

FourierSampler: Unlocking Non-Autoregressive Potential in Diffusion Language Models via Frequency-Guided Generation

Siyang He^{1,2,3*}, Qiqi Wang^{1,2,3*}, Xiaoran Liu^{1,2,3}, Hongnan Ma³,
Yiwei Shi³, Yuerong Song^{1,2,3}, Ying Zhu^{1,2,3}, Tianyi Liang^{2,3},
Zengfeng Huang^{1,2}, Ziwei He^{2,3†}, Xipeng Qiu^{1,2,3†}

¹Fudan University, ²Shanghai Innovation Institute, ³OpenMOSS Team,

Correspondence: ziwei.he@sii.edu.cn, xpqiu@fudan.edu.cn

Abstract

Despite the non-autoregressive potential of diffusion language models (dLLMs), existing decoding strategies exhibit positional bias, failing to fully unlock the potential of arbitrary-order generation. In this work, we delve into the inherent spectral characteristics of dLLMs and present the first frequency-domain analysis of dLLMs, showing that low-frequency components in hidden states primarily encode global structural information and long-range dependencies, while high-frequency components capture local details. Based on this observation, we propose FourierSampler, which leverages a frequency-domain sliding window mechanism to dynamically guide the model to achieve "structure-to-detail" generation. FourierSampler outperforms other inference enhancement strategies on LLaDA and SDAR, achieving relative task-specific gains up to 20.4% on LLaDA-1.5 and 16.0% on LLaDA-8B-Instruct, as well as average gains up to 8.6% across tasks.

1 Introduction

Diffusion Large Language Models (dLLMs) (Sahoo et al., 2024; Ou et al., 2025; Shi et al., 2024) have emerged as a notable non-autoregressive paradigm. Unlike autoregressive models (OpenAI, 2023; Grattafiori et al., 2024) constrained by prefix history and left-to-right decoding, dLLMs support arbitrary-order generation and can leverage global bidirectional context throughout generation. This flexibility brings several advantages over AR models, including mitigating the reversal curse (Berglund et al., 2024), supporting controllable generation such as text infilling, and enabling non-sequential planning for complex problems.

Despite these advantages, dLLMs still exhibit strong positional bias during practical generation (Huang et al., 2026). It's also shown that

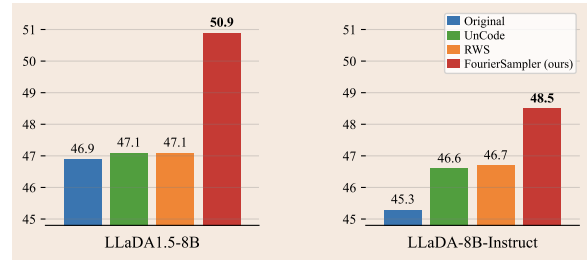


Figure 1: Average score across GSM8K, MATH, MBPP, HumanEval and Countdown for different decoding strategies in LLaDA (Nie et al., 2025; Zhu et al., 2026).

enforcing a left-to-right order can outperform standard confidence-based decoding on specific tasks. However, such a rigid order limits dLLMs' inherent arbitrary-order flexibility and their ability to leverage global bidirectional context for holistic planning. Recent rule-based (Huang et al., 2026) and reward-guided methods (Gwak et al., 2025) further show that optimizing the decoding schedule is critical for improving dLLM generation.

Unlike the above works that calibrate the sampling distribution with external priors, we introduce a principled approach to mitigate positional bias and unlock the intrinsic non-autoregressive potential of dLLMs. Instead of relying on external intervention signals, we delve into the frequency-domain characteristics of the model's internal representations. Motivated by the spectral observations in Fourier Transformer (He et al., 2023) and our theoretical analysis of frequency-dependent wavelength and local-variation contribution, we hypothesize that low-frequency components in textual representations primarily encode global structural information and long-range dependencies, whereas high-frequency components capture local details. We further support this hypothesis through case studies and quantitatively validate it with synonym perturbation experiments, revealing a structure-detail semantic hierarchy in the frequency domain.

Based on this intuition, we propose FourierSam-

* Equal contribution.

† Corresponding Author.

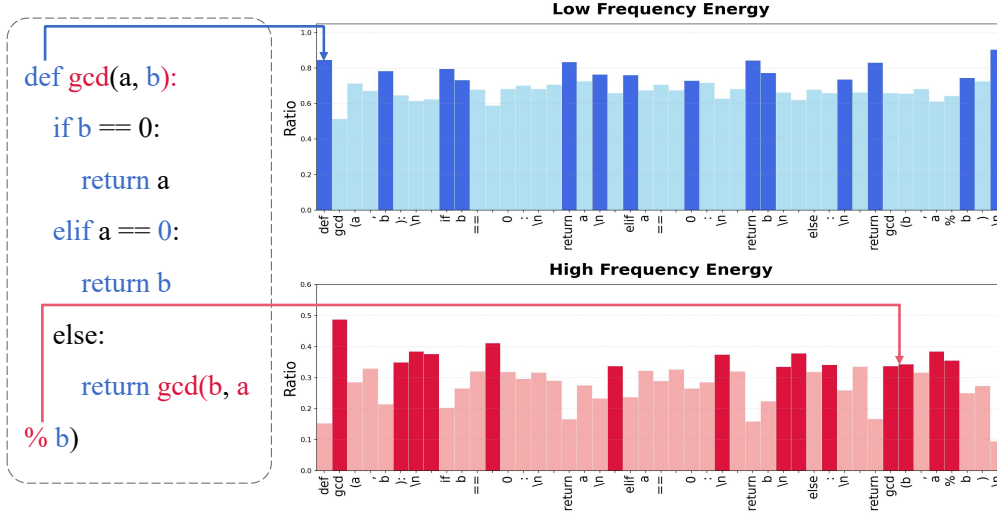


Figure 2: Visualization of the correspondence between frequency-domain analysis and textual information. In the hidden states after a forward pass, tokens dominated by low-frequency signals correspond to structural information like *if* and *elif*, while tokens dominated by high-frequency signals correspond to detailed information like *gcd*.

pler, a decoding strategy that dynamically focuses on different frequency bands at different decoding steps, guiding dLLMs to first decode structural content dominated by low-frequency signals and subsequently complement detailed content dominated by high-frequency signals. This produces a beneficial long-range influence on the generation trajectory, thereby enhancing the overall quality and logical coherence of the output. As shown in Figure 1, our method significantly outperforms the original model and other work relying on external signal-guided decoding on different tasks. Our contribution can be summarized as follows.

- We present the first frequential analysis of dLLM hidden states, showing that low-frequency components in temporal dimension encode structural information, while high-frequency components capture local details, providing internal guidance for decoding.
- We propose **FourierSampler**, a frequency-guided decoding scheme that promotes structure-to-detail generation with Translated Fourier Score and adaptively balances it with confidence via Adaptive Fourier Calibrator.
- We evaluate FourierSampler on LLaDA and SDAR, achieving stable gains on code and math tasks, with peak relative improvements up to 20.4%, 16.0%, 45.1%, and 26.5%, and average gains of 8.6%, 7.1%, 7.4%, and 6.5% on LLaDA-1.5, LLaDA-8B-Instruct, SDAR-1.7B-Chat, and SDAR-4B-Chat. FourierSam-

pler consistently outperforms other competitive methods, offering deep insights into dLLM decoding enhancement.

2 Related Work

2.1 Decoding Strategy for dLLMs

Diffusion Large Language Models (dLLMs) (Sahoo et al., 2024; Shi et al., 2024) have demonstrated remarkable scalability (Gong et al., 2025; Ni et al., 2025), as evidenced by models like LLaDA (Nie et al., 2025), Dream (Ye et al., 2025b) and SDAR (Cheng et al., 2026). Although capable of arbitrary decoding orders, their generation trajectories exhibit pronounced positional bias (Gwak et al., 2025). Recent works focus on optimizing token unmasking orders. These approaches typically prioritize tokens via maximum probability, entropy (Ben-Hamu et al., 2025), or confidence gaps (Kim et al., 2025), which outperform random sampling baselines (Austin et al., 2021a).

To mitigate positional bias and improve generation performance, advanced methods introduce external interventions. For example, UnCode (Huang et al., 2026) uses rule-based biases for specific positions and tokens, while RWS (Gwak et al., 2025) employs reward models to enhance coherence. Training-based approaches like DOT (Ye et al., 2024), DDPD (Liu et al., 2025a), and DCoLT (Huang et al., 2025) optimize generation trajectories via post-training or reinforcement learning. However, these methods rely on external signals or costly training, while leaving the internal

representations of dLLMs underexplored for decoding guidance.

2.2 Spectral Foundations of Transformers

The representation power of Transformers can be deeply understood through the lens of signal processing, particularly by analyzing their behavior in the frequency domain. While FNet (Lee-Thorp et al., 2022) and its successors (Zhuang et al., 2022; Scribano et al., 2023) demonstrate the efficacy of spectral token-mixing, theoretical evidence suggests that multi-head self-attention inherently functions as a low-pass filter (Wang et al., 2022; Park and Kim, 2022). This is further corroborated by Fourier Transformer (He et al., 2023), which observes that power spectra concentrate in low-frequency bins within deeper layers. FourierAttention (Liu et al., 2025b) also exploits truncated low-frequency Fourier representations for efficient KV-cache modeling. While these spectral properties have been integrated into autoregressive architectures, their potential to guide the decoding process in dLLMs remains unexplored.

3 Method

3.1 Spectral Basis for Hierarchical Generation

Fourier analysis offers a principled view for relating frequency components to the semantic hierarchy. Let $H = [h_0, \dots, h_{L-1}] \in \mathbb{R}^{L \times D}$ denote a hidden-state sequence, where L is the sequence length and D is the hidden dimension. The unitary discrete Fourier transform along the sequence dimension is defined as

$$\hat{H}_k = \frac{1}{\sqrt{L}} \sum_{t=0}^{L-1} H_t e^{-i2\pi kt/L}, \quad k = 0, \dots, L-1. \quad (1)$$

The Fourier basis in Eq. 1 is $e^{-i2\pi kt/L}$, whose angular frequency along the token dimension is $2\pi k/L$. For $k > 0$, its wavelength is

$$\lambda_k = \frac{2\pi}{2\pi k/L} = \frac{L}{k}.$$

Hence, low-frequency components have long wavelengths and remain coherent over large token spans, making them suitable for long-range dependencies. High-frequency components have short wavelengths and vary locally, making them sensitive to details such as variables.

To further connect Fourier frequency with locality, we consider the first-order difference along the

token dimension, $(\Delta H)_t = H_{t+1} - H_t$. Its energy measures the aggregate adjacent-token variation:

$$\|\Delta H\|_F^2 = \sum_{t=0}^{L-1} \|(\Delta H)_t\|_2^2 = \sum_{t=0}^{L-1} \|H_{t+1} - H_t\|_2^2. \quad (2)$$

Thus, the squared Frobenius norm quantifies local fluctuations along the sequence.

Applying the transform in Eq. 1 to the differenced sequence ΔH , and following the derivation in Appendix B, the k -th Fourier coefficient of the differenced signal is

$$\widehat{\Delta H}_k = (e^{i2\pi k/L} - 1) \hat{H}_k. \quad (3)$$

Since the transform in Eq. 1 is unitary, Parseval’s theorem preserves energy between the sequence domain and the frequency domain:

$$\|\Delta H\|_F^2 = \|\widehat{\Delta H}\|_F^2 = \sum_{k=0}^{L-1} \|\widehat{\Delta H}_k\|_2^2.$$

Substituting Eq. 3 yields

$$\begin{aligned} \|\Delta H\|_F^2 &= \sum_{k=0}^{L-1} |e^{i2\pi k/L} - 1|^2 \|\hat{H}_k\|_2^2 \\ &= \sum_{k=0}^{L-1} 4 \sin^2\left(\frac{\pi k}{L}\right) \|\hat{H}_k\|_2^2. \end{aligned} \quad (4)$$

Combining Eq. 2 and Eq. 4, local token-to-token variation can be viewed as a frequency-weighted sum of spectral energy, where the k -th frequency component is weighted by $4 \sin^2(\pi k/L)$.

Since the hidden states are real-valued, the real FFT retains the non-redundant one-sided spectrum $k = 0, \dots, \lfloor L/2 \rfloor$. Over this range, the weight $4 \sin^2(\pi k/L)$ increases with k . Thus, higher rFFT bins contribute more to local variation.

In summary, low-frequency components, with long wavelengths and small local-variation weights, capture global layouts and can provide structural guidance; high-frequency components, with short wavelengths and large local-variation weights, capture local changes for detail refinement.

3.2 Frequential Analysis on dLLM

Inspired by the spectral properties of Transformers discussed in Section 2.2 and the theoretical support for frequency-domain semantics in Section 3.1, we suppose that dLLM hidden states from a forward

pass follow the pattern that low-frequency components encode structural information, while high-frequency components capture local details.

To validate this, we analyze two examples with distinct structural features: a Python script (Figure 2) and a math derivation (Figure 5). From a single forward pass, we extract the final-layer hidden states $\mathbf{H} \in \mathbb{R}^{L \times D}$ (L : sequence length, D : hidden dimension). We compute the low-frequency component $\mathbf{H}'_{\text{low}}$ by applying a binary mask $\mathbf{M} \in \{0, 1\}^W$ to retain the lower half of W frequency components, defining the low-frequency ratio r_{low} to measure its spectral energy proportion:

$$\mathbf{H}'_{\text{low}} = \mathcal{F}_r^{-1}(\mathcal{F}_r(\mathbf{H}) \odot \mathbf{M}), \quad r_{\text{low}} = \frac{\|\mathbf{H}'_{\text{low}}\|_2^2}{\|\mathbf{H}\|_2^2},$$

$$\mathbf{M}_k = \begin{cases} 1, & 0 \leq k < \lfloor W/2 \rfloor, \\ 0, & \text{otherwise.} \end{cases} \quad (5)$$

where $\mathcal{F}_r$ denotes the real-valued Fourier transform, and $\|\cdot\|_2$ denotes the Euclidean norm along the feature dimension. Similarly, we define the high-frequency ratio $r_{\text{high}} = 1 - r_{\text{low}}$.

Figures 2 and 5 highlight tokens with extreme frequency ratios. Across both tasks, structural skeletons (e.g., *if/elif/return*, guiding math text) consistently dominate the low-frequency band, while local details (function names, values, formulas) exhibit high-frequency characteristics.

A part-of-speech analysis in Appendix G further shows that relation-bearing structural markers, e.g., conjunctions, adverbs, and predicates, are more low-frequency dominant, whereas nouns exhibit stronger high-frequency feature, consistent with their role in specifying concrete entities and details.

Besides, to quantitatively validate our hypothesis, we conduct a synonym perturbation experiment. On 100 LongBench documents (Bai et al., 2024), we replace 1%, 5%, and 10% of words with synonyms and compute the frequency-domain cosine similarity of final-layer hidden states. As Table 1 shows, high-frequency similarity drops sharply under stronger perturbations, indicating sensitivity to local details, while low-frequency similarity remains robust, suggesting that the invariant global structure is preserved in low-frequency spectrum.

These analyses provide the first evidence of frequency-domain semantic hierarchy in dLLMs. It suggests a hierarchical generation process: prioritizing low-frequency structural tokens establishes a stable global context that constrains the search

Model	Ratio	Similarity	
		Low-Freq	High-Freq
LLaDA-1.5	1%	0.98	0.88
	5%	0.91	0.62
	10%	0.83	0.43
LLaDA-8B-Instruct	1%	0.98	0.88
	5%	0.92	0.66
	10%	0.86	0.49

Table 1: Similarity of low-frequency and high-frequency components under different replacement ratios.

space for subsequent detail tokens, while decoding details before skeleton may cause error propagation. Such paradigm is supported by diffusion-based image generation (Besnier et al., 2025; Baade et al., 2026), where models form low-frequency global layouts and coarse content before refining details. Inspired by such hierarchy, we propose Fourier-Sampler, which applies a frequency-domain sliding window from low to high frequencies during decoding to enforce structure-to-detail order.

3.3 Translated Fourier Score

The Translated Fourier Score provides a frequency-domain criterion for ranking token positions during decoding. At each decoding step, it retains a fixed-width frequency band shifting uniformly from low-frequency end toward high-frequency end of the spectrum, and scores tokens based on their energy within the retained band.

Assuming a decoding block size B and S steps per block, let $\mathbf{H}(s) \in \mathbb{R}^{B \times D}$ denote the hidden states of the current block at decoding step s . We apply a real-valued Fourier transform along the sequence dimension, filter the spectrum using a step-dependent binary mask $\mathbf{g}(s)$, and then transform back to the original space:

$$\mathbf{H}'(s) = \mathcal{F}_r^{-1}(\mathcal{F}_r(\mathbf{H}(s)) \odot \mathbf{g}(s)), \quad (6)$$

where $\odot$ denotes element-wise multiplication. The filtered hidden state $\mathbf{H}'(s)$ is the reconstructed time-domain representation after retaining only the frequency components selected at step s .

Let W denote the number of frequency components. We set the width of frequency window as

$$w = \max(1, \lfloor \rho W \rfloor), \quad (7)$$

where $\rho \in (0, 1]$ controls the relative bandwidth.

The window translates uniformly along the frequency axis over decoding steps. So the starting

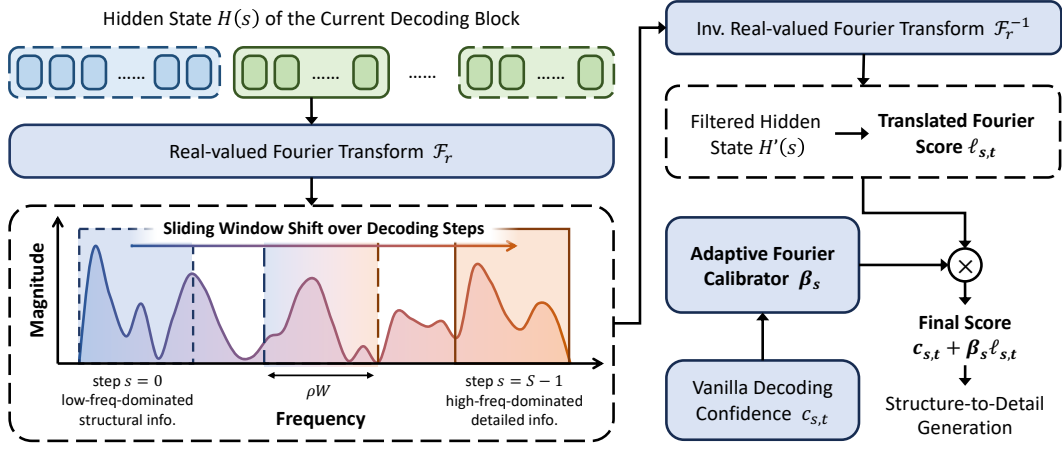


Figure 3: Overview of our FourierSampler. A sliding window in the frequency domain, retaining the low frequency at the beginning and the high frequency at the end based on the decoding step s , guides the dLLM to decode structural content first, then detailed content via Translated Fourier Score and Adaptive Fourier Calibrator.

position of the window at decoding step s is

$$o_s = \left\lfloor \frac{s}{S-1} (W - w) \right\rfloor. \quad (8)$$

The frequency mask $\mathbf{g}(s) \in \{0, 1\}^W$ is defined as

$$\mathbf{g}_p(s) = \begin{cases} 1, & o_s \leq p < o_s + w, \\ 0, & \text{otherwise,} \end{cases} \quad (9)$$

for frequency index $p \in \{0, \dots, W-1\}$.

As s increases, the retained frequency band shifts monotonically from the beginning toward the end of the spectrum, resulting in a step-dependent spectral preference during decoding, favoring low-frequency-dominated tokens in the earlier stage and high-frequency-dominated tokens later.

For a token at position t within the block ($0 \leq t \leq B-1$), we compute its energy under the filtered representation and normalize it within the block to obtain the Translated Fourier Score $\ell_{s,t}$, where ϵ is a small constant added for numerical stability:

$$\ell_{s,t} = \frac{\sum_{d=1}^D \left(\mathbf{H}'_{[t,d]}(s) \right)^2}{\max_{t'} \left(\sum_{d=1}^D \left(\mathbf{H}'_{[t',d]}(s) \right)^2 \right) + \epsilon}. \quad (10)$$

The score $\ell_{s,t}$ measures token t 's relative intensity under the frequency band selected at step s , and is used to guide decoding priority within the block.

3.4 Adaptive Fourier Calibrator

To dynamically adjust the guidance strength of the translated Fourier score $\ell_{s,t}$ in the sampler, we introduce an adaptive weight β_s based on the original

decoding confidence. At decoding step s , let the output prediction distribution be $p_{s,t}$. For the set of masked positions, $\mathcal{M}_s$, we take the maximum probability $q_{s,t} = \max_v p_{s,t}(v)$ in the prediction distribution at each position $t \in \mathcal{M}_s$, and treat its variance over $\mathcal{M}_s$ as the ability of dLLM to distinguish between the decoding priorities of different positions in the current decoding state.

$$\sigma_s^2 = \text{Var} \left(\{q_{s,t}\}_{t \in \mathcal{M}_s} \right). \quad (11)$$

We record the values of σ_s^2 over past decoding steps, compute the percentile P_s of the current variance within the history, and normalize it into $w_s \in (0, 1)$, the cumulative distribution function of the normal distribution. The normalization process is detailed in Appendix C. Finally, the adaptive weight is defined as follows, where $\beta_{\min}$ and $\beta_{\max}$ are the minimum and maximum values of the adaptive weight.

$$\beta_s = \beta_{\min} + (1 - w_s)(\beta_{\max} - \beta_{\min}). \quad (12)$$

With the translated Fourier score and adaptive weight, we add them to the original confidence $c_{s,t}$ at step s for token t to obtain the fusion score:

$$\tilde{c}_{s,t} = c_{s,t} + \beta_s \ell_{s,t}. \quad (13)$$

This design ensures that when the confidence differences between different positions are large, the model's own decoding intention is relatively clear, and the frequency guidance automatically weakens. Conversely, the frequential prior is strengthened, thereby achieving an adaptive decoding scheduler.

	GSM8k		Math		MBPP		HE		CD		Avg.	
<i>Llama3.1-8B-Instruct</i>	81.0		41.6		65.4		54.3		0.0		48.4	
<i>Qwen2.5-7B-Instruct</i>	81.7		49.2		66.9		52.4		4.3		50.9	
<i>LLaDA-1.5</i>	79.8		41.4		42.0		40.9		30.5		46.9	
+ UnCode	81.2	+1.7%	43.0	+3.9%	51.4	+22.2%	39.0	-4.5%	21.1	-30.8%	47.1	+0.5%
+ RWS	80.7	+1.1%	42.0	+1.4%	43.2	+2.8%	39.6	-3.0%	30.1	-1.3%	47.1	+0.4%
+ FourierSampler (ours)	81.8	+2.5%	44.2	+6.8%	50.6	+20.4%	43.3	+6.0%	34.8	+14.1%	50.9	+8.6%
<i>LLaDA-8B-Instruct</i>	78.2		42.2		41.3		39.6		25.4		45.3	
+ UnCode	79.0	+1.0%	40.4	-4.3%	49.8	+20.8%	39.6	0.0%	24.2	-4.6%	46.6	+2.8%
+ RWS	79.6	+1.8%	42.8	+1.4%	42.0	+1.9%	39.0	-1.5%	30.1	+18.5%	46.7	+3.0%
+ FourierSampler (ours)	79.6	+1.8%	45.2	+7.1%	47.9	+16.0%	40.9	+3.1%	28.9	+13.8%	48.5	+7.1%

Table 2: Results on LLaDA-1.5 and LLaDA-8B-Instruct (best values bolded; relative improvement over vanilla confidence decoding in subscripts). FourierSampler achieves the best average performance among different methods, bridging the gap with similarly sized AR models such as Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct, and slightly outperforming them on average with LLaDA-1.5.

	GSM8k		Math		MBPP		HE		CD		Avg.	
<i>Llama3.2-3B-Instruct</i>	69.4		36.6		51.0		26.8		0.0		36.8	
<i>Qwen2.5-3B-Instruct</i>	76.4		39.2		47.5		32.9		10.9		41.4	
<i>SDAR-4B-Chat</i>	86.6		48.2		42.0		57.9		13.3		49.6	
+ RWS	87.4	+1.0%	49.2	+2.1%	39.7	−5.5%	60.4	+4.2%	16.0	+20.6%	50.5	+1.9%
+ FourierSampler (ours)	87.6	+1.2%	50.0	+3.7%	47.5	+13.0%	62.2	+7.4%	16.8	+26.5%	52.8	+6.5%
<i>SDAR-1.7B-Chat</i>	72.9		39.6		35.4		37.8		15.6		40.3	
+ RWS	75.6	+3.6%	41.0	+3.5%	35.8	+1.1%	31.7	−16.1%	18.0	+15.0%	40.4	+0.3%
+ FourierSampler (ours)	73.8	+1.2%	40.0	+1.0%	36.6	+3.3%	43.3	+14.5%	22.7	+45.1%	43.3	+7.4%

Table 3: Results on SDAR-4B-Chat and SDAR-1.7B-Chat. FourierSampler achieves the best average performance, outperforming other methods and some similarly sized AR models including Llama3.2-3B-Instruct and Qwen2.5-3B-Instruct on average.

4 Experiment

4.1 Setup

We conduct experiments on widely used dLLMs, including LLaDA-1.5 (Zhu et al., 2026) and LLaDA-8B-Instruct (Nie et al., 2025). To further demonstrate that our method applies to other dLLM architectures, which adopt block-wise causal attention, we also evaluate our FourierSampler on SDAR-4B-Chat and SDAR-1.7B-Chat (Cheng et al., 2026). The evaluation benchmarks include GSM8K (4-shot) (Cobbe et al., 2021), MATH (4-shot) (Hendrycks et al., 2021), MBPP (3-shot) (Austin et al., 2021b), HumanEval (0-shot) (Chen et al., 2021), and Countdown(0-shot) (Ye et al., 2025a,c). During evaluation, we set the default block size to 64 for all dLLMs and adopt OpenCompass (Contributors, 2023) as the evaluation framework. We use 512 generation steps for GSM8K, MATH, HumanEval, and MBPP, and 128 steps for Countdown.

For LLaDA Series, we use UnCode (Huang et al., 2026) and RWS (Gwak et al., 2025) as baselines.

For SDAR Series, since UnCode lacks the token frequency distribution, we use RWS as the primary baseline. UnCode coefficients follow the original paper. For RWS, we use GRM-Llama3.2-3B (Yang et al., 2024b) as its reward model, recommended by its paper. We also compare against similar-sized AR models from the Llama and Qwen series under the same evaluation settings (Grattafiori et al., 2024; Meta, 2024; Yang et al., 2024a).

4.2 Main Results

For LLaDA-1.5 and LLaDA-8B-Instruct, dLLMs with full bidirectional attention, the results in Table 2 demonstrate that our method consistently and significantly outperforms the baseline across different tasks. We observe improvements up to 7.1% on MATH, 20.4% on MBPP, and 14.1% on Countdown. Moreover, our approach achieves the highest average performance, surpassing other methods.

Notably, FourierSampler enables LLaDA-1.5 to surpass some similarly sized autoregressive models on average, such as Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct, which the original model

and other methods fail to achieve. This highlights its effectiveness in unlocking the non-autoregressive potential of dLLMs.

For dLLMs with block-wise causal attention, including SDAR-4B-Chat and SDAR-1.7B-Chat, our method consistently outperforms the baseline across all benchmarks in Table 3. Specifically, it achieves relative improvements of 3.7% on MATH, 14.5% on HumanEval, and 45.1% on Countdown, surpassing competing methods on average. The above results further show that our method generalizes well across different dLLM architectures.

For broader NLP generation tasks, we further evaluate FourierSampler on LongBench summarization (GovReport, QMSum, and Multi-News) (Bai et al., 2024), MuSR multi-step reasoning (Sprague et al., 2024), and LongBench-Write (Bai et al., 2025) creative writing. As shown in Table 11, FourierSampler consistently improves performance across these diverse tasks; for example, on LongBench-Write, it improves LLaDA-8B-Instruct from 55.7 to 65.9, while achieving an overall relative improvement of 9.4% across all evaluated tasks. These results also show its generalization across different linguistic domains.

4.3 Ablation Study

We conduct component-wise ablations to isolate the effects of the Translated Fourier Score and Adaptive Fourier Calibrator. Fixing β removes adaptive calibration while preserving the Translated Fourier Score. As shown in Table 4, all fixed- β variants still substantially outperform vanilla decoding, e.g., by up to 19.4% on MBPP, showing that the main gains come from frequency-domain guidance. Meanwhile, fixed weights consistently underperform the adaptive weighting strategy, showing that the adaptive calibrator further improves performance by adjusting the guidance strength under different decoding states.

We further study the effect of the frequency-window ratio ρ , which controls the bandwidth of the sliding window in frequency domain. Table 5 shows that overly large windows may blur the distinction between low- and high-frequency bands, weakening the intended low-to-high frequency-guiding preference. Based on the overall performance, we use $\rho = 0.2$ for LLaDA-1.5 and 0.4 for LLaDA-8B-Instruct. Besides, the detailed selection process for β range can be found in Appendix D.

	GSM8k	MBPP	Avg.
LLaDA-1.5	79.8	42.0	60.9
+ Fixed $\beta = 0.4$	81.1	49.8	65.5
+ Fixed $\beta = 0.5$	81.1	47.9	64.5
+ Fixed $\beta = 0.6$	81.2	50.2	65.7
+ Adaptive $\beta \in (0.4, 0.6)$	81.8	50.6	66.2
LLaDA-8B-Instruct	78.2	41.3	59.8
+ Fixed $\beta = 0.4$	78.9	44.8	61.8
+ Fixed $\beta = 0.5$	79.4	46.7	63.0
+ Fixed $\beta = 0.6$	79.2	47.1	63.2
+ Adaptive $\beta \in (0.4, 0.6)$	79.6	47.9	63.7

Table 4: Component ablation study. Primary improvement comes from frequency-domain guidance while adaptive calibration provides additional gains.

	GSM8k	MBPP	Avg.
LLaDA-1.5	79.8	42.0	60.9
+ $\rho = 0.2$	81.8	50.6	66.2
+ $\rho = 0.4$	81.4	42.8	62.1
+ $\rho = 0.6$	81.1	43.0	62.0
LLaDA-8B-Instruct	78.2	41.3	59.8
+ $\rho = 0.2$	80.2	46.7	63.5
+ $\rho = 0.4$	79.6	47.9	63.7
+ $\rho = 0.6$	80.4	45.9	63.1

Table 5: Ablation of the frequency-window ratio ρ .

5 Discussion

5.1 Analysis of Decoding Block Size

To evaluate FourierSampler under different decoding block sizes, we report results in Table 6. FourierSampler achieves larger gains as block size increases, since larger blocks provide more complete and continuous signals for frequency-domain analysis, enabling the Fourier transform to better capture low-frequency global semantics and structural information. In contrast, small blocks fragment the sequence and obscure such patterns. Thus, larger blocks enable early extraction of global structure to guide subsequent fine-grained generation, making the advantages of FourierSampler more evident. Notably, FourierSampler at block size 64 outperforms the baseline at block sizes 32 or 16 on both tasks, suggesting that it mitigates, and sometimes avoids, the degradation caused by larger blocks.

5.2 Decoding Trajectory Comparison with Vanilla Decoding

Case study of decoding trajectory. To verify whether FourierSampler’s generation trajectory exhibits the expected semantically hierarchical decoding pattern, we visualize LLaDA-8B-Instruct’s step-wise decoding in Figure 4. For two decod-

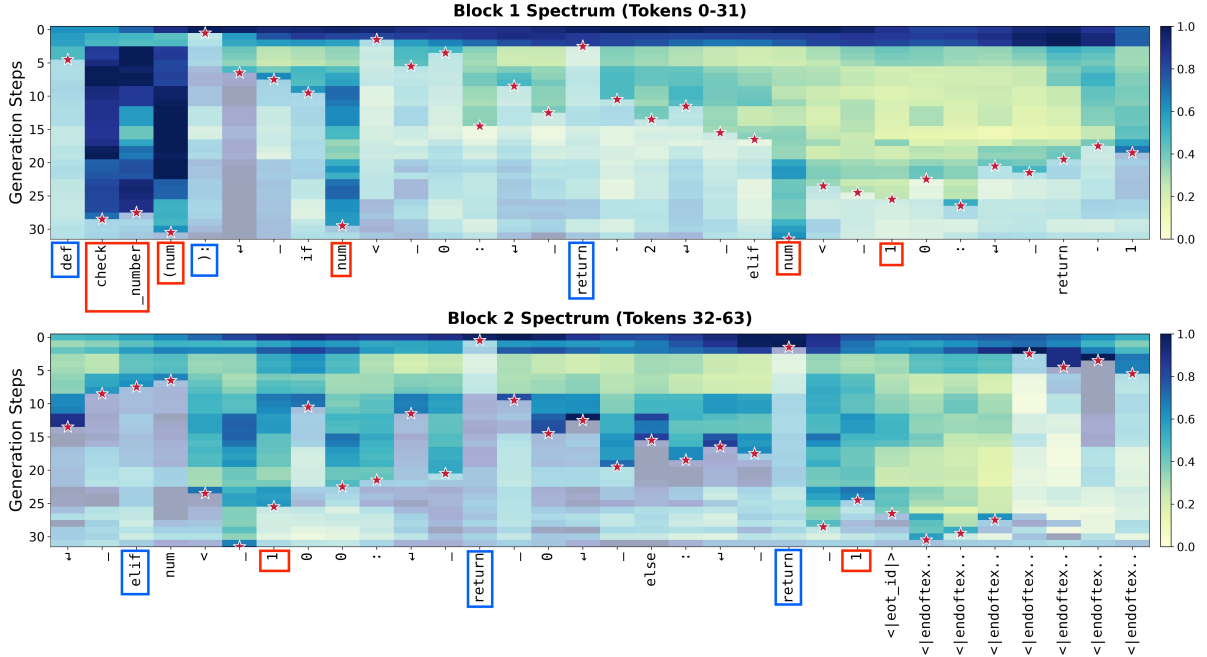


Figure 4: Step-wise generation trajectory for a structured-function prompt on LLaDA-8B-Instruct. The heatmap shows the Translated Fourier Score ($\in (0, 1)$), with red stars marking decoded steps. Early-decoded structure words (blue) and late-filled details (red) validate FourierSampler’s “structure-to-detail” pattern.

	GSM8K			MBPP		
	Vanilla	Ours	$\Delta(\%)$	Vanilla	Ours	$\Delta(\%)$
$B = 16$	80.8	81.1	+0.3%	49.0	48.3	-1.6%
$B = 32$	80.1	80.4	+0.3%	49.8	51.4	+3.1%
$B = 64$	79.8	81.8	+2.5%	42.0	50.6	+20.4%
$B = 128$	68.2	72.6	+9.5%	37.7	44.8	+18.6%

Table 6: Results with different block sizes on LLaDA-1.5. $\Delta(\%)$ denotes relative improvement.

ing blocks, the heatmap displays the Translated Fourier Score $\ell_{s,t}$ at each token position under the frequency band selected for each generation step. Red stars indicate when each token is finalized.

We use a structured-function prompt with conditional branches, as shown in Appendix E. In Block 1, structural keywords such as *def* (pos. 0), *if* (pos. 7), and *return* (pos. 15) are finalized early within the first ten steps, while details such as the function name *check_number* (pos. 1–2), variable *num* (pos. 3, 9, and 21), and values (e.g., pos. 24) are finalized near the end of the block. Block 2 shows the same trend: *elif* (pos. 2) and *return* (pos. 12 and 21) appear early, whereas concrete values such as pos. 23 are determined later.

It’s also shown that tokens are typically finalized with high $\ell_{s,t}$ values. This indicates that the frequency-domain guidance actively shapes the generation trajectory, leading to the observed

structure-to-detail decoding pattern.

Quantitative trajectory analysis. Beyond the qualitative visualization above, we further quantify the trajectory shift induced by FourierSampler on MBPP. Specifically, we record the finalized step of each token under vanilla confidence decoding and FourierSampler, and identify core structural tokens, including *def*, *if*, *elif*, *for*, *return*, etc. Across all decoding blocks, FourierSampler advances the first core structural token by **7.6** decoding steps on average, while the average finalization step of all core structural tokens is advanced by **6.4** steps. The shift is more pronounced on wrong-to-correct samples, where vanilla decoding fails but FourierSampler succeeds: the first structural token is advanced by **12.2** steps on average.

We further define late-structure failures as incorrect baseline samples whose first core structural token is finalized after step 40. They account for 56.3% of baseline errors and 66.7% of wrong-to-correct samples; in these repaired cases, the first structural token is advanced by **14.8** steps on average, and all structural tokens by **10.0** steps. To characterize the relative ordering between structure and detail, we define the structure–detail gap as

$$G = R_{\text{detail}} - R_{\text{struct}}, \quad (14)$$

where R_{struct} and R_{detail} denote the average final-

ized ranks of structural and detail tokens, respectively. FourierSampler consistently enlarges this gap, with the increase in G on wrong-to-correct samples being approximately $1.7\times$ the overall increase, and reaching $2.1\times$ on repaired late-structure failures. These results quantitatively confirm that FourierSampler systematically shifts decoding toward a stronger structure-to-detail order, and that this trajectory shift is particularly pronounced in samples where decoding correctness is improved.

5.3 Compatibility with Parallel Decoding

To further investigate FourierSampler’s compatibility with dynamic parallel decoding strategies, we combine it with Fast-dLLM (Wu et al., 2025b), an efficient dLLM inference framework that enables KV caching and decodes multiple masked tokens simultaneously based on a confidence criterion.

In detail, at each decoding step, we rank masked positions in current block by the fusion score $\tilde{c}_{s,t}$ in descending order and decode them sequentially until encountering a position whose initial confidence $c_{s,t}$ falls below the default confidence threshold adopted in the official Fast-dLLM implementation. This ensures that parallel unmasking remains governed by the structure-to-detail trajectory.

	GSM8K		MBPP	
	Score	TPS	Score	TPS
LLaDA-1.5	79.8	13.5	42.0	21.9
+ Fast-dLLM	79.8	145.9 (10.8 $\times$)	41.4	463.5 (21.2 $\times$)
+ Fast-dLLM + Ours	81.4	139.7 (10.4 $\times$)	51.4	481.6 (22.0 $\times$)
LLaDA-8B-Instruct	78.2	18.2	41.3	21.8
+ Fast-dLLM	77.2	214.8 (11.8 $\times$)	38.5	426.4 (19.6 $\times$)
+ Fast-dLLM + Ours	78.9	212.5 (11.7 $\times$)	46.3	458.2 (21.0 $\times$)

Table 7: Comparison between parallel decoding and its FourierSampler-enhanced variant on LLaDA-1.5 and LLaDA-8B-Instruct.

Table 7 shows that when combined with parallel decoding, our method achieves $10\times$ to $22\times$ speedups compared with vanilla static decoding, while entirely mitigating Fast-dLLM’s typical performance degradation caused by parallel decoding, still outperforming the static baseline.

Notably, on MBPP, it yields higher throughput than Fast-dLLM alone (481.6 vs. 463.5 tokens/s). This additional efficiency gain comes from further

reducing the average decoding steps from 35.5 to 32.5, highlighting its potential to accelerate convergence without sacrificing logical coherence.

Similarly, FourierSampler is also compatible with other uncertainty-based dLLM samplers like EB-Sampler (Ben-Hamu et al., 2025), improving both generation quality and decoding throughput; detailed results are provided in Appendix I.

We further apply FourierSampler to Fast-dLLM v2 (Wu et al., 2025a), a 7B block-diffusion language model natively designed for efficient parallel decoding. As shown in Table 8, FourierSampler consistently improves generation quality on representative tasks, including GSM8K and MBPP, while maintaining comparable decoding throughput. These results demonstrate that our frequency-guided scheduling remains effective on a natively parallel-decoding model without sacrificing its efficiency advantage.

	GSM8K		MBPP	
	Score	TPS	Score	TPS
Fast-dLLM v2	79.4	188.4	59.5	191.8
+ Ours	81.3	188.6	62.3	196.5

Table 8: Compatibility of FourierSampler with Fast-dLLM v2. FourierSampler improves generation quality while preserving the high throughput of the original parallel decoding model.

6 Conclusion

We present the first internal frequency-domain analysis of dLLMs, showing that low-frequency components of their hidden representations primarily capture global structure, while high-frequency components encode local details. Building on this, FourierSampler uses the Translated Fourier Score and Adaptive Fourier Calibrator to dynamically guide structure-to-detail generation.

Experiments on full-bidirectional-attention (LLaDA Series) and block-wise causal-attention (SDAR Series) architectures show consistent gains across various NLP tasks. Further analyses of block size and generation trajectory comparison support FourierSampler’s rationale, while its compatibility with parallel decoding enables a favorable quality-efficiency trade-off. Overall, FourierSampler offers an endogenous and effective path to unlocking dLLMs’ arbitrary-order generation potential.

Limitations

Although FourierSampler achieves larger improvements with larger decoding block sizes, its gains become relatively limited when the block size is very small (e.g., 4 or 8). This is consistent with our analysis in Section 5.1: reliable spectral signals require sufficient continuity along the sequence dimension to capture structural information. When the block size is small, hidden states become fragmented into short segments, making it harder to distinguish structural tokens from detailed ones during decoding. In addition, our spectral guidance is derived from the final-layer hidden states. While this choice is computationally efficient and effective, it does not exploit potential complementary signals across multiple layers. Exploring multi-layer frequency fusion may provide richer structural cues for decoding, which we leave for future work.

Ethics Statement

This research follows established ethical standards and established practices. It does not process sensitive personal data, involve human subjects, or target risky applications. All experiments and analyses are conducted with integrity, transparency, and reliability being ensured.

Acknowledgements

This work was supported by the National Natural Science Foundation of China (No. U24B20181 and 62525602). We sincerely thank all authors for their contributions and efforts throughout this work.

References

- Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. 2021a. [Structured denoising diffusion models in discrete state-spaces](#). In *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc.
- 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.
- Alan Baade, Eric Ryan Chan, Kyle Sargent, Changan Chen, Justin Johnson, Ehsan Adeli, and Li Fei-Fei. 2026. [Latent forcing: Reordering the diffusion trajectory for pixel-space image generation](#). *Preprint*, arXiv:2602.11401.
- Yushi Bai, Xin Lv, Jiajie Zhang, Hongchang Lyu, Jiankai Tang, Zhidian Huang, Zhengxiao Du, Xiao Liu, Aohan Zeng, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2024. [LongBench: A bilingual, multi-task benchmark for long context understanding](#). In *Proceedings of the 62nd Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 3119–3137, Bangkok, Thailand. Association for Computational Linguistics.
- Yushi Bai, Jiajie Zhang, Xin Lv, Linzhi Zheng, Siqi Zhu, Lei Hou, Yuxiao Dong, Jie Tang, and Juanzi Li. 2025. [Longwriter: Unleashing 10,000+ word generation from long context llms](#). In *International Conference on Learning Representations*, volume 2025, pages 36528–36546.
- Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas S Nolte, and Brian Karrer. 2025. [Accelerated sampling from masked diffusion models via entropy bounded unmasking](#). In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 55981–56007. Curran Associates, Inc.
- Lukas Berglund, Meg Tong, Maximilian Kaufmann, Mikita Balesni, Asa Stickland, Tomek Korbak, and Owain Evans. 2024. The reversal curse: LLMs trained on “a is b” fail to learn “b is a”. In *International Conference on Learning Representations*, volume 2024, pages 18623–18642.
- Victor Besnier, Mickael Chen, David Hurych, Eduardo Valle, and MATTHIEU CORD. 2025. [Halton scheduler for masked generative image transformer](#). In *International Conference on Learning Representations*, volume 2025, pages 83027–83047.
- 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.
- Shuang Cheng, Yihan Bian, Dawei Liu, Yuhua Jiang, Yihao Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and Bowen Zhou. 2026. [SDAR: A synergistic diffusion-AutoRegression paradigm for scalable sequence generation](#). In *Findings of the Association for Computational Linguistics: ACL 2026*, pages 22058–22075, San Diego, California, United States. Association for Computational Linguistics.
- Yung-Sung Chuang, Yujia Xie, Hongyin Luo, Yoon Kim, James Glass, and Pengcheng He. 2023. Dola: Decoding by contrasting layers improves factuality in large language models. *arXiv preprint arXiv:2309.03883*.
- 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.
- OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. <https://github.com/open-compass/opencompass>.
- 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 *International Conference on Learning Representations*, volume 2025, pages 5046–5073.
- Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, and 542 others. 2024. [The llama 3 herd of models](#). *Preprint*, arXiv:2407.21783.
- Daehoon Gwak, Minseo Jung, Junwoo Park, Minhoo Park, ChaeHun Park, Junha Hyung, and Jaegul Choo. 2025. [Reward-weighted sampling: Enhancing non-autoregressive characteristics in masked diffusion LLMs](#). In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 34574–34594, Suzhou, China. Association for Computational Linguistics.
- Ziwei He, Meng Yang, Minwei Feng, Jingcheng Yin, Xinbing Wang, Jingwen Leng, and Zhouhan Lin. 2023. [Fourier transformer: Fast long range modeling by removing sequence redundancy with FFT operator](#). In *Findings of the Association for Computational Linguistics: ACL 2023*, pages 8954–8966, Toronto, Canada. Association for Computational Linguistics.
- 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*.
- Pengcheng Huang, Tianming Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Tong Xiao, Zulong Chen, and Maosong Sun. 2026. [Empirical analysis of decoding biases in masked diffusion models](#). In *Proceedings of the 64th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 6853–6876, San Diego, California, United States. Association for Computational Linguistics.
- Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. 2025. [Reinforcing the diffusion chain of lateral thought with diffusion language models](#). In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 152677–152710. Curran Associates, Inc.
- Aaron Hurst, Adam Lerer, Adam P. Goucher, Adam Perelman, Aditya Ramesh, Aidan Clark, AJ Ostrow, Akila Welihinda, Alan Hayes, Alec Radford, Aleksander Madry, Alex Baker-Whitcomb, Alex Beutel, Alex Borzunov, Alex Carney, Alex Chow, Alex Kirillov, Alex Nichol, Alex Paino, and 399 others. 2024. [Gpt-4o system card](#). *Preprint*, arXiv:2410.21276.
- Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. 2025. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*.
- James Lee-Thorp, Joshua Ainslie, Ilya Eckstein, and Santiago Ontañón. 2022. [FNet: Mixing tokens with Fourier transforms](#). In *Proceedings of the 2022 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies*, pages 4296–4313, Seattle, United States. Association for Computational Linguistics.
- Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gomez-Bombarelli. 2025a. [Think while you generate: Discrete diffusion with planned denoising](#). In *International Conference on Learning Representations*, volume 2025, pages 81979–82026.
- Xiaoran Liu, Siyang He, Qiqi Wang, Ruixiao Li, Yuerong Song, Zhigeng Liu, Linlin Li, Qun Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. 2025b. [Beyond homogeneous attention: Memory-efficient llms via fourier-approximated kv cache](#). *Preprint*, arXiv:2506.11886.
- Xiaoran Liu, Kai Lv, Qipeng Guo, Hang Yan, Conghui He, Xipeng Qiu, and Dahua Lin. 2024. Longwanjuan: Towards systematic measurement for long text quality. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5709–5725.
- Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. [Pointer sentinel mixture models](#). *Preprint*, arXiv:1609.07843.
- Meta. 2024. [Llama 3.2: Revolutionizing edge ai and vision with open, customizable models](#).
- Jinjie Ni, Qian Liu, Chao Du, Longxu Dou, Hang Yan, Zili Wang, Tianyu Pang, and Michael Qizhe Shieh. 2025. Training optimal large diffusion language models. *arXiv preprint arXiv:2510.03280*.
- Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan LI. 2025. [Large language diffusion models](#). In *Advances in Neural Information Processing Systems*, volume 38, Main Conference, pages 50608–50646. Curran Associates, Inc.
- OpenAI. 2023. Gpt-4 technical report. Technical report.

- Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2025. [Your absorbing discrete diffusion secretly models the conditional distributions of clean data](#). In *International Conference on Learning Representations*, volume 2025, pages 64972–65009.
- Namuk Park and Songkuk Kim. 2022. How do vision transformers work? *arXiv preprint arXiv:2202.06709*.
- Dongyu Ru, Lin Qiu, Xipeng Qiu, Yue Zhang, and Zheng Zhang. 2023. Distributed marker representation for ambiguous discourse markers and entangled relations. *arXiv preprint arXiv:2306.10658*.
- Subham Sekhar 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](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 130136–130184. Curran Associates, Inc.
- Carmelo Scribano, Giorgia Franchini, Marco Prato, and Marko Bertogna. 2023. Dct-former: Efficient self-attention with discrete cosine transform. *Journal of Scientific Computing*, 94(3):67.
- Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. 2024. [Simplified and generalized masked diffusion for discrete data](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 103131–103167. Curran Associates, Inc.
- Oscar Skean, Md Rifat Arefin, Dan Zhao, Niket Patel, Jalal Naghiyev, Yann LeCun, and Ravid Shwartz-Ziv. 2025. Layer by layer: Uncovering hidden representations in language models. *arXiv preprint arXiv:2502.02013*.
- Zayne Sprague, Xi Ye, Kaj Bostrom, Swarat Chaudhuri, and Greg Durrett. 2024. [Musr: Testing the limits of chain-of-thought with multistep soft reasoning](#). In *International Conference on Learning Representations*, volume 2024, pages 14670–14728.
- Peihao Wang, Wenqing Zheng, Tianlong Chen, and Zhangyang Wang. 2022. Anti-oversmoothing in deep vision transformers via the fourier domain analysis: From theory to practice. *arXiv preprint arXiv:2203.05962*.
- Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. 2025a. [Fast-dllm v2: Efficient block-diffusion llm](#). *Preprint*, arXiv:2509.26328.
- Chengyue Wu, Hao Zhang, Shuchen Xue, Zhijian Liu, Shizhe Diao, Ligeng Zhu, Ping Luo, Song Han, and Enze Xie. 2025b. Fast-dllm: Training-free acceleration of diffusion llm by enabling kv cache and parallel decoding. *arXiv preprint arXiv:2505.22618*.
- An Yang, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chengyuan Li, Dayiheng Liu, Fei Huang, Haoran Wei, Huan Lin, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Yang, Jiayi Yang, Jingren Zhou, Junyang Lin, Kai Dang, and 22 others. 2024a. [Qwen2.5 technical report](#). *arXiv preprint arXiv:2412.15115*.
- Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024b. Regularizing hidden states enables learning generalizable reward model for llms. In *Advances in Neural Information Processing Systems*.
- Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025a. [Beyond autoregression: Discrete diffusion for complex reasoning and planning](#). In *International Conference on Learning Representations*, volume 2025, pages 77875–77898.
- Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and Lingpeng Kong. 2024. [Diffusion of thought: Chain-of-thought reasoning in diffusion language models](#). In *Advances in Neural Information Processing Systems*, volume 37, pages 105345–105374. Curran Associates, Inc.
- Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025b. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*.
- Xi Ye, Fangcong Yin, Yinghui He, Joie Zhang, Howard Yen, Tianyu Gao, Greg Durrett, and Danqi Chen. 2025c. Longproc: Benchmarking long-context language models on long procedural generation. *arXiv preprint arXiv:2501.05414*.
- Hongxiang Zhang, Hao Chen, Muhao Chen, and Tianyi Zhang. 2025. Active layer-contrastive decoding reduces hallucination in large language model generation. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pages 3028–3046.
- 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, San Diego, California, United States. Association for Computational Linguistics.
- Yimeng Zhuang, Jing Zhang, and Mei Tu. 2022. Long-range sequence modeling with predictable sparse attention. In *Proceedings of the 60th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 234–243.
- Andy Zou, Long Phan, Sarah Chen, James Campbell, Phillip Guo, Richard Ren, Alexander Pan, Xuwang Yin, Mantas Mazeika, Ann-Kathrin Dombrowski,

Shashwat Goel, Nathaniel Li, Michael J. Byun, Zifan Wang, Alex Mallen, Steven Basart, Sanmi Koyejo, Dawn Song, Matt Fredrikson, and 2 others. 2025. [Representation engineering: A top-down approach to ai transparency](#). *Preprint*, arXiv:2310.01405.

A Parameter Settings and Robustness Analysis

FourierSampler mainly involves two hyperparameters: the window ratio ρ , which controls the bandwidth of the translated frequency window, and the adaptive weight range $(\beta_{\min}, \beta_{\max})$, which determines the dynamic strength of frequency guidance. In our experiments, for the LLaDA series, we set the adaptive weight range to $(0.4, 0.6)$, with $\rho = 0.2$ for LLaDA-1.5 and $\rho = 0.4$ for LLaDA-8B-Instruct, while for SDAR-4B-Chat, we set the adaptive weight range to $(0, 0.2)$ and use $\rho = 0.5$.

These hyperparameters are selected through a lightweight calibration process. In practice, we evaluate a small set of candidate values on a few representative tasks and choose the range that yields better overall performance. This process is not intended to perform an exhaustive grid search, but rather to identify a reasonable operating range for each model. The detailed hyperparameter selection process is provided in Section 4.3 and Appendix D.

Importantly, FourierSampler remains consistently better than vanilla decoding under different parameter choices, as shown in Tables 5, 9 and 10. This suggests that the performance gain primarily comes from frequency-guided structure-to-detail decoding, rather than from complex hyperparameter tuning. Nevertheless, different models may exhibit different confidence distributions, decoding dynamics, and spectral energy profiles. Therefore, a small amount of model-specific calibration can further improve performance and better unlock the non-autoregressive generation potential of each model.

B Additional Theoretical Analysis: Derivation of Eq. 3

Let $\omega_k = 2\pi k/L$. Applying the same DFT as Eq. 1 to the differenced sequence ΔH , we have

$$\widehat{\Delta H}_k = \frac{1}{\sqrt{L}} \sum_{t=0}^{L-1} (\Delta H)_t e^{-i\omega_k t}.$$

Since $(\Delta H)_t = H_{t+1} - H_t$, this becomes

$$\widehat{\Delta H}_k = \frac{1}{\sqrt{L}} \sum_{t=0}^{L-1} (H_{t+1} - H_t) e^{-i\omega_k t}.$$

By linearity of the summation,

$$\widehat{\Delta H}_k = \frac{1}{\sqrt{L}} \sum_{t=0}^{L-1} H_{t+1} e^{-i\omega_k t} - \frac{1}{\sqrt{L}} \sum_{t=0}^{L-1} H_t e^{-i\omega_k t}.$$

The second term is exactly $\widehat{H}_k$. For the first term, let $u = t + 1$, so $t = u - 1$. Then

$$\frac{1}{\sqrt{L}} \sum_{t=0}^{L-1} H_{t+1} e^{-i\omega_k t} = \frac{1}{\sqrt{L}} \sum_{u=1}^L H_u e^{-i\omega_k (u-1)}.$$

Rearranging the exponential term gives

$$\frac{1}{\sqrt{L}} \sum_{u=1}^L H_u e^{-i\omega_k (u-1)} = e^{i\omega_k} \frac{1}{\sqrt{L}} \sum_{u=1}^L H_u e^{-i\omega_k u}.$$

Under the standard periodic convention of the DFT, $H_L = H_0$, and since $e^{-i\omega_k L} = e^{-i2\pi k} = 1$, we have

$$\sum_{u=1}^L H_u e^{-i\omega_k u} = \sum_{u=0}^{L-1} H_u e^{-i\omega_k u}.$$

Therefore,

$$\frac{1}{\sqrt{L}} \sum_{t=0}^{L-1} H_{t+1} e^{-i\omega_k t} = e^{i\omega_k} \widehat{H}_k.$$

Substituting the two terms back, we obtain

$$\widehat{\Delta H}_k = e^{i\omega_k} \widehat{H}_k - \widehat{H}_k = (e^{i\omega_k} - 1) \widehat{H}_k.$$

Finally, substituting $\omega_k = 2\pi k/L$ yields

$$\widehat{\Delta H}_k = (e^{i2\pi k/L} - 1) \widehat{H}_k.$$

C Details of Method

As we have presented in Section 3, we introduce an adaptive weight β_s based on the original decoding confidence. At the decoding step s , it is calculated based on the variance σ_s^2 over the maximum probability $q_{s,t}$ in the prediction distribution at each masked position $t \in \mathcal{M}_s$. We record the σ_s^2 over the past 20 decoding steps, compute the percentile P_s of the current variance within the history, and linearly map it to the effective support interval $[-3, 3]$ of the standard normal distribution, then obtain a smooth value $w_s \in (0, 1)$ through the cumulative distribution function of the normal distribution, $F(x) = \frac{1}{2} * \left(1 + \operatorname{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$. The pseudocode of the whole process is shown in Alg 1.

Algorithm 1 Compute Adaptive Weight β_s

```

1: procedure COMPUTEADAPTIVEWEIGHT
2:   if not  $\mathcal{M}_s.any()$  then
3:     return  $\beta_{\min}$ 
4:    $\sigma_s^2 = \text{Var}(\{q_{s,t}\}_{t \in \mathcal{M}_s})$ 
5:    $v\_list.append(\sigma_s^2)$ 
6:   if  $\text{len}(v\_list) > 20$  then
7:      $v\_list.pop(0)$ 
8:    $v\_list\_ = [\_ \text{ for } \_ \text{ in } v\_list \text{ if } \_ < \sigma_s^2]$ 
9:    $P_s = \frac{\text{len}(v\_list\_)}{\text{len}(v\_list)}$ 
10:   $z_s = (P_s - \frac{1}{2}) * 6$ 
11:   $w_s = \frac{1}{2} * \left(1 + \text{erf}\left(\frac{z_s}{\sqrt{2}}\right)\right)$ 
12:   $\beta_s = \beta_{\min} + (1 - w_s) * (\beta_{\max} - \beta_{\min})$ 
13:  return  $\beta_s, v\_list$ 

```

D Selection of the Adaptive Weight Range

In this section, we provide additional experiments to justify the selection of the adaptive weight range $\beta_{\min}$ and $\beta_{\max}$ used in the Adaptive Fourier Calibrator.

In our method, the weight β_s dynamically adjusts the influence of the Translated Fourier Score during decoding and varies within the interval $[\beta_{\min}, \beta_{\max}]$ (Eq.13). Therefore, determining appropriate values for $\beta_{\min}$ and $\beta_{\max}$ requires first identifying a reasonable fluctuation range for β .

To this end, we first conduct experiments using fixed β values during decoding. Since different models may exhibit different intrinsic confidence distributions, we expect the appropriate guidance strength to be primarily model-specific while remaining transferable across tasks. To verify this, we perform lightweight offline tuning on a held-out dataset and evaluate the selected setting on other downstream tasks. Specifically, for LLaDA-1.5, we use Countdown, which emphasizes global reasoning and planning, as the tuning set, evaluate three representative fixed weights $\beta \in \{0.2, 0.5, 0.8\}$, and then evaluate the same settings on GSM8K and MATH. As shown in Table 9, $\beta = 0.5$ achieves the best performance on Countdown, suggesting that a suitable central value for this model is around 0.5.

We then directly evaluate the same fixed weights on GSM8K and MATH without further tuning. As shown in Table 10, $\beta = 0.5$, selected on the held-out tuning set, also achieves the best average performance across the two downstream tasks. This confirms that lightweight held-out tuning can iden-

	Countdown
LLaDA-1.5	30.5
+ FourierSampler (Fixed $\beta = 0.2$)	31.6
+ FourierSampler (Fixed $\beta = 0.5$)	33.2
+ FourierSampler (Fixed $\beta = 0.8$)	32.8

Table 9: Performance comparison under different fixed β values on Countdown using LLaDA-1.5.

tify a suitable β range that transfers well across tasks. Based on these results, we set $\beta_{\min} = 0.4$ and $\beta_{\max} = 0.6$, forming a symmetric range centered at 0.5.

	GSM8K	MATH	Avg.
LLaDA-1.5	79.9	41.4	60.7
+ Ours (Fixed $\beta = 0.2$)	80.9	42.0	61.5
+ Ours (Fixed $\beta = 0.5$)	81.1	44.6	62.9
+ Ours (Fixed $\beta = 0.8$)	80.7	43.2	62.0

Table 10: Performance comparison under different fixed β values on GSM8K and MATH using LLaDA-1.5.

We further verify the effectiveness of the adaptive mechanism. Dynamically adjusting β_s within $[0.4, 0.6]$ according to the model’s positional certainty yields better performance than using a fixed β , as also shown in Table 4. This result demonstrates that adaptive weighting better balances frequency-domain guidance with the model’s intrinsic decoding confidence.

E Details of Sample Analysis

Beyond code generation tasks, we also conducted a frequency-domain analysis on a mathematical derivation passage concerning the **difference of squares formula**. As shown in Figure 5, narrative text appears as low-frequency components, while specific formulas and variables emerge as high frequency. This observation further corroborates our experimental findings in Section 3.2.

For the generation-order analysis in Section 5.2, we use the following structured Python-function prompt: *Write a python function that returns -2 when a number is below 0, -1 when it is below 10, 0 when it is below 100, and 1 otherwise.* The prompt contains multiple conditional branches, making it suitable for visualizing the structure-to-detail generation trajectory.

F Evaluation on Additional NLP Tasks

While the distinction between structure and detail is most explicit in formal languages such as code

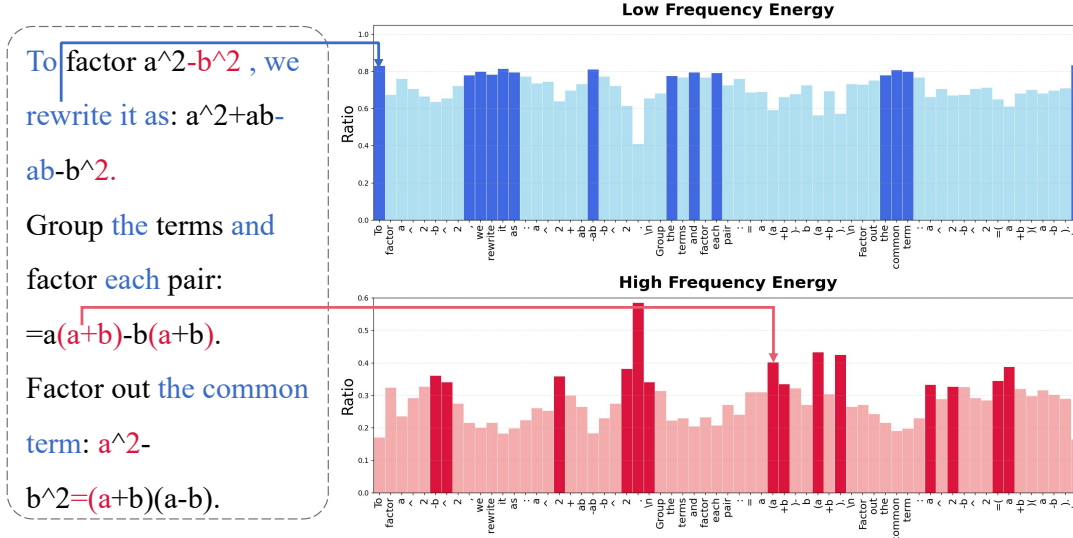


Figure 5: Visualization of the correspondence between frequency-domain analysis and textual information.

and mathematics, we hypothesize that long-form natural language generation also exhibits a hierarchical organization. In many NLP generation tasks, high-level discourse structure—such as narrative flow, reasoning chains, or document organization—forms the global scaffold of the output, while specific entities, descriptions, and stylistic expressions correspond to finer-grained details.

Based on this observation, we further evaluate FourierSampler on several long-form NLP generation tasks. Specifically, we evaluate on three benchmarks: LongBench summarization tasks (Bai et al., 2024), including GovReport, QMSum, and Multi-News; MuSR (Sprague et al., 2024), which contains murder mysteries, object placement, and team allocation subtasks for multi-hop deductive reasoning; and LongBench-Write (Bai et al., 2025), an open-ended creative writing benchmark where we sample prompts with generation lengths under 500 tokens and use GPT-4o (Hurst et al., 2024) as an automatic evaluator.

Results in Table 11 show that FourierSampler achieves the best average performance across both models. Compared with the vanilla decoding strategy, our method yields relative improvements of **9.4%** for LLaDA-8B-Instruct and **6.7%** for LLaDA-1.5 across these open-domain tasks.

These results provide additional evidence that the structure-to-detail generation pattern is not limited to highly structured domains such as code or mathematics. Instead, it appears to be a more general characteristic of how diffusion language models represent hierarchical information across di-

verse linguistic tasks, allowing FourierSampler to adaptively optimize generation trajectories across different scenarios.

G Analysis of Part-of-Speech

To investigate which words in natural language likely correspond to high-frequency components and which favor low-frequency ones, we conducted a detailed statistical analysis on the WikiText-103 dataset (Merity et al., 2016). For each paragraph, we extract the final-layer hidden state sequence $\mathbf{H} \in \mathbb{R}^{L \times D}$, apply the spectral filtering mechanism, and calculate the low-frequency ratio r_{low} as defined in Equation 5. Based on this, tokens with $r_{\text{low}} > 0.5$ (indicating low-frequency dominance) are classified into the *low group*, while others are assigned to the *high group*. Figure 6 illustrates the distribution ratios of different parts of speech across these two groups.

Function words and connectives, which are primarily responsible for constructing sentence logic and structural scaffolding (Ru et al., 2023; Liu et al., 2024), occupy a larger proportion of low-frequency group. In particular, conjunctions, prepositions, and adverbs (e.g., *firstly*) exhibit the highest ratios of low-frequency dominance. In addition, verbs, as the core predicates of sentences, also show a strong low-frequency tendency. This observation explains why control-flow tokens such as *def* and *return* are preferentially generated in the code-generation heatmap presented in Figure 4.

In contrast, nouns exhibit the strongest high-frequency characteristics. Typically referring to

Model	Summary			MuSR			LongBench-Write	Avg.
	gov	qm	multi	murder	object	team		
LLaDA-8B-Instruct	32.0	5.2	26.4	63.6	53.5	56.0	55.7	41.8
+ UnCode	32.4	5.3	27.0	62.0	53.9	44.8	61.1	40.9
+ RWS	32.3	9.4	27.2	64.0	54.7	53.6	64.2	43.6
+ FourierSampler (Ours)	32.4	10.5	27.3	68.8	57.8	57.2	65.9	45.7
LLaDA-1.5	33.2	9.1	27.1	63.6	55.9	56.0	70.4	45.1
+ UnCode	32.8	20.7	26.8	66.0	52.7	51.2	71.0	45.9
+ RWS	32.7	14.2	27.2	67.2	57.0	53.6	64.5	45.2
+ FourierSampler (Ours)	33.6	14.4	27.4	70.8	57.8	60.8	71.9	48.1

Table 11: Evaluation results on extensive NLP generation benchmarks including LongBench summarization, MuSR reasoning tasks, and LongBench-Write creative writing.

	GSM8k		Math		MBPP		HE		CD		Avg.	
Dream-7B-Chat	76.7		42.2		63.4		57.3		24.2		52.8	
+ RWS	76.4	−0.3%	41.6	−1.4%	63.0	−0.6%	56.7	−1.1%	23.8	−1.6%	52.3	−0.8%
+ UnCode	78.8	+2.8%	44.6	+5.7%	65.8	+3.7%	53.1	−7.4%	26.2	+8.1%	53.7	+1.7%
+ FourierSampler (ours)	78.8	+2.8%	43.2	+2.4%	67.3	+6.1%	59.2	+3.2%	32.0	+32.2%	56.1	+6.3%

Table 12: Results on Dream-7B-Chat. Best values are bolded, and relative improvements over vanilla confidence decoding are shown in subscripts. FourierSampler achieves the best average performance, demonstrating its extensibility to other dLLM families.

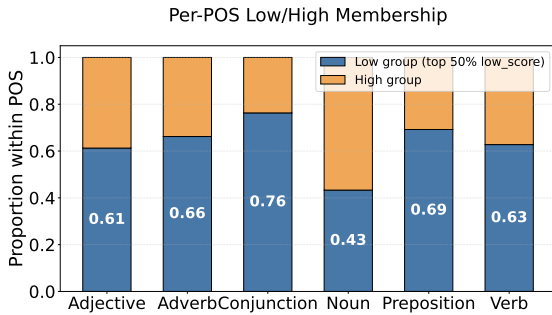


Figure 6: Low and high-frequency features of different parts of speech in the frequency domain. The blue bars represent the proportion of tokens classified into the *low* group, while the orange bars represent the *high* group.

concrete entities and values (e.g., *fib*, *0*), they fill the syntactic skeleton. This aligns with our distinction between *framework* and *detail* words. FourierSampler leverages this property to transform implicit hierarchies into explicit generation planning.

H Extension to Other dLLM Families

We further adapt FourierSampler to other dLLM families beyond LLaDA and SDAR, such as Dream-7B-Chat. As shown in Table 12, FourierSampler consistently outperforms vanilla decoding across different tasks and achieves the best average performance among all compared methods. In particular, it obtains relative improvements of 6.1%

on MBPP and 32.2% on Countdown, with an average improvement of 6.3%. These results show that FourierSampler can be effectively extended to different dLLM families, demonstrating its good adaptability and generality across different dLLM architectures.

I Compatibility with Other Uncertainty-Based Samplers

We further investigate whether FourierSampler can be combined with other uncertainty-based parallel decoding strategies such as EB-Sampler (Ben-Hamu et al., 2025), which dynamically determines the number of tokens to unmask at each decoding step by bounding the approximate error with an entropy tolerance γ . Following the experimental settings explored in the original paper, we set $\gamma = 0.01$ and incorporate our frequency-domain guidance into its token selection process, using the same integration strategy as in Section 5.3.

As shown in Table 13, incorporating FourierSampler into EB-Sampler yields consistent gains across different tasks. Overall, the combined method improves the average score by 10.0% relative to vanilla decoding and by 9.6% relative to EB-Sampler alone, showing that frequency-domain trajectory guidance provides complementary information beyond such parallel decoding strategies.

We additionally report decoding throughput in

Method	GSM8K	MATH	MBPP	HE	CD	Avg.
LLaDA-1.5	79.8	41.4	42.0	40.9	30.5	46.9
+ EB-Sampler	80.8	43.8	43.2	36.6	30.9	47.1
+ EB-Sampler + FourierSampler	82.2	44.0	50.6	41.5	39.8	51.6

Table 13: Compatibility with EB-Sampler on LLaDA-1.5. FourierSampler provides complementary frequency-domain guidance to uncertainty-based parallel decoding.

Table 14. While EB-Sampler substantially accelerates generation through dynamic parallel unmasking, incorporating FourierSampler preserves and even further improves its throughput. These results suggest that FourierSampler is largely orthogonal to efficient uncertainty-based parallel decoding and can be naturally integrated with it, not only improving downstream task performance but also showing the potential to further enhance inference efficiency.

Method	GSM8K TPS	MBPP TPS
LLaDA-1.5	13.5	21.9
+ EB-Sampler	55.4	169.8
+ EB-Sampler + Ours	72.9	237.5

Table 14: Decoding throughput when combining FourierSampler with EB-Sampler on LLaDA-1.5.

J Impact on Decoding Throughput

To evaluate the computational overhead introduced by our method, we measure the decoding throughput of different sampling strategies. Using LLaDA-1.5 as the test model, we conduct experiments on GSM8K with 512 decoding steps and a block size of 64, and report the average generation throughput in tokens per second. All these experiments are conducted on NVIDIA H200 GPU.

Method	Throughput (tokens/s)
LLaDA-1.5	13.5
+ FourierSampler	13.3
+ RWS	9.5
+ UnCode	13.4

Table 15: Decoding throughput comparison on GSM8K with 512 decoding steps and block size 64.

As shown in Table 15, although FourierSampler introduces additional computational overhead due to the Fourier transform required at each decoding step, the decoding efficiency is only slightly lower than that of the original model (a decrease of merely 0.2 tokens/s). This is largely because these Fourier operations are implemented with

highly optimized cuFFT kernels on GPU, making the additional spectral computation lightweight in practice. Besides, FourierSampler achieves significantly higher throughput than RWS and maintains decoding efficiency comparable to UnCode. Meanwhile, FourierSampler achieves notably higher average performance across tasks than both the original model and other methods. These results demonstrate that our method achieves a favorable trade-off between efficiency and effectiveness.

K Analysis of Layer Selection for Frequency-Domain Guidance

In the main text, FourierSampler is applied exclusively to the final-layer hidden states. This design choice is grounded in the premise that the final layer provides the most mature semantic space, as it is directly coupled with the model’s output distribution. Existing inference-time intervention methods in AR, such as DoLa (Chuang et al., 2023), Representation Engineering (Zou et al., 2025) and Active Layer-Contrastive Decoding (Zhang et al., 2025), similarly utilize the final layer as the primary anchor for behavioral intervention. Furthermore, extracting frequency signals from a single layer maintains high computational efficiency by avoiding the overhead of cross-layer propagation.

To empirically validate the effectiveness of the final layer and explore the potential utility of intermediate representations, we conduct an ablation study regarding layer selection. Specifically, the inputs to the frequency-domain sliding-window filtering and the Translated Fourier Score computation are replaced with the input hidden states of specified intermediate Transformer blocks. Table 16 presents the evaluation results on LLaDA-1.5 across layers 0, 8, 16, 24, 31, and the final layer (Layer 32) on GSM8K and MBPP.

As shown in Table 16, utilizing the final-layer hidden state achieves the best average generation performance across metrics, aligning with the theoretical intuition of its direct mapping to the output space. Consequently, the final layer serves as a

Layer Index	GSM8K	MBPP	Avg.
32 (Ours)	81.8	50.6	66.2
31	80.2	49.8	65.0
24	81.4	45.9	63.6
16	82.2	46.3	64.2
8	82.0	47.1	64.5
0	81.0	48.6	64.8

Table 16: Performance comparison of FourierSampler applied to different hidden state layers on LLaDA-1.5.

simple yet robust foundation for structure-to-detail decoding.

The results also demonstrate that applying frequency-domain guidance to shallower or intermediate layers (e.g., layers 16) still yields performance gains compared to the baseline. This observation is consistent with representation probing literature (Skean et al., 2025), which indicates that intermediate layers capture a rich hierarchy of linguistic and structural abstractions. Therefore, exploring multi-layer frequency-domain fusion—such as cross-layer band consistency or adaptive layer-weighted aggregation—represents a viable direction for future work.