DEPTH-ENTROPY GUIDED SAMPLING FOR TRAINING-FREE LLM REASONING

Anonymous authors
Paper under double-blind review

ABSTRACT

Reinforcement learning (RL) has become the dominant paradigm for improving the reasoning capabilities of large language models, but it requires expensive training, curated data, and reward signals. Recent work shows that sampling from sharpened base-model distributions at test time recovers much of the RL gain, yet existing methods rely solely on output-layer likelihoods and ignore the transformer's internal forward-pass dynamics. We introduce **Depth-Entropy Guided Sampling (DEGS)**, a training-free, test-time method that exploits *layer-wise entropy collapse* as an intrinsic quality signal. We observe that stronger reasoners—including RL-posttrained variants—exhibit a distinctive "late collapse": logit-lens–decoded entropy stays elevated until deeper layers before converging. We define a per-sequence *collapse depth* $D(\mathbf{x})$ and a joint objective $\pi(\mathbf{x}) \propto p(\mathbf{x})^{\alpha} \exp\big(\beta\, D(\mathbf{x})\big)$ that combines sequence likelihood with this depth-entropy structure, instantiated inside an MCMC power-sampling framework (DEGS-MCMC). Across three open-weight models and four reasoning benchmarks, this near-chance per-candidate signal compounds over the sampling trajectory into state-of-the-art training-free accuracy, with gains largest out of domain and on the harder splits—exactly where likelihood alone falls short—at single-digit-percent wall-clock overhead. DEGS narrowly trails an in-house GRPO reference on the math splits GRPO was trained for, yet surpasses it out of domain on GPQA for all three models, without any training, reward model, or labeled data.

1 INTRODUCTION

Reinforcement learning (RL) with verifiable rewards—e.g., Group Relative Policy Optimization (GRPO) (Shao et al., 2024)—is the dominant paradigm for improving the reasoning capabilities of large language models (LLMs), posttraining frontier models to sizeable gains on mathematical, scientific, and coding benchmarks (Guo et al., 2025). Yet these gains are costly: RL posttraining requires curated data, extensive tuning, and—most critically—a reliable reward signal, which is unavailable in many domains of practical interest (Prabhudesai et al., 2025). A growing body of evidence further suggests they may not reflect fundamentally new capabilities: RL-posttrained models concentrate probability mass on traces that *already* have high likelihood under the base model (He et al., 2025; Yue et al., 2025), and base models can even outperform them in the multi-shot regime due to degraded diversity (Song et al., 2025). This *distribution sharpening* hypothesis (Shao et al., 2025) casts RL mainly as a filter, redistributing pass@k capability into single-shot performance rather than teaching genuinely novel reasoning.

This has inspired *training-free* methods that sharpen the base distribution at inference time, part of a broader move toward scaling test-time computation (Snell et al., 2024; Welleck et al., 2024; Brown et al., 2024). Karan and Du (2025) formalize this through the *power distribution* $p(\mathbf{x})^{\alpha}$ ($\alpha > 1$), targeted by Metropolis–Hastings (MH) sampling, to match GRPO without training, data, or reward models; Scalable Power Sampling (Ji et al., 2026) and Power-SMC (Azizi et al., 2026) improve efficiency via token-level approximation and a batch-parallel SMC scheme. Together they show base models are far more capable at single-shot reasoning than standard sampling reveals.

Despite their success, all existing training-free methods rely exclusively on the *output-layer likelihood* $p(\mathbf{x})$, ignoring the structured intermediate representations of the forward pass (Nostalgebraist, 2020; Belrose et al., 2023; Wendler et al., 2024). This is a missed opportunity: Wendler et al. (2024) identify

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.

three forward-pass phases (input processing, a middle "concept space," and output mapping), and Tan et al. (2025) find that entropy evolves from high-entropy exploration in early layers to deterministic refinement at the top.

We contribute an empirical observation bridging these lines: comparing *layer-wise entropy trajectories* of base models against their RL-posttrained counterparts, stronger reasoners consistently exhibit **later entropy collapse** (Figure 1)—base models lock in predictions at shallow layers, while RL-posttrained models sustain elevated internal uncertainty until much deeper. This extended "deliberation phase" is associated with downstream accuracy (a small but consistent effect, quantified in Section 5) and offers a new lens on what RL posttraining achieves internally. This motivates a simple idea: *if late entropy collapse is characteristic of strong reasoning, we can use it as a training-free quality signal at test time*. We formalize it through a scalar *collapse depth* $D(\mathbf{x})$ and construct a joint objective $\pi(\mathbf{x}) \propto p(\mathbf{x})^{\alpha} \cdot \exp(\beta\, D(\mathbf{x}))$ that augments the power distribution with this signal. The resulting method, **Depth-Entropy Guided Sampling (DEGS)**, integrates into the MH power-sampling framework (Algorithm 1)—without parameter updates, reward models, or labeled data.

We keep two linked claims distinct: the *motivation* (Figure 1) is a cross-stage contrast—posttrained reasoners collapse later than their base models—while the *operative* claim our reweighting of base-model samples exploits is internal to a single base model: among its candidates, the correct ones collapse later than the incorrect ones, which we verify directly in Section 5 (Figure 2).

Our contributions are as follows:

1. We identify **late entropy collapse** as an empirical signature of strong reasoning models and show that the collapse depth $D(\mathbf{x})$ is a statistically significant predictor of solution correctness across models and benchmarks (Section 3).

2. We propose **DEGS**, built on a joint likelihood–depth objective $\pi(\mathbf{x}) \propto p(\mathbf{x})^{\alpha} \exp(\beta D(\mathbf{x}))$ (Section 4), in two decoders: a final-layer reranker, *Best-of-N-Entropy*, which beats log-likelihood Best-of-N in all twelve cells by modest margins, and our main method **DEGS-MCMC**, which exploits the full depth profile to improve over likelihood-only power sampling in all twelve cells by substantially larger margins (median ≈ 4 points; Table 1).

3. We characterize **how a weak per-candidate signal becomes a reliable decoding gain**: collapse depth is only a near-chance discriminator in isolation (AUC ≈ 0.6; Section 3), yet by Proposition 8 it can only reorder near-equal-likelihood candidates, so compounding this tiebreak over the MCMC trajectory yields a gain positive in all twelve cells and largest on the harder, out-of-domain splits where likelihood is weakest.

To our knowledge, DEGS is the first method to use the *depth-wise evolution of internal uncertainty*—rather than output-layer signals alone—as a test-time selection criterion for reasoning, indicating that intermediate representations carry quality signals not reducible to sequence-level likelihood.

2 Related Work

RL posttraining and distribution sharpening. RL with verifiable rewards (GRPO) (Shao et al., 2024; Guo et al., 2025), building on RL from human feedback (Ouyang et al., 2022) and policy-gradient methods such as PPO (Schulman et al., 2017), is now the dominant approach for improving LLM reasoning, but growing evidence attributes its gains primarily to *distribution sharpening* rather than new capabilities (He et al., 2025; Yue et al., 2025; Song et al., 2025), motivating the training-free power-sampling line above (Karan and Du, 2025; Ji et al., 2026; Azizi et al., 2026); verifier-based reranking, process/outcome reward models, self-consistency, and pure repeated sampling (Cobbe et al., 2021; Uesato et al., 2022; Lightman et al., 2024; Wang et al., 2023; Brown et al., 2024) are simpler alternatives. Using a quality signal *inside* MH connects to QuEST (Faria et al., 2024), SMC steering of language-model programs (Lew et al., 2023), and twist-guided sequential Monte Carlo (Zhao et al., 2024), but those steer the chain with an *external*, task-specific estimator (a learned MT metric, a trained twist), whereas DEGS reads its signal off the base model's own intermediate representations—no auxiliary model, reward, or training. All these methods rely on *output-layer* signals; DEGS complements them with *internal* depth-entropy dynamics.

Internal representations and logit lens. The logit lens (Nostalgebraist, 2020), tuned lens (Belrose et al., 2023), and unifying frameworks such as Patchscopes (Ghandeharioun et al., 2024) decode intermediate hidden states into token distributions, revealing evolving internal beliefs; Wendler et al. (2024) show these carry information orthogonal to output tokens, and Tan et al. (2025) find early layers maintain high entropy while top layers converge. Closest to our use of depth is DoLa (Chuang et al., 2024), which contrasts logit-lens projections of a later and an earlier layer to reduce hallucination; unlike DoLa's per-token, two-layer contrast, DEGS leaves the token distribution unchanged and instead aggregates the full depth profile of entropy collapse into a sequence-level reweighting signal, used as a *test-time selection signal* rather than for interpretation or RL optimization.

Entropy signals for reasoning. Prabhudesai et al. (2025) maximize base-model confidence via RL, Zhao et al. (2025) use self-entropy as an RL reward, and Cui et al. (2025) analyze *output-policy* entropy collapse during RL training; all operate on output-layer entropy within a training framework, whereas DEGS exploits entropy *across the depth dimension* entirely at *test time*.

Confidence, calibration, and adaptive depth. Models' *own* uncertainty is also a usable signal: large models are often well calibrated and can partly predict their own correctness (Kadavath et al., 2022), and output-distribution measures such as semantic entropy flag unreliable generations (Farquhar et al., 2024). Closest in spirit to our per-token collapse depth is *confidence-based early exiting* (Schuster et al., 2022), which halts computation at a shallow layer once a per-token confidence criterion is met; DEGS inverts this efficiency framing—rather than exiting early to save compute, it *reads how deep* the network defers commitment as a test-time quality signal, leaving the forward pass intact.

Notation. We consider an autoregressive LLM $p(\mathbf{x}) = \prod_t p(x_t \mid x_{<t})$ with L transformer layers and hidden states $\mathbf{h}_{l,t}$. The *logit lens* (Nostalgebraist, 2020) decodes layer l via the unembedding U, $p_l(v \mid x_{\leq t}) = \mathrm{softmax}(U\mathbf{h}_{l,t})_v$, with layer-wise entropy $H_l(t) = -\sum_v p_l(v) \log p_l(v)$; the *power distribution* (Karan and Du, 2025) $p^\alpha(\mathbf{x}) \propto p(\mathbf{x})^\alpha$ ($\alpha \geq 1$) is targeted by MH sampling. Full definitions are in Appendix B.

3 Motivating Observation: Late Entropy Collapse in Strong Reasoners

Comparing layer-wise entropy trajectories $\{H_l(t)\}_{l=1}^L$ across training stages (Figure 1), **base models** collapse early (entropy drops to near zero at shallow layers), while **RL-posttrained models** sustain elevated entropy significantly deeper before converging—consistent with the three-phase structure of Wendler et al. (2024). In Figures 1(a)–(b), base Qwen2.5-Math-7B drops below 2.5 nats by layer 10 whereas its RL-posttrained variant stays above 5 nats until layer 15–20; the same pattern holds for DeepSeek-Math-7B (Figures 1(c)–(d)), and panel (e) confirms higher mean collapse depth for posttrained models across all four settings. Combined with the finding that base models already contain high-quality reasoning traces (Karan and Du, 2025), this motivates our hypothesis: *among base-model candidates, those whose entropy collapses later are more likely to be correct*.

This is the cross-stage *motivation*; the base-internal claim our method uses is verified in Section 5 (Figure 2). Appendix A (Figure 3) extends the contrast to all twelve overlays, including GSM8K (Cobbe et al., 2021) and Qwen2.5-Math-72B (motivation only).

4 Method: Depth-Entropy Guided Sampling

We formalize the observation into a test-time algorithm: we define *collapse depth*, construct the DEGS objective, and present a lightweight reranking variant alongside the MCMC variant we ship.

4.1 Collapse Depth

Per-token collapse depth. Consider a generated sequence $\mathbf{x} = (x_0, \ldots, x_T)$ from a base model p. To reduce cost, we compute entropy on a *layer subset*

$$\mathcal{L}_{\mathrm{sub}} = \{l_1, \ldots, l_K\} \subset \{1, \ldots, L\}, \tag{1}$$

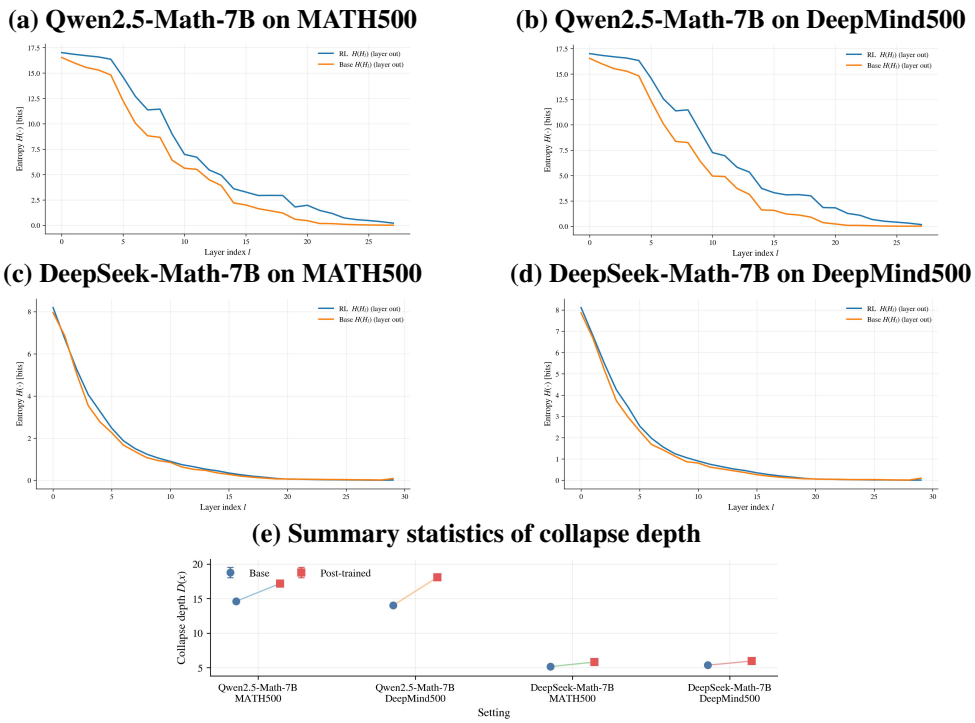


Figure 1: Late entropy collapse in stronger reasoners. Panels (a)–(d) plot layer-wise entropy trajectories for representative base and posttrained models on mathematical benchmarks; panel (e) summarizes the effect via the mean collapse *layer* $\overline{d_t}$ (average over tokens of the shallowest probed layer reaching the threshold, in absolute layer index—not the normalized $D(\mathbf{x}) \in (0, 1]$ of Definition 2). Stronger reasoners collapse later in depth—equivalently, a larger $D(\mathbf{x})$—suggesting that internal uncertainty dynamics may provide a useful test-time signal.

with $K \ll L$; by default we use evenly spaced $\mathcal{L}_{\text{sub}} = \{4, 8, 12, 16, 20, 24, 28\}$ for a 28-layer model ($K = 7$).

Definition 1 (Per-token collapse depth)**.** *Given an entropy threshold $\tau > 0$, the* collapse depth *of token position t is*

$$d_t = \min\{l \in \mathcal{L}_{\text{sub}} \mid H_l(t) \le \tau\}, \tag{2}$$

with the convention $d_t = L$ if no layer in $\mathcal{L}_{\text{sub}}$ satisfies the threshold.

Intuitively d_t measures how deep the network processes before becoming "confident" about token t: small d_t indicates (possibly premature) early commitment, large d_t sustained deliberation.

Sequence-level collapse depth.

Definition 2 (Sequence-level collapse depth)**.** *The* normalized collapse depth *of $\mathbf{x} = (x_0, \ldots, x_T)$ is*

$$D(\mathbf{x}) = \frac{1}{T+1} \sum_{t=0}^{T} \frac{d_t}{L}. \tag{3}$$

By construction $D(\mathbf{x}) \in (0, 1]$; larger values indicate internal uncertainty persisting to deeper layers on average. Our central hypothesis (validated in Section 5) is that *correct solutions tend to have larger $D(\mathbf{x})$*. The threshold τ (default 0.25 nats) controls sensitivity; DEGS is robust across a range of values (Table 8).

4.2 The DEGS Objective

Standard power sampling (Karan and Du, 2025) targets $p(\mathbf{x})^\alpha$ using only the output-layer likelihood. We augment it with the depth-entropy structure captured by $D(\mathbf{x})$.

Definition 3 (DEGS target distribution)**.** *The DEGS target distribution over sequences is*

$$\pi(\mathbf{x}) \propto p(\mathbf{x})^{\alpha} \cdot \exp\big(\beta \cdot D(\mathbf{x})\big), \tag{4}$$

where $\alpha \geq 1$ is the likelihood exponent and $\beta \geq 0$ the depth weight.

The **likelihood term** $p(\mathbf{x})^{\alpha}$ favors globally coherent sequences (as in power sampling); the **depth-entropy term** $\exp(\beta\, D(\mathbf{x}))$ favors sequences whose internal processing resembles strong reasoners'—elevated uncertainty across layers before committing. It reduces to power sampling at $\beta = 0$ and depth-only scoring at $\alpha = 0$ (not expected to perform well alone). Equivalently, the log-score for ranking is

$$S(\mathbf{x}) = \alpha \cdot \log p(\mathbf{x}) + \beta \cdot D(\mathbf{x}). \tag{5}$$

Defaults: α=4, β=5, τ=0.25 nats, $\mathcal{L}_{\text{sub}}$=$\{4, 8, \ldots, 28\}$, N=16 candidates, T_{MCMC}=10 MH steps; full ranges in Table 8.

4.3 Reranking Variants

Both variants use one scoring signal but apply it differently: reranking applies the depth tiebreak once, whereas DEGS-MCMC compounds it across the sampling trajectory. **Best-of-N-Entropy** ranks a fixed pool of N candidates by final-layer entropy alone; it beats log-likelihood Best-of-N in all twelve cells (Table 1), but by modest margins and well below the MCMC family. **DEGS reranking** instead uses the full depth score $S(\mathbf{x}) = \alpha \log p(\mathbf{x}) + \beta D(\mathbf{x})$ (Algorithm 2, Appendix C), computed from one teacher-forced forward that caches hidden states at $\mathcal{L}_{\text{sub}}$. By Proposition 8 the depth term only reorders near-equal-likelihood candidates, so a single argmax over N=16 extracts little from it; we therefore keep DEGS reranking as a controlled probe of *where* in depth the signal acts (Table 7) and ship DEGS-MCMC, which compounds the same tiebreak over the trajectory at a few percent overhead relative to Best-of-N (Table 6).

4.4 DEGS-MCMC: Integration with Power Sampling

For a more sample-efficient exploration of equation 4, we integrate DEGS into the MCMC power-sampling framework of Karan and Du (2025).

Acceptance ratio. In standard power sampling the target is $p(\mathbf{x})^{\alpha}$ with MH acceptance ratio $A_{\text{std}}(\mathbf{x}', \mathbf{x}) = \min\big(1, \frac{p(\mathbf{x}')^{\alpha}\, q(\mathbf{x}|\mathbf{x}')}{p(\mathbf{x})^{\alpha}\, q(\mathbf{x}'|\mathbf{x})}\big)$, where q is the proposal (random-index resampling). Replacing the target with $\pi(\mathbf{x})$ from equation 4 yields

$$A_{\text{DEGS}}(\mathbf{x}', \mathbf{x}) = \min\left(1,\ \frac{p(\mathbf{x}')^{\alpha}\, \exp\big(\beta\, D(\mathbf{x}')\big)\, q(\mathbf{x} \mid \mathbf{x}')}{p(\mathbf{x})^{\alpha}\, \exp\big(\beta\, D(\mathbf{x})\big)\, q(\mathbf{x}' \mid \mathbf{x})}\right). \tag{6}$$

All other components—proposal, block structure, chain selection—remain identical to Algorithm 1 of Karan and Du (2025); the only addition is an entropy-probing pass to compute $D(\cdot)$.

Algorithmic structure. Following Karan and Du (2025), we divide the sequence into blocks of size B with intermediate targets $\pi_k(x_{0:kB}) \propto p(x_{0:kB})^{\alpha} \exp\big(\beta D(x_{0:kB})\big)$, extending each prefix with the proposal model and then running N_{MCMC} MH steps with equation 6 (Algorithm 1).

Convergence and cost. The chain converges to π as $N_{\text{MCMC}} \to \infty$ by standard MCMC theory (Neal, 1993)—$\pi(\mathbf{x})$ has nonzero mass wherever $p(\mathbf{x}) > 0$ and the random-index resampling proposal is irreducible and aperiodic (Karan and Du, 2025) (full statement and proof in Appendix F.2). Cost is dominated by the $\frac{N_{\text{MCMC}}\, T^2}{4B}$ generated tokens of power sampling, plus $O(N_{\text{MCMC}} \cdot K)$ entropy computations at the probed layers (Table 6); design choices and defaults are in Appendices D–E.

5 Experiments

Setup. We evaluate DEGS on mathematical reasoning (MATH500 (Hendrycks et al., 2021; Lightman et al., 2024) and DeepMind500, the latter a held-out subset we construct from the DeepMind

Algorithm 1 DEGS-MCMC (Entropy-Guided Power Sampling)

Require: Base model p; proposal model p_{prop}; prompt; α, β, τ; block size B; max length T; MCMC steps N_{MCMC}; layer subset $\mathcal{L}_{\text{sub}}$
Ensure: Sequence $\mathbf{x}_{0:T} \sim \pi$

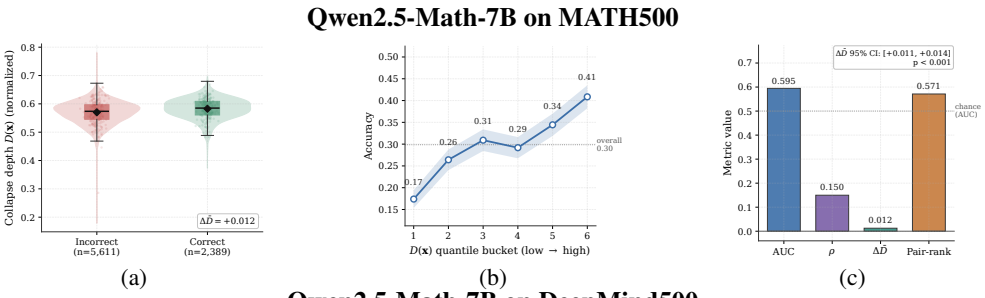
for $k = 0$ **to** $\lceil T/B \rceil - 1$ **do**
  Initialize: Extend prefix $\mathbf{x}_{0:kB}$ by sampling $x_t \sim p_{\text{prop}}(\cdot \mid x_{<t})$ for $t = kB{+}1, \ldots, (k{+}1)B$
  Set current state $\mathbf{x} \leftarrow \mathbf{x}_{0:(k+1)B}$
  for $n = 1$ **to** N_{MCMC} **do**
    Sample index $m \sim \text{Uniform}\{1, \ldots, (k{+}1)B\}$
    Construct proposal $\mathbf{x}'$: keep prefix $x_{0:m-1}$, resample $x'_t \sim p_{\text{prop}}(\cdot \mid x'_{<t})$ for $t = m, \ldots, (k{+}1)B$
    Compute $\log p(\mathbf{x})$, $\log p(\mathbf{x}')$, $D(\mathbf{x})$, $D(\mathbf{x}')$, $q(\mathbf{x} \mid \mathbf{x}')$, $q(\mathbf{x}' \mid \mathbf{x})$
    $A \leftarrow \min\Big(1,\ \frac{p(\mathbf{x}')^{\alpha} \exp(\beta\, D(\mathbf{x}'))\, q(\mathbf{x}|\mathbf{x}')}{p(\mathbf{x})^{\alpha} \exp(\beta\, D(\mathbf{x}))\, q(\mathbf{x}'|\mathbf{x})}\Big)$
    if $\text{Uniform}(0,1) \le A$ **then**
      $\mathbf{x} \leftarrow \mathbf{x}'$ {accept proposal}
    end if
  end for
  Fix prefix $\mathbf{x}_{0:(k+1)B} \leftarrow \mathbf{x}$
end for
return $\mathbf{x}_{0:T}$

mathematics dataset (Saxton et al., 2019)), code generation (HumanEval (Chen et al., 2021)), and scientific reasoning (GPQA-Diamond (Rein et al., 2024)) with three open-weight base models—Qwen2.5-7B (Qwen, 2025), Qwen2.5-Math-7B (Yang et al., 2024), and DeepSeek-Math-7B (Shao et al., 2024) (full benchmark, grading, and training details in Appendix E). The zero-shot chain-of-thought prompts (Wei et al., 2022; Kojima et al., 2022), answer parsers, and graders are held *identical* across all decoding methods, so accuracy differences reflect the decoder rather than the harness. The *GRPO (MATH)* rows are not transcribed from prior work: we train one GRPO reference per base model under a standard RLVR recipe (Lambert et al., 2024) (DeepScaleR prompts (Luo et al., 2025), a ground-truth math verifier, group size 16) and evaluate it under the same protocol, so GRPO functions as a fair, in-domain reference trained under matched conditions. Because DEGS adds an entropy-probing pass, all methods are compared at matched test-time compute, normalized to forward-equivalent units (Table 6, Section 5.3).

Baselines. We compare against: (1) base decoding, (2) low-temperature sampling, (3) Best-of-N with log-likelihood, (4) MCMC power sampling (Karan and Du, 2025), (5) Scalable Power Sampling (Ji et al., 2026), (6) Power-SMC (Azizi et al., 2026), and (7) GRPO-posttrained models (Shao et al., 2024). Our methods are *Best-of-N-Entropy* (final-layer-entropy reranking) and *DEGS-MCMC*.

5.1 Does Collapse Depth Alone Predict Correctness?

Before using $D(\mathbf{x})$ as a scoring signal, we verify it correlates with correctness: for each problem we sample $N{=}16$ candidates ($T{=}0.8$, top-$p{=}0.9$), judge correctness, and compute $D(\mathbf{x})$. On Qwen2.5-Math-7B / MATH500 (Figure 2a–c), correct candidates have significantly higher mean collapse depth than incorrect ones ($p < 0.001$), and accuracy climbs monotonically across $D(\mathbf{x})$ buckets, from 0.17 in the lowest to 0.41 in the highest—a more than twofold spread that recurs, more sharply, on the harder DeepMind500 split (d–f, top bucket $\approx 2.6\times$ the bottom). Taken on its own, however, collapse depth is a *weak per-candidate* discriminator: near-chance in isolation (AUC ≈ 0.6), with a low Spearman $\rho \approx 0.15$ and pairwise-ranking accuracy barely above one-half (c,f). The significance therefore reflects the size of the $n{=}8000$ candidate pool rather than a large effect. This is precisely the regime DEGS targets: by Proposition 8 a moderate β shifts each log-score by at most β nats, so the depth term acts only as a tiebreaker among near-equal-likelihood candidates—where being right slightly more often than chance suffices once compounded over the MCMC trajectory (Section 5.2).



(a) (b) (c)

(d) (e) (f)

$D(\mathbf{x})$ for correct vs. incorrect | Accuracy by $D(\mathbf{x})$ bucket | Predictive metrics for $D(\mathbf{x})$

Figure 2: Collapse depth $D(\mathbf{x})$ as a predictor of correctness, on two representative settings: Qwen2.5-Math-7B on MATH500 (a–c) and DeepMind500 (d–f). Each row shows the distribution of $D(\mathbf{x})$ for correct vs. incorrect candidates, the empirical accuracy of candidates by $D(\mathbf{x})$ quantile bucket, and the predictive statistics (AUC, Spearman ρ, mean gap $\Delta\bar{D}$, pairwise ranking accuracy). In both settings, later collapse is positively associated with correctness.

5.2 Main Results

DEGS is the strongest training-free decoder in 11 of 12 cells. DEGS-MCMC attains the best training-free accuracy on all four benchmarks for both Qwen backbones and on three of four for DeepSeek-Math-7B. On Qwen2.5-Math-7B it improves over MCMC power sampling (Karan and Du, 2025) on every benchmark and beats the strongest prior training-free method, Power-SMC, on all four—though the margin narrows to 0.5–2 points on the math splits, where the depth term refines rather than replaces the likelihood signal. Every power-sampling method far outstrips naive decoding (on Qwen2.5-7B / DeepMind500, $0.250 \rightarrow 0.573$). The lone exception is DeepSeek-Math-7B / GPQA, where low-temperature decoding is anomalously strong (0.430, beating every power-sampling variant including DEGS)—tellingly the setting where the late-collapse signal is weakest (Figure 1(c–e)). This is a boundary of the method: when the depth signal is uninformative, the likelihood-driven power distribution governs.

DEGS narrows the gap to GRPO on math and surpasses it out of domain. GRPO is posttrained on MATH and stays strongest in-domain, edging DEGS by 1.8 points on MATH500 and a comparable margin on DeepMind500; even so, with no training or reward signal, DEGS closes most of the 3.7- and 5.8-point gaps plain power sampling leaves on those splits. Out of domain the picture reverses: DEGS exceeds GRPO on GPQA for all three models and on HumanEval for both Qwen models—reaching 0.769 on Qwen2.5-7B HumanEval, well clear of GRPO—with GRPO keeping only a slight edge on DeepSeek-Math-7B HumanEval. This supports our framing: GRPO redistributes capability within the distribution it was optimized on, whereas DEGS reweights base-model samples and preserves cross-domain coverage. The per-cell pattern tracks the late-collapse signal: the two Qwen backbones, with the largest base-versus-posttrained collapse-depth separation (Figure 1), show the clearest gains, while DeepSeek-Math-7B—where that separation is smallest—shows almost none.

Takeaway and statistical reliability. The depth-entropy term never lowers accuracy relative to the likelihood-only power sampling it extends: it improves on that baseline in all twelve cells and is the best training-free decoder in eleven, concentrating its gains exactly where likelihood is weakest—out-of-domain GPQA and the harder DeepMind500 split—rather than acting as a new capability. The comparison is *within-framework*: toggling only β (5 vs. 0) holds decoder, harness, and budget fixed,

Table 1: Main benchmark results across models, tasks, and decoding methods, comparing standard decoding baselines, power-sampling baselines, our entropy-guided variants, and GRPO references. For DeepSeek-Math-7B, the Base / Low-temperature rows reflect the un-tuned base checkpoint under our standardized decoding protocol; with no instruction tuning the model frequently fails to emit a parsable boxed answer.

Method	MATH500 ↑	DeepMind500 ↑	HumanEval Pass@1 ↑	GPQA ↑
Qwen2.5-7B				
Base	0.498	0.250	0.329	0.278
Low-temperature	0.628	0.342	0.524	0.303
Best-of-N	0.650	0.395	0.609	0.282
MCMC Power Sampling	0.706	0.537	0.622	0.318
Scalable Power Sampling	0.708	0.549	0.756	0.349
Power-SMC	0.714	0.563	0.761	0.351
Best-of-N-Entropy (ours)	**0.664**	**0.412**	**0.632**	**0.348**
DEGS-MCMC (ours)	**0.728**	**0.573**	**0.769**	**0.393**
GRPO (MATH)	**0.740**	**0.598**	**0.561**	**0.354**
Qwen2.5-Math-7B				
Base	0.496	0.332	0.329	0.278
Low-temperature	0.690	0.482	0.512	0.353
Best-of-N	0.684	0.465	0.512	0.343
MCMC Power Sampling	0.748	0.574	0.573	0.389
Scalable Power Sampling	0.758	0.593	0.604	0.409
Power-SMC	0.762	0.601	0.612	0.413
Best-of-N-Entropy (ours)	**0.687**	**0.479**	**0.537**	**0.369**
DEGS-MCMC (ours)	**0.767**	**0.619**	**0.618**	**0.426**
GRPO (MATH)	**0.785**	**0.632**	**0.537**	**0.399**
DeepSeek-Math-7B				
Base	0.362	0.212	0.415	0.333
Low-temperature	0.366	0.269	0.427	0.430
Best-of-N	0.420	0.276	0.433	0.338
MCMC Power Sampling	0.424	0.307	0.470	0.345
Scalable Power Sampling	0.464	0.314	0.487	0.364
Power-SMC	0.467	0.329	0.498	0.386
Best-of-N-Entropy (ours)	**0.423**	**0.298**	**0.453**	**0.341**
DEGS-MCMC (ours)	**0.475**	**0.336**	**0.508**	**0.392**
GRPO (MATH)	**0.492**	**0.352**	**0.524**	**0.333**

so the uniformly positive per-cell deltas of Table 1 (median ≈ 4 points) are not confounded by the implementation differences a head-to-head with an external baseline would introduce. We claim only their *uniform direction*, not any single magnitude: with ≈ 1.9 points of binomial error on a 500-item split we report no per-cell variance, and although the twelve-cell direction passes a one-sided sign test ($p = 2^{-12} \approx 2.4 \times 10^{-4}$), we read this as corroborating the β-toggle rather than as a standalone test, since the cells share models and benchmarks and are not independent.

5.3 Accuracy versus Compute Budget

Matched-budget protocol. Because DEGS adds an entropy-probing pass, we normalize all methods to forward-equivalent units (Table 6): each reranking method is charged N generation passes plus probing, each MCMC-family method $B \cdot T_{\mathrm{MCMC}} = 160$ proposal forwards, and layer-subset probing K/L of a forward per probed candidate.

The depth signal is nearly free. The probe is teacher-forced and touches only K of L layers, so it costs a fraction of a full forward pass rather than a whole one. On Qwen2.5-Math-7B / MATH500 it

Table 2: Core ablation of DEGS components on Qwen2.5-Math-7B, isolating likelihood-based scoring, depth-based entropy scoring, and alternative entropy signals under a matched-compute protocol ($N = 16$, MCMC steps $= 10$, block size $B = 16$).

Method	$\alpha \log p(\mathbf{x})$	$\beta D(\mathbf{x})$	Final-layer entropy	Selection mode	MATH500 ↑	DeepMind500 ↑	GPQA ↑
Best-of-N (log-likelihood)	✓	–	–	Rerank	68.4	46.5	34.3
Power Sampling ($\beta = 0$)	✓	–	–	MCMC	74.8	57.4	38.9
Depth-only scoring ($\alpha = 0$)	–	✓	–	Rerank	63.2	41.4	32.8
Best-of-N-Entropy (ours)	–	–	✓	Rerank	68.7	47.9	36.9
DEGS reranking	✓	✓	–	Rerank	71.6	54.8	36.2
DEGS-MCMC	✓	✓	–	MCMC	76.7	61.9	42.6

adds just $+3.3\%$ per-problem wall-clock to DEGS reranking and $+8.8\%$ to DEGS-MCMC over their likelihood-only counterparts (Table 6). The accuracy gains of Table 1 therefore come at single-digit-percent overhead—far below the near-doubling that a naive count of one extra forward per candidate would imply.

The signal is localizable in depth. Sweeping $\mathcal{L}_{\text{sub}}$ (full sweep in Table 7, Appendix E) separates two effects that "probe fewer layers" conflates—how *many* layers enter the grid versus *where* they sit. Count is almost free: thinning the default seven-layer grid down to five leaves MATH500 and GPQA accuracy unchanged, and probing all 28 layers buys only $+0.4$ point on MATH500 at $1.05\times$ cost. Placement, by contrast, is decisive: holding the count at four and sliding the window toward the input lowers accuracy monotonically—$76.6 \to 72.8$ on MATH500, with the same monotone decline on GPQA—until an input-only probe falls below even the $\beta{=}0$ baselines (Table 2). Because this ordering recurs across both out-of-domain GPQA and in-domain MATH500, it reflects *where* entropy collapse carries discriminative content rather than a benchmark quirk, and the probe can safely be restricted to those layers at no accuracy penalty.

5.4 Ablation Studies

Both terms are necessary, and the signal lives in the depth profile, not the final layer. DEGS-MCMC improves over likelihood-only power sampling ($\beta{=}0$) on all three benchmarks—up to $+4.5$ on DeepMind500 (Table 2)—and over depth-only scoring by a wide margin; neither term alone reproduces the combined result. The gain is not merely an entropy effect: final-layer-entropy reranking lands barely above Best-of-N on MATH500 yet trails DEGS-MCMC by up to 14 points on DeepMind500, so *when* a sequence's predictions stabilize is more informative than its final-layer entropy alone (Appendix D)—the distinction that motivates collapse depth. Added to either selection mode, the depth term yields a positive marginal gain over its matched likelihood-only baseline on all three benchmarks, largest in reranking on the harder DeepMind500 split, as Proposition 8 predicts; but applying the tiebreak once (reranking) versus at each of ~ 160 accept/reject steps (DEGS-MCMC) is what separates them—DEGS-MCMC dominates reranking on all three benchmarks, and reranking serves only as a controlled probe of *where* the signal acts (Table 7).

6 Discussion and Conclusion

We introduced DEGS, a training-free method that turns layer-wise entropy collapse depth into a test-time quality signal: stronger reasoners collapse later, which we operationalize as a pseudo-reward $D(\mathbf{x})$ tilting the power objective. DEGS does not merely favor uncertain sequences—the ablation (Table 2) confirms both terms are necessary.

Limitations and future work. Our evaluation centers on mathematical reasoning, with preliminary coding and scientific-QA results; generalization to open-ended or factual tasks remains open. DEGS also requires hidden-state access (open-weight models only), and the logit lens is an imperfect probe (Wendler et al., 2024). Promising extensions include richer entropy-curve features via learned scorers, token-selective collapse depth, cross-architecture validation, and integration with efficient samplers such as Power-SMC (Azizi et al., 2026).

AI USE STATEMENT

In this work, generative AI tools were used to aid or polish the writing of this paper. The authors reviewed and edited all AI-assisted text and take full responsibility for the final content of this work.

REPRODUCIBILITY STATEMENT

All experiments use publicly available open-weight base models (Qwen2.5-7B, Qwen2.5-Math-7B, DeepSeek-Math-7B) and public benchmarks (MATH500, DeepMind500, HumanEval, GPQA-Diamond). The collapse-depth definition (Definitions 1–2), the DEGS objective equation 4, and both decoding variants are specified in full in Section 4 and Algorithms 1–2. Appendix E reports the complete harness—per-backbone layer counts and probe grids, the prompt/parser/grader for each benchmark, decoding knobs, the entropy-probing procedure, the GRPO training recipe, and the matched-budget protocol—together with default hyperparameters and search ranges (Table 8) and the per-step MCMC protocol (Table 4); every prompt, parser, and grader is shared verbatim across all decoding methods. Formal statements and complete proofs of all theoretical claims are in Appendix F.

ETHICS STATEMENT

This work studies training-free decoding for reasoning on standard public benchmarks using open-weight models, and involves no new data collection or human subjects. It inherits the general risks of LLM reasoning systems (e.g., confidently stated incorrect answers) but, to our knowledge, introduces no additional ethical concerns; because DEGS requires hidden-state access, it applies only to open-weight models.

REFERENCES

Seyedarmin Azizi, et al. Power-SMC: Low-latency sequence-level power sampling for training-free LLM reasoning. *arXiv preprint arXiv:2602.10273*, 2026.

Nora Belrose, Zach Furman, Logan Smith, Danny Halawi, Igor Ostrovsky, Lev McKinney, Stella Biderman, and Jacob Steinhardt. Eliciting latent predictions from transformers with the tuned lens. *arXiv preprint arXiv:2303.08112*, 2023.

Bradley Brown, Jordan Juravsky, Ryan Ehrlich, Ronald Clark, Quoc V. Le, Christopher Ré, and Azalia Mirhoseini. Large language monkeys: Scaling inference compute with repeated sampling. *arXiv preprint arXiv:2407.21787*, 2024.

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.

Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James R. Glass, and Pengcheng He. DoLa: Decoding by contrasting layers improves factuality in large language models. In *International Conference on Learning Representations (ICLR)*, 2024.

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. *arXiv preprint arXiv:2110.14168*, 2021.

Ganqu Cui, Yuchen Zhang, Jiacheng Chen, Lifan Yuan, Zhi Wang, Yuxin Zuo, Haozhan Li, Yuchen Fan, Huayu Chen, Weize Chen, Zhiyuan Liu, Hao Peng, Lei Bai, Wanli Ouyang, Yu Cheng, Bowen Zhou, and Ning Ding. The entropy mechanism of reinforcement learning for reasoning language models. *arXiv preprint arXiv:2505.22617*, 2025.

Gonçalo R. A. Faria, Sweta Agrawal, Antonio Farinhas, Ricardo Rei, José G. C. de Souza, and André F. T. Martins. QuEST: Quality-aware Metropolis–Hastings sampling for machine translation. In *NeurIPS*, 2024.

Sebastian Farquhar, Jannik Kossen, Lorenz Kuhn, and Yarin Gal. Detecting hallucinations in large language models using semantic entropy. *Nature*, 630(8017):625–630, 2024.

Asma Ghandeharioun, Avi Caciularu, Adam Pearce, Lucas Dixon, and Mor Geva. Patchscopes: A unifying framework for inspecting hidden representations of language models. In *International Conference on Machine Learning (ICML)*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, et al. DeepSeek-R1: Incentivizing reasoning capability in LLMs via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025.

Andre 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 (EMNLP)*, pp. 25559–25571, 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. In *Advances in Neural Information Processing Systems (NeurIPS) Datasets and Benchmarks Track*, 2021.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations (ICLR)*, 2020.

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.

Saurav Kadavath, Tom Conerly, Amanda Askell, Tom Henighan, Dawn Drain, Ethan Perez, Nicholas Schiefer, Zac Hatfield-Dodds, Nova DasSarma, Eli Tran-Johnson, Scott Johnston, Sheer El-Showk, Andy Jones, Nelson Elhage, Tristan Hume, Anna Chen, Yuntao Bai, Sam Bowman, Stanislav Fort, Deep Ganguli, Danny Hernandez, Josh Jacobson, Jackson Kernion, Shauna Kravec, Liane Lovitt, Kamal Ndousse, Catherine Olsson, Sam Ringer, Dario Amodei, Tom Brown, Jack Clark, Nicholas Joseph, Ben Mann, Sam McCandlish, Chris Olah, and Jared Kaplan. Language models (mostly) know what they know. *arXiv preprint arXiv:2207.05221*, 2022.

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

Takeshi Kojima, Shixiang Shane Gu, Machel Reid, Yutaka Matsuo, and Yusuke Iwasawa. Large language models are zero-shot reasoners. In *NeurIPS*, 2022.

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

Alexander K. Lew, Tan Zhi-Xuan, Gabriel Grand, and Vikash K. Mansinghka. Sequential Monte Carlo steering of large language models using probabilistic programs. *arXiv preprint arXiv:2306.03081*, 2023.

Hunter Lightman, Vineet Kosaraju, Yura 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 (ICLR)*, 2024.

Michael Luo, Sijun Tan, Justin Wong, Xiaoxiang Shi, William Y. Tang, Manan Roongta, Colin Cai, Jeffrey Luo, Tianjun Zhang, Li Erran Li, Raluca Ada Popa, and Ion Stoica. DeepScaleR: Surpassing O1-Preview with a 1.5B model by scaling RL, 2025. Notion Blog. `https://github.com/agentica-project/rllm`.

Radford M. Neal. Probabilistic inference using Markov chain Monte Carlo methods. Technical report, University of Toronto, 1993.

Nostalgebraist. Interpreting GPT: The logit lens. *LessWrong*, 2020.

Long Ouyang, Jeffrey Wu, Xu Jiang, et al. Training language models to follow instructions with human feedback. In *NeurIPS*, volume 35, pages 27730–27744, 2022.

Mihir Prabhudesai, Lili Chen, Alex Ippoliti, Katerina Fragkiadaki, Hao Liu, and Deepak Pathak. Maximizing confidence alone improves reasoning. *arXiv preprint arXiv:2505.22660*, 2025.

Qwen: An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, et al. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*, 2025.

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 *Conference on Language Modeling (COLM)*, 2024.

David Saxton, Edward Grefenstette, Felix Hill, and Pushmeet Kohli. Analysing mathematical reasoning abilities of neural models. In *International Conference on Learning Representations (ICLR)*, 2019.

John Schulman, Filip Wolski, Prafulla Dhariwal, Alec Radford, and Oleg Klimov. Proximal policy optimization algorithms. *arXiv preprint arXiv:1707.06347*, 2017.

Tal Schuster, Adam Fisch, Jai Gupta, Mostafa Dehghani, Dara Bahri, Vinh Q. Tran, Yi Tay, and Donald Metzler. Confident adaptive language modeling. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2022.

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.

Rulin Shao, Shuyue Stella Li, Rui Xin, et al. Spurious rewards: Rethinking training signals in RLVR. *arXiv preprint arXiv:2506.10947*, 2025.

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.

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

Yuqiao Tan, Minzheng Wang, Shizhu He, Huanxuan Liao, Chengfeng Zhao, Qiunan Lu, Tian Liang, Jun Zhao, and Kang Liu. Bottom-up policy optimization: Your language model policy secretly contains internal policies. *arXiv preprint arXiv:2512.19673*, 2025.

Jonathan Uesato, Nate Kushman, Ramana Kumar, Francis Song, Noah Siegel, Lisa Wang, Antonia Creswell, Geoffrey Irving, and Irina Higgins. Solving math word problems with process- and outcome-based feedback. *arXiv preprint arXiv:2211.14275*, 2022.

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. In *ICLR*, 2023.

Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, Brian Ichter, Fei Xia, Ed H. Chi, Quoc V. Le, and Denny Zhou. Chain-of-thought prompting elicits reasoning in large language models. In *NeurIPS*, 2022.

Sean Welleck, Amanda Bertsch, Matthew Finlayson, Hailey Schoelkopf, Alex Xie, Graham Neubig, Ilia Kulikov, and Zaid Harchaoui. From decoding to meta-generation: Inference-time algorithms for large language models. *Transactions on Machine Learning Research (TMLR)*, 2024.

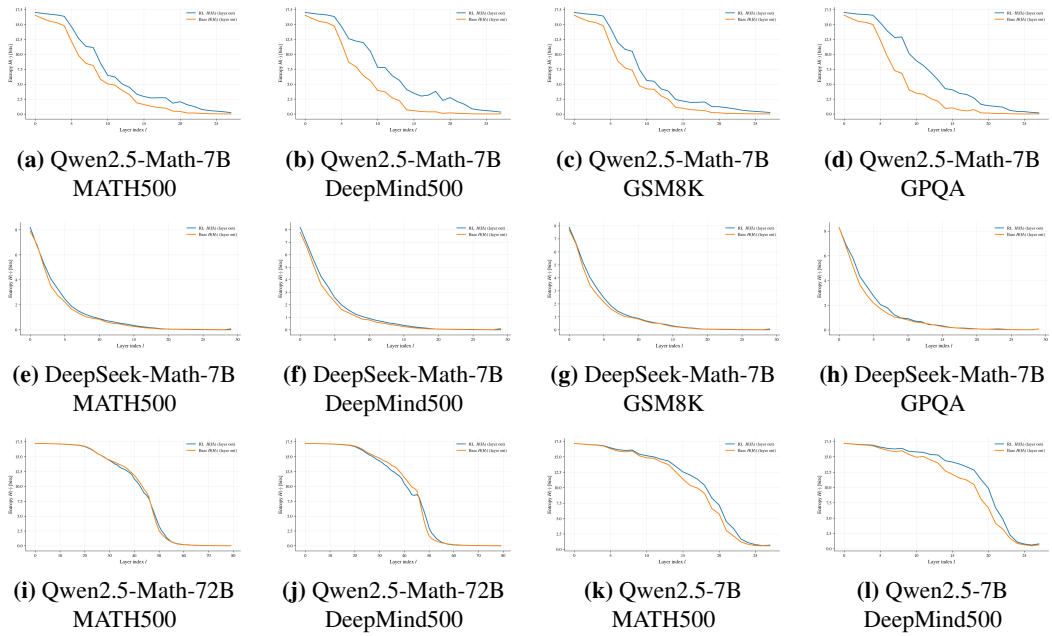
Chris Wendler, Veniamin Veselovsky, Giovanni Monea, and Robert West. Do Llamas work in English? On the latent language of multilingual transformers. In *Proceedings of ACL*, pages 15366–15394, 2024.

An Yang, Beichen Zhang, Binyuan Hui, Bofei Gao, Bowen Yu, Chengpeng Li, Dayiheng Liu, Jianhong Tu, Jingren Zhou, Junyang Lin, Keming Lu, Mingfeng Xue, Runji Lin, Tianyu Liu, Xingzhang Ren, and Zhenru Zhang. Qwen2.5-Math technical report: Toward mathematical expert model via self-improvement. *arXiv preprint arXiv:2409.12122*, 2024.

Yang Yue, Zhiqi Chen, Rui Lu, et al. Does reinforcement learning really incentivize reasoning capacity in LLMs beyond the base model? In *Advances in Neural Information Processing Systems (NeurIPS)*, 2025.

Xuandong Zhao, Zhewei Kang, Aosong Feng, Sergey Levine, and Dawn Song. Learning to reason without external rewards. *arXiv preprint arXiv:2505.19590*, 2025.

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.



(a) Qwen2.5-Math-7B MATH500 **(b)** Qwen2.5-Math-7B DeepMind500 **(c)** Qwen2.5-Math-7B GSM8K **(d)** Qwen2.5-Math-7B GPQA

(e) DeepSeek-Math-7B MATH500 **(f)** DeepSeek-Math-7B DeepMind500 **(g)** DeepSeek-Math-7B GSM8K **(h)** DeepSeek-Math-7B GPQA

(i) Qwen2.5-Math-72B MATH500 **(j)** Qwen2.5-Math-72B DeepMind500 **(k)** Qwen2.5-7B MATH500 **(l)** Qwen2.5-7B DeepMind500

Figure 3: Complete entropy-versus-layer visualizations across all twelve (model, benchmark) pairs. Each panel overlays the layer-wise token entropy H_l in nats for the posttrained variant against its base counterpart, measured on the subset where the posttrained model is correct and the base model is wrong; this isolates the depths at which posttraining adds discriminative content. This appendix figure complements Figure 1 by expanding coverage and reducing the possibility that the main-paper examples are selective.

A Additional Figures and Tables

B Preliminaries (Full)

Autoregressive language models. Let $\mathcal{X}$ denote a finite vocabulary. An autoregressive LLM defines $p(\mathbf{x}) = \prod_{t=0}^{T} p(x_t \mid x_{<t})$. In a transformer with L layers, hidden states $\mathbf{h}_{l,t} \in \mathbb{R}^d$ are updated via $\mathbf{h}_{l,t} = \mathbf{h}_{l-1,t} + f_l(\mathbf{h}_{l-1,1}, \ldots, \mathbf{h}_{l-1,t})$, where f_l is the l-th transformer block. After layer L, the unembedding matrix $U \in \mathbb{R}^{|\mathcal{X}| \times d}$ produces next-token logits.

Logit lens. The logit lens (Nostalgebraist, 2020) applies U at intermediate layers: $p_l(v \mid x_{\leq t}) = \mathrm{softmax}(U\mathbf{h}_{l,t})_v$. The layer-wise entropy is $H_l(t) = -\sum_v p_l(v \mid x_{\leq t}) \log p_l(v \mid x_{\leq t})$. Prior work shows entropy is high in early layers and drops sharply in later layers (Wendler et al., 2024).

Power distributions. The power distribution $p^\alpha(\mathbf{x}) \propto p(\mathbf{x})^\alpha$ ($\alpha \geq 1$) sharpens the base distribution. Karan and Du (2025) show that p^α differs from low-temperature sampling and propose MH sampling to target it.

C DEGS Reranking Algorithm

Algorithm 2 gives the depth-based reranking procedure referenced in Section 4.3. Given a problem prompt, sample N candidate solutions $\{\mathbf{x}^{(i)}\}_{i=1}^{N}$ from the base model p using moderate randomness (temperature $\in [0.7, 1.0]$, top-$p = 0.95$), recording the per-token log-probabilities. For each candidate, compute the log-likelihood from the recorded log-probs and the collapse depth $D(\mathbf{x}^{(i)})$ from one additional teacher-forced forward pass (logit-lens entropy at layers $\mathcal{L}_{\mathrm{sub}}$), then the combined score $S^{(i)} = \alpha \log p(\mathbf{x}^{(i)}) + \beta D(\mathbf{x}^{(i)})$, and return $\mathbf{x}^* = \arg\max_i S^{(i)}$. As discussed in Section 4.3, at N=16 this is not a competitive standalone decoder (Proposition 8); we retain it as a controlled probe of *where* in depth the collapse signal carries discriminative content (Table 7).

Table 3: Full benchmark results with all reported settings. Same data as Table 1, with the decoding budget reported as N=candidates, M=MCMC steps, T=temperature. Power-SMC is included as an additional training-free decoding baseline (Azizi et al., 2026).

Model	Method	Budget	MATH500 ↑	DeepMind500 ↑	HumanEval Pass@1 ↑	GPQA ↑
Qwen2.5-7B	Base	T=0	0.498	0.250	0.329	0.278
Qwen2.5-7B	Low-temperature	T=0.25	0.628	0.342	0.524	0.303
Qwen2.5-7B	Best-of-N	N=16	0.650	0.395	0.609	0.282
Qwen2.5-7B	MCMC Power Sampling	N=16, M=10	0.706	0.537	0.622	0.318
Qwen2.5-7B	Scalable Power Sampling	(T=0.25)	0.708	0.549	0.756	0.349
Qwen2.5-7B	Power-SMC	N=16	0.714	0.563	0.761	0.351
Qwen2.5-7B	Best-of-N-Entropy	N=16	0.664	0.412	0.632	0.348
Qwen2.5-7B	DEGS-MCMC	N=16, M=10, β=5	0.728	0.573	0.769	0.393
Qwen2.5-7B	GRPO (MATH)	T=0	0.740	0.598	0.561	0.354
Qwen2.5-Math-7B	Base	T=0	0.496	0.332	0.329	0.278
Qwen2.5-Math-7B	Low-temperature	T=0.25	0.690	0.482	0.512	0.353
Qwen2.5-Math-7B	Best-of-N	N=16	0.684	0.465	0.512	0.343
Qwen2.5-Math-7B	MCMC Power Sampling	N=16, M=10	0.748	0.574	0.573	0.389
Qwen2.5-Math-7B	Scalable Power Sampling	(T=0.25)	0.758	0.593	0.604	0.409
Qwen2.5-Math-7B	Power-SMC	N=16	0.762	0.601	0.612	0.413
Qwen2.5-Math-7B	Best-of-N-Entropy	N=16	0.687	0.479	0.537	0.369
Qwen2.5-Math-7B	DEGS-MCMC	N=16, M=10, β=5	0.767	0.619	0.618	0.426
Qwen2.5-Math-7B	GRPO (MATH)	T=0	0.785	0.632	0.537	0.399
DeepSeek-Math-7B	Base	T=0	0.362	0.212	0.415	0.333
DeepSeek-Math-7B	Low-temperature	T=0.25	0.366	0.269	0.427	0.430
DeepSeek-Math-7B	Best-of-N	N=16	0.420	0.276	0.433	0.338
DeepSeek-Math-7B	MCMC Power Sampling	N=16, M=10	0.424	0.307	0.470	0.345
DeepSeek-Math-7B	Scalable Power Sampling	(T=0.25)	0.464	0.314	0.487	0.364
DeepSeek-Math-7B	Power-SMC	N=16	0.467	0.329	0.498	0.386
DeepSeek-Math-7B	Best-of-N-Entropy	N=16	0.423	0.298	0.453	0.341
DeepSeek-Math-7B	DEGS-MCMC	N=16, M=10, β=5	0.475	0.336	0.508	0.392
DeepSeek-Math-7B	GRPO (MATH)	T=0	0.492	0.352	0.524	0.333

Algorithm 2 DEGS Reranking (Best-of-N)

Require: Base model p; prompt; candidate budget N; layer subset $\mathcal{L}_{\text{sub}}$; threshold τ; weights α, β
Ensure: Selected solution $\mathbf{x}^*$

Sample N candidates $\{\mathbf{x}^{(i)}\}_{i=1}^{N} \sim p(\cdot \mid \text{prompt})$; record log-probs
for $i = 1, \ldots, N$ **do**
  $\log p(\mathbf{x}^{(i)}) \leftarrow \sum_t \log p(x_t^{(i)} \mid x_{<t}^{(i)})$ {from sampling}
  Run forward pass on prompt $\oplus\, \mathbf{x}^{(i)}$ with hidden states at layers $\mathcal{L}_{\text{sub}}$
  for each layer $l \in \mathcal{L}_{\text{sub}}$ and position t **do**
    $H_l(t) \leftarrow -\sum_v p_l(v \mid x_{\le t}^{(i)}) \log p_l(v \mid x_{\le t}^{(i)})$ {logit-lens entropy}
  end for
  for each position t **do**
    $d_t \leftarrow \min\{l \in \mathcal{L}_{\text{sub}} : H_l(t) \le \tau\}$ (default L if none)
  end for
  $D(\mathbf{x}^{(i)}) \leftarrow \frac{1}{T^{(i)}+1} \sum_t d_t / L$
  $S^{(i)} \leftarrow \alpha \cdot \log p(\mathbf{x}^{(i)}) + \beta \cdot D(\mathbf{x}^{(i)})$
end for
return $\mathbf{x}^* \leftarrow \arg\max_i S^{(i)}$

D Design Choices and Practical Considerations

Relationship to existing signals. DEGS differs from final-layer entropy reranking in that it captures the *depth profile* of entropy evolution, not merely its endpoint. A candidate may have low final-layer entropy yet collapse early, reflecting superficial rather than deep reasoning. Table 2 isolates the contribution of each component.

Table 4: DEGS-MCMC protocol details: the practical setup of the entropy-guided MCMC variant, including proposal construction, chain length, acceptance computation, and stopping rules.

Component	Specification
Target distribution	$\pi(\mathbf{x}) \propto p(\mathbf{x})^{\alpha} \exp(\beta D(\mathbf{x}))$, with $\alpha{=}1/\tau_{\text{prop}}$ (default $\alpha{=}4$, $\tau_{\text{prop}}{=}0.25$) and $\beta{\geq}0$ (default $\beta{=}5$; $\beta{=}0$ recovers plain MCMC power sampling)
Proposal family	Local suffix resampling: at each MH step, draw a cut index i and regenerate all tokens from i to the end of the current completion with the base model at temperature τ_{prop} (multinomial, no top-k/top-p)
Proposal granularity	Token-level from a random cut point; the resampled span length is implicitly variable and is corrected for in the MH ratio
Editable span	Only the generated completion $[\lvert\text{prompt}\rvert, \lvert\text{gen}\rvert)$ is editable; the prompt is frozen, so the cut index is drawn uniformly from $\{\lvert\text{prompt}\rvert, \ldots, \lvert\text{gen}\rvert{-}1\}$
Initialization	Block-wise warm start: the chain is built incrementally over $B{=}16$ blocks of 192 tokens each, with each block sampled at temperature τ_{prop} before MH steps run
Chain length T_{MCMC}	$T_{\text{MCMC}}{=}10$ MH steps per block, $B{=}16$ blocks; total 160 proposals per sample at default
Burn-in / warm start	No explicit burn-in: block-wise warm start already places the chain in a high-likelihood region
Acceptance ratio	$\log r = \alpha\big(\log p(\mathbf{x}') - \log p(\mathbf{x})\big) + \big(\log q(\mathbf{x} \mid \mathbf{x}') - \log q(\mathbf{x}' \mid \mathbf{x})\big) + \beta\big(D(\mathbf{x}') - D(\mathbf{x})\big)$; accept with probability $\min(1, e^{\log r})$. The DEGS correction $\beta(D(\mathbf{x}') - D(\mathbf{x}))$ is additive and active only when $\beta{>}0$. Measured acceptance at the canonical setting ($\alpha{=}4$, $\beta{=}5$): deepseek_math/deepmind: 78%; qwen_math/math: 57%.
Likelihood evaluation	Single teacher-forced forward per proposal; base logprobs are summed over the completion without length normalization
Entropy probing for $D(\mathbf{x})$	Separate no-gradient forward with hidden-state hooks on $\mathcal{L}_{\text{sub}}$; at each layer the logit lens is applied after final RMSNorm, and exact token entropy is computed in vocab chunks of 8192; $D(\mathbf{x})$ is the mean normalized depth over generated (non-EOS) tokens
Caching / reuse strategy	Current state's $D(\mathbf{x})$ is cached and refreshed only when a new block is appended or a proposal is accepted; each proposal recomputes $D(\mathbf{x}')$ from scratch
Early stopping	Chain terminates when EOS appears; otherwise the full $B{\cdot}T_{\text{MCMC}}$ proposals are run
Selection from chain	Final chain state is returned (no chain averaging or best-so-far selection)
Budget normalization	Reported in forward-equivalent units as defined in Table 6

E Implementation Details

This appendix records the decoding, probing, training, and evaluation configuration behind Section 5. The tables in this section are referenced from the main text; the prose below specifies what the tables do not, so that every reported number is reproducible from a single fixed harness.

Base models. All experiments use three open-weight base checkpoints: Qwen2.5-7B and Qwen2.5-Math-7B, each with $L{=}28$ transformer layers, and DeepSeek-Math-7B with $L{=}30$ layers. Because collapse depth is read off a fixed grid of layer indices, the default probe $\mathcal{L}_{\text{sub}}$ is rescaled to each backbone's depth— $\{4, 8, 12, 16, 20, 24, 28\}$ for the two 28-layer Qwen models and $\{4, 9, 14, 18, 22, 26, 30\}$ for the 30-layer DeepSeek model—so that the seven probed layers occupy comparable *relative* depths l/L across architectures rather than a fixed absolute index.

Benchmarks, prompts, and grading. Table 5 gives the output format and answer-processing pipeline for each benchmark; the governing principle is that the harness is held fixed while only the decoder varies. MATH500 and DeepMind500 share an identical zero-shot chain-of-thought prompt, parser, and grader, with the final answer read from the last `\boxed{}` span and scored by a sympy-based symbolic/numeric matcher; the two splits therefore differ only in problem difficulty, which is what licenses reading the larger DEGS gains on DeepMind500 (Section 5.2) as a difficulty effect rather than a harness artifact. GPQA (Diamond) is posed as a four-way multiple-choice task whose option order is shuffled once and held fixed across methods—so every decoder is scored against the same gold position—with the predicted letter read from the boxed answer, matched exactly against the gold letter, and completions capped at 3072 tokens. HumanEval is generated by direct completion from the function signature, without chain-of-thought, and graded for functional

correctness (Pass@1) by the official test harness. Every prompt template, parser, and grader is shared verbatim across all decoding methods and all models, so that accuracy differences between rows of Table 1 isolate the decoder.

Decoding configuration. Each method samples from the same base model under a shared set of decoding knobs. We disable nucleus (Holtzman et al., 2020) and top-k truncation throughout (top-p=1.0): the power-sampling family targets the untruncated base distribution, and matching this across baselines keeps the comparison on equal footing. The likelihood exponent is realized as an inverse sampling temperature ($\alpha=1/T$), so the default $\alpha=4$ corresponds to $T=0.25$ on MATH500, DeepMind500, and GPQA, whereas HumanEval is decoded at $T=0.5$ ($\alpha=2$) to retain the diversity program synthesis requires. The reranking methods (Best-of-N, Best-of-N-Entropy, and DEGS reranking) draw $N=16$ candidates and return one; the MCMC family (power sampling and DEGS-MCMC) runs $T_{\mathrm{MCMC}}=10$ Metropolis–Hastings steps in each of $B=16$ blocks of 192 tokens, i.e. 160 proposals per sample. All DEGS variants use depth weight $\beta=5$ and their likelihood-only counterparts $\beta=0$, so that any DEGS-versus-baseline difference is attributable solely to the depth-entropy term. Defaults and search ranges are collected in Table 8.

Collapse-depth probing. Scoring a completion with $D(\mathbf{x})$ adds one teacher-forced forward pass that caches hidden states at the layers in $\mathcal{L}_{\mathrm{sub}}$. At each probed layer the model's final RMSNorm is applied before the logit lens—matching the normalization used at the output head—and exact token entropy is computed over the full vocabulary in chunks of 8192 entries to bound peak memory. The per-token collapse depth d_t is the shallowest probed layer whose logit-lens entropy is at or below $\tau=0.25$ nats ($d_t=L$ if none qualifies), and $D(\mathbf{x})$ averages d_t/L over all generated tokens, excluding EOS and scoring the full completion span. In DEGS-MCMC the current state's $D(\mathbf{x})$ is cached and refreshed only when a block is appended or a proposal is accepted, whereas every proposal recomputes $D(\mathbf{x}')$ from scratch; the complete per-step protocol, including the acceptance ratio and stopping rule, is given in Table 4.

Predictive-validity sampling. The analysis of whether $D(\mathbf{x})$ predicts correctness (Figure 2) deliberately uses a higher-entropy regime than the decoding runs—$N=16$ candidates per problem at $T=0.8$, top-p=0.9—so that the candidate pool spans a wide range of $D(\mathbf{x})$ and the correlation is not estimated on a near-degenerate set of near-identical samples.

GRPO reference. Following Karan and Du (2025), our GRPO references are produced rather than quoted: we posttrain each base model with GRPO (Shao et al., 2024) and evaluate the resulting policy through the same harness as every other method. The training setup mirrors theirs—we adopt the GRPO implementation of Shao et al. (2025) with its default hyperparameters, optimize against a ground-truth math verifier on the MATH training split, and use a group size of 16 rollouts per prompt. For the two Qwen backbones this configuration coincides with Karan and Du (2025); DeepSeek-Math-7B is outside their study, so its GRPO reference is trained by us under the same recipe. Decoding is greedy throughout, so that GRPO functions as an in-domain reference obtained under matched conditions.

Sharding and compute. Generation is distributed over a heterogeneous GPU cluster of RTX 5880 and RTX 3090 nodes, one worker per GPU; the shards—five for MATH500 and DeepMind500, six for GPQA, and four for HumanEval—are a throughput split over the problem set only. Hardware is split by experiment: the main accuracy results are measured on the RTX 5880 nodes, whereas the compute-efficiency sweep behind Table 7 and Table 6 runs on a single RTX 3090—a difference that moves wall-clock alone and not accuracy, since the decoder and harness are identical across nodes. Test-time compute is normalized to forward-equivalent units as defined in Table 6, under which each reranking and MCMC method is matched against its same-family baseline; the wall-clock figures in that table are per-problem means on Qwen2.5-Math-7B / MATH500, and the corresponding per-problem mean on GPQA under the MCMC setting is approximately 0.63 ks.

Table 5: Benchmarks, metrics, and evaluation setup, with output format and answer-processing details for each benchmark.

Benchmark	Task type	Size	Primary metric	Output format	Evaluation / answer-processing setup
MATH500	Math reasoning	500	Accuracy	CoT + `\boxed{}`	Zero-shot CoT prompt; answer parsed from last `\boxed{}` span; graded by sympy-based symbolic/numeric matcher; temperature 0.25
DeepMind500	Math reasoning	500	Accuracy	CoT + `\boxed{}`	Held-out subset we construct from the DeepMind mathematics dataset; identical prompt, parser, and grader as MATH500; temperature 0.25
HumanEval	Code generation	164	Pass@1	Python function	Direct completion from function signature (no CoT); code extracted via regex; graded by official functional-correctness harness (per-test timeout 3 s); temperature 0.5
GPQA (Diamond)	Scientific QA	198	Accuracy	MC (A/B/C/D) + CoT	Zero-shot CoT MC prompt; letter extracted from final `\boxed{}` span; temperature 0.25

Table 6: Compute accounting and matched-budget protocol: how test-time compute is normalized across methods. All MCMC-family methods use $B{=}16$ blocks with $T_{\mathrm{MCMC}}{=}10$ MH steps per block (160 total proposals), and all layer-subset probing is charged as K/L of a full forward pass per probed candidate (default $K{=}7$, $L{\in}\{28, 30\}$). The wall-clock column reports per-problem mean elapsed seconds on Qwen2.5-Math-7B / MATH500 measured during the compute-efficiency sweep reported in Table 1.

Method	Candidate budget	Extra forward passes	Hidden-state probing	MCMC steps	Budget unit	Matched-budget note	Wall-clock
Base decoding	1 sample	0	No	0	$1\times$ decode pass	Single-sample reference	–
Low-temperature decoding	$N{=}16$ samples	0	No	0	$N\times$ decode passes	Matched by candidate count	–
Best-of-N (log-likelihood)	$N{=}16$ samples	0	No	0	N decode passes	Same candidate set as DEGS rerank	228.1 s
Best-of-N-Entropy	$N{=}16$ samples	N	Yes (final layer)	0	$N{+}N$ forward-equivalents	Same candidate pool and rerank budget	230.5 s
MCMC Power Sampling / SPS	$B{\cdot}T_{\mathrm{MCMC}}{=}160$ proposals	0	No	$T_{\mathrm{MCMC}}{=}10$/block	Total proposal forwards	Matched by forward-equivalent cost	330.5 s
Power-SMC	Particle-dependent	0	No	$T_{\mathrm{MCMC}}{=}10$/block	Total proposal forwards	Matched against MCMC family	–
DEGS reranking	$N{=}16$ samples	N ($\mathcal{L}_{\mathrm{aub}}$)	Yes ($\lvert\mathcal{L}_{\mathrm{aub}}\rvert{=}7$)	0	$N{+}N{\cdot}(K/L)$ fwd-eq.	Matched against rerank baselines	235.6 s
DEGS-MCMC	$B{\cdot}T_{\mathrm{MCMC}}{=}160$ proposals	1 probe/MH step ($\mathcal{L}_{\mathrm{aub}}$)	Yes ($\lvert\mathcal{L}_{\mathrm{aub}}\rvert{=}7$)	$T_{\mathrm{MCMC}}{=}10$/block	Proposals $+ B{\cdot}T_{\mathrm{MCMC}}{\cdot}(K/L)$ probes	Matched against MCMC family	359.5 s

F Proofs and Derivations

This appendix collects formal statements and proofs for the claims used in the main text. Throughout, p is the base model, $D(\mathbf{x}) \in (0, 1]$ is the normalized collapse depth (Definition 2), and $\pi(\mathbf{x}) \propto p(\mathbf{x})^{\alpha} \exp(\beta D(\mathbf{x}))$ is the DEGS target (Definition 3) with $\alpha \geq 1$, $\beta \geq 0$.

F.1 Normalizability of the DEGS target

Proposition 1 (Normalizability)**.** *Let the support $\mathcal{S} = \{\mathbf{x} : p(\mathbf{x}) > 0\}$ be finite (e.g. sequences of bounded length over a finite vocabulary). Then $Z = \sum_{\mathbf{x}\in\mathcal{S}} p(\mathbf{x})^{\alpha} \exp(\beta D(\mathbf{x}))$ satisfies $0 < Z < \infty$, so $\pi(\mathbf{x}) = p(\mathbf{x})^{\alpha} \exp(\beta D(\mathbf{x}))/Z$ is a well-defined probability distribution.*

Proof. Each term is nonnegative, and positive for at least one $\mathbf{x}$ (any $\mathbf{x} \in \mathcal{S}$), so $Z > 0$. Since $D(\mathbf{x}) \leq 1$, each term is bounded by $p(\mathbf{x})^{\alpha} e^{\beta}$; summing over the finite set $\mathcal{S}$ gives $Z \leq e^{\beta} \sum_{\mathbf{x}\in\mathcal{S}} p(\mathbf{x})^{\alpha} \leq e^{\beta} \sum_{\mathbf{x}\in\mathcal{S}} p(\mathbf{x}) = e^{\beta} < \infty$, using $\alpha \geq 1$ and $p(\mathbf{x}) \leq 1$. Hence $0 < Z < \infty$. □

F.2 Convergence of DEGS-MCMC

Theorem 1 (Convergence to the DEGS target)**.** *Consider the Metropolis–Hastings chain with target π and the random-index resampling proposal q of Karan and Du (2025), using the acceptance ratio equation 6. If q is irreducible and aperiodic on $\mathcal{S}$, then the chain has π as its unique stationary distribution and the marginal law of the n-th state converges to π in total variation as $n \to \infty$.*

Proof. The acceptance ratio equation 6 is exactly the Metropolis–Hastings ratio for target π, because the unknown normalizer Z cancels:

$$\frac{\pi(\mathbf{x}')\, q(\mathbf{x} \mid \mathbf{x}')}{\pi(\mathbf{x})\, q(\mathbf{x}' \mid \mathbf{x})} = \frac{p(\mathbf{x}')^{\alpha} \exp(\beta D(\mathbf{x}'))\, q(\mathbf{x} \mid \mathbf{x}')}{p(\mathbf{x})^{\alpha} \exp(\beta D(\mathbf{x}))\, q(\mathbf{x}' \mid \mathbf{x})}.$$

Table 7: Accuracy–overhead trade-off for the layer subset $\mathcal{L}_{\text{sub}}$: representative layer-subset choices for entropy probing with their measured wall-clock cost (per-problem mean, averaged over both (model, benchmark) evaluation settings used in Table 6) and downstream accuracy. The DeepMind500 column is reported on DeepSeek-Math-7B; see Table 6 for the matched-budget protocol.

$\mathcal{L}_{\text{sub}}$	K	Relative overhead	MATH500 ↑	GPQA ↑	DeepMind500 ↑
All layers $\{1, \ldots, L\}$	28	1.05×	0.771	0.426	0.339
$\{4, 8, 12, 16, 20, 24, 28\}$	7	1.00×	0.767	0.426	0.336
$\{8, 12, 16, 20, 24, 28\}$	6	0.98×	0.767	0.426	0.336
$\{12, 16, 20, 24, 28\}$	5	0.98×	0.767	0.426	0.336
$\{16, 20, 24, 28\}$	4	0.99×	0.766	0.426	0.334
$\{12, 16, 20, 24\}$	4	0.98×	0.765	0.424	0.334
$\{8, 12, 16, 20\}$	4	0.99×	0.762	0.421	0.321
$\{4, 8, 12, 16\}$	4	0.99×	0.759	0.418	0.320
$\{1, 4, 8, 12\}$	4	0.98×	0.746	0.403	0.306
$\{1, 2, 3, 4\}$	4	0.98×	0.728	0.385	0.275

Table 8: Full DEGS hyperparameters and search ranges. The likelihood exponent α is implemented as $1/\text{temperature}$ inside the power-sampling objective.

Parameter	Default	Search range
Likelihood exponent α	4	$\{2,\ 4,\ 6,\ 8\}$
Depth weight β	5	$\{0,\ 1,\ 2,\ 5,\ 10\}$
Entropy threshold τ (nats)	0.25	$\{0.05,\ 0.10,\ \ldots,\ 0.50\}$
$\mathcal{L}_{\text{sub}}$ (Qwen, L=28)	$\{4, 8, 12, 16, 20, 24, 28\}$	$K \in \{4, 5, 6, 7\}$ + shifted
$\mathcal{L}_{\text{sub}}$ (DeepSeek, L=30)	$\{4, 9, 14, 18, 22, 26, 30\}$	Depth-matched rescaling
Candidates N	16	$\{4,\ 8,\ 16,\ 32\}$
MCMC blocks B	16	fixed
MCMC steps/block T_{MCMC}	10	$\{1, 2, 5, 10, 20, 40\}$
Block length	192 tokens	fixed
Sampling temperature	0.25; 0.5 (HumanEval)	α=$1/T$
Top-p	1.0	fixed
Scoring span	Completion (excl. prompt)	{full, compl., reasoning, answer, numeric}
Apply final RMSNorm	Yes	fixed
Exclude EOS from entropy avg.	Yes	fixed
Vocab chunk size	8192	fixed

By Proposition 1, π is a well-defined distribution, and standard Metropolis–Hastings theory (Neal, 1993) guarantees that the chain satisfies detailed balance with respect to π. Detailed balance makes π stationary; irreducibility and aperiodicity of the proposal (inherited by the accepted chain, since the acceptance probability is positive wherever π is) make the stationary distribution unique and yield convergence in total variation by the ergodic theorem for Markov chains. □

F.3 Power vs. temperature sampling

Proposition 2 (Power sampling is not temperature sampling)**.** *Let $p(\mathbf{x}) = \prod_{t=1}^{T} p(x_t \mid x_{<t})$ be autoregressive. The sequence-level power distribution $p^{\alpha}(\mathbf{x}) \propto p(\mathbf{x})^{\alpha}$ is in general* not *equal to the autoregressive distribution obtained by temperature sampling at temperature $1/\alpha$, i.e. the model $r(\mathbf{x}) = \prod_t r(x_t \mid x_{<t})$ with $r(x_t \mid x_{<t}) \propto p(x_t \mid x_{<t})^{\alpha}$, except in degenerate cases.*

Proof. Temperature sampling renormalizes *per step*: $r(x_t \mid x_{<t}) = p(x_t \mid x_{<t})^{\alpha} / Z_t(x_{<t})$ with $Z_t(x_{<t}) = \sum_v p(v \mid x_{<t})^{\alpha}$. Hence

$$r(\mathbf{x}) = \prod_{t=1}^{T} \frac{p(x_t \mid x_{<t})^{\alpha}}{Z_t(x_{<t})} = \frac{p(\mathbf{x})^{\alpha}}{\prod_{t=1}^{T} Z_t(x_{<t})}.$$

The sequence-level power distribution instead uses a single global normalizer, $p^{\alpha}(\mathbf{x}) = p(\mathbf{x})^{\alpha}/Z$ with $Z = \sum_{\mathbf{x}} p(\mathbf{x})^{\alpha}$. These coincide only if $\prod_t Z_t(x_{<t})$ is the same constant Z for every $\mathbf{x}$ in

the support, which fails whenever the per-step normalizers $Z_t(x_{<t})$ depend on the prefix $x_{<t}$—the generic case for a non-uniform model. Thus the two distributions differ in general. □

F.4 Why collapse depth concentrates on hard tokens

Proposition 3 (Concentrated futures collapse early)**.** *Fix a token position t and consider its logit-lens distributions $p_l(\cdot \mid x_{\le t})$ across layers l. Suppose at some layer l_0 the distribution is already concentrated, $\max_v p_{l_0}(v \mid x_{\le t}) \ge 1-\epsilon$ for small $\epsilon \in (0,1)$. Then the logit-lens entropy at l_0 satisfies $H_{l_0}(t) \le H_b(\epsilon) + \epsilon \log(|\mathcal{X}|-1)$, where H_b is the binary entropy function. In particular, for small ϵ the entropy is below a modest threshold τ, so $d_t \le l_0$.*

Proof. This is Fano's inequality applied to the logit-lens distribution. Let $v^\star = \arg\max_v p_{l_0}(v \mid x_{\le t})$ with mass $1-\delta$ for some $\delta \le \epsilon$. Writing the entropy as that of a mixture between $v^\star$ and the remaining mass,

$$H_{l_0}(t) = H_b(\delta) + \delta \cdot H\big(p_{l_0}(\cdot \mid x_{\le t},\, v \ne v^\star)\big) \le H_b(\delta) + \delta \log(|\mathcal{X}|-1),$$

since the conditional distribution over the other $|\mathcal{X}|-1$ tokens has entropy at most $\log(|\mathcal{X}|-1)$. Both $H_b(\delta)$ and $\delta\log(|\mathcal{X}|-1)$ are nondecreasing in δ on $[0,1/2]$, so the bound holds with δ replaced by ϵ. For ϵ small enough that $H_b(\epsilon) + \epsilon\log(|\mathcal{X}|-1) \le \tau$, we get $H_{l_0}(t) \le \tau$, hence $d_t = \min\{l \in \mathcal{L}_{\text{sub}} : H_l(t) \le \tau\} \le l_0$. □

F.5 Reranking approximates the DEGS target

Proposition 4 (Self-normalized importance sampling)**.** *Let $\{\mathbf{x}^{(i)}\}_{i=1}^N$ be drawn i.i.d. from a proposal g with $g(\mathbf{x}) > 0$ on $\mathcal{S}$. Define importance weights $w_i \propto \pi(\mathbf{x}^{(i)})/g(\mathbf{x}^{(i)})$, normalized to sum to one. Then for any bounded test function h, the self-normalized estimator $\sum_i w_i h(\mathbf{x}^{(i)})$ converges almost surely to $\mathbb{E}_\pi[h]$ as $N \to \infty$. Choosing $g = p$ gives unnormalized weights $w_i \propto p(\mathbf{x}^{(i)})^{\alpha-1}\exp(\beta D(\mathbf{x}^{(i)}))$.*

Proof. Self-normalized importance sampling is consistent whenever $\text{supp}(\pi) \subseteq \text{supp}(g)$ (Neal, 1993). The unnormalized weight is $\pi(\mathbf{x})/g(\mathbf{x}) \propto p(\mathbf{x})^\alpha \exp(\beta D(\mathbf{x}))/p(\mathbf{x}) = p(\mathbf{x})^{\alpha-1}\exp(\beta D(\mathbf{x}))$ when $g = p$, and the global constant Z cancels under normalization. Almost-sure convergence follows from the strong law of large numbers applied to numerator and denominator separately, both of which have finite expectation since h is bounded and the weights are integrable on the finite support $\mathcal{S}$. DEGS reranking returns the maximizer $\arg\max_i[\alpha \log p(\mathbf{x}^{(i)}) + \beta D(\mathbf{x}^{(i)})]$, the mode of the reweighted set; the same weights underlie the full self-normalized estimator. □

F.6 Cost accounting

Proposition 5 (Probing overhead)**.** *Let a full forward pass over a length-T sequence cost C_{fwd} (dominated by L transformer layers). Computing $D(\mathbf{x})$ by caching hidden states at $K = |\mathcal{L}_{\text{sub}}|$ layers and applying the logit lens adds entropy computations at those K layers only. The marginal cost of probing is $C_{\text{probe}} = K \cdot c_{\text{lens}}$, where c_{lens} is the per-layer logit-lens-plus-entropy cost, independent of L; relative to a generation forward pass that already computes all L layers, probing reuses the same hidden states and adds only the lens/entropy arithmetic.*

Proof. The forward pass computes hidden states $\mathbf{h}_{l,t}$ at all layers regardless of probing. Probing reads the cached states at the K layers in $\mathcal{L}_{\text{sub}}$ and, for each, applies the unembedding and computes entropy, costing c_{lens} per layer (an $O(|\mathcal{X}|d)$ matrix-vector product plus an $O(|\mathcal{X}|)$ entropy sum, optionally chunked over the vocabulary). Summing over the K probed layers gives $C_{\text{probe}} = K\, c_{\text{lens}}$, with no dependence on the total depth L beyond $K \le L$. The teacher-forced scoring pass is a single additional forward, so the total scoring cost is $C_{\text{fwd}} + C_{\text{probe}}$. □

Proposition 6 (Token budget of DEGS-MCMC)**.** *With block size B, sequence length T, and N_{MCMC} Metropolis–Hastings steps per block, the expected number of generated tokens is $\Theta(N_{\text{MCMC}}T^2/B)$ up to the per-step resampling fraction, matching the accounting of Karan and Du (2025).*

Proof. There are T/B blocks. Within block k, the state has length $\Theta(kB)$, and a random-index proposal resamples on average half the suffix from the chosen cut point, i.e. $\Theta(kB)$ tokens per accepted/attempted proposal. Running N_{MCMC} steps per block and summing over $k = 1, \ldots, T/B$ gives $\sum_{k=1}^{T/B} N_{\mathrm{MCMC}}\,\Theta(kB) = N_{\mathrm{MCMC}}\,\Theta\big(B \cdot \frac{(T/B)^2}{2}\big) = \Theta(N_{\mathrm{MCMC}} T^2/B)$. This reproduces the quadratic-in-T cost of block-wise power sampling. □

F.7 Entropy decomposition for the depth signal

Definition 4 (Layer-conditional entropy contribution)**.** *For token t, define the inter-layer change in logit-lens belief by the KL divergence $\Delta_l(t) = \mathrm{KL}\big(p_l(\cdot \mid x_{\le t}) \,\|\, p_{l-1}(\cdot \mid x_{\le t})\big)$ between consecutive layers $l-1$ and l.*

Proposition 7 (Collapse depth tracks belief stabilization)**.** *If the logit-lens beliefs stabilize after layer $l^\star$ in the sense that $\Delta_l(t) \le \eta$ for all $l > l^\star$ and some small η, and the entropy $H_l(t)$ is non-increasing for $l > l^\star$, then the per-token collapse depth satisfies $d_t \le l^\star$ whenever $H_{l^\star}(t) \le \tau$. Consequently $D(\mathbf{x})$ is governed by the depth at which beliefs stabilize, not by post-stabilization fluctuations.*

Proof. By definition $d_t = \min\{l \in \mathcal{L}_{\mathrm{sub}} : H_l(t) \le \tau\}$. If $H_{l^\star}(t) \le \tau$ then the minimizing layer is at most $l^\star$, so $d_t \le l^\star$. The stabilization condition $\Delta_l(t) \le \eta$ for $l > l^\star$ ensures the beliefs (and hence the entropy) change negligibly beyond $l^\star$, so the threshold crossing that defines d_t is determined at or before $l^\star$; later near-constant layers cannot move the crossing earlier or later. Averaging d_t/L over tokens, $D(\mathbf{x})$ depends on the stabilization depths $\{l_t^\star\}$ rather than on fluctuations in deeper layers. □

F.8 Bounded perturbation of the likelihood ranking

Proposition 8 (Depth term is a bounded reranking perturbation)**.** *Fix two candidate sequences $\mathbf{x}, \mathbf{x}'$. Under the DEGS score $S(\mathbf{x}) = \alpha \log p(\mathbf{x}) + \beta D(\mathbf{x})$ with $D(\mathbf{x}) \in (0, 1]$, the depth term can change the score gap by at most β in absolute value: $\big|[S(\mathbf{x}') - S(\mathbf{x})] - \alpha[\log p(\mathbf{x}') - \log p(\mathbf{x})]\big| = \beta\,|D(\mathbf{x}') - D(\mathbf{x})| \le \beta$. Hence if $\alpha|\log p(\mathbf{x}') - \log p(\mathbf{x})| > \beta$, the DEGS ranking of $\mathbf{x}, \mathbf{x}'$ agrees with the pure-likelihood ranking; the depth signal can only reorder candidates whose α-scaled log-likelihoods lie within β of each other.*

Proof. By definition $S(\mathbf{x}') - S(\mathbf{x}) = \alpha[\log p(\mathbf{x}') - \log p(\mathbf{x})] + \beta[D(\mathbf{x}') - D(\mathbf{x})]$, so the deviation from the pure-likelihood gap is exactly $\beta[D(\mathbf{x}') - D(\mathbf{x})]$. Since $D(\mathbf{x}), D(\mathbf{x}') \in (0, 1]$, their difference lies in $(-1, 1)$, giving the bound $\beta|D(\mathbf{x}') - D(\mathbf{x})| \le \beta$. If $\alpha[\log p(\mathbf{x}') - \log p(\mathbf{x})] > \beta$, then $S(\mathbf{x}') - S(\mathbf{x}) \ge \alpha[\log p(\mathbf{x}') - \log p(\mathbf{x})] - \beta > 0$, preserving the likelihood order; symmetrically for the reverse inequality. Thus reordering requires $\alpha|\log p(\mathbf{x}') - \log p(\mathbf{x})| \le \beta$. □

Remark 1 (Interpretation)**.** *Proposition 8 formalizes the design intent of DEGS: the depth-entropy term acts as a* tiebreaker *among candidates of comparable likelihood rather than overriding the base model's likelihood signal. With the default $\beta = 5$ and $\alpha = 4$, two candidates are reordered only if their log-likelihoods differ by at most $\beta/\alpha = 1.25$ nats—a small gap on the scale of full-sequence log-likelihoods—so DEGS preserves the strong likelihood ranking almost everywhere and intervenes only where the base model is itself nearly indifferent. This is consistent with the weak per-candidate AUC of collapse depth (Section 3) coexisting with a consistent decoding gain: a near-chance signal, applied only at genuine likelihood ties and compounded over many MCMC steps, shifts the sampled distribution without destabilizing it.*

Remark 2 (Scope of the theory)**.** *The results above are deliberately modest: Propositions 1–6 establish that the DEGS target is well-defined, that DEGS-MCMC targets it correctly, and that the probing overhead is depth-independent, while Proposition 8 bounds the influence of the depth term. None of these results claims that larger $D(\mathbf{x})$* causes *higher correctness; that link is the empirical hypothesis tested in Section 5 (Figure 2), and the theory only delimits how a weak correctness signal, once assumed, is incorporated into sampling without distorting the likelihood objective beyond a controlled margin.*