by sampling from a sharpened distribution over the model's outputs (Karan & Du, 2026; Ji et al., 2026). Such sequence-level targets naturally depend on Monte Carlo methods for controllable generation and blockwise resampling (Mireshghallah et al., 2022; Forristal et al., 2023). The cornerstone hypothesis is that reasoning trajectories are latent in models, and thus, sequence-level sharpening amplifies these trajectories, enabling sampling from them without additional RL post-training while yielding comparable inference-time reasoning performance.

However, power-sharpened sampling introduces a fundamental inference-time challenge. Sharpening concentrates the sampling distribution around high-likelihood responses, which can improve final-answer quality, but excessive concentration can trap the sampler in locally plausible yet incorrect reasoning paths (Karan & Du, 2026; Ji et al., 2026). This behavior is especially consequential for multi-step reasoning tasks, where an early mistake can lead to a coherent but wrong solution (Li et al., 2025). Conversely, flatter distributions explore a broader range of alternative reasoning paths, but may also generate noisy or lower-quality responses that are unsuitable as final outputs (Troshin et al., 2025). Thus, power-sharpened LLM sampling faces a central **exploration–exploitation trade-off**: the sampler must explore diverse reasoning trajectories while still exploiting the sharpened distribution to produce high-quality answers.

To address this challenge, we introduce **Parallel Power Tempering (PPT)**, a power-sharpened parallel tempering sampler for LLM inference. Our method adapts classical parallel tempering, also known as replica exchange, from multi-modal MCMC (Swendsen & Wang, 1986; Geyer, 1991; Hukushima & Nemoto, 1996; Earl & Deem, 2005) to sequence-level power-sharpened LLM sampling. Parallel tempering transfers naturally to the LLM regime. Instead of running a single sampler at one sharpening level, PPT runs replicas in parallel over a ladder of sharpening levels. Unlike prior LLM replica-exchange methods that vary prefix length (He et al., 2026), all replicas share a common horizon, which enables whole-record swaps. The lower-sharpening replicas act as exploratory chains that can search broadly across possible responses, while the higher-sharpening replicas act as exploitative chains that concentrate on responses favored by the sharpened objective. The replicas are coupled through swap moves, which allow neighboring replicas to exchange their generated responses according to a principled and tractable Metropolis–Hastings acceptance rule (Deng et al., 2020; Syed et al., 2022). Every swap decision therefore reduces to the base-model log-probabilities of the two responses being exchanged, which are already cached from generation, so a swap requires no additional model evaluations. In this way, promising reasoning paths discovered by exploratory replicas can be transferred to sharper replicas, while sharper replicas can escape poorly mixed regions by exchanging with flatter chains. We evaluate PPT across a range of small open-source models on benchmarks spanning mathematics, coding, and STEM reasoning. Our main contributions can be summarized as follows:

- We introduce PPT, a novel power-sharpened parallel tempering sampler specially adapted to LLM sampling that runs a ladder of replicas at various sharpening levels and couples adjacent replicas through swap moves, letting exploratory low-power chains feed diverse states to exploitative high-power chains, mitigating the central limitation on the exploration–exploitation trade-off.
- We identify a *structural truncation bias* in prior implementations of early-stopped power sampling, which parallel tempering cannot repair, and eliminate it via a fixed-horizon construction that preserves the true sharpened target. Since every swap schedule is exact under this construction, we further study efficient swap strategies subject to memory and compute constraints.
- Extensive experiments across math, STEM, and code reasoning benchmarks on multiple open models show that PPT substantially outperforms sampling and RL baselines in nearly all settings. Notably, with Qwen3.5-9B, it even matches or exceeds the performance of frontier models, as illustrated in Figure 1.

2 PRELIMINARIES

2.1 NOTATIONS

Let $\mathbf{x}_0$ denote a given prompt and let $\mathbf{x}_{0:T} = (\mathbf{x}_0, x_1, \ldots, x_T)$ be a sequence of generated tokens appended to that prompt, where each x_t belongs to a finite vocabulary $\mathcal{V}$. An autoregressive LLM factorizes the joint distribution over $\mathbf{x}_{0:T}$ via the chain rule $p_0(\mathbf{x}_{1:T}|\mathbf{x}_0) = \prod_{t=1}^{T} p_0(x_t \mid \mathbf{x}_{0:t-1})$. At fixed horizon T, let $\mathcal{S}_T$ be the finite state space containing sequences of length-T and records terminated at $\tau \leq T$. Post-terminal entries are filled with deterministic padding leading to a state $(x_1, \ldots, x_{\tau-1}, \text{EOS}, \perp, \ldots, \perp)$, so $p_0(\mathbf{x}_{1:T} \mid \mathbf{x}_0)$ through T or until EOS. Sampling proceeds sequentially: at each step $t = 1, \ldots, T$, a token is drawn from $x_t \sim p_0(\cdot \mid \mathbf{x}_{0:t-1})$.

2.2 POWER SHARPENING FOR LLMS

Recent work proposes sampling from a *power-sharpened* variant of this distribution (Karan & Du, 2026; Ji et al., 2026):

$$\pi_\alpha(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \frac{p_0(\mathbf{x}_{1:T} \mid \mathbf{x}_0)^\alpha}{Z_\alpha(\mathbf{x}_0)}, \qquad Z_\alpha(\mathbf{x}_0) = \sum_{\mathbf{x}'_{1:T} \in \mathcal{V}^T} p_0(\mathbf{x}'_{1:T} \mid \mathbf{x}_0)^\alpha, \tag{1}$$

where $\alpha > 1$ is the sharpening parameter and $Z_\alpha(\mathbf{x}_0)$ is the normalizing constant. Raising the base distribution to a power of α suppresses low-likelihood continuations while amplifying high-likelihood ones (Karan & Du, 2026). However, direct autoregressive sampling from π_α is intractable, since the normalizing constant in Eq. 1 requires summing the powered likelihood over all possible future completions. For this reason Metropolis–Hastings sampling was employed.

Metropolis–Hastings (MH) is an MCMC algorithm for sampling from a distribution known only up to a normalizing constant (Metropolis et al., 1953; Hastings, 1970). Given a target density $\pi_\alpha(\mathbf{x})$, MH constructs a Markov chain $\mathbf{x}^{(0)} \to \mathbf{x}^{(1)} \to \cdots \to \mathbf{x}^{(n)}$ as follows. From the current state $\mathbf{x}^{(i)}$, a candidate $\mathbf{y}$ is drawn from a proposal distribution. Let g_α be the tokenwise-powered proposal

$$g_\alpha(x_t \mid \mathbf{x}_0, \mathbf{x}_{<t}) = \frac{p_0(x_t \mid \mathbf{x}_0, \mathbf{x}_{<t})^\alpha}{\sum_{v \in \mathcal{V}} p_0(v \mid \mathbf{x}_0, \mathbf{x}_{<t})^\alpha}. \tag{2}$$

After EOS, its only admissible output is $\perp$, with probability one. Although tractable, g_α is a tokenwise proposal and is not the sequence-level target π_α. Following Karan & Du (2026), draw a restart position r from a state-independent distribution, retain the prefix $\mathbf{x}_{<r}$, and resample the suffix using

$$q_r(\mathbf{y} \mid \mathbf{x}) = \mathbf{1}\{\mathbf{y}_{<r} = \mathbf{x}_{<r}\} \prod_{t=r}^{T} g_\alpha(y_t \mid \mathbf{x}_0, \mathbf{y}_{<t}).$$

If r lies after the current EOS, the padding convention makes this a self-transition. Conditional on the sampled r, accept the proposal with probability

$$A_r(\mathbf{x}, \mathbf{y}) = \min\left\{1, \frac{\pi_\alpha(\mathbf{y} \mid \mathbf{x}_0) q_r(\mathbf{x} \mid \mathbf{y})}{\pi_\alpha(\mathbf{x} \mid \mathbf{x}_0) q_r(\mathbf{y} \mid \mathbf{x})}\right\}. \tag{3}$$

Each MH restart preserves π_α; thus, the state-independent mixture over r also preserves π_α.

2.3 PARALLEL TEMPERING

Parallel tempering (PT) is an MCMC technique designed to improve mixing for target distributions that are difficult to explore directly, such as multimodal or highly concentrated distributions (Swendsen & Wang, 1986; Earl & Deem, 2005). Instead of running one chain, PT runs multiple chains, or *replicas*, in parallel over a ladder of tempered distributions.

PT introduces a sequence of auxiliary distributions $\pi_1(\mathbf{x}), \pi_2(\mathbf{x}), \ldots, \pi_K(\mathbf{x})$. The joint state of all replicas is denoted by $\mathbf{X} = \left(\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}\right)$, where $\mathbf{x}^{(k)}$ is the current state of replica k. The replicas are coupled together through swaps between neighboring replicas k and $k+1$ proposing state exchange as $(\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)}) \to (\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)})$, with acceptance probability

$$A_{\text{swap}} = \min\left\{1, \frac{\pi_k\left(\mathbf{x}^{(k+1)}\right) \pi_{k+1}\left(\mathbf{x}^{(k)}\right)}{\pi_k\left(\mathbf{x}^{(k)}\right) \pi_{k+1}\left(\mathbf{x}^{(k+1)}\right)}\right\}. \tag{4}$$

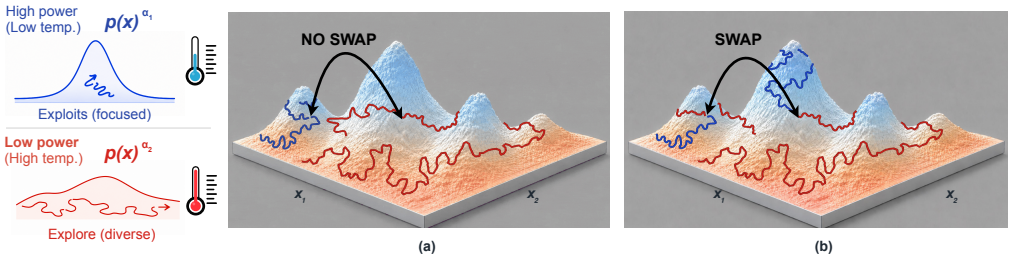

Figure 2: In our PPT, a flat, low-power chain $p(\mathbf{x})^{\alpha_2}$ (red) *explores freely* while a sharp, high-power chain $p(\mathbf{x})^{\alpha_1}$ (blue) exploits local modes to *reason sharply*; periodically the chains propose to swap states. In (a) the swap is rejected and both stay put. In (b), it is accepted, letting the focused chain teleport to a high-probability region the explorer found, rather than climbing over the barrier itself.

3 Methodology

3.1 Motivation: The Exploration Bottleneck

Recall that at horizon T, the sharpened power target has the form $\pi_\alpha(\mathbf{x}) \propto p_0(\mathbf{x})^\alpha$. The power α encodes an exploration–exploitation trade-off that no single value resolves: raising it concentrates mass on high-likelihood responses—desirable when high-likelihood traces are more coherent or reliable—but sharpens the landscape and impedes local exploration, while lowering it flattens the landscape and eases exploration at the price of leaving good answers diffuse.

To address this exploration–exploitation trade-off, we introduce **Parallel Power Tempering (PPT)**, which couples multiple replicas through swaps. Specializing the parallel-tempering construction from Section 2 to sequence-level power targets, we choose a ladder of sharpening powers $1 \le \alpha_1 < \cdots < \alpha_K$ and assign replica k the target $\pi_{\alpha_k}(\mathbf{x}) \propto p_0(\mathbf{x})^{\alpha_k}$ and couple the replicas through state swaps. Replicas with small α_k act as high-temperature explorers that range across alternative reasoning paths, while swaps let the promising trajectories they discover travel up the ladder to the target chain—the sharpest rung α_K—which thus reaches high-probability regions by exchange rather than by climbing over reasoning barriers itself. See Figure 2 for a visual example.

3.2 Structural Truncation Bias and the Fixed-Horizon Correction

To apply PPT to variable-length sequences, refinement must allow early termination to be revised. Naïvely extending the early-stopping implementation of Karan & Du (2026) to the parallel-tempering setting inherits a one-way truncation bias. Each suffix proposal regenerates only through the current realized length, so a shorter candidate $\mathbf{y}$ can satisfy $q(\mathbf{y} \mid \mathbf{x}) > 0$ while $q(\mathbf{x} \mid \mathbf{y}) = 0$. The exact MH rule should reject this move, but the implementation can accept it using token scores truncated to the candidate's endpoint. Therefore, the accepted shortenings create uncompensated probability flow toward shorter records, preventing the sampler from preserving the true target.

Under a uniform accepted-shortening condition, this one-way truncation also yields an asymptotic bias floor. More formally, let $\mathcal{B}_h$ denote the records of length at most h, and let $b_h^{(k)} = \pi_{\alpha_k}(\mathcal{B}_h^c)$ denote the probability mass of longer records at rung k. Assume the MH updates accept a move shortening a record from above h to at most h, with probability $\delta > 0$, but can never lengthen records. Then for any initial distribution ν, after N iterations, we show in App. C.2 that

$$\liminf_{N\to\infty} \mathrm{TV}(\nu_N, \Pi) \ge 1 - \prod_{k=1}^{K} (1 - b_h^{(k)}). \tag{5}$$

This inequality quantifies the asymptotic error due to the violation of the target preservation by the unbalanced MH rule. Crucially, this truncation would *hinder PPT's ability to explore*, as shortened records transferred to lower power rung would retain their restricted generation budget.

Remark 1. To remove this one-way truncation, we maintain a fixed horizon T, through deterministic post-EOS padding $\perp$, allowing suffix proposals starting at or before EOS to regenerate up to T regardless of the current completion length, and explore more effectively. We show in App. C that this fixed-horizon correction removes structural truncation bias, while local MH updates composed with replica swaps preserve the true sharpened target.

Error Floor: Example. To better understand the effect of the one-way truncation, consider one chain at a fixed horizon $T = 2$, with power $\alpha = 2$ and vocabulary $\{a, e\}$, where e is the terminal token. The model assigns probability $1/2$ to each token at every unfinished prefix. On the common record space $(X_1, X_2, X_3, X_4) = \big((e, \perp), (a), (a, e), (a, a)\big)$, and the powered target is $\pi = (\frac{2}{3}, 0, \frac{1}{6}, \frac{1}{6})$, assigning zero target mass on the unfinished one-token record X_2. Assume, both samplers start from $X_3 = (a, e)$ and perform N MH updates. Once unbalanced refinement accepts a one-token record, it cannot revisit either two-token continuation. Fixed-horizon refinement retains this possibility: from X_1, it returns to a two-token record with probability $1/8$ per update. Thus an early EOS remains revisable. For $N \geq 1$, the fixed-horizon and truncating laws ν_N and $\widetilde{\nu}_N$ satisfy

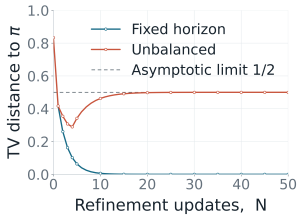

Figure 3: Comparison between fixed-horizon and unbalanced refinement starting from X_3.

$$\mathrm{TV}(\nu_N, \pi) = \frac{2}{3}\left(\frac{5}{8}\right)^N \xrightarrow[N\to\infty]{} 0, \qquad \mathrm{TV}(\widetilde{\nu}_N, \pi) \xrightarrow[N\to\infty]{} \frac{1}{2},$$

Although, unbalanced refinement initially reduces the error, as illustrated in Figure 3, exploration is eventually confined to the one-token records. More details are left for App. C.

3.3 Parallel Power Tempering

Following the progressive schedule of Karan & Du (2026), we choose a block width B and define

$$T_m = \min\{mB, T\}, \qquad m = 0, \ldots, M, \qquad M = \lceil T/B \rceil.$$

The fixed-horizon construction of Remark 1 is applied at every stage. More specifically, at stage m, all replicas share horizon T_m, with positions after EOS padded by $\perp$. Let us denote the record of replica k by $\mathbf{x}^{(k)} \in \mathcal{S}_{T_m}$, targeting $\pi_k(\mathbf{x}) \propto p_0(\mathbf{x})^{\alpha_k}$, with stage-$m$ joint target being expressed as

$$\Pi_m\left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K)}\right) := \prod_{k=1}^{K} \pi_k\left(\mathbf{x}^{(k)}\right). \tag{6}$$

Because Π_m factorizes, its kth marginal is π_k. Since $T_M = T$, the designated rung $k^\star$ has marginal $\pi_{k^\star} = \pi_{\alpha^\star}$, which is exactly the target at the horizon-T (Section 2). The auxiliary replicas provide potential transport paths without altering this output marginal; their mixing advantage is conditional on their local kernels being faster than the output-rung kernel.

(1) Block proposal. Starting from its stage-$(m-1)$ record, we extend each open replica by up to B tokens using the tokenwise-powered proposal g_{α_k} from Eq. 2:

$$x_t^{(k)} \sim g_{\alpha_k}\left(\cdot \mid \mathbf{x}_{<t}^{(k)}\right), \qquad t = T_{m-1} + 1, \ldots, T_m. \tag{7}$$

Generation stops at EOS, so only non-terminated records require model calls. Sampling from g_{α_k} corresponds to token-level decoding at temperature $1/\alpha_k$, which differs from the sequence-level target. Therefore, block extension initializes stage m but does not draw from target Π_m.

(2) Local refinement. At stage m and rung k, fix the current record $\mathbf{x} \in \mathcal{S}_{T_m}$. Each local update draws a resampling position $r \sim \mathrm{Unif}\{1, \ldots, T_m\}$, retains the prefix $\mathbf{x}_{<r}$, and regenerates the suffix, yielding $\mathbf{y} \sim q_r(\cdot \mid \mathbf{x})$, where

$$q_r(\mathbf{y} \mid \mathbf{x}) := \mathbf{1}\{\mathbf{y}_{<r} = \mathbf{x}_{<r}\} \prod_{t=r}^{T_m} g_{\alpha_k}(y_t \mid \mathbf{y}_{<t}). \tag{8}$$

The MH correction accepts the candidate with probability

$$A^{\text{refine}} = \min\left\{1, \frac{\pi_k(\mathbf{y})\, q_r(\mathbf{x} \mid \mathbf{y})}{\pi_k(\mathbf{x})\, q_r(\mathbf{y} \mid \mathbf{x})}\right\}. \tag{9}$$

For each fixed r, the resulting MH kernel preserves π_k; hence, so does the state-independent uniform mixture over r. Because r may fall anywhere in the record, refinement can revise decisions before the newest block. Forward and reverse probabilities are both evaluated through T_m.

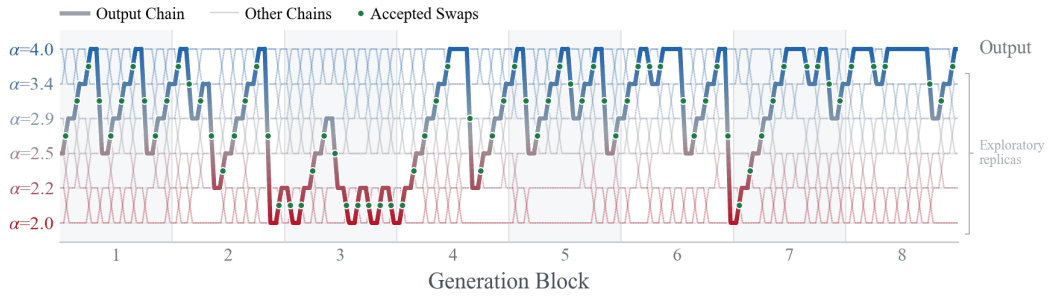

Figure 4: **Parallel tempering across the sharpening ladder.** Green dots mark accepted adjacent swaps; the colored gradient line traces a sampled trajectory across replicas and generation blocks.

(3) Swaps. At horizon T_m, to couple the chains along the ladder, we sweep through the adjacent pairs in order $k = 1, \dots, K-1$ and propose exchanging the records at rungs k and $k+1$,

$$(\mathbf{x}^{(k)}, \mathbf{x}^{(k+1)}) \mapsto (\mathbf{x}^{(k+1)}, \mathbf{x}^{(k)}),$$

committing each accepted swap before proceeding to the next pair. This sequential ordering allows a state to traverse multiple rungs within one sweep. Related ordered swap schedules are studied in classical parallel tempering (Deng et al., 2023).

In Figure 4, the sequence that is ultimately returned has been carried by accepted swaps through every rung of the ladder, from the most exploratory replica to the sharpest, before reaching the output rung. The MH acceptance probability is

$$A^{\text{swap}} = \min\left\{1, \frac{\pi_k(\mathbf{x}^{(k+1)})\pi_{k+1}(\mathbf{x}^{(k)})}{\pi_k(\mathbf{x}^{(k)})\pi_{k+1}(\mathbf{x}^{(k+1)})}\right\}. \tag{10}$$

Substituting $\pi_k \propto p_0^{\alpha_k}$ cancels the normalizing constants and gives

$$A^{\text{swap}} = \min\left\{1, \exp\left[(\alpha_{k+1} - \alpha_k)\left(\log p_0(\mathbf{x}^{(k)}) - \log p_0(\mathbf{x}^{(k+1)})\right)\right]\right\}. \tag{11}$$

Free Exploration. Swaps can transport promising trajectories from exploratory rungs to a sharper rung whose local updates can complete them, potentially *exponentially* accelerating discovery of better modes (Dong & Tong, 2022), rather than improving only linearly with additional independent samples. Notably, this exploration comes with minimal computational overhead as each adjacent exchange is an MH move evaluated from cached sequence likelihoods, so the exchange itself requires no additional model passes (Table 3). Algorithm 1 specifies the complete schedule, and Appendix B gives implementation details; after stage M, the record at rung $k^\star$ is returned.

3.4 Ladder Design

To effectively capitalize the benefit of Parallel Tempering, adjacent chains overlap sufficiently. Powers placed too far apart exchange rarely and cut the ladder into isolated pieces; powers placed too close spend replicas without sufficient exploration. For fixed endpoints α_1, α_K and K chains, an optimal ladder design is the *equi-accepting* ladder (Rathore et al., 2005; Deng et al., 2023). For expected acceptance $\mathcal{A}_k$ between rungs k and $k+1$, the intermediate powers are chosen such that

$$\mathcal{A}_1 = \mathcal{A}_2 = \cdots = \mathcal{A}_{K-1}. \tag{12}$$

A pair with substantially lower acceptance is a bottleneck that throttles transport along the whole ladder, while unusually high acceptance signals two powers that are needlessly close; equalizing acceptances maximizes the weakest link (Appendix D.2).

Remark 2. The acceptance rate $\mathcal{A}_k$ in Eq. 11 is governed by the gap $\alpha_{k+1} - \alpha_k$ and the spread of the sequence log-likelihood $\log p_0$. When this spread scales as $1/\alpha$, then $\mathcal{A}_k$ becomes a function of the ratio α_{k+1}/α_k, and the equi-accepting condition of Eq. 12 reduces to the geometric ladder

$$\alpha_k = \alpha_1 \left(\frac{\alpha_K}{\alpha_1}\right)^{\frac{k-1}{K-1}}, \qquad k = 1, \dots, K,$$

In practice, we initialize with a geometric ladder and tune the intermediate powers empirically until the measured swap acceptances are approximately equal. More details are left for App. D.

Table 1: Pass@1 accuracy (%) with one returned completion per item. **Bold** marks the best result, including ties, within each model group

Method	MATH500	GPQA	HumanEval	GSM8K	AIME 24&25	LCB v5
Qwen3-4B						
Standard	83.3	51.0	85.4	94.6	70.7	45.6
Lower Temperature	85.1	48.5	86.6	94.5	72.0	44.8
Power Sampling	83.5	51.0	90.2	94.2	73.3	55.1
PowerSMC	83.2	47.5	87.9	94.3	61.7	52.3
GRPO	85.6	50.0	91.6	94.6	73.3	52.7
PPT (Ours)	**86.0**	**53.6**	**93.9**	**94.8**	**78.3**	**57.9**
Qwen3-8B						
Standard	83.9	55.8	84.9	94.6	72.7	48.1
Lower Temperature	84.6	54.5	87.2	95.0	73.0	48.4
Power Sampling	82.2	57.6	90.7	95.2	73.3	53.5
PowerSMC	84.1	58.0	90.2	**95.5**	65.0	53.6
GRPO	86.4	53.0	93.3	94.6	74.0	54.1
PPT (Ours)	**88.0**	**60.1**	**94.9**	**95.5**	**78.3**	**58.4**

Schedule Each adjacent-pair swap is itself an MH move on the joint target, so any composition of swaps preserves Π_m at every stage m, and hence Π. Therefore, the communication schedule cannot change what we sample, only how fast records travel across the ladder. Choosing one is an efficiency question, and its answer depends on the regime. In our implementation, we employ an ordered sweep of adjacent exchanges, applying each accepted swap immediately, as presented in Section 3.3. This allows a state to traverse several rungs within a single sweep. An alternative schedule is an asymptotically-optimal swap schedule in classical parallel tempering: the deterministic even–odd (DEO) communication defined in (Okabe et al., 2001; Syed et al., 2022), which attempts a single parity matching of disjoint interfaces and alternates between the two matchings, so that the swaps within a round can run in parallel.

Remark 3. If acceptance probabilities are history-independent and equal to a constant $\bar{A}$, then the expected iteration counts for a tagged-replica round trip are

$$\mathbb{E}T_{\mathrm{ADJ}} = K\left(1 + (K-1)\frac{1-\bar{A}}{\bar{A}}\right), \qquad \mathbb{E}T_{\mathrm{DEO}} = 2K\left(1 + (K-1)\frac{1-\bar{A}}{\bar{A}}\right), \tag{13}$$

Thus, ordered ADJ requires half as many expected iterations as DEO, where an ADJ iteration comprises a complete adjacent sweep and a DEO iteration comprises one parity layer. This reduction comes at the cost of executing all $K-1$ swap attempts sequentially. In our regime, where we use few replicas and communication overhead is small relative to local refinement, this tradeoff favors ADJ. Conversely, DEO becomes faster for sufficiently large K, since ADJ's sequential communication overhead grows quadratically with the number of replicas, compared to the DEO's linear scaling.

4 Experiments

4.1 Main Results

We compare PPT with standard and low-temperature decoding, Power Sampling (Karan & Du, 2026), PowerSMC (Azizi et al., 2026), and GRPO (Shao et al., 2024), which is a training-based reference. Table 1 reports results for Qwen3-4B and Qwen3-8B on MATH500 (Lightman et al., 2024), GPQA (Rein et al., 2024), HumanEval (Chen et al., 2021), GSM8K (Cobbe et al., 2021), AIME 24&25 (Zhang & Math-AI, 2024; Zhang & Team, 2025), and LiveCodeBench (LCB) v5 (Jain et al., 2025), with a common benchmark-specific completion cap shared by all methods. Table 2 extends the comparison to the more recent and capable Qwen3.5-9B on the three most challenging benchmarks: GPQA, AIME 24&25, and LCB v5.

Across both tables, PPT attains the best or tied-best accuracy in all 15 model–benchmark settings, spanning mathematics, science, and code; its only tie is on the saturated GSM8K. It is demonstrated that single-chain sharpening is unreliable on strong models, as Power Sampling falls below standard decoding in most Qwen3.5-9B benchmarks, PowerSMC trails standard decoding by a wide margin on AIME 24&25, and lower-temperature decoding yields only modest gains. Conversely, PPT targets the same powered distribution and outperforms every sampling-based baseline, never regresses

Table 2: Pass@1 Accuracy using Qwen3.5-9B on GPQA, AIME 24&25 and LCB v5.

Qwen3.5-9B	GPQA	AIME 24&25	LCB
Standard	82.4	82.3	81.9
Power Sampling	77.6	78.7	82.0
Lower Temperature	81.8	84.3	82.4
GPT-5†	85.4	95.0	84.6
Opus 4.5†	86.6	93.3	87.1
GLM 4.7†	85.9	95.0	89.4
PPT(Ours)	**85.9**	**93.3**	**84.9**

† Reported values.

Table 3: Per-example runtime and peak memory for our implementation.

Method	Peak KV cache memory (GiB)	Local-update time (sec)	Swap overhead (sec)
Qwen3-8B			
Single chain	0.86	36.1	–
PPT ($K = 4$)	3.21	37.2	0.037
Qwen3.5-9B			
Single chain	3.06	43.8	–
PPT ($K = 3$)	7.56	75.4	0.08

below standard decoding, and even achieves higher accuracy than GRPO without additional parameter updates or rewards. Notably, using Qwen3.5-9B, we observed performance that matches and even exceeds frontier models. Additional results are in Appendix H.

Runtime and Memory Cost Furthermore, we provide a breakdown of the computational cost of PPT in the LCB v5 dataset. Table 3 reports per problem: peak-memory usage, local-update time, and replica-exchange overhead for the single chain ($K = 1$) and PPT ($K = 4$ for Qwen3-8B, $K = 3$ for Qwen3.5-9B). The efficiency of our implementation is highlighted as batching local updates across replicas accelerates wall-clock inference-time. As shown in Table 3, the multi-replica runs require only $1.03\times$ and $1.72\times$ the single-chain local-update wall-clock time for Qwen3-8B and Qwen3.5-9B, respectively, while replica exchange accounts for only 0.1% of the total time in both cases. However, this efficiency comes at the expense of higher peak memory usage: approximately $3.73\times$ and $2.47\times$ higher than the single-chain values.

4.2 Compute-matched Controls

Additional single-chain refinement. To test whether additional refinement of a single chain can recover the gains of PPT, we increase the number of MCMC steps in single chain settings ($K = 1$) to match the total compute in terms of decode tokens generated by our multi-chain PPT on Qwen3-8B. Table 4 shows that PPT remains consistently more accurate across all four benchmarks. The largest gaps occur on the hardest tasks, namely LCB v5 and AIME 24&25, where PPT reaches 58.4% and 78.3% compared with 54.9% and 75.3% for the extended single chain, respectively. Thus, additional single-chain refinement does not close the performance gap at the matched compute budget.

Table 4: Compute-matched Qwen3-8B accuracy (%) between our multi-chain PPT ($K > 1$) and single chain ($K = 1$) samplers.

Benchmark	PPT	Single-chain
MATH500	**88.0**	85.6
GPQA	**60.1**	59.1
AIME 24&25	**78.3**	75.3
LCB v5	**58.4**	54.9

K uncoupled chains. To isolate the contribution of parallel tempering, Table 5 compares PPT with an uncoupled ladder on both Qwen3 models. The uncoupled ladder runs the same number of chains, each refined independently over a fixed horizon with no swaps. We report two metrics: i) *Pass@1* scores a single trace; the coldest rung for PPT, and the rung with the highest accumulated log-likelihood for the uncoupled ladder, and ii) *Vote* takes the majority answer across all traces. Table 5 shows that enabling swaps consistently improves both the Pass@1 and voting accuracy in every model–benchmark setting. Overall, we can see that significant gains are attributed to the exchange mechanism, not merely running and aggregating more replicas.

4.3 Ablation Studies

Number of replicas. We sweep $K = 1$–5 with fixed sharpening range, block size, and MCMC steps, using our fixed-horizon kernel at $K = 1$ (Figure 5, left). Compute is measured in wall-clock time in seconds per example. Parallel batching lowers the cost per replica, so running K chains scales more favorably than K-fold times a single-chain run. For example, with $K = 4$, the total cost is only $2.1\times$ and $1.2\times$ that of a single chain on MATH500 and HumanEval, respectively. These results further demonstrate the favorable runtime scaling of PPT as the number of chains increases. Beyond five replicas, per-replica accuracy plateaus, with diminishing returns for further increasing.

Table 5: Compute-matched accuracy (%) with and without swaps using the same number of chains. Pass@1 scores a single selected trace; Vote takes the majority answer across traces. **Bold** marks the better result.

Method	MATH 500		AIME 24&25		GPQA	
	Pass@1	Vote	Pass@1	Vote	Pass@1	Vote
Qwen3-4B						
Uncoupled ladder	84.2	87.6	75.7	78.3	52.3	48.5
PPT (Ours)	**86.0**	**88.8**	**78.3**	**80.0**	**53.6**	**55.1**
Qwen3-8B						
Uncoupled ladder	86.0	87.4	76.7	80.0	57.6	58.9
PPT (Ours)	**88.0**	**90.0**	**78.3**	**81.7**	**60.1**	**61.6**

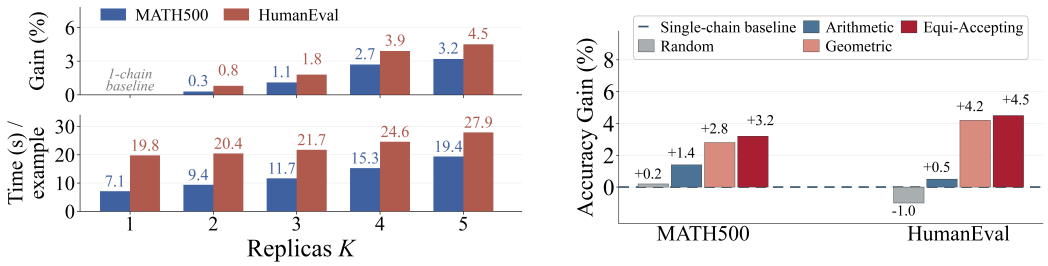

Figure 5: **Replica count and ladder design.** *Left:* (Top): accuracy gain as the number of replicas increases, and (Bottom): the corresponding compute in wall-clock time per example; the favorable scaling reflects the batching of replicas in parallel. *Right:* effect of ladder design on accuracy across two datasets, MATH500 and HumanEval, with $K = 5$ replicas.

We found that accuracy saturates around four to five replicas: $K = 4$, gains 2.7 and 3.9 percentage points over $K = 1$, while $K = 5$ peaks at $+3.2$ and $+4.5$.

Ladder design. We examine the effect of appropriate ladder design in the efficacy of the model. We compare: i) randomly picked ladder temperatures, ii) arithmetic sequences , iii) geometric sequences, and iv) equi-accepting sharpening ladders with $K = 5$ and fixed block size and decoding budget (Figure 5, right). Ladder spacing controls how readily states move between exploratory and strongly sharpened replicas. Random and arithmetic spacing offer limited gains, with random spacing even harming HumanEval. Geometric spacing, which uses equal power ratios (Remark 2), improves both benchmarks, while equi-accepting performs best. By approximately equalizing adjacent swap-acceptance rates, the latter aims to prevent any single pair from becoming a communication bottleneck. These results suggest that balancing communication across the ladder helps replicas share useful exploration more effectively.

5 Conclusion

We introduced PPT, a parallel tempering sampler for power-sharpened LLM distributions that addresses the exploration–exploitation trade-off of single-chain power sampling. Low-power replicas explore diverse reasoning trajectories, high-power replicas refine them, and swaps couple the ladder power moving promising traces toward the sharpest rung at negligible cost. In our analysis, we identified a structural truncation bias in early-stopped power samplers and removed it with a fixed-horizon construction that restores preservation of the true sharpened target, and provides all chains the same exploration bandwidth. Empirically, PPT is best or tied-best in all 15 model–benchmark settings and outperforms GRPO without parameter updates or rewards, with gains that come from the swaps rather than extra compute or chains. With Qwen3.5-9B, it performs comparably to frontier models, suggesting that small frozen models can reason without additional parameter updates.

AI USE STATEMENT

We used generative AI tools to assist with manuscript editing, mathematical-notation consistency. For the theoretical analysis, AI tools assisted in checking arguments. The authors reviewed the AI-assisted material and take full responsibility for the correctness of the theoretical arguments and the accuracy, integrity, and content of the final manuscript.

ETHICS STATEMENT

All authors have read and commit to adhering to the ICLR Code of Ethics. This work uses only publicly available datasets and open-source models, and does not involve human subjects or human subjects data.

REPRODUCIBILITY STATEMENT

Appendix B describes the implementation of the proposed PPT module. Appendix C provides the detailed theoretical justification. Appendix G gives the experimental settings, for Section 4, and Appendix H collects additional experimental results.

REFERENCES

Emmanuel Anaya Gonzalez, Sairam Vaidya, Kanghee Park, Ruyi Ji, Taylor Berg-Kirkpatrick, and Loris D'Antoni. Constrained sampling for language models should be easy: An mcmc perspective. *Advances in Neural Information Processing Systems*, 38:59951–59981, 2026.

Seyedarmin Azizi, Erfan Baghaei Potraghloo, Minoo Ahmadi, Souvik Kundu, and Massoud Pedram. Power-SMC: Low-Latency Sequence-Level Power Sampling for Training-Free LLM Reasoning. *CoRR*, abs/2602.10273, 2026.

M. Chen, J. Tworek, H. Jun, Q. Yuan, H. P. de Oliveira Pinto, J. Kaplan, H. Edwards, Y. Burda, N. Joseph, G. Brockman, A. Ray, R. Puri, G. Krueger, M. Petrov, H. Khlaaf, G. Sastry, P. Mishkin, B. Chan, S. Gray, N. Ryder, M. Pavlov, A. Power, L. Kaiser, M. Bavarian, C. Winter, P. Tillet, F. P. Such, D. Cummings, M. Plappert, F. Chantzis, E. Barnes, A. Herbert-Voss, W. H. Guss, A. Nichol, A. Paino, N. Tezak, J. Tang, I. Babuschkin, S. Balaji, S. Jain, W. Saunders, C. Hesse, A. N. Carr, J. Leike, J. Achiam, V. Misra, E. Morikawa, A. Radford, M. Knight, M. Brundage, M. Murati, K. Mayer, P. Welinder, B. McGrew, D. Amodei, S. McCandlish, I. Sutskever, and W. Zaremba. Evaluating Large Language Models Trained on Code. *CoRR*, abs/2107.03374, 2021.

Yilei Chen, Souradip Chakraborty, Lorenz Wolf, Yannis Paschalidis, and Aldo Pacchiano. Post-training Large Language Models for Diverse High-Quality Responses. *arXiv preprint arXiv:2509.04784*, 2025.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, Christopher Hesse, and John Schulman. Training Verifiers to Solve Math Word Problems. *CoRR*, abs/2110.14168, 2021.

Wei Deng, Qi Feng, Liyao Gao, Faming Liang, and Guang Lin. Non-convex Learning via Replica Exchange Stochastic Gradient MCMC. In *Proceedings of the 37th International Conference on Machine Learning*, volume 119, pp. 2474–2483, 2020.

Wei Deng, Qian Zhang, Qi Feng, Faming Liang, and Guang Lin. Non-reversible Parallel Tempering for Deep Posterior Approximation. In *Thirty-Seventh AAAI Conference on Artificial Intelligence*, pp. 7332–7339, 2023.

Jing Dong and Xin T Tong. Spectral Gap of Replica Exchange Langevin Diffusion on Mixture Distributions. *Stochastic Processes and their Applications*, 151:451–489, 2022.

Weihua Du, Yiming Yang, and Sean Welleck. Optimizing Temperature for Language Models with Multi-Sample Inference. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 14648–14668, 2025.

David J Earl and Michael W Deem. Parallel Tempering: Theory, Applications, and New Perspectives. *Physical Chemistry Chemical Physics*, 7(23):3910–3916, 2005.

Gonçalo R Faria, Sweta Agrawal, António Farinhas, Ricardo Rei, José G de Souza, and André F Martins. QUEST: Quality-aware Metropolis-Hastings Sampling for Machine Translation. *Advances in Neural Information Processing Systems*, 37:89042–89068, 2024.

Shengyu Feng, Xiang Kong, Shuang Ma, Aonan Zhang, Dong Yin, Chong Wang, Ruoming Pang, and Yiming Yang. Step-by-Step Reasoning for Math Problems via Twisted Sequential Monte Carlo. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jarad Forristal, Fatemehsadat Mireshghallah, Greg Durrett, and Taylor Berg-Kirkpatrick. A block metropolis-hastings sampler for controllable energy-based text generation. In *Proceedings of the 27th Conference on Computational Natural Language Learning (CoNLL)*, pp. 403–413, 2023.

Charles J. Geyer. Markov Chain Monte Carlo Maximum Likelihood. In *Computing Science and Statistics: Proceedings of the 23rd Symposium on the Interface*, pp. 156–163, 1991.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Ruoyu Zhang, Runxin Xu, Qihao Zhu, Shirong Ma, Peiyi Wang, Xiao Bi, et al. Deepseek-R1: Incentivizing Reasoning Capability in LLMs via Reinforcement Learning. *arXiv preprint arXiv:2501.12948*, 2025.

W Keith Hastings. Monte Carlo Sampling Methods using Markov Chains and Their Applications. *Biometrika*, 57(1):97–109, 1970.

Andre Wang He, Daniel Fried, and Sean Welleck. Rewarding the Unlikely: Lifting GRPO Beyond Distribution Sharpening. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 25548–25560, 2025.

Jiajun He, Zongyu Guo, José Miguel Hernández-Lobato, and Yuanqi Du. Recipes for Steering and Scaling LLMs via Sampling. *arXiv preprint arXiv:2608.26120*, 2026.

Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. Measuring Mathematical Problem Solving with the MATH Dataset. *arXiv preprint arXiv:2103.03874*, 2021.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The Curious Case of Neural Text Degeneration. In *8th International Conference on Learning Representations*, 2020.

Chuxuan Hu, Yuxuan Zhu, Antony Kellermann, Caleb Biddulph, Suppakit Waiwitlikhit, Jason Benn, and Daniel Kang. Breaking Barriers: Do Reinforcement Post Training Gains Transfer To Unseen Domains? In *The Fourteenth International Conference on Learning Representations*, 2026.

Jingcheng Hu, Yinmin Zhang, Qi Han, Daxin Jiang, Xiangyu Zhang, and Heung-Yeung Shum. Open-Reasoner-Zero: An Open Source Approach to Scaling Up Reinforcement Learning on the Base Model. In *Advances in Neural Information Processing Systems*, volume 38, pp. 162239–162262, 2025.

Audrey Huang, Adam Block, Qinghua Liu, Nan Jiang, Akshay Krishnamurthy, and Dylan J. Foster. Is Best-of-N the Best of Them? Coverage, Scaling, and Optimality in Inference-Time Alignment. In *Proceedings of the 42nd International Conference on Machine Learning*, volume 267, pp. 25075–25126, 2025.

Koji Hukushima and Koji Nemoto. Exchange Monte Carlo Method and Application to Spin Glass Simulations. *Journal of the Physical Society of Japan*, 65(6):1604–1608, 1996.

Naman Jain, King Han, Alex Gu, Wen-Ding Li, Fanjia Yan, Tianjun Zhang, Sida Wang, Armando Solar-Lezama, Koushik Sen, and Ion Stoica. LiveCodeBench: Holistic and Contamination Free Evaluation of Large Language Models for Code. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Zhewei Kang, Xuandong Zhao, and Dawn Song. Scalable best-of-n selection for large language models via self-certainty. In D. Belgrave, C. Zhang, H. Lin, R. Pascanu, P. Koniusz, M. Ghassemi, and N. Chen (eds.), *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 19720–19745. Curran Associates, Inc., 2025. doi: 10.52202/085713-0663. URL https://proceedings.neurips.cc/paper_files/paper/2025/file/1c7eff166a8e345f664f0faa8f4e4d2e-Paper-Conference.pdf.

Aayush Karan and Yilun Du. Reasoning with Sampling: Your Base Model is Smarter Than You Think. In *The Fourteenth International Conference on Learning Representations*, 2026.

David A Kofke. On the acceptance probability of replica-exchange monte carlo trials. *The Journal of chemical physics*, 117(15):6911–6914, 2002.

Aminata Kone and David A. Kofke. Selection of Temperature Intervals for Parallel-Tempering Simulations. *The Journal of Chemical Physics*, 122(20):206101, 2005.

Nathan Lambert, Jacob Morrison, Valentina Pyatkin, Shengyi Huang, Hamish Ivison, Faeze Brahman, Lester James V Miranda, Alisa Liu, Nouha Dziri, Shane Lyu, et al. Tülu 3: Pushing frontiers in open language model post-training. *arXiv preprint arXiv:2411.15124*, 2024.

Marvin Li, Aayush Karan, and Sitan Chen. Blink of an Eye: a Simple Theory for Feature Localization in Generative Models. In *Forty-second International Conference on Machine Learning*, volume 267, pp. 35047–35080, 2025.

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

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

Clara Meister, Tiago Pimentel, Gian Wiher, and Ryan Cotterell. Locally Typical Sampling. *Transactions of the Association for Computational Linguistics*, 11:102–121, 2023.

Nicholas Metropolis, Arianna W. Rosenbluth, Marshall N. Rosenbluth, Augusta H. Teller, and Edward Teller. Equation of State Calculations by Fast Computing Machines. *Journal of Chemical Physics*, 21(6):1087–1092, 1953.

Fatemehsadat Mireshghallah, Kartik Goyal, and Taylor Berg-Kirkpatrick. Mix and Match: Learning-free Controllable Text Generation using Energy Language Models. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 401–415, 2022.

Minh Nhat Nguyen, Andrew Baker, Clement Neo, Allen Roush, Andreas Kirsch, and Ravid Shwartz-Ziv. Turning Up the Heat: Min-p Sampling for Creative and Coherent LLM Outputs. In *The Thirteenth International Conference on Learning Representations*, 2025.

Tsuneyasu Okabe, Masaaki Kawata, Yuko Okamoto, and Masuhiro Mikami. Replica-exchange monte carlo method for the isobaric–isothermal ensemble. *Chemical physics letters*, 335(5-6): 435–439, 2001.

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, John Schulman, Jacob Hilton, Fraser Kelton, Luke Miller, Maddie Simens, Amanda Askell, Peter Welinder, Paul F Christiano, Jan Leike, and Ryan Lowe. Training Language Models to Follow Instructions with Human Feedback. In *Advances in Neural Information Processing Systems*, volume 35, pp. 27730–27744, 2022.

Nitin Rathore, Manan Chopra, and Juan J de Pablo. Optimal Allocation of Replicas in Parallel Tempering Simulations. *The Journal of chemical physics*, 122(2):024111, 2005.

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

Zhihong Shao, Peiyi Wang, Qihao Zhu, Runxin Xu, Junxiao Song, Xiao Bi, Haowei Zhang, Mingchuan Zhang, Y. K. Li, Y. Wu, and Daya Guo. DeepSeekMath: Pushing the Limits of Mathematical Reasoning in Open Language Models. *arXiv preprint arXiv:2402.03300*, 2024.

Yuji Sugita and Yuko Okamoto. Replica-Exchange Molecular Dynamics Method for Protein Folding. *Chemical Physics Letters*, 314(1–2):141–151, 1999.

Robert H. Swendsen and Jian-Sheng Wang. Replica Monte Carlo Simulation of Spin-Glasses. *Physical Review Letters*, 57(21):2607–2609, 1986.

Saifuddin Syed, Vittorio Romaniello, Trevor Campbell, and Alexandre Bouchard-Côté. Parallel tempering on optimized paths. In *International Conference on Machine Learning*, pp. 10033–10042. PMLR, 2021.

Saifuddin Syed, Alexandre Bouchard-Côté, George Deligiannidis, and Arnaud Doucet. Non-reversible Parallel Tempering: A Scalable Highly Parallel MCMC scheme. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)*, 84(2):321–350, 2022.

Chenxia Tang, Jianchun Liu, Hongli Xu, and Liusheng Huang. Top-$n\sigma$: Eliminating Noise in Logit Space for Robust Token Sampling of LLM. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 10758–10774, 2025.

Akiyoshi Tomihari and Issei Sato. Power Distribution Bridges Sampling, Self-Reward RL, and Self-Distillation. *arXiv preprint arXiv:2605.04542*, 2026.

Sergey Troshin, Wafaa Mohammed, Yan Meng, Christof Monz, Antske Fokkens, and Vlad Niculae. Control the Temperature: Selective Sampling for Diverse and High-Quality LLM Outputs. In *Second Conference on Language Modeling*, 2025.

Tianchun Wang, Zichuan Liu, Yuanzhou Chen, Jonathan Light, Weiyang Liu, Haifeng Chen, Xiang Zhang, and Wei Cheng. On the Effect of Sampling Diversity in Scaling LLM Inference. *arXiv preprint arXiv:2502.11027*, 2025.

Yang Yue, Zhiqi Chen, Rui Lu, Andrew Zhao, Zhaokai Wang, Yang Yue, Shiji Song, and Gao Huang. Does Reinforcement Learning Really Incentivize Reasoning Capacity in LLMs Beyond the Base Model? In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pp. 57654–57689, 2025.

Yifan Zhang and Team Math-AI. American Invitational Mathematics Examination (AIME) 2024, 2024.

Yifan Zhang and Math-AI Team. American Invitational Mathematics Examination (AIME) 2025, 2025. URL https://huggingface.co/datasets/math-ai/aime25.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Baker Grosse. Probabilistic Inference in Language Models via Twisted Sequential Monte Carlo. In *Proceedings of the 41st International Conference on Machine Learning*, volume 235, pp. 60704–60748, 2024.

CONTENTS

1 Introduction **1**

2 Preliminaries **3**
2.1 Notations . . . 3
2.2 Power Sharpening for LLMs . . . 3
2.3 Parallel Tempering . . . 3

3 Methodology **4**
3.1 Motivation: The Exploration Bottleneck . . . 4
3.2 Structural Truncation Bias and the Fixed-Horizon Correction . . . 4
3.3 Parallel Power Tempering . . . 5
3.4 Ladder Design . . . 6

4 Experiments **7**
4.1 Main Results . . . 7
4.2 Compute-matched Controls . . . 8
4.3 Ablation Studies . . . 8

5 Conclusion **9**

A Notation and Preliminaries **15**
A.1 Prompt-conditioned autoregressive model . . . 15
A.2 Sequence-level power target . . . 16
A.3 Sequence-level powering is not tokenwise temperature decoding . . . 16

B Implementation Details **16**
B.1 Step 1: Block extension . . . 17
B.2 Step 2: Local refinement . . . 17
B.3 Step 3: Swap exchange . . . 18
B.4 Putting the implementation together . . . 18

C Theoretical Properties: Bias and Exactness of PPT **19**
C.1 Fixed-horizon targets and PPT kernels . . . 19
C.2 Structural bias from one-way truncation in PPT . . . 21
C.3 Target preservation and convergence of fixed-horizon PPT . . . 23
C.4 A finite illustration of target preservation and one-way truncation bias . . . 25

D Power-Ladder Design **26**
D.1 Thermodynamic coordinate and exact swap overlap . . . 26
D.2 Max–min design and the unique equi-accepting ladder . . . 27

E Chain Communication Protocol **28**

F Extended Related Work **29**

G Experimental Settings **30**
G.1 Choice of Baselines . . . 30
G.2 Choice of Datasets . . . 31
G.3 Evaluation Metrics . . . 31
G.4 Configuration . . . 32

H Extended Experimental Results **33**
H.1 Extended Main Results . . . 33
H.2 Ablation Studies . . . 34

A NOTATION AND PRELIMINARIES

This section establishes the fixed-horizon representation used throughout the paper. It introduces prompt-conditioned completion records, the sequence-level power target, the tokenwise-powered proposal distribution, and the sequence of stage horizons used by the progressive sampler.

A.1 PROMPT-CONDITIONED AUTOREGRESSIVE MODEL

Fix a tokenized prompt $\mathbf{x}_0$, a finite language-model vocabulary $\mathcal{V}_{\mathrm{LM}}$, a set of configured terminal tokens $\mathcal{E} \subseteq \mathcal{V}_{\mathrm{LM}}$, and a maximum completion horizon $T \geq 1$. Let $\perp \notin \mathcal{V}_{\mathrm{LM}}$ be an auxiliary post-terminal symbol used to express the fixed-horizon state space, and define the augmented vocabulary

$$\mathcal{V} := \mathcal{V}_{\mathrm{LM}} \cup \{\perp\}.$$

The symbol $\perp$ serves exclusively as bookkeeping for the configured post-terminal convention. A *completion record* stores the generated completion tokens, while the prompt $\mathbf{x}_0$ enters every model distribution as fixed conditioning context. For every supported pre-terminal history h, the language model defines normalized next-token probabilities $p_{\mathrm{LM}}(v \mid \mathbf{x}_0, h)$, $v \in \mathcal{V}_{\mathrm{LM}}$. We define a completion history *open* through the position preceding its first configured terminal token and *terminated* from that token onward. The configured post-terminal convention extends the model distribution to $\mathcal{V}$ through

$$p_0(v \mid \mathbf{x}_0, h) := \begin{cases} p_{\mathrm{LM}}(v \mid \mathbf{x}_0, h), & h \text{ is supported and open and } v \in \mathcal{V}_{\mathrm{LM}}, \\ 1, & h \text{ is supported and terminated and } v = \perp, \\ 0, & \text{in all remaining cases.} \end{cases} \tag{14}$$

In Eq. 14, invalid constructions, such as a $\perp$ before the first terminal token or a non-$\perp$ token after termination, are assigned zero probability. Generation ends at the first configured terminal token. The post-terminal convention then fills the remaining horizon with $\perp$, each padding position contributing a factor of one.

A fixed-horizon record $\mathbf{x} = x_{1:T} \in \mathcal{V}^T$ then has prompt-conditioned probability

$$p_0(\mathbf{x} \mid \mathbf{x}_0) := \prod_{t=1}^{T} p_0\big(x_t \mid \mathbf{x}_0, \mathbf{x}_{<t}\big), \tag{15}$$

where each factor is given by Eq. 14, so any record containing an invalid construction has zero probability. The positive-support state space at horizon T is denoted

$$\mathcal{S}_T := \left\{\mathbf{x} \in \mathcal{V}^T : p_0(\mathbf{x} \mid \mathbf{x}_0) > 0\right\}. \tag{16}$$

For a sequence $\mathbf{x}$, we define its first terminal position by

$$\tau(\mathbf{x}) := \min\{t \in \{1, \ldots, T\} : x_t \in \mathcal{E}\},$$

with its completion length given by $\mathrm{len}(\mathbf{x}) := \min\{\tau(\mathbf{x}), T\}$. For $\mathbf{x} \in \mathcal{S}_T$, when $\tau(\mathbf{x}) \leq T$, the fixed-horizon state satisfies $\mathbf{x}_{\tau(\mathbf{x})+1:T} = (\perp, \ldots, \perp)$, and its probability simplifies to

$$p_0(\mathbf{x}_{1:T} \mid \mathbf{x}_0) = \prod_{t=1}^{\tau(\mathbf{x})} p_{\mathrm{LM}}(x_t \mid \mathbf{x}_0, \mathbf{x}_{<t}). \tag{17}$$

For $\mathbf{x} \in \mathcal{S}_T$, when no terminal token is emitted, the language model generates all T positions and the completion is right-censored at the declared horizon. The implementation supports both explicit deterministic tails and equivalent implicit-tail representations. In the latter representation, the stored completion record together with the configured post-terminal convention uniquely specifies the corresponding element of $\mathcal{S}_T$. During progressive construction, an open completion record of length below T serves as an intermediate record, and generation through the remaining positions produces its fixed-horizon representation.

A.2 SEQUENCE-LEVEL POWER TARGET

For a sharpening power $\alpha \geq 1$, we define the sequence-level power target on $\mathcal{S}_T$ by

$$\pi_{\alpha,T}(\mathbf{x} \mid \mathbf{x}_0) := \frac{p_0(\mathbf{x} \mid \mathbf{x}_0)^\alpha}{Z_{\alpha,T}(\mathbf{x}_0)}, \qquad Z_{\alpha,T}(\mathbf{x}_0) := \sum_{\mathbf{z} \in \mathcal{S}_T} p_0(\mathbf{z} \mid \mathbf{x}_0)^\alpha. \tag{18}$$

The finite state space and its positive support give $0 < Z_{\alpha,T}(\mathbf{x}_0) < \infty$, so Equation 18 defines a probability distribution on the fixed-horizon completion space. Following the above, we define the sequence energy as $U_T(\mathbf{x}) := -\log p_0(\mathbf{x} \mid \mathbf{x}_0)$. The target then takes the Gibbs form

$$\pi_{\alpha,T}(\mathbf{x} \mid \mathbf{x}_0) = Z_{\alpha,T}(\mathbf{x}_0)^{-1} \exp\{-\alpha U_T(\mathbf{x})\}.$$

Now consider a power ladder with $1 \leq \alpha_1 < \alpha_2 < \cdots < \alpha_K$, associating a replica with each power level. The joint replica state is given by $\mathbf{X} = \left(\mathbf{x}^{(1)}, \ldots, \mathbf{x}^{(K)}\right) \in \mathcal{S}_T^K$, with its joint product target being written as

$$\Pi_T(\mathbf{X} \mid \mathbf{x}_0) := \prod_{k=1}^{K} \pi_{\alpha_k,T}\left(\mathbf{x}^{(k)} \mid \mathbf{x}_0\right). \tag{19}$$

Each power remains associated with its rung index. A predesignated output rung $k^\star$ therefore has target marginal $\pi_{\alpha_{k^\star},T}$. The choice of $k^\star$ is fixed independently of the realized completion records. The horizon T forms part of the target specification, and every target statement in this appendix refers to this declared finite horizon. For $K = 1$, the product target reduces to a single sequence-level target and the adjacent-exchange operation becomes the identity.

A.3 SEQUENCE-LEVEL POWERING IS NOT TOKENWISE TEMPERATURE DECODING

Sequence-level powering generally differs from tokenwise temperature decoding because its next-token conditional depends on possible continuations; see Tomihari & Sato (2026, Section 4.1) for the analysis and derivation. For a supported prefix $h_t := (\mathbf{x}_0, \mathbf{x}_{<t})$, define the tokenwise powered normalizer and proposal used by our sampler:

$$z_{\alpha,t}(h_t) := \sum_{v \in \mathcal{V}} p_0(v \mid h_t)^\alpha, \tag{20}$$

$$g_{\alpha,t}(v \mid h_t) := \frac{p_0(v \mid h_t)^\alpha}{z_{\alpha,t}(h_t)}. \tag{21}$$

Conditioning on h_t abbreviates conditioning on $(\mathbf{x}_0, \mathbf{x}_{<t})$ throughout. After a terminal token, $g_{\alpha,t}$ places unit mass on $\perp$, following the configured post-terminal convention. In our notation, the powered continuation mass is

$$H_{\alpha,t}(v \mid h_t) := \begin{cases} \displaystyle\sum_{\mathbf{u}_{t+1:T} \in \mathcal{V}^{T-t}} p_0(\mathbf{u}_{t+1:T} \mid h_t, v)^\alpha, & p_0(v \mid h_t) > 0, \\ 0, & p_0(v \mid h_t) = 0. \end{cases} \tag{22}$$

At $t = T$, the continuation is empty and carries mass one, so $H_{\alpha,T}(v \mid h_T) = 1$ whenever $p_0(v \mid h_T) > 0$. Therefore, the sequence-level target induces the next-token conditional

$$\pi_{\alpha,T}(v \mid h_t) = \frac{p_0(v \mid h_t)^\alpha H_{\alpha,t}(v \mid h_t)}{\displaystyle\sum_{u \in \mathcal{V}} p_0(u \mid h_t)^\alpha H_{\alpha,t}(u \mid h_t)}. \tag{23}$$

The tractable proposal $g_{\alpha,t}$ omits the continuation factor $H_{\alpha,t}$ and equals this conditional only when that factor is constant over supported next tokens.

B IMPLEMENTATION DETAILS

A naive implementation runs the K replicas sequentially, giving K-fold wall-clock time increase for K replicas. Instead, we execute block extensions and suffix-resampling proposals for all replicas

jointly on a paged-attention engine (vLLM), while preserving each replica's proposal temperature $\tau_k = 1/\alpha_k$. Continuous batching handles the variable-length suffix proposals with little padding, and KV-cache reuse allows proposed suffixes to be generated and scored using cached prefix states. In other words, each replica stores its current sequence, cached base-model log-probability, and KV-cache handle, so an accepted swap merely exchanges record and cache without additional model evaluation. Total generated tokens still grow with K, but per-refinement-round wall-clock scales more favorably, until GPU saturation, after which scaling approaches the K-fold limit. This optimization comes solely from parallel execution and leaves the local MH and swap kernels unchanged.

The implementation combines three operations at each progressive stage: block extension, local refinement, and parallel tempering swaps. The first operation carries every completion record to the current stage horizon. The second operation refreshes a suffix of each record through a Metropolis–Hastings transition. Lastly, the third operation swaps whole records between adjacent powers.

B.1 Step 1: Block extension

Our implementation follows the blockwise construction of (Karan & Du, 2026). We define $B \geq 1$ to be the block width and further write $M := \lceil \frac{T}{B} \rceil, \quad T_m := \min\{mB, T\}, \quad m = 0, \ldots, M$, with $T_0 = 0$ and $T_M = T$. All fixed-horizon definitions apply at T_m by replacing T with T_m. In particular, we define the block-wise state space $\mathcal{S}_{T_m}$, with the terminal position and length on that space. For stage m and rung k, abbreviate

$$\pi_{m,k}(\cdot) := \pi_{\alpha_k, T_m}(\cdot \mid \mathbf{x}_0), \qquad \Pi_m(\mathbf{X}) := \Pi_{T_m}(\mathbf{X} \mid \mathbf{x}_0) = \prod_{k=1}^{K} \pi_{m,k}\big(\mathbf{x}^{(k)}\big). \tag{24}$$

Thus, we can infer that the terminal-stage target is $\Pi_M = \Pi_T$. For the rest of this section, the fixed conditioning on $\mathbf{x}_0$ is suppressed in stage notation.

Block extension from T_{m-1} to T_m initializes the state on the enlarged space. Sampling new tokens from $g_{\alpha_k,t}$ are generally not distributed from the target law of $\pi_{m,k}$. Therefore, we can interpret block extension as a warm start for the subsequent Metropolis–Hastings refinement. A finite number of refinement steps produces an approximation to Π_m. The exact invariance and convergence results below concern the local and parallel tempering kernels after the stage horizon has been fixed.

A record is active at the start of block extension exactly when its current state is open. Activity is recomputed after all preceding local refinements and parallel tempering swaps. Thus, if an accepted refinement removes a terminal token, the record becomes active and is extended at the next stage. A currently terminated record is extended deterministically through its post-terminal tail. At rung k, every model-generated token is drawn according to

$$x_t^{(k)} \sim g_{\alpha_k,t}\left(\cdot \mid \mathbf{x}_0, \mathbf{x}_{<t}^{(k)}\right), \qquad t = T_{m-1}+1, \ldots, T_m. \tag{25}$$

The generation for each rung stops after the first terminal token, and following the post-terminal convention the remaining positions are filled up to T_m deterministically. Requests may be batched across prompts and rungs without altering the record-specific law in Equation 25.

For each model-generated token, the implementation stores its base-model log probability

$$\ell_t(\mathbf{x}) := \log p_0(x_t \mid \mathbf{x}_0, \mathbf{x}_{<t}). \tag{26}$$

For every rung j, the corresponding tokenwise normalizer is

$$\zeta_{t,j}(\mathbf{x}) := \log z_{\alpha_j,t}(\mathbf{x}_0, \mathbf{x}_{<t}), \qquad j = 1, \ldots, K. \tag{27}$$

Finally, deterministic post-terminal positions satisfy $\ell_t(\mathbf{x}) = \zeta_{t,j}(\mathbf{x}) = 0$, so their tails may be stored explicitly or reconstructed when evaluating fixed-horizon sums.

B.2 Step 2: Local refinement

After extension, each rung receives its configured number of local updates. Fix a stage m and rung k. The implementation uses the state-independent uniform restart law

$$r \sim \omega_{m,k}, \qquad \omega_{m,k}(r) = \frac{1}{T_m}, \qquad r \in \{1, \ldots, T_m\}.$$

The theory below allows any state-independent restart law and requires $\omega_{m,k}(1) > 0$ only for convergence. Conditioned on r, the proposal retains the prefix before r and regenerates the suffix through the common endpoint T_m:

$$q_{m,k,r}(\mathbf{y} \mid \mathbf{x}) := \mathbf{1}\{\mathbf{y}_{<r} = \mathbf{x}_{<r}\} \prod_{t=r}^{T_m} g_{\alpha_k,t}\left(y_t \mid \mathbf{x}_0, \mathbf{y}_{<t}\right). \tag{28}$$

If $r > \tau(\mathbf{x})$, the retained prefix is already terminated and the proposal is a self-transition. The implementation detects this from the stored terminal position and issues no model call. Nevertheless, the restart law is applied over all of $\{1, \ldots, T_m\}$ rather than over $\{1, \ldots, \tau(\mathbf{x})\}$. Restricting it to the open positions would make ω depend on the current state and would require a $\tau(\mathbf{x})/\tau(\mathbf{y})$ factor in the acceptance ratio. If for the new position $r \le \tau(\mathbf{x})$, the proposal move could remove, replace, or relocate the terminal token. Because all records lie in $\mathcal{S}_{T_m}$, both directions use the same restart set and are evaluated through the same horizon; post-terminal factors equal one.

The proposal is accepted with probability

$$A_{m,k,r}(\mathbf{x}, \mathbf{y}) := 1 \wedge \frac{\pi_{m,k}(\mathbf{y}) q_{m,k,r}(\mathbf{x} \mid \mathbf{y})}{\pi_{m,k}(\mathbf{x}) q_{m,k,r}(\mathbf{y} \mid \mathbf{x})}. \tag{29}$$

The usual MH convention is used when the reverse proposal probability vanishes. The factor $\omega_{m,k}(r)$ cancels because the restart law is independent of the current state. For supported $\mathbf{x}, \mathbf{y}$ with $\mathbf{x}_{<r} = \mathbf{y}_{<r}$, the powered base-model terms also cancel, as shown next. The resulting cached acceptance ratio is

$$\log R^{\mathrm{loc}}_{m,k,r}(\mathbf{x}, \mathbf{y}) := \log \frac{\pi_{m,k}(\mathbf{y})\, q_{m,k,r}(\mathbf{x} \mid \mathbf{y})}{\pi_{m,k}(\mathbf{x})\, q_{m,k,r}(\mathbf{y} \mid \mathbf{x})} = \sum_{t=r}^{T_m} \left[\zeta_{t,k}(\mathbf{y}) - \zeta_{t,k}(\mathbf{x})\right]. \tag{30}$$

An accepted proposal replaces the suffix and all aligned ℓ- and ζ-entries. A rejected proposal retains the current completion record and its cache. Updates across rungs use independent auxiliary randomness but may be batched into a shared model evaluation.

B.3 Step 3: Swap exchange

Local refinement updates the completion records within their current rungs, while parallel tempering swaps transport complete records across the power ladder. For $k \in \{1, \ldots, K-1\}$, let σ_k denote the transposition of coordinates k and $k+1$. The proposed exchange is accepted with probability

$$A^{\mathrm{swap}}_{m,k}(\mathbf{X}) := 1 \wedge \frac{\Pi_m(\sigma_k \mathbf{X})}{\Pi_m(\mathbf{X})}. \tag{31}$$

Writing $\mathbf{x} := \mathbf{x}^{(k)}$ and $\mathbf{y} := \mathbf{x}^{(k+1)}$, the log acceptance ratio is

$$\log \frac{\Pi_m(\sigma_k \mathbf{X})}{\Pi_m(\mathbf{X})} = (\alpha_{k+1} - \alpha_k)\left[\log p_0(\mathbf{x} \mid \mathbf{x}_0) - \log p_0(\mathbf{y} \mid \mathbf{x}_0)\right]. \tag{32}$$

The cached base-model scores satisfy

$$\log p_0(\mathbf{x} \mid \mathbf{x}_0) = \sum_{t=1}^{T_m} \ell_t(\mathbf{x}),$$

so the cached completion records supply the full exchange ratio.

An ordered adjacent sweep attempts exchanges for $k = 1, \ldots, K-1$ in increasing order, so each attempt acts on the result of the preceding ones. The powers remain attached to their rung indices, while an accepted exchange moves the complete record and every aligned cache. For $K = 1$, no exchange is attempted.

B.4 Putting the implementation together

Let $n_{\mathrm{local},m}$ denote the number of synchronized local rounds in one schedule period, and let $n_{\mathrm{sweep},m}$ denote the number of ordered adjacent sweeps that follow. At stage m, block extension

Algorithm 1 Progressive fixed-horizon parallel tempering

Require: Fixed prompt $\mathbf{x}_0$, horizon T, block width B, ladder powers $\alpha_{1:K}$, stage counts $N_{1:M}$, update schedules $n_{\mathrm{local},1:M}$ and $n_{\mathrm{sweep},1:M}$, output rung $k^\star$

Ensure: A completion from rung $k^\star$

Initialize records $\mathbf{x}^{(1:K)}$ and their aligned caches as empty.
for stage $m = 1, \dots, M$ **do**
  Set the current horizon $T_m \leftarrow \min\{mB, T\}$.

▷ **Step 1: Block extension.** Expose the next block of tokens and extend each replica to T_m.
  for $k = 1, \dots, K$ **do**
    while $\mathbf{x}^{(k)}$ is open and $|\mathbf{x}^{(k)}| < T_m$ **do**
      Sample the next token from $g_{\alpha_k,t}(\cdot \mid \mathbf{x}_0, \mathbf{x}^{(k)}_{<t})$.
      Append the token and its cached ℓ- and ζ-values to $\mathbf{x}^{(k)}$.
    Apply the post-terminal convention through T_m.
  for $n = 1, \dots, N_m$ **do**

▷ **Step 2: Local refinement.** Resample suffixes within each rung, preserving its target distribution.
    for $a = 1, \dots, n_{\mathrm{local},m}$ **do**
      for $k = 1, \dots, K$ **do**
        Sample $r \sim \omega_{m,k}$.
        Propose $\mathbf{y}$ by resampling the corresponding suffix through T_m from $q_{m,k,r}(\cdot \mid \mathbf{x}^{(k)})$.
        With probability $A_{m,k,r}(\mathbf{x}^{(k)}, \mathbf{y})$, replace $\mathbf{x}^{(k)}$ and its cache by $\mathbf{y}$ and its cache.

▷ **Step 3: Replica swaps.** Exchange adjacent rungs so records can move between power levels.
    for $b = 1, \dots, n_{\mathrm{sweep},m}$ **do**
      for $k = 1, \dots, K-1$ **do**
        With probability $A^{\mathrm{swap}}_{m,k}(\mathbf{X})$, swap replicas k and $k+1$, including their caches.
Let $\mathbf{x} \leftarrow \mathbf{x}^{(k^\star)}$.
return $\mathbf{x}$ through its first configured terminal token, or all of $\mathbf{x}$ through T if no such token occurs.

first advances every record from T_{m-1} to T_m. The sampler then repeats N_m schedule periods, each consisting of $n_{\mathrm{local},m}$ local rounds followed by $n_{\mathrm{sweep},m}$ ordered sweeps. A zero count skips the corresponding operation. After stage M, the predesignated rung $k^\star$ supplies the returned record, which is presented through its first configured terminal token or through T if it is right-censored.

Algorithm 1 summarizes this execution order. The formal Markov kernels induced by these operations are introduced once, in Section C.

C Theoretical Properties: Bias and Exactness of PPT

In this section, we study the exactness of PPT. First, we define the complete transition kernel and show how one-way truncating local updates lead to structural bias in both the ensemble and the returned rung, despite the chain swaps. Then, we establish that PPT can preserve the true target for fixed-horizon MH updates and prove asymptotic convergence to the intended target under supported full restarts.

C.1 Fixed-horizon targets and PPT kernels

We start by fixing a stage m, after block extension and hold its horizon T_m fixed. We will suppress the stage index throughout for notation simplicity. Therefore, denote by $\mathcal{S}_T$ the space of records with horizon T, including records that terminate earlier and are padded with $\perp$. Subsequently, we denote by $\mathrm{len}(\mathbf{x})$ the number of generated tokens before padding, including the terminal token when present. An unterminated record in $\mathcal{S}_T$ has length T.

Then, at rung k, we write the marginal target as $\pi_k = \pi_{\alpha_k,T}$ defined in Eq. 24. The ensemble state and target are

$$\mathbf{X} = (\mathbf{x}^{(1)}, \dots, \mathbf{x}^{(K)}), \qquad \Pi(\mathbf{X}) = \prod_{k=1}^{K} \pi_k(\mathbf{x}^{(k)}).$$

Lastly, for any joint law μ, we express its marginal at rung k as $\mu^{(k)}$. Distributional deviations are measured by the total variation,

$$\mathrm{TV}(\mu, \nu) = \frac{1}{2} \sum_{\mathbf{x}} |\mu(\mathbf{x}) - \nu(\mathbf{x})| = \sup_{A} |\mu(A) - \nu(A)|.$$

The sum is over the common finite space of the two laws. Results for $\mu^{(k)}$ apply directly when rung k is selected as the output.

Local refinement. For a fixed prompt $\mathbf{x}_0$, rung k, and restart position $r \in \{1, \dots, T\}$, the proposal retains positions before r and regenerates the remaining record:

$$q_{k,r}(\mathbf{y} \mid \mathbf{x}) = \mathbf{1}\{\mathbf{y}_{<r} = \mathbf{x}_{<r}\} \prod_{t=r}^{T} g_{\alpha_k,t}(y_t \mid \mathbf{x}_0, \mathbf{y}_{<t}).$$

Here $\mathbf{y}_{<r} = (y_1, \dots, y_{r-1})$, and $g_{\alpha_k,t}$ is the normalized token proposal defined in Subsection B.2. Once a terminal token appears, each remaining padding token has proposal probability one. Thus forward and reverse proposals share the endpoint T, even when the two records terminate at prior positions. For a pair with $q_{k,r}(\mathbf{y} \mid \mathbf{x}) > 0$, the MH acceptance probability is

$$A_{k,r}(\mathbf{x}, \mathbf{y}) = 1 \wedge \frac{\pi_k(\mathbf{y}) q_{k,r}(\mathbf{x} \mid \mathbf{y})}{\pi_k(\mathbf{x}) q_{k,r}(\mathbf{y} \mid \mathbf{x})}.$$

A proposed move with zero reverse probability is rejected. Set $A_{k,r}(\mathbf{x}, \mathbf{y}) = 0$ when $q_{k,r}(\mathbf{y} \mid \mathbf{x}) = 0$, since such a pair is never proposed. The resulting transition kernel is

$$L_{k,r}(\mathbf{x}, \mathbf{y}) = \begin{cases} q_{k,r}(\mathbf{y} \mid \mathbf{x}) A_{k,r}(\mathbf{x}, \mathbf{y}), & \mathbf{y} \neq \mathbf{x}, \\ 1 - \sum_{\mathbf{z} \neq \mathbf{x}} q_{k,r}(\mathbf{z} \mid \mathbf{x}) A_{k,r}(\mathbf{x}, \mathbf{z}), & \mathbf{y} = \mathbf{x}. \end{cases}$$

Its diagonal includes both self-proposals and rejected proposals.

Restart positions have state-independent probabilities $\omega_k(r) \geq 0$, with $\sum_{r=1}^{T} \omega_k(r) = 1$. The rung-level kernel and one synchronized local sweep are

$$L_k(\mathbf{x}, \mathbf{y}) = \sum_{r=1}^{T} \omega_k(r) L_{k,r}(\mathbf{x}, \mathbf{y}), \qquad \mathcal{R}(\mathbf{X}, \mathbf{Y}) = \prod_{k=1}^{K} L_k(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}).$$

The product expresses conditional independence of the complete local updates across rungs.

Replica exchange. Let $\sigma_k \mathbf{X}$ exchange coordinates k and $k+1$. The adjacent rung MH acceptance probability is given by

$$A_k^{\mathrm{swap}}(\mathbf{X}) = 1 \wedge \frac{\Pi(\sigma_k \mathbf{X})}{\Pi(\mathbf{X})} = 1 \wedge \frac{\pi_k(\mathbf{x}^{(k+1)}) \pi_{k+1}(\mathbf{x}^{(k)})}{\pi_k(\mathbf{x}^{(k)}) \pi_{k+1}(\mathbf{x}^{(k+1)})},$$

defining the swap kernel as follows:

$$W_k(\mathbf{X}, \mathbf{Y}) = A_k^{\mathrm{swap}}(\mathbf{X}) \mathbf{1}\{\mathbf{Y} = \sigma_k \mathbf{X}\} + [1 - A_k^{\mathrm{swap}}(\mathbf{X})] \mathbf{1}\{\mathbf{Y} = \mathbf{X}\}.$$

Therefore, the ordered adjacent sweep is defined as the composition of all W_k, namely: $\mathcal{C} = W_1 \cdots W_{K-1}$. Kernels act on laws from the right, so W_1 is applied first and each exchange acts on the records produced by the preceding exchanges.

Complete PPT transition. Fix integers $n_{\text{local}} \geq 1$ and $n_{\text{sweep}} \geq 0$. One refinement round has kernel

$$P = \mathcal{R}^{\,n_{\text{local}}} \mathcal{C}^{\,n_{\text{sweep}}}.$$

Thus n_{local} counts synchronized local sweeps per round, and n_{sweep} counts adjacent exchange sweeps. Starting from a law μ_0, the law after N rounds is $\mu_N = \mu_0 P^N$. The horizon, targets, restart probabilities, and schedule remain fixed during these rounds. Block extension changes the state space and supplies the initializer; it is not part of P.

C.2 Structural bias from one-way truncation in PPT

The structural bias arises when a record's current length becomes the generation budget for subsequent local proposals. For instance, if a proposal replaces a 10-token sequence with one that terminates after 4 tokens, then next proposal is capped at 4 tokens. It would be unable to reconstruct the original 10-token record, implying that the reverse proposal has zero probability. Computing the acceptance ratio only using token scores through the shorter endpoint can miss this loss of reverse support and accept the shortening move. The exact MH rule would reject it. Accepting shortening moves while rejecting length-increasing moves results in one-way probability leakage from longer records to shorter ones with no return flow.

In PPT, a swap can give a rung a longer record from another rung, but it only moves a record that already exists. Hence, accepting shortening moves at the target violates target invariance, with parallel tempering unable to undo this violation. Furthermore, under a uniform accepted-entry condition, we show that repeated shortening also yields an asymptotic bias floor for the ensemble and returned rung.

Truncating kernel and comparison space. We denote the local transition kernel implemented by the truncating sampler at rung k by $\widetilde{L}_k$, including acceptance and rejection. Since truncation may produce unterminated records shorter than T, we compare the sampler and target on a common finite space $\widetilde{\mathcal{S}}_T \supseteq \mathcal{S}_T$ containing these records. Additionally, each marginal target π_k is extended by zero outside $\mathcal{S}_T$, and $\text{len}(\mathbf{x})$ counts generated tokens before padding.

The synchronized kernel $\widetilde{\mathcal{R}}$ applies the local updates independently across rungs. A complete round $\widetilde{P}$ consists of n_{local} such sweeps followed by n_{sweep} exchange sweeps $\widetilde{\mathcal{C}}$:

$$\widetilde{\mathcal{R}}(\mathbf{X}, \mathbf{Y}) = \prod_{k=1}^{K} \widetilde{L}_k(\mathbf{x}^{(k)}, \mathbf{y}^{(k)}), \qquad \widetilde{P} = \widetilde{\mathcal{R}}^{\,n_{\text{local}}} \widetilde{\mathcal{C}}^{\,n_{\text{sweep}}}.$$

Here $\widetilde{\mathcal{C}}$ only exchanges or retains records; it need not preserve Π. Starting from an initial law $\widetilde{\mu}_0$, the ensemble law after N complete rounds is $\widetilde{\mu}_N = \widetilde{\mu}_0 \widetilde{P}^N$.

Short records and excluded target mass. Fix a length threshold $1 \leq h < T$. We denote the set of records of length at most h by $\mathcal{B}$, and the target probability of a longer record at rung k by b_k:

$$\mathcal{B} = \{\mathbf{x} \in \widetilde{\mathcal{S}}_T : \text{len}(\mathbf{x}) \leq h\}, \qquad b_k = \pi_k(\mathcal{B}^c).$$

An ensemble belongs to $\mathcal{B}^K$ precisely when every record is short. Under the product target Π, this event has probability $\prod_k (1 - b_k)$, so the target probability of at least one long record is

$$\Pi((\mathcal{B}^K)^c) = 1 - \prod_{k=1}^{K} (1 - b_k).$$

This joint target mass and the individual masses b_k will determine the bias bounds for the ensemble and returned rung, respectively.

One-way drift. Assume local updates under $\widetilde{P}$ never increase record length. A short record then remains in $\mathcal{B}$ after every local update. To track shortening across the ensemble, define $V(\mathbf{X})$ as the number of long records:

$$V(\mathbf{X}) = \sum_{j=1}^{K} \mathbf{1}\{\mathbf{x}^{(j)} \notin \mathcal{B}\}$$

Note, that local updates under $\widetilde{P}$ cannot increase this count, and that swaps with $\widetilde{C}$ preserve it. For a record initially drawn from a law ν on $\widetilde{\mathcal{S}}_T$, let $D_k(\nu)$ be the probability that an accepted local update at rung k crosses from $\mathcal{B}^c$ into $\mathcal{B}$.

$$D_k(\nu) = \sum_{\mathbf{x}\notin\mathcal{B}} \nu(\mathbf{x})\widetilde{L}_k(\mathbf{x},\mathcal{B}).$$

Then, for any initial law ν, one local update decreases the probability of a long record by exactly the accepted crossing probability $D_k(\nu)$:

$$(\nu\widetilde{L}_k)(\mathcal{B}^c) = \nu(\mathcal{B}^c) - D_k(\nu).$$

Since no record in $\mathcal{B}$ can leave it under a local update, the loss of mass from $\mathcal{B}^c$ equals $D_k(\nu)$.

Additionally, consider an ensemble initialized from Π. We define $\varphi_k := D_k(\nu)$, when the record is drawn from the marginal target π_k. Summing over target marginals gives a decrease of $\sum_k \varphi_k$ in the expected count after the first local sweep. This loss gives the following bounds on the expected count and the TV error after one complete round:

$$\mathbb{E}_{\Pi}[V] - \mathbb{E}_{\Pi\widetilde{P}}[V] \geq \sum_k \varphi_k, \qquad \mathrm{TV}(\Pi\widetilde{P},\Pi) \geq \frac{1}{K}\sum_k \varphi_k.$$

Thus $\sum_k \varphi_k > 0$ implies $\Pi\widetilde{P} \neq \Pi$. Positive shortening flow at the target is sufficient to violate invariance; no uniform shortening probability is needed. Crossings observed under a transient law ν estimate $D_k(\nu)$, so they alone do not establish the target flow φ_k.

Uniform accepted entry. To obtain an asymptotic bias floor from any initializer, we additionally assume that every long record becomes short with probability at least $\delta \in (0,1]$ in one local update, uniformly over records and rungs. Together with the closure of $\mathcal{B}$, this gives

$$\widetilde{L}_k(\mathbf{x},\mathcal{B}) = 1 \quad (\mathbf{x}\in\mathcal{B}), \qquad \widetilde{L}_k(\mathbf{x},\mathcal{B}) \geq \delta \quad (\mathbf{x}\notin\mathcal{B}). \tag{33}$$

The lower bound concerns accepted transitions: positive probability of proposing a short record alone does not guarantee entry into $\mathcal{B}$.

Theorem 4 (Structural bias of truncating PPT)**.** *Under Eq. 33, after $N \geq 1$ rounds from any initializer $\widetilde{\mu}_0$, the ensemble and each rung k satisfy the following TV lower bounds:*

$$\mathrm{TV}(\widetilde{\mu}_N,\Pi) \geq \left[1 - \prod_{j=1}^{K}(1-b_j) - K(1-\delta)^N\right]_+, \tag{34}$$

$$\mathrm{TV}(\widetilde{\mu}_N^{(k)},\pi_k) \geq \left[b_k - K(1-\delta)^N\right]_+, \tag{35}$$

Here $[u]_+ = \max\{u,0\}$, and the correction term $K(1-\delta)^N$ bounds the probability that any long record remains. As this term vanishes, the excluded target masses give the asymptotic bias floors:

$$\liminf_{N\to\infty} \mathrm{TV}(\widetilde{\mu}_N,\Pi) \geq 1 - \prod_{j=1}^{K}(1-b_j), \tag{36}$$

$$\liminf_{N\to\infty} \mathrm{TV}(\widetilde{\mu}_N^{(k)},\pi_k) \geq b_k. \tag{37}$$

Proof. Let $\mathbf{X}'$ be the ensemble after the MH-update per rung from $\mathbf{X}$. Each long record remains long with probability at most $1-\delta$, while short records remain short. Thus the expected number of long records contracts as follows:

$$\begin{aligned}\mathbb{E}[V(\mathbf{X}') \mid \mathbf{X}] &= \sum_{j=1}^{K} \widetilde{L}_j(\mathbf{x}^{(j)},\mathcal{B}^c) \\ &\leq (1-\delta)\sum_{j=1}^{K} \mathbf{1}\{\mathbf{x}^{(j)}\notin\mathcal{B}\} = (1-\delta)V(\mathbf{X}).\end{aligned}$$

Since swaps preserve V, only local refinement can affect this count. Therefore, repeating this contraction for N iterations yields

$$\mathbb{E}_{\widetilde{\mu}_N} V \le K(1-\delta)^N.$$

Now, we consider at least one long record at any rung k which means $V \ge 1$. Since $\mathbf{1}_{V\ge 1} \le V$, the probability of this event is bounded by the expected count of long records

$$\widetilde{\mu}_N^{(k)}(\mathcal{B}^c) \le \widetilde{\mu}_N((\mathcal{B}^K)^c) \le \mathbb{E}_{\widetilde{\mu}_N} V \le K(1-\delta)^N.$$

From joint target Π, the probabilities of these event is b_k and $1-\prod_j(1-b_j)$, respectively.

Computing the TV distance between the two events above yields

$$\mathrm{TV}(\widetilde{\mu}_N^{(k)}, \pi_k) \ge [b_k - \mathbb{E}_{\widetilde{\mu}_N} V]_+ , \tag{38}$$

$$\mathrm{TV}(\widetilde{\mu}_N, \Pi) \ge \left[1 - \prod_{j=1}^{K}(1-b_j) - \mathbb{E}_{\widetilde{\mu}_N} V\right]_+ . \tag{39}$$

Substituting the preceding bound on $\mathbb{E}_{\widetilde{\mu}_N} V$ proves the finite-run inequalities. Then as $N \to \infty$, this expected count tends to zero, yielding the stated asymptotic bias floors. □

The quantities b_k determine the structural bias floors, while δ controls how fast, they are approached. The inequalities above quantify structural bias caused by violation of the target preservation for the one-way truncating MH-update law.

C.3 Target preservation and convergence of fixed-horizon PPT

In our implementation, we eliminate the structural bias and one-way truncation shown above, by keeping a common horizon T fixed throughout refinement. Every record and both directions of every local proposal use this endpoint, with deterministic padding after a terminal token. Consequently, early termination does not reduce the budget for later proposals, since a restart at or before the terminal position can regenerate beyond it, up to T, when proposal support permits. MH acceptance computed on this common space preserves the intended targets, and supported full restarts ensure convergence to them. This removes the asymptotic truncation bias.

We now analyze P on $\mathcal{S}_T^K$, where the intended target is strictly positive and both proposal directions use the same horizon.

Proposition 5 (Target preservation)**.** *Each fixed-position kernel $L_{k,r}$ is reversible with respect to π_k. The local sweep $\mathcal{R}$, each swap W_k, the ordered swap sweep $\mathcal{C}$, and the complete PPT kernel P preserve* Π.

Proof. For distinct $\mathbf{x}, \mathbf{y}$, the accepted local flow is

$$\pi_k(\mathbf{x}) L_{k,r}(\mathbf{x},\mathbf{y}) = \min\{\pi_k(\mathbf{x}) q_{k,r}(\mathbf{y} \mid \mathbf{x}), \pi_k(\mathbf{y}) q_{k,r}(\mathbf{x} \mid \mathbf{y})\},$$

which is symmetric in $\mathbf{x}, \mathbf{y}$. Thus each fixed-position kernel preserves π_k, as does its state-independent mixture L_k. Independence across rungs then gives $\Pi\mathcal{R} = \Pi$. Similarly, the accepted swap flow is

$$\Pi(\mathbf{X}) A_k^{\mathrm{swap}}(\mathbf{X}) = \min\{\Pi(\mathbf{X}), \Pi(\sigma_k \mathbf{X})\},$$

which is symmetric under the exchange. Hence $\Pi W_k = \Pi$. Composing these invariant kernels proves $\Pi\mathcal{C} = \Pi$ and $\Pi P = \Pi$. The ordered compositions need not be reversible. □

In contrast, the error of the one-way truncation violates target invariance. If $\sum_k b_k > 0$, then starting from Π gives a positive expected count $\mathbb{E}_\Pi V = \sum_k b_k$, which one complete truncating round strictly decreases. Hence $\Pi\widetilde{P} \ne \Pi$.

Supported full restart. At $r = 1$, no part of the current record is retained, so

$$q_k^{\text{full}}(\mathbf{y}) := q_{k,1}(\mathbf{y} \mid \mathbf{x}) = \prod_{t=1}^{T} g_{\alpha_k,t}(y_t \mid \mathbf{x}_0, \mathbf{y}_{<t})$$

is independent of $\mathbf{x}$. Suppose every rung selects this restart with probability $\omega_k(1) > 0$, and $q_k^{\text{full}}(\mathbf{y}) > 0$ for every $\mathbf{y} \in \mathcal{S}_T$. Define

$$\varepsilon_k = \omega_k(1) \min_{\mathbf{y} \in \mathcal{S}_T} \frac{q_k^{\text{full}}(\mathbf{y})}{\pi_k(\mathbf{y})}, \qquad \varepsilon = \prod_{k=1}^{K} \varepsilon_k.$$

The minimum ratio measures worst-case proposal coverage: it is the largest constant for which the proposal assigns at least that multiple of the target probability to every record. Multiplication by $\omega_k(1)$ accounts for how often a full restart is selected. The proof below shows that, regardless of the current record, the local transition contains a component of weight ε_k distributed as π_k. Independence makes ε the corresponding weight of the product target in a joint local sweep. Finiteness, positive support, and normalization imply $0 < \varepsilon_k \leq 1$ and $0 < \varepsilon \leq 1$.

Theorem 6 (Convergence of the PPT ensemble and returned rung)**.** *Under the supported-full-restart conditions above,* Π *is the unique invariant law of* P*. For any initializer* μ_0 *on* $\mathcal{S}_T^K$*, any rung* k*, and* $N \geq 1$*,*

$$\mathrm{TV}(\mu_N^{(k)}, \pi_k) \leq \mathrm{TV}(\mu_N, \Pi) \leq (1-\varepsilon)^N \, \mathrm{TV}(\mu_0, \Pi) \xrightarrow[N \to \infty]{} 0. \tag{40}$$

Proof. The full-restart component of the local MH kernel satisfies

$$\begin{aligned} L_k(\mathbf{x}, \mathbf{y}) &\geq \omega_k(1) q_k^{\text{full}}(\mathbf{y}) \min\left\{1, \frac{\pi_k(\mathbf{y}) q_k^{\text{full}}(\mathbf{x})}{\pi_k(\mathbf{x}) q_k^{\text{full}}(\mathbf{y})}\right\} \\ &= \omega_k(1) \pi_k(\mathbf{y}) \min\left\{\frac{q_k^{\text{full}}(\mathbf{y})}{\pi_k(\mathbf{y})}, \frac{q_k^{\text{full}}(\mathbf{x})}{\pi_k(\mathbf{x})}\right\} \geq \varepsilon_k \pi_k(\mathbf{y}). \end{aligned}$$

For $\mathbf{y} = \mathbf{x}$, self-proposals alone provide this lower bound; rejections only add diagonal mass. Multiplying over rungs gives

$$\mathcal{R}(\mathbf{X}, \mathbf{Y}) \geq \varepsilon \Pi(\mathbf{Y}) \quad \text{for all } \mathbf{X}, \mathbf{Y} \in \mathcal{S}_T^K.$$

If $\varepsilon = 1$, one local sweep has law Π, and later invariant updates preserve it. Otherwise define the residual kernel

$$Q(\mathbf{X}, \mathbf{Y}) = \frac{\mathcal{R}(\mathbf{X}, \mathbf{Y}) - \varepsilon \Pi(\mathbf{Y})}{1 - \varepsilon}.$$

The lower bound makes Q nonnegative, its rows sum to one, and $\Pi\mathcal{R} = \Pi$ implies $\Pi Q = \Pi$. Consequently, it holds that

$$\mathrm{TV}(\nu\mathcal{R}, \Pi) = (1-\varepsilon) \, \mathrm{TV}(\nu Q, \Pi Q) \leq (1-\varepsilon) \, \mathrm{TV}(\nu, \Pi).$$

This implies that each local refinement step contracts the error by a factor of at most $1 - \varepsilon$. Additionally, since swaps preserve Π and cannot increase TV, we conclude that the joint bound is proved for the entire algorithm. Marginalization proves the rung bound. Convergence from every initializer establishes uniqueness of the invariant law. □

Scope. It is noted that the supported full restarts are a sufficient condition for convergence. Additionally, the coefficient ε is a worst-case guarantee and can be very small. Uniform restart selection alone contributes a factor T^{-K} through $\omega_k(1) = 1/T$, and the minimum proposal-to-target ratio can be smaller still. Theorem 6 therefore establishes asymptotic exactness; it does not certify a small error at a practical refinement budget. At the terminal horizon, the law produced by the preceding block extensions serves as μ_0, and further fixed-horizon refinement converges to the output target without requiring consistency between stage targets at different horizons. The statement concerns refinement at a fixed terminal horizon, not the number of block extensions.

C.4 A FINITE ILLUSTRATION OF TARGET PRESERVATION AND ONE-WAY TRUNCATION BIAS

Lastly, we present an example instantiating the lower bound of Subsection 3.2. It contrasts a fixed-horizon Metropolis–Hastings transition with the unbalanced current-length-capped transition discussed above. The example makes both effects explicit on a common 4-state comparison space.

Four-state example. Take one rung with a fixed horizon $T = 2$, $\alpha = 2$, and vocabulary $\{a, e\}$, where e is terminal and $p_0(a \mid h) = p_0(e \mid h) = 1/2$ at every open prefix. Let $X_1 = (e, \perp)$, $X_2 = (a)$, $X_3 = (a, e)$, and $X_4 = (a, a)$, with respective lengths $(1, 1, 2, 2)$. The valid fixed-horizon space is $\mathcal{S}_2 = \{X_1, X_3, X_4\}$; X_2 is prematurely censored and appears only in the enlarged comparison space. Write $p_{0,2}$ for the base record law at horizon two, extended by zero on X_2. Its probabilities and powered target, obtained by normalizing $p_{0,2}^2$ with $Z = 3/8$, are

$$(p_{0,2}(X_i))_{i=1}^4 = \left(\frac{1}{2}, 0, \frac{1}{4}, \frac{1}{4}\right), \qquad \pi = \left(\frac{2}{3}, 0, \frac{1}{6}, \frac{1}{6}\right). \tag{41}$$

Here X_3 terminates at the horizon, whereas X_4 reaches the horizon without a terminal token. Both are valid records; the shorter unfinished record X_2 has zero target mass, despite its nonzero prefix likelihood. Symmetry gives the full-restart proposal $q^{\text{full}} = p_{0,2}$, but $q^{\text{full}} \neq \pi$: the proposal is normalized tokenwise, whereas π uses a single sequence-level normalizer. Deterministic padding leaves these weights unchanged.

The corrected update chooses a restart position uniformly from $\{1, 2\}$ and regenerates through the common endpoint T, padding after EOS. Its full-restart component is an independence proposal with acceptance probability $A_1(\mathbf{x}, \mathbf{y}) = 1 \wedge p_{0,2}(\mathbf{y})/p_{0,2}(\mathbf{x})$ on valid records. The unbalanced update instead chooses the restart uniformly from $\{1, \dots, \text{len}(\mathbf{x})\}$, caps regeneration at $\text{len}(\mathbf{x})$, and applies the token-score acceptance rule through the candidate endpoint without checking reverse support. The two kernels are found to be

$$P^{\text{fix}} = \begin{pmatrix} 7/8 & 0 & 1/16 & 1/16 \\ 0 & 1 & 0 & 0 \\ 1/4 & 0 & 3/8 & 3/8 \\ 1/4 & 0 & 3/8 & 3/8 \end{pmatrix}, \qquad \widetilde{P} = \begin{pmatrix} 1/2 & 1/2 & 0 & 0 \\ 1/2 & 1/2 & 0 & 0 \\ 1/4 & 0 & 3/8 & 3/8 \\ 1/4 & 0 & 3/8 & 3/8 \end{pmatrix}. \tag{42}$$

The self-loop at X_2 in P^{fix} only embeds the corrected kernel in the comparison space; this zero-target state is unreachable from $\mathcal{S}_2$.

Then, we consider the laws $\nu_N = \delta_{X_4}(P^{\text{fix}})^N$ and $\widetilde{\nu}_N = \delta_{X_4}\widetilde{P}^N$, with no block extensions during refinement. Starting from X_3 gives the same TV curves: $\pi(X_3) = \pi(X_4)$, and their transition rows coincide in each kernel. Direct calculation gives $\pi P^{\text{fix}} = \pi$ and $\text{TV}(\nu_N, \pi) \to 0$. In contrast, $\widetilde{P}$ neither preserves π nor converges to it: $\widetilde{\nu}_N \to \frac{1}{2}\delta_{X_1} + \frac{1}{2}\delta_{X_2}$, and

$$\begin{aligned} \pi\widetilde{P} &= \left(\frac{5}{12}, \frac{1}{3}, \frac{1}{8}, \frac{1}{8}\right) \neq \pi, \\ \lim_{N\to\infty} \text{TV}(\widetilde{\nu}_N, \pi) &= \frac{1}{2}\left(\frac{1}{6} + \frac{1}{2} + \frac{1}{6} + \frac{1}{6}\right) = \frac{1}{2}. \end{aligned} \tag{43}$$

More explicitly, for $N \geq 1$,

$$\begin{aligned} \text{TV}(\nu_N, \pi) &= \frac{2}{3}\left(\frac{5}{8}\right)^N, \\ \text{TV}(\widetilde{\nu}_N, \pi) &= \max\left\{\frac{1}{6} + \frac{1}{3}\left(\frac{3}{4}\right)^N, \frac{1}{2} - \frac{2}{3}\left(\frac{3}{4}\right)^N\right\}. \end{aligned} \tag{44}$$

At $N = 2$, the unbalanced TV error has decreased to $17/48$, but remains above the corrected error $25/96$. The unbalanced error reaches its minimum $37/128$ at $N = 4$, then rises toward $1/2$. Finite-step improvement therefore does not establish target preservation. Figure 6 shows this transient behavior and the subsequent asymptotic separation.

(a) Selected TV values

N	Fixed horizon	Unbalanced
0	0.83333	0.83333
1	0.41667	0.41667
2	0.26042	0.35417
3	0.16276	0.30729
4	0.10173	0.28906
5	0.06358	0.34180
10	0.00606	0.46246
∞	0	1/2

(b) Deviation from π

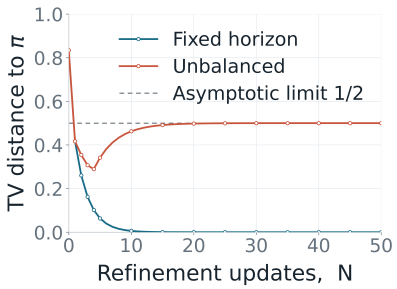

Figure 6: Selected values and total-variation trajectories from X_4 (identical TV values from X_3). The unbalanced curve uses the unbalanced kernel $\widetilde{P}$. The dashed line is the attained asymptotic floor $1/2$; TV can lie below it at finite N.

Recovery of the structural lower bound. The example realizes the confinement mechanism of Subsection C.2 with the closed set of one-token records. From either longer record, the chain enters this set in one step with probability $1/4$, while the intended target assigns mass $1/3$ to its complement:

$$\mathcal{B} = \mathcal{B}_1 = \{X_1, X_2\}, \quad \delta = \frac{1}{4}, \quad b_1 = \pi(\mathcal{B}^c) = \frac{1}{3}.$$

With $K = n_{\text{local}} = 1$, Eq. 35 gives the finite-step lower bound $\left[\frac{1}{3} - (\frac{3}{4})^N\right]_+$, and hence the asymptotic floor $1/3$. This length-only bound can be sharpened by using the zero target mass of the prematurely censored record X_2. The second coordinate of $\widetilde{\nu}_N$ gives, for $N \geq 1$,

$$\mathrm{TV}(\widetilde{\nu}_N, \pi) \geq \widetilde{\nu}_N(X_2) = \frac{1}{2} - \frac{2}{3}\left(\frac{3}{4}\right)^N. \tag{45}$$

For every $N \geq 4$, the mass on each valid record is at most its target mass, so all excess mass lies on X_2 and this bound is an equality. It therefore recovers exactly the asymptotic floor $1/2$ attained by the unbalanced trajectory; the confinement bound $1/3$ remains a valid but weaker guarantee.

Takeaway. At a fixed refinement stage, a current-endpoint kernel—one whose proposals cannot extend past the current record's length—can only shorten a record or preserve its length. If it accepts shortening moves with positive probability under the intended target π, mass flows irreversibly toward shorter records and π cannot be invariant; the early-stopping implementation's truncated-score acceptance rule permits exactly this. Fixed-horizon refinement removes this obstruction by defining proposals on a common state space in which both directions have positive probability, so the MH correction is non-degenerate and the chain can move across record lengths.

D Power-Ladder Design

This section develops a principled design rule for the power ladder at a fixed horizon T, following (Rathore et al., 2005; Syed et al., 2022; Kofke, 2002). Recall that we denote the fixed-horizon state space with $\mathcal{S}_T$, the sequence energy by $U(\mathbf{x}) := -\log p_0(\mathbf{x} \mid \mathbf{x}_0)$, and the the sequence-level power target by $\pi_\alpha(\mathbf{x}) := Z_\alpha(\mathbf{x}_0)^{-1} e^{-\alpha U(\mathbf{x})}$. Fix the number of replicas K and the endpoint powers α_1 and α_K. Every adjacent pair satisfies $\alpha_k < \alpha_{k+1}$. The analysis assumes equilibrium draws at the two rungs involved in each exchange and studies the resulting expected swap acceptance.

The design criterion below uses equilibrium overlap (Rathore et al., 2005; Deng et al., 2023). Thus, each sequence from adjacent temperatures is represented by independent draws from its two power targets. This criterion isolates the geometry of the ladder; observed acceptance in a finite run also reflects warm-up, local mixing, and Monte Carlo variability.

D.1 Thermodynamic coordinate and exact swap overlap

For $\alpha < \beta$, let $X_\alpha \sim \pi_\alpha$ and $X_\beta \sim \pi_\beta$ independently. The swap log ratio is $G_{\alpha,\beta} := (\beta - \alpha)\{U(X_\beta) - U(X_\alpha)\}$, and the exact equilibrium acceptance is $\bar{a}(\alpha, \beta) := \mathbb{E}[1 \wedge e^{G_{\alpha,\beta}}]$.

Define $\sigma^2(s) := \mathrm{Var}_{\pi_s}[U(X)]$. The exponential-family identity $\frac{d}{ds}\mathbb{E}_{\pi_s}[U(X)] = -\sigma^2(s)$ and independence yield

$$\mathbb{E}[G_{\alpha,\beta}] = -(\beta-\alpha)\int_\alpha^\beta \sigma^2(s)\,ds, \qquad \mathrm{Var}(G_{\alpha,\beta}) = (\beta-\alpha)^2\{\sigma^2(\alpha)+\sigma^2(\beta)\}. \tag{46}$$

To describe the local scale, let $\sigma(\alpha)$ denote the energy standard deviation under π_α. The corresponding thermodynamic distance and total ladder length are

$$d(\alpha,\beta) = \int_\alpha^\beta \sigma(s)\,ds, \qquad D_{\mathrm{tot}} = d(\alpha_1,\alpha_K).$$

On the finite state space, the power family varies smoothly with α, and its mean energy satisfies $\frac{d}{d\alpha}\mathbb{E}_{\pi_\alpha}[U(X)] = -\sigma^2(\alpha)$. Together with the independence of X_α and X_β, this gives the following local behavior when $\beta = \alpha+\delta$:

$$\mathbb{E}[G_{\alpha,\alpha+\delta}] = -d(\alpha,\alpha+\delta)^2 + O(\delta^3), \qquad \mathrm{Var}(G_{\alpha,\alpha+\delta}) = 2d(\alpha,\alpha+\delta)^2 + O(\delta^3). \tag{47}$$

These expansions identify thermodynamic distance as the local scale of the swap log ratio. They do not imply Gaussianity, so Gaussian shape is an additional finite-gap approximation (Kofke, 2002). Specifically, we use

$$G_{\mathrm{G}}(\alpha,\beta) \sim \mathcal{N}\big(-d(\alpha,\beta)^2,\, 2d(\alpha,\beta)^2\big)\,,$$

and denote its predicted acceptance by $a_{\mathrm{G}}(\alpha,\beta) = \mathbb{E}[1 \wedge e^{G_{\mathrm{G}}(\alpha,\beta)}]$. All optimality claims below refer to this Gaussian surrogate.

D.2 Max–min design and the unique equi-accepting ladder

Adjacent exchanges form a path, so every trip between the endpoint rungs must cross every interface. When interfaces are attempted equally often, the smallest predicted adjacent acceptance is a natural bottleneck proxy for end-to-end transport. For fixed K, α_1, and α_K, we consequently consider

$$\max_{\substack{\alpha_2,\dots,\alpha_{K-1}:\\ \alpha_1<\alpha_2<\cdots<\alpha_{K-1}<\alpha_K}} \min_{1\le k<K} a_{\mathrm{G}}(\alpha_k,\alpha_{k+1}). \tag{48}$$

This criterion optimizes the weakest interface rather than the average acceptance across the ladder.

Lemma 7 (Optimal Gaussian equi-accepting ladder)**.** *Suppose U takes at least two values on $\mathcal{S}_T$. Under the Gaussian surrogate, the predicted acceptance between $\alpha<\beta$ is*

$$a_{\mathrm{G}}(\alpha,\beta) = A(d(\alpha,\beta)), \qquad A(d) := 2\Phi\left(-\frac{d}{\sqrt{2}}\right),$$

where Φ is the standard normal cumulative distribution function. The unique solution of Eq. 48 divides the total thermodynamic length equally among the $K-1$ adjacent interfaces:

$$\int_{\alpha_k^{\mathrm{opt}}}^{\alpha_{k+1}^{\mathrm{opt}}} \sigma(s)\,ds = \frac{D_{\mathrm{tot}}}{K-1}, \qquad k=1,\dots,K-1. \tag{49}$$

Consequently, every interface has the common predicted acceptance $A(D_{\mathrm{tot}}/(K-1))$.

Proof. For $d = d(\alpha,\beta) > 0$ and $G_{\mathrm{G}} \sim \mathcal{N}(-d^2, 2d^2)$, the Gaussian tail and its exponential tilt give $\mathbb{P}(G_{\mathrm{G}} \ge 0) + \mathbb{E}[e^{G_{\mathrm{G}}}\mathbf{1}\{G_{\mathrm{G}}<0\}] = 2\Phi(-d/\sqrt{2})$. This proves the stated acceptance formula, and A is strictly decreasing (Kofke, 2002).

For any feasible ladder, let $d_k = d(\alpha_k,\alpha_{k+1})$. Positive support and nonconstant U imply $\sigma(\alpha)>0$, while additivity gives $\sum_{k=1}^{K-1} d_k = D_{\mathrm{tot}}$. Consequently, $\min_k a_{\mathrm{G}}(\alpha_k,\alpha_{k+1}) = A(\max_k d_k)$. Since $\max_k d_k \ge D_{\mathrm{tot}}/(K-1)$, with equality if and only if all gaps equal their average, the equal-gap vector is uniquely optimal.

Finally, the map $F(\alpha) = d(\alpha_1,\alpha)/D_{\mathrm{tot}}$ is continuous and strictly increasing, so the optimal gaps determine the unique rungs $\alpha_k^{\mathrm{opt}} = F^{-1}((k-1)/(K-1))$, $k=1,\dots,K$. □

Thus equi-acceptance follows from the max–min objective: fixed total thermodynamic length forces the optimal ladder to balance every bottleneck. When U is constant on $\mathcal{S}_T$, every power target is the same, $\sigma(\alpha) = 0$, and each swap is accepted. Every feasible ladder is then max–min optimal. Lemma 7 covers the varying-energy regime in which rung placement affects transport (Rathore et al., 2005; Syed et al., 2022).

Corollary 8 (Arithmetic and geometric ladder templates)**.** *Fix K, α_1, and α_K, and apply the max–min construction of Lemma 7 to either of the following idealized thermodynamic design curves on $[\alpha_1, \alpha_K]$. Each curve yields a unique surrogate ladder:*

$$\begin{aligned} \sigma(\alpha) = \sigma_0 > 0 \quad &\Longrightarrow \quad \alpha_k^{\mathrm{opt}} = \alpha_k^{\mathrm{arith}} := \alpha_1 + \frac{k-1}{K-1}(\alpha_K - \alpha_1), \qquad k = 1, \dots, K, \\ C(\alpha) = \alpha^2\sigma^2(\alpha) = c > 0 \quad &\Longrightarrow \quad \alpha_k^{\mathrm{opt}} = \alpha_k^{\mathrm{geom}} := \alpha_1 \left(\frac{\alpha_K}{\alpha_1}\right)^{(k-1)/(K-1)}, \qquad k = 1, \dots, K. \end{aligned} \tag{50}$$

Proof. When $\sigma(\alpha) = \sigma_0$, integration gives $d(\alpha, \beta) = \sigma_0(\beta - \alpha)$. Equal thermodynamic gaps are equal raw-power gaps, and the fixed endpoints make each gap $(\alpha_K - \alpha_1)/(K-1)$. When $C(\alpha) = c$, one has $\sigma(\alpha) = \sqrt{c}/\alpha$, so integration gives $d(\alpha, \beta) = \sqrt{c}\log(\beta/\alpha)$. Equal thermodynamic gaps are then equal log-power gaps, and the fixed endpoints make each log gap $\log(\alpha_K/\alpha_1)/(K-1)$. These two substitutions yield Equation 50; by Lemma 7, every interface then has the common predicted acceptance $A(D_{\mathrm{tot}}/(K-1))$, i.e., $A\big(\sigma_0(\alpha_K - \alpha_1)/(K-1)\big)$ for the arithmetic template and $A\big(\sqrt{c}\,\log(\alpha_K/\alpha_1)/(K-1)\big)$ for the geometric one. □

These exact curves are idealized design models. Approximate flatness of $\sigma(\alpha)$ gives a near-arithmetic initialization, while approximate constancy of $C(\alpha)$ gives a near-geometric initialization. Pilot calibration can refine either template toward equal thermodynamic gaps (Kofke, 2002).

In a finite state space, exact constancy of $\sigma^2(\alpha)$ on an open interval forces $\sigma^2(\alpha) = 0$ and constant energy: analyticity extends the constant value across the positive power domain, while concentration on minimum-energy states sends the variance to zero as $\alpha \to \infty$. Therefore, positive constant variance is an idealized local model, and arithmetic spacing is a local empirical template.

E Chain Communication Protocol

Each adjacent-pair swap is itself an MH move on the joint target, so any composition of swaps preserves Π_m at every stage m, and hence Π: the communication schedule cannot change what we sample, only how fast records travel across the ladder. Hence, choosing one is an efficiency question, and its answer depends on the regime.

Rungs are indexed $1, \dots, K$ as in the main text, and edge $k \in \{1, \dots, K-1\}$ joins rungs k and $k+1$. An attempt on edge k is accepted with probability $a_k \in (0, 1]$; as in the remark of Section 3.4, acceptance decisions are taken to be independent with these fixed, history-independent probabilities, and we write

$$r_k = 1 - a_k, \qquad \rho_k = \frac{r_k}{a_k}, \qquad R = \sum_{k=1}^{K-1} \rho_k.$$

The schedules are defined as follows.

Ordered ADJ: one sweep attempts edges $1, 2, \dots, K-1$ in order, applying each accepted exchange immediately. Observe the tagged replica at complete-sweep boundaries. Its round trip starts at rung 1, visits rung K, and then returns to rung 1. Let T_{ADJ} count sweeps, referring to one tagged replica's trip, rather than completion of a trip by every replica. We proceed to derive the expected iteration count for a round trip $\mathbb{E}T_{\mathrm{ADJ}}$.

Upward passage. Follow the first upward crossing of each edge k. Successful crossings can cascade through the entire ladder within one sweep, so the passage takes one sweep plus delays caused by rejections. After a rejection at edge k, the tag stays in the prefix $\{1, \dots, k\}$ until its next attempt at that edge. Observe the tag just before edge k is scheduled in each sweep. Until the next attempt, its motion is the ordered scan confined to this prefix. Starting from k, the mean time to return to k and

retry is also k sweeps. The number of rejections before the successful crossing is geometric, with mean $\rho_k = r_k/a_k$. Independence and the Markov property make the expected delay at this edge $k\rho_k$. Summing gives

$$H_{1\to K}^{\mathrm{ADJ}} = 1 + \sum_{k=1}^{K-1} k\,\rho_k. \tag{51}$$

Downward passage. A successful move from $k+1$ to k occurs after edge $k-1$ has already been processed. Thus the tag can move down by at most one rung per sweep, so the passage takes $K-1$ sweeps plus rejection delays. After rejecting edge k from rung $k+1$, the tag makes an excursion in the suffix $\{k+1,\dots,K\}$ before retrying. Observing just before edge k is scheduled, the return-time fact gives a mean retry delay of $K-k$ sweeps. There are on average ρ_k rejections before the successful downward crossing, hence

$$H_{K\to 1}^{\mathrm{ADJ}} = (K-1) + \sum_{k=1}^{K-1} (K-k)\rho_k. \tag{52}$$

An endpoint reached during a sweep remains occupied until its end, so these passage counts agree with the stated sweep-boundary convention. Adding the two expectations gives

$$\mathbb{E}T_{\mathrm{ADJ}} = K(1+R). \tag{53}$$

Lastly, we can compare the expected iteration count above with the corresponding expected count for the asymptotically optimal schedule deterministic even–odd (DEO) introduced by (Okabe et al., 2001). More specifically in DEO, one layer attempts all even or all odd edges, alternating the two matchings deterministically. In the lifted state (k,ε), the sign $\varepsilon \in \{-1,+1\}$ is the next proposed direction. Acceptance preserves the sign; rejection reverses it. An outward direction at an endpoint reverses in one idle layer. We count the round trip, including both turnaround layers. This is the recurring endpoint-arrival convention.

The corresponding expected count is

$$\mathbb{E}T_{\mathrm{DEO}} = 2K(1+R). \tag{54}$$

Subsequently, assume d_{ADJ} and d_{DEO} denote the iteration durations, including refinement and swap communication. When the additional communication cost of ordered ADJ is negligible relative to d_{DEO}, the expected round-trip elapsed times satisfy

$$\frac{\mathbb{E}C_{\mathrm{ADJ}}}{\mathbb{E}C_{\mathrm{DEO}}} = \frac{d_{\mathrm{ADJ}}}{2d_{\mathrm{DEO}}} = \frac{1}{2} + \frac{d_{\mathrm{ADJ}} - d_{\mathrm{DEO}}}{2d_{\mathrm{DEO}}} \approx \frac{1}{2}.$$

Remark 9. DEO becomes preferable when the additional sequential communication outweighs this iteration-count advantage, namely when $d_{\mathrm{ADJ}} > 2d_{\mathrm{DEO}}$. In particular, as $K \to \infty$, the quadratic scaling $\Theta(K^2(1+R))$ of the ordered sweep dominates. Under these timing assumptions, DEO is asymptotically faster as the number of replicas grows.

F Extended Related Work

Distribution sharpening for LLMs. A major goal of LLM post-training is to shift probability mass toward higher-quality responses. RLHF optimizes against learned human-preference rewards (Ouyang et al., 2022), while RLVR methods such as GRPO optimize against verifiable task rewards and have improved mathematical and coding performance (Shao et al., 2024; Guo et al., 2025; Hu et al., 2025; Lambert et al., 2024). However, post-training is expensive, reward-dependent, and can reduce output diversity by reinforcing already likely solution modes (He et al., 2025; Yue et al., 2025; Chen et al., 2025). This motivates training-free inference-time methods that sharpen or reshape the base distribution without updating model weights.

Inference-time scaling. Best-of-N and reward-reranking improve generation by sampling multiple completions and selecting a high-scoring one (Wang et al., 2025; Huang et al., 2025). Temperature and truncation methods, including low-temperature, modify local token probabilities to balance quality and diversity (Meister et al., 2023; Troshin et al., 2025; Du et al., 2025). These approaches

are effective but are generally heuristic: they do not sample from a specified sequence-level target distribution.

Power distribution sampling. Power distribution sampling replaces these heuristics with an explicit sequence-level target, the power distribution $\pi_\alpha \propto p_0^\alpha$ with $\alpha > 1$ (Eq. 1), which sharpens the base model over whole completions rather than token by token. Karan & Du (2026) introduced Power Sampling, which targets π_α with a Metropolis–Hastings chain whose proposals regenerate a uniformly chosen suffix under the tokenwise-powered proposal g_α (Eq. 2), applied blockwise over progressively longer prefixes. Without training, rewards, or verifiers, it nearly matches and in some cases exceeds GRPO on MATH500, HumanEval, and GPQA, but repeated suffix regeneration makes it expensive at inference time. Two follow-ups reduce this cost. Ji et al. (2026) show that π_α can be approximated autoregressively by the low-temperature distribution rescaled with token-level factors that capture the quality of future continuations; they estimate these factors with Monte Carlo rollouts and a jackknife bias correction, which removes the MCMC loop and reduces latency by more than an order of magnitude. PowerSMC (Azizi et al., 2026) instead targets π_α with sequential Monte Carlo: a small set of particles is advanced in parallel under g_α, which they show is the unique prefix-only proposal minimizing the incremental weight variance, with token-by-token importance-weight correction, resampling when needed, and an exponent-bridging schedule that stabilizes the particles. It matches or exceeds MH-based Power Sampling on MATH500 at a fraction of its latency. PPT builds on the local MH refinement of Power Sampling, and we compare against both Power Sampling and PowerSMC in our experiments.

Monte Carlo methods for LLM inference. MCMC and SMC provide principled alternatives to heuristic decoding for unnormalized sequence-level targets. MCMC constructs a Markov chain with the desired stationary distribution, and has been used for energy-based controllable text generation, blockwise text resampling, quality-aware machine translation, and constrained LM sampling (Mireshghallah et al., 2022; Forristal et al., 2023; Faria et al., 2024; Anaya Gonzalez et al., 2026). SMC instead maintains weighted particles across a sequence of intermediate targets and has recently been applied to twisted SMC for language-model inference, mathematical reasoning, and reward-guided decoding (Zhao et al., 2024; Feng et al., 2025; Markovic-Voronov et al., 2026).

Parallel tempering. Parallel tempering is designed to improve MCMC mixing by running multiple tempered replicas and swapping their states with a Metropolis correction. It originated in spin-glass simulation (Swendsen & Wang, 1986), was generalized to statistical MCMC (Geyer, 1991), and became standard in exchange Monte Carlo and molecular simulation (Hukushima & Nemoto, 1996; Sugita & Okamoto, 1999; Earl & Deem, 2005). Later work tunes the temperature ladder (Kone & Kofke, 2005), optimizes the annealing path (Syed et al., 2021), and uses non-reversible replica dynamics (Okabe et al., 2001; Syed et al., 2022). These methods exploit high-temperature replicas for exploration and low-temperature replicas for target concentration. Existing LLM samplers rarely use this exchange mechanism: they typically rely on a single MH chain (Karan & Du, 2026), independent best-of-N samples (Huang et al., 2025), or SMC particles (Azizi et al., 2026). PPT adapts parallel tempering to LLM sampling by transferring states from exploratory replicas to sharpened targets, with the aim of improving finite-budget mixing while exploiting parallel execution.

Recently, He et al. (2026) developed replica-exchange sampling for LLMs using intermediate targets at a common sharpening temperature indexed by prefix length, with exchange proposals that extend and truncate token chunks. Instead, PPT couples replicas at different sharpening powers over a common horizon at each generation stage, exchanging entire sequence records using cached likelihoods without additional model evaluations. This construction explicitly couples exploration under flatter distributions with concentration under sharper targets over the same completion space.

G Experimental Settings

G.1 Choice of Baselines

- **Standard Decoding.** We sample directly from the autoregressive model p_0 using standard ancestral decoding at temperature one, corresponding to the unsharpened target $\alpha = 1$. This baseline measures the raw capability of the underlying model and serves as the reference against which all sharpening gains are reported.

- **Low Temperature Decoding.** We decode from the autoregressive model at a reduced sampling temperature $\tau = 1/\alpha < 1$, the standard heuristic for concentrating probability on high-likelihood tokens. Because temperature sharpens each next-token conditional independently, it does not reproduce the sequence-level power target $\pi_\alpha \propto p_0^\alpha$ (Equation 23).
- **Power Sampling (Blockwise MH Sampler)** (Karan & Du, 2026)**.** For the *Power Sampling* results of Table 1, we use the authors' released implementation without modification. In particular it retains the early stopping analyzed in Section 3.2, where each proposal is generated and scored only through the current realized length.
- **Power Sequential Monte Carlo**: A particle-based alternative that approximates the same tempered target by maintaining a population of partial sequences and periodically resampling them in proportion to their power-reweighted importance weights. It represents the sequential Monte Carlo counterpart to our parallel tempering scheme, allowing us to contrast population resampling against exchange-based transport.
- **GRPO**: Group Relative Policy Optimization (Shao et al., 2024) is a reinforcement-learning method that fine-tunes the model weights to concentrate probability mass on high-reward trajectories. Unlike the inference-time samplers above, GRPO modifies the model itself and requires a verifiable reward signal. we include it as a strong training-based reference point to assess how far a purely inference-time sampler can close the gap to a method that pays the additional cost of post-training.

For Power Sampling (Karan & Du, 2026) and PowerSMC, we based our experiments on the respective official implementations, left their sampling kernels unmodified, and tuned their method-specific hyperparameters independently for each model to obtain the strongest performance we could achieve under the common evaluation protocol.

G.2 Choice of Datasets

- **MATH500**: The MATH dataset (Hendrycks et al., 2021) consists of competition math problems spanning seven categories including geometry, number theory, and precalculus. There are 12500 problems total, with 7500 training problems and 5000 test problems. MATH500 is a fixed subset of 500 problems from the test set (Lightman et al., 2024).
- **HumanEval**: HumanEval is a set of 164 handwritten programming problems covering algorithms, reasoning, mathematics, and language comprehension Chen et al. (2021). Each problem has an average of 7.7 associated unit tests, where solving the problem corresponds to passing all unit tests.
- **GPQA**: GPQA Rein et al. (2024) is a dataset of multiple-choice science questions (physics, chemistry, and biology) which require advanced reasoning skills to solve. We use the GPQA Diamond subset for evaluation, which consists of 198 questions which represent the highest quality subset of the GPQA dataset.
- **GSM8K**: GSM8K is a dataset of grade-school math word problems that require multi-step arithmetic reasoning to solve (Cobbe et al., 2021). Each problem is paired with a natural-language solution and a single numeric final answer, and correctness is scored by exact match on that answer. We evaluate on the standard 1319-problem test split.
- **AIME 24&25**: The AIME 24&25 sets collect the problems from the American Invitational Mathematics Examination for those years, a high-difficulty olympiad-style competition whose questions each have an integer answer between 0 and 999 (Zhang & Math-AI, 2024; Zhang & Team, 2025). We evaluate on the combined 60-problem set, both examinations (I and II) from 2024 and 2025 respectively. This benchmark probes performance on the hardest, most reasoning-intensive mathematics in our suite, where exploration of distinct solution paths matters most.
- **LCB v5**: LiveCodeBench is a contamination-aware coding benchmark that continuously collects problems from competitive-programming contests (Jain et al., 2025).

G.3 Evaluation Metrics

Task accuracy. Our primary metric is strict pass@1 accuracy: for each benchmark item, every method returns exactly one completion, and we report the percentage of items solved. No method uses reward- or verifier-based reranking or oracle filtering. For GSM8K and AIME 24&25, we extract the final numeric answer and require an exact match after standard normalization. GPQA

Table 6: PPT hyperparameters for the `Qwen3-4B` and `Qwen3-8B` evaluations. B is the generation block size, and N_{MCMC} is the number of local refinement steps per block.

Benchmark	K	$\{\alpha_k\}$	T	B	N_{MCMC}
MATH500	4	$\{2.35, 2.5, 2.7, 2.9\}$	8192	4096	10
AIME 24&25	4	$\{1.8, 1.95, 2.15, 2.4\}$	32768	8192	10
GPQA-Diamond	3	$\{1.15, 1.3, 1.5\}$	16384	4096	6
LiveCodeBench v5	4	$\{1.8, 2.0, 2.2, 2.4\}$	8192	2048	8
HumanEval	3	$\{1.5, 1.6, 1.8\}$	4096	2048	20
GSM8K	6	$\{2.0, 2.3, 2.6, 3.0, 3.5, 4.0\}$	3072	192	10

is scored by exact match on the selected multiple-choice option. MATH500 is scored with the PRM800K/Hendrycks-MATH equivalence grader.

Code correctness. For HumanEval, and LiveCodeBench v5, we report execution-based pass@1. A completion is correct only if it compiles or executes successfully and passes every associated unit test; invalid programs, runtime errors, and timeouts receive zero credit.

Diagnostic and efficiency metrics. To characterize the generated reasoning traces, we additionally report the average token log-likelihood under the base model, $T^{-1}\sum_{t=1}^{T}\log p_0(x_t \mid x_{<t})$. Confidence is measured as the mean negative next-token entropy,

$$C(x) = \frac{1}{T}\sum_{t=1}^{T}\sum_{v\in\mathcal{V}} p_0(v \mid x_{<t})\log p_0(v \mid x_{<t}),$$

with values closer to zero indicating greater confidence; we first average over generated positions within each completion and then average the resulting completion-level scores across benchmark items.

G.4 Configuration

Models and prompting. The evaluated checkpoints range from 3.8B to 9B parameters and include base, instruction-tuned, and GRPO-trained models from the Qwen2.5, Qwen3, Qwen3.5, Phi-3.5, and Tulu-3 families. No additional training is performed as part of our evaluation. The Qwen3 models use their tokenizer-provided chat templates, whereas the Qwen2.5 models use raw prompts. Experiments were run on NVIDIA H100 hardware. The requested completion budget is capped per item by the checkpoint context length minus the prompt length and a 64-token safety margin. The Qwen2.5-Math experiments use $T = 3072$. For Qwen3, Table 6 shows: GSM8K use $T = 3072$, HumanEval use $T = 4096$, MATH500 and LCB v5 experiments use $T = 8192$, Qwen3 GPQA experiments use $T = 16384$, and Qwen3 AIME 24&25 experiments use $T = 32768$. Lastly, for Qwen3.5 GPQA and LCB v5 use $T = 131072$, and AIME 24&25 use $T = 65536$.

PowerSMC implementation. PowerSMC maintains normalized particle weights $\bar{w}_i$ and computes

$$\text{ESS} = \frac{1}{\sum_{i=1}^{N}\bar{w}_i^2}.$$

ESS is checked at the configured token interval and at the final horizon. When it falls below $0.5N$, the implementation performs systematic resampling and resets the particle weights. At the end of generation, one particle is sampled according to the final normalized weights. Particles are internal sampler states and are not counted as additional evaluation completions. Lastly, PowerSMC uses a per-particle EOS handling.

Across all configurations, replica K is the returned replica and has target power $\alpha^\star$. Local proposals draw the restart position uniformly from $\{1, \ldots, T_m\}$ and regenerate the suffix using temperature $\tau_k = 1/\alpha_k$, top-$p = 1$, disabled top-k, and min-$p = 0$. We apply one ordered adjacent sweep, $(1, 2), (2, 3), \ldots, (K-1, K)$, after every synchronized local-MH round.

Table 7: PPT hyperparameters for the `Qwen3.5-9B` evaluations. B is the generation block size, and N_{MCMC} is the number of local refinement steps per block.

Benchmark	K	$\{\alpha_k\}$	T	B	N_{MCMC}
GPQA-Diamond	3	$\{1.05, 1.10, 1.25\}$	131,072	131,072	10
AIME 24&25	3	$\{1.25, 1.4, 1.6\}$	65,536	65,536	5
LiveCodeBench v5	3	$\{1.1, 1.25, 1.4\}$	131,072	131,072	5

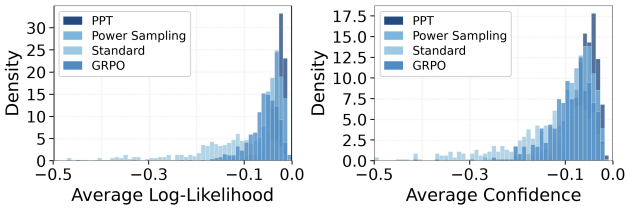

Figure 7: Confidence comparison across methods on MATH500 using Qwen3-8B; values closer to zero indicate more plausible traces.

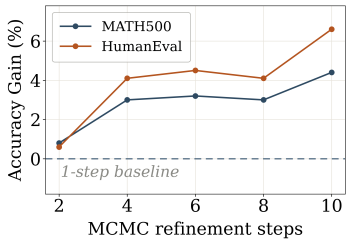

Figure 8: Accuracy gain over the one-step baseline for varying number of refinement steps.

Common Comparison Protocol Two entries are compared, in Table 1 or in Appendix H, only when they satisfy the following conditions:

- **Grading.** Public or official graders throughout.
- **Prompt.** Every baseline is compared with PPT under an identical prompt.
- **Completion cap.** Identical horizon for all baselines.

H Extended Experimental Results

H.1 Extended Main Results

Table 8 shows that PPT is the most consistent method across model families and task types, attaining the best or tied-best performance in the large majority of settings. Its advantage is clearest for the Qwen3 models on the hardest tasks: the largest margin over the strongest baseline is on AIME 24&25 for Qwen3-4B (5.0 points) and on LCB v5 for Qwen3-8B (4.3 points). Results from other model families are more mixed. On tasks where the model is already near saturation, PPT remains competitive. This pattern is consistent with the central intuition behind parallel tempering: exploratory replicas can search across different reasoning modes, while exchange lets the output rung inherit promising trajectories instead of having to discover them on its own. Because the main comparison changes more than one component relative to Power Sampling, the table establishes the breadth of the improvement; the matched ablations below isolate the effects of fixed-horizon refinement and replica exchange.

Acceptance Rates Table 9 reports the mean acceptance rate of swap proposals between adjacent rungs, averaged over all exchange attempts in the run, across six benchmarks and six models. Two regularities stand out. First, acceptance is governed by the model and by the task. Second, the Qwen3 models accept far fewer swaps $(0.18 - 0.37)$ than the other models $(0.35 - 0.58)$. Both regimes nonetheless sit in the range classically recommended for parallel tempering (roughly $0.15 - 0.6$), so the ladders are communicating rather than idling; a denser ladder for the newer and more capable models would raise acceptance further at the cost of more replicas, and we leave that trade-off to the ladder-design discussion.

Confidence of Trace Figure 7 complements task accuracy by asking whether the returned traces lie in regions that the model itself regards as plausible. As illustrated, for both average log-likelihood and negative next-token entropy, PPT concentrates more of the distribution near zero and suppresses the diffuse low-score tail seen most clearly under standard decoding. This is the behavior expected from sequence-level sharpening: trajectories discovered across the ladder can move to the output

Table 8: Single-sample performance (%) across reasoning, coding, and knowledge benchmarks. **Bold** indicates the best result within each base-model group. Scores are averaged over 5 random seeds and reported as mean ± standard deviation. Dashes '—' indicate that no GRPO result is available for Phi 3.5 instruct.

Method	MATH 500	GPQA	Human Eval	GSM8K	AIME 24&25	LCB v5
Qwen 2.5 MATH						
Standard	51.6 ±1.8	31.6 ±1.1	30.5 ±1.8	71.4 ±0.9	4.3 ±0.9	2.5 ±0.9
Lower Temperature	72.7 ±0.7	35.9 ±2.3	53.0 ±2.2	85.1 ±1.2	11.0 ±1.5	7.6 ±2.1
Power Sampling	76.7 ±0.4	35.9 ±1.8	56.7 ±0.4	86.7 ±0.3	15.7 ±1.5	8.9 ±1.2
PowerSMC	78.0 ±0.7	32.8 ±1.4	59.1 ±2.6	85.4 ±1.0	13.3 ±1.2	10.7 ±1.0
GRPO	77.3 ±0.6	36.5 ±2.7	53.7 ±1.1	86.4 ±0.4	13.3 ±1.7	2.8 ±1.1
PPT (Ours)	**79.6** ±0.8	**37.6** ±1.1	**60.6** ±1.3	**91.4** ±1.1	**17.7** ±1.5	**12.5** ±1.9
Qwen 2.5						
Standard	43.0 ±3.0	28.6 ±2.0	20.7 ±0.9	59.2 ±1.1	4.3 ±1.9	2.5 ±0.3
Lower Temperature	63.8 ±2.2	32.8 ±2.1	51.8 ±0.7	81.0 ±0.5	6.7 ±2.0	15.1 ±1.8
Power Sampling	69.6 ±2.1	33.2 ±2.9	58.7 ±2.5	86.1 ±0.5	9.0 ±1.5	6.6 ±0.8
PowerSMC	70.3 ±0.7	34.0 ±0.6	59.4 ±0.8	86.3 ±0.4	12.7 ±1.9	15.0 ±0.7
GRPO	**74.0** ±1.1	33.5 ±2.5	57.3 ±0.9	84.2 ±0.4	13.3 ±1.2	**16.9** ±1.2
PPT (Ours)	**74.0** ±1.5	**34.9** ±0.7	**61.6** ±0.9	**88.1** ±0.5	**15.7** ±0.9	9.3 ±0.5
Phi 3.5 instruct						
Standard	37.9 ±1.4	31.6 ±1.4	37.9 ±1.1	80.4 ±0.5	2.3 ±0.9	7.2 ±1.3
Lower Temperature	41.1 ±0.8	31.3 ±0.4	61.0 ±1.2	82.3 ±0.4	4.3 ±1.5	8.1 ±1.8
Power Sampling	38.2 ±1.2	29.8 ±2.6	67.0 ±0.8	81.9 ±0.7	**6.7** ±1.2	11.7 ±1.9
PowerSMC	45.1 ±0.8	31.8 ±1.1	63.4 ±0.4	86.7 ±0.6	4.3 ±0.9	8.4 ±2.2
GRPO	—	—	—	—	—	—
PPT (Ours)	**48.4** ±1.1	**33.4** ±0.9	**68.3** ±0.7	**88.6** ±0.4	4.3 ±1.5	**12.0** ±2.2
Tulu						
Standard	44.9 ±1.0	26.8 ±0.5	59.8 ±1.8	85.4 ±0.3	2.3 ±1.5	9.8 ±0.9
Lower Temperature	45.2 ±1.1	26.3 ±0.5	63.4 ±1.5	87.3 ±0.2	2.3 ±1.5	10.0 ±1.3
Power Sampling	49.9 ±0.9	32.3 ±2.2	59.8 ±0.9	87.8 ±0.1	4.3 ±0.9	11.1 ±0.9
PowerSMC	50.3 ±0.6	31.8 ±0.6	60.2 ±0.9	87.7 ±0.2	4.3 ±0.9	12.2 ±0.7
GRPO	47.0 ±0.8	24.7 ±0.4	61.3 ±1.1	87.6 ±0.1	4.3 ±1.5	11.0 ±0.5
PPT (Ours)	**51.8** ±0.6	**35.9** ±0.7	**64.0** ±1.6	**89.7** ±0.3	**6.7** ±3.1	**12.6** ±0.6
Qwen3-4B						
Standard	83.3 ±1.6	51.0 ±1.3	85.4 ±2.6	94.6 ±0.1	70.7 ±1.9	45.6 ±0.4
Lower Temperature	85.1 ±0.4	48.5 ±1.2	86.6 ±1.8	94.5 ±0.2	72.0 ±1.8	44.8 ±1.5
Power Sampling	83.5 ±2.5	51.0 ±0.7	90.2 ±1.5	94.2 ±0.1	73.3 ±1.2	55.1 ±1.7
PowerSMC	83.2 ±1.2	47.5 ±1.4	87.9 ±0.9	94.3 ±0.1	61.7 ±1.2	52.3 ±1.5
GRPO	85.6 ±0.6	50.0 ±0.6	91.6 ±0.7	94.6 ±0.1	73.3 ±1.7	52.7 ±1.8
PPT (Ours)	**86.0** ±0.7	**53.6** ±0.9	**93.9** ±2.3	**94.8** ±0.2	**78.3** ±2.4	**57.9** ±1.3
Qwen3-8B						
Standard	83.9 ±1.2	55.8 ±1.7	84.9 ±1.2	94.6 ±0.3	72.7 ±3.8	48.1 ±1.7
Lower Temperature	84.6 ±0.4	54.5 ±2.1	87.2 ±0.6	95.0 ±0.2	73.0 ±4.6	48.4 ±2.5
Power Sampling	82.2 ±1.8	57.6 ±1.6	90.7 ±1.6	95.2 ±0.5	73.3 ±2.4	53.5 ±1.2
PowerSMC	84.1 ±0.6	58.0 ±0.6	90.2 ±0.4	**95.5** ±0.1	65.0 ±1.2	53.6 ±1.0
GRPO	86.4 ±0.6	53.0 ±0.7	93.3 ±1.2	94.6 ±0.4	74.0 ±0.9	54.1 ±2.3
PPT (Ours)	**88.0** ±0.6	**60.1** ±0.8	**94.9** ±1.5	**95.5** ±0.2	**78.3** ±2.6	**58.4** ±1.3

rung when they are compatible with the sharper target. These diagnostics measure the model's internal preference rather than calibration, so they support the intended concentration mechanism but do not by themselves establish correctness.

H.2 Ablation Studies

Impact of MCMC steps. The number of local refinement steps N_{MCMC} specifies the number of MH-update attempts per replica per block. Figure 8 demonstrates the accuracy gain over a single step, as most of the improvement arrives by 4 steps (+3.0 on MATH500, +4.1 on HumanEval), and the best results come at 10 steps (+4.4 and +6.6). More refinement therefore helps overall but the

Table 9: Mean swap acceptance of the replica ladder.

Dataset	Qwen3-8B	Qwen3-4B	Qwen 2.5 MATH	Qwen 2.5	Tulu	Phi 3.5 instruct
MATH500	0.199	0.242	0.446	0.500	0.487	0.373
AIME 24&25	0.192	0.276	0.404	0.533	0.449	0.392
GSM8K	0.274	0.332	0.427	0.455	0.505	0.382
GPQA-D	0.203	0.240	0.346	0.405	0.498	0.400
HumanEval	0.205	0.374	0.428	0.580	0.478	0.421
LiveCodeBench	0.222	0.260	0.530	0.511	0.482	0.425

gain is not monotone in N_{MCMC} and since local updates dominate runtime (Table 3), N_{MCMC} is the main knob for trading accuracy against cost.

Compute-matched comparison In this section, we consider two complementary compute-matched controls: a single chain given more MCMC refinement steps, and multiple uncoupled chains.

More refinement in a single chain. First, Table 10 compares PPT with $K > 1$ replicas against a single chain ($K = 1$). We increase the number of MCMC steps in the single-chain run until its total generated-token budget matches or even exceeds that of the multi-replica run. Both methods return one trace, so this comparison tests whether the same token budget is better spent on additional refinement of one chain or on interacting replicas. PPT achieves consistently higher accuracy across all eight comparisons, with the largest gains on AIME.

Table 10: Compute-matched single-trace comparison. Entries report Accuracy (Tokens$\times 10^3$), with accuracy in percent and decode tokens per problem in parentheses, including refinement proposals. The $K = 1$ control uses additional MCMC steps to match the decode-token budget of PPT ($K > 1$). **Bold** marks the higher accuracy for each benchmark and model.

Method	**MATH 500**	**GPQA**	**AIME 24&25**	**LCB v5**
Qwen3-4B				
Single chain ($K = 1$)	84.8 (112)	53.0 (212)	72.7 (95)	56.0 (255)
PPT (Ours) ($K > 1$)	**86.0** (83)	**53.6** (146)	**78.3** (87)	**57.9** (233)
Qwen3-8B				
Single chain ($K = 1$)	85.6 (131)	59.1 (363)	75.3 (215)	54.9 (213)
PPT (Ours) ($K > 1$)	**88.0** (89)	**60.1** (349)	**78.3** (151)	**58.4** (210)

Multiple traces. Subsequently, Table 11 compares PPT against two multi-trace baselines, each given the same decode-token budget as the coupled run. The *uncoupled ladder* keeps the replica count, powers $\{\alpha_k\}$, horizon, block size, and refinement schedule of PPT but disables all swap attempts, so it isolates the effect of exchanging records during generation. *Best-of-N* (BoN) draws independent samples from the model until the budget is used, so it represents the standard way of spending extra inference compute across many traces. For our metrics, we use: *Pass@1* returns one final record; *Vote* takes the majority over final answers, breaking ties by cumulative base-model log-likelihood.

For Pass@1, PPT returns its coldest-rung ($k^\star$) trace and *Best-of-N* its highest-log-likelihood trace; the *uncoupled ladder* is reported under both rules for a like-for-like comparison. The single-trace readout gives a clear comparison of sampling quality, and there PPT is the strongest method in every setting. The records returned by PPT consistently outperform both the traces from the uncoupled ladder and from BoN at the same compute. Compared with the uncoupled ladder, exchange improves the single-trace and vote accuracy in every model–benchmark setting, and on AIME 24&25 it raises all three settings for both models, by a considerable margin. This shows that the gains come from how exchange reshapes each chain's trajectory, not from spending more tokens.

Taken together, these controls show that the benefit of PPT holds whether compute is matched through deeper refinement of one chain, several chains without exchange, or independent base-model samples.

Table 11: Compute-matched accuracy (%) of PPT, the uncoupled ladder (no swaps), and Best-of-N under the same decode-token budget. For the uncoupled ladder, the two Pass@1 values correspond to returning the highest-likelihood trace (Likelihood Rank) and the trace at rung $k^\star$ (Coldest Rung). **Bold** marks the best result within each column and model group.

Method	MATH 500		GPQA		AIME 24&25	
	Pass@1	Vote	Pass@1	Vote	Pass@1	Vote
Qwen3-4B						
Uncoupled ladder (Likelihood / Coldest Rung)	84.2 / 83.6	87.6	52.3 / 52.9	48.5	75.7 / 74.3	78.3
Best-of-N	84.7	**89.2**	52.4	54.7	71.3	77.7
PPT (Ours)	**86.0**	88.8	**53.6**	**55.1**	**78.3**	**80.0**
Qwen3-8B						
Uncoupled ladder (Likelihood / Coldest Rung)	86.0 / 84.9	87.4	57.6 / 57.6	58.9	76.7 / 76.7	80.0
Best-of-N	84.6	**90.3**	56.5	60.1	72.8	78.6
PPT (Ours)	**88.0**	90.0	**60.1**	**61.6**	**78.3**	**81.7**

Table 12: Pass@1 accuracy (%) and hit-capacity rate (%, share of returned responses that exhaust the completion budget without emitting EOS) on MATH500 for the truncating single-chain sampler (Power Sampling), a fixed-horizon single chain, and PPT (fixed horizon, $K > 1$ replicas with swaps). **Bold** marks the best accuracy for each model.

Method	Qwen3-4B		Qwen3-8B	
	Acc. ↑	Hit cap. ↓	Acc. ↑	Hit cap. ↓
Truncating, $K = 1$ (Power Sampling)	83.5	9.2	82.2	12.2
Fixed-horizon, $K = 1$	84.5	8.0	85.3	9.5
PPT (Ours) (fixed-horizon, $K > 1$)	**86.0**	7.6	**88.0**	9.6

Fixed Horizon vs One-Way Truncation Table 12 separates the effect of the fixed-horizon correction from the effect of replica exchange. With exchange disabled, moving to a common fixed-horizon state space improves accuracy for both models. Retaining reversible suffix moves lets refinement reconsider where a response should end, rather than making an early EOS an irreversible change of state space. Enabling exchange further improves accuracy. The strongest results consequently come from two complementary ingredients—a valid fixed-horizon kernel that can revise trajectories in both directions, and communication that transports useful trajectories across sharpening levels.

The two kernels treat termination asymmetrically. Under the truncating kernel, an accepted early EOS permanently shrinks the proposal budget of every later local move, while a record that has not terminated can only be shortened by a proposal that itself terminates. Consequently, the terminal position is effectively frozen after a few updates. Conversely, under the fixed-horizon kernel, a restart at or before the terminal position regenerates the suffix through T_m in both directions, so refinement can move the EOS earlier or later and can convert a right-censored record into a terminated one.

Consistent with this, moving from the truncating to the fixed-horizon kernel at $K = 1$ lowers the hit-capacity rate from 9.2% to 8.0% on Qwen3-4B and from 12.2% to 9.5% on Qwen3-8B, so more responses terminate within budget, and accuracy rises at the same time. Enabling exchange on top of the fixed-horizon kernel leaves the hit-capacity rate essentially unchanged (7.6% and 9.6%) while improving accuracy further, which is the expected signature of swaps acting on *which* trajectories reach the output rung rather than on termination behavior. The correction is therefore practically useful as well as theoretically necessary, and the two ingredients of PPT contribute through distinct mechanisms.

Communication Schedule: DEO versus Ordered Adjacent Sweeps Appendix E compares the ordered adjacent sweep (ADJ) with deterministic even–odd communication (DEO). Table 13 presents an empirical comparison between the schedules showing that, for the same finite iteration budget, ADJ consistently achieves higher accuracy than DEO across all evaluated models and

Table 13: Communication-schedule ablation: single-sample accuracy (%) of PPT with DEO versus the ordered adjacent sweep (ADJ, our default), under otherwise identical configurations. **Bold** marks the better schedule.

Schedule	MATH500	GPQA	HumanEval	AIME 24&25	LCB v5
Qwen 2.5 MATH					
PPT with DEO	77.8	32.3	57.6	15.7	11.3
PPT with ADJ (default)	**79.6**	**37.6**	**60.6**	**17.7**	**12.5**
Qwen3-4B					
PPT with DEO	84.7	53.0	90.2	73.3	53.8
PPT with ADJ (default)	**86.0**	**53.6**	**93.9**	**78.3**	**57.9**
Qwen3-8B					
PPT with DEO	85.9	57.1	90.2	75.7	55.3
PPT with ADJ (default)	**88.0**	**60.1**	**94.9**	**78.3**	**58.4**

benchmarks, suggesting that its greater exploration per iteration translates into improved predictive performance.