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

Anonymous ACL submission

Abstract

Diffusion Large Language Models, or dLLM, have become a widely discussed topic in NLP recently due to their arbitrary-order decoding feature and their potential to capture more complex semantics and achieve generation from structure to detail. Despite this, existing work finds that dLLMs demonstrate positional bias or fail to fully unlock the potential of non-autoregressive generation, which has sparked research on dLLM decoding strategies. Current decoding strategies primarily rely on external signal intervention to optimize dLLM decoding, lacking sufficient exploration of the dLLM's internal characteristics. Inspired by signal processing theory and its applications in NLP, we first introduce frequency-domain analysis into dLLM and propose FourierSampler, which leverages a frequency-domain sliding window on hidden states to guide dLLMs to first decode structural content dominated by low-frequency signals, then decode detailed content dominated by high-frequency signals. We conduct validation experiments on LLaDA and SDAR, and find that FourierSampler can consistently achieve improvements in code and math tasks, surpassing existing methods as well as auto-regressive models of the same size.

1 Introduction

Diffusion Large Language Models (dLLMs) (Sahoo et al., 2024; Ou et al., 2024; Shi et al., 2024) have become a hot topic in NLP. Models such as LLaDA (Nie et al., 2025), Dream (Ye et al., 2025a), Mercury (Inception, 2025), Gemini Diffusion (Gemini, 2025), and SDAR (Cheng et al., 2025) confirm the scalability of this paradigm (Nie et al., 2024; Gong et al., 2024; Ni et al., 2025) and fuel intensive follow-up work on long-context modeling (Liu et al., 2025b; He et al., 2025), inference efficiency (Wu et al., 2025b,a; Song et al., 2025), multimodal extensions (You et al., 2025; Yang et al., 2025), and post-training strategies (Zhu et al., 2025a; Zhao et al., 2025; Wang et al., 2025; Zhu et al., 2025b). Unlike conventional autoregressive LLMs (AR) decoding left-to-right (OpenAI, 2023; Sun et al., 2024; Cai et al., 2024; Dubey et al., 2024), dLLMs take an arbitrary-order generation strategy, which has been verified in solving the reversal curse (Berglund et al., 2023), and maintaining coherence across contexts (Bachmann and Nagarajan, 2024; Ye et al., 2024a; Li et al., 2022), and is expected to capture richer semantics and support structure-to-detail generation (Li et al., 2025; Yu et al., 2025).

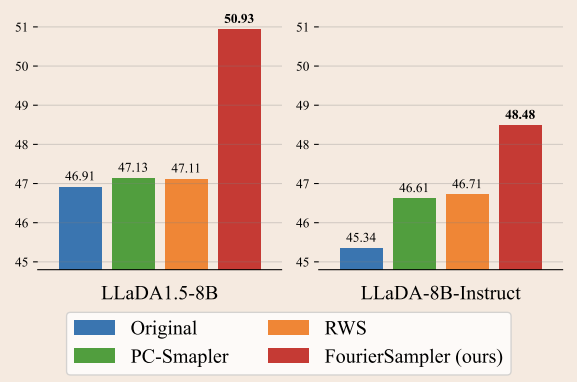

Figure 1: The average score in different tasks of FourierSampler compared with other decoding strategies in LLaDA (Nie et al., 2025; Zhu et al., 2025a).

Despite this, existing work finds that dLLMs demonstrate a tendency toward left-to-right decoding, and forcing left-to-right decoding can even achieve results superior to the original confidence-based decoding in some cases (Gwak et al., 2025), which has also sparked exploration into the decoding planning or sampling strategies of dLLMs. Existing work primarily focuses on the design of external intervention signals, such as rule-based biases for specific positions or tokens in the vocabulary (Huang et al., 2025a), or model-based external reward weighting (Gwak et al., 2025). None of these approaches delves deeply into the inherent

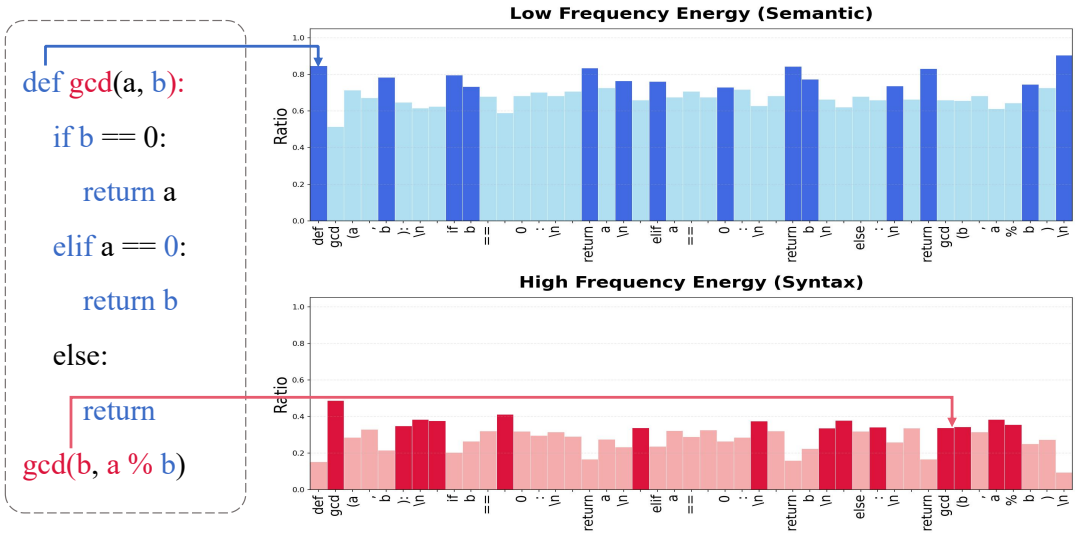

Figure 2: Visualization of the correspondence between frequency-domain analysis and textual information. In the hidden states after a forward pass, tokens dominated by low-frequency signals correspond to structural information like *if* and *elif*, while tokens dominated by high-frequency signals correspond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

characteristics of dLLMs to unlock the arbitrary-order decoding potential of dLLMs. This raises a question: can the dLLM itself have the ability to plan better decoding strategies before generation?

Inspired by frequency-domain signal processing theory (Cooley and Tukey, 1965; Sorensen et al., 2003) and its applications in NLP (Lee-Thorp et al., 2022; He et al., 2023; Liu et al., 2025a), we propose **FourierSampler** which leverages frequency-domain filtering of the hidden states to guide the model to first decode structural content dominated by low-frequency signals, then decode detailed content dominated by high-frequency signals through the signal strength of different frequencies of tokens. Specifically, we apply the Fourier Transform to tokens within decoding blocks along the sequence dimension, design the **Translated Fourier Score**, examining different frequency bands at different decoding steps for each token, and implement an **Adaptive Fourier Calibrator**, dynamically adjusting guidance strength based on the fluctuations of decoding confidence. As shown in Figure 1, our method can statistically outperform the original model and other work relying on external signal-guided decoding on various math and code tasks. Our contribution can be summarized as follows.

- We conduct the first frequency analysis in dLLMs that the low-frequency components of hidden states in the temporal dimension correspond to structural information in the output, while the high-frequency components correspond to detailed information, which provides internal guidance for dLLM decoding.

- We propose **FourierSampler**, our dLLM decoding sampling scheme, which guides the model to achieve a structure-to-detail decoding with the Translated Fourier Score, and balances the guidance with the original confidence by an Adaptive Fourier Calibrator.

- We conduct validation experiments on two types of dLLMs, LLaDA and SDAR, and find that we can stably achieve improvements in code and math tasks, surpassing existing decoding strategies as well as autoregressive models with similar sizes and providing new insights for an in-depth understanding of dLLM decoding enhancement.

2 Related Work

2.1 Decoding Strategy for dLLMs

Non-autoregressive dLLMs can exhibit a tendency towards autoregressive-like decoding behavior for positional bias (Gwak et al., 2025). Consequently, recent work focuses on optimizing token unmasking orders to enhance generation planning. Mainstream models like LLaDA (Nie et al., 2025) and SDAR (Cheng et al., 2025) adopt confidence-based unmasking. Variants prioritize unmasking based on maximum probability, entropy (Ben-Hamu et al.,

2025), or confidence gaps (Kim et al., 2025), with random sampling baselines (Austin et al., 2021a).

To enhance generation performance, advanced methods introduce external interventions. For example, PC-Sampler (Huang et al., 2025a) uses rules-based biases for specific positions, while RWS (Gwak et al., 2025) employs reward models to enhance coherence. Alternatively, training-based approaches like DOT (Ye et al., 2024b), DDPD (Liu et al., 2024a), and DCoLT (Huang et al., 2025b) optimize generation trajectories via post-training or reinforcement learning. However, these methods rely on complex external signals or costly training. They overlook the potential of mining dLLM internal representations for effective decoding guidance.

2.2 Frequency Analysis in Transformers

Frequency analysis, originated from signal processing (Cooley and Tukey, 1965; Brigham, 1988), is pivotal for understanding and optimizing neural networks. The theoretical spectral bias (Rahaman et al., 2019) manifests in Transformers, where frequency transforms effectively substitute self-attention. FNet (Lee-Thorp et al., 2022) and its successors (Zhuang et al., 2022; Scribano et al., 2023) demonstrate the efficacy of such frequency representations. Physically, this aligns with multi-head self-attention acting as a low-pass filter (Wang et al., 2022; Park and Kim, 2022). Corroborating this, FourierTransformer (He et al., 2023) observes that hidden state power spectra concentrate in low-frequency bins in deeper layers. While these studies incorporate frequency transforms into AR architectures, leveraging this spectral property to actively guide dLLM decoding remains unexplored.

3 Method

3.1 Frequential Analysis on dLLM

Leveraging established insights from signal processing (Cooley and Tukey, 1965; Sorensen et al., 2003; Brigham, 1988) and their increasing adoption in NLP (Lee-Thorp et al., 2022; He et al., 2023; Liu et al., 2025a), we observe that low-frequency components of signals typically carry global trends and coarse-grained information, while high-frequency components encode detailed information. Based on this, we hypothesize that in dLLMs, the frequency spectrum of hidden states in decoding exhibits a similar semantic stratification.

To validate this, we select two text samples possessing distinct structural features for analysis, a Python script for calculating the Greatest Common Divisor shown in Figure 2, and a mathematical derivation of the difference of squares formula in Figure 6. After a single forward in dLLMs, we extract the final-layer hidden states $\boldsymbol{H} \in \mathbb{R}^{L\times D}$ where L is the sequence length and D is the hidden dimension, and compute the low-frequency component $\boldsymbol{H}_{\text{low}}$ by applying a binary mask $\boldsymbol{M}$ that only retains the lower 50% spectral energy.

$$\boldsymbol{H}_{\text{low}} = \mathcal{F}_r^{-1}\left(\mathcal{F}_r(\boldsymbol{H}) \odot \boldsymbol{M}\right), \tag{1}$$

where $\mathcal{F}_r$ denotes the real-valued Fourier transform, since $\boldsymbol{H}$ is real-valued (Sorensen et al., 2003). We then define the low-frequency ratio r_{low} as:

$$r_{\text{low}} = \frac{\|\boldsymbol{H}_{\text{low}}\|_2^2}{\|\boldsymbol{H}\|_2^2}, \tag{2}$$

where $\|\cdot\|_2$ denotes the Euclidean norm along the feature dimension D. Similarly, we also define the high-frequency ratio r_{high} as $1 - r_{\text{low}}$

Based on this metric, we highlight the Top-14 tokens with the highest low-frequency and high-frequency ratios in Figure 2 and Figure 6. Observations indicate that in the code task, reserved keywords constituting the logical skeleton of the program (*if*, *elif*, *return*) exhibit energy significantly concentrated in the low-frequency band. Conversely, specific function names and numerical values display distinct high-frequency characteristics. Similarly, in the math task, natural language text guiding the derivation logic is dominated by low-frequency signals, whereas specific mathematical formulas account for the majority of the high-frequency energy. This analysis confirms, for the first time in dLLMs, the correspondence where ***low-frequency implies structure, and high-frequency implies detail***. Based on these observations, we implement a dynamic spectral filtering method called FourierSampler with a frequency-domain sliding window that gradually transitions the passband from low to high frequencies. This allows the model to generate from structure to detail.

3.2 Translated Filtering Score

Our FourierSampler consists of two parts, the Translated Fourier Score, which adjusts the dLLMs' decoding based on energy in the frequency domain, and the Adaptive Fourier Calibrator, which controls the adjustment strength based on the original decoding confidence, as shown in Figure 3.

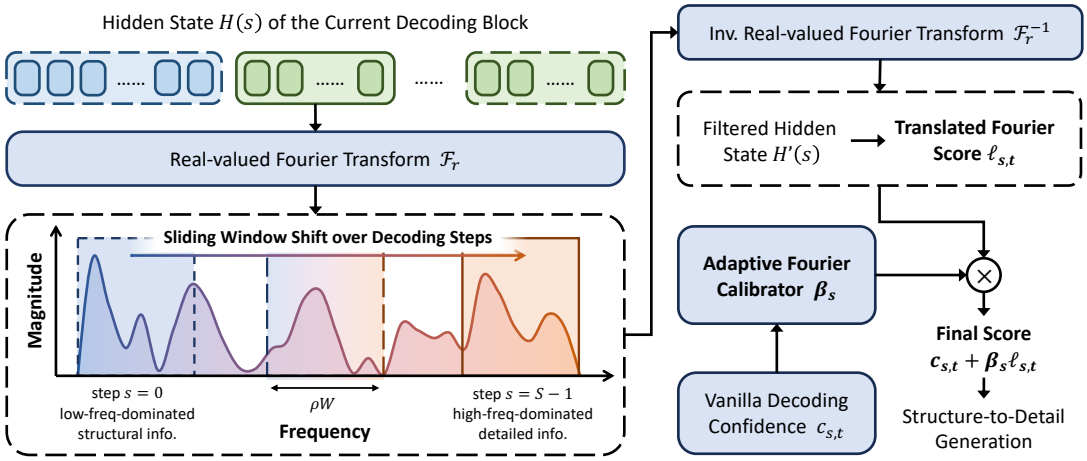

Figure 3: Overview of our FourierSampler. A sliding window in the frequency domain, retaining the low frequency at the beginning and the high frequency at the end based on the decoding step s, guides the dLLM to decode structural content first, then detailed content via Translated Fourier Score and Adaptive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

Assuming the decoding block size of the dLLM is B, and the number of decoding steps per block is S, let $\boldsymbol{H}(s) \in \mathbb{R}^{L\times D}$ represent the hidden state at decoding step s. For the token interval $[b, e)$ corresponding to the current decoding block, we perform real-valued Fourier transform $\mathcal{F}_r$, sliding window filtering $\boldsymbol{g}$, and inverse real-valued Fourier transform $\mathcal{F}_r^{-1}$ along the token dimension on the hidden state of the current block.

$$\boldsymbol{H}'(s) = \mathcal{F}_r^{-1}\left(\mathcal{F}_r\left(\boldsymbol{H}_{[b:e,:]}(s)\right) \odot \boldsymbol{g}(s)\right). \quad (3)$$

When the FT is applied for real-valued input, the negative frequency terms of the output are exact the complex conjugates of the corresponding positive-frequency terms, so the length of $\mathcal{F}_r$-transformed axis W equals to $B/2+1$. Furthermore, the filter $\boldsymbol{g}(s) \in \{0,1\}^W$ retains only the low-frequency parts at the beginning and only the high-frequency parts at the end based on the decoding step s. The starting position o_s and size of retained frequency band are determined by window ratio ρ, with the specified formula as follows.

$$\boldsymbol{g}_p(s) = \begin{cases} 1 & o_s \le p \le o_s + \rho W \\ 0 & \text{otherwise} \end{cases}, \qquad o_s = \left\lfloor \frac{W-\rho W}{S} \right\rfloor s. \quad (4)$$

This design forms a sliding window with fixed bandwidth and translated over decoding steps in the frequency domain. When the decoding step s is small, the window concentrates on the low-frequency region, emphasizing structural information. As s increases, the window gradually shifts toward high frequencies, making the model focus more on detailed content (He et al., 2023).

For token t in block $[b, e)$, namely $0 \le t \le B-1$, we calculate the energy of each token in the filtered frequency band and perform maximum normalization on the energy within the block, to eliminate scale differences between different samples and decoding block size, with ϵ to avoid division-by-zero exceptions. Then we obtain the Translated Fourier Score $\ell_{s,t}$:

$$\ell_{s,t} = \frac{\sum_{d=1}^{D}\left(\boldsymbol{H}'_{[b+t,d]}(s)\right)^2}{\max_{t'}\sum_{d=1}^{D}\left(\boldsymbol{H}'_{[b+t',d]}(s)\right)^2 + \epsilon}. \quad (5)$$

This score can be understood as the intensity of the token position under the retained frequency band. A larger $\ell_{s,t}$ indicates that the hidden states at position t in step s has a stronger intensity in the currently emphasized frequency band, thus better matching the frequential preference in current step.

3.3 Adaptive Fourier Calibrator

To dynamically adjust the guidance strength of the translated filtering score ℓ_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 writing priorities of different

positions in the current decoding state.

$$\sigma_s^2 = \mathrm{Var}\left(\{q_{s,t}\}_{t\in\mathcal{M}_s}\right). \tag{6}$$

We record the values of this method over the past 20 decoding steps, compute the percentile P_s of the current variance within the history, and normalize it into $w_s \in (0,1)$ interval the cumulative distribution function of the normal distribution. The process of normalization is detailed in Appendix A, and, finally, the adaptive weight is defined as follows, where $\beta_{\min}$ and $\beta_{\max}$ are the minimum and maximum values of adaptive weight, respectively.

$$\beta_s = \beta_{\min} + (1 - w_s)\big(\beta_{\max} - \beta_{\min}\big). \tag{7}$$

Based on the above translated filtering score and adaptive weight calculation, we add it to the original confidence $c_{s,t}$ as step s for token t to obtain the fusion score, shown as follows.

$$\tilde{c}_{s,t} = c_{s,t} + \beta_s \ell_{s,t}. \tag{8}$$

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.

4 Experiment

4.1 Setup

We conduct experiments on widely used diffusion-based dLLMs, including LLaDA1.5-8B (Zhu et al., 2025a) and LLaDA-8B-Instruct (Nie et al., 2025). In addition, to demonstrate that our method applies to models with other dLLM architectures, namely dLLMs with block-wise causal attention, we also evaluate our FourierSampler on SDAR-4B-Chat and SDAR-1.7B-Chat (Cheng et al., 2025). 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., 2024a, 2025b). 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 generation steps for Countdown. All experiments are conducted on NVIDIA H200 GPUs.

For LLaDA Series, we use PC-Sampler (Huang et al., 2025a) and RWS (Gwak et al., 2025) as the main baselines. For the SDAR Series, since PC-Sampler does not provide the token frequency distribution for SDAR, we use RWS as the primary baseline. The relevant coefficients in PC-Sampler follow the settings in its original paper. For RWS, we adopt GRM-Llama3.2-3B (Yang et al., 2024) as its reward model, recommended by its paper. We also compare dLLMs enhanced via different decoding strategies with similarly sized autoregressive models from Llama and Qwen Seires (Dubey et al., 2024; Meta, 2024; Qwen et al., 2024).

4.2 Main Results

For LLaDA1.5-8B and LLaDA-8B-Instruct, which are dLLMs based on full bidirectional attention, the experimental results in Table 1 demonstrate that our method consistently and significantly outperforms the baseline across different math and code tasks. In particular, we observe substantial relative improvements of up to 7.2% on MATH, 20.4% on MBPP, and 14.1% on Countdown compared to the baseline. Moreover, our approach achieves the highest average performance across all benchmarks, surpassing other competitive methods.

Notably, our FourierSampler further enables LLaDA1.5-8B to bridge and exceed the performance gap with similarly sized autoregressive models, such as Llama3.1-8B-Instruct and Qwen2.5-7B-Instruct, where LLaDA1.5-8B originally underperformed on average, and other decoding strategies fail to achieve it. This result highlights that our method can more effectively unlock the potential of non-autoregressive generation in dLLMs.

For dLLMs with block-wise causal attention, including SDAR-4B-Chat and SDAR-1.7B-Chat, our method also consistently outperforms the baseline across all evaluated benchmarks in Table 2. Specifically, we achieve relative improvements of 3.7% on MATH, 14.5% on HumanEval, and 45.1% on Countdown. In addition, the average performance across tasks is superior to that of other competing approaches. These results further demonstrate that our method generalizes well across different dLLM designs, including both full-bidirectional-attention and block-wise causal attention.

4.3 Ablation Study

To verify the rationality of the Adaptive Fourier Calibrator module design in our method, we conduct ablation studies by fixing the parameter β to

	GSM8k		Math		MBPP		HE		CD		Avg.	
Llama3.1-8B-Instruct	80.97		41.60		65.37		54.27		0.00		48.44	
Qwen2.5-7B-Instruct	81.65		49.20		66.93		52.44		4.30		50.90	
LLaDA1.5-8B	79.83		41.40		42.02		40.85		30.47		46.91	
+ PC-Sampler	81.20	+1.7%	43.00	+3.9%	**51.36**	+22.2%	39.02	−4.5%	21.09	−30.8%	47.13	+0.5%
+ RWS	80.67	+1.1%	42.00	+1.4%	43.19	+2.8%	39.63	−3.0%	30.08	−1.3%	47.11	+0.4%
+ FourierSampler (ours)	**81.80**	+2.5%	**44.20**	+6.8%	50.58	+20.4%	**43.29**	+6.0%	**34.77**	+14.1%	**50.93**	+8.6%
LLaDA-8B-Instruct	78.24		42.20		41.25		39.63		25.39		45.34	
+ PC-Sampler	79.00	+1.0%	40.40	−4.3%	**49.81**	+20.8%	39.63	0.0%	24.22	−4.6%	46.61	+2.8%
+ RWS	**79.61**	+1.8%	42.80	+1.4%	42.02	+1.9%	39.02	−1.5%	**30.08**	+18.5%	46.71	+3.0%
+ FourierSampler (ours)	**79.61**	+1.8%	**45.20**	+6.8%	47.86	+16.0%	**40.85**	+3.1%	28.90	+13.8%	**48.48**	+7.1%

Table 1: Results on LLaDA Series including LLaDA1.5-8B (Zhu et al., 2025a) and LLaDA-8B-Instruct (Nie et al., 2025) with best values in bold and relative improvement over vanilla decoding based on confidence in subscripts. Our FourierSampler achieves the best average performance, surpassing other competitive methods and even similarly sized AR models including Llama3.1-8B-Instruct (Dubey et al., 2024) and Qwen2.5-7B-Instruct (Qwen et al., 2024).

	GSM8k		Math		MBPP		HE		CD		Avg.	
Llama3.2-3B-Instruct	69.37		36.60		50.97		26.83		0.00		36.75	
Qwen2.5-3B-Instruct	76.42		39.20		47.47		32.93		10.94		41.39	
SDAR-4B-Chat	86.58		48.20		42.02		57.93		13.28		49.60	
+ RWS	87.41	+1.0%	49.20	+2.1%	39.69	−5.5%	60.37	+4.2%	16.02	+20.6%	50.54	+1.9%
+ FourierSampler (ours)	**87.64**	+1.2%	**50.00**	+3.7%	**47.47**	+13.0%	**62.20**	+7.4%	**16.80**	+26.5%	**52.82**	+6.5%
SDAR-1.7B-Chat	72.93		39.60		35.41		37.80		15.62		40.27	
+ RWS	**75.59**	+3.6%	**41.00**	+3.5%	35.80	+1.1%	31.71	−16.1%	17.97	+15.0%	40.41	+0.3%
+ FourierSampler (ours)	73.84	+1.2%	40.00	+1.0%	**36.58**	+3.3%	**43.29**	+14.5%	**22.66**	+45.1%	**43.27**	+7.4%

Table 2: Results on SDAR Series including SDAR-4B-Chat and SDAR-1.7B-Chat (Zhu et al., 2025a) with best values in bold and relative improvement over vanilla decoding based on confidence in subscripts. Our FourierSampler achieves the best average performance, surpassing other competitive methods and even similarly sized AR models including Llama3.2-3B-Instruct (Meta, 2024) and Qwen2.5-3B-Instruct (Qwen et al., 2024).

the maximum, minimum, and mean values of the adaptive weights, respectively, and evaluate the performance on GSM8K and MBPP. The results in Table 3 show that different tasks on the same model may prefer different weight values. However, using fixed weights consistently underperforms the adaptive weighting strategy. These observations further validate the effectiveness of our method.

We also conduct ablation studies regarding the choice of the sliding window size in the frequency domain, denoted as the window ratio ρ, for each model. Table 3 presents the experimental results for LLaDA-1.5B and LLaDA-8B-Instruct under different settings. Based on the overall performance across downstream tasks, we selected 0.2 and 0.4 as the window ratios for the two models, respectively.

	GSM8k	MBPP	Avg.
LLaDA1.5-8B	79.83	42.02	60.97
+ FourierSampler	**81.80**	**50.58**	**66.19**
+ Fixed $\beta = 0.4$	81.12	49.81	65.47
+ Fixed $\beta = 0.5$	81.12	47.86	64.49
+ Fixed $\beta = 0.6$	81.20	<u>50.19</u>	<u>65.70</u>
+ $\rho = 0.4$	<u>81.35</u>	42.80	62.08
+ $\rho = 0.6$	81.05	43.00	62.03
LLaDA-8B-Instruct	78.24	41.25	59.75
+ FourierSampler	<u>79.61</u>	**47.86**	**63.74**
+ Fixed $\beta = 0.4$	78.85	44.75	61.80
+ Fixed $\beta = 0.5$	79.38	46.69	63.04
+ Fixed $\beta = 0.6$	79.23	<u>47.08</u>	<u>63.16</u>
+ $\rho = 0.6$	**80.36**	45.91	63.14

Table 3: Results on LLaDA Series (Nie et al., 2025; Zhu et al., 2025a) for the ablation study of FourierSampler.

5 Discussion

5.1 Analysis of Decoding Block Size

To verify the effectiveness of FourierSampler under different decoding block sizes for dLLMs, we further conduct an experiment shown in Table 4 and observe that as the block size increases, applying FourierSampler to downstream tasks leads to more pronounced performance gains. This is because larger blocks provide a more complete and continuous signal for frequency-domain analysis, allow-

	GSM8K			MBPP		
	Vanilla	**Ours**	$\Delta(\%)$	**Vanilla**	**Ours**	$\Delta(\%)$
$B = 16$	80.82	81.05	$_{+0.3\%}$	49.03	48.25	$_{-1.6\%}$
$B = 32$	80.06	80.36	$_{+0.3\%}$	49.81	51.36	$_{+3.1\%}$
$B = 64$	79.83	81.80	$_{+2.5\%}$	42.02	50.58	$_{+20.4\%}$
$B = 128$	68.16	72.55	$_{+9.5\%}$	37.74	44.75	$_{+18.6\%}$

Table 4: Results with different block sizes. Here we use LLaDA1.5-8B as the base model. B denotes the block size and $\Delta(\%)$ denotes the relative improvement over vanilla decoding based on confidence.

ing the Fourier transform to more accurately capture low-frequency components that correspond to global semantics and structural information. When the block size is too small, the sequence is frequently segmented, which can fragment frequency-domain representations across blocks and make low-frequency information difficult to localize reliably. In contrast, larger blocks allow frame-level information to be fully distilled and well modeled at early stages, and to consistently guide subsequent fine-grained generation. As a result, the advantages of the FourierSampler become more evident in downstream task performance. It can be further observed that after applying FourierSampler, the scores on MBPP and GSM8K at block size $B = 64$ are both higher than the baseline at block size 32 or 16, suggesting that our method effectively mitigates, and in some cases even avoids the severe performance degradation when block size increases.

5.2 Analysis of Generation Order

To investigate whether the generation trajectory under FourierSampler aligns with the spectral characteristics observed in the static forward pass discussed in Section 3.1 and truly activates the non-autoregressive potential of dLLMs, we visualize the step-by-step decoding process of LLaDA-8B-Instruct (Nie et al., 2025) on a code generation task. The heatmap visualizes, for two decoding blocks (Block 1 and Block 2), the Translated Fourier Score $\ell_{s,t}$ computed at each token position under the frequency band selected for each generation step. Red stars indicate the generation step at which each token is ultimately finalized. A structure-to-detail generation pattern can be observed in Figure 4.

From the spectrum of Block 1, it is evident that keywords representing the logical skeleton of the program—such as *if* (position 9), and *return* (positions 17 and 28)—achieve high scores and are determined at very early decoding stages (Steps 0–10). This indicates that the model prioritizes constructing the overall structural framework of the code. In contrast, specific variable names (e.g., *fib*, *n*) and numerical values (e.g., *0*, *1*) are generally generated at later decoding stages (Steps 15–30). For instance, in Block 2, although *else* (position 33) and *return* (position 37) appear relatively early, the subsequent concrete computation logic involving *fib* (position 45) and *n* (position 47) does not emerge until around Step 20.

This structure-to-detail generation trajectory provides intuitive evidence that our Translated Fourier Score successfully maps low-frequency energy in the frequency domain to structural information in text. As a result, dLLM can plan global logic first and subsequently fill in local details.

5.3 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 $\boldsymbol{H} \in \mathbb{R}^{L\times D}$, apply the spectral filtering mechanism defined in Equation 1, and calculate the low-frequency ratio r_{low} as defined in Equation 2. 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 5 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 (Carnap, 1937; Ru et al., 2023; Liu et al., 2024b), occupy a larger proportion of low-frequency group. In particular, conjunctions (e.g., *but*, *if*, *because*), prepositions (e.g., *in*, *for*), 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 the previous section.

In contrast, nouns exhibit the strongest high-frequency characteristics among all part-of-speech categories. Nouns typically refer to concrete entities, variables, or values (e.g., *fib*, *n*, *0*), serving as the specific content that fills the syntactic skeleton. These results are consistent with our distinction be-

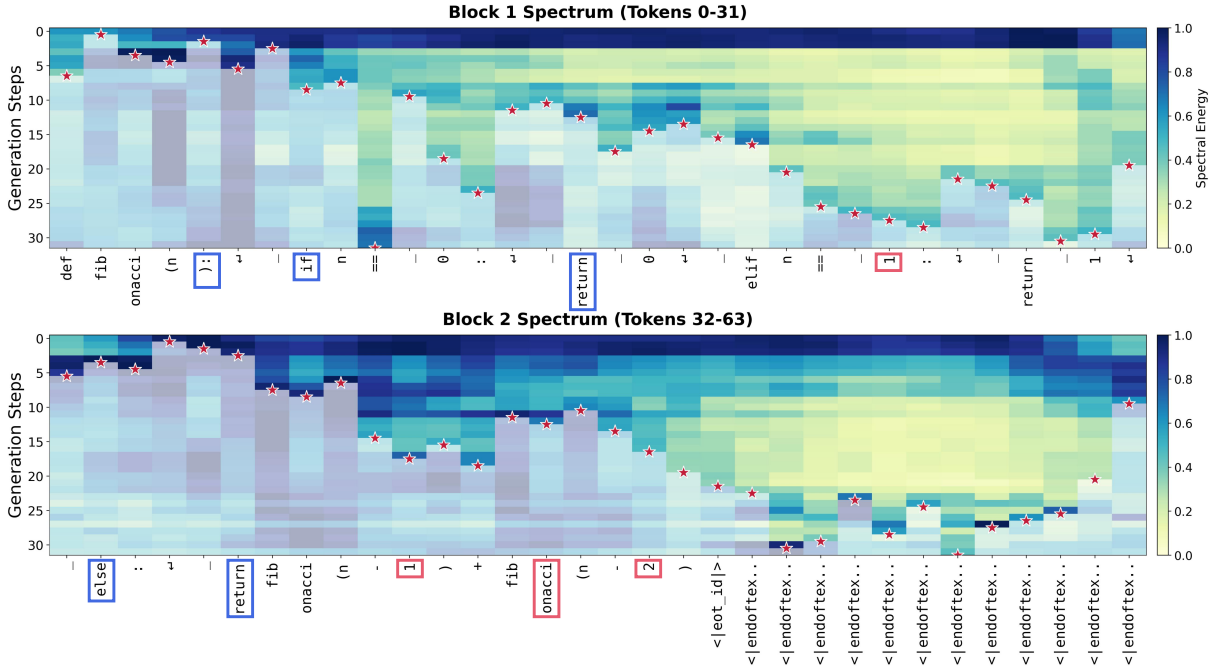

Figure 4: Visualization of step-wise generation trajectory for the prompt "Write a python function to compute Fibonacci sequence" on LLaDA-8B-Instruct (Nie et al., 2025). The heatmap displays the Translated Fourier Score $\ell_{s,t}$ at each step, with red stars marking the precise step where each token is decoded. We highlight blue boxes that correspond to structure words and are decoded in the early stages, and red boxes that correspond to detail words and are filled in later stages, which validates the "structure-to-detail" decoding pattern of FourierSampler.

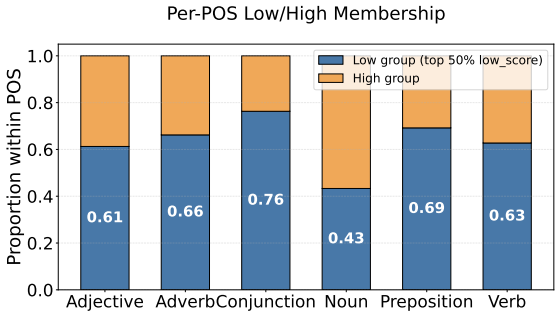

Figure 5: Low- and high-frequency features of different parts of speech in the frequency domain. The blue bars represent the proportion of tokens classified into the *low group*, while the orange bars represent the *high group*. Observations indicate that functional words responsible for syntactic structure, like conjunctions, exhibit dominant low-frequency features. In contrast, nouns, which typically serve as specific content fillers, show the strongest high-frequency tendency. This distribution corroborates our hypothesis that low-frequency components encode the structural skeleton, while high-frequency components correspond to detailed entities.

tween *framework words* and *detail words* in natural language. FourierSampler leverages this property to transform implicit linguistic hierarchies into explicit generation planning.

6 Conclusion

In this work, we investigate the internal decoding mechanisms of dLLMs from the perspective of signal processing. We conduct the first frequency analysis in dLLMs, showing that low-frequency implies structure, and high-frequency implies detail. Then, we propose FourierSampler. By leveraging the Translated Fourier Score and Adaptive Fourier Calibrator, our method dynamically guides the dLLMs to achieve a structure-to-detail generation.

Extensive experiments across full-bidirectional-attention (LLaDA Series) and block-wise causal attention (SDAR Series) architectures demonstrate that FourierSampler achieves consistent performance improvements in different tasks, such as math and code. Furthermore, our analyses regarding different decoding block sizes, generation order and part-of-speech distributions further corroborate the rationality of FourierSampler. This study not only experimentally surpasses similarly sized auto-regressive models but also paves an endogenous way for future research to unlock the arbitrary-order generation potential of dLLMs.

Limitation

Although our FourierSampler exhibits pronounced improvements with larger block sizes, the gains become relatively limited under small block sizes (e.g., block size 4 or 8). This behavior is consistent with our analysis in Section 5.1 that spectral signals require a certain level of continuity along the sequence dimension to reliably capture structural information. When the block size is small, the hidden states are fragmented into short segments, which constrains the ability to prioritize generating structural tokens before generating details.

Besides, our spectral analysis and decoding guidance are applied to the final-layer hidden states. While this design choice is simple and effective, we have not further investigated the frequency feature of intermediate layers. Different layers may encode structural and semantic information at varying levels of abstraction, and incorporating multi-layer frequency signals could potentially provide richer guidance for decoding. We leave a further analysis of intermediate layers to 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.

References

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*.

Gregor Bachmann and Vaishnavh Nagarajan. 2024. The pitfalls of next-token prediction. *arXiv preprint arXiv:2403.06963*.

Heli Ben-Hamu, Itai Gat, Daniel Severo, Niklas Nolte, and Brian Karrer. 2025. Accelerated sampling from masked diffusion models via entropy bounded unmasking. *arXiv preprint arXiv:2505.24857*.

Lukas Berglund, Meg Tong, Max Kaufmann, Mikita Balesni, Asa Cooper Stickland, Tomasz Korbak, and Owain Evans. 2023. The reversal curse: Llms trained on" a is b" fail to learn" b is a". *arXiv preprint arXiv:2309.12288*.

E Oran Brigham. 1988. *The fast Fourier transform and its applications*. Prentice-Hall, Inc.

Zheng Cai, Maosong Cao, Haojiong Chen, Kai Chen, Keyu Chen, Xin Chen, Xun Chen, Zehui Chen, Zhi Chen, Pei Chu, Xiaoyi Dong, Haodong Duan, Qi Fan, Zhaoye Fei, Yang Gao, Jiaye Ge, Chenya Gu, Yuzhe Gu, Tao Gui, and 81 others. 2024. Internlm2 technical report. *arXiv preprint arXiv:2403.17297*.

R Carnap. 1937. The logical syntax of language.

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*.

Shuang Cheng, Yihan Bian, Dawei Liu, Linfeng Zhang, Qian Yao, Zhongbo Tian, Wenhai Wang, Qipeng Guo, Kai Chen, Biqing Qi, and 1 others. 2025. Sdar: A synergistic diffusion-autoregression paradigm for scalable sequence generation. *arXiv preprint arXiv:2510.06303*.

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*.

OpenCompass Contributors. 2023. Opencompass: A universal evaluation platform for foundation models. https://github.com/open-compass/opencompass.

James W Cooley and John W Tukey. 1965. An algorithm for the machine calculation of complex fourier series. *Mathematics of computation*, 19(90):297–301.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, and 1 others. 2024. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*.

Gemini. 2025. Gemini diffusion, our state-of-the-art, experimental text diffusion model.

Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An, Peilin Zhao, Wei Bi, Jiawei Han, and 1 others. 2024. Scaling diffusion language models via adaptation from autoregressive models. *arXiv preprint arXiv:2410.17891*.

Daehoon Gwak, Minseo Jung, Junwoo Park, Minho 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 34562–34582.

Guangxin He, Shen Nie, Fengqi Zhu, Yuankang Zhao, Tianyi Bai, Ran Yan, Jie Fu, Chongxuan Li, and Binhang Yuan. 2025. Ultrallada: Scaling the context length to 128k for diffusion large language models. *arXiv preprint arXiv:2510.10481*.

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. *arXiv preprint arXiv:2305.15099*.

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, Shuhao Liu, Zhenghao Liu, Yukun Yan, Shuo Wang, Zulong Chen, and Tong Xiao. 2025a. Pc-sampler: Position-aware calibration of decoding bias in masked diffusion models. *arXiv preprint arXiv:2508.13021*.

Zemin Huang, Zhiyang Chen, Zijun Wang, Tiancheng Li, and Guo-Jun Qi. 2025b. Reinforcing the diffusion chain of lateral thought with diffusion language models. *arXiv preprint arXiv:2505.10446*.

Inception. 2025. Introducing mercury, the world's first commercial-scale diffusion language model.

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 Ontanon. 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.

Tianyi Li, Mingda Chen, Bowei Guo, and Zhiqiang Shen. 2025. A survey on diffusion language models. *arXiv preprint arXiv:2508.10875*.

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.

Sulin Liu, Juno Nam, Andrew Campbell, Hannes Stärk, Yilun Xu, Tommi Jaakkola, and Rafael Gómez-Bombarelli. 2024a. Think while you generate: Discrete diffusion with planned denoising. *arXiv preprint arXiv:2410.06264*.

Xiaoran Liu, Siyang He, Qiqi Wang, Ruixiao Li, Yuerong Song, Zhigeng Liu, Linlin Li, Qun Liu, Zengfeng Huang, Qipeng Guo, and 1 others. 2025a. Beyond homogeneous attention: Memory-efficient llms via fourier-approximated kv cache. *arXiv preprint arXiv:2506.11886*.

Xiaoran Liu, Kai Lv, Qipeng Guo, Hang Yan, Conghui He, Xipeng Qiu, and Dahua Lin. 2024b. Longwanjuan: Towards systematic measurement for long text quality. In *Findings of the Association for Computational Linguistics: EMNLP 2024*, pages 5709–5725.

Xiaoran Liu, Yuerong Song, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. 2025b. Longllada: Unlocking long context capabilities in diffusion llms. *arXiv preprint arXiv:2506.14429*.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. 2016. Pointer sentinel mixture models. *Preprint*, arXiv:1609.07843.

AI Meta. 2024. Llama 3.2: Revolutionizing edge ai and vision with open, customizable models. *Meta AI*.

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, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. 2024. Scaling up masked diffusion models on text. *arXiv preprint arXiv:2410.18514*.

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. *arXiv preprint arXiv:2502.09992*.

OpenAI. 2023. Gpt-4 technical report. Technical report.

Jingyang Ou, Shen Nie, Kaiwen Xue, Fengqi Zhu, Jiacheng Sun, Zhenguo Li, and Chongxuan Li. 2024. Your absorbing discrete diffusion secretly models the conditional distributions of clean data. *arXiv preprint arXiv:2406.03736*.

Namuk Park and Songkuk Kim. 2022. How do vision transformers work? *arXiv preprint arXiv:2202.06709*.

Qwen, :, 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, Jiaxi Yang, Jingren Zhou, and 24 others. 2024. Qwen2.5 technical report. *arXiv preprint arXiv:2412.15115*.

Nasim Rahaman, Aristide Baratin, Devansh Arpit, Felix Draxler, Min Lin, Fred Hamprecht, Yoshua Bengio, and Aaron Courville. 2019. On the spectral bias of neural networks. In *International conference on machine learning*, pages 5301–5310. PMLR.

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

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. *Advances in neural information processing systems*, 37:103131–103167.

Yuerong Song, Xiaoran Liu, Ruixiao Li, Zhigeng Liu, Zengfeng Huang, Qipeng Guo, Ziwei He, and Xipeng Qiu. 2025. Sparse-dllm: Accelerating diffusion llms with dynamic cache eviction. *arXiv preprint arXiv:2508.02558*.

H V Sorensen, D Jones, Michael Heideman, and C Burrus. 2003. Real-valued fast fourier transform algorithms. *IEEE Transactions on acoustics, speech, and signal processing*, 35(6):849–863.

Tianxiang Sun, Xiaotian Zhang, Zhengfu He, Peng Li, Qinyuan Cheng, Xiangyang Liu, Hang Yan, Yunfan Shao, Qiong Tang, Shiduo Zhang, Xingjian Zhao, Ke Chen, Yining Zheng, Zhejian Zhou, Ruixiao Li, Jun Zhan, Yunhua Zhou, Linyang Li, Xiaogui Yang, and 5 others. 2024. Moss: An open conversational large language model. *Machine Intelligence Research*.

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*.

Yinjie Wang, Ling Yang, Bowen Li, Ye Tian, Ke Shen, and Mengdi Wang. 2025. Revolutionizing reinforcement learning framework for diffusion large language models. *arXiv preprint arXiv:2509.06949*.

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. *arXiv 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*.

Ling Yang, Ye Tian, Bowen Li, Xinchen Zhang, Ke Shen, Yunhai Tong, and Mengdi Wang. 2025. Mmada: Multimodal large diffusion language models. *arXiv preprint arXiv:2505.15809*.

Rui Yang, Ruomeng Ding, Yong Lin, Huan Zhang, and Tong Zhang. 2024. 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. 2024a. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv:2410.14157*.

Jiacheng Ye, Shansan Gong, Liheng Chen, Lin Zheng, Jiahui Gao, Han Shi, Chuan Wu, Xin Jiang, Zhenguo Li, Wei Bi, and 1 others. 2024b. Diffusion of thought: Chain-of-thought reasoning in diffusion language models. *Advances in Neural Information Processing Systems*, 37:105345–105374.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. 2025a. 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. 2025b. Longproc: Benchmarking long-context language models on long procedural generation. *arXiv preprint arXiv:2501.05414*.

Zebin You, Shen Nie, Xiaolu Zhang, Jun Hu, Jun Zhou, Zhiwu Lu, Ji-Rong Wen, and Chongxuan Li. 2025. Llada-v: Large language diffusion models with visual instruction tuning. *arXiv preprint arXiv:2505.16933*.

Runpeng Yu, Qi Li, and Xinchao Wang. 2025. Discrete diffusion in large language and multimodal models: A survey. *arXiv preprint arXiv:2506.13759*.

Siyan Zhao, Devaansh Gupta, Qinqing Zheng, and Aditya Grover. 2025. d1: Scaling reasoning in diffusion large language models via reinforcement learning. *arXiv preprint arXiv:2504.12216*.

Fengqi Zhu, Rongzhen Wang, Shen Nie, Xiaolu Zhang, Chunwei Wu, Jun Hu, Jun Zhou, Jianfei Chen, Yankai Lin, Ji-Rong Wen, and 1 others. 2025a. Llada 1.5: Variance-reduced preference optimization for large language diffusion models. *arXiv preprint arXiv:2505.19223*.

Ying Zhu, Jiaxin Wan, Xiaoran Liu, Siyanag He, Qiqi Wang, Xu Guo, Tianyi Liang, Zengfeng Huang, Ziwei He, and Xipeng Qiu. 2025b. Dirl: An efficient post-training framework for diffusion language models. *arXiv preprint arXiv:2512.22234*.

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.

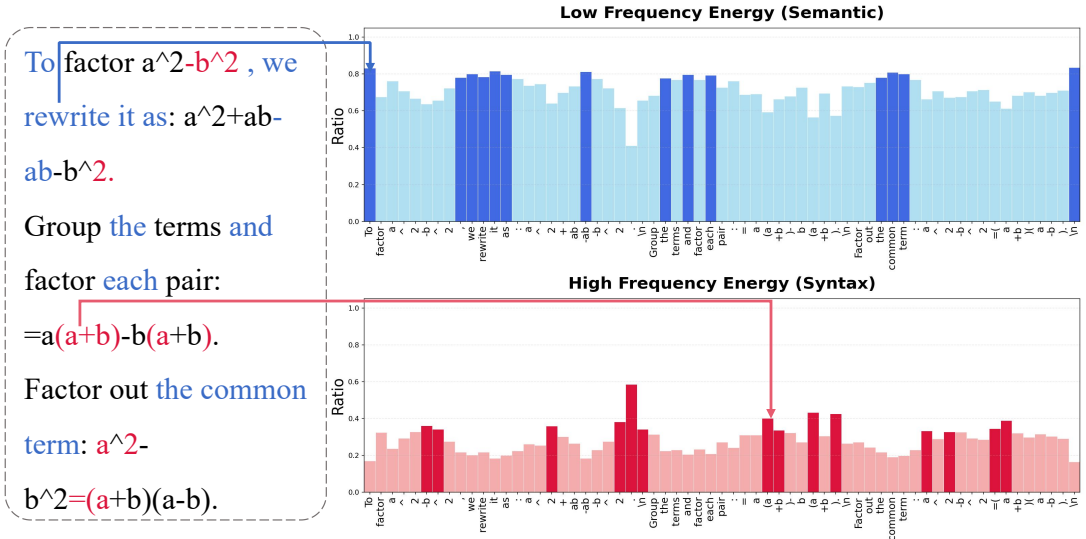

Figure 6: Visualization of the correspondence between frequency-domain analysis and textual information.

A Details of Method

The key hyperparameters of our FourierSampler for different dLLMs are shown in Table 5.

	ϵ	ρ	$\beta_{\min}$	$\beta_{\max}$
LLaDA1.5-8B	1e-5	0.2	0.4	0.6
LLaDA-8B-Instruct	1e-5	0.4	0.4	0.6

Table 5: Hyper-parameters.

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, $F(x) = \frac{1}{2} * \left(1 + \text{erf}\left(\frac{x}{\sqrt{2}}\right)\right)$. The pseudocode of the whole process is shown in Alg 1.

Algorithm 1 Compute Adaptive Weight β_s

procedure COMPUTEADAPTIVEWEIGHT
  if not $\mathcal{M}_s$.any() **then**
    return $\beta_{\min}$
  $\sigma_s^2 = \text{Var}\left(\{q_{s,t}\}_{t \in \mathcal{M}_s}\right)$
  v_list.append(σ^2)
  if len(v_list) > 20 **then**
    v_list.pop(0)
  if len(v_list) $== 0$ **then**
    $p_s = \frac{1}{2}$
  else
    v_list_ = [_ for _ in v_list if _ $< \sigma_s^2$)]
    $p_s = \frac{\text{len (v\_list\_)}}{\text{len (v\_list)}}$
  $z_s = \left(p_s - \frac{1}{2}\right) * 3$
  $w_s = \frac{1}{2} * \left(1 + \text{erf}\left(\frac{z_s}{\sqrt{2}}\right)\right)$
  $\beta_s = \beta_{\min} + (1 - w_s) * (\beta_{\max} - \beta_{\min})$
  return β_s, v_list

B Details of 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 6, 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 3.1.