Supportive Token Revealing for Fast Diffusion Language Model Decoding

Giries Abu Ayoub[1], Mario Barbara[1], Lluís Pastor-Pérez[2], Tanja Bien[2], Aneesh Barthakur[2], Alaa Maalouf[1], Loay Mualem[2,3]*

[1]Department of Computer Science, University of Haifa
[2]Institute for AI, University of Stuttgart [3]IMPRS-IS
Corresponding authors: jerryabuayob@gmail.com and loaymua@gmail.com

Abstract

Discrete diffusion language models can generate text efficiently by updating multiple masked positions in parallel, but this parallelism introduces a quality-latency trade-off. Aggressive decoding may commit mutually dependent tokens too early, while conservative decoding requires many denoising steps. Existing methods address this tension by deciding which tokens are safe to reveal using confidence or dependency criteria. However, avoiding unsafe commits does not necessarily make the remaining masked sequence easy to decode, since uncertain tokens may depend on masked tokens, creating a bottleneck for denoising steps. We propose AXON, a training-free module that can be added on top of existing parallel decoding strategies for diffusion language models. Rather than replacing the base decoder, AXON monitors the remaining uncertain masked tokens and intervenes only when their current state suggests that additional context is needed. It then shifts the criterion from which tokens are safest to reveal to which confident reveals would best support later denoising. AXON selects anchors, confident masked tokens that uncertain positions attend to, using attention, uncertainty, and confidence signals. Experiments on reasoning and code-generation benchmarks across multiple diffusion language models show that AXON improves the quality-latency trade-off of existing parallel decoders, often reducing the number of function evaluations while maintaining or improving accuracy.

https://github.com/Jerryaa98/AXON.git

1 Introduction

Autoregressive language models (Radford et al., 2019; Tom et al., 2020) generate text one token at a time, which makes their inference inherently sequential. Diffusion large language models (dLLMs) (Shi et al., 2024; Sahoo et al., 2024; Li et al., 2022; Gong et al., 2022; Lou et al., 2023) offer a different generation paradigm: they start from a masked sequence and iteratively denoise it, predicting all masked positions at each step. This bidirectional refinement process makes dLLMs attractive for low-latency text generation since multiple positions can be updated in parallel. Recent dLLMs such as LLaDA (Nie et al., 2026) and Dream (Ye et al.) demonstrate that this paradigm can produce competitive text generations while exposing a natural source of parallelism that is absent in autoregressive left-to-right decoding. However, unmasking arbitrary tokens simultaneously can produce inconsistent configurations as tokens may be coupled through syntax, semantics, arithmetic constraints, or multi-token entities. Existing training-free parallel-decoding methods mainly address this issue by deciding which tokens can be safely committed together. Confidence-based methods (Wu et al., 2026; Ben-Hamu et al., 2026; Kim et al., 2026b; Shu et al., 2026) reveal tokens whose marginal probabilities are sufficiently high, implicitly assuming that high confidence tokens are both reliable and uncoupled/mutually independent. More recent dependency-aware methods (Luo et al., 2026; Ringel et al., 2026; Zhou et al., 2026; Kim et al., 2026a) use attention or distributional similarity to estimate interactions between masked positions and to avoid simultaneously committing strongly coupled tokens. However, even when unsafe commits are avoided, the remaining masked positions may still lack sufficient visible context for reliable parallel decoding. This creates a bottleneck where decoding becomes increasingly conservative or stalls altogether.

In this work, we take a complementary view of fast dLLM decoding. Instead of only asking which tokens are safe to reveal immediately, we ask which masked tokens would provide the most useful context for the remaining uncertain positions if revealed early. Some tokens are already easy to

*Corresponding author

Proceedings of the 15th International Joint Conference on Natural Language Processing and the 5th Conference of the Asia-Pacific Chapter of the Association for Computational Linguistics, pages 683–704
November 6-10, 2026 ©2026 Association for Computational Linguistics

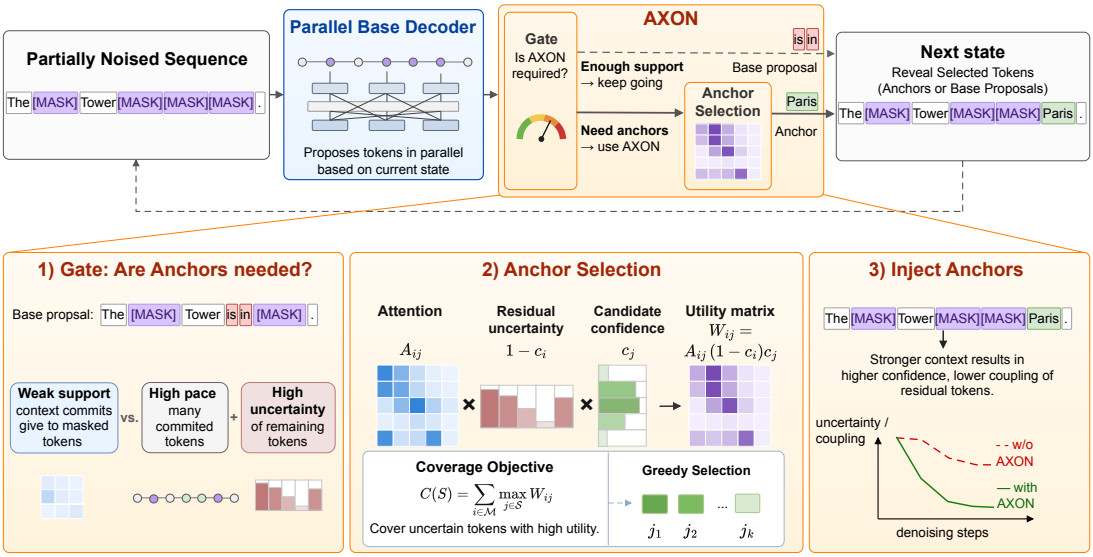


Figure 1: Overview of AXON. A fast parallel decoder proposes tokens from the current masked state. AXON uses a lightweight gate to detect insufficient contextual support, selects influential anchors using attention, uncertainty, and confidence signals, and reveals them as context for the next denoising step.

commit because they are confident and weakly coupled to others. Other tokens are influential because many uncertain positions attend to them or depend on them. Revealing such tokens can improve contextual support for later denoising steps and reduce decoding bottlenecks.

Motivated by this view, we propose AXON[1], a training-free module that can be added on top of existing parallel decoding strategies. AXON treats influential-token revealing as a small set-selection problem. It scores candidate tokens using attention, uncertainty, and confidence signals, and selects a small non-redundant subset through a coverage-based objective. The coverage view naturally leads to a submodular function objective, since once an uncertain position is already supported by one selected token (i.e. covered), another token supporting the same position should contribute less additional value. Since this selection is not needed at every step, AXON uses a lightweight gate that activates it only when the current masked tokens appears under-supported. Figure 1 provides an overview of this process.

We evaluate AXON on reasoning and code-generation benchmarks across multiple diffusion language models and parallel decoders. The results show that AXON improves the quality-latency trade-off over strong baselines. Explicitly, our contributions are as follows :

- We introduce a complementary view of fast dLLM decoding: beyond selecting tokens that are safe to commit, a decoder can reveal informative masked tokens that provide useful context for the remaining positions.
- We propose AXON, a training-free module for existing parallel dLLM decoders. AXON intervenes only when the current masked sequence appears under-supported. It first applies a gate to detect such states, then identifies confident masked tokens that are strongly attended to by uncertain positions, and finally selects a non-redundant subset of these anchors through a coverage-based objective. Revealing the selected anchors provides targeted context to the remaining masked positions without modifying the underlying model.
- We evaluate AXON across multiple backbones, reasoning and code-generation benchmarks, and several families of parallel decoders, including confidence-based, locality-aware, and dependency-aware strategies. Across these settings, AXON improves the quality-latency trade-off while keeping the backbone frozen.

2 Related Work

Fast dLLM inference methods primarily differ in how masked positions are selected and revealed during denoising. In the sequel, we first review dis-

[1]We call our method AXON by analogy to axons in neural systems: it reveals a few high-impact tokens that transmit useful context to uncertain positions during denoising.

crete dLLMs, followed by training-free decoding methods for frozen backbones and dependency-aware parallel decoding methods.

Discrete dLLMs. Extending denoising diffusion to discrete spaces, discrete dLLMs parameterize the reverse process through per-token marginal predictions (Shi et al., 2024; Hoogeboom et al., 2021; Austin et al., 2021a; Sahoo et al., 2024; Müller et al., 2026). Recent large-scale masked dLLMs like LLaDA (Nie et al., 2026; Bie et al., 2025, 2026) and Dream (Ye et al.) offer a competitive alternative to autoregressive generation.

Fast decoding and commit expansion for frozen dLLMs. A way to accelerate decoding is to predict multiple tokens per model call. In autoregressive models, speculative decoding and related verification-based methods propose multiple future tokens and validate them in parallel with the target model (Leviathan et al., 2023). For dLLMs, the challenge is to decide which masked positions can be reliably committed together. Early approaches make this choice using local uncertainty measures such as entropy, margin, or confidence (Ghazvininejad et al., 2019). Fast-dLLM (Wu et al., 2026) combines block-wise Key-Value caching with a rule that commits positions whose marginal confidence exceeds a threshold, using high confidence as a proxy for reliable parallel commitment. EB-Sampler (Ben-Hamu et al., 2026) uses an entropy bound to adaptively choose which and how many tokens to unmask at each step. KLASS (Kim et al., 2026b) requires the token's distribution to be both confident and stable across successive denoising steps. LocalLeap (Kong et al., 2025) expands commits around confident anchors by relaxing nearby positions within a local window when local predictions appear reliable. These methods are lightweight and training-free, and they provide different criteria for expanding or accepting commits in a frozen dLLM.

Dependency-aware parallel decoding. Recent work has shown that aggressive parallel decoding can suffer from joint inconsistency, where tokens that look plausible individually become incompatible when committed together (Wu et al., 2026; Ben-Hamu et al., 2026; Zhang et al., 2026). Dependency-aware methods address this by estimating interactions between masked positions during decoding. APD (Israel et al., 2026) uses an auxiliary autoregressive model to estimate dependencies between tokens committed in parallel. Similarly, DEMASK (Ringel et al., 2026) uses a lightweight pairwise dependency predictor trained on hidden states. Instead of training a model, other methods utilize the already available attention maps to estimate dependencies. Attn-Sampler (Zhou et al., 2026) uses attention column sums as a global importance score for ordering commits. DAPD (Kim et al., 2026a) builds an attention-induced Markov random field and picks per-step commits as a greedy independent set. DAWN (Luo et al., 2026) constructs an attention dependency graph and uses committed anchors to support decoding of related positions, with conflict-aware scheduling that avoids jointly updating strongly coupled positions.

These methods highlight the role of dependency structure in safe parallel commitment, but they primarily use it to decide which positions should or should not be updated together. AXON instead focuses on the complementary problem of anchor selection: identifying confident reveals that are not only reliable, but also informative for the remaining uncertain positions. Additional related work can be found in Appendix A .

3 Background

Diffusion language models generate a sequence by gradually resolving masked positions. Let X denote the prompt and let $y = (y_1, \ldots, y_L)$ denote the response to be generated, where each y_i belongs to a discrete vocabulary $\mathcal{V}$. Decoding starts from a fully masked response, $y^{(0)} = (\texttt{[MASK]}, \ldots, \texttt{[MASK]})$, and proceeds through a sequence of denoising steps. At step t, the current response $y^{(t)}$ contains both revealed tokens and still-masked positions. We denote the set of masked positions by $M^{(t)} = \{i \in \{1, \ldots, L\} : y_i^{(t)} = \texttt{[MASK]}\}$. For every $i \in M^{(t)}$, the model predicts a categorical distribution over the vocabulary, $p_\theta(y_i = v \mid X, y^{(t)}), v \in \mathcal{V}$, and we define its confidence as the probability assigned to the most likely non-mask token. A decoding rule then chooses a subset of masked positions $R^{(t)} \subseteq M^{(t)}$ to reveal. The selected positions are filled with their top-1 predictions and remain visible in the following denoising steps. Thus, the main inference-time decision is how to choose $R^{(t)}$ at each step.

Parallel decoding. The advantage of dLLMs comes from the ability to reveal more than one position per forward pass. Revealing many positions reduces the number of denoising steps, but it can also introduce errors because the model

gives per-position marginal predictions while the selected positions may be interdependent. A parallel decoder effectively relies on the approximation $p_\theta(y_R \mid X, y^{(t)}) \approx \prod_{i \in R} p_\theta(y_i \mid X, y^{(t)})$, for the selected set R. This approximation is reliable when the positions in R are close to conditionally independent under the current context. In practice, masked tokens can depend on one another through syntax, semantics, arithmetic constraints, formatting, or multi-token entities. Thus, tokens that look plausible individually may form an inconsistent joint assignment when revealed together.

Notation. Let X denotes the conditioning prompt. At a given step t, let $y^{(t)}$ the set of tokens already revealed, $B^{(t)}$ denote the set of all token positions in the current decoding block, $M^{(t)}$ be the set of positions that are still masked, and let $U^{(t)} \subseteq M^{(t)}$ be the set of (*uncertain*) positions whose predicted confidence falls below a threshold. Let $S^{(t)}_{\text{base}}$ denote the set of tokens that is proposed by the base decoder, and $S^{\star(t)}$ be the set of (final) tokens returned by AXON. We denote by $c_i^{(t)} \in [0, 1]$ the predicted confidence at position i, by $A_{ij}^{(t)}$ the attention weight that position i assigns to position j, aggregated across all attention layers. Finally $\alpha \in [0, 1]$ denotes the gate hyperparameter.

4 Methodology

Current fast parallel decoders may reveal several positions simultaneously if they are approximately conditionally independent. However, "safe" combinations don't always provide enough context for future steps, leading to states where the remaining positions are still too uncertain to decode safely. AXON targets this limitation by augmenting a given decoder with a two stage strategy. First, AXON computes a lightweight condition to determine if the base decoder needs additional contextual support. Second, if the condition is satisfied (i.e. the gate activates), then AXON exposes one or more *anchors*—which are masked positions that are both confident, and receive strong attention from uncertain positions. By revealing these anchors, AXON exposes the necessary information required to reduce the uncertainty of the masked tokens. Figure 2 illustrates this process.

Anchors. Ideally, an anchor should be a position that reduces the uncertainty of several masked positions once revealed. Given a masked position i, the improvement of revealing position j is ideally measured by the quantity $\Delta_i(j) = H(y_i \mid X, y^{(t)}) - H(y_i \mid X, y^{(t)}, y_j)$, where H denotes the entropy of the predictive distribution and y_i, y_j are the tokens at positions i and j. Optimizing $\Delta_i(j)$ directly is infeasible because the second term requires re-evaluating the model for each candidate anchor position.

Inspired by previous attention-based decoding methods (Luo et al., 2026; Zhou et al., 2026), we circumvent this limitation by using a *lightweight* score based on the attention maps of the dLLM which are already available as a by-product of the denoising step. Explicitly, define

$$w_{ij}^{(t)} = A_{ij}^{(t)} \big(1 - c_i^{(t)}\big) c_j^{(t)}, \tag{1}$$

The confidence term $c_j^{(t)}$ selects for anchors that are confident and reliable, while the uncertainty term $(1 - c_i^{(t)})$ emphasizes uncertain tokens that have the most need for additional context. Finally, the attention score $A_{ij}^{(t)}$ captures the fact that the uncertainty for a masked position i has the most potential for improvement when it is strongly coupled with the anchor j. Indeed, the connection between attention and confidence improvement can be empirically verified (see Figure 7), further supported by the attention-dynamics and coverage analyses (see Appendix C), and made more rigorous for single-layer dLLMs (see Appendix G).

Coverage objective. The anchor score above measures the usefulness of one candidate anchor for one uncertain position. However, when selecting multiple anchors at a denoising step (see Section 5), their value should not be treated independently. Two anchors may both appear useful, but if they support the same uncertain positions, revealing both gives little additional benefit. We therefore select anchors by maximizing a coverage objective under a cardinality constraint:

$$\max_{S \subseteq V^{(t)}, |S| \le k} C^{(t)}(S) = \sum_{i \in U^{(t)}} \max_{j \in S} w_{ij}^{(t)}, \tag{2}$$

where S is the selected anchor set, and k is the anchor budget. Each uncertain position receives credit only from its best selected anchor, encouraging selected anchors to cover different parts of the residual state.

Proposition 1 in Appendix D proves that $C(S)$ (Eq. 2) is *monotone submodular*, giving diminishing returns when anchors cover overlapping uncertain positions. Formally, a set function C is

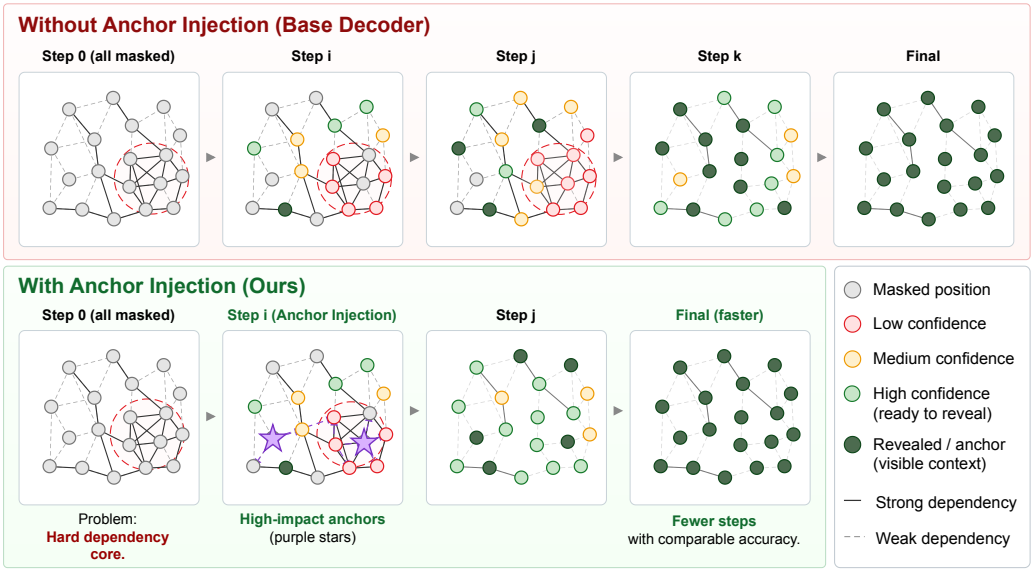


Figure 2: **Effect of anchor injection in AXON.** Without extra context, the base decoder struggles with a hard dependency core (red circle), requiring more denoising steps. When an under-supported state is detected, AXON injects informative anchors (purple stars) as visible context. This breaks down unresolved dependencies, simplifying subsequent predictions and accelerating decoding.

submodular if for every $X \subseteq Y$ and $j \notin Y$,

$$C(X \cup \{j\}) - C(X) \geq C(Y \cup \{j\}) - C(Y).$$

Submodular functions enjoy the property that under a cardinality budget k, simple greedy maximization gives a $(1 - 1/e)$ approximation guarantee (Nemhauser et al., 1978). Overall, $C(S)$ encourages early (greedy) anchors to target the largest pockets of uncertainty, leading to total support rising quickly and then saturating.

4.1 Gating mechanisms

To effectively augment different base decoders with AXON, we develop a mechanism to conditionally activate AXON (i.e. a gate) when the base decoder needs additional support. Our gate is computationally efficient as it depends solely on signals already computed during the forward pass. It detects two common failure modes of base decoders, namely (a) a "pace deficit", and (b) a "context deficit"—which we detail in the immediate sequel.

Pace deficit criterion. It is often desirable to enforce a fixed budget of diffusion steps, say $T \in \mathbb{N}$. At time step t there are $|M^{(t)}|$ tokens that still need to be revealed and only $T - t$ steps remaining. An estimate of the *fraction* of the remaining masked tokens that should be revealed at step t to enforce this budget is, $\rho^{(t)} = \frac{1}{T-t}$. We interpret $\rho^{(t)}$ as the required *pace*. Analogously, the base decoder's pace

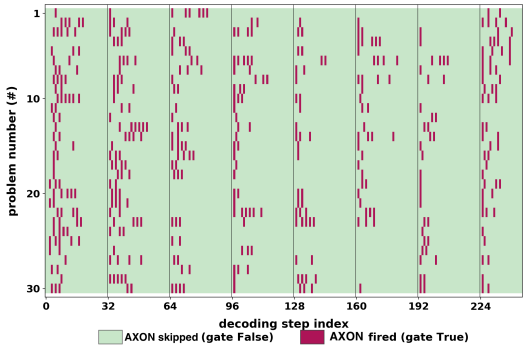


Figure 3: **Per-Step gate activity for LLaDA-1.5 + AXON on GSM8K.** Rows are tasks in the dataset, columns are denoising steps (8 blocks of 32), red means that AXON fired the gate and green means it did not.

is simply the fraction of masked tokens revealed at step t, i.e. $r^{(t)} = \frac{|S^{(t)}_{\text{base}}|}{|M^{(t)}|}$. Thus, to detect a *deficit* in the pace, we compute the following quantity,

$$d^{(t)}_{\text{pace}} = \left(\rho^{(t)} - r^{(t)}\right) \mathbb{I}\left[r^{(t)} \leq r^{(t-1)}\right]. \quad (3)$$

Here $\mathbb{I}[\cdot]$ denotes an indicator function. The term $\rho^{(t)} - r^{(t)}$ is a straightforward realization of the pace deficit, while the condition $r^{(t)} \leq r^{(t-1)}$ allows AXON to ignore the pace deficit when the base decoder's pace is still increasing, i.e. AXON is optimistic that the pace deficit will be solved by the base decoder in subsequent steps.

Context deficit criterion. One of the core goals of AXON is to improve the contextual support pro-

Algorithm 1 AXON decoding step at time t.

Input: Masked $M^{(t)}$, Uncertain $U^{(t)}$, Base $S^{(t)}_{\text{base}}$, Confidences $c^{(t)}$, Attention $A^{(t)}$, Block $B^{(t)}$, Gate weight α, Budget T, Pace at step t $r^{(t)}$, Coverage objective $C^{(t)}$

Output: Committed token set $S^{\star(t)}$

```
# compute decoding paces
r^(t) ← |S^(t)_base| / |M^(t)|
ρ^(t) ← 1/(T − t)
# compute residual confidence and contextual support
c̄^(t)_M ← (1/|U^(t)|) Σ_{i∈U^(t)} c^(t)_i
g^(t) ← (1/|U^(t)|) Σ_{i∈U^(t)} (Σ_{j∈S^(t)_base} A^(t)_ij) / (Σ_{j∈B^(t)} A^(t)_ij + ε)   (Eq. 4)
# calculate deficits
d^(t)_pace ← (ρ^(t) − r^(t)) 𝕀[r^(t) ≤ r^(t−1)]   (Eq. 3)
d^(t)_ctx ← max(r^(t), 1 − c̄^(t)_M) − g^(t)   (Eq. 5)
# evaluate unified gate (Eq. 6)
if α (d^(t)_ctx)_+ + (1 − α) (d^(t)_pace)_+ > 0 then
    # gate open: dynamic submodular selection
    τ^(t) ← C^(t)(S^(t)_base)   (Eq. 2)
    S ← ∅
    while C^(t)(S) < τ^(t) and M^(t) \ S ≠ ∅ do
        # greedily add anchor with max marginal gain
        Δ_j ← C^(t)(S ∪ {j}) − C^(t)(S)
        j* ← arg max_{j∈M^(t)\S} Δ_j
        S ← S ∪ {j*}
    end while
    S^⋆(t) ← S
else
    # gate closed: commit base proposal
    S^⋆(t) ← S^(t)_base
end if
return S^⋆(t)
```

vided by decoded tokens. Given the base decoder's proposal, we can compute a normalized measure of the contextual support as follows,

$$g^{(t)} = \frac{1}{|U^{(t)}|} \sum_{i \in U^{(t)}} \frac{\sum_{j \in S^{(t)}_{\text{base}}} A^{(t)}_{ij}}{\sum_{j \in B^{(t)}} A^{(t)}_{ij} + \varepsilon}. \quad (4)$$

Each term in the outer sum measures the support (in terms of attention) provided by $S^{(t)}_{\text{base}}$ to position i, relative to the total support provided by all the tokens in the block. To detect low contextual support, we compare $g^{(t)}$ to a dynamic threshold to obtain our context deficit criterion, namely,

$$d^{(t)}_{\text{ctx}} = \max\big(r^{(t)}, u^{(t)}_M\big) - g^{(t)}, \quad (5)$$

where $r^{(t)} = \frac{|S^{(t)}_{\text{base}}|}{|M^{(t)}|}$ as before, and $u^{(t)}_M = \frac{1}{|U^{(t)}|} \sum_{i \in U^{(t)}} (1 - c^{(t)}_i)$ is the mean uncertainty among uncertain tokens. The dynamic threshold captures two scenarios where we want to increase the contextual support further (by activating AXON). Firstly, when $r^{(t)}$ is high, it means that the base decoder has committed a large number of positions, and so the contextual support should be proportionately high to avoid committing tokens that provide low support. Secondly, when $u^{(t)}_M$ is high, this means that the remaining masked tokens are highly uncertain, and so again, the contextual support should be increased proportionately. We refer the reader to Section K in the Appendix for an ablation and additional analysis of alternative scalings of these two terms. In particular, one can introduce separate weights or normalization factors for the reveal ratio and uncertainty terms. However, our experiments show that keeping both signals on their natural scale, i.e., using the unweighted form in Eq. (5), gives the best and most stable performance. This shows that the two quantities are already comparable enough for gating purposes.

Hybrid gate. We combine both gates into an α-weighted hybrid gate:

$$G^{(t)}_{\alpha} = \mathbb{I}\Big[\alpha \left(d^{(t)}_{\text{ctx}}\right)_+ + (1-\alpha)\left(d^{(t)}_{\text{pace}}\right)_+ > 0\Big], \quad (6)$$

where $(x)_+ = \max(0, x)$ ensures that a negative deficit (surplus) in one criterion does not offset the other. When the gate is activated, AXON's anchor selection procedure kicks into action; otherwise, AXON commits the base proposal. The effect of changing α, which modulates the emphasis on the pace deficit and context criteria, is studied in Appendix F. Activation patterns for $G^{(t)}_{\alpha}$ as a function of t is visualized in Figure 3. For additional analysis on gate firing patterns see Appendix I.

5 Experiments

We evaluate our framework against standard discrete diffusion benchmarks. We focus on 3 metrics: the quality-latency trade-off and the number of computational steps of our dynamic mechanism.

5.1 Setups

Models and datasets. We evaluate AXON on three dLLM backbones spanning two model families: Dream-v0-Instruct-7B[2] (Ye et al.), LLaDA-1.5[3] (Zhu et al., 2026), and LLaDA-8B-Instruct[4] (Zhu et al., 2026). This selection allows us to assess whether AXON generalizes

[2]https://huggingface.co/Dream-org/Dream-v0-Instruct-7B

[3]https://huggingface.co/GSAI-ML/LLaDA-1.5

[4]https://huggingface.co/GSAI-ML/LLaDA-8B-Instruct

Method	HumanEval			MBPP			GSM8K			Minerva-Math		
	Acc (↑)	TPS(↑)	NFE(↓)	Acc (↑)	TPS(↑)	NFE(↓)	Acc (↑)	TPS(↑)	NFE(↓)	Acc (↑)	TPS(↑)	NFE(↓)
LLaDA-1.5												
Original	43.90	6.04	256.0	38.80	2.39	256.0	80.36	6.70	256.0	33.36	8.35	256.0
Confidence	43.29	15.61	100.6	38.00	14.11	42.5	80.44	22.95	74.8	**33.38**	22.31	95.4
+ AXON	43.90	18.53	93.6	**39.20**	14.39	42.2	**81.50**	22.57	72.1	33.24	22.61	92.7
+ AXON$_{\text{CVR}}$	43.29	19.74	91.0	38.20	14.52	42.1	81.27	23.30	70.3	33.02	23.06	91.1
LocalLeap	41.46	19.83	75.2	38.80	18.50	32.5	80.06	29.46	57.3	32.40	27.64	75.7
+ AXON	**44.51**	20.42	77.6	38.20	16.77	35.3	80.74	26.74	60.2	32.88	25.86	79.7
+ AXON$_{\text{CVR}}$	**44.51**	22.91	73.0	37.60	17.13	34.6	81.12	27.64	58.1	32.34	26.47	78.1
DAWN	42.07	22.96	77.3	37.80	19.37	30.7	80.82	30.55	54.2	31.80	28.63	72.9
+ AXON	42.68	26.96	62.9	37.20	20.97	29.3	81.20	33.02	49.5	32.00	30.64	68.0
+ AXON$_{\text{CVR}}$	39.02	**29.11**	**59.6**	37.20	**21.49**	**28.3**	80.52	**34.41**	**47.6**	31.30	**31.60**	**66.0**
LLaDA-8B-Instruct												
Original	40.24	18.09	256.0	29.40	6.48	256.0	77.94	7.14	256.0	**33.20**	9.51	256.0
Confidence	40.85	58.96	77.9	29.60	23.69	69.8	78.39	23.48	77.7	32.98	25.14	96.9
+ AXON	40.55	65.45	74.6	32.00	23.51	62.2	**79.30**	23.39	74.7	32.76	25.75	92.7
+ AXON$_{\text{CVR}}$	40.55	66.23	73.6	31.40	23.69	61.6	79.15	24.16	72.2	32.30	26.23	91.0
LocalLeap	39.63	71.20	61.4	29.40	29.84	55.0	77.33	30.55	58.9	32.42	31.61	75.6
+ AXON	**42.68**	76.66	63.1	31.40	28.05	51.5	77.48	27.97	62.1	32.62	29.75	79.1
+ AXON$_{\text{CVR}}$	42.06	79.39	60.2	30.20	28.61	50.6	75.89	28.90	59.9	32.16	30.52	77.2
DAWN	40.24	80.26	63.2	30.80	31.19	50.0	77.94	31.95	55.3	32.36	32.83	72.5
+ AXON	39.98	92.00	53.2	**32.40**	31.69	44.4	77.41	34.33	50.6	31.84	35.36	67.0
+ AXON$_{\text{CVR}}$	39.58	**92.22**	**52.4**	30.60	**32.35**	**42.4**	76.57	**35.89**	**48.2**	30.84	**36.57**	**64.7**
Dream-v0-Instruct-7B												
Original	53.66	16.62	256.0	54.20	5.42	256.0	**76.35**	4.85	256.0	37.80	11.50	256.0
Confidence	**57.32**	38.30	90.4	55.20	26.38	62.7	75.20	19.23	62.3	38.24	24.66	114.8
+ AXON	57.30	41.68	75.4	55.40	25.19	56.0	76.12	18.53	61.6	**38.50**	24.04	112.1
+ AXON$_{\text{CVR}}$	57.30	42.90	70.2	55.20	24.98	55.1	75.82	18.44	61.2	37.74	24.20	110.0
LocalLeap	54.27	42.79	76.1	54.60	28.69	44.7	73.24	22.05	54.2	38.10	28.22	97.7
+ AXON	54.90	47.60	65.1	55.60	28.47	41.4	75.06	21.97	51.5	37.90	28.38	91.2
+ AXON$_{\text{CVR}}$	54.27	47.80	70.0	55.00	28.25	**40.6**	74.91	22.04	**51.1**	37.42	28.32	**89.5**
DAWN	54.88	44.77	76.2	55.40	29.76	42.8	73.16	22.20	52.08	38.22	28.65	94.5
+ AXON	55.60	49.70	**64.9**	**55.80**	**30.09**	42.6	74.83	22.59	51.3	38.28	**28.89**	93.0
+ AXON$_{\text{CVR}}$	54.80	**50.10**	68.8	55.00	29.89	41.8	74.53	**22.67**	51.7	37.76	28.86	91.2

Table 1: **Main quantitative evaluations.** We evaluate across selection rules our submodular anchor framework. We report task Accuracy (%), Throughput (TPS), and Number of Function Evaluations (NFE). Green cell background shading next to a AXON value indicates an improvement over the parent baseline decoder (↑ for Acc and TPS, ↓ for NFE). Best values overall per backbone and per dataset metric are in **bold**.

across architectures and training recipes. In all cases, we use the publicly released checkpoints as frozen backbones. We report results on four benchmarks covering reasoning and code generation: GSM8K (Cobbe et al., 2021) and Minerva-Math (Hendrycks et al., 2021) for mathematical problem solving, and HumanEval (Chen et al., 2021) and MBPP (Austin et al., 2021b) for Python program synthesis. All methods are evaluated under identical model weights, data splits, decoding budgets, and stopping criteria, so any observed differences can be attributed to the decoding strategy rather than the evaluation protocol. Additional implementation details can be found in Appendix B. We additionally test generalization to a non-reasoning benchmark and a third backbone in Appendix L.

Baselines. We compare against three recent training-free decoding strategies that cover the dominant design axes in the dLLM literature: confidence-based acceleration (Wu et al., 2026), locality-aware scheduling (LocalLeap) (Kong et al., 2025), and dependency-aware parallel commitment (DAWN) (Luo et al., 2026). LocalLeap selects anchors by confidence and commits nearby positions within a local window, while DAWN schedules parallel commitments based on inter-token dependencies. These baselines are well aligned with our setting because they are training-free and operate through explicit anchor or commitment selection, allowing AXON to be layered on top without modifying the underlying decoder.

Metrics. We report task accuracy, tokens per second (TPS), and number of function evaluations (NFE). Accuracy follows the standard evaluation protocol of each benchmark. TPS is the number of generated tokens divided by wall-clock decoding time, and NFE is the number of model forward

passes required to complete generation. A desirable decoder improves accuracy or TPS while reducing NFE, or improves one metric without a comparable drop in the others.

Number of Anchors. AXON is intended to provide minimal contextual support rather than to replace the base decoder's commit rule. We therefore evaluate a conservative default variant with a fixed budget of $k = 1$ anchor per gated intervention. We also report AXON_{CVR}, an adaptive variant that greedily adds anchors until their coverage matches the contextual support induced by the base decoder's proposal, i.e., until $C^{(t)}(S_m) \geq C^{(t)}(S^{(t)}_{\text{base}})$. See Appendix J for a study of the effect of the anchor budget.

5.2 Results

Table 1 summarizes accuracy and latency across all baselines, comparing each base decoder with the fixed-budget AXON variant and the adaptive AXON_{CVR} variant. Across models, tasks, and decoders, AXON improves accuracy or TPS in nearly every setting while consistently reducing NFE.

Notably, several settings show simultaneous gains in accuracy and throughput. On LLaDA-8B-Instruct with LocalLeap, AXON improves HumanEval accuracy by 3.05 points while increasing TPS, and AXON_{CVR} further improves throughput while remaining 2.43 accuracy points above the baseline. Similar trends appear on LLaDA-1.5. On mathematical reasoning, AXON improves both accuracy and TPS on GSM8K and Minerva-Math with LLaDA-1.5 and DAWN, while reducing NFE. On Dream-v0-Instruct-7B, DAWN+AXON improves GSM8K accuracy by 1.67 points with higher TPS, and improves both accuracy and throughput on MBPP with lower NFE. Additional per-decoder analysis is provided in Section H.

5.3 Ablations

A natural alternative to our objective (2) is additive coverage, $\text{GC}(S) = \frac{1}{|U^{(t)}|} \sum_{i \in U^{(t)}} \sum_{j \in S} w_{ij}$, which is also a submodular function. The key difference is that objective (2) (also known as *facility-location*) assigns credit from each uncertain position only to its best selected anchor, while additive coverage rewards all anchors which have positive weight. As a result, optimizing $\text{GC}(S)$ can repeatedly select anchors that support the same uncertain positions, whereas facility-location naturally favors diverse and non-redundant reveals through diminishing returns. Table 2 compares the two objectives. Facility-location consistently achieves lower NFE while maintaining comparable accuracy, indicating that explicitly encouraging diverse anchor coverage improves decoding efficiency. In Appendix E, We further tested whether adding an explicit redundancy penalty improves selection quality. In practice, the effect was negligible, suggesting that facility-location already discourages redundant anchors sufficiently through its saturating coverage structure. We therefore use the monotone facility-location objective throughout the paper. In addition to the anchor selection objective, the performance of AXON relies on the unified gate weight α, which balances the framework's sensitivity between context and pace deficits. Based on our evaluations, we configure AXON with $\alpha = 0.2$ for the LLaDA and $\alpha = 0.9$ for Dream-Instruct across all main-text experiments; a comprehensive ablation of these gating dynamics and their impact on throughput and accuracy is provided in Appendix F.

		$C(S)$			$\text{GC}(S)$		
Model	**Task**	Acc	TPS	NFE	Acc	TPS	NFE
Dream-7B	MBPP	55.00	29.76	**41.81**	54.60	29.98	42.25
LLaDA-1.5	MBPP	35.20	21.23	**28.30**	35.00	20.69	29.33
LLaDA-8B	MBPP	30.40	32.01	**42.40**	29.80	31.69	43.46
LLaDA-1.5	HumanEval	39.02	28.73	**59.49**	41.46	28.41	60.49
LLaDA-8B	HumanEval	39.98	90.64	**52.43**	39.98	88.27	54.22
Mean across cells		*33.14*	***40.47***	***44.89***	***33.39***	*39.81*	*45.95*

Table 2: **Ablating coverage form.** Results comparing our objective, monotone facility-location ($C(S)$) against the graph-cut (GC) across tasks and models.

6 Conclusion and Future Work

We introduced AXON, a training-free module for improving parallel decoding in diffusion language models. AXON monitors the decoding state and intervenes only when the current revealed tokens provide insufficient support to the remaining masked positions. When needed, it reveals a small set of influential and non-redundant anchors that provide targeted context for subsequent denoising steps.

Across reasoning and code-generation benchmarks, multiple dLLM backbones, and different decoding strategies, AXON improves the quality-latency trade-off without changing the backbone or base decoder. Our ablations show that state monitoring and coverage-based anchor selection matter, suggesting that efficient decoding should go beyond commit rules and construct useful context.

More broadly, our work sheds light on the im-

portance of state monitoring and anchor selection in future parallel decoding, whether through gating mechanisms or self-monitoring strategies detecting when contextual support is needed.

7 Limitations

AXON improves existing parallel decoders through gated anchor interventions, but it has several limitations. First, it relies on attention and confidence signals from the frozen dLLM. These signals are lightweight and available during decoding, but they are only proxies for token influence and uncertainty. When confidence is poorly calibrated or attention does not reflect useful dependencies, anchor selection may provide limited benefit.

Second, AXON adds scoring and selection overhead. In our setting this cost is controlled because the gate fires only when needed and anchor selection is performed over a limited candidate set. For longer sequences or larger candidate pools, more approximate selection may be required.

Finally, AXON still has a quality-latency tradeoff through the gate, anchor budget, and base decoder configuration. Our experiments show favorable operating points across multiple models, benchmarks, and decoder families, but the best configuration may vary by task and decoding budget. We focus on text-only reasoning and code-generation tasks; long-context, dialogue, multilingual, and multimodal settings remain future work.

8 AI Use Statement

In this work, we used generative AI tools for text polishing and for writing code. Additionally, we used generative AI tools for part of the ideation and to assist with results analysis.

We have reviewed all AI-assisted work. AI-generated code was verified and stress tested by three of the authors. AI-polished text was read and edited by an author to confirm that it preserves the intended technical meaning and introduces no claim beyond what our experiments support.

Acknowledgments and Disclosure of Funding

The authors thank the Stuttgart Center for Simulation Science (SimTech) and the International Max Planck Research School for Intelligent Systems (IMPRS-IS) for supporting Lluís, Aneesh, Tanja, and Loay. This research was funded by the Ministry of Science, Research and the Arts Baden-Wuerttemberg through the Artificial Intelligence Software Academy (AISA). We also acknowledge the support of the Stuttgart Center for Simulation Science (SimTech). Tanja was supported by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2120/1 – 390831618. Aneesh was supported and funded by the Deutsche Forschungsgemeinschaft (DFG, German Research Foundation) under Germany's Excellence Strategy – EXC 2075 – 390740016. L. Mualem was supported by a postdoctoral scholarship from the Planning and Budgeting Committee (PBC) of the Council for Higher Education in Israel. L. Mualem gratefully acknowledge the computing time provided on the high-performance computer HoreKa by the National High-Performance Computing Center at KIT (NHR@KIT). This center is jointly supported by the Federal Ministry of Education and Research and the Ministry of Science, Research and the Arts of Baden-Württemberg, as part of the National High-Performance Computing (NHR) joint funding program (https://www.nhr-verein.de/en/our-partners). HoreKa is partly funded by the German Research Foundation (DFG). A. Maalouf acknowledges support from the Neubauer Family Foundation and from the MAOF Fellowship of the Council for Higher Education.

2

References

Sudhanshu Agrawal, Risheek Garrepalli, Raghavv Goel, Mingu Lee, Christopher Lott, and Fatih Porikli. 2025. Spiffy: Multiplying diffusion llm acceleration via lossless speculative decoding. *arXiv e-prints*, pages arXiv–2509.

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. 2021a. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993.

Jacob Austin, Augustus Odena, Maxwell Nye, Maarten Bosma, Henryk Michalewski, David Dohan, Ellen Jiang, Carrie Cai, Michael Terry, Quoc Le, and 1 others. 2021b. Program synthesis with large language models. *arXiv preprint arXiv:2108.07732*.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. 2026. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *Advances in Neural Information Processing Systems*, 38:55981–56007.

Tiwei Bie, Maosong Cao, Xiang Cao, Bingsen Chen, Fuyuan Chen, Kun Chen, Lun Du, Daozhuo Feng, Haibo Feng, Mingliang Gong, and 1 others. 2026. Llada2. 1: Speeding up text diffusion via token editing. *arXiv preprint arXiv:2602.08676*.

Tiwei Bie, Maosong Cao, Kun Chen, Lun Du, Mingliang Gong, Zhuochen Gong, Yanmei Gu, Jiaqi Hu, Zenan Huang, Zhenzhong Lan, and 1 others. 2025. Llada2. 0: Scaling up diffusion language models to 100b. *arXiv preprint arXiv:2512.15745*.

Mark Chen, Jerry Tworek, Heewoo Jun, Qiming Yuan, Henrique Ponde De Oliveira Pinto, Jared Kaplan, Harri Edwards, Yuri Burda, Nicholas Joseph, Greg Brockman, and 1 others. 2021. Evaluating large language models trained on code. *arXiv preprint arXiv:2107.03374*.

Yixin Chen, Ankur Nath, Chunli Peng, and Alan Kuhnle. 2024. Guided combinatorial algorithms for submodular maximization. *arXiv preprint arXiv:2405.05202*.

Yuxin Chen and Andreas Krause. 2013. Near-optimal batch mode active learning and adaptive submodular optimization. In *International Conference on Machine Learning*, pages 160–168. PMLR.

Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, and 1 others. 2026. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 22058–22075.

Karl Cobbe, Vineet Kosaraju, Mohammad Bavarian, Mark Chen, Heewoo Jun, Lukasz Kaiser, Matthias Plappert, Jerry Tworek, Jacob Hilton, Reiichiro Nakano, and 1 others. 2021. Training verifiers to solve math word problems. *arXiv preprint arXiv:2110.14168*.

Marwa El Halabi, Suraj Srinivas, and Simon Lacoste-Julien. 2022. Data-efficient structured pruning via submodular optimization. *Advances in Neural Information Processing Systems*, 35:36613–36626.

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.

Shansan Gong, Mukai Li, Jiangtao Feng, Zhiyong Wu, and LingPeng Kong. 2022. Diffuseq: Sequence to sequence text generation with diffusion models. *arXiv preprint arXiv:2210.08933*.

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. *arXiv preprint arXiv:2103.03874*.

Feng Hong, Geng Yu, Yushi Ye, Haicheng Huang, Huangjie Zheng, Ya Zhang, Yanfeng Wang, and Jiangchao Yao. 2026. Wide-in, narrow-out: Revokable decoding for efficient and effective dllms. In *International Conference on Learning Representations*, volume 2026, pages 40132–40150.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. 2021. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. 2026. Accelerating diffusion llms via adaptive parallel decoding. *Advances in neural information processing systems*, 38:52870–52888.

Bumjun Kim, Dongjae Jeon, Moongyu Jeon, and Albert No. 2026a. Dependency-aware parallel decoding via attention for diffusion llms. *arXiv e-prints*, pages arXiv–2603.

Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. 2026b. Klass: Kl-guided fast inference in masked diffusion models. *Advances in Neural Information Processing Systems*, 38:92267–92301.

Fanheng Kong, Jingyuan Zhang, Yahui Liu, Zirui Wu, Yu Tian, Guorui Zhou, and 1 others. 2025. Accelerating diffusion llm inference via local determinism propagation. *arXiv preprint arXiv:2510.07081*.

Yaniv Leviathan, Matan Kalman, and Yossi Matias. 2023. Fast inference from transformers via speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286. PMLR.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. 2022. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35:4328–4343.

Stephanie Lin, Jacob Hilton, and Owain Evans. 2022. Truthfulqa: Measuring how models mimic human falsehoods. In *Proceedings of the 60th annual meeting of the association for computational linguistics (volume 1: long papers)*, pages 3214–3252.

Aaron Lou, Chenlin Meng, and Stefano Ermon. 2023. Discrete diffusion modeling by estimating the ratios of the data distribution. *arXiv preprint arXiv:2310.16834*.

Lizhuo Luo, Zhuoran Shi, Jiajun Luo, Zhi Wang, Shen Ren, Wenya Wang, and Tianwei Zhang. 2026. Dawn: Dependency-aware fast inference for diffusion llms. *arXiv preprint arXiv:2602.06953*.

Loay Raed Mualem, Ethan R Elenberg, Moran Feldman, and Amin Karbasi. 2024. Submodular minimax optimization: Finding effective sets. In *International Conference on Artificial Intelligence and Statistics*, pages 1081–1089. PMLR.

Adrian Müller, Antoine Gonon, Zebang Shen, Ya-Ping Hsieh, and Niao He. 2026. Support before frequency in discrete diffusion. *arXiv preprint arXiv:2605.13999*.

George L Nemhauser, Laurence A Wolsey, and Marshall L Fisher. 1978. An analysis of approximations for maximizing submodular set functions—i. *Mathematical programming*, 14(1):265–294.

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. *Advances in Neural Information Processing Systems*, 38:50608–50646.

Yu-Yang Qian, Junda Su, Lanxiang Hu, Peiyuan Zhang, Zhijie Deng, Peng Zhao, and Hao Zhang. 2026. d3llm: Ultra-fast diffusion llm using pseudo-trajectory distillation. *arXiv preprint arXiv:2601.07568*.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, and 1 others. 2019. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9.

Liran Ringel, Ameen Ali, and Yaniv Romano. 2026. Dependency-guided parallel decoding in discrete diffusion language models. *arXiv preprint arXiv:2604.02560*.

Subham S Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. 2024. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. 2024. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37:103131–103167.

Yingte Shu, Yuchuan Tian, Chao Xu, Yunhe Wang, and Hanting Chen. 2026. Deferred commitment decoding for diffusion language models with confidence-aware sliding windows. *arXiv e-prints*, pages arXiv–2601.

Brown Tom, Mann Benjamin, Ryder Nick, Subbiah Melanie, Dhariwal Prafulla, Neelakantan Arvind, Shyam Pranav, Sastry Girish, Askell Amanda, Agarwal Sandhini, and 1 others. 2020. Language models are few-shot learners. *Advances in neural information processing systems*, 33(1):1877–1901.

Murad Tukan, Fares Fares, Yotam Grufinkle, Ido Talmor, Loay Mualem, Vladimir Braverman, and Dan Feldman. 2023. Orbslam3-enhanced autonomous toy drones: Pioneering indoor exploration. *arXiv preprint arXiv:2312.13385*.

Murad Tukan, Loay Mualem, and Moran Feldman. 2024. Practical 0.385-approximation for submodular maximization subject to a cardinality constraint. In *Advances in Neural Information Processing Systems*. NeurIPS 2024.

Xinyun Wang, Min Zhang, Sen Cui, Zhikang Chen, Bo Jiang, Kun Kuang, and Mingbao Lin. 2026. Reversible diffusion decoding for diffusion language models. *arXiv preprint arXiv:2602.00150*.

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 *International Conference on Learning Representations*, volume 2026, pages 4340–4354.

Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2026. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. In *International Conference on Learning Representations*, volume 2026, pages 57027–57051.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models, 2025. *URL https://arxiv. org/abs/2508.15487*.

Shaorong Zhang, Longxuan Yu, Rob Brekelmans, Luhan Tang, Salman Asif, and Greg Ver Steeg. 2026. Generation order and parallel decoding in masked diffusion models: An information-theoretic perspective. *arXiv preprint arXiv:2602.00286*.

Lifeng Zhou and Pratap Tokekar. 2022. Risk-aware submodular optimization for multirobot coordination. *IEEE Transactions on Robotics*, 38(5):3064–3084.

Yuyan Zhou, Kai Syun Hou, Weiyu Chen, and James Kwok. 2026. Attention-based sampler for diffusion language models. *arXiv preprint arXiv:2604.08564*.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. 2026. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11425–11460.

A Additional Related Work

Adaptive and revocable commitment. A related line changes how aggressive commitments are organized or revised over time. WINO (Hong et al., 2026) re-masks suspicious commits to make aggressive drafts less risky, while RDD (Wang et al., 2026) rolls back previous blocks when the current block stalls. DCD (Shu et al., 2026) replaces rigid block boundaries with a confidence-aware sliding window. SlowFast (Wei et al., 2026) alternates between an exploratory stage that identifies stable high-confidence spans and an accelerated stage that decodes them in parallel. Each controls the risk, consequence, or timing of a commit. AXON does not replace these mechanisms. It decides which additional reveal is worth making once the decoder's safety rule has been applied.

Model-side and systems-level acceleration. d3LLM (Qian et al., 2026) accelerates dLLMs by retraining the model so it can expose early-decodable tokens sooner. This places d3LLM on the model-training side of the acceleration landscape, whereas AXON keeps the backbone frozen and modifies only the inference-time commit rule.

An orthogonal line reduces wall-clock cost by changing how the model is evaluated or verified, not which positions are committed. Key-value caching for dLLMs (Wu et al., 2026) avoids repeated computation across denoising steps, and speculative systems such as Spiffy (Agrawal et al., 2025) accepts many drafted reveals in parallel under a lossless distribution-preserving test. This is orthogonal to the commit-selection problem we address: systems methods make individual steps cheaper, while AXON changes the commit mask when the host decoder's own rule leaves the residual under-covered. The two compose, so we treat system acceleration as complementary to anchor selection.

Submodular optimization. Submodular functions model diminishing returns, where the marginal gain of adding a new element decreases as the selected set grows. In our setting, once an uncertain position is already well supported by one selected anchor, adding another anchor that supports the same position should provide less additional value. This naturally motivates a facility-location objective, where each uncertain position receives credit only from its best selected anchor. Beyond our setting, submodular objectives are widely used in machine learning for selecting representative and non-redundant subsets, including data summarization (Chen et al., 2024; Mualem et al., 2024), active learning (Chen and Krause, 2013), feature selection (Tukan et al., 2024), model compression (El Halabi et al., 2022), and sensor placement (Zhou and Tokekar, 2022; Tukan et al., 2023). Greedy optimization of monotone submodular functions further admits the standard $(1 - 1/e)$ approximation guarantee under cardinality constraints (Nemhauser et al., 1978).

B Hardware and Implementation Details.

All experiments were implemented within the DAWN [5] diffusion decoding codebase, reusing its model weights, tokenizer settings, decoding budgets, stopping criteria, and block-sampling protocol. Throughput is measured end-to-end and includes the overhead introduced by AXON's commit-selection step during decoding. Experiments are conducted on a cluster of 8×NVIDIA A100 80GB GPUs. We set $\beta_r = \beta_u = 1$ since all the involved quantities are normalized in the interval $[0, 1]$.

C Attention Dynamics and Coverage Analysis

To better understand the mechanistic impact of AXON on the diffusion decoding process, we probe the internal attention dynamics of the model during generation. Specifically, we analyze how attention patterns evolve across inference steps, how the revelation of structural anchors shifts these patterns, and how AXON accelerates the accumulation of contextual coverage. In Section C.1 evaluates LLaDA-8B Instruct on the HumanEval dataset, comparing the base decoders (Confidence, LocalLeap, and DAWN) with and without our proposed method, AXON. In Section C.2, we evaluate LLaDA-1.5 on the MBPP dataset, comparing the base DAWN decoder against DAWN equipped with AXON. Similarly, Section C.3 evaluates LLaDA-1.5 on the HumanEval dataset, comparing the base DAWN decoder against DAWN with AXON.

C.1 Qualitative Analysis of Token Dynamics

Figure 4 visualizes the step-by-step decoding trajectories to illustrate how AXON optimizes the parallel reveal mechanism compared to standard

[5] https://github.com/lizhuo-luo/DAWN

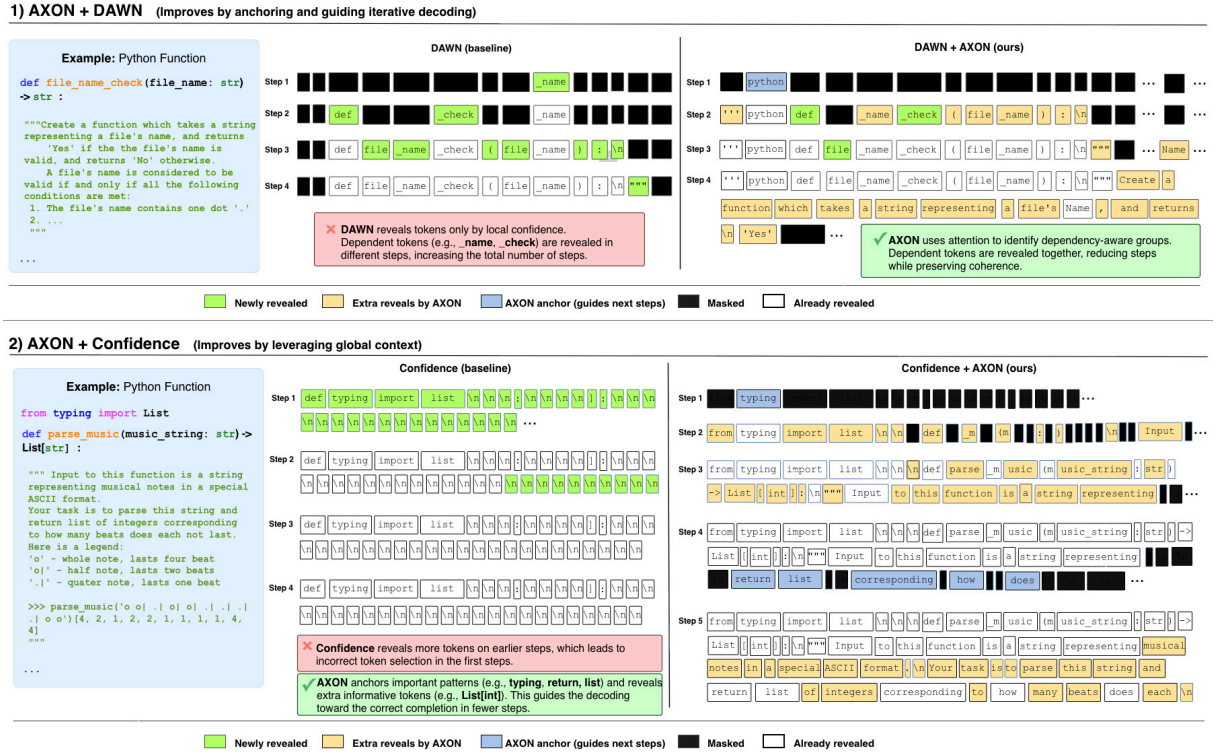


Figure 4: **Comparison of token reveals across decoding methods.** (1) shows the trajectory of DAWN, (2) tracks the Confidence decoder. In both sets, the base decoder sits on the left figure and our anchor-guided AXON variant sits to the right. Green: tokens newly unmasked at the current step. Blue: anchor tokens selected by AXON at step i to guide reveals at steps $> i$. Orange: tokens unmasked because of the anchors selected by AXON. blocks: remaining masked positions. AXON's anchor mechanism unlocks semantically coherent token groups significantly earlier, reducing total NFE while enhancing context coverage.

decoders. In the baseline configurations (left panels), both DAWN and Confidence decoders exhibit fragmented reveals. DAWN gets bottlenecked by local token dependencies, forcing sequential steps to resolve closely coupled syntax like _name and _check. Confidence decoding (bottom left) suffers from early global scattering, unmasking generic syntax elements (\n) prematurely without semantic coordination. In contrast, when using AXON on top of the decoders (right panels), it intercepts these under-supported states. By injecting critical structural anchors (blue tokens like python or typing), AXON leverages global attention paths to safely unmask entire dependent blocks simultaneously (orange tokens). This effectively resolves hard dependency cores blocks ahead of schedule, thus minimizing the total NFE required to achieve coherent code generation.

C.2 Mechanistic Impact of Anchor Commits

When AXON fires, it explicitly selects masked tokens that are highly attended to by the remaining uncertain positions. Figure 5 illustrates the immediate structural impact of revealing these anchors.

We visualize the raw 32×32 attention slice (where rows indicate the attending position and columns indicate the attended position) just before and immediately after AXON commits its selected anchors. In the pre-commit state, attention is largely dominated by local, diagonal structures. However, in the post-commit state, distinct vertical streaks appear corresponding exactly to the positions AXON just unmasked. Once these structural anchors are converted from [MASK] to real tokens, they become highly informative landmarks. The remaining residual tokens rapidly route their attention mass to these new visible contexts, confirming that the chosen anchors successfully absorb the unresolved dependency structure of the block.

C.3 Token Unmasking Dynamics

To isolate AXON's efficiency gains (NFE), we analyze the unmasking profiles within a single decoding block in Figure 6. Both DAWN and AXON maintain a remarkably flat per-step profile (Figure 6b). However, AXON consistently achieves

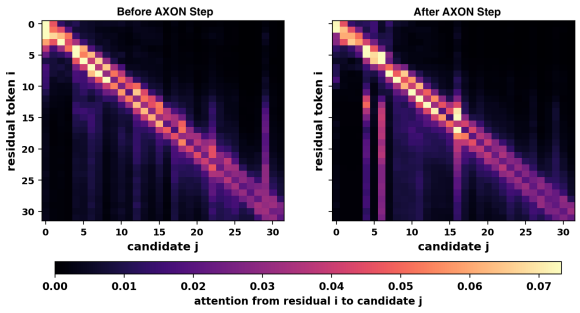


Figure 5: **Attention routing before and after a AXON intervention.** The 32×32 attention slice for a single block (Block 0, Step 0 to Step 1) of an MBPP problem. **Left:** The step immediately before AXON fires. **Right:** The step immediately after AXON anchors are committed. The emergence of bright vertical streaks, such as columns 4, 6, and 16, shows newly revealed anchor tokens absorbing attention mass from the remaining masked residuals.

a stable $\sim$9% advantage at every step, resolving roughly 3.5 tokens compared to DAWN's 3.2. As shown in the cumulative trajectory (Figure 6a), this minor dividend systematically compounds, allowing AXON to resolve $28/32$ tokens by the end of the block versus DAWN's $26/32$. Aggregated over a full decoding horizon, this saves roughly 16 forward passes within the same token budget, directly explaining AXON's lower overall NFE. The consistent gap, caused by AXON firing , confirms that submodular coverage steadily guides the model toward high-yield positions.

D Submodularity of the Coverage Objective

Proposition 1. The facility-location coverage function in Eq. 2 is monotone submodular.

Proof. Fix a denoising step t, and let U be the residual positions that the anchor set should support and V be the candidate anchor positions. Assume all weights are nonnegative, $w_{ij} \geq 0$, which holds for $w_{ij}^{(t)} = A_{ij}^{(t)}(1-c_i^{(t)})c_j^{(t)}$ because attention weights and confidence terms are nonnegative. For any selected anchor set $S \subseteq V$, recall that

$$C(S) = \sum_{i \in U} \max_{j \in S} w_{ij},$$

with the convention that $\max_{j \in \emptyset} w_{ij} = 0$.

Monotonicity. Let $A \subseteq B \subseteq V$. For each residual position i,

$$\max_{j \in A} w_{ij} \leq \max_{j \in B} w_{ij},$$

because B contains every candidate in A and possibly more. Summing over $i \in U$ gives $C(A) \leq C(B)$. Therefore, C is monotone.

Submodularity. We show diminishing returns. Let $A \subseteq B \subseteq V$ and let $e \in V \setminus B$. For any set $S \subseteq V$, write

$$m_i(S) = \max_{j \in S} w_{ij}.$$

The marginal gain from adding e to S is

$$\begin{aligned} C(S \cup \{e\}) - C(S) &= \sum_{i \in U} \left(\max\{m_i(S), w_{ie}\} - m_i(S)\right) \\ &= \sum_{i \in U} \max\{0, w_{ie} - m_i(S)\}. \end{aligned}$$

Since $A \subseteq B$, we have $m_i(A) \leq m_i(B)$ for every $i \in U$. Thus,

$$\max\{0, w_{ie} - m_i(A)\} \geq \max\{0, w_{ie} - m_i(B)\}$$

for every i. Summing over all residual positions yields

$$C(A \cup \{e\}) - C(A) \geq C(B \cup \{e\}) - C(B).$$

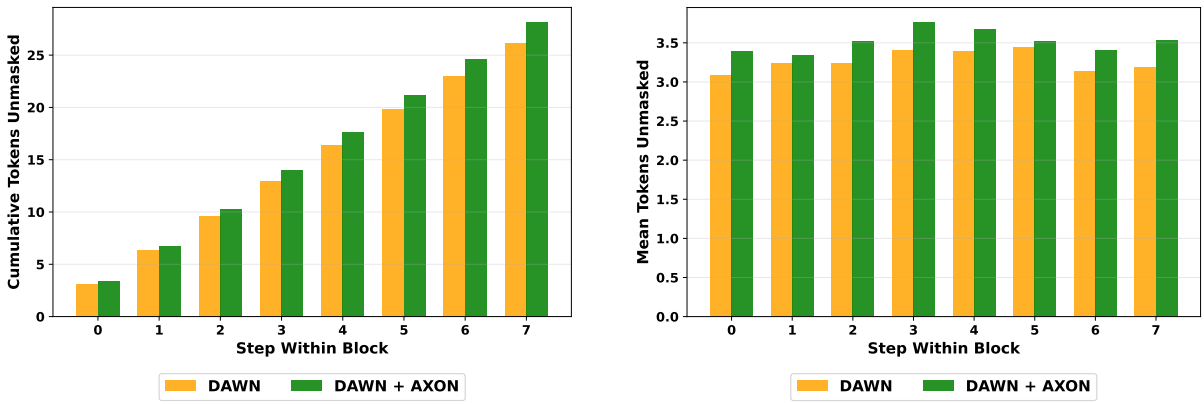
This is exactly the submodularity condition. Hence C is monotone submodular. □

E Submodular-Objective Ablation

We ablate AXON's set-level anchor-selection objective along two axes.

1. **Coverage form.** Facility-location (FL) $C_{\mathrm{FL}}^{(t)}(S) = \sum_{i \in U} \max_{j \in S} w_{ij}^{(t)}$, in which each uncertain position draws its coverage from the best-matching anchors within the selected set S, versus Graph-Cut (GC) $C_{\mathrm{GC}}^{(t)}(S) = \sum_{i \in U} \sum_{j \in S} w_{ij}^{(t)}$, which instead sums each position's affinity over all selected anchors.

2. **Monotonicity.** The monotone objective $F_{\mathrm{mono}}(S) = C^{(t)}(S)$ versus the non-monotone $F_{\mathrm{nonmono}}(S) = C^{(t)}(S) - \lambda R(S)$, where $R(S) = \sum_{\{j,k\} \subseteq S} q_{jk}$ is a pairwise redundancy/conflict penalty over the selected set and its weight, $\lambda \geq 0$.



(a) Cumulative tokens unmasked

(b) Mean tokens unmasked per individual step

Figure 6: **Token unmasking dynamics.** Analysis of generation mechanics within a single decoding block on Humaneval dataset, utilizing LlaDA1.5, and on top of DAWN, comparing (a) the cumulative trajectory of resolved tokens and (b) the mean unmasking rate per discrete step. AXON consistently unmask a larger number of tokens per step.

Protocol. All ablation runs use the standard AXON configuration. We sweep on the two datasets. MBPP on all three models, and HumanEval on the two LLaDA models, giving five (model, task) cells. For each cell, we run both objectives in $\{\text{FL}, \text{GC}\}\times\{\text{mono}, \text{nonmono}\}$, and for the non-monotone variants we sweep $\lambda \in \{0.2, 0.3, 0.4, 0.5\}$, giving 10 configurations per cell.

Objective form: FL vs. GC (monotone)

We first compare the monotone facility-location (FL) objective against a graph-cut (GC). As we showed in Table 2, FL consistently outperforms GC in computational efficiency while maintaining comparable accuracy. FL achieves a lower number of function evaluations (NFE) across all five evaluation cells, yielding a lower mean NFE overall (*44.89* vs. *45.95*). This confirms that explicitly enforcing structural diversity prevents the model from generating additive and redundant anchor coverage, thus streamlining the generation process without sacrificing task performance.

Non-monotone penalty: λ sweep on FL

We test whether explicitly penalizing redundant anchors improves selection. As shown in Table 3, we sweep the penalty weight λ on the FL objective to observe if non-monotone regularization yields a better accuracy-NFE tradeoff compared to the purely monotone baseline ($\lambda = 0$). The results demonstrate that introducing a non-monotone redundancy penalty provides no meaningful benefit to either task accuracy or generation efficiency.

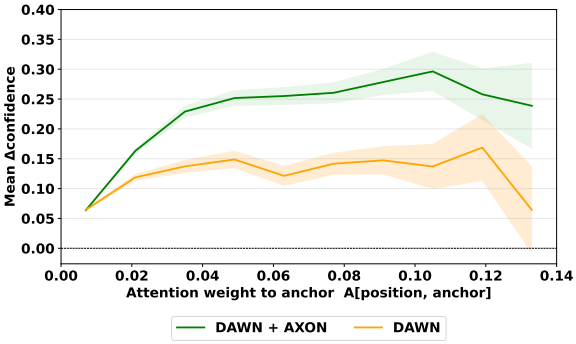


Figure 7: **Confidence propagation from AXON anchors scales with attention connectivity.** For each decoding step where a submodular anchor token is committed, we track the per-position confidence gain $\Delta\text{confidence} = x_0^p[t+1] - x_0^p[t]$ for all remaining masked positions, binned by their attention weight to that anchor $A[\text{position}, \text{anchor}]$. Results are averaged over HumanEval dataset, and utilizing LLaDA-1.5.

Across all five evaluation cells, shifting λ from 0 up to 0.5 leaves the mean accuracy virtually unchanged (fluctuating negligibly between *39.88%* and *39.96%*). Similarly, the computational cost remains entirely stagnant, with the mean NFE locked tightly between *44.86* and *44.91*. This flat trend empirically justifies our choice of the purely monotone objective: because complex pairwise coordination penalties do not yield a superior improvement, the simpler, hyperparameter-free, monotone objective ($\lambda = 0$) remains the optimal configuration.

		λ=0 (mono)		λ=0.2		λ=0.3		λ=0.4		λ=0.5	
Model	**Task**	Acc	NFE	Acc	NFE	Acc	NFE	Acc	NFE	Acc	NFE
Dream-7B	MBPP	55.00	41.81	55.00	41.82	54.80	41.88	54.80	41.90	54.80	41.91
LLaDA-1.5	MBPP	35.20	28.30	35.40	28.19	35.40	28.19	35.40	28.18	35.20	28.19
LLaDA-8B	MBPP	30.40	42.40	30.40	42.40	30.40	42.41	30.40	42.42	30.40	42.60
LLaDA-1.5	HumanEval	39.02	59.49	39.02	59.43	39.02	59.43	39.02	59.30	39.02	59.30
LLaDA-8B	HumanEval	39.98	52.43	39.98	52.59	39.98	52.65	39.98	52.65	39.98	52.29
Mean across cells		*39.92*	*44.89*	*39.96*	*44.89*	*39.92*	*44.91*	*39.92*	*44.89*	*39.88*	*44.86*

Table 3: **Ablating monotonicity.** Results comparing facility-location (FL) with monotone objective ($\lambda = 0$) against the nonmonotone objective ($\lambda \in \{0.2, 0.3, 0.4, 0.5\}$) across tasks and model configurations.

Confidence Propagation From AXON

Figure 7 validates the core structural premise of AXON's facility-location objective: anchors should be chosen not merely for their own high confidence, but for their structural capacity to propagate information to the unmasked positions that attend to them. At near-zero attention connectivity ($A \approx 0$), both methods perform identically, yielding a minimal confidence gain (~ 0.04). However, as attention weight increases, AXON's confidence gain (Δconfidence) rises steeply and plateaus at ~ 0.25–0.30. This represents a 2x improvement over DAWN's flat plateau (~ 0.12–0.15), proving that AXON's submodular selection successfully routes information to highly-attending positions to accelerate model confidence.

F Gate Weight α Ablation

The hybrid gate of AXON combines two deficit signals through a single weight α: the context deficit $d_{\text{ctx}}^{(t)}$ Eq. 5 and the pace deficit $d_{\text{pace}}^{(t)}$ Eq. 3. It fires whenever the convex combination of their positive rectifications exceeds zero (see Eq. 6). The two endpoints recover the pure regimes: $\alpha = 1$ gates solely on the context deficit (ensuring contextual support scales with residual uncertainty and commit volume), while $\alpha = 0$ gates solely on the pace deficit (ensuring the base decoder maintains the required completion schedule).

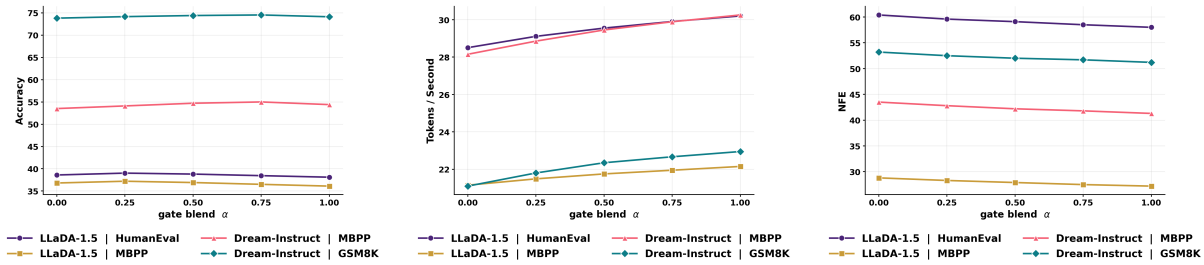
To quantify how sensitive AXON is to this choice, we sweep $\alpha \in \{0, 0.25, 0.5, 0.75, 1\}$ on top of the DAWN base decoder (Figure 8). We evaluate four model $\times$ task configurations: LLaDA-1.5 on HumanEval and MBPP, and Dream-Instruct on MBPP and GSM8K, reporting accuracy, TPS, and NFE as functions of α in Figures 8a, 8b, and 8c respectively.

Accuracy. As illustrated in Figure 8a, the impact of the gate weight α on generation accuracy reveals a clear, model-specific optimum, demonstrating that performance is non-monotonic. Pushing the gate to either extreme relying solely on the pace schedule ($\alpha = 0$) or exclusively on contextual support ($\alpha = 1.0$) is often sub-optimal, validating the need to interpolate between both deficit checks. We configure AXON with $\alpha = 0.2$ and $\alpha = 0.9$ for the LLaDA and Dream-Instruct backbones respectively across all main-text evaluations, tuning the framework to the specific failure modes of each base model.

TPS Throughput and NFE. Generation throughput and computational overhead scale dynamically with α (see Figures 8b and 8c). By shifting the gate's sensitivity, the framework becomes more or less tolerant of rapid, under-supported commits, thereby firing the submodular selection at different frequencies. Tuning this parameter safely accelerates generation speed and streamlines compute cost, offering a predictable trade-off trajectory against downstream quality depending on the strictness of the deployment budget.

G Relationship between confidence gain and attention

Overview and bound. In this section we analyze how revealing a chosen anchor affects the confidence of the model at a masked position. Explicitly, we will prove an upper bound on the increase in confidence at a masked position after the anchor is revealed, in terms of the attention map A. Let the masked position be i, and the chosen anchor be at position s and let $y_s \in \mathcal{V} \setminus \{\texttt{[MASK]}\}$ be the revealed (anchor) token. For simplicity, we assume that the prompt X is empty, i.e. the input to the dLLM consists of only the response vector $y^{(t)} = (y_1^{(t)}, \ldots, y_L^{(t)})$. Note that $y_s^{(t)} = \texttt{[MASK]}$ prior to revealing the anchor. Denote the response



(a) Accuracy vs. α (b) Throughput vs. α (c) Average NFE vs. α

Figure 8: **Ablating gate weight α.** Performance metrics across multiple models and tasks evaluated on top of DAWN under varying gate configurations, $\alpha \in \{0.0, 0.25, 0.5, 0.75, 1.0\}$. This illustrates the simultaneous impact of α on (a) downstream task accuracy, (b) token generation throughput (TPS), and (c) total computational cost measured in average NFE.

vector after the reveal as $\tilde{y}^{(t)}$, where

$$\tilde{y}_j^{(t)} = \begin{cases} y_s & \text{if } j = s, \\ y_j^{(t)} & \text{otherwise .} \end{cases}$$

Let $\mathcal{V}' = \mathcal{V} \setminus \{\texttt{[MASK]}\}$. Thus, the increase in confidence (*confidence gain*) for position i after revealing position s can be defined as,

$$\delta_i = \max_{v \in \mathcal{V}'} p_\theta(y_i = v | \tilde{y}^{(t)}) - \max_{v \in \mathcal{V}'} p_\theta(y_i = v | y^{(t)}).$$

While it is intuitive that the confidence of position i should be more sensitive to the reveal of token s when the corresponding attention entry A_{is} is large, it is challenging to make this intuition rigorous due to the complexity of the transformer model p_θ. To make progress in this direction, we adopt certain simplifying assumptions following prior work (Zhou et al., 2026), namely,

1. Following (Zhou et al., 2026), we assume that the model p_θ is parametrized by a single layer transformer.

2. We assume that the attention map A does not change after revealing position s (similar to (Zhou et al., 2026) Assumption 3.1).

Under these assumptions, we can prove that,

$$\delta_i \leq \beta A_{is}, \tag{7}$$

for a positive constant $\beta > 0$.

Discussion of bound. The upper bound in (7) motivates our use of attention as a measure for anchor usefulness, and the inclusion of A_{ij} in the score w_{ij}. Indeed, (7) implies that the improvements for masked tokens (with respect to confidence) is limited by their dependency on the anchor position s (as measured by A_{is}). This is corroborated by Figure 7, where low attention values are correlated with low confidence gains. Thus, naively choosing an anchor s may have only a minimal effect on the confidence of remaining tokens if the column $A_{\cdot,s}$ has small entries. This potentially explains the superior performance of AXON with respect to confidence gain (see Figure 7), since AXON's submodular objective explicitly addresses this factor.

However, our analysis is limited by our assumptions. In particular, assuming a static attention map fails to account for the case where revealing an anchor s strongly affects a masked position i, that was earlier coupled with the anchor only through a positional dependency (e.g. adjacent tokens). For example, if the anchor is revealed to be a word that primarily occurs in a multi-token entity, e.g. "York", the attention map can potentially change a lot for certain positions (e.g. the preceding position). Extending this analysis to consider such effects is left for future work. We now explicitly discuss the assumed architecture of p_θ, followed by a proof of (7).

Architecture assumption. Explicitly, let $\text{Embed} : \mathcal{V} \to \mathbb{R}^d$ be a d-dimensional embedding of the token space, and $\text{Positional} : \{1, \ldots, L\} \to \mathbb{R}^d$ be a positional encoding function. As before, let $y^{(t)} = (y_1, \ldots, y_L)$ be the partially masked response at diffusion step t. Let $p_\theta(y_i | y^{(t)}) \in \mathbb{R}^{|\mathcal{V}|}$ denote the full categorical distribution predicted by the model.

For our assumed architecture, the predicted distribution is computed in the following steps. First, the embeddings for each position i are computed as follows,

$$h_i = \text{emb}(y_i, i) = \text{Embed}(y_i) + \text{Positional}(i).$$

Given the query, key and value projection matrices

W_Q, W_K, W_V, the attention maps A_{ij} and output representations for the attention block are computed as follows,

$$q_i = W_Q h_i, \quad k_i = W_K h_i, \quad v_i = W_V h_i,$$
$$A_{ij} = \frac{\exp(q_i^\top k_j/\sqrt{d})}{\sum_j \exp(q_i^\top k_j/\sqrt{d})},$$
$$z_i = \sum_j A_{ij} v_j.$$

Finally, a fully connected layer (with parameters (W, b)), followed by a softmax activation, is applied to each position independently to obtain the predicted distributions as follows :

$$p_\theta(y_i|y^{(t)}) = \text{Softmax}(W z_i + b).$$

Define the notation,

$$g(z) = \text{Softmax}(W z + b).$$

Since g is differentiable everywhere, therefore g is also locally Lipschitz continuous. Note that since $\mathcal{V}$ and $\{1, \ldots, L\}$ are finite sets, given that $\text{emb}(v, i)$ is always well-defined, there exists some $B_e > 0$ such that $\|h_i\|_2 \leq B_e$ for every position i. Next, since W_V is a (bounded) linear mapping and $A_{ij} \in [0, 1]$, there also exists some $B > 0$ such that $\|z_i\|_2 \leq B$ for every position i. Since g is locally Lipschitz, it is globally Lipschitz on the bounded ball $\|z\|_2 \leq B$. Let this global Lipschitz constant be L, therefore,

$$\|g(z_1) - g(z_2)\|_1 \leq L\|z_1 - z_2\|_2. \tag{8}$$

Note that we use the absolute value norm on the left hand side. Since all norms are topologically equivalent in Euclidean spaces, (8) is equivalent to expressing it purely in terms of the Euclidean norm on both sides. This Lipschitz property is the main tool we will use to prove the upper bound (7) in the next part.

Derivation of upper bound. Clearly,

$$\begin{aligned}
&\delta_i \\
&\leq |\delta_i| \\
&= |\max_{v \in \mathcal{V}'} p_\theta(y_i = v|\tilde{y}^{(t)}) - \max_{v \in \mathcal{V}'} p_\theta(y_i = v|y^{(t)})| \\
&\leq \max_{v \in \mathcal{V}'} |p_\theta(y_i = v|\tilde{y}^{(t)}) - p_\theta(y_i = v|y^{(t)})|,
\end{aligned}$$

where the last bound follows from the identity $|\max_x f(x) - \max_x g(x)| \leq \max_x |f(x) - g(x)|$. Next, clearly the maximum entry of a positive valued vector is smaller than the sum of its entries, therefore,

$$\begin{aligned}
&\max_{v \in \mathcal{V}'} |p_\theta(y_i = v|\tilde{y}^{(t)}) - p_\theta(y_i = v|y^{(t)})| \\
\leq &\|p_\theta(y_i|\tilde{y}^{(t)}) - p_\theta(y_i = v|y^{(t)})\|_1 \\
= &\|g(\tilde{z}_i) - g(z_i)\|_1,
\end{aligned}$$

where $\tilde{z}_i$ is the output representation at position i for the input $\tilde{y}^{(t)}$ (i.e. after unmasking the anchor). We can thus apply the Lipschitz bound (8) to obtain,

$$\begin{aligned}
\delta_i &\leq L\|\tilde{z}_i - z_i\|_2 \\
&= L\left\|\sum_j A_{ij}(\tilde{v}_j - v_j)\right\|_2,
\end{aligned}$$

where $\tilde{v}_j$ is the value computed at position j for input $\tilde{y}^{(t)}$. Note that we have used the assumption here that the attention map remains the same for inputs $\tilde{y}^{(t)}$ and $y^{(t)}$. Since only position s is different, therefore, $\tilde{v}_j = v_j$ iff $j \neq s$. Therefore,

$$\delta_i \leq L A_{is} \|\tilde{v}_s - v_s\|_2$$

Let

$$C = \max_{\substack{x,y \in \mathcal{V} \\ i,j \in \{1,\ldots,L\}}} \|W_V \text{emb}(x, i) - W_V \text{emb}(y, j)\|.$$

Clearly $C > 0$, and since $\mathcal{V}$ and $\{1, \ldots, L\}$ are finite sets, therefore $C < \infty$. Therefore,

$$\begin{aligned}
&\|\tilde{v}_s - v_s\|_2 \\
= &\|W_V \text{emb}(y_s, s) - W_V \text{emb}(\texttt{[MASK]}, s)\|_2 \leq C.
\end{aligned}$$

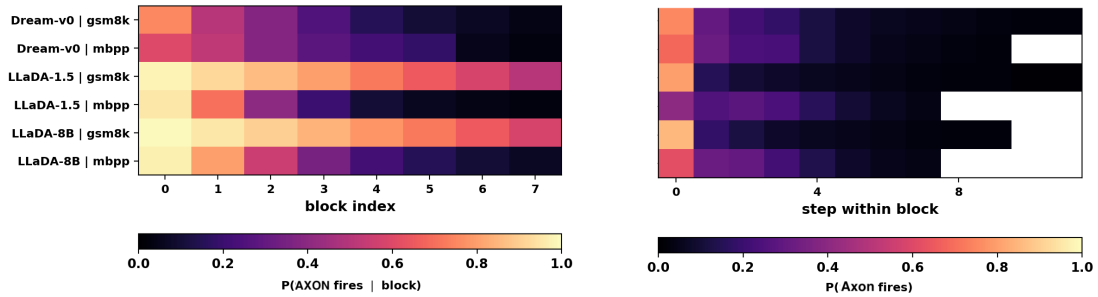
Thus, we obtain

$$\delta_i \leq LCA_{is},$$

which proves (7).

H Effects Of AXON On The Decoders

The interaction between anchor selection and accelerated decoding strategies is heavily dependent on the structural representation mechanism of the method. For the simple thresholding-based methods, which make decisions to immediately commit a token only based on a flat probability threshold, AXON acts as a co-dependency filter; by evaluating global information entropy, it penalizes mutual context redundancy, eliminating simultaneous-masking conflicts and yielding a steady recovery



(a) AXON activation per block **(b) AXON average activation per step**

Figure 9: **AXON gate firing patterns across generations.** Complementary heatmaps show the fraction of evaluation steps where the gate activates ($G^{(t)} = 1$) across Dream-Instruct and LLaDA backbones on MBPP and GSM8K. **(a) Within block activation.** AXON intervention heavily concentrate in the early blocks of a 256-token sequence, where the base proposer requires the most structural support. **(b) Within step activation.** Within any given block, AXON firing peaks during the first few denoising steps (at the block opening) and sharply declines once the base proposer catches up.

in both generation quality and Throughput. For heuristically-driven spatial methods (LocalLeap), which expand unmasking zones natively by lowering confidence thresholds uniformly within a fixed, linear geometric radius around predicted anchors, AXON enforces a diversity constraint that eliminates anchor clumping, preventing localized text degradation and primarily optimizing Accuracy. For graph-driven structural methods (DAWN), which is based on model token interdependencies by extracting raw attention maps and utilizing a Maximum Independent Set solver to parallelize non-conflicting tokens, AXON provides global informational pillars that clarify noisy attention maps, allowing the dependency solver to uncover massive parallel sets and maximize throughput (TPS).

I Analysis of Gate Firing Patterns

Figure 9 visualizes the probability of the AXON gate firing ($G^{(t)} = 1$).

Across blocks (Macro). Figure 9a shows that activations are heavily front-loaded. Early in the generation, causal context is sparse and residual uncertainty is high, requiring AXON to intervene frequently. As the sequence progresses and reliable context accumulates, the gate naturally backs off.

Within blocks (Micro). Figure 9b reveals that interventions concentrate almost entirely at Step 0. When a new block opens, structural support is minimal. AXON immediately injects high-leverage anchors to clear this bottleneck, allowing the base decoder to rapidly resolve the rest of the block uninterrupted. The truncated right edge of the heatmap

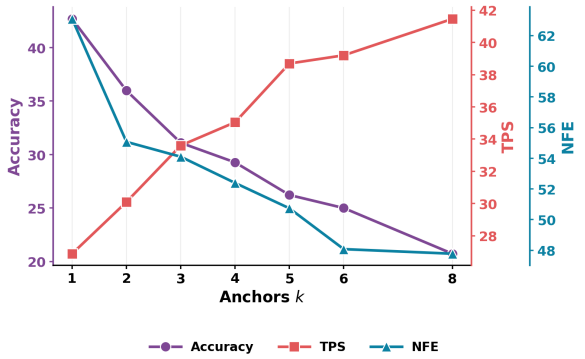


Figure 10: **Impact of anchor budget on decoding dynamics on Llada-1.5 and HumanEval.** Accuracy, throughput (TPS), and NFE across varying anchor set sizes.

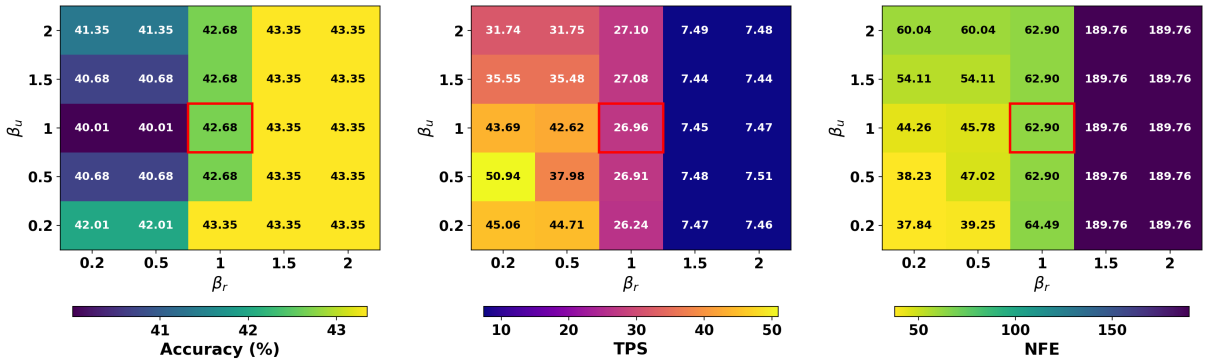
visually confirms that this early intervention reduces the total steps required to finish the block.

J Effect of the Number of Anchors

AXON uses a small anchor budget because the goal is not to maximize the number of additional committed tokens, but to reveal a small number of tokens that provide high contextual support to uncertain residual positions. The default variant uses a fixed budget of $k = 1$ anchor whenever the gate fires. This setting keeps the intervention cheap and conservative, while the facility-location objective discourages redundant reveals by giving each uncertain position credit only for its best selected anchor.

We also evaluate AXON_{CVR}, an adaptive coverage-matching variant. Let

$$\tau^{(t)} = C^{(t)}\left(S^{(t)}_{\text{base}}\right)$$



(a) Accuracy vs. β_u and β_r (b) Throughput vs. β_u and β_r (c) Average NFE vs. β_u and β_r

Figure 11: **Ablating scaling coefficients β_u and β_r.** Performance metrics utilizing LlaDa1.5 backbone, on HumanEval task, on top of DAWN decoder under varying scaling coefficients, $\{\beta_u, \beta_r\} \in \{0.2, 0.5, 1.0, 1.5, 2.0\} \times \{0.2, 0.5, 1.0, 1.5, 2.0\}$. This illustrates the simultaneous impact of β_u and β_r on (a) downstream task accuracy, (b) token generation throughput (TPS), and (c) total computational cost measured in average NFE.

denote the contextual coverage induced by the base decoder's proposal. Starting from $S_0 = \varnothing$, AXON$_{\text{CVR}}$ greedily adds the candidate anchor with largest marginal gain under $C^{(t)}$ until

$$C^{(t)}(S_m) \geq \tau^{(t)}.$$

Equivalently, its final cardinality is

$$|S^{\star(t)}_{\text{CVR}}| = \min\left\{m \geq 1 :\ C^{(t)}(S_m) \geq \tau^{(t)}\right\}.$$

This gives the adaptive variant a per-step scale for how much contextual support is needed, while still selecting anchors according to the same non-redundant coverage objective. For any realized cardinality m, the greedy prefix retains the standard $(1 - 1/e)$ approximation guarantee for monotone submodular maximization under a cardinality budget of m.

In Figure 10, we ablate the fixed per-fire anchor budget and report its effect on NFE, tokens per second, and accuracy.

Larger budgets commit more tokens per intervention, shrinking the residual mask $M^{(t)}$ and lowering NFE monotonically. Throughput tracks NFE initially but plateaus once the greedy submodular search overhead begins to dominate the savings. Accuracy follows an inverted U-shape: a small set of high-confidence anchors absorbs the attention of uncertain residuals and resolves stalled positions, while larger budgets force selection from the low-confidence tail of the candidate pool $V^{(t)}$, injecting noise that propagates through subsequent denoising steps. The settings deployed in Section 5 sit at the Pareto frontier of this trade-off.

K Context Deficit Normalization β

The context deficit $d^{(t)}_{\text{ctx}}$ (Eq. 5) evaluates the contextual support $g^{(t)}$ against two distinct signals: the decoding pace $r^{(t)}$ and the residual uncertainty $u^{(t)}_M$. While all three quantities naturally fall within the $[0, 1]$ interval, they originate from different distributions and lack a universally scaled theoretical equivalence. As discussed in Appendix G, this correlates between attention mass and uncertainty reduction. However, to account for potential distribution shifts across different model architectures, one could introduce scaling coefficients $\beta_r, \beta_u > 0$ to explicitly calibrate these thresholds:

$$d^{(t)}_{\text{ctx}} = \max\left(\beta_r r^{(t)},\ \beta_u u^{(t)}_M\right) - g^{(t)}. \tag{9}$$

Setting $\beta_r = \beta_u = 1$ establishes a parameter-free uniform-prior baseline which we use in our experiments in Table 1. To evaluate the sensitivity of AXON to these normalization constants, we perform a 2D grid search over $\beta_r, \beta_u \in \{0.2, 0.5, 1.0, 1.5, 2.0\}$ (Figure 11), reporting Accuracy, TPS, and NFE.

Accuracy. In Figure 11a, the grid search reveals a clear trade-off spectrum depending on the strictness of the gate. At lower threshold values ($\beta \leq 0.5$), the gate is overly permissive; it frequently fails to intervene when the base decoder lacks sufficient structural context, causing generation accuracy to drop. As the β coefficients increase, the gate becomes stricter, forcing the model to rely more heavily on the injected anchors which pushes accuracy.

TPS Throughput and NFE. While aggressive gating (high β) yields marginal accuracy gains, it

Backbone	Decoder	plain	+AXON	Δacc	NFE (plain→+AXON)	TPS (plain→+AXON)	Speedup
Dream-7B	Confidence	45.04	52.26	+7.22	256.0 → 30.9	1.67 → 8.58	5.13×
	DAWN	50.43	52.39	+1.96	26.3 → 25.8	9.12 → 10.53	1.16×
	LocalLeap	50.80	52.26	+1.47	26.4 → 25.9	8.80 → 10.12	1.15×
LLaDA-1.5	Confidence	50.92	51.16	+0.24	256.0 → 36.9	1.37 → 9.36	6.81×
	DAWN	49.82	50.31	+0.49	33.9 → 27.4	10.36 → 12.79	1.23×
	LocalLeap	50.80	50.92	+0.12	37.0 → 31.5	9.27 → 10.66	1.15×

Table 4: **TruthfulQA.** Open-ended, non-coding, non-reasoning evaluation on Dream-7B and LLaDA-1.5.

	HumanEval			TruthfulQA		
Decoder	Acc.	TPS	NFE	Acc.	TPS	NFE
baseline	74.39	7.1	160.5	42.96	9.3	106.7
Confidence	76.22	7.7	164.7	43.33	8.5	109.6
+AXON	79.27	8.7	165.8	44.43	9.6	111.4
+AXON$_{\text{CVR}}$	79.27	8.7	165.8	44.43	9.7	111.4
LocalLeap	73.78	8.4	150.6	43.45	8.9	104.9
+AXON	77.44	9.4	152.0	44.55	10.0	107.2
+AXON$_{\text{CVR}}$	77.43	9.4	152.0	44.55	10.1	107.2
DAWN	74.39	9.1	148.5	43.70	10.7	105.3
+AXON	74.39	10.0	147.2	45.57	11.0	104.9
+AXON$_{\text{CVR}}$	74.39	10.0	147.2	45.57	11.0	104.9

Table 5: **SDAR-8B backbone results.** An additional dLLM backbone on HumanEval and TruthfulQA.

incurs a catastrophic penalty to efficiency. Thresholds above 1.0 force the gate to fire almost continuously which inflates the number of forward passes (NFE) and decreasing throughput (TPS), as depicted in Figures 11c and 11b respectively, but with a direct trade-off with accuracy.

L Generalization to Additional Datasets and Backbones

To test whether AXON's gains extend beyond code generation and math reasoning, we evaluate on TruthfulQA (Lin et al., 2022), an open-ended, non-coding, non-reasoning benchmark spanning 38 real-world categories that targets factual accuracy and hallucination resistance. Table 4 reports results on Dream-7B and LLaDA-1.5 across all three base decoders. AXON improves accuracy in every configuration, with the largest speedups on top of Confidence decoding (5.13× on Dream-7B, 6.81× on LLaDA-1.5), since Confidence's largely sequential commit pattern on this dataset leaves the most headroom for AXON's anchors.

We additionally evaluate a third dLLM backbone, SDAR-8B (Cheng et al., 2026), on HumanEval and TruthfulQA (MBPP and GSM8K runs were still in progress at submission time). Table 5 shows the same pattern observed on Dream and LLaDA: AXON improves accuracy, throughput, or both across every decoder, indicating the anchor-selection mechanism generalizes across backbone families rather than exploiting properties specific to LLaDA or Dream.