# Beyond Static Cutoffs: One-Shot Dynamic Thresholding for Diffusion Language Models

**Jucheng Shen**
Rice University
js237@rice.edu

**Yeonju Ro**
The University of Texas at Austin
yro@cs.utexas.edu

## Abstract

Masked Diffusion Language Models (MDLM) are becoming competitive with their autoregressive counterparts but commonly decode with fixed steps and sequential unmasking. To accelerate decoding, recent works like Fast-dLLM enables parallel decoding via a static global confidence threshold, yet we observe strong block/step-wise confidence fluctuations and, within a dataset, near-identical confidence trajectories across inputs indicated by cosine similarity. Inspired by these two observations, we introduce **One-Shot Dynamic Thresholding (OSDT)**, which calibrates thresholds on a single sequence and applies them to subsequent inputs with negligible overhead. On GPQA, GSM8K, and HumanEval, OSDT attains superior accuracy–throughput trade-offs (**+24%** tokens/s on GSM8K at the **best** accuracy, **+45%** on GPQA with comparable accuracy, and **+50%** on HumanEval with a modest accuracy gap). Beyond these results, our findings suggest broader opportunities to leverage reusable task-level confidence signatures for more general-purpose algorithmic and systems innovations in diffusion decoding.

## 1 Introduction

Masked diffusion language models (MDLM) have recently advanced rapidly, with pre-trained models such as LLaDA-8B [1] and Dream-7B [2] scaling discrete diffusion and demonstrating strong performance across language, math, and code. Inference in these models is semi-autoregressive: the sequence is partitioned into contiguous blocks $B_1, \dots, B_T$ that are generated left-to-right (autoregressive across blocks), while within each block decoding proceeds by diffusion-style denoising steps $s = 1, \dots, S$ that can unmask any subset of still-masked positions in a non-left-to-right order. At each step, a mask predictor proposes token distributions and a fixed per-step quota of token positions is filled with topk by confidence or randomly, while the remaining positions stay masked for later steps.

To push throughput beyond fixed number of tokens per step, recent works like Fast-dLLM [3] enables parallel decoding by unmasking all tokens above a **static** global confidence threshold. However, our empirical observations show that confidence is highly non-uniform: it varies across blocks and denoising steps, yet confidence trajectories are remarkably stable across inputs within the same dataset (or, the same task). The first observation calls for adaptivity over the course of generation; the second indicates that a small amount of computation can capture a task-level "confidence signature." Together, they reveal substantial space for more efficient parallel decoding in MDLM than static schedules can offer.

We therefore study **dynamic thresholding for parallel decoding in masked diffusion models**. Formally, given a remasking decoder and per-step confidence scores, the goal is to design an inference-time policy $\pi$ that selects thresholds $\{\tau_b\}$ or $\{\tau_{b,s}\}$ to decide which masked tokens to unmask, maximizing throughput while maintaining task accuracy (or, equivalently, improving the

39th Conference on Neural Information Processing Systems (NeurIPS 2025) Workshop: Efficient Reasoning Workshop.

accuracy–throughput Pareto frontier). The policy must be **task-aware, training-free, and add negligible overhead**.

Our contributions can be summarized as follows:

- We empirically establish that within a dataset the block/step-wise confidence vectors are highly correlated across inputs (cosine similarity near 1), revealing a stable task-level signature.
- We introduce **One-Shot Dynamic Thresholding (OSDT)**, a two-phase decoder that calibrates thresholds on a single sequence and reuses them for subsequent inputs at either block or step-block granularity.
- We provide a comprehensive evaluation on GPQA, GSM8K, and HumanEval, showing that OSDT consistently improves tokens/s at comparable accuracy and often sets a better Pareto frontier than static thresholds as in Fast-dLLM.

## 2 Observation

Prior work [3] assumes confidence follows a generalizable pattern and applies a static threshold across tasks. In practice, as shown in Figure 1, benchmarks as different as GPQA (expert-level Q&A), GSM8K (grade-school math), and HumanEval (code generation) exhibit distinct signatures. This task-dependent variability undermines any "one-size-fits-all" thresholding strategy, highlighting the need for adaptation at inference time.

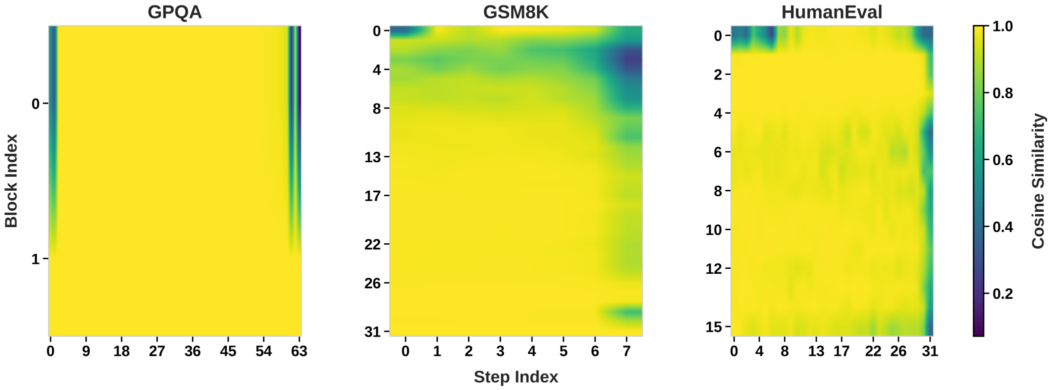

Figure 1: **Step-block mean token confidence.** Across GPQA, GSM8K, and HumanEval, confidence starts low, peaks mid-process, and drops near the final steps. These structured U-shaped dynamics highlight the limits of static thresholding.

A natural question is whether these dynamics remain stable within a dataset. If confidence patterns are consistent across inputs, the profile of one sequence could proxy for the rest. To test this, we measured cosine similarity between average step-blockwise confidence vectors for all input pairs.

Results in Figure 2 show striking consistency: cosine values are near 1.0 across GPQA, GSM8K, and HumanEval, producing nearly uniform bright heatmaps. This indicates that while confidence levels differ across tasks, the relative trajectories are highly stable within a dataset—a task-level rather than instance-level property.

This stability implies strong predictive power: the behavior on one sequence forecasts that of others in the same dataset. It provides the basis for adaptive thresholding methods that calibrate once and generalize broadly, enabling more efficient inference without sacrificing accuracy.

## 3 Dynamic Threshold

These findings expose the limits of static cutoffs: they fail to adapt to structured, task-dependent confidence dynamics, yet within-dataset patterns are highly consistent. This motivates a method that

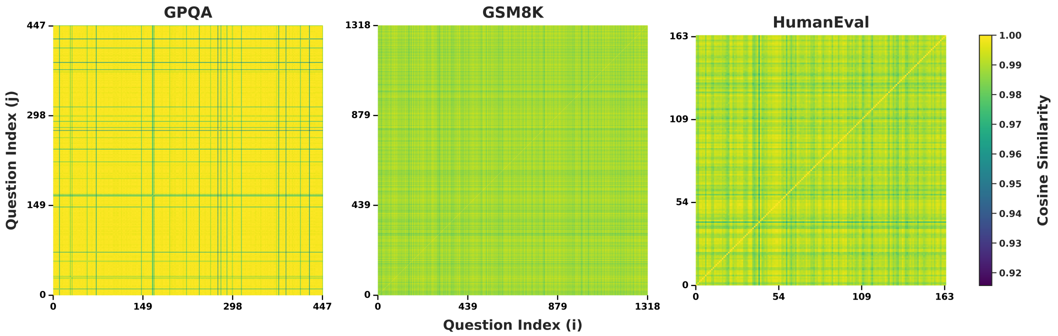

Figure 2: **Pairwise cosine similarity of step-block mean token confidence.** Confidence trajectories are nearly identical across inputs of the same dataset, suggesting a single calibration run can generalize to the entire benchmark.

is both dynamic and sample-efficient—able to adapt thresholds to a dataset's confidence profile with minimal overhead.

We propose **One-Shot Dynamic Thresholding (OSDT)**, a two-phase approach:

- **Phase 1 (Calibration):** The first sequence is decoded with standard static-thresholding as in Fast-dLLM, and block- or step-block-wise confidence vectors are collected.
- **Phase 2 (Dynamic Inference):** Subsequent sequences use thresholds derived from the calibration profile, with safeguards such as a cap $\kappa$ and slack ratio $\epsilon$ to balance efficiency and robustness.

This mechanism dynamically aligns decoding decisions to the task's confidence landscape. Unlike static thresholding, OSDT tailors thresholds to dataset-level dynamics while incurring negligible extra cost. Algorithm 1 (see Appendix) outlines the procedure. Its reliance on a single calibration run makes OSDT practical, efficient, and well-suited to real-world scenarios where both accuracy and latency matter.

# 4 Evaluation

We evaluate **One-Shot Dynamic Thresholding (OSDT)** on three benchmarks: GPQA (expert-level Q&A), GSM8K (grade-school math), and HumanEval (code generation). OSDT calibrates a confidence profile on the first sequence of a dataset and applies dynamically adjusted thresholds to subsequent inputs. We study both its hyperparameter behavior and its performance against Fast-dLLM [3]. All experiments use batch size 1 on a single NVIDIA H100 GPU.

## 4.1 Hyperparameters

OSDT exposes four hyperparameters that trade accuracy for throughput:

- **Dynamic Mode ($M$):** Threshold per block (*block*) or per denoising step within each specific block (*step-block*).
- **Threshold Metric ($\mu$):** Statistic over calibration confidences (mean, Q1, median, Q3, min-whisker).
- **Threshold Cap ($\kappa$):** Upper bound to avoid overly strict thresholds.
- **Slack Ratio ($\epsilon$):** Downscales thresholds to increase parallelism.

We perform a grid search over $\mu$, $\kappa \in \{0.75, 0.8, 0.85, 0.9, 0.95\}$, $\epsilon \in \{0.01, 0.05, 0.1, 0.15, 0.2\}$. Results (see Appendix for more details) show task-dependent optima: GPQA benefits from fine-grained *step-block* thresholds, whereas GSM8K and HumanEval prefer the simpler *block* mode. We use the following configurations in all following comparisons:

- **GPQA:** *step-block*, $q2$, $\kappa = 0.75$, $\epsilon = 0.20$.
- **GSM8K:** *block*, $q1$, $\kappa = 0.75$, $\epsilon = 0.20$.
- **HumanEval:** *block*, $q1$, $\kappa = 0.80$, $\epsilon = 0.10$.

### 4.2 Comparative Results

Table 1 compares OSDT to Fast-dLLM's fixed-threshold ($\tau = 0.9$) and best factor-based settings. OSDT consistently achieves a better accuracy–throughput trade-off across diverse reasoning and code generation benchmarks:

- **GSM8K:** OSDT achieves the highest accuracy (76.0%) while running 24% faster than the fixed-threshold baseline, indicating that dynamic thresholding effectively balances precision and decoding speed for step-by-step numerical reasoning.
- **GPQA:** OSDT maintains comparable accuracy (29.2% vs. 29.9%) with 45% higher throughput, suggesting that OSDT adapts well even in multi-hop question answering tasks with varying reasoning depth.
- **HumanEval:** OSDT delivers similar pass rate (40.9% vs. 43.3%) yet achieves 50% faster throughput, demonstrating that dynamic control does not compromise code-generation reliability.

Overall, OSDT achieves more favorable Pareto points, accelerating inference without significant quality degradation. The results highlight that adaptive thresholding improves computational efficiency by selectively reducing redundant forward passes—particularly when model confidence stabilizes—yielding consistent gains across both reasoning and code generation domains.

Table 1: **Comparative results.** Best values in **bold**. Throughput in tokens/s.

| **Benchmark** | **OSDT (Ours)** | | **Fast-dLLM (Fixed)** | | **Fast-dLLM (Factor)** | |
|---|---|---|---|---|---|---|
| | **Acc. (%)** | **Thru.** | Acc. (%) | Thru. | Acc. (%) | Thru. |
| GPQA | 29.24 | **63.27** | 28.12 | 42.69 | **29.91** | 43.58 |
| GSM8K | **76.00** | **230.75** | 74.75 | 172.74 | 75.00 | 186.63 |
| HumanEval | 40.85 | **172.25** | 39.63 | 152.51 | **43.29** | 114.71 |

## 5 Related Work

Recent advances in masked diffusion language models (DLMs) demonstrate their potential as an alternative to autoregressive generation. **LLaDA** [1] scales discrete diffusion to 8B parameters with block-wise decoding and low-confidence remasking, while **Dream** [2] improves training via AR initialization and context-adaptive token-level noise rescheduling. Both rely on fixed-step schedules that unmask tokens one by one, limiting efficiency.

**Fast-dLLM** [3] addresses this by formally proving when greedy parallel decoding with product-of-marginals is equivalent to sequential decoding in high-confidence regimes, enabling safe parallel generation. It then proposes a *confidence-aware parallel decoding* rule that unmasks all tokens above a global static cutoff, and introduces prefix and dual (prefix+suffix) KV-Cache designs to improve throughput. However, its thresholding remains task-agnostic and static.

In contrast, our work focuses on *dynamic, task-aware thresholding*. Instead of relying on fixed schedules or static cutoffs, we calibrate thresholds from a single input and adapt them to dataset-level confidence patterns, yielding more efficient and accurate decoding.

## 6 Conclusion

In this paper, we identify a simple but overlooked property of MDLMs: confidence evolves in structured ways over time yet is strikingly consistent across inputs within the same task. Building on this, we introduce **One-Shot Dynamic Thresholding (OSDT)**, a training-free, dataset-aware decoding scheme that calibrates thresholds on a single sequence and applies them with negligible overhead; On LLaDA-8B across GPQA, GSM8K, and HumanEval, OSDT improves the accuracy–throughput frontier, e.g., +24% tokens/s at the best accuracy on GSM8K, +45% throughput on GPQA at comparable accuracy, and +50% on HumanEval with a modest accuracy gap.

# References

[1] Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models, 2025.

[2] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025.

[3] Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding, 2025.

# A Technical Appendices and Supplementary Material

## A.1 OSDT Algorithm

**Algorithm 1** One-Shot Dynamic Thresholding (OSDT)

```
Input: Prompts $Q = \{q_1, \dots, q_n\}$; mode $M \in \{\texttt{block}, \texttt{step-block}\}$; metric $\mu$; cap $\kappa$; slack $\epsilon$
Output: Answers $A = \{a_1, \dots, a_n\}$
$A \leftarrow \varnothing$;   $\mathcal{T} \leftarrow \varnothing$
Notation: $\mathcal{B}(b)$ denotes the indices in block $b$.
for $i \leftarrow 1$ to $n$ do
    if $i = 1$ then                                          ▷ one-shot calibration
        $(a_1, conf) \leftarrow$ STANDARDGENERATE($q_1$)
        $\mathcal{T} \leftarrow$ CALIBRATE($conf, M, \mu$)
        APPEND($A, a_1$)
    else
        $x \leftarrow$ INITSEQUENCE($q_i$)
        for $b \leftarrow 1$ to num_blocks do
            step $\leftarrow 0$
            while MASKED($x, b$) do
                conf $\leftarrow$ CONFIDENCE(PREDICT($x$), $\mu$)
                if $M =$ step-block then
                    $\tau \leftarrow \mathcal{T}[b][step]$           ▷ step-block-level
                else
                    $\tau \leftarrow \mathcal{T}[b]$                 ▷ block-level
                $\tau \leftarrow \min(\tau, \kappa)$;   $\tau_{\text{eff}} \leftarrow \tau(1 - \epsilon)$
                $S \leftarrow \{\, j \in \mathcal{B}(b) \;:\; conf[j] > \tau_{\text{eff}} \,\}$      ▷ indices in block $b$ above threshold
                if ISEMPTY($S$) then
                    idx $\leftarrow$ INDEXOFMAX(conf, block $b$)
                    $S \leftarrow \{idx\}$           ▷ fallback: unmask most confident index in block $b$
                UNMASKANDUPDATE($x, S$);   step $\leftarrow$ step $+ 1$
        $a_i \leftarrow x$;   APPEND($A, a_i$)
return $A$
```

## A.2 Full Hyperparameter Sweep Results

We provide detailed accuracy–throughput trade-offs for OSDT across GPQA, GSM8K, and HumanEval. Each plot visualizes all combinations of dynamic mode ($M$), threshold metric ($\mu$), threshold cap ($\kappa$), and slack ratio ($\epsilon$). Marker shape denotes $\mu$, marker size denotes $\kappa$, color indicates $\epsilon$, and line style distinguishes *step-block* (solid) from *block* (dashed).

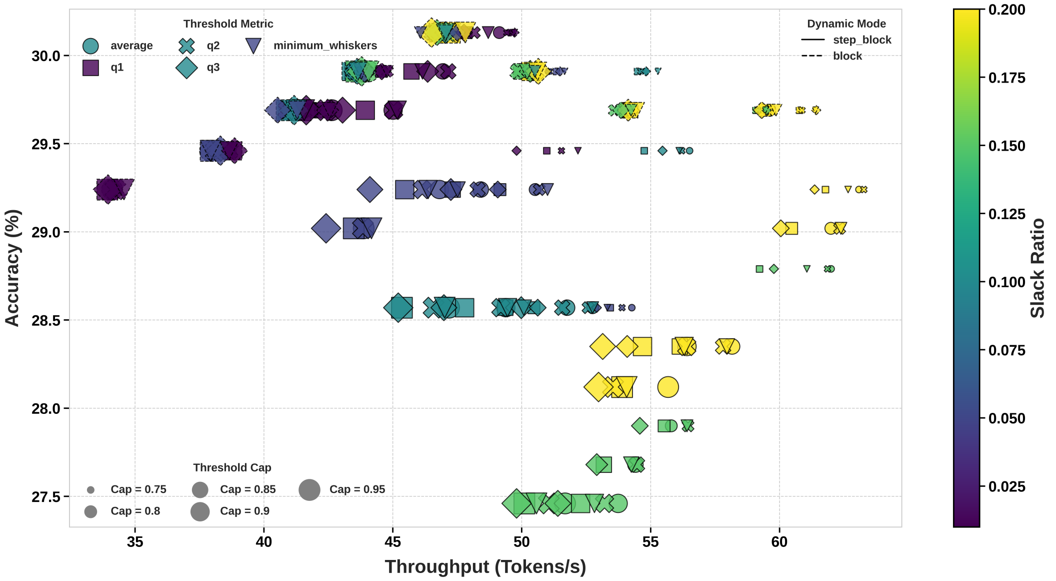

Figure 3: **GPQA hyperparameter sweep.** Accuracy peaks near 30% but varies only slightly across settings, while throughput is strongly influenced by $\epsilon$ and $\kappa$. Step-block mode provides finer adaptation and better trade-offs in high-accuracy regions.

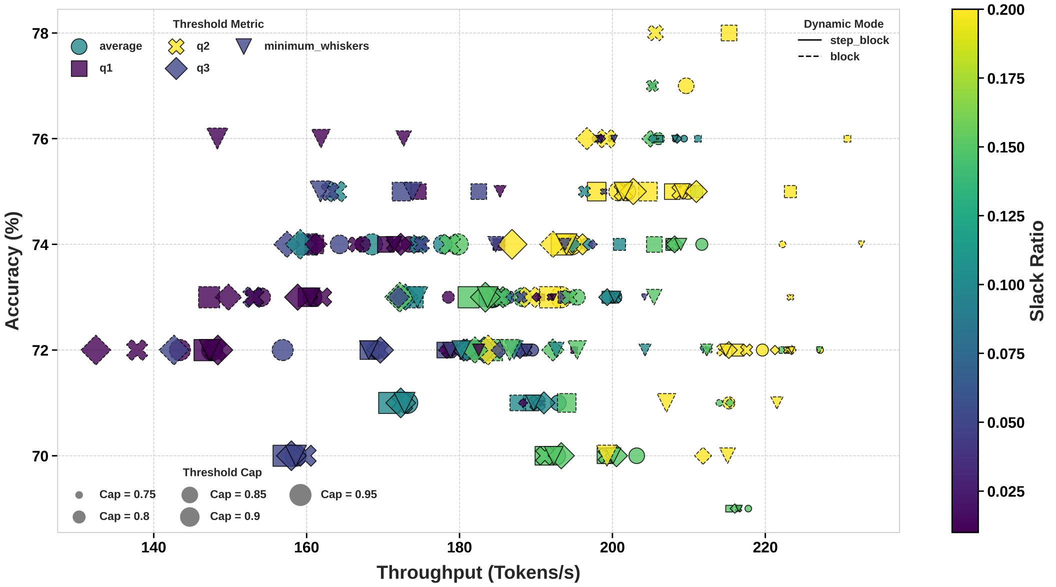

Figure 4: **GSM8K hyperparameter sweep.** Structured reasoning tasks benefit most from block-level thresholds, which achieve higher accuracy (up to 76%) while maintaining strong throughput. Step-block offers little advantage here, confirming block mode suffices for GSM8K.

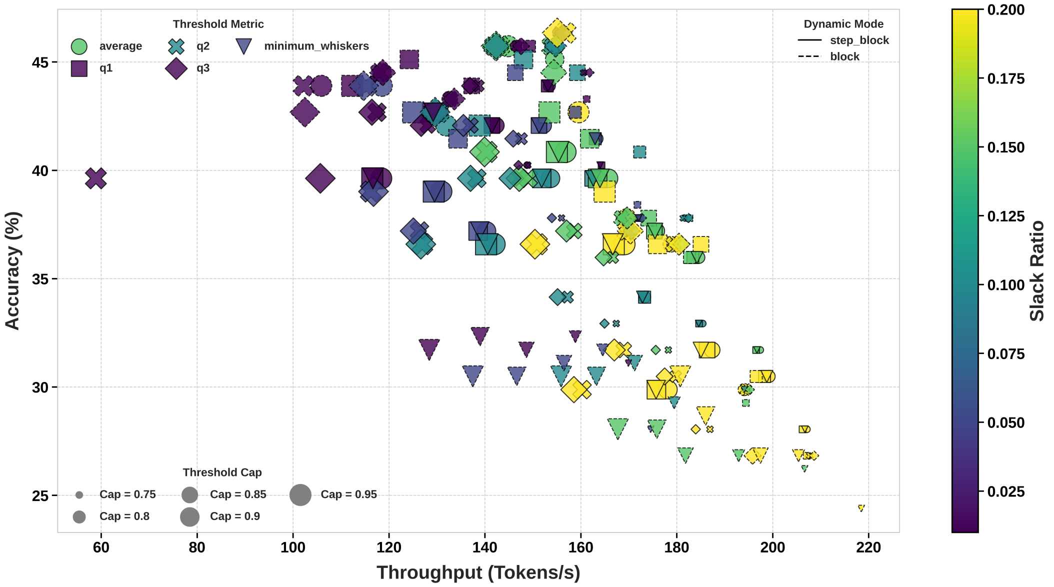

Figure 5: **HumanEval hyperparameter sweep.** Code generation shows a sharper accuracy–throughput trade-off: aggressive settings yield large speedups but accuracy drops quickly. Block-level thresholds dominate the Pareto frontier, offering simpler yet more efficient schedules.