Early-Exit Power Sampling with Delayed Acceptance: Leveraging Mid-Layer Computation for Efficient Reasoning

Khash, Ganni

Abstract

Power-distribution sampling (*a.k.a.* sampling from $p_\alpha \propto p^\alpha$ with $\alpha > 1$) has recently shown strong reasoning gains using blockwise Metropolis–Hastings (MH), but at the cost of many full forward passes through all layers of a large language model (LLM). A growing body of evidence suggests that mid layers already encode most task-relevant information, while later layers largely refine and sharpen predictions. Motivated by these observations, we propose *Early-Exit Power Sampling with Delayed Acceptance* (EE-PS): a blockwise MH sampler for $p_\alpha(\cdot)$ that replaces most likelihood evaluations with truncated *mid-layer* passes and uses a calibrated *mid-layer head* to cheaply score proposals. To preserve exactness with respect to p_α, EE-PS employs a standard two-stage *delayed-acceptance* (DA) correction: *every* proposal that passes the cheap test is immediately verified with a full-model acceptance step. We outline the method, discuss calibration and gating to keep corrections rare, analyze compute including KV-cache effects, and propose an evaluation plan on math and code benchmarks. The goal is to retain the quality gains of power sampling while cutting FLOPs relative to full-depth MH.

1 Motivation

Recent depth analyses indicate a "guess-then-refine" pattern in decoder-only Transformers: early/mid layers form competitive token predictions, whereas later layers mostly sharpen or re-rank them. Mid-layer readouts (via tuned/logit-lens probes) often approximate the final head well after simple calibration. In parallel, *power sampling* improves reasoning by exploring a sharpened joint distribution using blockwise MH with resample-the-tail proposals. However, the method's repeated full-depth likelihood evaluations are expensive. We hypothesize that accurate *relative* scoring of MH proposals can be achieved using early exit at a mid layer, with occasional but principled full corrections to maintain the exact target.

2 Preliminaries

Model. Consider a pre-LayerNorm Transformer with residual stream $h_l \in \mathbb{R}^{T\times d}$:

$$a_l = \mathrm{SelfAttention}_l(\mathrm{Norm}(h_l)), \quad \hat{h}_l = h_l + a_l, \; m_l \quad = \mathrm{MLP}_l(\mathrm{Norm}(\hat{h} * l)), h * l + 1 \quad = \hat{h} * l + m_l.$$

Let $W^{\mathrm{out}} \in \mathbb{R}^{d\times|\mathcal{V}|}$ be the LM head. For an early-exit layer $L^*! \in !1, \ldots, L$, we define a calibrated mid-layer head

$$\tilde{z} * t = \mathrm{Norm}(h * L^*, t), A + b, \qquad \tilde{p}(x_t \mid x * < t) = \mathrm{softmax}(\tau^{-1}\tilde{z}_t), \tag{1}$$

where (A, b, τ) are fit on a small held-out set. We consider (i) temperature scaling $A = W^{\mathrm{out}}$, (ii) an affine probe (*tuned lens*) trained by ridge regression.

Power target and blockwise MH. Let $p(x_{1:T})$ be the base model likelihood for a sequence $x_{1:T}! \in! \mathcal{V}^T$. Power sampling targets

$$p_\alpha(x_{1:T}); \propto; p(x_{1:T})^\alpha, \qquad \alpha > 1, \tag{2}$$

implemented via blockwise MH over contiguous blocks of size B with *resample-the-tail* proposals from the base model at temperature $1/\alpha$.

3 Method: EE-PS

3.1 Two-stage delayed-acceptance MH (DA-MH)

For current sequence x and proposal x', the full MH acceptance targeting $p_\alpha(\cdot)$ is

$$A(x! \to! x'); =; \min! \left(1, ; \frac{p_\alpha(x'), q(x \mid x')}{p_\alpha(x), q(x' \mid x)}\right). \tag{3}$$

EE-PS introduces a surrogate (mid-layer) target $\tilde{p} * \alpha \propto \tilde{p}^\alpha$ for *screening*. The DA-MH acceptance factorizes into two stages:

$$\tilde{A}(x! \to! x') = \min! \left(1, ; \frac{\tilde{p} * \alpha(x'), q(x \mid x')}{\tilde{p} * \alpha(x), q(x' \mid x)}\right), \; A^\star(x! \to! x') \quad = \min! \left(1, ; \frac{p_\alpha(x'), \tilde{p} * \alpha(x)}{p_\alpha(x), \tilde{p}_\alpha(x')}\right). \tag{4}$$

Algorithmic rule: draw $U! \sim! \text{Unif}[0, 1]$. If $U > \tilde{A}$, *reject* using only the truncated pass. If $U \le \tilde{A}$, *immediately* perform the full-depth evaluation and accept with probability $A^\star$.

Proposition 1 (Correctness)**.** *Assuming the proposal $q(\cdot \mid \cdot)$ is the same in both stages and $A^\star$ is applied* whenever *a proposal passes* (**??**)*, the two-stage kernel leaves $p_\alpha(\cdot)$ invariant; hence EE-PS is an exact sampler for p_α.*

3.2 Gating and calibration

To keep full corrections rare and reliable, we use:

- **Uncertainty gating:** escalate to the full model (even before stage-1) when mid-layer predictions are diffuse: $H(\tilde{p}) > \eta$ or margin $\tilde{z} * (1) - \tilde{z} * (2) < \delta$.
- **Periodic diagnostics:** monitor $\text{KL}(\tilde{p}, |, p)$ on a small subsample to adjust (A, b, τ) or L^* if drift is detected.
- **Choice of L^*:** a short sweep selects the shallowest L^* with small mid–final KL on validation (often $\sim!$50–60

3.3 Proposal distribution

Within each block, we sample a resampling index and redraw the suffix from the base model at temperature $1/\alpha$. This yields proposals closer to the power target, improving the stage-1 pass rate and the overall acceptance versus temperature-1 proposals.

Algorithm 1 EE-PS (block length B, DA-MH for p_α)

Initialize block prefix (e.g., base model, temp $1/\alpha$).
for $n = 1$ to N_{MCMC} **do**
  Sample resampling index $m \in 1, \ldots, B$ within the block.
  Propose x' by regenerating tokens m:end at temp $1/\alpha$.
  if $H(\tilde{p}) > \eta$ or margin $< \delta$ **then go to full-depth check**.
  Stage-1 (truncated): run to L^* for x, x', compute $\tilde{A}$ via (**??**); draw $U! \sim! U[0, 1]$.
  if $U > \tilde{A}$ **then**
    Reject x' and continue.
  else
    Stage-2 (full): complete to depth L, compute $A^\star$ via (4).
    Accept x' with probability $A^\star$ else keep x.
Output final block; proceed to next block.

3.4 Algorithm

4 Compute Analysis

Let C_{full} denote the FLOPs for a full-depth likelihood evaluation per proposal (including KV-cache updates), and $C_{\text{mid}} \approx (L^*/L), C_{\text{full}}$ for a truncated pass. Let ρ be the fraction of proposals that *reach* stage-2 (i.e., pass stage-1 or trigger gating). Expected cost per proposal:

$$C_{\text{EE-PS}}; \approx; (1 - \rho), C_{\text{mid}}; +; \rho, (C_{\text{mid}} + C_{\text{full-tail}}), \tag{5}$$

where $C_{\text{full-tail}} \leq C_{\text{full}} - C_{\text{mid}}$ accounts for reusing prefill/KV-cache and continuing from L^* to L. Add small overheads for (i) calibration (A, b, τ) and (ii) periodic diagnostic KL on a subsample. Using $L^*/L! \approx! 0.55$ and $\rho! \in! [0.1, 0.3]$, the *per-proposal* FLOPs can drop by $\sim$!35–50

5 Evaluation Plan

Benchmarks. GSM8K, MATH-500, GPQA-Diamond (reasoning), HumanEval/MBPP (code), plus a constrained generation task (e.g., equation completion). **Baselines.** Greedy/temperature/nucleus; standard power sampling (same α, blocks, steps); early-exit/self-speculative decoding baselines (e.g., LayerSkip, EESD); and DoLa (contrastive layers) as a mid-/late-layer decoding comparator. **Metrics.** Exact match / pass@k; MH acceptance; stage-1 pass rate; number of stage-2 calls per generated token; mid–final KL; FLOPs/token and wall-clock (including gating/diagnostics). **Ablations.** Exit depth L^*; calibration (none vs. temperature vs. affine probe); proposal temperature; block size/steps; gating thresholds.

6 Related Work

Power distribution sampling. The blockwise MH approach for sampling from p_α has shown strong gains on math/code tasks without training. EE-PS builds on this objective but introduces a mid-layer surrogate and a DA-MH correction to reduce full-depth evaluations while remaining exact.

Mid- vs. late-layer signals. Tuned/Logit Lens decode intermediate representations and show that mid-layer predictions align well with final outputs after lightweight calibration; DoLa exploits

contrasts between earlier and later layers to improve factuality during decoding. We use these insights to design an early-exit surrogate likelihood.

Early-exit and self-speculative decoding. Self-speculative and early-exit methods (LayerSkip, EESD) use a (partial) model run to draft tokens and the remaining layers to verify, aiming for speedups with quality preserved. In contrast, EE-PS is a Markov-chain sampler for a *different target* (p_α), using mid layers for *proposal screening* and DA for exactness.

Compute and KV-cache. Practical savings depend on cache reuse and scheduling. Systems such as vLLM (PagedAttention) highlight how memory management and chunked prefill affect latency and throughput for advanced decoding schemes; these considerations apply to EE-PS as well.

7 Limitations

If the surrogate mis-ranks proposals, stage-1 may pass too many candidates, increasing stage-2 calls and eroding savings. Some tasks may rely on late-layer refinements more heavily, reducing the advantage of early exit. Light per-model calibration of (A, b, τ) and the choice of L^* are required.

8 Expected Contributions

1. An exact DA-MH sampler for p_α that leverages mid-layer competence to avoid many full-depth evaluations.
2. A practical recipe (exit layer, calibration, gating) showing where early exit maintains fidelity for sampling-based reasoning.
3. Empirical quality–compute trade-offs competitive with full power sampling, without any additional training.

References

[1] A. Karan and Y. Du. Reasoning with Sampling: Your Base Model is Smarter Than You Think. *arXiv:2510.14901*, 2025. URL: https://arxiv.org/abs/2510.14901(https://arxiv.org/abs/2510.14901).

[2] J. A. Christen and C. Fox. Markov Chain Monte Carlo Using an Approximation. *Journal of Computational and Graphical Statistics*, 14(4):795–810, 2005. doi:10.1198/106186005X76983.

[3] M. Banterle, A. L. Gilks, S. Grazian, and C. P. Robert. Accelerating Metropolis–Hastings Algorithms by Delayed Acceptance. *arXiv:1503.00996*, 2015. URL: https://arxiv.org/abs/1503.00996(https://arxiv.org/abs/1503.00996).

[4] N. Belrose, Z. Furman, L. Smith, D. Halawi, I. Ostrovsky, L. McKinney, S. Biderman, and J. Steinhardt. Eliciting Latent Predictions from Transformers with the Tuned Lens. *arXiv:2303.08112*, 2023. URL: https://arxiv.org/abs/2303.08112(https://arxiv.org/abs/2303.08112).

[5] Y.-S. Chuang, Y. Xie, H. Luo, Y. Kim, J. Glass, and P. He. DoLa: Decoding by Contrasting Layers Improves Factuality in Large Language Models. *arXiv:2309.03883*, 2023. URL: https://arxiv.org/abs/2309.03883.

[6] M. Elhoushi, A. Shrivastava, D. Liskovich, B. Hosmer, B. Wasti, L. Lai, A. Mahmoud, B. Acun, S. Agarwal, A. Roman, A. Aly, B. Chen, and C.-J. Wu. LayerSkip: Enabling Early Exit Inference and Self-Speculative Decoding. In *ACL 2024*, 2024. Preprint *arXiv:2404.16710*. URL: https://arxiv.org/abs/2404.16710.

[7] J. Liu, Q. Wang, J. Wang, and X. Cai. Speculative Decoding via Early-exiting for Faster LLM Inference with Thompson Sampling Control Mechanism (EESD). *arXiv:2406.03853*, 2024. URL: https://arxiv.org/abs/2406.03853.

[8] R. Csord'as, C. D. Manning, and C. Potts. Do Language Models Use Their Depth Efficiently? *arXiv:2505.13898*, 2025. URL: https://arxiv.org/abs/2505.13898.

[9] W. Kwon, Z. Li, S. Zhuang, Y. Sheng, L. Zheng, C. H. Yu, J. E. Gonzalez, H. Zhang, and I. Stoica. Efficient Memory Management for Large Language Model Serving with PagedAttention. *arXiv:2309.06180*, 2023. URL: https://arxiv.org/abs/2309.06180.