# Stopping Computation for Converged Tokens in Masked Diffusion-LM Decoding

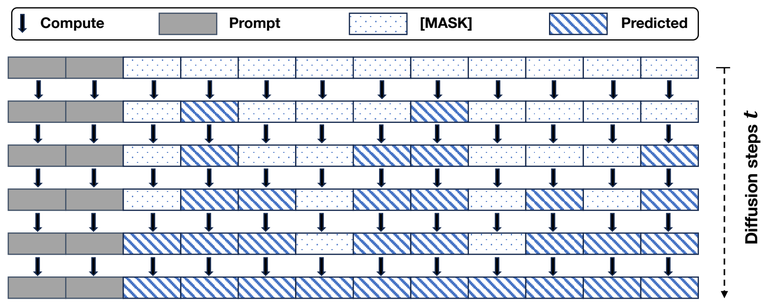
Daisuke Oba[1] Danushka Bollegala[2,3,†] Masahiro Kaneko[4] Naoaki Okazaki[1]
[1]Institute of Science Tokyo [2]University of Liverpool [3]Amazon [4]MBZUAI

***Stable positions do not need later diffusion updates! Lock them and move on — SureLock***

## Motivation

**Every diffusion step recomputes the whole sequence**

Standard masked diffusion decoding still recomputes attention and FFN for the whole sequence at every step, even after many positions have been unmasked/stabilized




The dominant per-step compute remains $O(N^2 d)$, where $N$ is the sequence length, $d$ is the model dimension

## Key question

**Can later diffusion steps avoid recomputing stable positions without losing context?**

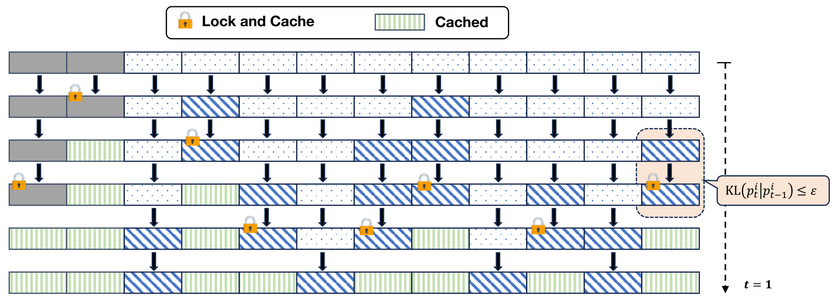
## Gap

**Existing acceleration asks how to decode faster, not what later steps can stop computing**

Prior work i) reduces the number of diffusion steps, or ii) reuses computation across steps. But it does not directly ask "*what can be removed from later computation*"

## SureLock

**Lock stable positions, cache their K/V, and update only the rest**




**Step 1.** **Examine posterior stability across consecutive steps**

$D_t^i = \mathrm{KL}(p_t^i | p_{t-1}^i)$ where $p_t^i$ is the posterior at position $i$ at step $t$

**Step 2.** **Lock stable positions and cache their K/V**

$D_t^i \le \varepsilon \implies$ lock the position $i$ and cache its $K/V$

**Step 3.** **Update only the remaining positions in later steps**

$\forall i \in A_t$: update $i$, $\forall i \in L_t$: reuse $K_i/V_i$, skip $Q_i$ , $FFN_i$

**Compute after locking**

Attention: $O(N^2 d) \longrightarrow O(|A_t| N d)$
FFN: $O(N d^2) \longrightarrow O(|A_t| d^2)$

## Why use step-wise KL for locking rule?

**Lock-time KL bounds the final deviation after locking**

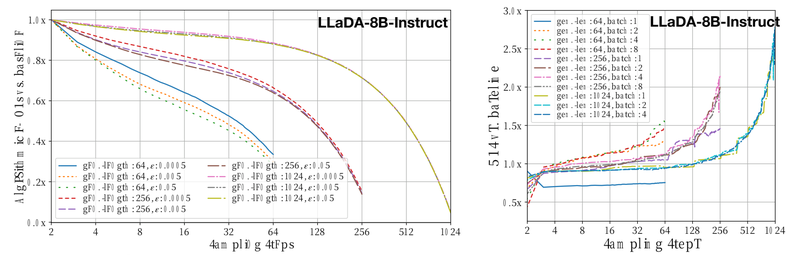
$$D_{t^{\backslash *}}^i \le \varepsilon \implies \left\| \log p_{base,final}^i - \log p_{lock,final}^i \right\|_\infty \le C_{tail} \sqrt{\varepsilon}$$

$t^{\backslash *}$ : the locking step index
$p_{base,final}^i$ / $p_{lock,final}^i$: the final posteriors without / with locking

This gives a theoretical rationale for locking by step-wise KL: a smaller $\varepsilon$ yields a tighter worst-case bound on the final deviation

## Efficiency and Generation Quality

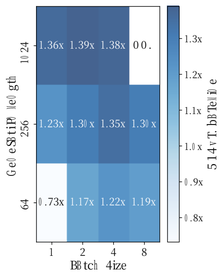
**Later diffusion steps become both lighter and faster**




**Generation quality is largely preserved**

| Model | Benchmark | Quality vs. baseline (avg.) | FLOPs vs. baseline (avg.) |
|---|---|---|---|
| LLaDA-8B | WikiText-103 | **1.13x** (Gen.-PPL) | **0.548x** |
| LLaDA-8B-Instruct | MT-Bench | **1.02x** (llm-as-a-judge) | **0.576x** |
| LLaDA-8B-Instruct | HumanEval | **1.05x** (pass@1) | **0.464x** |

## Practical note




**Observed speedups depend on settings**

Compute-bound settings benefit most; Lighter settings are limited by overheads (e.g., non-contiguous memory access)

## Illustrative example

**Question:** Benjamin went to a bookstore and purchased a variety of books. He bought 5 copies of a sci-fi novel, each priced at $20, 3 copies of a history book priced at $30 each, and 2 copies of a philosophy book for $45 each. What was the total cost of his purchases?.

**Baseline** (score: 10)
Benjamin bought 5 copies of a sci-fi novel, each priced at $20, so he spent 5 * $20 = $100. He bought 3 copies of a history book, each priced at $30, so he spent 3 * $30 = $90. He bought 2 copies of a philosophy book, each priced at $45, so he spent 2 * $45 = $90. The total cost of his purchases is $100 + $90 + $90 = $280. Summing up,

**SureLock** (score: 10)
Benjamin bought 5 copies of a sci-fi novel for $20 each, so the total cost is 5 * $20 = $100. He also bought 3 copies of a history book for $30 each, so the total cost is 3 * $30 = $90. Lastly, he bought 2 copies of a philosophy book for $45 each, so the total cost is 2 * $45 = $90. The total cost of his purchases is $100 + $90 + $90 = $280. Con

† He holds a concurrent appointment as an Amazon Scholar. But this research is not associated with Amazon