

# From Decoding Knobs to Commit Frontiers: Understanding Trajectory Control in Discrete Diffusion Language Models

Junzhuo Li<sup>†‡</sup>, Bo Wang<sup>†‡</sup>, Tianhao Li<sup>¶</sup>, Haoran Wei<sup>¶</sup>,  
Baosong Yang<sup>¶</sup>, and Xuming Hu<sup>†‡\*</sup>

<sup>†</sup>The Hong Kong University of Science and Technology (Guangzhou)

<sup>‡</sup>The Hong Kong University of Science and Technology

<sup>¶</sup>Qwen Team, Alibaba Group

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

xuminghu@hkust-gz.edu.cn

## Abstract

Discrete diffusion language models generate text by iteratively committing masked positions, making decoding a trajectory-level process rather than a sequence of independent token decisions. Existing analyses often focus on surface-level knobs such as temperature, step budget, block size, or scheduler choice, but these knobs do not explain why the same change can induce different behaviors across models. We analyze discrete diffusion decoding through high-influence commitment events and define an observed trajectory influence measure based on downstream entropy reduction over unresolved positions. Across Dream-v0-Instruct-7B and LLaDA-8B-Instruct on reasoning, code, and question-answering tasks, we find that influence is sparse, early relative to each model’s commit frontier, and stable under tested seed and step-budget variations. We further show that criticality emerges from the interaction between commit timing and token role, explaining why decoding knobs reshape trajectories in model-dependent ways. Finally, lightweight margin-based interventions validate the analysis by correcting diagnosed failure modes without additional forward passes.

## 1 Introduction

Discrete diffusion language models (DLMs) (Austin et al., 2021a; Sahoo et al., 2024; Nie et al., 2026; Ye et al., 2025; Zhu et al., 2025a,b) generate text by iteratively denoising a masked canvas. Unlike autoregressive models, which commit tokens in a fixed left-to-right order, diffusion language models must decide which masked positions to resolve at each step. Decoding is therefore not merely a token prediction problem, but a trajectory control problem: early position commitments can reshape the uncertainty of many positions that remain unresolved.

Most practical decoding procedures expose this process through surface-level knobs, such as temperature, step budget, block size, generation length, and scheduler choice. These knobs are useful for controlling generation, but they are limited as explanations. The same nominal change can induce different trajectory structures across models, because each scheduler maps uncertainty estimates into position commitments in a different way. This suggests that understanding diffusion decoding requires looking beneath decoding knobs and analyzing the commitment trajectory they induce.

In this paper, we study discrete diffusion decoding through high-influence position commitments. We define an observed influence measure that quantifies how much a commitment event is associated with downstream entropy reduction over unresolved positions. This measure is descriptive rather than counterfactual: it does not claim to isolate the causal effect of a single position, but it provides a trajectory-level diagnostic for identifying commitments that coincide with large reductions in future uncertainty. We use this diagnostic to characterize the commit frontier, the set of positions exposed early enough by a scheduler to affect many later decisions. Figure 1 provides an overview of our framework, from commitment trajectories to observed influence, commit frontiers, and lightweight validation.

Our analysis follows a three-stage path. We first show that trajectory influence is highly concentrated: a small number of commitments accounts for a large fraction of total observed influence. These high-influence commitments appear early relative to each model’s own commit frontier, rather than early in absolute sequence position, and the resulting structure is stable under the tested seed and step-budget variations. We then analyze what makes these frontier commitments critical. Criticality is not determined by token category or position alone; instead, it emerges from the interaction

\*Corresponding author

between commit timing and token role. Boundary-like tokens dominate under some schedulers, while semantic content, numerical tokens, delimiters, and whitespace become critical under others.

This commit-frontier view also clarifies the effect of decoding knobs. Temperature, generation length, step budget, and block size do not act as universal controls. Rather, they reshape, relocate, or preserve the commit frontier in model-dependent ways. In our experiments, temperature has the strongest structural effect, but its direction differs across models: it tends to concentrate and relocate influence in Dream, while often diffusing or preserving influence in LLaDA. Thus, the behavioral effect of a decoding knob is best understood through the scheduler-specific trajectory structure it induces.

Finally, we use lightweight margin-based interventions as validation cases for the analysis. The commit-frontier view diagnoses different failure modes in different models: premature EOS commitments in Dream and unreliable low-margin commitments in LLaDA. Simple targeted interventions address these failures without additional forward passes, showing that trajectory analysis can provide actionable model-specific insight. These interventions are not intended as a universal decoding algorithm; they serve as evidence that commit-frontier analysis can connect trajectory understanding with lightweight control.

Our contributions are summarized as follows:

- We propose a trajectory-level framework for analyzing discrete diffusion decoding through observed influence among position commitments.
- We show that high-influence commitments are sparse, early relative to model-specific commit frontiers, and stable in the tested settings.
- We identify a commit-timing–token-role interaction that explains why different schedulers expose different critical positions.
- We show that common decoding knobs reshape trajectories through the commit frontier, and validate the analysis with lightweight model-specific interventions.

## 2 Related Work

**Discrete diffusion language models, iterative refinement, and decoding control.** Our work

builds on discrete diffusion language models and non-left-to-right iterative generation. Discrete diffusion models define denoising processes over discrete or absorbing masked states (Austin et al., 2021a; Hooeboom et al., 2022; Sahoo et al., 2024), and recent work has scaled this paradigm to instruction-tuned language models, code generation, and block- or architecture-adapted diffusion models (Gong et al., 2025; Arriola et al., 2025; Nie et al., 2026; Ye et al., 2025; Xie et al., 2025). Related non-autoregressive models, including Mask-Predict, insertion models, and edit-based models, also generate by iteratively refining partially specified outputs (Ghazvininejad et al., 2019; Stern et al., 2019; Gu et al., 2019). Recent inference-time methods further improve dLM decoding by revising commitments, adapting computation, changing stopping criteria, or searching for better schedules (Wang et al., 2026; Zhai et al., 2026; Li et al., 2026; Mohamed et al., 2025; Horvitz et al., 2025; Chen et al., 2025a,b; Bansal and Sanghavi, 2026). In contrast to these algorithmic directions, we analyze realized commitment trajectories diagnostically, asking how existing schedulers and decoding knobs reshape the commit frontier.

**Influence, attribution, and trajectory diagnostics.** Our use of influence is related to attribution and interpretability, but differs in object and goal. Influence functions trace predictions to training examples (Koh and Liang, 2017), while causal tracing and model-editing methods localize internal activations or parameters involved in model behavior (Meng et al., 2022; Wu et al., 2023; Li et al., 2025). These approaches are typically counterfactual or intervention-based. By contrast, our observed trajectory influence is defined over the decoding trajectory itself: it measures how the uncertainty of unresolved positions changes after a commitment event. We therefore treat it as a descriptive trajectory signal, not as the isolated causal effect of a single position, and use it to compare how different schedulers and decoding knobs expose commit-frontier structures.

## 3 Trajectory Influence in Discrete Diffusion Decoding

Discrete diffusion language models generate text by iteratively refining a masked canvas. Let  $M_t$  denote the set of unresolved positions before decoding step  $t$ . At this step, the model predicts a token distribution for each position in  $M_t$ , and a

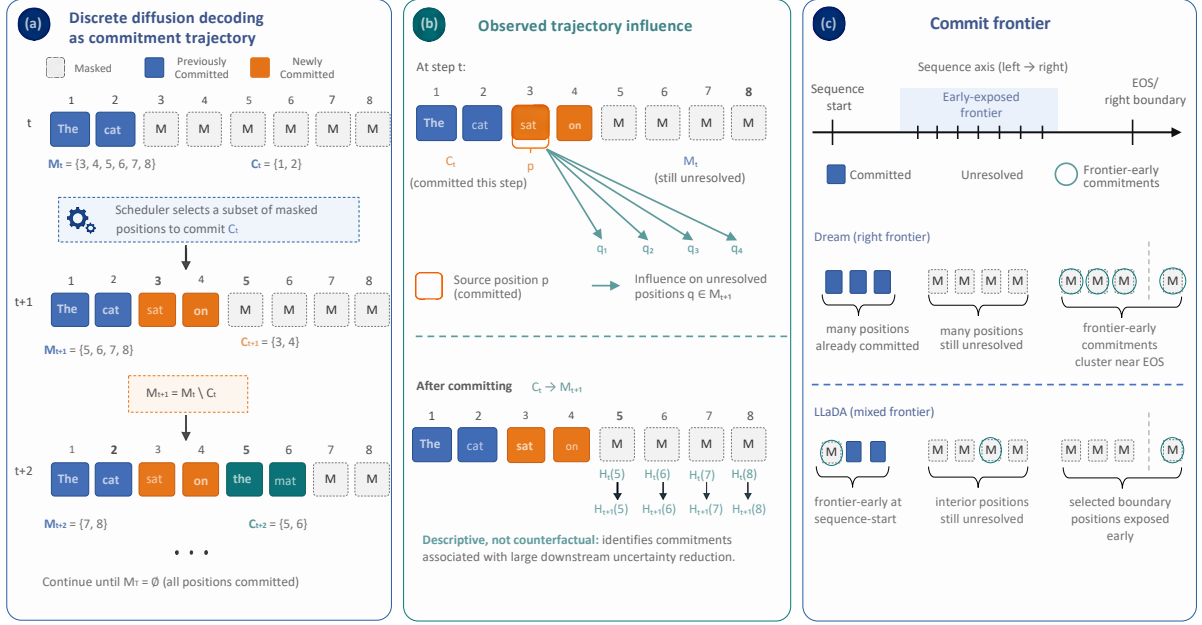


Figure 1: Overview of our commit-frontier analysis. Discrete diffusion decoding is formulated as a commitment trajectory in which schedulers iteratively expose masked positions. We define observed trajectory influence by measuring downstream entropy reduction over unresolved positions, and use it to identify scheduler-specific commit frontiers. The framework explains sparse critical commitments, frontier-role interaction, model-dependent knob effects, and lightweight intervention validation.

scheduler selects a subset of positions  $C_t \subseteq M_t$  to commit. Once committed, these positions are written into the canvas and removed from the unresolved set:

$$M_{t+1} = M_t \setminus C_t. \quad (1)$$

The resulting generation process is therefore a commitment trajectory rather than a fixed left-to-right sequence of token decisions.

This trajectory determines which positions become available as context while many other positions remain unresolved. We refer to the early exposed region of this commitment order as the model’s *commit frontier*. Operationally, the commit frontier is not a fixed absolute span in the sequence. Instead, it is the scheduler-induced set of positions committed early enough to potentially affect many later decisions. We characterize this frontier through commit order, high-influence source positions, source-region overlap, and the concentration of downstream uncertainty reduction.

### 3.1 Observed Trajectory Influence

We use entropy change to describe how commitment events are associated with future uncertainty reduction. Let  $H_t(q)$  denote the entropy of the model distribution at unresolved position  $q$  immedi-

ately before step  $t$ . After the scheduler commits  $C_t$ , the entropy at a remaining unresolved position  $q$  may change from  $H_t(q)$  to  $H_{t+1}(q)$ . For a committed position  $p \in C_t$ , we define its observed directed influence on an unresolved position  $q \in M_{t+1}$  as

$$w(p \rightarrow q) = \frac{H_t(q) - H_{t+1}(q)}{|C_t|}, \quad p \in C_t. \quad (2)$$

The position-level influence of  $p$  is then the mean outgoing influence over positions that remain unresolved after the commitment event:

$$I(p) = \mathbb{E}_{q \in M_{t+1}} [w(p \rightarrow q)]. \quad (3)$$

This score is descriptive rather than counterfactual. Since multiple positions may be committed at the same step, the entropy change from  $H_t$  to  $H_{t+1}$  is induced by the whole commit set  $C_t$ . The division by  $|C_t|$  provides a simple co-commitment attribution that avoids assigning the full step-level entropy reduction to every committed position. We therefore do not interpret  $I(p)$  as the isolated causal effect of forcing position  $p$  to be committed. Instead, it identifies positions that participate in commitment events associated with large downstream uncertainty reductions along the realized trajectory.

### 3.2 Trajectory Summaries and Baselines

We summarize each trajectory using several complementary statistics. Influence Gini measures how concentrated observed influence is across positions. Top- $k$  influence coverage measures how much total influence is accounted for by the most influential positions. Source-position overlap compares whether the same positions remain highly influential across decoding configurations, while rank correlation measures the similarity of commitment orders.

To distinguish trajectory-level influence from local prediction uncertainty, we also compare against a local decision-gap baseline. For a position committed at step  $t$ , the decision gap is the probability difference between the top-1 and top-2 predicted tokens at commit time. A small gap indicates local ambiguity, but it does not directly measure how much committing the position changes the uncertainty of other unresolved positions. This comparison allows us to test whether high-influence commitments capture structure beyond local confidence.

### 3.3 Experimental Setup

We analyze two representative discrete diffusion language models: Dream-v0-Instruct-7B (Ye et al., 2025), which uses an entropy-based scheduler, and LLaDA-8B-Instruct (Nie et al., 2026), which uses a confidence-based scheduler. Our tasks include GSM8K (Cobbe et al., 2021) and MATH-500 (Hendrycks et al., 2021b) for mathematical reasoning, HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021b) for code generation, and MMLU (Hendrycks et al., 2021a) for multiple-choice question answering. Generation length ranges from 64 to 512 tokens, decoding steps from 64 to 256, and temperature from 0.0 to 0.9. Unless otherwise stated, each configuration contains 100 examples.

For each decoding trajectory, we record step-level token distributions, entropy, commitment order, committed token values, and final outputs. In total, our analysis covers 27 model-task-decoding configurations. We additionally evaluate cross-seed stability on Dream GSM8K using four random seeds. These records allow us to compare how different models, tasks, and decoding knobs reshape the commit frontier.

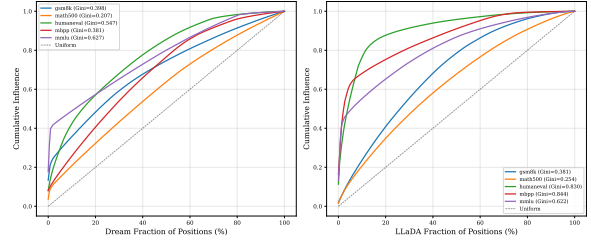


Figure 2: Lorenz curves of position influence for Dream and LLaDA at length 256 and  $T = 0.0$ .

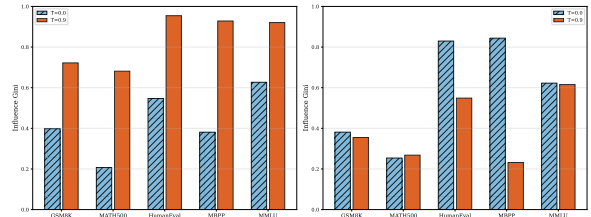


Figure 3: Cross-task influence Gini comparison at length 256.

## 4 Critical Commitments Are Sparse, Frontier-Early, and Stable

We first examine whether diffusion decoding is shaped by many small commitments or by a small number of high-influence events. We focus on three properties of the observed trajectory influence: concentration, frontier position, and stability.

### 4.1 Influence Is Highly Concentrated

Observed influence is highly uneven across positions. Figure 2 shows Lorenz curves for Dream and LLaDA at length 256 and  $T = 0.0$ , and Figure 3 summarizes cross-task influence Gini. Even under deterministic decoding, the top 20% of positions account for 32–89% of total observed influence, depending on the model and task.

The degree of concentration is both task- and scheduler-dependent. For LLaDA at  $T = 0.0$ , influence Gini ranges from 0.25 on MATH500 to 0.84 on MBPP, indicating much stronger concentration in code generation. Model choice also matters: on HumanEval, Dream has influence Gini 0.55, while LLaDA reaches 0.83. Thus, criticality is not only a property of the task, but also of the scheduler-induced commitment trajectory.

This structure is not captured by local ambiguity alone. Across our trajectory records, the Gini coefficient of the top-1/top-2 decision gap remains nearly flat around 0.025, whereas influence Gini ranges from 0.08 to 0.95. Observed influence therefore captures trajectory-level structure beyond local

|                                |  | Dream                   | LLaDA                               |
|--------------------------------|--|-------------------------|-------------------------------------|
| First five committed positions |  | 255, 252, 251, 163, 178 | 0, 91, 2, 249, 30                   |
| Commit pattern                 |  | EOS-region anchored     | Mixed with sequence-start positions |
| Top influence sources          |  | 255, 252, 251, 163, 178 | 0, 91, 2, 249, 30                   |

Table 1: Representative GSM8K comparison at length 256 and  $T = 0.0$ . Criticality follows each model’s commit frontier rather than a shared absolute position pattern.

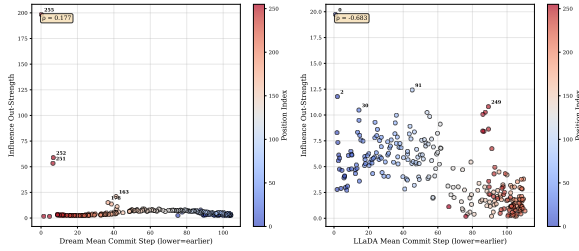


Figure 4: Per-position influence versus mean commit step. High-influence positions appear near the early commit frontier of each model.

prediction confidence.

## 4.2 Criticality Follows the Commit Frontier

High-influence commitments appear early relative to each model’s own commitment order. However, “early” does not mean early in absolute sequence position. A position becomes influential when it is committed while many other positions remain unresolved.

Table 1 shows a representative GSM8K comparison at length 256 and  $T = 0.0$ . Dream’s early influential commitments concentrate near the right boundary and EOS region, whereas LLaDA exposes a mixed frontier with sequence-start and selected boundary positions. Figure 4 shows the same pattern by plotting position influence against mean commit step.

This contrast suggests that Dream and LLaDA do not traverse the same dependency structure at different speeds. Instead, their schedulers expose different frontier positions, and these early commitments become the main channels through which downstream uncertainty is reduced.

## 4.3 Critical Structure Is Stable in Tested Settings

The critical structure is stable under the tested seed and step-budget variations. In the Dream GSM8K seed sweep, influence concentration, commitment order, and critical-set overlap remain unchanged across four seeds at both  $T = 0.0$  and  $T = 0.9$ . Across 64, 128, and 256 decoding steps, commit-order correlation remains above 0.99 for both Dream and LLaDA, suggesting that step budget mainly changes temporal granularity rather than the underlying commitment order.

At the same time, stability is scheduler-specific rather than universal. On GSM8K at length 256 and  $T = 0.0$ , Dream and LLaDA have negatively correlated commitment orders ( $\rho = -0.46$ ), reflecting opposite scheduling strategies. Thus, high-influence commitments are stable within the tested scheduler settings, but their locations and roles differ sharply across models.

## 4.4 Summary

This section establishes that diffusion decoding contains sparse, frontier-early, and stable high-influence commitments. These commitments are not simply the most locally ambiguous positions; they are positions exposed early enough by a scheduler to affect many unresolved decisions. The next section asks what makes these frontier commitments critical.

## 5 Criticality Emerges from Frontier–Role Interaction

Section 4 shows that high-influence commitments are sparse and frontier-dependent. We now ask what makes these frontier commitments critical. Our main finding is that criticality is not a property of token type or sequence position alone. Instead, it emerges from the interaction between *when* a position is committed and *what role* the committed token plays.

### 5.1 Critical Token Roles Depend on the Commit Frontier

We group committed tokens into broad role families. Structural and boundary tokens include EOS, whitespace, punctuation, and code delimiters. Semantic content tokens include identifiers, reasoning words, numbers, and programming keywords. Table 2 summarizes representative configurations.

| Configuration          | Dominant family        | Share                         | Interpretation                                                                           |
|------------------------|------------------------|-------------------------------|------------------------------------------------------------------------------------------|
| Dream GSM8K / MATH500  | Structural, mainly EOS | 98–100%                       | Dream exposes an EOS-heavy early frontier in math settings.                              |
| Dream HumanEval / MBPP | Structural + semantic  | 74% / 18%                     | EOS and formatting anchors dominate, with function names and keywords also contributing. |
| LLaDA GSM8K / MATH500  | Semantic               | 60–95%                        | The confidence-based frontier exposes reasoning words and numeric content.               |
| LLaDA HumanEval / MBPP | Mixed                  | 60% structural / 40% semantic | Delimiters and whitespace provide code structure, while identifiers carry content.       |
| LLaDA MMLU             | Structural, mainly EOS | 93%                           | Short-answer QA resembles EOS-heavy frontier settings.                                   |

Table 2: Dominant role families among high-influence positions. Critical token roles vary with the scheduler-induced commit frontier.

The key pattern is not that one token family is universally important. A token role becomes influential when it is placed on an early frontier where many positions remain unresolved. Dream’s entropy-based scheduler often exposes EOS-like boundary positions early in math tasks, whereas LLaDA exposes semantic content in math and a mixture of structural and semantic tokens in code. Thus, the identity of critical positions depends on how the scheduler constructs the commit frontier.

## 5.2 Influence Share and Enrichment Reveal Different Effects

Raw influence share can overstate the importance of frequent categories. We therefore compute frequency-normalized enrichment:

$$\text{Enrichment}(c) = \frac{\text{InfluenceShare}(c)}{\text{PositionShare}(c)}. \quad (4)$$

Figure 5 visualizes this distinction across token categories and configurations. This distinction is crucial in EOS-heavy settings. In Dream math tasks, EOS tokens account for most observed influence, but their enrichment is close to  $1.0\times$ . Individual EOS instances are therefore not unusually influential relative to their frequency; they dominate because the early frontier contains many EOS-like positions.

By contrast, rare code-structural tokens can be highly enriched. In LLaDA code tasks, code delimiters reach  $43.0\times$  enrichment on MBPP, while whitespace reaches  $28.8\times$  on HumanEval. These tokens are rare, but when committed early, they impose syntactic or formatting constraints on many unresolved positions. This supports the frontier–role view: influence depends jointly on what information a token carries and when that information becomes available.

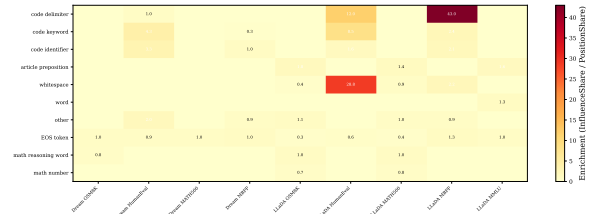


Figure 5: Frequency-normalized enrichment of token categories. Values above 1.0 indicate overrepresentation among high-influence positions, highlighting rare code-structural tokens in LLaDA code tasks.

## 5.3 Feature Analysis Supports the Frontier–Role View

We further test whether token role remains predictive when considered alongside structural position and graph-derived features. Figure ?? reports the adjusted  $R^2$  of ridge models using different feature groups. Semantic-role features explain substantial variance in LLaDA math and code settings, with adjusted  $R^2$  values between 0.34 and 0.48. Removing semantic features reduces  $R^2$  by 0.140 on LLaDA HumanEval and 0.091 on LLaDA MBPP. In contrast, Dream math tasks have near-zero semantic  $R^2$ , consistent with their EOS-heavy frontier structure.

Graph-coupling features achieve higher  $R^2$  values, ranging from 0.79 to 0.98, but they are partly circular because they are computed from the same influence graph as the target. We therefore use them only as an upper reference, not as an independent causal explanation.

## 5.4 Synthesis: Frontier $\times$ Role

These results suggest a compact mechanism: the commit frontier creates the opportunity for influ-

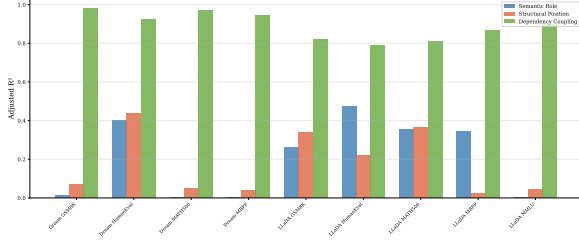


Figure 6: Feature groups explaining position criticality. Bars show adjusted  $R^2$  from ridge regressions predicting position-level influence. Semantic features are informative in LLaDA math and code settings, while dependency-coupling features provide an upper reference because they are derived from the influence graph.

ence, while token role determines what information the commitment contributes. A position becomes critical when it is committed early enough to affect many unresolved positions and when its token role carries constraints relevant to those positions.

This mechanism explains the observed model differences. Dream often places EOS-like positions on the early frontier, yielding high structural influence but low per-instance EOS enrichment. LLaDA exposes semantic content in math tasks and rare syntax-bearing tokens in code tasks, making reasoning words, numbers, delimiters, and whitespace predictive of influence. Criticality is therefore not a universal property of a token category or an absolute sequence location; it is produced by the interaction between scheduler-induced commit timing and the role of the committed token.

## 6 Decoding Knobs Reshape Commit Frontiers

Sections 4 and 5 show that diffusion decoding is shaped by sparse, frontier-dependent commitments whose influence depends on token role. We now return to the decoding knobs that practitioners directly control. Rather than treating temperature, generation length, step budget, and block size as independent mechanisms, we analyze them through three frontier-level effects: whether they change influence concentration, relocate high-influence source regions, or preserve the same commitment order while changing temporal granularity.

### 6.1 Temperature Has Model-Dependent Structural Effects

Temperature is the strongest source of frontier reconfiguration in our experiments, but its effect is not universal. Figure 7 shows that Dream and

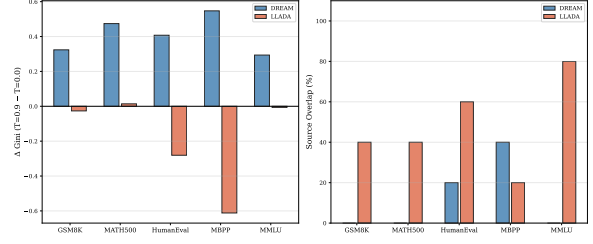


Figure 7: Model-dependent effects of temperature on commit-frontier structure. Higher temperature tends to concentrate and relocate influence in Dream, while it often diffuses or preserves the influence structure in LLaDA. This shows that the same decoding knob can reshape trajectories in scheduler-specific ways.

LLaDA often respond in opposite ways when temperature increases. For Dream, higher temperature consistently increases influence concentration and often relocates the high-influence source region. On GSM8K, influence Gini rises from 0.398 at  $T = 0.0$  to 0.428 at  $T = 0.3$ , 0.656 at  $T = 0.6$ , and 0.722 at  $T = 0.9$ . The dominant source region also moves from the EOS side toward the sequence-start side, indicating a restructuring of the commit frontier rather than a simple strengthening of the same sources.

For LLaDA, higher temperature more often diffuses influence or preserves the existing structure. The effect is most visible on code tasks, where influence Gini drops by 0.612 on MBPP and 0.281 on HumanEval, while changes on GSM8K, MATH500, and MMLU are much smaller. Thus, temperature is not a generic “more randomness means more diffuse influence” knob. Its effect depends on how a scheduler converts perturbed token distributions into position commitments.

### 6.2 Knob Effects Form a Frontier-Level Hierarchy

Different knobs affect the commit frontier to different degrees. Table 3 summarizes representative changes in influence Gini. Temperature has the largest effect because it directly perturbs token distributions and can therefore change which positions the scheduler commits early. Generation length has a moderate effect because it changes the available position space and the amount of influence each position can carry. Step count and block size are weaker in the tested settings, mainly changing temporal resolution rather than the underlying commitment order.

This hierarchy is consistent with the frontier–

| Knob comparison                       | $\Delta$ Gini | Relative |
|---------------------------------------|---------------|----------|
| Temp., Dream MATH500                  | +0.449        | +571%    |
| Temp., Dream GSM8K                    | +0.324        | +81%     |
| Temp., LLaDA MBPP                     | -0.612        | -73%     |
| Gen. length, Dream MATH500, 64 to 512 | +0.110        | +16%     |
| Steps, Dream GSM8K, 64 to 256         | +0.030        | +8%      |
| Block size, LLaDA MATH500, 64 to 256  | -0.028        | -11%     |

Table 3: Representative effect-size ranking for decoding knobs. Temperature induces the largest frontier-level changes; generation length has a moderate effect; step count and block size are smaller in the tested settings.

role mechanism in Section 5. A knob has a large structural effect when it changes which token roles are exposed early enough to influence many unresolved positions. It has a smaller effect when it mostly redistributes a similar commitment order over more or fewer refinement steps.

### 6.3 Source Regions Shift Structurally Rather Than Randomly

Frontier relocation is also structured. Exact top-source overlap can be low when high-influence positions move, but this does not imply that the new sources are random. Since nearby positions often share functional roles, we also compute windowed overlap by treating positions within a  $\pm 5$  window as regional matches. Between  $T = 0.0$  and  $T = 0.9$ , exact top-5 overlap ranges from 0% to 80% across model-task settings; in low-overlap cases, windowed overlap rises to 13–40%, suggesting regional shifts rather than arbitrary jumps.

Dream provides the clearest case of full frontier relocation: high-influence positions often migrate from the EOS side to the sequence-start side. LLaDA usually shows higher overlap, consistent with temperature changing influence concentration more than fully relocating the frontier. These results suggest that decoding knobs reshape trajectories in interpretable ways: they can move the frontier, change its concentration, or preserve its region while altering influence strength.

### 6.4 Summary

Decoding knobs act through the commit frontier. Temperature has the strongest and most model-dependent effect, generation length has a moderate effect, and step count or block size mainly changes temporal granularity in the tested settings. Overall, decoding behavior is better explained by how

schedulers expose high-influence commitments than by treating knobs as universal controls.

## 7 Discussion: Lightweight Control as Validation

Finally, we ask whether commit-frontier analysis can provide actionable guidance. We use lightweight interventions only as validation cases, not as a new universal decoding algorithm. The key idea is simple: if high-influence frontier commitments reveal a model-specific failure mode, then a small targeted correction should help most in the settings where that failure mode is diagnosed.

This is what we observe. For Dream, the EOS-heavy early frontier in low-temperature math settings suggests a premature termination failure, so we apply an EOS-margin correction that replaces uncertain EOS commitments with the runner-up token. For LLaDA, where the same EOS trap is not observed, we instead use a token-agnostic margin deferral rule for uncertain commitments. Both rules use only the current token distribution and add no extra forward passes. Dream’s correction improves GSM8K by 18.4 points and MATH500 by 26.2 points at  $T = 0.0$ , while LLaDA’s deferral improves GSM8K by 6.0 points at  $T = 0.9$ .

These results support the diagnostic value of the framework: the same commit-frontier analysis leads to different controls for different schedulers. Detailed intervention rules and results are provided in Appendix A.

## 8 Conclusion

We presented a trajectory-level analysis of discrete diffusion language model decoding, showing that schedulers expose sparse high-influence commitment events along model-specific commit frontiers. Using observed downstream entropy reduction, we found that these commitments are early relative to each model’s frontier, stable under tested variations, and shaped by the interaction between commit timing and token role. This commit-frontier view explains why temperature, generation length, step budget, and block size reshape decoding trajectories in model-dependent ways rather than acting as universal controls. Lightweight interventions further validate that the analysis can diagnose model-specific failure modes without additional forward passes, suggesting that discrete diffusion decoding is best understood as trajectory control over scheduler-dependent critical commitments.

## Limitations

This work analyzes discrete diffusion decoding through observed trajectory influence, which is descriptive rather than causal. Since multiple positions may be committed at the same decoding step, the entropy reduction after a commitment event cannot be uniquely attributed to a single position without additional counterfactual interventions. Our measure is therefore intended as a diagnostic signal for realized trajectories, not as an estimate of isolated causal effects.

Our experiments cover two representative instruction-tuned dLMs, Dream and LLaDA, across reasoning, code, and question-answering tasks. Although these models use different schedulers and exhibit distinct frontier structures, broader validation on additional architectures, schedulers, languages, and generation settings is needed. In addition, our token-role analysis relies on coarse role families such as EOS, semantic content, whitespace, and delimiters; more fine-grained linguistic or program-structural categories may reveal additional patterns.

Finally, the lightweight interventions are used only as validation cases for the commit-frontier analysis. They are manually designed, model-specific, and not intended to be universal decoding algorithms. Future work could learn frontier-aware schedulers directly, incorporate richer uncertainty signals, and test whether such methods can improve generation quality while preserving the efficiency advantages of discrete diffusion decoding.

## 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

- Marianne Arriola, Subham Sekhar Sahoo, Aaron Gokaslan, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Justin T Chiu, and Volodymyr Kuleshov. 2025. [Block diffusion: Interpolating between autoregressive and diffusion language models](#). In *The Thirteenth International Conference on Learning Representations*.
- 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.
- Parikshit Bansal and Sujay Sanghavi. 2026. [Enabling approximate joint sampling in diffusion lms](#). *Preprint*, arXiv:2509.22738.
- Kecheng Chen, Ziru Liu, Xijia Tao, Hui Liu, Xinyu Fu, Suiyun Zhang, Dandan Tu, Lingpeng Kong, Rui Liu, and Haoliang Li. 2025a. [Beyond confidence: Adaptive and coherent decoding for diffusion language models](#). *Preprint*, arXiv:2512.02044.
- 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.
- Sitan Chen, Kevin Cong, and Jerry Li. 2025b. [Optimal inference schedules for masked diffusion models](#). *Preprint*, arXiv:2511.04647.
- 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.
- Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. 2019. [Mask-predict: Parallel decoding of conditional masked language models](#). In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing and the 9th International Joint Conference on Natural Language Processing (EMNLP-IJCNLP)*, pages 6112–6121, Hong Kong, China. Association for Computational Linguistics.
- Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, Hao Peng, and Lingpeng Kong. 2025. [Scaling diffusion language models via adaptation from autoregressive models](#). In *The Thirteenth*

- International Conference on Learning Representations*.
- Jiatao Gu, Changhan Wang, and Jake Zhao Junbo. 2019. *Levenshtein transformer*. Curran Associates Inc., Red Hook, NY, USA.
- Dan Hendrycks, Collin Burns, Steven Basart, Andy Zou, Mantas Mazeika, Dawn Song, and Jacob Steinhardt. 2021a. *Measuring massive multitask language understanding*. In *International Conference on Learning Representations*.
- Dan Hendrycks, Collin Burns, Saurav Kadavath, Akul Arora, Steven Basart, Eric Tang, Dawn Song, and Jacob Steinhardt. 2021b. *Measuring mathematical problem solving with the MATH dataset*. In *Thirty-fifth Conference on Neural Information Processing Systems Datasets and Benchmarks Track (Round 2)*.
- Emiel Hooeboom, Alexey A. Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. 2022. *Autoregressive diffusion models*. In *International Conference on Learning Representations*.
- Zachary Horvitz, Raghav Singhal, Hao Zou, Carles Domingo-Enrich, Zhou Yu, Rajesh Ranganath, and Kathleen McKeown. 2025. *No compute left behind: Rethinking reasoning and sampling with masked diffusion models*. *Preprint*, arXiv:2510.19990.
- Pang Wei Koh and Percy Liang. 2017. *Understanding black-box predictions via influence functions*. In *Proceedings of the 34th International Conference on Machine Learning*, volume 70 of *Proceedings of Machine Learning Research*, pages 1885–1894. PMLR.
- Junzhuo Li, Bo Wang, Xiuze Zhou, Peijie Jiang, Jia Liu, and Xuming Hu. 2025. *Decoding knowledge attribution in mixture-of-experts: A framework of basic-refinement collaboration and efficiency analysis*. In *Proceedings of the 63rd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 22431–22446, Vienna, Austria. Association for Computational Linguistics.
- Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Yi Liang, Soroush Vosoughi, and Shiwei Liu. 2026. *Diffusion language model knows the answer before it decodes*. In *The Fourteenth International Conference on Learning Representations*.
- Kevin Meng, David Bau, Alex J Andonian, and Yonatan Belinkov. 2022. *Locating and editing factual associations in GPT*. In *Advances in Neural Information Processing Systems*.
- Amr Mohamed, Yang Zhang, Michalis Vazirgiannis, and Guokan Shang. 2025. *Fast-decoding diffusion language models via progress-aware confidence schedules*. *Preprint*, arXiv:2512.02892.
- 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*.
- 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*.
- Mitchell Stern, William Chan, Jamie Kiros, and Jakob Uszkoreit. 2019. *Insertion transformer: Flexible sequence generation via insertion operations*. In *Proceedings of the 36th International Conference on Machine Learning*, volume 97 of *Proceedings of Machine Learning Research*, pages 5976–5985. PMLR.
- Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. 2026. *Remasking discrete diffusion models with inference-time scaling*. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*.
- Xinwei Wu, Junzhuo Li, Minghui Xu, Weiling Dong, Shuangzhi Wu, Chao Bian, and Deyi Xiong. 2023. *DEPN: Detecting and editing privacy neurons in pre-trained language models*. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, pages 2875–2886, Singapore. Association for Computational Linguistics.
- 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.
- 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.
- Kevin Zhai, Sabbir Mollah, Zhenyi Wang, and Mubarak Shah. 2026. *Core: Context-robust re-masking for diffusion language models*. *Preprint*, arXiv:2602.04096.
- 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, Tiejuan 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.

| Model | Diagnosed pattern                                                        | Lightweight rule                                                           | Main effect                                          |
|-------|--------------------------------------------------------------------------|----------------------------------------------------------------------------|------------------------------------------------------|
| Dream | EOS-heavy early frontier; premature EOS under low temperature            | If EOS is selected with a small margin, commit the runner-up token instead | +18.4pp on GSM8K and +26.2pp on MATH500 at $T = 0.0$ |
| LLaDA | No EOS trap; high temperature produces unreliable low-margin commitments | Defer low-margin positions regardless of token type                        | +6.0pp on GSM8K at $T = 0.9$                         |

Table 4: Commit-frontier diagnosis suggests different lightweight interventions for different scheduler profiles.

| Setting          | Baseline | + Deferral | $\Delta$ | Best $\tau_{\text{def}}$ |
|------------------|----------|------------|----------|--------------------------|
| GSM8K, $T = 0.0$ | 74.0%    | 75.0%      | +1.0pp   | 0.05–0.10                |
| GSM8K, $T = 0.9$ | 73.0%    | 79.0%      | +6.0pp   | 0.20                     |

Table 5: LLaDA margin-based deferral. The intervention helps most at high temperature, where confidence estimates are more strongly perturbed.

| Setting            | Baseline | + EOS correction | $\Delta$ |
|--------------------|----------|------------------|----------|
| GSM8K, $T = 0.0$   | 25.9%    | 44.3%            | +18.4pp  |
| MATH500, $T = 0.0$ | 3.6%     | 29.8%            | +26.2pp  |
| GSM8K, $T = 0.6$   | 46.4%    | 47.9%            | +1.5pp   |
| GSM8K, $T = 0.9$   | 46.3%    | 54.8%            | +8.5pp   |

Table 6: Dream EOS-margin correction. The largest gains occur at  $T = 0.0$ , where commit-frontier analysis diagnoses an EOS-dominated early frontier and premature termination failure.

## A Lightweight Intervention Details

This appendix provides the detailed rules and results for the lightweight interventions discussed in Section 7. These interventions are used only as validation cases for commit-frontier analysis. They are not intended to be universal decoding algorithms.

### A.1 Intervention Principle

The interventions follow a simple diagnosis-to-control principle. First, we use observed trajectory influence to identify which positions dominate uncertainty reduction under a given scheduler. Second, we check whether these positions correspond to a systematic failure mode under specific decoding settings. Third, we intervene only when the model’s local decision is uncertain, using the top-1/top-2 probability margin as an online signal.

This principle leads to different controls for different models. Dream exhibits an EOS-heavy early frontier in low-temperature math settings, suggesting a premature boundary-commitment failure. LLaDA does not show the same EOS trap; instead, its confidence-based scheduler can become unreliable under high temperature. We therefore use an EOS-specific correction for Dream and a token-agnostic deferral rule for LLaDA.

### A.2 Dream: EOS-Margin Correction

Dream’s entropy-based scheduler commits low-entropy positions. In low-temperature math settings, the model can assign high confidence to EOS-region positions before resolving content tokens. This produces a premature termination failure mode, especially on MATH500 at  $T = 0.0$ ,

where the baseline accuracy is only 3.6%.

We apply the correction after the scheduler selects committed positions but before tokens are written into the canvas. Let  $p_1(y | p)$  and  $p_2(y | p)$  denote the top-1 and top-2 token probabilities at position  $p$ , and let

$$m(p) = p_1(y | p) - p_2(y | p)$$

be the local decision margin. If the selected token is EOS and  $m(p) < \tau_{\text{eos}}$ , we write the runner-up token instead of EOS. Otherwise, we keep the original selected token. The rule does not require any additional forward pass and changes only locally ambiguous EOS decisions.

The results match the diagnosis. The correction is most effective in the low-temperature settings where Dream’s early frontier is EOS-heavy. The gain is smaller once the frontier has shifted away from the EOS region, suggesting that the rule is not simply a generic EOS penalty but is most useful when the commit-frontier analysis predicts an EOS-dominated failure mode.

### A.3 LLaDA: Margin-Based Deferral

LLaDA uses a confidence-based scheduler and does not exhibit the same low-temperature EOS trap. At  $T = 0.0$ , it already achieves strong GSM8K performance. However, under high temperature, token probabilities are perturbed and confidence estimates can become less reliable. The scheduler may then commit positions whose top alternatives remain close.

We therefore use a token-agnostic margin-based deferral rule. For each candidate position selected by the scheduler, we compute the top-1/top-2 probability margin  $m(p)$ . If  $m(p) < \tau_{\text{def}}$ , the position remains masked and is deferred to a later step. Otherwise, the top-1 token is committed. Unlike the Dream intervention, this rule does not target EOS or any specific token category. It simply delays uncertain commitments so they can be resolved with more context.

The effect is smaller than Dream’s EOS correction, as expected. Dream’s rule addresses a binary empty-output failure, while LLaDA’s rule refines an already functional scheduler. The larger gain at  $T = 0.9$  supports the diagnosis that high temperature makes low-margin commitments less reliable.

We omit code-task rows from the main result tables because those measurements only showed that margin deferral did not substantially change influence Gini on HumanEval and MBPP, without providing a comparable accuracy improvement. This negative result is still informative: LLaDA’s code behavior may require more targeted controls over syntax-bearing tokens such as delimiters and whitespace, rather than a token-agnostic margin rule.

#### A.4 Takeaway

The two interventions validate the explanatory role of commit-frontier analysis. The same diagnostic framework leads to different corrective rules for different schedulers: Dream benefits from an EOS-specific correction, whereas LLaDA benefits from uncertainty-based deferral under high temperature. These results suggest that future decoding methods could explicitly account for high-influence frontier commitments, avoid premature boundary decisions, and stabilize rare syntax-bearing tokens when they appear early in the trajectory.