DECOUPLED MARGINAL SHARPENING FOR TRAINING-FREE INFERENCE-TIME SCALING

Anonymous authors
Paper under double-blind review

ABSTRACT

We study how to improve LLM reasoning at inference time without updating model weights or using external reward models. Recent reward-free power-sampling methods define the sequence-level power target $\pi_\alpha(y \mid x) \propto p_0(y \mid x)^\alpha$, where $\alpha > 1$, which increases the relative probability of completions that are already likely under the base model. Existing methods use Metropolis–Hastings or Sequential Monte Carlo (SMC) to approximately sample from this target. However, two challenges arise at different levels. At the target level, full-sequence sharpening can concentrate probability on a few exact reasoning–answer trajectories, suppressing alternative paths that support the same answer. At the sampler level, finite-particle SMC approximations can suffer from path collapse when resampling repeatedly duplicates a few high-weight prefixes. To address these challenges, we propose **Decoupled Marginal Sharpening (DMS)**, which separately controls how strongly the model favors particular reasoning paths and how strongly it favors final answers after aggregating support across those paths. At the target level, DMS first samples from a sharpened reasoning distribution, then marginalizes the model's answer probabilities over the resulting weighted reasoning population and sharpens the induced answer marginal. Separate exponents allow mild reasoning sharpening to encourage exploration of diverse paths, followed by stronger answer-marginal sharpening to concentrate probability on answers with greater aggregate support across those paths. At the sampler level, we develop Fully Adapted Annealed SMC, which combines an annealed sequence of power targets with locally fully adapted proposals and predictive ancestor resampling. Experiments on MATH500, GPQA-Diamond, HumanEval, and AIME24/25 across three base models show that DMS achieves the highest mean pass@1 among the compared methods in all tasks.

1 INTRODUCTION

Large language models can improve their reasoning performance through additional training (Zelikman et al., 2022; Ouyang et al., 2022; Hsieh et al., 2023; Shao et al., 2024; Zhang et al., 2026) or additional computation at inference time (Wang et al., 2022; Yao et al., 2023; Snell et al., 2024; Wang et al., 2026). We focus on inference-time scaling methods that keep the model fixed and use its own probabilities to guide generation, without additional model updates. These methods can be broadly divided into reward-based and reward-free approaches. Reward-based methods use an external reward model or verifier to guide generation toward higher-quality solutions (Zhao et al., 2024; Feng et al., 2025; Markovic-Voronov et al., 2026). Their effectiveness depends on the availability and accuracy of these signals (Zhu & Lu, 2026). Reward-free methods instead rely on signals already contained in the base model. A prominent approach is sequence-level power sampling, which defines the target distribution $\pi_\alpha(y \mid x) \propto p_0(y \mid x)^\alpha, \alpha > 1$, where p_0 is the base model, x is the prompt, and y is a complete generated sequence. This power transformation *sharpens* the base distribution by increasing the relative probability of more likely completions, making the distribution more concentrated. Sampling each token according to this target requires summing powered sequence probabilities over all possible future continuations, which is generally intractable. Existing methods approximate this sharpened distribution through Metropolis–Hastings updates to generated sequences (Karan & Du, 2025), token sampling guided by probabilities estimated from future continuations (Ji et al., 2026), or Sequential Monte Carlo (SMC) (Azizi et al., 2026).

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.

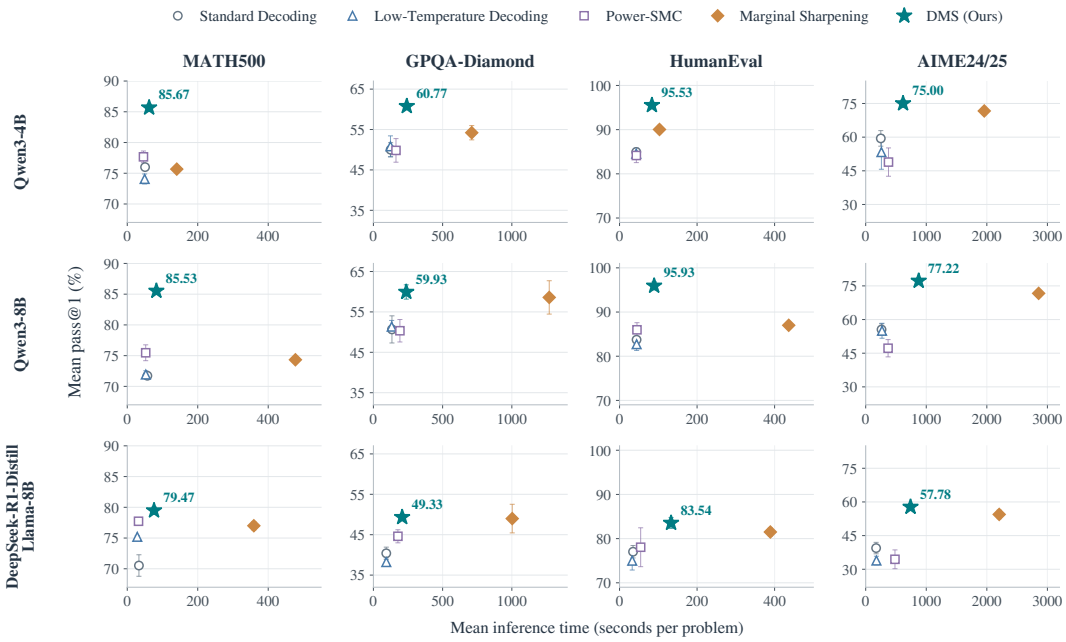

Figure 1: **Accuracy and inference latency across three models and four reasoning benchmarks.** Our method **DMS** achieves the highest accuracy in all settings while requiring less inference time than the strong baseline Marginal Sharpening. Vertical error bars indicate ± 1 standard deviation in pass@1. In each subplot, higher accuracy and lower latency are better (toward the upper left).

For mathematical and coding tasks, an LLM typically generates a reasoning path followed by a final answer. Sharpening the full sequence therefore favors individual reasoning–answer pairs with high joint likelihood. However, different reasoning paths can lead to the same correct answer, which may receive substantial combined support even when each individual path has relatively low probability. Sharpening these pairs separately can consequently reduce the probability of an answer supported by many such paths. Moreover, higher sequence likelihood does not necessarily imply correctness (Zenn & Geiping, 2026). Self-consistency and marginal sharpening account for support across reasoning paths (Wang et al., 2022; Arzhantsev et al., 2026); in particular, marginal sharpening aggregates conditional answer probabilities under the original reasoning distribution before sharpening the resulting answer marginal. We seek to retain this aggregation while also guiding reasoning toward more likely paths. The two stages thus call for different sharpening strengths: mild sharpening during reasoning favors more likely paths while retaining alternatives for aggregation, whereas stronger sharpening after aggregation concentrates probability on answers supported by their combined contributions.

To this end, we propose **Decoupled Marginal Sharpening (DMS)**, a training-free framework that separately controls sharpening at the reasoning and answer stages. Given a prompt x, let z denote a reasoning path and a denote its answer. DMS first sharpens the reasoning distribution, averages the model's conditional answer probabilities over the resulting distribution of reasoning paths, and then sharpens the resulting answer marginal:

$$\pi_Z(z|x) \propto p_0(z|x)^{\alpha_Z}, \quad m_{\alpha_Z}(a|x) = \textstyle\sum_z \pi_Z(z|x) p_0(a|z,x), \quad \pi_A(a|x) \propto m_{\alpha_Z}(a|x)^{\alpha_A}.$$

The reasoning exponent α_Z controls how much each reasoning path contributes to the answer marginal, while the answer exponent α_A controls how strongly the final distribution concentrates on answers with greater aggregate support. Choosing these exponents separately allows answer selection to become more concentrated without requiring equally strong sharpening of the reasoning distribution. In practice, we estimate the answer marginal from a finite population of reasoning paths. Repeated SMC resampling can duplicate a few high-weight prefixes and discard alternatives, causing path collapse and reducing the distinct reasoning paths available for aggregation. To address this sampling challenge, we develop Fully Adapted Annealed SMC (FAA-SMC), which combines

annealed power targets, predictive ancestor resampling, and locally fully adapted proposals. The resulting weighted reasoning population defines an empirical answer marginal. DMS samples answer candidates from this marginal, evaluates each complete candidate under all reasoning paths, and approximates the sharpened answer target.

Our contributions are summarized as follows:

- We propose **DMS**, a training-free two-stage marginal sharpening framework with separate control over reasoning-path concentration and answer-level consensus. DMS sharpens the reasoning distribution and then sharpens the answer marginal obtained by aggregating probabilities across reasoning paths, using different exponents for the two stages.
- We develop a practical sampling procedure for the two-stage target. FAA-SMC combines exact annealing bridges, predictive resampling, and locally fully adapted propagation to approximate the sharpened reasoning distribution. A finite-particle answer sampler then approximates the sharpened answer marginal through cross-reasoning likelihood evaluation.
- We evaluate DMS on MATH500, GPQA-Diamond, HumanEval, and AIME24/25 across three base models. DMS achieves the highest mean pass@1 among the compared methods in all twelve model–benchmark settings.

2 PRELIMINARIES

LLM Generation and Sequence-Level Power Sharpening. Given a problem prompt x, an autoregressive language model p_0 generates each token conditioned on the prompt and the preceding tokens. The probability of a sequence $y_{1:T} = (y_1, \dots, y_T)$ is therefore $p_0(y_{1:T} \mid x) = \prod_{t=1}^{T} p_0(y_t \mid x, y_{<t})$, where $y_{<t} = (y_1, \dots, y_{t-1})$. We use T as the maximum generation length. If generation ends before T, we pad the sequence with a fixed terminal symbol and assign conditional probability one to each padding step, leaving the sequence probability unchanged.

Huang et al. (2025) observe that correct responses tend to have higher likelihoods under the model and that selecting responses with higher likelihoods can improve accuracy, although the gains depend on the base model and task. These observations motivate *distribution sharpening*: keeping the model parameters fixed while changing the sampling distribution to favor responses that the model already assigns higher probability.

For training-free LLM reasoning, Karan & Du (2025) propose sampling from a sequence-level power distribution that implements this principle while keeping the base model fixed:

$$\pi_\alpha(y_{1:T} \mid x) = \mathcal{Z}_\alpha(x)^{-1} p_0(y_{1:T} \mid x)^\alpha, \qquad \alpha \geq 1, \tag{2.1}$$

where $\mathcal{Z}_\alpha(x) = \sum_{y_{1:T}} p_0(y_{1:T} \mid x)^\alpha$ is the normalizing constant. When $\alpha = 1$, this target recovers the base model distribution. For $\alpha > 1$, it favors outputs the model is more confident in, as reflected by their higher likelihood.

To sample a sequence from π_α one token at a time, we need the conditional probability of each next token under π_α. Computing these probabilities requires summing $p_0(y_{1:T} \mid x)^\alpha$ over all possible future continuations, which is generally intractable. Simply lowering the temperature of the base model's next-token distribution does not account for the powered probability mass of these continuations and therefore does not generally produce samples from π_α. We therefore use sequential Monte Carlo (SMC), which generates tokens from tractable proposal distributions and uses importance weights to correct for the difference between the proposal and target distributions.

Sequential Monte Carlo. SMC approximates the target distribution using K particles, each consisting of a generated prefix and an importance weight (Zhao et al., 2024; Azizi et al., 2026). At step t, the weighted particles approximate the distribution proportional to $\gamma_t(y_{1:t} \mid x)$, where $\{\gamma_t\}_{t=0}^{T}$ is a sequence of unnormalized intermediate targets satisfying

$$\gamma_0(\varnothing \mid x) = 1, \qquad \gamma_T(y_{1:T} \mid x) = p_0(y_{1:T} \mid x)^\alpha.$$

The normalized terminal target is therefore π_α in (2.1).

All particles start from the empty prefix with equal weights $w_0^k = 1$. At each step, a proposal distribution $q_t(\cdot \mid x, y_{<t})$ specifies how to sample the next token. To account for the proposal

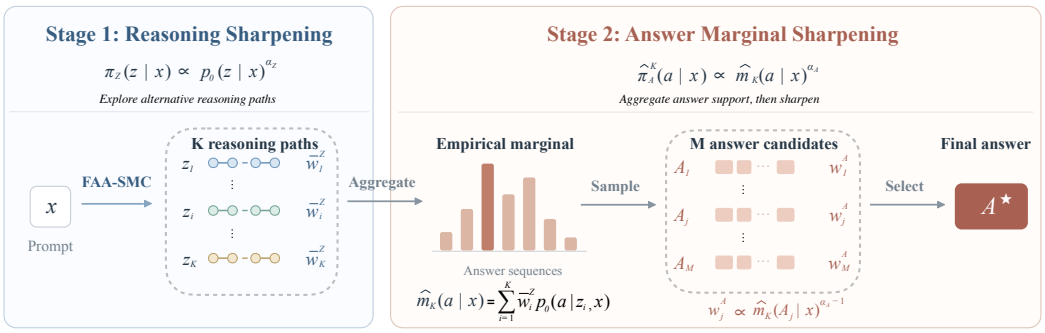

Figure 2: Overview of DMS. FAA-SMC constructs K weighted reasoning paths that define an empirical answer marginal. We sample M candidates from this marginal and select the final answer according to their normalized answer-sharpening weights.

probability and the change from γ_{t-1} to γ_t, we use the incremental importance weight

$$G_t(y_{1:t}) = \frac{\gamma_t(y_{1:t} \mid x)}{\gamma_{t-1}(y_{<t} \mid x) q_t(y_t \mid x, y_{<t})}. \tag{2.2}$$

Particle k is then extended and reweighted according to

$$Y_t^k \sim q_t(\cdot \mid x, Y_{<t}^k), \qquad w_t^k = w_{t-1}^k G_t(Y_{1:t}^k).$$

To determine when to resample, we normalize the weights and compute the effective sample size:

$$\bar{w}_t^k = \frac{w_t^k}{\sum_{\ell=1}^K w_t^\ell}, \qquad \mathrm{ESS}_t = \frac{1}{\sum_{k=1}^K (\bar{w}_t^k)^2}.$$

A small ESS indicates that a few particles carry most of the total weight. When $\mathrm{ESS}_t < \tau K$ for a threshold $\tau \in (0, 1]$, SMC selects K prefixes according to their normalized weights, allowing the same prefix to be selected multiple times. The selected prefixes form a new particle population with equal weights, and generation continues separately from each selected prefix. At termination, the weighted population $\{(Y_{1:T}^k, \bar{w}_T^k)\}_{k=1}^K$ approximates π_α.

With a finite particle budget, repeated resampling can lead to *path collapse*: high-weight prefixes are repeatedly copied while other prefixes are discarded, leaving many particles descended from only a few earlier prefixes. Although resampling resets the weights, it does not recover the discarded paths. This loss of reasoning-path diversity motivates the sampler design introduced in our methodology.

3 Methodology: Decoupled Marginal Sharpening

We propose **Decoupled Marginal Sharpening (DMS)**, which first sharpens the reasoning distribution and then sharpens the answer marginal constructed from it. To approximate these targets, we develop Fully Adapted Annealed SMC (FAA-SMC) to generate a weighted population of reasoning paths. We sample answer candidates from the resulting empirical marginal, evaluate each candidate under all paths, and reweight them for final selection. Figure 2 illustrates the overall procedure.

3.1 Two-Stage Sharpened Target

We write the complete generation as $Y = (Z, A)$, where Z is the reasoning path and A is the complete answer token sequence generated after it, such as a mathematical response or a code completion. The base model distribution factorizes as $p_0(z, a \mid x) = p_0(z \mid x) p_0(a \mid z, x)$.

Sequence-level power sharpening applies the same exponent to both factors: $p_0(z, a \mid x)^\alpha = p_0(z \mid x)^\alpha p_0(a \mid z, x)^\alpha$. This ties reasoning and conditional answer sharpening to the same strength. However, reasoning and answer selection serve different purposes. During reasoning, we want to favor

likely paths while retaining alternatives for exploration; when selecting the final answer, we want to concentrate on candidates with strong model support. We therefore introduce separate exponents, allowing mild or no additional reasoning sharpening together with stronger answer sharpening.

Reasoning-Level Sharpening. We first define the reasoning target with its own exponent:

$$\pi_Z(z \mid x) = \mathcal{Z}_Z(x)^{-1} p_0(z \mid x)^{\alpha_Z}, \qquad \alpha_Z \geq 1, \tag{3.1}$$

where $\mathcal{Z}_Z(x) = \sum_z p_0(z \mid x)^{\alpha_Z}$ is the normalizing constant. Setting $\alpha_Z = 1$ retains the base reasoning distribution, while increasing α_Z gives greater relative probability to more likely reasoning paths. Choosing this exponent independently lets us control the preference for these paths during reasoning exploration without tying it to the strength used for answer selection.

Reasoning-Induced Answer Marginal. With the reasoning distribution in (3.1) specified, we next construct the corresponding answer marginal. Marginalization combines support for the same answer across different reasoning paths, so that subsequent answer sharpening acts on their total contribution. We define the answer marginal distribution as follows,

$$m_{\alpha_Z}(a \mid x) = \sum_z \pi_Z(z \mid x) p_0(a \mid z, x).$$

The marginal measures the combined support for this answer under chosen reasoning distribution.

Marginal Answer Sharpening. We then sharpen the answer marginal with a separate exponent:

$$\pi_A(a \mid x) = \mathcal{Z}_A(x)^{-1} m_{\alpha_Z}(a \mid x)^{\alpha_A}, \qquad \alpha_A \geq 1, \tag{3.2}$$

where $\mathcal{Z}_A(x) = \sum_a m_{\alpha_Z}(a \mid x)^{\alpha_A}$ is the normalizing constant. For a fixed α_Z, increasing α_A gives greater relative probability to answers with larger marginal probability while preserving their ranking under m_{α_Z}. The final answer distribution can therefore be sharpened more strongly while the reasoning distribution remains fixed.

Remark 3.1. The order of marginalization and answer sharpening matters. Under sequence-level sharpening, the probability assigned to an answer is proportional to $\sum_z p_0(z, a \mid x)^{\alpha}$, so each trajectory is sharpened before contributions to the same answer are combined. This can make a single likely trajectory outweigh several less likely trajectories even when their combined probability was originally greater, reversing the ranking of their associated answers.[1] In DMS, answer sharpening acts on the combined probability already obtained under π_Z. Consequently, even setting $\alpha_Z = \alpha_A > 1$ does not generally recover the answer marginal of sequence-level sharpening. When $\alpha_Z = 1$, we have $m_1(a \mid x) = p_0(a \mid x)$, and the ideal target in (3.2) reduces to marginal sharpening under the original reasoning distribution (Arzhantsev et al., 2026).

To approximate the target in (3.2), we first estimate m_{α_Z} using a finite weighted population of reasoning paths. We next describe FAA-SMC, a stable and efficient sampler designed to mitigate path collapse when constructing this population.

3.2 Reasoning Sampling with FAA-SMC

FAA-SMC approximates the reasoning target in (3.1) with final exponent $\alpha = \alpha_Z$. It combines two mechanisms: annealing introduces sharpening gradually during generation, while a resample–propagate update allows copies of a selected prefix to generate their next tokens independently.

Annealed Intermediate Targets. Applying the final sharpening strength from the start can concentrate particles before alternative reasoning paths have been explored. We therefore use a nondecreasing schedule $1 = \beta_0 \leq \beta_1 \leq \cdots \leq \beta_T = \alpha$ and define

$$\gamma_t(y_{1:t} \mid x) = p_0(y_{1:t} \mid x)^{\beta_t}. \tag{3.3}$$

[1] Consider a simple example: suppose three trajectories leading to answer a each have probability 0.2, while one trajectory leading to answer b has probability 0.4. Their base answer probabilities are 0.6 and 0.4. With $\alpha = 2$, sequence-level sharpening gives unnormalized answer masses 0.12 and 0.16, reversing their ranking. Sharpening the base answer marginal instead gives 0.36 and 0.16.

Under these targets, the incremental importance weight in (2.2) becomes

$$G_t(y_{1:t}) = p_0(y_{<t} \mid x)^{\beta_t - \beta_{t-1}} \frac{p_0(y_t \mid x, y_{<t})^{\beta_t}}{q_t(y_t \mid x, y_{<t})}. \tag{3.4}$$

The prefix factor accounts for the change in exponent applied to tokens already generated. Since $\beta_T = \alpha$, (3.3) ends at the specified power target. Annealing therefore changes how sharpening is introduced during generation while preserving the final target.

Fully Adapted Resample–Propagate. We next choose a proposal that allows the importance correction to be computed before generating the next token. Let $h_t = (x, y_{<t})$ denote the current context. We use the locally fully adapted proposal

$$q_t^{\mathrm{FA}}(v \mid h_t) = \frac{p_0(v \mid h_t)^{\beta_t}}{\mathcal{Z}_t(h_t)},$$

where $\mathcal{Z}_t(h_t) = \sum_{v \in \mathcal{V}} p_0(v \mid h_t)^{\beta_t}$ and $\mathcal{V}$ is the token vocabulary. Substituting this proposal into (3.4) cancels the dependence on the sampled token:

$$G_t(y_{1:t}) = M_t(h_t), \qquad M_t(h_t) = p_0(y_{<t} \mid x)^{\beta_t - \beta_{t-1}} \mathcal{Z}_t(h_t).$$

The remaining correction depends only on the current prefix. We can therefore compute the predictive weights $\widetilde{w}_{t-1}^k = w_{t-1}^k M_t(h_t^k)$ and use them to select ancestor prefixes before propagation.

At resampling checkpoints, we resample prefixes according to their normalized predictive weights if $\mathrm{ESS} < \tau K$ and reset their weights to equal values. Each prefix then independently samples its next token from q_t^{FA}, with no further weight correction since M_t already accounts for it. Resampling before propagation allows copies of the same prefix to generate different next tokens instead of inheriting the same newly sampled token. Algorithm 2 in Appendix B summarizes the procedure.

Remark 3.2 (Comparison with propagate–resample SMC)**.** Consider a prefix selected twice at a resampling step. A propagate–resample update in standard SMC copies the same newly generated token into both particles. FAA-SMC instead selects the prefix first and then draws a next token independently for each copy, allowing the copies to follow different continuations immediately. For example, if two tokens each have probability $1/2$ under q_t^{FA}, two copies of the same prefix independently draw different tokens with probability $1/2$.

Island-DMS. On difficult problems requiring long reasoning paths, repeated resampling can lead to more severe path collapse, especially when the particle population is small. In extreme cases, all final particles may descend from only one or two initial particles. To preserve multiple particle diversity, we adopt an island variant (Vergé et al., 2015). We divide the total population into L independent islands of K_{island} particles, with $K = LK_{\mathrm{island}}$. Each island runs FAA-SMC and resamples only within its own population. Since each island retains at least one initial-particle ancestor, pooling all islands guarantees at least L distinct ancestors, even if all particles within each island share the same ancestor.

We retain all final reasoning paths and assign equal total weight to each island. If $\bar{w}_{\ell,k}^Z$ is the normalized weight of particle k within island ℓ, its pooled weight is $\bar{w}_{\ell,k}^{Z,\mathrm{island}} = \bar{w}_{\ell,k}^Z / L$. This pooled population defines the empirical answer marginal used in the next subsection.

3.3 Sampling from the Sharpened Answer Marginal

Let $\mathcal{B}_K^Z = \{(z_i, \bar{w}_i^Z)\}_{i=1}^K$ denote the weighted reasoning population returned by FAA-SMC, with $\sum_{i=1}^K \bar{w}_i^Z = 1$. For Island-DMS, we use the pooled population defined above. Each path ends at the reasoning–answer boundary and provides the context for both answer generation and scoring. We replace the sum over all reasoning paths in m_{α_Z} with the weighted sum over this population:

$$\widehat{m}_K(a \mid x) = \textstyle\sum_{i=1}^K \bar{w}_i^Z p_0(a \mid z_i, x). \tag{3.5}$$

This empirical marginal defines the answer target $\widehat{\pi}_A^K(a \mid x) \propto \widehat{m}_K(a \mid x)^{\alpha_A}$, which approximates the ideal target in (3.2). We approximate sampling from this target by generating candidates from $\widehat{m}_K$ and correcting the difference through importance weighting.

Algorithm 1 DMS: Decoupled Marginal Sharpening

Require: Prompt x, base model p_0, reasoning particle count K, answer candidate count M, exponents α_Z, α_A, reasoning schedule $1 = \beta_0 \le \cdots \le \beta_T = \alpha_Z$, resampling threshold τ, and resampling checkpoints.

1: Run FAA-SMC in Algorithm 2 over the reasoning segment to obtain $\mathcal{B}_K^Z = \{(z_i, \bar{w}_i^Z)\}_{i=1}^K$.
2: **for** $j = 1, \dots, M$ **do**
3: Sample $I_j \sim \mathrm{Cat}(\bar{w}_{1:K}^Z)$.
4: Generate a complete answer $A_j \sim p_0(\cdot \mid z_{I_j}, x)$.
5: Score A_j under all K reasoning paths and compute $\widehat{m}_K(A_j \mid x)$ using (3.5).
6: Compute w_j^A using (3.6).
7: **end for**
8: Normalize the answer weights to obtain $\bar{w}_{1:M}^A$.
9: Sample $J^\star \sim \mathrm{Cat}(\bar{w}_{1:M}^A)$ and **return** $A^\star = A_{J^\star}$.

Generating Answer Candidates. Since $\widehat{m}_K$ is a mixture, we can sample from it by first selecting a reasoning path according to its weight and then generating an answer conditioned on that path. Repeating these two draws independently produces M candidates:

$$I_j \sim \mathrm{Cat}(\bar{w}_{1:K}^Z), \qquad A_j \sim p_0(\cdot \mid z_{I_j}, x), \qquad j = 1, \dots, M.$$

Each candidate therefore follows $\widehat{m}_K$, even though its generation uses only one reasoning path. This mixture serves as the proposal distribution for answer sharpening.

Evaluating Answer Marginal Probabilities. Generating a candidate requires only one selected path, but evaluating its marginal probability requires contributions from the entire reasoning population. We therefore score the same complete answer A_j under every retained context (x, z_i) and combine the conditional probabilities $p_0(A_j \mid z_i, x)$ using (3.5). This allows paths other than the one used for generation to contribute to the candidate's probability for final selection. Additional detail of derivation is described in Appendix B.

Answer Reweighting and Selection. The proposal is $\widehat{m}_K$, whereas the unnormalized answer target is $\widehat{m}_K^{\alpha_A}$. Their ratio gives the importance weight

$$w_j^A = \frac{\widehat{m}_K(A_j \mid x)^{\alpha_A}}{\widehat{m}_K(A_j \mid x)} = \widehat{m}_K(A_j \mid x)^{\alpha_A - 1}. \tag{3.6}$$

Candidate generation already samples answers in proportion to their marginal probabilities, so the correction uses only the remaining power $\alpha_A - 1$. We normalize the weights as $\bar{w}_j^A = w_j^A / \sum_{\ell=1}^M w_\ell^A$, draw $J^\star \sim \mathrm{Cat}(\bar{w}_{1:M}^A)$, and return $A^\star = A_{J^\star}$.

Conditional on the reasoning population, this sampling-importance-resampling procedure approximates $\widehat{\pi}_A^K$ using M candidates. Algorithm 1 summarizes DMS.

4 Experiments

4.1 Experimental Setup

Models and baselines. We evaluate DMS using Qwen3-4B, Qwen3-8B (Yang et al., 2025), and DeepSeek-R1-Distill-Llama-8B (Guo et al., 2025). The Qwen3 models use their native chat templates with thinking mode enabled. We compare against standard decoding, low-temperature decoding, Power-SMC (Azizi et al., 2026), and Marginal Sharpening (Arzhantsev et al., 2026). Power-SMC targets a sharpened distribution over complete sequences, while Marginal Sharpening sharpens the answer marginal induced by the original reasoning distribution.

Benchmarks and evaluation. We evaluate on MATH500 (500 problems) (Hendrycks et al., 2021; Lightman et al., 2024), GPQA-Diamond (198 questions) (Rein et al., 2023), HumanEval (164 problems) (Chen et al., 2021), and AIME24/25 (60 problems from AIME I and II in both years).

Table 1: Pass@1 (%) across models and benchmarks. Shaded rows denote DMS, and boldface indicates the highest mean within each model–benchmark setting. The DMS results on AIME24/25 use the Island-DMS variant.

Model	Method	MATH500	GPQA-Diamond	HumanEval	AIME24/25
Qwen3-4B	Standard Decoding	76.00 ± 0.53	50.00 ± 1.75	84.96 ± 0.35	59.44 ± 3.47
	Low-Temperature Decoding	74.07 ± 0.81	50.84 ± 2.59	84.55 ± 1.27	53.33 ± 7.64
	Power-SMC	77.67 ± 0.95	49.83 ± 2.92	84.15 ± 1.61	48.89 ± 6.31
	Marginal Sharpening	75.67 ± 0.23	54.21 ± 1.77	90.04 ± 0.93	71.67 ± 1.67
	DMS (Ours)	$\mathbf{85.67 \pm 0.58}$	$\mathbf{60.77 \pm 0.77}$	$\mathbf{95.53 \pm 0.70}$	$\mathbf{75.00 \pm 1.67}$
Qwen3-8B	Standard Decoding	71.73 ± 0.58	50.67 ± 3.36	83.74 ± 0.35	55.56 ± 2.55
	Low-Temperature Decoding	71.93 ± 0.61	51.35 ± 1.54	82.72 ± 1.41	55.00 ± 3.33
	Power-SMC	75.47 ± 1.29	50.34 ± 2.78	85.98 ± 1.61	47.22 ± 3.85
	Marginal Sharpening	74.33 ± 0.12	58.59 ± 4.13	86.99 ± 0.70	71.67 ± 1.67
	DMS (Ours)	$\mathbf{85.53 \pm 0.42}$	$\mathbf{59.93 \pm 1.77}$	$\mathbf{95.93 \pm 0.93}$	$\mathbf{77.22 \pm 0.96}$
DeepSeek-R1-Distill-Llama-8B	Standard Decoding	70.53 ± 1.75	40.40 ± 1.52	77.03 ± 1.41	39.44 ± 2.55
	Low-Temperature Decoding	75.20 ± 0.20	38.22 ± 1.05	75.00 ± 2.11	33.89 ± 1.92
	Power-SMC	77.73 ± 0.42	44.61 ± 1.62	78.05 ± 4.40	34.44 ± 4.19
	Marginal Sharpening	77.00 ± 0.69	48.99 ± 3.54	81.50 ± 0.93	54.44 ± 1.92
	DMS (Ours)	$\mathbf{79.47 \pm 0.61}$	$\mathbf{49.33 \pm 0.77}$	$\mathbf{83.54 \pm 1.22}$	$\mathbf{57.78 \pm 0.96}$

These benchmarks cover mathematical reasoning, scientific question answering, and code generation. Mathematical predictions are extracted from the final boxed expression and graded using normalized exact matching and symbolic equivalence. GPQA-Diamond uses exact option matching, while HumanEval completions are evaluated against the original unit tests in isolated processes with a three-second timeout.

Sampling protocol. DMS uses $K = 16$ reasoning particles and $M = 32$ answer candidates per problem. We use standard DMS on MATH500, GPQA-Diamond, and HumanEval, and Island-DMS with eight islands of two particles on AIME24/25. Candidates are sampled from the weighted reasoning mixture and cross-scored under all K paths to estimate their marginal probabilities and compute answer-level importance weights. Generation budgets, sharpening exponents, and implementation details are provided in Appendix C.

Metrics. Our primary metric is pass@1 of the final returned answer. For DMS, one answer is sampled categorically from the M candidates according to their normalized importance weights. We report the mean and sample standard deviation over three random seeds, $\{10080, 10081, 10082\}$.

4.2 Main Results

Table 1 reports pass@1 across three models and four benchmarks. DMS achieves the highest mean accuracy in all twelve settings, improving over the strongest baseline by 0.34–10.07 percentage points. On MATH500, DMS reaches 85.67% and 85.53% with Qwen3-4B and Qwen3-8B, exceeding Power-SMC by 8.00 and 10.07 points, respectively. On HumanEval, it achieves 95.53% and 95.93%, improving over Marginal Sharpening by 5.49 and 8.94 points.

Standard DMS achieves the highest mean accuracy on MATH500, GPQA-Diamond, and HumanEval for all three models. It also exceeds the strongest baseline on AIME24/25, reaching 72.78%, 74.44%, and 55.00% with Qwen3-4B, Qwen3-8B, and DeepSeek-R1-Distill-Llama-8B, respectively. Island-DMS further raises these results to 75.00%, 77.22%, and 57.78%, adding 2.22–2.78 percentage points while retaining the same total reasoning particle count.

The strongest baseline varies across tasks: Power-SMC performs best on MATH500, while Marginal Sharpening leads on the other three benchmarks. DMS achieves the highest mean pass@1 in each setting. Its comparison with Marginal Sharpening is particularly relevant because this baseline already aggregates answer probabilities across reasoning paths. The results support the complete DMS procedure, which combines a separately controlled reasoning distribution with marginal-probability weighting of complete answer candidates. DMS also has lower measured inference latency than Marginal Sharpening in all twelve settings; timing details are provided in Appendix G.

4.3 Ablation Study on AIME24/25

We examine the contributions of the two sharpening stages and island sampling on AIME24/25 using Qwen3-8B. All two-stage variants share the same prompts, grader, and 16,384-token limit per reasoning–answer trajectory, with $K = 16$ reasoning particles and $M = 32$ answer candidates. In Table 2, the reference DMS configuration uses $(\alpha_Z, \alpha_A) = (1.005, 4)$. Setting either exponent to one disables the corresponding sharpening operation; setting both to one gives structured decoding without sharpening. Island-DMS keeps the exponents and total particle count fixed while dividing the reasoning population into eight independent islands of two particles.

Table 2: Component analysis on AIME24/25 using Qwen3-8B. Expected accuracy is the probability of selecting a correct answer under the normalized answer weights, averaged over problems. Pass@1 evaluates one sampled answer per problem, while oracle pass@32 is the fraction of problems with at least one correct answer among the 32 candidates. All accuracy metrics are percentages.

Variant	α_Z	α_A	Expected	Pass@1	Oracle pass@32
Structured decoding only	1	1	66.48	67.78	**82.78**
Answer sharpening only	1	4	**76.89**	**77.22**	**82.78**
Reasoning sharpening only	1.005	1	71.77	72.78	78.33
Standard DMS	1.005	4	74.55	74.44	78.33
Island-DMS	1.005	4	76.83	**77.22**	81.67

With unsharpened reasoning ($\alpha_Z = 1$), increasing α_A from 1 to 4 raises expected accuracy from 66.48% to 76.89%. At $\alpha_A = 1$, mild reasoning sharpening raises average candidate correctness from 66.48% to 71.77%, but reduces oracle pass@32 from 82.78% to 78.33%. Reasoning sharpening therefore improves average candidate correctness while reducing correct-answer coverage in this setting. At fixed exponents and particle count, Island-DMS raises expected accuracy from 74.55% to 76.83% and oracle pass@32 from 78.33% to 81.67%. However, answer sharpening alone achieves similar expected accuracy (76.89%), and both obtain 77.22% pass@1. Thus, additional reasoning sharpening is not required for the best observed pass@1 in this setting, supporting independent choices of the two exponents.

Due to space limit, we defer additional experiments to the appendix. In particular, we report sensitivity to the two sharpening exponents in Appendix D and ablations of annealing and resampling in Appendix E. We also provide a diagnostic using a higher-budget reasoning sampler in Appendix F, together with inference latency comparison and a breakdown of DMS runtime in Appendix G.

5 Conclusion

We studied training-free inference-time scaling for LLM reasoning, aiming to improve answer accuracy without updating model weights or relying on external reward models. Sequence-level power sampling faces both sampler-level and target-level challenges: finite-particle SMC can suffer from path collapse, while full-sequence sharpening can suppress alternative reasoning paths that support the same answer. We proposed **DMS**, a two-stage marginal sharpening framework with separate sampler-level and target-level designs. At the sampler level, FAA-SMC combines annealed power targets, predictive resampling, and locally fully adapted proposals. At the target level, DMS sharpens the reasoning distribution, marginalizes answer probabilities over the resulting reasoning population, and then sharpens the induced answer marginal. Separate exponents control reasoning-path concentration and answer-level consensus, allowing answers to be weighted according to their aggregate support across reasoning paths. Experiments on MATH500, GPQA-Diamond, HumanEval, and AIME24/25 across three base models show that DMS achieves the highest mean pass@1 among the compared methods in all twelve settings, including comparisons with Power-SMC and Marginal Sharpening. Future work includes improving answer-marginal estimation, reducing particle degeneracy during reasoning, and developing adaptive choices of sharpening strengths and inference-time budgets.

AI USE STATEMENT

We used generative AI tools to assist with manuscript revision and polishing, as well as code implementation and debugging. We also used these tools to summarize experimental results and prepare tables. The authors reviewed the AI-assisted material and take responsibility for the final content of this work, including its text, code, and experimental results.

REPRODUCIBILITY STATEMENT

We provide code for reproducing the reported experiments in the supplementary material. Section 3 describes the algorithms, and Appendix B provides additional derivations and numerical implementation details. Section 4 and Appendix C document the models, benchmarks, generation budgets, evaluation protocols, hyperparameter settings, random seeds, and hardware configuration. Appendices D–G provide the settings and results for the sensitivity analyses, sampler ablations, higher-budget reasoning-sampler diagnostic, and time-cost analysis.

REFERENCES

Aleksei Arzhantsev, Otmane Sakhi, and Nicolas Chopin. Self-consistency via marginal sharpening. *arXiv preprint arXiv:2605.28142*, 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. *arXiv preprint arXiv:2602.10273*, 2026.

Robin Shing Moon Chan, Tianyu Liu, Samuel Kiegeland, Clemente Pasti, Jacob Hoover Vigly, Timothy J O'Donnell, Ryan Cotterell, and Tim Vieira. Ensembling language models with sequential monte carlo. *arXiv preprint arXiv:2603.05432*, 2026.

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

Yun-Chen Cheng, Che-Yu Lin, and Cheng-Lin Yang. Score × decoder: A unified view of unsupervised inference-time scaling for hallucination mitigation. *arXiv preprint arXiv:2606.00739*, 2026.

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. URL https://openreview.net/forum?id=Ze4aPP0tIn.

Pieralberto Guarniero, Adam M Johansen, and Anthony Lee. The iterated auxiliary particle filter. *Journal of the American Statistical Association*, 112(520):1636–1647, 2017.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

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.

Cheng-Yu Hsieh, Chun-Liang Li, Chih-Kuan Yeh, Hootan Nakhost, Yasuhisa Fujii, Alex Ratner, Ranjay Krishna, Chen-Yu Lee, and Tomas Pfister. Distilling step-by-step! outperforming larger language models with less training data and smaller model sizes. In *Findings of the association for computational linguistics: ACL 2023*, pp. 8003–8017, 2023.

Audrey Huang, Adam Block, Dylan Foster, Dhruv Rohatgi, Cyril Zhang, Max Simchowitz, Jordan Ash, and Akshay Krishnamurthy. Self-improvement in language models: The sharpening mechanism. In *International Conference on Learning Representations*, volume 2025, pp. 76687–76739, 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.

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

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

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

Long Ouyang, Jeffrey Wu, Xu Jiang, Diogo Almeida, Carroll Wainwright, Pamela Mishkin, Chong Zhang, Sandhini Agarwal, Katarina Slama, Alex Ray, et al. Training language models to follow instructions with human feedback. *Advances in neural information processing systems*, 35: 27730–27744, 2022.

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. *arXiv preprint arXiv:2311.12022*, 2023.

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

Charlie Snell, Jaehoon Lee, Kelvin Xu, and Aviral Kumar. Scaling llm test-time compute optimally can be more effective than scaling model parameters. *arXiv preprint arXiv:2408.03314*, 2024.

Christelle Vergé, Cyrille Dubarry, Pierre Del Moral, and Eric Moulines. On parallel implementation of sequential monte carlo methods: the island particle model. *Statistics and Computing*, 25(2): 243–260, 2015.

Weixin Wang, Yu Yang, Wei Deng, and Pan Xu. Inference-time alignment of diffusion models via trust-region iterative twisted sequential monte carlo. *arXiv preprint arXiv:2605.25123*, 2026.

Xuezhi Wang, Jason Wei, Dale Schuurmans, Quoc Le, Ed Chi, Sharan Narang, Aakanksha Chowdhery, and Denny Zhou. Self-consistency improves chain of thought reasoning in language models. *arXiv preprint arXiv:2203.11171*, 2022.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025.

Shunyu Yao, Dian Yu, Jeffrey Zhao, Izhak Shafran, Tom Griffiths, Yuan Cao, and Karthik Narasimhan. Tree of thoughts: Deliberate problem solving with large language models. *Advances in neural information processing systems*, 36:11809–11822, 2023.

Eric Zelikman, Yuhuai Wu, Jesse Mu, and Noah Goodman. Star: Bootstrapping reasoning with reasoning. *Advances in Neural Information Processing Systems*, 35:15476–15488, 2022.

Johannes Zenn and Jonas Geiping. When are likely answers right? on sequence probability and correctness in llms. *arXiv preprint arXiv:2606.27359*, 2026.

Ruixiang Zhang, Richard He Bai, Huangjie Zheng, Navdeep Jaitly, Ronan Collobert, and Yizhe Zhang. Embarrassingly simple self-distillation improves code generation. *arXiv preprint arXiv:2604.01193*, 2026.

Stephen Zhao, Rob Brekelmans, Alireza Makhzani, and Roger Grosse. Probabilistic inference in language models via twisted sequential monte carlo. *arXiv preprint arXiv:2404.17546*, 2024.

Youheng Zhu and Yiping Lu. On the power of (approximate) reward models for inference-time scaling. *arXiv preprint arXiv:2602.01381*, 2026.

A Related Work

Inference-Time Scaling for LLM Reasoning. Inference-time scaling improves LLM reasoning by allocating additional computation during decoding while keeping the base model fixed. Existing methods can be broadly divided into reward-based and reward-free approaches. Reward-based methods use an external reward or verifier to define a reward-tilted sequence distribution. Building on auxiliary and twisted particle filtering (Guarniero et al., 2017), Zhao et al. (2024) formulate language-model inference as sampling from unnormalized sequence distributions and introduce twisted SMC with learned estimates of future potential. Feng et al. (2025) apply this framework to mathematical reasoning, while Markovic-Voronov et al. (2026) develop a general reward-guided SMC decoder with prefix-based and look-ahead intermediate targets. Zhu & Lu (2026) further study how errors in approximate reward models affect inference-time scaling.

Reward-free methods instead use signals already contained in the base model, such as sequence likelihood or agreement among sampled solutions. One line of work uses sequence-level power sampling, which defines the target $\pi_\alpha(y \mid x) \propto p_0(y \mid x)^\alpha$ for $\alpha > 1$. Karan & Du (2025) approximately sample from this target using an iterative Metropolis–Hastings-style procedure. Ji et al. (2026) improve sampling efficiency by approximating the future correction associated with the sequence-level power objective during autoregressive decoding. Sequential Monte Carlo provides a parallel alternative by maintaining and resampling a population of weighted partial generations. Azizi et al. (2026) propose Power-SMC, which uses token-level importance corrections, adaptive resampling, and exponent bridging to approximate the sequence-level power target. Nguyen et al. (2026) further improve particle allocation using short-horizon rollouts or an amortized estimate of future probability mass. Related reward-free SMC methods also consider sampling from ensemble distributions constructed from multiple language models or prompts (Chan et al., 2026). Complementary studies examine how model-intrinsic scores interact with sampling, optimization, and consensus-based decoders (Cheng et al., 2026), and show that increasing sequence likelihood does not necessarily improve reasoning accuracy (Zenn & Geiping, 2026).

Beyond sampling individual sequences, reward-free methods can aggregate support across reasoning paths at the answer level. Self-consistency samples multiple reasoning trajectories and selects the answer with the greatest aggregate support (Wang et al., 2022). Arzhantsev et al. (2026) formulate self-consistency as marginal sharpening over final answers, constructing the answer marginal under the original reasoning distribution. DMS extends this formulation by allowing the reasoning distribution used to construct the marginal to be sharpened independently of the subsequent answer sharpening.

B Additional Details for DMS

Algorithm 2 summarizes FAA-SMC. DMS applies this sampler to the reasoning segment with terminal exponent $\alpha = \alpha_Z$.

Target Preservation under Annealing. Let $Q(y_{1:T} \mid x) = \prod_{t=1}^{T} q_t(y_t \mid x, y_{<t})$ denote the sequence proposal obtained from the token-level proposal kernels. Along any fixed sequence with positive proposal probability, the incremental importance ratios telescope:

$$\prod_{t=1}^{T} G_t(y_{1:t}) = \frac{\gamma_T(y_{1:T} \mid x)}{\gamma_0(\varnothing \mid x) Q(y_{1:T} \mid x)} = \frac{p_0(y_{1:T} \mid x)^\alpha}{Q(y_{1:T} \mid x)}.$$

Thus, annealing changes the intermediate targets while preserving the specified final power target. For a terminated particle, subsequent steps append the fixed terminal symbol with probability one, so $\mathcal{Z}_t(h_t) = 1$. Its incremental corrections then reduce to $p_0(y_{<t} \mid x)^{\beta_t - \beta_{t-1}}$, which must still be applied until the final exponent is reached.

Augmented Representation of the Answer Target. The answer target in (3.2) admits the joint representation

$$\widetilde{\pi}_A(z, a \mid x) = \mathcal{Z}_A(x)^{-1} \pi_Z(z \mid x) p_0(a \mid z, x) m_{\alpha_Z}(a \mid x)^{\alpha_A - 1}.$$

Algorithm 2 FAA-SMC: Fully Adapted Annealed SMC

Require: Prompt x, base model p_0, exponent schedule $1 = \beta_0 \leq \cdots \leq \beta_T = \alpha$, particle count K, resampling threshold τ, and resampling checkpoints.

Initialize $Y_{1:0}^k = \varnothing$ and $w_0^k = 1$ for $k = 1, \ldots, K$.
for $t = 1, \ldots, T$ **do**
  Set $h_t^k = (x, Y_{<t}^k)$ and compute $q_t^{\mathrm{FA}}(\cdot \mid h_t^k)$ and $M_t(h_t^k)$ for all k.
  Compute $\widetilde{w}_{t-1}^k = w_{t-1}^k M_t(h_t^k)$ and normalize to obtain $\widetilde{\bar{w}}_{t-1}^{1:K}$.
  if t is a resampling checkpoint and $\mathrm{ESS}(\widetilde{\bar{w}}_{t-1}^{1:K}) < \tau K$ **then**
    Resample the prefixes and their associated proposals according to $\widetilde{\bar{w}}_{t-1}^{1:K}$, and set $w_t^k = 1$ for all k.
  else
    Set $w_t^k = \widetilde{w}_{t-1}^k$ for all k.
  end if
  For prefixes that terminated before step t, set Y_t^k to the fixed terminal symbol.
  For all other particles, independently sample $Y_t^k \sim q_t^{\mathrm{FA}}(\cdot \mid x, Y_{<t}^k)$.
end for
Normalize the final weights and **return** $\{(Y_{1:T}^k, \bar{w}_T^k)\}_{k=1}^K$.

Summing over reasoning paths gives

$$\sum_z \widetilde{\pi}_A(z, a \mid x) = \mathcal{Z}_A(x)^{-1} m_{\alpha_Z}(a \mid x)^{\alpha_A} = \pi_A(a \mid x).$$

The additional power factor depends on the answer marginal and is shared by all reasoning paths paired with the same answer sequence.

Numerical Evaluation of Answer Weights. We compute the answer importance weights using the empirical marginal $\widehat{m}_K$, which approximates m_{α_Z} from the finite weighted reasoning population. For each candidate A_j, its conditional log probability under reasoning path z_i is

$$\ell_{j,i} = \log p_0(A_j \mid z_i, x) = \sum_{t=1}^{|A_j|} \log p_0(A_{j,t} \mid x, z_i, A_{j,<t}).$$

Only answer tokens contribute to this score; the reasoning path provides the conditioning context. To evaluate the empirical marginal in log space, we set $b_j = \max_{1 \leq i \leq K} (\log \bar{w}_i^Z + \ell_{j,i})$ and compute

$$\log \widehat{m}_K(A_j \mid x) = b_j + \log \left(\sum_{i=1}^K \exp\left(\log \bar{w}_i^Z + \ell_{j,i} - b_j \right) \right).$$

The log importance weight is $\log w_j^A = (\alpha_A - 1) \log \widehat{m}_K(A_j \mid x)$. We similarly normalize these weights using $c = \max_{1 \leq r \leq M} \log w_r^A$:

$$\bar{w}_j^A = \frac{\exp(\log w_j^A - c)}{\sum_{r=1}^M \exp(\log w_r^A - c)}.$$

Subtracting the maximum improves numerical stability without changing the marginal probabilities or normalized importance weights.

C Experimental Setup and Implementation Details

Models and benchmarks. We evaluate Qwen3-4B, Qwen3-8B, and DeepSeek-R1-Distill-Llama-8B using their native chat templates, with thinking mode enabled for Qwen3. Inference uses `bfloat16` on one NVIDIA H100 80 GB GPU per task. We evaluate all 500 MATH500 problems, 198 GPQA-Diamond questions, 164 HumanEval problems, and 60 AIME24/25 problems. Results are averaged over seeds $\{10080, 10081, 10082\}$.

Generation and evaluation. Within each model–benchmark setting, all methods use the same model checkpoint, tokenizer, user-level problem prompt, maximum length per reasoning–answer trajectory, answer extractor, and grader. The maximum generation lengths are 4224 tokens for MATH500, 8192 for GPQA-Diamond, 4608 for HumanEval, and 16384 for AIME24/25. Standard and low-temperature decoding generate a single completion containing both reasoning and the final answer, using the full generation limit. Power-SMC likewise generates complete reasoning–answer particles within the same per-particle limit. These methods do not reserve a separate answer budget or insert a reasoning–answer boundary. Marginal Sharpening and DMS allocate the trajectory budget as $4096 + 128$ tokens for MATH500, $8064 + 128$ for GPQA-Diamond, $4096 + 512$ for HumanEval, and $16256 + 128$ for AIME24/25, where the first and second terms specify the reasoning and answer limits, respectively. If a reasoning trace reaches its limit without producing a reasoning–answer boundary, we append a single `</think>` token. This token is included in the continuation context but excluded from the reasoning-path likelihood and residual importance-weight calculation. For non-code benchmarks, the separated answer stage is initialized with a boxed-answer prefix and terminates after a complete boxed expression; HumanEval uses a native code continuation. Answer generation and cross-scoring use the same continuation context, including any inserted boundary token and fixed answer prefix; only generated answer tokens contribute to the conditional answer likelihood. Mathematical predictions are evaluated by normalized exact matching and symbolic equivalence, GPQA uses exact option matching, and HumanEval executes the original base tests in an isolated process with a three-second timeout.

Baselines. Standard and low-temperature decoding use temperatures 1 and 0.25, respectively, with top-$p = 1$. Power-SMC uses $N = 16$ particles and checks the particle weights every 64 tokens, applying systematic resampling when $\mathrm{ESS} < 0.5N$. Its final power exponents for (MATH500, GPQA-Diamond, HumanEval, AIME24/25) are $(4, 1.05, 4, 4)$ for Qwen3-4B, $(2, 1.25, 4, 4)$ for Qwen3-8B, and $(1.5, 1.01, 1.01, 1.01)$ for DeepSeek-R1-Distill-Llama-8B. By default, the proposal uses temperature 0.25 and top-$p = 0.9$, and the exponent increases linearly from 1 to its final value over 100 tokens. For Qwen3-8B on GPQA-Diamond and DeepSeek-R1-Distill-Llama-8B on GPQA-Diamond, HumanEval, and AIME24/25, the proposal instead uses temperature 1 and top-$p = 1$, with a 256-token linear annealing schedule. Marginal Sharpening uses a sharpening exponent of 4 and 8 independent Monte Carlo groups, each containing 4 reasoning traces, for 32 traces in total. It generates a single answer token by token using the resulting prefix-weighted marginal token probabilities.

DMS configuration. Unless stated otherwise, DMS uses $K = 16$ reasoning particles and $M = 32$ answer candidates. The power-two annealing schedule begins after 1024 reasoning tokens and reaches the terminal exponent at the benchmark-specific maximum reasoning length. For standard DMS, we check the ESS of the predictive weights every 64 tokens and apply systematic resampling when it falls below $K/2$. Each complete answer candidate is cross-scored under all retained reasoning contexts, and the final answer is sampled according to its normalized answer importance weight. Table 3 lists the exponents used in the main results.

Table 3: DMS reasoning and answer exponents (α_Z, α_A) used in the main results (Table 1).

Model	MATH500	GPQA-Diamond	HumanEval	AIME24/25
Qwen3-4B	$(1.25, 3)$	$(1.025, 2)$	$(1.25, 1.5)$	$(1.025, 3.5)$
Qwen3-8B	$(1.025, 2.5)$	$(1.025, 2)$	$(1.25, 1.5)$	$(1.005, 4)$
DeepSeek-R1-Distill-Llama-8B	$(1.001, 6)$	$(1.001, 5)$	$(1.25, 1.5)$	$(1.001, 8)$

Island implementation. On AIME24/25, all three models use eight islands of two particles, giving $K = 16$ reasoning particles in total. Resampling occurs independently within each island. At the reasoning–answer boundary, we normalize the particle weights within each island and assign equal total weight to all islands. Specifically, particle k in island ℓ receives pooled weight $\bar{w}_{\ell k}/L$, where $\sum_k \bar{w}_{\ell k} = 1$ and $L = 8$. These pooled weights define the empirical answer marginal used for candidate generation and selection.

D SENSITIVITY TO THE TWO SHARPENING EXPONENTS

We study the interaction between reasoning and answer sharpening using standard DMS with Qwen3-8B on AIME24/25. We evaluate

$$\alpha_Z \in \{1, 1.001, 1.005, 1.01, 1.025\}, \qquad \alpha_A \in \{1, 1.5, 2, 2.5, 3, 3.5, 4, 5, 6, 8\}.$$

All configurations use $K = 16$, $M = 32$, and the same generation budgets and sampler settings.

We report expected accuracy, the probability that categorical selection returns a correct answer, averaged over problems. For candidate correctness indicators $c_m(x) \in \{0, 1\}$ and normalized answer weights $\bar{w}_m(x; \alpha_A)$, this is

$$\mathrm{Acc}_{\mathrm{exp}} = \frac{1}{|\mathcal{D}|} \sum_{x \in \mathcal{D}} \sum_{m=1}^{M} \bar{w}_m(x; \alpha_A) c_m(x),$$

where $\mathcal{D}$ is the benchmark problem set. We average this metric over three seeds. At $\alpha_A = 1$, all answer weights are equal, so expected accuracy equals average candidate correctness. Oracle pass@32 is the fraction of problems with at least one correct candidate and remains unchanged across answer exponents within a fixed bank.

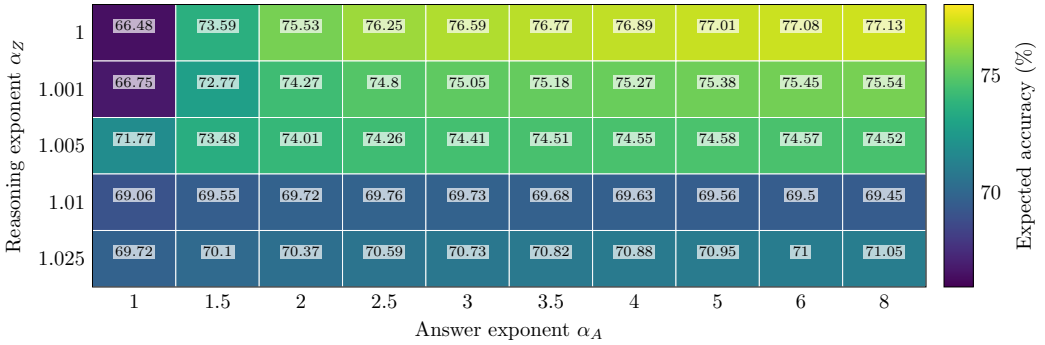

Figure 3: Expected accuracy (%) of standard DMS with Qwen3-8B on AIME24/25 across reasoning and answer exponents, averaged over three seeds.

Figure 3 shows that answer sharpening improves expected accuracy over uniform answer selection across all tested reasoning exponents. At $\alpha_Z = 1$, expected accuracy rises from 66.48% at $\alpha_A = 1$ to 76.89% at $\alpha_A = 4$ and 77.13% at $\alpha_A = 8$. Thus, most of the improvement is obtained by $\alpha_A = 4$, with limited additional gains from stronger answer sharpening.

Mild reasoning sharpening can also improve average candidate correctness: setting $\alpha_Z = 1.005$ raises it from 66.48% to 71.77%. However, the highest expected accuracy obtained from this reasoning bank is 74.58%, below the 77.13% achieved with $\alpha_Z = 1$. Improving average candidate correctness therefore does not necessarily improve accuracy after answer reweighting. In this standard (non-island) setting, the strongest result combines an unsharpened reasoning distribution with a larger answer exponent. These results support choosing the two sharpening strengths independently, allowing strong answer sharpening without requiring additional concentration of the reasoning distribution.

E ANNEALING AND RESAMPLING ABLATIONS

We evaluate annealing and reasoning-stage resampling in standard DMS using Qwen3-8B on AIME24/25, with $K = 16$, $M = 32$, and $(\alpha_Z, \alpha_A) = (1.005, 4)$. The reference sampler uses a quadratic annealing schedule that remains at $\beta_t = 1$ through token 1024 and reaches α_Z at the maximum reasoning length. At 64-token checkpoints, it performs systematic resampling whenever the predictive ESS falls below $K/2$.

The no-annealing variant applies α_Z from the first generated token, using the locally fully adapted proposal $q_t^{\mathrm{FA}}(v \mid h_t) \propto p_0(v \mid h_t)^{\alpha_Z}$ and the corresponding incremental importance weights. The no-resampling variant retains the reference annealing schedule, proposal, and sequential weight updates, but disables ancestor resampling. Its final reasoning weights are still used to construct the empirical answer marginal.

All variants approximate the same terminal reasoning target. Completed reasoning traces stop generating tokens but continue to receive annealing weight corrections as β_t increases. If all particles finish before the terminal exponent is reached, the remaining annealing updates are completed without generating additional tokens.

Table 4: Sampler ablations for non-island DMS with Qwen3-8B on AIME24/25, averaged over three seeds. Expected accuracy and oracle pass@32 are percentages. Final ESS is computed from the final reasoning weights and divided by $K = 16$. Resamples and unique roots report the mean number of reasoning-stage resampling events and surviving initial-particle ancestors, respectively.

Variant	Expected	Oracle pass@32	Final ESS$/K$	Resamples	Unique roots
standard DMS	**74.55**	78.33	0.765	1.39	8.17
No annealing	71.72	77.78	0.773	2.26	7.11
No resampling	72.09	**80.00**	0.252	0	**16.00**

Table 4 shows that the reference sampler achieves the highest mean expected accuracy, exceeding the no-annealing and no-resampling variants by 2.83 and 2.46 percentage points, respectively. Without annealing, resampling occurs more frequently and fewer distinct initial ancestors remain in the final population. This pattern is consistent with gradual annealing limiting ancestor loss during resampling.

Disabling resampling preserves all 16 initial ancestors and increases oracle pass@32 from 78.33% to 80.00%. However, the final reasoning weights become more concentrated, with normalized ESS falling to 0.252, and expected accuracy decreases to 72.09%. Thus, preserving more ancestors and greater correct-answer coverage does not necessarily improve weighted answer selection.

Final ESS and the number of surviving ancestors measure different aspects of the reasoning population. Resampling can balance particle weights without recovering ancestors that have already been discarded. Similar final ESS values can therefore coexist with different numbers of surviving ancestors, as observed for the reference and no-annealing variants.

F Higher-Budget Reasoning-Sampler Diagnostic

We further examine sensitivity to reasoning sharpening using a higher-budget replica-exchange Markov chain Monte Carlo sampler in place of FAA-SMC. Each replica targets $\pi_{\alpha_Z}(z \mid x) \propto p_0(z \mid x)^{\alpha_Z}$ through local Metropolis–Hastings updates that revise previously generated reasoning traces. Adjacent replicas also propose exchanging their traces to improve mixing across the exponent ladder. We use the resulting finite-chain samples as a diagnostic reference rather than as exact samples from the target distributions.

We evaluate Qwen3-4B on HumanEval using seeds $\{10080, 10081, 10082\}$, with reasoning and answer limits of 4096 and 512 tokens, respectively. For each problem and seed, we run 16 independent replica ladders and retain one final reasoning trace per exponent from each ladder. After the initial generation block, each replica receives 50 local Metropolis–Hastings updates. Each update selects a restart position uniformly within the generated reasoning trace and regenerates the remaining suffix at temperature $1/\alpha_Z$, followed by the corresponding Metropolis–Hastings acceptance step. After each round of local updates, the sampler performs a complete sweep of adjacent replica-exchange proposals. We run the fine ladder $(1.1, 1.05, 1.025, 1)$ and the extended ladder $(4, 3, 2, 1.5, 1.25, 1.1)$ independently. These configurations maintain 64 and 96 replica chains per problem and seed, respectively, in addition to the repeated suffix generation required by the local updates.

For each reasoning exponent, the $K = 16$ retained traces define a uniformly weighted reasoning mixture. We generate $M = 32$ answer candidates from this mixture and cross-score each candidate under all retained reasoning contexts. The same candidates and cross-score matrix are reused across

Table 5: Reasoning-exponent sensitivity with a higher-budget replica-exchange sampler using Qwen3-4B on HumanEval. Expected accuracy and oracle pass@32 are reported as percentages averaged over three seeds. The fine and extended ladders are run independently.

α_Z	Expected accuracy ($\alpha_A = 4$)	Oracle pass@32
Fine ladder		
1	95.06	98.17
1.025	96.17	97.97
1.05	96.59	97.97
1.1	96.39	97.97
Extended ladder		
1.1	96.06	98.17
1.25	96.02	98.17
1.5	96.61	98.17
2	95.88	97.97
3	96.01	97.97
4	96.23	97.97

the answer-exponent grid. Candidates are evaluated using the same full-program-preserving HumanEval extractor and isolated execution protocol as the main results, with a three-second timeout. Table 5 reports expected accuracy at $\alpha_A = 4$ and oracle pass@32.

Both ladders achieve their highest expected accuracy at moderate reasoning exponents within their respective ranges. In the fine ladder, expected accuracy increases from 95.06% at $\alpha_Z = 1$ to 96.59% at $\alpha_Z = 1.05$, before slightly decreasing to 96.39% at $\alpha_Z = 1.1$. In the extended ladder, expected accuracy likewise peaks at a moderate exponent, reaching 96.61% at $\alpha_Z = 1.5$.

Oracle pass@32 decreases slightly from 98.17% to 97.97% as α_Z increases within each ladder. This pattern is consistent with the trade-off introduced by sharpening: concentrating probability on more likely reasoning paths can reduce diversity and the chance of finding a 'lucky' correct answer through a less likely path. Meanwhile, the gains in expected accuracy show that moderate reasoning sharpening can improve weighted answer selection while maintaining high correct-answer coverage.

G Time-Cost Analysis

DMS inference consists of reasoning-particle generation, answer-candidate generation, and cross-scoring the $K \times M$ reasoning–answer pairs. Cross-scoring reuses cached reasoning prefixes to evaluate answer continuations. We measure wall-clock inference time per problem using the hardware configuration described in Appendix C. The measurements include all three stages and exclude model loading and offline grading.

Table 6: Mean wall-clock inference time in seconds per problem, averaged over seeds.

Model	Method	MATH500	GPQA-Diamond	HumanEval	AIME24/25
Qwen3-4B	Standard Decoding	50.57	124.41	43.06	249.27
	Low-Temperature Decoding	49.36	123.26	43.09	259.47
	Power-SMC	46.01	163.14	43.33	376.40
	Marginal Sharpening	140.51	710.42	103.05	1957.74
	DMS (Ours)	62.09	241.09	83.49	617.09
Qwen3-8B	Standard Decoding	57.09	133.99	43.92	261.37
	Low-Temperature Decoding	52.62	131.98	44.05	269.33
	Power-SMC	52.36	191.54	44.95	369.65
	Marginal Sharpening	477.17	1270.15	436.42	2855.41
	DMS (Ours)	82.69	237.56	89.30	874.87
DeepSeek-R1-Distill-Llama-8B	Standard Decoding	33.13	92.97	34.79	172.13
	Low-Temperature Decoding	28.47	93.20	32.66	177.10
	Power-SMC	31.97	177.11	54.48	483.29
	Marginal Sharpening	359.23	1002.83	389.05	2204.19
	DMS (Ours)	76.03	207.87	132.94	740.26

Table 6 reports the mean inference time per problem. Relative to standard decoding, non-island DMS requires 1.23–3.82 times the latency, while Island-DMS on AIME24/25 requires 2.48–4.30 times the latency. Across all twelve model–benchmark settings, DMS has lower measured latency than Marginal Sharpening.

For Island-DMS with Qwen3-8B on AIME24/25, reasoning generation, answer generation, and cross-scoring take 705.38 s (80.63%), 25.24 s (2.89%), and 144.23 s (16.49%) per problem, respectively. Reasoning generation accounts for more than four-fifths of the measured latency, followed by cross-scoring, while answer generation contributes less than 3%. These timings reflect the evaluated implementations and configurations.