

When to Change the Canvas? Safe Adaptive Variable-Length Decoding for Diffusion Language Models

Junzhuo Li^{†‡}, Tianhao Li[¶], Hao-Ran Wei[¶], Xiang Li^{†‡},
Baosong Yang[¶], and Xuming Hu^{†‡*}

[†]The Hong Kong University of Science and Technology (Guangzhou)

[‡]The Hong Kong University of Science and Technology

[¶]Qwen Team, Alibaba Group

jz.li@connect.hkust-gz.edu.cn

xuminghu@hkust-gz.edu.cn

Abstract

Diffusion language models generate text by denoising a masked canvas, but usually require the canvas length to be fixed before decoding. This leads to an allocation dilemma: short canvases may truncate outputs, while overlong canvases waste computation and introduce noisy continuations. Existing variable-length methods adapt the canvas through expansion or length search, but largely ignore whether such intervention is safe. We argue that canvas editing can perturb partially stabilized denoising trajectories, and propose Safe Adaptive Canvas Decoding (SACD), a training-free framework for conservative canvas control. SACD diagnoses allocation, estimates intervention risk, and chooses among EXPAND, SHRINK, and KEEP. Experiments across reasoning, code generation, and infilling tasks show that SACD improves accuracy, latency, and token utilization across multiple DLM backbones, highlighting safe sparse intervention as a key principle for variable-length diffusion decoding.

1 Introduction

Diffusion language models (DLMs) (Austin et al., 2021a; Shi et al., 2024; Sahoo et al., 2024; Nie et al., 2026; Ye et al., 2025; Zhu et al., 2025a,b) generate text through iterative denoising over a masked response canvas, enabling bidirectional context modeling and parallel token refinement. Despite their strong generation ability, most existing DLMs still rely on a fixed response canvas specified before decoding begins. This creates a fundamental allocation dilemma: insufficient canvases may truncate reasoning, while over-allocated canvases waste computation and introduce redundant continuation regions. The problem is especially severe for reasoning and code generation tasks, where output lengths vary substantially across examples (Wei et al., 2022; Chen et al., 2021).

Recent work attempts to address this limitation through variable-length diffusion decoding, including dynamic expansion (Wu et al., 2026; Yang et al., 2025; Kim et al., 2026) and length-search strategies (Li et al., 2026; Cheng et al., 2026). However, these methods largely assume that more adaptive editing leads to better generation (Song et al., 2021). We argue that this assumption overlooks a key property of diffusion decoding: canvas editing itself perturbs the denoising trajectory.

Diffusion generation is a progressive refinement process where token confidence and semantic structure gradually stabilize across denoising steps (Wu et al., 2025; Wei et al., 2026). Modifying the canvas changes these dynamics directly. For example, a partially denoised sequence may already form a stable local structure. Inserting new mask positions between existing tokens forces the model to reorganize partially converged token dependencies and re-distribute uncertainty over neighboring regions. Similarly, aggressive shrinking may remove semantic regions whose boundaries have not yet stabilized (Ren et al., 2020). Frequent expansion, truncation, or length search can therefore disrupt denoising continuity, destabilize termination boundaries, and introduce unnecessary inference overhead.

These observations suggest that the central challenge of variable-length diffusion decoding is not simply how to adapt the canvas, but when intervention is truly necessary and safe. Effective variable-length decoding should minimize unnecessary trajectory perturbation rather than maximize adaptation frequency.

To address this problem, we propose Safe Adaptive Canvas Decoding (SACD), a training-free framework that performs sparse and conservative canvas intervention through EXPAND, SHRINK, and KEEP actions. SACD first diagnoses whether the current canvas is under-allocated, over-allocated, or already self-regulated, and then estimates whether intervention is sufficiently safe before modifying

*Corresponding author

Method	Train-free	Model-agn.	Task-agn.	Parallel	Shrink	Risk-aware
DAEDAL	✓	✗	✗	✓	✗	✗
DreamOn	✗	✗	✓	✓	✓	✗
FlexMDM	✗	✗	✓	✓	—	✗
dLLM-Var	✗	✗	✗	✓	—	✗
LR-DLLM	✓	✓	✓	✗	✓*	✗
SACD	✓	✓	✓	✓	✓	✓

Table 1: Comparison of variable-length decoding methods for diffusion language models. *LR-DLLM can adjust remaining length through local search, but commits tokens serially rather than preserving parallel denoising.

the canvas. Importantly, KEEP is treated as the default action under uncertainty, allowing SACD to preserve stable denoising trajectories whenever intervention is unnecessary. Experiments across reasoning, code generation, and infilling benchmarks show that SACD consistently improves accuracy, token utilization, and latency across multiple DLM families, while remaining robust under both short- and long-canvas initialization.

Our contributions are summarized as follows:

- We identify trajectory perturbation as an overlooked challenge in variable-length diffusion decoding, showing that frequent canvas editing can disrupt denoising continuity and destabilize generation.
- We propose SACD, a training-free framework that performs sparse and conservative canvas control through risk-aware EXPAND, SHRINK, and KEEP decisions.
- We demonstrate that minimizing unnecessary intervention improves generation stability, efficiency, and output quality across reasoning, code generation, and infilling benchmarks.

2 Background and Related Work

2.1 Fixed-Canvas Decoding in DLMs

Diffusion language models (DLMs) generate text by iteratively denoising a masked response canvas. Given an input context $\mathbf{c}$, a standard DLM initializes a canvas of predefined length L as $\mathbf{y}_T = [\mathbf{M}]^L$, and progressively commits tokens according to a confidence- or entropy-based scheduler. This decoding paradigm enables bidirectional context modeling and parallel token refinement (Sahoo et al., 2024; Shi et al., 2024; Ye et al., 2025; Nie et al., 2026), but it also requires the response length to be allocated before decoding begins.

This fixed-canvas assumption creates a length-allocation dilemma (Gong et al., 2023). A short

canvas may truncate reasoning or force overly compressed outputs, while a long canvas increases bidirectional attention cost and may introduce redundant or noisy continuation regions. The problem is especially pronounced in reasoning, code generation, and infilling tasks, where the required output length varies substantially across examples.

2.2 Variable-Length Decoding

Recent work relaxes the fixed-canvas assumption through either training-based or training-free variable-length generation. Training-based methods introduce explicit length modeling, termination mechanisms, or task-specific objectives, as in DreamOn, FlexMDM, and dLLM-Var (Wu et al., 2026; Kim et al., 2026; Yang et al., 2025). While effective, these methods usually require additional training or model-specific adaptation.

Training-free methods instead adjust the canvas during inference. DAEDAL dynamically expands the canvas from a short initialization using completion signals (Li et al., 2026), while LR-DLLM treats length as an inference-time variable and performs length-regularized local search (Cheng et al., 2026). These methods show that DLMs contain useful internal signals for length adaptation. However, they mainly focus on how to change the canvas length: DAEDAL is primarily expansion-oriented, and LR-DLLM relies on repeated length search that may weaken the parallel denoising advantage.

2.3 Toward Safe Canvas Intervention

Existing variable-length methods leave underexplored a complementary question: *when is it safe to intervene in the canvas?* Canvas editing is not a neutral operation. Expanding, shrinking, or repeatedly searching over lengths changes the denoising state on which subsequent predictions depend. Therefore, variable-length decoding should not only detect whether the current canvas is insufficient or redundant, but also assess whether modifying it may disrupt the ongoing denoising trajectory.

SACD follows this perspective by formulating variable-length decoding as conservative canvas control. Instead of continuously optimizing length, it intervenes only when the canvas is clearly misallocated and the corresponding edit is estimated to be safe. This leads to a risk-aware policy over EXPAND, SHRINK, and KEEP, where KEEP serves as the default action under uncertainty.

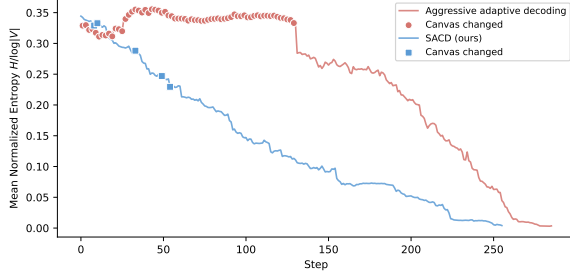


Figure 1: Motivation for conservative canvas control. Diamonds mark canvas changes made by an aggressive adaptive decoding variant. Frequent intervention is associated with persistently high entropy, while SACD yields a smoother entropy decay, suggesting that unnecessary canvas editing may delay denoising stabilization.

3 Methodology

3.1 Motivation and Overview

Given a prompt or context c , a standard diffusion language model initializes a fixed-length response canvas

$$\mathbf{y}_T = [\mathbf{M}]^L, \quad (1)$$

and iteratively denoises it by predicting token distributions and committing a subset of positions according to a base scheduler. While this enables parallel refinement, the predefined canvas may be either insufficient or redundant for a given instance.

A natural remedy is to modify the canvas during decoding, but such editing is not a neutral length operation. If a partially denoised sequence has already formed a local structure, inserting new mask positions between existing tokens changes token alignment and redistributes uncertainty. Similarly, shrinking before the termination boundary stabilizes may remove useful content. Thus, variable-length decoding should decide not only whether the canvas is misallocated, but also whether intervention is safe.

Figure 1 provides empirical motivation for this view. Compared with SACD, an aggressive adaptive variant that modifies the canvas more frequently keeps the mean normalized entropy high for a longer portion of the denoising trajectory, especially around repeated canvas changes. This suggests that frequent intervention may delay uncertainty reduction, motivating a sparse policy that preserves the current trajectory under uncertainty.

SACD generalizes fixed-canvas denoising by allowing the canvas length to vary over time:

$$\mathbf{y}_t \in (\mathcal{V} \cup \{\mathbf{M}\})^{L_t}, \quad (2)$$

where L_t is the canvas length at step t . At sparse detection steps, SACD selects

$$a_t \in \{\text{EXPAND}, \text{SHRINK}, \text{KEEP}\}, \quad (3)$$

where EXPAND appends mask tokens, SHRINK removes redundant suffix positions, and KEEP leaves the canvas unchanged. Importantly, KEEP is the default action when intervention evidence is uncertain.

As illustrated in Figure 2, SACD decomposes canvas control into necessity diagnosis and safety estimation. It first assigns the current state as

$$d_t \in \{\text{UNDER}, \text{OVER}, \text{SELF}\}, \quad (4)$$

indicating whether the canvas is insufficient, redundant, or already self-regulated. It then estimates the risk of the corresponding edit. SACD modifies the canvas only when the diagnosed state indicates a clear allocation problem and the intervention risk is low; otherwise, it selects KEEP. This conservative pipeline adapts the canvas while minimizing unnecessary perturbation to the denoising trajectory.

3.2 Canvas State Diagnosis

At each detection step, SACD first determines whether the current canvas is likely to be under-allocated, over-allocated, or already self-regulated. This stage estimates the *necessity* of intervention, rather than directly modifying the canvas.

Let $\mathbf{y}_t \in (\mathcal{V} \cup \{\mathbf{M}\})^{L_t}$ denote the current canvas. For each position i , the DLM produces a token distribution

$$p_t(v | i) = p_\theta(y_i = v | c, \mathbf{y}_t), \quad v \in \mathcal{V}. \quad (5)$$

We use two lightweight signals to summarize the model’s belief about canvas sufficiency: the end-of-text probability $E_t(i) = p_t(\text{eot} | i)$ and the normalized entropy $H_t(i)$. Intuitively, high EOT probability suggests that the model has located a possible termination region, while high entropy indicates that the content at a position remains uncertain.

Since position-wise predictions can be noisy, SACD computes locally smoothed statistics and estimates a tentative termination boundary:

$$\hat{b}_t = \arg \max_i [z(\log(\bar{E}_t(i) + \epsilon)) - \lambda z(\bar{H}_t(i))], \quad (6)$$

where $\bar{E}_t(i)$ and $\bar{H}_t(i)$ are window-smoothed eot probability and entropy, and $z(\cdot)$ denotes per-sample standardization across positions. A reliable

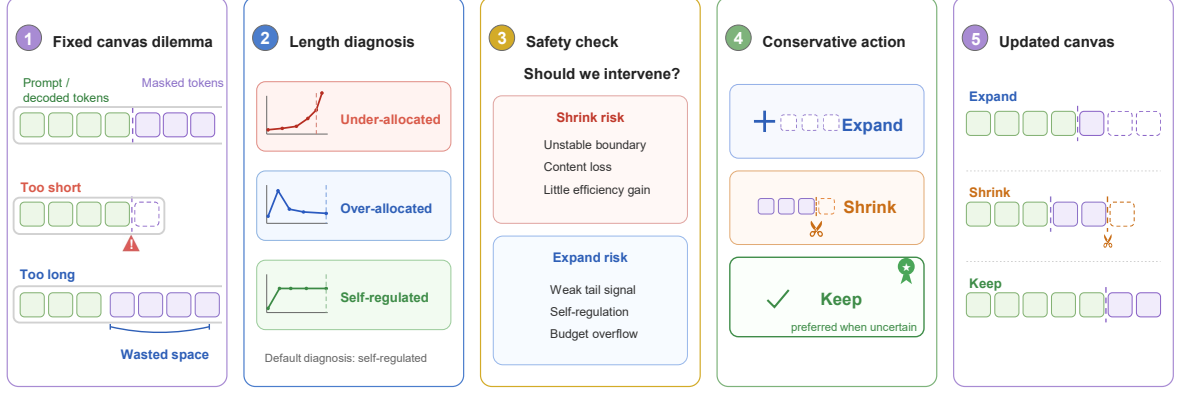


Figure 2: Overview of SACD. At sparse detection steps, SACD first diagnoses the current canvas state as under-allocated, over-allocated, or self-regulated. It then estimates whether a possible edit is safe and conservatively chooses among EXPAND, SHRINK, and KEEP. By treating KEEP as the default action under uncertainty, SACD adapts the canvas only when necessary while preserving the denoising trajectory.

boundary should have high eot confidence and low uncertainty.

However, a single boundary estimate is insufficient for safe intervention. SACD therefore maintains a short history of recent boundary estimates and measures boundary stability. A stable early boundary provides evidence that the suffix may be redundant, while an unstable boundary indicates that the denoising trajectory has not yet settled and should not be interrupted.

Based on these signals, SACD assigns the canvas state as

$$d_t \in \{\text{UNDER}, \text{OVER}, \text{SELF}\}. \quad (7)$$

The rule is intentionally conservative. UNDER is triggered only when the estimated boundary remains near the right edge and the tail region is still uncertain. OVER is triggered only when a stable boundary appears before the right edge and the post-boundary suffix has high EOT confidence and low entropy. All ambiguous cases are assigned to SELF, indicating that the current canvas should be preserved.

3.3 Risk-Aware Intervention Policy

After diagnosing the canvas state, SACD further estimates whether the corresponding edit is safe. This step reflects the central premise of SACD: even when the canvas appears misallocated, editing it may perturb the ongoing denoising trajectory. Therefore, an intervention is allowed only when both necessity and safety conditions are satisfied.

In implementation, SACD evaluates intervention safety using Boolean unsafe predicates rather

than a single scalar risk threshold. Let $\mathcal{U}_t^{\text{SHRINK}}$ and $\mathcal{U}_t^{\text{EXPAND}}$ denote whether shrinking or expanding is considered unsafe at step t . Each predicate is triggered when any corresponding risk condition is satisfied, such as unstable boundaries, possible content removal, insufficient efficiency gain, budget violation, or evidence that the model is already self-regulating.

The final action is selected by a conservative policy:

$$a_t = \begin{cases} \text{SHRINK}, & d_t = \text{OVER} \text{ and } \neg \mathcal{U}_t^{\text{SHRINK}}, \\ \text{EXPAND}, & d_t = \text{UNDER} \text{ and } \neg \mathcal{U}_t^{\text{EXPAND}}, \\ \text{KEEP}, & \text{otherwise.} \end{cases} \quad (8)$$

Thus, KEEP is not a passive fallback but an explicit safety action. When the allocation evidence is ambiguous or any unsafe condition is triggered, SACD preserves the current canvas to avoid unnecessary trajectory perturbation. Detailed definitions of the unsafe predicates and their thresholds are provided in Appendix A.

3.4 Conservative Canvas Editing

Given the selected action a_t , SACD edits the canvas using conservative operations designed to minimize disruption to the denoising trajectory.

Shrink. When $a_t = \text{SHRINK}$, SACD removes the redundant suffix after the estimated termination boundary. To avoid cutting off unresolved content, the new length is not set exactly to $\hat{b}_t$, but preserves an adaptive safety margin:

$$L'_t = \hat{b}_t + m_t, \quad (9)$$

where m_t increases when the boundary is unstable or weak. In practice, we instantiate

$$m_t = m_0 + \alpha \sqrt{\text{Var}(\mathcal{B}_t)} + \beta(1 - s_t), \quad (10)$$

where $\mathcal{B}_t$ is the recent boundary history and s_t measures boundary strength. The canvas is then updated as

$$\mathbf{y}_t \leftarrow \mathbf{y}_t[1 : L'_t]. \quad (11)$$

After shrinking, SACD disables subsequent expansion for the remainder of the current generation. This one-way lock prevents shrink-expand oscillation and preserves the shortened trajectory once the redundant suffix has been removed.

Expand. When $a_t = \text{EXPAND}$, SACD appends a small number of mask positions:

$$\mathbf{y}_t \leftarrow \mathbf{y}_t \parallel [\mathbf{M}]^{\Delta L_t}. \quad (12)$$

The expansion size ΔL_t is bounded by a maximum step size and the global length budget. SACD uses gradual expansion rather than jumping to a large canvas, allowing the model to reassess the enlarged canvas in later denoising steps.

Keep. When $a_t = \text{KEEP}$, the canvas remains unchanged:

$$\mathbf{y}_t \leftarrow \mathbf{y}_t. \quad (13)$$

This action is deliberately treated as part of the control policy. Since many denoising states are ambiguous, keeping the canvas unchanged often provides the safest way to preserve partially stabilized token dependencies.

Overall, SACD edits the canvas asymmetrically: shrinking is margin-preserving and followed by a lock, while expansion is gradual and budget-constrained. These design choices make canvas adaptation sparse, stable, and conservative. We provide the full implementation details in Appendix A and the complexity analysis in Appendix C.

4 Experiments

We evaluate whether SACD can adapt canvas length while preserving the denoising trajectory. Our experiments are designed to answer three questions: (1) Can SACD safely remove redundant canvas regions without harming generation quality? (2) Does conservative intervention improve the accuracy-efficiency trade-off compared with training-free variable-length baselines? (3) Is SACD robust when the initial canvas is either severely under-allocated or heavily over-allocated?

4.1 Experimental Setup

Models and tasks. We evaluate SACD on LLaDA-8B-Instruct and Dream-7B-Instruct, two representative diffusion language models with different decoding behaviors. For standard generation, we use GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021) for mathematical reasoning, and HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021b) for code generation. For code infilling, we evaluate on HumanEval-Infilling (Bavarian et al., 2022) with both single-line and multi-line settings. We report accuracy for mathematical reasoning and execution-based Pass@1 for code generation and infilling.

Evaluation settings. We consider three complementary settings. First, to test safe shrinking under over-allocation, we compare SACD with fixed-length denoising using a reference canvas of $L = 1024$. SACD starts from the same canvas and is allowed to shrink only when the suffix is diagnosed as redundant and safe to remove. This setting directly evaluates whether SACD can reduce wasted canvas regions while preserving output quality.

Second, to compare with training-free variable-length baselines, we evaluate DAEDAL (Li et al., 2026), LR-DLLM (Cheng et al., 2026), and SACD on LLaDA-8B-Instruct under an extremely short initialization, $L_{\text{init}} = 1$. This setting stresses short-to-long adaptation and highlights the cost of different intervention strategies: DAEDAL expands based on completion signals, LR-DLLM performs length-regularized search, and SACD uses sparse risk-aware intervention.

Third, to study robustness to initialization, we vary the initial canvas length over $L_{\text{init}} \in \{64, 128, 256, 512, 1024\}$, with $L_{\text{init}} = 1$ used as a stress test. This setting examines whether a method remains stable when the initial canvas is either insufficient or redundant.

Metrics. In addition to task performance and latency, we report final canvas length and token utilization, defined as the ratio between the effective length before the first EOT token and the final canvas length. To measure intervention safety, we report Intervention Harm Rate (IHR), Rescue Rate (RR), and No-Op Rate (NOR). IHR measures how often adaptive decoding harms examples that fixed-length decoding solves, RR measures how often it rescues examples fixed-length decoding fails, and

Benchmark	Fixed-Length Denoising			SACD				
	Ref. L	Acc.	Util.	Canvas	Acc.	Lat. (Ref.→SACD)	Util.	IHR/RR/NOR
<i>Part I: LLaDA-8B-Instruct</i>								
GSM8K	1024	83.8	34.2%	1024→387	85.6	106.2→30.9	88.0%	0.0 / 1.8 / 95.5
MATH-500	1024	40.4	55.7%	1024→466	44.2	109.7→32.3	95.3%	0.0 / 3.8 / 92.8
HumanEval [†]	1024	46.3	64.1%	1024→434	48.2	109.6→31.8	86.4%	1.2 / 3.0 / 96.1
MBPP [†]	1024	37.2	33.6%	1024→412	38.6	108.5→32.5	89.8%	0.0 / 1.4 / 94.2
<i>Part II: Dream-7B-Instruct</i>								
GSM8K	1024	81.7	27.7%	1024→335	84.4	94.2→26.9	79.2%	0.0 / 2.7 / 94.7
MATH-500	1024	40.8	32.9%	1024→461	42.4	97.4→30.3	81.0%	0.0 / 1.6 / 97.1
HumanEval [†]	1024	42.1	53.9%	1024→304	44.5	96.7→27.6	83.6%	0.6 / 3.0 / 95.5
MBPP [†]	1024	34.4	31.8%	1024→453	38.6	97.3→27.2	74.6%	0.0 / 4.2 / 92.8

Table 2: Main generation results under over-allocated canvas initialization. All methods start from the same reference length $L = 1024$. SACD safely shrinks the canvas to a more compact natural length while improving utilization, latency, and accuracy. Util. denotes token utilization. IHR/RR/NOR denote Intervention Harm Rate, Rescue Rate, and No-Op Rate. [†]HumanEval and MBPP accuracy uses execution-based Pass@1; IHR/RR/NOR use structural correctness checks.

NOR measures how often the controller chooses KEEP. A high NOR indicates that the method intervenes sparsely rather than frequently perturbing the denoising trajectory. Formal metric definitions are provided in Appendix B.

4.2 Main Results on Standard Generation

Table 2 evaluates SACD in the over-allocated setting, where fixed-length denoising uses a reference canvas of $L = 1024$ and SACD starts from the same canvas. This setting tests whether SACD can remove redundant canvas regions without perturbing otherwise successful denoising trajectories.

Across all benchmarks and both DLM backbones, SACD shrinks the canvas to substantially shorter natural lengths while preserving or improving accuracy. On LLaDA-8B-Instruct, the final canvas is reduced from 1024 tokens to 387–466 tokens, with accuracy gains on GSM8K, MATH-500, HumanEval, and MBPP. On Dream-7B-Instruct, SACD similarly reduces the canvas to 304–461 tokens and improves accuracy across all four tasks. These results suggest that over-allocated canvases are not merely inefficient: redundant continuation regions may also interfere with the denoising process, and removing them safely can improve generation quality.

SACD also yields large efficiency gains. On LLaDA, latency decreases from roughly 106–110 seconds to 31–33 seconds; on Dream, it decreases from 94–97 seconds to 27–30 seconds. Token utilization increases consistently, e.g., from 34.2% to 88.0% on LLaDA GSM8K and from 27.7% to

79.2% on Dream GSM8K. This shows that SACD does not simply shorten outputs, but converts over-allocated canvases into compact generation regions with substantially less wasted space.

The safety metrics further support the conservative-intervention principle. IHR remains near zero across almost all settings, indicating that SACD rarely harms examples solved by fixed-length decoding. Meanwhile, RR is consistently positive, showing that safe canvas adjustment can rescue some failures caused by noisy or redundant canvas regions. NOR remains above 92% in most settings, demonstrating that SACD obtains its gains through sparse high-confidence interventions rather than frequent trajectory perturbation. Together, these results support our central claim: effective variable-length diffusion decoding should intervene only when the canvas is clearly misallocated and the intervention is safe.

4.3 Comparison under Severe Short-Canvas Initialization

We next compare SACD with two training-free variable-length decoding baselines, DAEDAL and LR-DLLM, on LLaDA-8B-Instruct. All methods are initialized with an extremely short canvas, $L_{\text{init}} = 1$. This setting is not intended to reproduce the default operating point of every baseline; rather, it serves as a severe under-allocation stress test that evaluates whether a decoder can recover from minimal initial generation space while maintaining low inference overhead. We further examine more moderate initial lengths, including $L_{\text{init}} = 64$, in the

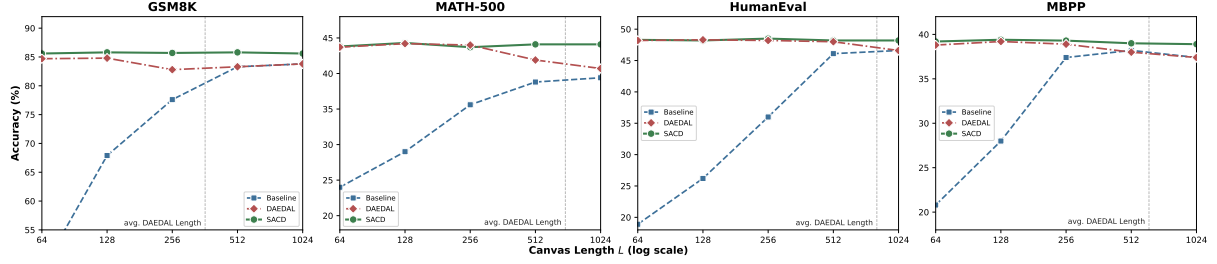


Figure 3: Length sensitivity on LLaDA-8B-Instruct across four benchmarks. Fixed-length denoising is highly sensitive to the initial canvas length and improves as severe under-allocation is relieved. DAEDAL effectively expands short canvases, but its benefit becomes less consistent once the initial canvas is already near or beyond the sufficient-length region. SACD remains stable across different initial lengths by performing conservative EXPAND, SHRINK, and KEEP decisions. The vertical dashed line denotes the average final canvas length reached by DAEDAL.

Benchmark	Method	Acc.	Lat.	NOR
GSM8K	DAEDAL	63.3	38.8	73.0
	LR-DLLM	75.1	84.6	21.1
	SACD	84.5	34.6	75.6
MATH-500	DAEDAL	26.4	35.2	76.4
	LR-DLLM	39.8	92.7	13.6
	SACD	42.2	34.3	85.6
HumanEval	DAEDAL	27.4	32.1	74.1
	LR-DLLM	31.7	83.2	33.7
	SACD	42.1	35.8	75.6
MBPP	DAEDAL	30.2	34.9	81.5
	LR-DLLM	32.0	89.3	23.0
	SACD	37.4	34.6	75.6

Table 3: Comparison with training-free variable-length decoding baselines on LLaDA-8B-Instruct under severe short-canvas initialization. All methods are initialized with $L_{\text{init}} = 1$ to stress-test their ability to recover from extreme under-allocation and to compare inference overhead. This setting is intentionally more challenging than the commonly used $L_{\text{init}} = 64$ setting for expansion-based decoding; we further analyze sensitivity to initial canvas length in Figure 3.

length-sensitivity analysis in Section 4.4.

As shown in Table 3, SACD achieves the best accuracy on all four benchmarks under this stress-test setting. Compared with DAEDAL, SACD obtains substantially higher accuracy while maintaining comparable latency. This suggests that expansion triggers alone may be insufficient when the initial canvas is severely under-allocated, since the decoder must not only add space but also decide when intervention is reliable. Compared with LR-DLLM, SACD achieves both higher accuracy and much lower latency. This reflects the difference between repeated length search and sparse trajectory-

preserving intervention: LR-DLLM explores candidate lengths and commits tokens more serially, whereas SACD preserves the base parallel denoising schedule and intervenes only at selected detection steps.

The NOR results provide additional evidence for the conservative nature of SACD. SACD maintains a high no-op rate across tasks, indicating that it does not recover from short canvases through continuous editing. Instead, it performs sparse interventions while keeping most denoising steps unchanged. Compared with LR-DLLM, SACD has a substantially higher NOR and lower latency, showing that effective recovery from under-allocation does not require frequent length search. Compared with DAEDAL, SACD achieves much stronger accuracy at a similar level of intervention sparsity, suggesting that the timing and safety of intervention are more important than expansion alone.

Overall, this comparison supports our main claim: variable-length diffusion decoding benefits not from maximizing the number of canvas edits, but from intervening only when the canvas is clearly misallocated and the edit is unlikely to disrupt the denoising trajectory.

4.4 Length Sensitivity and Robustness to Canvas Initialization

We further study how decoding methods behave under different initial canvas lengths. This analysis complements the severe short-canvas stress test in Table 3: instead of evaluating a single initialization, it examines whether a variable-length decoder remains stable when the initial canvas is either insufficient, near-sufficient, or over-allocated.

Figure 3 shows the results on LLaDA-8B-Instruct. Fixed-length denoising is highly sensitive to the initial canvas length. Its accuracy im-

Model	Method	Pass@1	
		Single-line	Multi-line
LLaDA-8B	Baseline	48.3	21.1
	+ FlexMDM	65.0	-
	+ LR-DLLM	69.4	35.9
	+ SACD	73.1	38.5
LLaDA 1.5	Baseline	48.8	32.2
	+ LR-DLLM	68.9	34.6
	+ SACD	72.4	36.7
Dream-7B	Baseline	48.2	38.4
	+ LR-DLLM	76.7	46.3
	+ SACD	78.5	47.2
DreamCoder-7B	Baseline	55.5	43.2
	+ DreamOn	73.5	36.1
	+ LR-DLLM	81.6	53.8
	+ SACD	85.3	67.0

Table 4: HumanEval-Infilling results under severe short-canvas initialization. All methods start with an initial mask length of 1, requiring the decoder to expand the canvas during inference.

proves as the canvas becomes longer, especially from $L = 64$ to $L = 512$, indicating that short canvases under-allocate generation space. However, the gains eventually saturate, suggesting that simply increasing the canvas length does not continuously improve generation and may leave redundant regions unresolved.

DAEDAL substantially improves over short fixed-length baselines by expanding insufficient canvases. This confirms that expansion is useful when the initial canvas is clearly under-allocated. However, because DAEDAL is primarily expansion-oriented, its benefit becomes less consistent once the initial canvas is already close to or beyond the sufficient length region. In such cases, the main challenge is no longer only to add more space, but also to avoid unnecessary intervention and remove redundant canvas regions when appropriate.

In contrast, SACD remains consistently strong across different initial lengths. Its performance is much less sensitive to the starting canvas because it can choose among EXPAND, SHRINK, and KEEP according to both allocation evidence and intervention safety. When the canvas is short, SACD can expand conservatively; when the canvas is long, it can shrink redundant suffixes; and when the evidence is ambiguous, it preserves the current denoising trajectory through KEEP. These results support the view that robust variable-length diffusion decoding requires safe canvas control rather than expansion

alone.

4.5 Results on Code Infilling

Table 4 evaluates SACD on HumanEval-Infilling with an initial mask length of 1. This setting represents a severe under-allocation scenario: the missing span length is unknown in advance, and the decoder must create sufficient canvas space during inference while preserving the surrounding code context.

Across all evaluated backbones, SACD consistently improves over the fixed-length baseline and achieves the best performance among available variable-length methods. On LLaDA-8B, SACD reaches 73.1 single-line and 38.5 multi-line Pass@1, outperforming LR-DLLM in both settings. On Dream-7B, SACD further improves over LR-DLLM, and on DreamCoder-7B (Xie et al., 2025) it obtains especially large gains in multi-line infilling, improving Pass@1 from 53.8 to 67.0.

These results show that SACD is not limited to shrinking over-allocated canvases. It can also recover from severe under-allocation by expanding the canvas when additional space is necessary. Importantly, this adaptation is performed through sparse safety-aware intervention rather than continuous length search. Thus, SACD provides a unified canvas-control mechanism for both standard generation and infilling, where the required output length is highly uncertain.

5 Conclusion

We proposed SACD, a training-free framework for safe adaptive canvas control in diffusion language models. By treating canvas editing as a potentially disruptive intervention, SACD performs sparse EXPAND, SHRINK, and KEEP decisions based on both allocation and safety evidence. Experiments across reasoning, code generation, and infilling tasks show that SACD improves accuracy, latency, and token utilization across multiple DLM backbones. These results highlight safe sparse intervention as a key principle for variable-length diffusion decoding.

Limitations

SACD is a training-free framework and therefore relies on lightweight heuristic signals, including EOT probability, entropy, and boundary stability, to estimate canvas allocation and intervention risk. Although effective in our experiments, these signals may behave differently across model families,

decoding schedules, and tasks, and may require modest adjustment in new settings. Our evaluation mainly focuses on LLaDA- and Dream-style DLMs on reasoning, code generation, and infilling benchmarks; broader scenarios such as open-ended generation, dialogue, and multilingual tasks remain to be explored. In addition, SACD follows a conservative intervention principle, which improves safety but may miss cases where more aggressive adaptation is useful. Future work may develop learned or adaptive controllers that retain SACD’s safety principle while improving flexibility.

Acknowledgments

This work was supported by the National Natural Science Foundation of China (Grant No.62506318); Guangdong Provincial Department of Education Project (Grant No.2024KQNCX028); Scientific Research Projects for the Higher-educational Institutions (Grant No.2024312096), Education Bureau of Guangzhou Municipality; Guangzhou-HKUST(GZ) Joint Funding Program (Grant No.2025A03J3957), Education Bureau of Guangzhou Municipality.

References

- Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021a. [Structured denoising diffusion models in discrete state-spaces](#). In *Advances in Neural Information Processing Systems*.
- Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and Charles Sutton. 2021b. [Program synthesis with large language models](#). *Preprint*, arXiv:2108.07732.
- Mohammad Bavarian, Heewoo Jun, Nikolas Tezak, John Schulman, Christine McLeavey, Jerry Tworek, and Mark Chen. 2022. [Efficient training of language models to fill in the middle](#). *Preprint*, arXiv:2207.14255.
- Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde de Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, Alex Ray, Raul Puri, Gretchen Krueger, Michael Petrov, Heidy Khlaaf, Girish Sastry, Pamela Mishkin, Brooke Chan, Scott Gray, and 39 others. 2021. [Evaluating large language models trained on code](#). *Preprint*, arXiv:2107.03374.
- Zicong Cheng, Ruixuan Jia, Jia Li, Guo-Wei Yang, Meng-Hao Guo, and Shi-Min Hu. 2026. [Improving variable-length generation in diffusion language models via length regularization](#). *Preprint*, arXiv:2602.07546.
- 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. 2021. [Training verifiers to solve math word problems](#). *Preprint*, arXiv:2110.14168.
- Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and Lingpeng Kong. 2023. [Diffuseq: Sequence to sequence text generation with diffusion models](#). In *The Eleventh International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021. [Measuring mathematical problem solving with the MATH dataset](#). In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Jaeyeon Kim, Lee Cheuk Kit, Carles Domingo-Enrich, Yilun Du, Sham M. Kakade, Timothy Ngatiaoco, Sit-tan Chen, and Michael Samuel Albergo. 2026. [Any-order flexible length masked diffusion](#). In *The Fourteenth International Conference on Learning Representations*.
- Jinsong Li, Xiaoyi Dong, Yuhang Zang, Yuhang Cao, Jiaqi Wang, and Dahua Lin. 2026. [Beyond fixed: Training-free variable-length denoising for diffusion large language models](#). In *The Fourteenth International Conference on Learning Representations*.
- Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, JUN ZHOU, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2026. [Large language diffusion models](#). In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Yi Ren, Jinglin Liu, Xu Tan, Zhou Zhao, Sheng Zhao, and Tie-Yan Liu. 2020. [A study of non-autoregressive model for sequence generation](#). In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 149–159, Online. Association for Computational Linguistics.
- Subham Sekhar Sahoo, Marianne Arriola, Aaron Gokaslan, Edgar Mariano Marroquin, Alexander M Rush, Yair Schiff, Justin T Chiu, and Volodymyr Kuleshov. 2024. [Simple and effective masked diffusion language models](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. 2024. [Simplified and generalized masked diffusion for discrete data](#). In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*.
- Jiaming Song, Chenlin Meng, and Stefano Ermon. 2021. [Denoising diffusion implicit models](#). In *International Conference on Learning Representations*.
- Jason Wei, Xuezhi Wang, Dale Schuurmans, Maarten Bosma, brian ichter, Fei Xia, Ed H. Chi, Quoc V Le,

and Denny Zhou. 2022. [Chain-of-thought prompting elicits reasoning in large language models](#). In *Advances in Neural Information Processing Systems*.

Qingyan Wei, Yaojie Zhang, Zhiyuan Liu, Puyu Zeng, Yuxuan Wang, Biqing Qi, Dongrui Liu, and Linfeng Zhang. 2026. [Accelerating diffusion large language models with slowfast sampling: The three golden principles](#). In *The Fourteenth International Conference on Learning Representations*.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025. [Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding](#). *Preprint*, arXiv:2505.22618.

Zirui Wu, Lin Zheng, Zhihui Xie, Jiacheng Ye, Jiahui Gao, Shansan Gong, Yansong Feng, Zhenguo Li, Wei Bi, Guorui Zhou, and Lingpeng Kong. 2026. [Dream: Diffusion language models for code infilling beyond fixed-size canvas](#). In *The Fourteenth International Conference on Learning Representations*.

Zhihui Xie, Jiacheng Ye, Lin Zheng, Jiahui Gao, Jingwei Dong, Zirui Wu, Xueliang Zhao, Shansan Gong, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. [Dream-coder 7b: An open diffusion language model for code](#). *Preprint*, arXiv:2509.01142.

Yicun Yang, Cong Wang, Shaobo Wang, Zichen Wen, Biqing Qi, Hanlin Xu, and Linfeng Zhang. 2025. [Diffusion llm with native variable generation lengths: Let \[eos\] lead the way](#). *Preprint*, arXiv:2510.24605.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025. [Dream 7b: Diffusion large language models](#). *Preprint*, arXiv:2508.15487.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2025a. [Llada 1.5: Variance-reduced preference optimization for large language diffusion models](#). *Preprint*, arXiv:2505.19223.

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tiejun Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, and 7 others. 2025b. [Llada-moe: A sparse moe diffusion language model](#). *Preprint*, arXiv:2509.24389.

A Implementation Details of SACD

This appendix provides the implementation details of SACD, including the safety predicates used for intervention decisions and the corresponding hyperparameters. While the main text presents SACD as a conservative risk-aware policy, our implementation evaluates intervention safety through Boolean unsafe predicates rather than a single scalar risk

threshold. An edit is rejected whenever any corresponding unsafe condition is triggered.

A.1 Position-wise Signals

At each detection step t , the diffusion language model outputs a token distribution

$$p_t(v | i) = p_\theta(y_i = v | c, \mathbf{y}_t), \quad v \in \mathcal{V}, \quad (14)$$

for each response position i . We compute two position-wise signals: the EOT probability

$$E_t(i) = p_t(\text{eot} | i), \quad (15)$$

and the normalized predictive entropy

$$H_t(i) = -\frac{1}{\log |\mathcal{V}|} \sum_{v \in \mathcal{V}} p_t(v | i) \log p_t(v | i). \quad (16)$$

A high EOT probability suggests that the model regards position i as close to a termination region, while high entropy indicates unresolved content uncertainty.

To reduce local noise, we use window-smoothed statistics:

$$\begin{aligned} \bar{E}_t(i) &= \frac{1}{|W(i)|} \sum_{j \in W(i)} E_t(j), \\ \bar{H}_t(i) &= \frac{1}{|W(i)|} \sum_{j \in W(i)} H_t(j), \end{aligned} \quad (17)$$

where $W(i)$ is a local window centered at position i .

A.2 Boundary Estimation

SACD estimates a tentative termination boundary by combining EOT confidence and entropy:

$$B_t(i) = z(\log(\bar{E}_t(i) + \epsilon)) - \lambda z(\bar{H}_t(i)), \quad (18)$$

where $z(\cdot)$ denotes per-sample standardization over positions, ϵ is a small constant for numerical stability, and λ controls the entropy penalty. The estimated boundary is

$$\hat{b}_t = \arg \max_{1 \leq i \leq L_t} B_t(i). \quad (19)$$

Since a single boundary estimate may be noisy, SACD maintains a short boundary history

$$\mathcal{B}_t = \{\hat{b}_{t-K(h-1)}, \dots, \hat{b}_{t-K}, \hat{b}_t\}, \quad (20)$$

where K is the detection interval and h is the history size. Boundary stability is measured by

$$\text{Stab}_t = \text{Var}(\mathcal{B}_t). \quad (21)$$

A low variance indicates that the termination region has stabilized across recent detection steps.

A.3 Canvas State Diagnosis

SACD diagnoses the canvas state as UNDER, OVER, or SELF using conservative threshold rules.

Let $\mathcal{T}_t$ denote the tail region of the current canvas:

$$\mathcal{T}_t = \{L_t - w_{\text{tail}} + 1, \dots, L_t\}, \quad (22)$$

where w_{tail} is the tail-window size. We define tail uncertainty and tail EOT confidence as

$$\begin{aligned} H_t^{\text{tail}} &= \frac{1}{|\mathcal{T}_t|} \sum_{i \in \mathcal{T}_t} \bar{H}_t(i), \\ E_t^{\text{tail}} &= \frac{1}{|\mathcal{T}_t|} \sum_{i \in \mathcal{T}_t} \bar{E}_t(i). \end{aligned} \quad (23)$$

The canvas is diagnosed as under-allocated only when the estimated boundary is close to the right edge and the tail remains uncertain:

$$\begin{aligned} d_t = \text{UNDER} \quad \text{if} \quad & \hat{b}_t \geq (1 - \rho_{\text{edge}})L_t, \\ & H_t^{\text{tail}} \geq \tau_H^{\text{tail}}, \\ & E_t^{\text{tail}} \leq \tau_E^{\text{tail}}. \end{aligned} \quad (24)$$

For over-allocation, let $\mathcal{S}_t$ denote the candidate suffix after the estimated boundary:

$$\mathcal{S}_t = \{\hat{b}_t + m_{\text{min}}, \dots, L_t\}, \quad (25)$$

where m_{min} is the minimum preserved margin. We compute

$$\begin{aligned} E_t^{\text{suf}} &= \frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} \bar{E}_t(i), \\ H_t^{\text{suf}} &= \frac{1}{|\mathcal{S}_t|} \sum_{i \in \mathcal{S}_t} \bar{H}_t(i). \end{aligned} \quad (26)$$

The canvas is diagnosed as over-allocated only when the boundary is stable and the suffix is likely redundant:

$$\begin{aligned} d_t = \text{OVER} \quad \text{if} \quad & \text{Stab}_t \leq \tau_{\text{stab}}, \\ & E_t^{\text{suf}} \geq \tau_E^{\text{suf}}, \\ & H_t^{\text{suf}} \leq \tau_H^{\text{suf}}, \\ & L_t - \hat{b}_t \geq \Delta L_{\text{min}}. \end{aligned} \quad (27)$$

If neither condition is satisfied, SACD assigns

$$d_t = \text{SELF}. \quad (28)$$

This default assignment prevents SACD from editing the canvas under ambiguous evidence.

A.4 Intervention Safety Predicates

In the main text, SACD is described as a risk-aware intervention policy. In implementation, intervention safety is evaluated through Boolean unsafe predicates rather than a single scalar risk threshold. An edit is rejected whenever any corresponding unsafe condition is triggered.

Shrink safety predicate. For SHRINK, SACD rejects the edit if the candidate suffix may still contain useful content, if the estimated boundary is unstable, or if the expected efficiency gain is too small. Let σ_b^2 denote the variance of recent boundary estimates, $\mathcal{S}_t$ denote the candidate suffix to be removed, and $L_t - L'_t$ denote the expected shrink gain. The unsafe predicate for shrinking is:

$$\begin{aligned} \mathcal{U}_t^{\text{SHRINK}} &= [\sigma_b^2 \geq 0.02] \\ &\quad \vee \left[\max_{i \in \mathcal{S}_t} (1 - E_t(i)) > 0.3 \right] \\ &\quad \vee \left[\max_{i \in \mathcal{S}_t} H_t(i) > 0.4 \right] \\ &\quad \vee [L_t - L'_t < 32]. \end{aligned} \quad (29)$$

The four conditions correspond to boundary instability, possible non-EOT content in the removed suffix, high uncertainty in the removed suffix, and insufficient shrink gain, respectively.

Expand safety predicate. For EXPAND, SACD rejects the edit when the tail evidence for under-allocation is weak, when the model already shows signs of self-termination, or when expansion violates decoding-budget constraints. Let ΔH_t denote the entropy gap between the tail region and the main canvas region, E_t^{tail} denote the average EOT probability in the tail region, B_t^{commit} denote the estimated per-step commit budget, and L_t denote the current canvas length. The unsafe predicate for expansion is:

$$\begin{aligned} \mathcal{U}_t^{\text{EXPAND}} &= [\Delta H_t < 0.5] \\ &\quad \vee [E_t^{\text{tail}} > 0.35] \\ &\quad \vee [B_t^{\text{commit}} > 256] \\ &\quad \vee [L_t \geq 1024]. \end{aligned} \quad (30)$$

The first condition rejects expansion when tail uncertainty is not sufficiently stronger than the main canvas. The second rejects expansion when the model already predicts termination near the tail, suggesting possible self-regulation. The last two conditions enforce the per-step commit budget and maximum canvas length.

Shrink lock. After a SHRINK operation, SACD disables subsequent EXPAND actions for the remainder of the current generation. This lock is reset between different input examples, but it is not released within a single decoding trajectory. The design prevents shrink–expand oscillation after SACD has identified and removed a redundant suffix.

A.5 Action Selection

The final action is selected by combining the diagnosed canvas state with the intervention-safety predicates. Let $\mathcal{U}_t^{\text{SHRINK}}$ and $\mathcal{U}_t^{\text{EXPAND}}$ denote whether shrinking or expanding is considered unsafe at step t , respectively. SACD selects the action as:

$$a_t = \begin{cases} \text{SHRINK}, & d_t = \text{OVER} \text{ and } \neg \mathcal{U}_t^{\text{SHRINK}}, \\ \text{EXPAND}, & d_t = \text{UNDER} \text{ and } \neg \mathcal{U}_t^{\text{EXPAND}}, \\ \text{KEEP}, & \text{otherwise.} \end{cases} \quad (31)$$

Thus, SACD edits the canvas only when the diagnosis indicates a clear allocation problem and no corresponding unsafe condition is triggered. In all ambiguous or risky cases, SACD selects KEEP, preserving the current denoising trajectory.

A.6 Canvas Editing Details

Shrink. When $a_t = \text{SHRINK}$, SACD truncates the suffix after the estimated boundary while preserving an adaptive margin:

$$L'_t = \hat{b}_t + m_t. \quad (32)$$

The margin is computed as

$$m_t = m_0 + \alpha \sqrt{\text{Stab}_t} + \beta(1 - s_t), \quad (33)$$

where m_0 is the base margin, and s_t is the normalized boundary strength:

$$s_t = \sigma \left(B_t(\hat{b}_t) - \frac{1}{L_t} \sum_{i=1}^{L_t} B_t(i) \right). \quad (34)$$

The canvas is updated as

$$\mathbf{y}_t \leftarrow \mathbf{y}_t[1 : L'_t]. \quad (35)$$

After shrinking, SACD activates a shrink lock for K_{lock} detection steps, during which expansion is disabled to avoid edit oscillation.

Symbol	Description	Value
K	Detection interval	10
h	Boundary history size	5
w	Local smoothing window	5
w_{tail}	Tail-window size	10
λ	Entropy penalty in boundary score	0.5
ρ_{edge}	Threshold for under-allocation	0.90
τ_H^{tail}	Tail-entropy threshold	0.5
τ_E^{tail}	Tail-EOT threshold	0.5
ΔL_{max}	Maximum expansion size	128
L_{max}	Maximum canvas budget	1024
m_0	Base shrink margin	20
α, β	Margin coefficients	30.0, 15.0

Table 5: General hyperparameters of SACD. Values are fixed across tasks and models unless otherwise specified.

Expand. When $a_t = \text{EXPAND}$, SACD appends new mask tokens:

$$\mathbf{y}_t \leftarrow \mathbf{y}_t \parallel [\mathbf{M}]^{\Delta L_t}. \quad (36)$$

The expansion size is bounded by

$$\Delta L_t = \min(\Delta L_{\text{max}}, L_{\text{max}} - L_t). \quad (37)$$

This gradual expansion allows the model to reassess the enlarged canvas at later detection steps instead of jumping directly to a large canvas.

Keep. When $a_t = \text{KEEP}$, no canvas edit is performed.

A.7 Hyperparameters

Table 5 lists the hyperparameters used by SACD. We use a single set of hyperparameters across benchmarks unless otherwise specified.

A.8 SACD Algorithm

B Evaluation Metric Definitions

Let $z_i^{\text{ref}} \in \{0, 1\}$ denote whether the fixed-length reference answers example i correctly, and let $z_i^{\text{adapt}} \in \{0, 1\}$ denote whether the adaptive decoding method answers it correctly. We define the Intervention Harm Rate as

$$\text{IHR} = \frac{1}{N} \sum_{i=1}^N \mathbb{I} \left[z_i^{\text{ref}} = 1 \wedge z_i^{\text{adapt}} = 0 \right]. \quad (38)$$

The Rescue Rate is defined as

$$\text{RR} = \frac{1}{N} \sum_{i=1}^N \mathbb{I} \left[z_i^{\text{ref}} = 0 \wedge z_i^{\text{adapt}} = 1 \right]. \quad (39)$$

Condition / Symbol	Description	Value
<i>Shrink rejection conditions</i>		
σ_b^2	Boundary variance threshold	0.02
$\max_{i \in S_t} (1 - E_t(i))$	Max non-EOT probability in removed suffix	0.3
$\max_{i \in S_t} H_t(i)$	Max entropy in removed suffix	0.4
$\Delta L_{\min}$	Minimum shrink gain	32
<i>Expand rejection conditions</i>		
ΔH_t	Tail-main entropy gap threshold	0.5
E_t^{tail}	Tail-EOT threshold for expansion rejection	0.35
B_t^{commit}	Maximum per-step commit budget	256
$L_{\max}$	Maximum canvas length	1024
Shrink lock	Expansion disabled after shrink	Until generation ends

Table 6: Safety-related hyperparameters used in SACD. Shrink and expand decisions are implemented as Boolean unsafe predicates: an intervention is rejected if any corresponding condition is triggered.

Let a_t denote the canvas-control action at detection step t . The No-Op Rate is defined as

$$\text{NOR} = \frac{\sum_t \mathbb{I}[a_t = \text{KEEP}]}{\sum_t 1}. \quad (40)$$

We define token utilization as

$$\text{Util.} = \frac{L_{\text{eff}}}{L_{\text{final}}}, \quad (41)$$

where L_{eff} is the effective length before the first EOT token and L_{final} is the final canvas length after decoding.

When comparing against a fixed reference length L_{ref} , we also report canvas saving:

$$\text{Save} = 1 - \frac{L_{\text{final}}}{L_{\text{ref}}}. \quad (42)$$

C Complexity Analysis

We analyze the inference cost of SACD from two perspectives: the cost of denoising over a variable-length canvas and the additional overhead introduced by the canvas controller.

Let $C_\theta(L)$ denote the cost of one forward pass of the underlying DLM with canvas length L . Since DLMs use bidirectional attention over the response canvas, the dominant term scales quadratically with the canvas length:

$$C_\theta(L) = \mathcal{O}(L^2). \quad (43)$$

For fixed-length decoding with initial canvas length L_0 and T denoising steps, the total cost is

$$C_{\text{fixed}} = \sum_{t=1}^T C_\theta(L_0) = \mathcal{O}(TL_0^2). \quad (44)$$

SACD allows the canvas length to change during denoising. Let L_t be the canvas length at step t . The denoising cost becomes

$$C_{\text{SACD}} = \sum_{t=1}^T C_\theta(L_t) + C_{\text{ctrl}}, \quad (45)$$

where C_{ctrl} is the cost of the risk-aware controller. Since SACD invokes the controller only every K steps and computes diagnosis signals from the model’s current predictions, the controller does not require per-token length search. Its overhead is

$$C_{\text{ctrl}} = \mathcal{O}\left(\frac{T}{K} L_t\right), \quad (46)$$

when prediction statistics are reused, which is negligible compared with the quadratic attention cost. Therefore, SACD mainly changes the total cost through the sequence of canvas lengths $\{L_t\}_{t=1}^T$.

We define the relative attention saving as

$$\text{Saving} = 1 - \frac{\sum_{t=1}^T L_t^2}{TL_0^2}. \quad (47)$$

This quantity measures the reduction in bidirectional attention computation compared with fixed-length decoding using the same initial canvas length. When SACD safely shrinks an over-allocated canvas, later denoising steps operate on a shorter canvas, reducing the dominant attention cost.

Algorithm 1 Safe Adaptive Canvas Decoding

```
1: Initialize  $\mathbf{y}_T = [\mathbf{M}]^{L_{\text{init}}}$ 
2: Initialize boundary history  $\mathcal{B} \leftarrow \emptyset$ 
3: for denoising step  $t = T, \dots, 1$  do
4:   Run the base DLM scheduler to update
   masked positions
5:   if  $t$  is a detection step then
6:     Compute  $E_t(i)$  and  $H_t(i)$  for all posi-
     tions
7:     Estimate boundary  $\hat{b}_t$  and update  $\mathcal{B}$ 
8:     Diagnose canvas state  $d_t \in$ 
      $\{\text{UNDER}, \text{OVER}, \text{SELF}\}$ 
9:     Compute intervention risks
      $R_t(\text{EXPAND})$  and  $R_t(\text{SHRINK})$ 
10:    Select action  $a_t$  according to the con-
    servative policy
11:    if  $a_t = \text{SHRINK}$  then
12:      Truncate suffix with adaptive mar-
      gin and activate shrink lock
13:    else if  $a_t = \text{EXPAND}$  then
14:      Append  $\Delta L_t$  mask tokens if budget
      allows
15:    else
16:      Keep the canvas unchanged
17:    end if
18:  end if
19: end for
20: return decoded output
```
