CONTEXT TOKENS ARE ANCHORS: UNDERSTANDING THE REPETITION CURSE IN dMLLMS FROM AN INFORMATION FLOW PERSPECTIVE

Anonymous authors
Paper under double-blind review

ABSTRACT

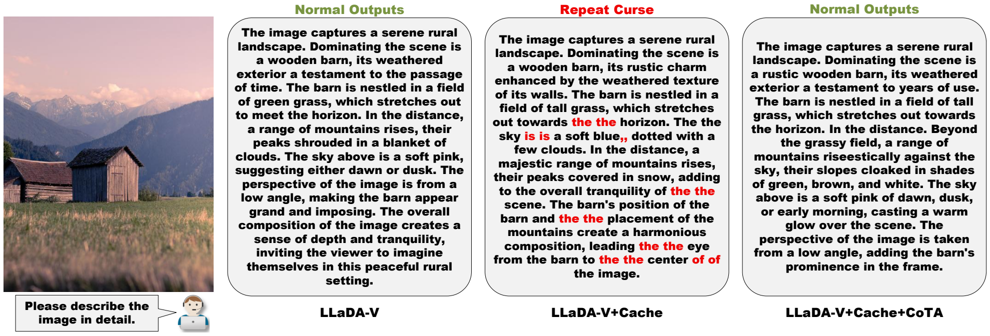
Recent diffusion-based Multimodal Large Language Models (dMLLMs) suffer from high inference latency and therefore rely on caching techniques to accelerate decoding. However, the application of cache mechanisms often introduces undesirable repetitive text generation, a phenomenon we term the **Repeat Curse**. To better investigate underlying mechanism behind this issue, we analyze repetition generation through the lens of information flow. Our work reveals three key findings: (1) context tokens aggregate semantic information as anchors and guide the final predictions; (2) as information propagates across layers, the entropy of context tokens converges in deeper layers, reflecting the model's growing prediction certainty; (3) Repetition is typically linked to disruptions in the information flow of context tokens and to the inability of their entropy to converge in deeper layers. Based on these insights, we present **CoTA**, a plug-and-play method for mitigating repetition. CoTA enhances the attention of context tokens to preserve intrinsic information flow patterns, while introducing a penalty term to the confidence score during decoding to avoid outputs driven by uncertain context tokens. With extensive experiments, CoTA demonstrates significant effectiveness in alleviating repetition and achieves consistent performance improvements on general tasks. Code will be made available.

1 INTRODUCTION

Recent advances in diffusion-based large language models (dLLMs) Nie et al. (2025); Zhu et al. (2025a); Ye et al. (2025); Gong et al. (2025) have demonstrated impressive reasoning and parallel decoding capabilities. Unlike autoregressive large language models Touvron et al. (2023a); Yang et al. (2024a); Chiang et al. (2023), which generate text by sequentially predicting the next token, dLLMs produce tokens in parallel by framing response generation as an iterative denoising process over a fully masked discrete token sequence. Building on this foundation, diffusion-based Multimodal Large Language Models (dMLLMs) You et al. (2025); Yang et al. (2025); Li et al. (2025b) have emerged as powerful multimodal systems that integrate vision instruction tuning with dLLMs, and have achieved performance comparable to leading autoregressive architectures across multiple benchmarks Liu et al. (2023c); Chen et al. (2024b); Fu et al. (2023).

Current efforts to accelerate dMLLMs Liu et al. (2025); Wu et al. (2025) primarily exploit customized caching strategies for both prefix and suffix tokens. While these approaches effectively reduce inference latency, our experiments demonstrate that they often introduce a severe side effect: the generated text exhibits substantial repetition, as shown in Figure 1. This repetition significantly reduces the performance and readability of the model outputs. We refer to this phenomenon as **"Repeat Curse"**. We propose four complementary metrics to quantitatively evaluate the "Repeat Curse" phenomenon. As illustrated in Figure 2.a, our empirical analysis shows that this issue consistently emerges in dMLLMs when cache techniques are employed.

However, the inherent black-box nature of dMLLMs presents a major obstacle to uncovering the internal mechanisms responsible for Repeat Curse. Recent advances in information flow analysis Yu et al. (2024); Wang et al. (2023); Chen et al. (2024a) have introduced an interpretable approach for understanding the relationship between model outputs and internal mechanisms, which has moti-



Please describe the image in detail.

The image captures a serene rural landscape. Dominating the scene is a wooden barn, its weathered exterior a testament to the passage of time. The barn is nestled in a field of green grass, which stretches out to meet the horizon. In the distance, a range of mountains rises, their peaks shrouded in a blanket of clouds. The sky above is a soft pink, suggesting either dawn or dusk. The perspective of the image is from a low angle, making the barn appear grand and imposing. The overall composition of the image creates a sense of depth and tranquility, inviting the viewer to imagine themselves in this peaceful rural setting.

The image captures a serene rural landscape. Dominating the scene is a wooden barn, its rustic charm enhanced by the weathered texture of its walls. The barn is nestled in a field of tall grass, which stretches out towards the the horizon. The the sky is is a soft blue,, dotted with a few clouds. In the distance, a majestic range of mountains rises, their peaks covered in snow, adding to the overall tranquility of the the scene. The barn's position of the barn and the the placement of the mountains create a harmonious composition, leading the the eye from the barn to the the center of of the image.

The image captures a serene rural landscape. Dominating the scene is a rustic wooden barn, its weathered exterior a testament to years of use. The barn is nestled in a field of tall grass, which stretches out towards the horizon. In the distance. Beyond the grassy field, a range of mountains riseestically against the sky, their slopes cloaked in shades of green, brown, and white. The sky above is a soft pink of dawn, dusk, or early morning, casting a warm glow over the scene. The perspective of the image is taken from a low angle, adding the barn's prominence in the frame.

Figure 1: **Motivation.** When cache is applied to accelerate dMLLMs, the generated responses often exhibit excessive token repetition—a phenomenon we term the Repeat Curse.

vated the development of numerous methods for mitigating abnormal output patterns (e.g., hallucination Tang et al. (2025c) and degeneration Yona et al. (2025)). Inspired by these successes, this work conducts an in-depth analysis from the perspective of information flow to reveal the connection between the internal mechanisms of dMLLMs and the "Repeat Curse."

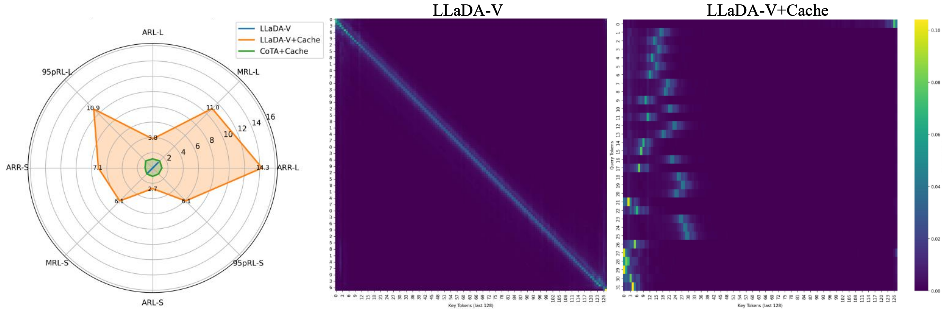
By visualizing the attention interaction patterns among tokens (as shown in Figure 3), we observe that in dMLLMs, the bidirectional attention mechanism enables context tokens adjacent to the query to act as anchors that progressively aggregate semantic information across layers, **causing attention to gradually concentrate on these context tokens**. We further introduce information entropy to analyze the impact of context tokens on decoding, as illustrated in Figure 4 a. We find that **the entropy of context tokens converges in deeper layers**, reflecting that as information is progressively aggregated across layers, the model's predictive certainty consistently increases. When repetition arises after applying the cache, the model exhibits random attention allocation (Figure 2.b). In addition, context tokens corresponding to repeated outputs sustain abnormally high entropy in deeper layers (Figure 4.b). Together, these findings suggest that caching disrupts the inherent mechanisms of dMLLMs, thereby triggering the Repeat Curse. We attribute this phenomenon to two key factors: 1. The introduction of caching disrupts the attention distribution and the inherent information flow patterns of context tokens; 2. Some context tokens whose entropy fails to converge in deeper layers induce the model to generate uncertain tokens, which are often accompanied by repetition.

Building on these insights, we propose **CoTA**[1], a plug-and-play approach that addresses the abnormal patterns underlying the Repeat Curse and mitigates repetition. CoTA is built on two key components: (1) Context-token Attention Enhancement (CTAE): a distance-aware attention intervention that strengthens attention to context tokens, thereby preserving the intended information flow during token interactions; (2) Context-token Entropy-guided Voting (CTEV): a mechanism that leverages the aggregated deep-layer entropy of context tokens as a penalty term in the confidence score, discouraging the model from generating uncertain and repetitive outputs. CoTA can be seamlessly integrated with baseline dMLLMs and existing caching strategies in a training-free manner, while incurring only modest computational overhead. Experimental results show that CoTA is highly effective in mitigating repetition, reducing the adjacent repetition rate by up to 92%. Furthermore, across several multimodal benchmarks, CoTA consistently surpasses the baseline, demonstrating substantial improvements in overall robustness and generalization.

Our contributions are three-fold.

- First, we identify the Repeat Curse phenomenon that emerges when caching is applied in dMLLMs and uncover its underlying causes through information flow analysis.
- Second, we introduce CoTA, a plug-and-play approach specifically designed to alleviate repetition.
- Third, we validate the effectiveness of CoTA through extensive experiments, demonstrating consistent performance improvements across multiple general multimodal tasks.

[1] The name CoTA is derived from our key finding: **CO**ntext **T**okens are **A**nchors.



(a) Quantitative Results of the "Repeat Curse"

(b) Visualizations of Attention Map in LLaDA-V and LLaDA-V+Cache

Figure 2: (a) presents the quantitative results of the "Repeat Curse". L and S indicate evaluations on long-text responses (512 tokens) and short-text responses (64 tokens). The evaluation metrics ARR, MRL, ARL, and 95pRL are introduced in Section 3.2 and Appendix J. (b) visualizes the attention maps of LLaDA-V and LLaDA-V+Cache.

2 Related Work

Diffusion-based Multimodal Large Language Models. The latest diffusion-based large language models (dLLMs) Nie et al. (2025); Zhu et al. (2025a); Ye et al. (2025); Gong et al. (2025) have been successfully scaled to 8B parameters, achieving performance comparable to state-of-the-art autoregressive large language models Dubey et al. (2024); Bi et al. (2024); Yang et al. (2024a). By combining visual instruction tuning Liu et al. (2023a) with dLLMs, You et al. (2025); Yang et al. (2025); Li et al. (2025b) successfully develops diffusion-based multimodal large language models.

Information Flow. Recent studies have underscored the importance of information flow as an intuitive means of representing the internal mechanisms of black-box models. Common approaches for analyzing information flow mainly include saliency scores Yu et al. (2024); Wang et al. (2023), attention maps Xiao et al. (2023); Huang et al. (2023), Grad-CAM Zhang et al. (2024), and massive values Jin et al. (2025), among others. Prior studies Yu et al. (2024); Chen et al. (2024a); Tang et al. (2025a) have demonstrated the existence of certain anchor tokens in autoregressive models, which aggregate information and play a crucial role in cross-layer information flow. Additionally, ADLM Rout et al. (2025) discusses the role of anchors in semantic guidance within diffusion language models. These findings inspire our exploration of information flow in dMLLMs.

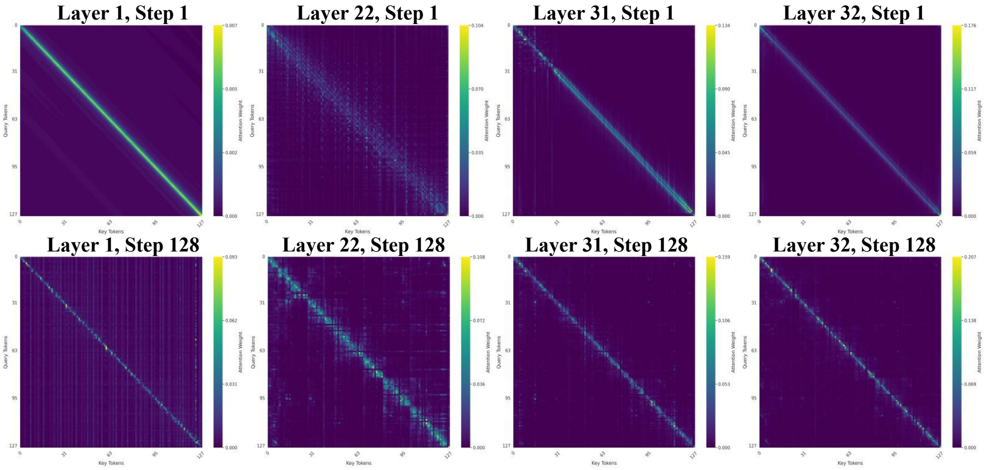
3 Motivation and Analysis

In this section, we begin with a brief overview of baseline dMLLMs and the cache mechanism[2]. We then investigate the 'Repeat Curse' through both quantitative and qualitative analyses. Finally, we compare the information flow in dMLLMs with and without cache, shedding light on the underlying cause of the Repeat Curse.

3.1 Preliminary

Diffusion-based Multimodal Large Language Models (dMLLMs). Typically, a dMLLM $\mathcal{F}$ consists of three main components: a pretrained vision encoder $\mathcal{F}_v$, a dLLM $\mathcal{F}_t$, and a projector f that maps visual features into the text embedding space. Given an image input I_v and an instruction prompt I_p (e.g., *"Please describe the image in detail"*), the model converts them into tokens and concatenates them into a multimodal sequence: $\{\mathcal{S}_v, \mathcal{S}_t\}$, where $\mathcal{S}_v = f(\mathcal{F}_v(I_v)) = \{w_i\}_{i=1}^{V}$ and $\mathcal{S}_t = \mathcal{F}_t(I_p) = \{w_i\}_{i=1}^{T}$ represent visual and instruction tokens of lengths V and T, respectively. Subsequently, a fully masked token sequence $\mathcal{S}_m = \{w_i\}_{i=1}^{M}$ of lengths M is initialized as the response sequence and concatenated with $\{\mathcal{S}_v, \mathcal{S}_t\}$ to form the final input sequence $\mathcal{S} = \{\mathcal{S}_v, \mathcal{S}_t, \mathcal{S}_m\}$. In such sequence, different tokens share the same dimension.

[2] This paper adopts LLaDA-V You et al. (2025) and dLLM-Cache Liu et al. (2025) as the baseline dMLLMs and cache method.



Layer 1, Step 1

Layer 22, Step 1

Layer 31, Step 1

Layer 32, Step 1

Layer 1, Step 128

Layer 22, Step 128

Layer 31, Step 128

Layer 32, Step 128

Figure 3: **Attention Maps Visualization of LLaDA-V**. Based on LLaDA-V 8B with a generation length of 128 and 32 decoding steps, we visualize the attention matrices corresponding to the masked (response) tokens, with the x-axis denoting key tokens and the y-axis denoting query tokens. Brighter colors indicate higher attention values. Context tokens act as anchors to aggregate information across layers and absorb attention.

Inference and Sampling in dMLLMs. Let $\mathcal{S}^t = \{\mathcal{S}_v^t, \mathcal{S}_t^t, \mathcal{S}_m^t\}$ denote the input sequence state at decoding step t. Starting from $\mathcal{S}^0$, where the response sequence $\mathcal{S}_m$ is fully masked, the dMLLM performs an iterative unmasking process over T discrete steps to generate the final text response. At each step t, the model computes a probability distribution $p_\theta(\mathcal{S}^t \mid \mathcal{S}^{t-1})$ for every masked token. From these distributions, the most likely token predictions $\hat{\mathcal{S}}_{(i)}^t$ and their corresponding confidence scores $c_{(i)}$ are determined:

$$\hat{\mathcal{S}}_{(i)}^t = \arg\max_{v \in V} p_\theta(\mathcal{S}_{(i)}^t = v \mid \mathcal{S}^{t-1}) \quad \text{and} \quad c_{(i)} = p_\theta(\mathcal{S}_{(i)}^t = \hat{\mathcal{S}}_{(i)}^t \mid \mathcal{S}^{t-1}), \quad i \in M_{t-1}, \tag{1}$$

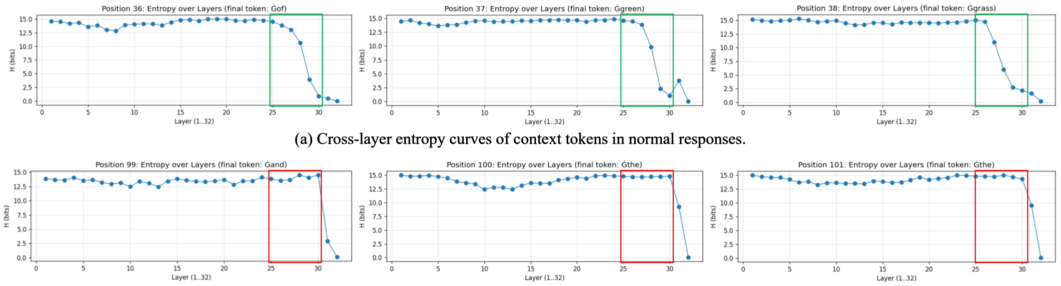
where M_{t-1} represents the index set of masked token positions at step $(t-1)$ and V is the vocabulary. Finally, the model selects the k positions in M_{t-1} with the highest confidence scores $c_{(i)}$ and obtains the set of indices to update U_t. The k selected tokens are unmasked at this step, producing the updated sequence:

$$\mathcal{S}_{(i)}^t = \begin{cases} \hat{\mathcal{S}}_{(i)}^t, & \text{if } i \in U_t, \\ \mathcal{S}_{(i)}^{t-1}, & \text{otherwise.} \end{cases} \tag{2}$$

The indices of the masked tokens at step t are updated by $M_t = M_{t-1} - U_t$.

Cache Mechanism for dMLLMs. During the iterative unmasking process of dMLLMs, attention needs to be computed over all tokens in the sequence $\mathcal{S}$ at each step, which leads to significant inference latency. Since dMLLMs adopt a bidirectional attention mechanism, the conventional KV-cache technique Xiao & et al. (2024) is not applicable. To address this, several caching methods Liu et al. (2025); Wu et al. (2025) leverage a similar observation: prefix tokens $\{\mathcal{S}_v, \mathcal{S}_t\}$ and parts of the suffix tokens $\mathcal{S}_m$ exhibit minimal changes in their attention values during inference, which enables the design of token state caching and reuse. For example, in Liu et al. (2025), the core caching mechanism is defined as follows:

$$\begin{cases} \text{Recompute}(\{\mathcal{S}_v^t, \mathcal{S}_t^t\}), & \text{if } t \bmod \frac{T}{\mathcal{E}_p} = 0, \\ \text{Recompute}(\mathcal{S}_m^t), & \text{if } t \bmod \frac{T}{\mathcal{E}_s} = 0, \\ \text{Recompute}(\mathcal{S}_{(i)}^t), & \text{if } \mathcal{S}_{(i)}^t \in \arg\min_\alpha \Big(\text{Sim}\big(\mathcal{S}_m^t, \mathcal{S}_m^{t-1}\big) \Big). \end{cases} \tag{3}$$



(a) Cross-layer entropy curves of context tokens in normal responses.

(b) Cross-layer entropy curves of context tokens in Repetition responses.

Figure 4: (a) and (b) correspond to context tokens from normal decoding and from decoding with repeated tokens, respectively. We define the set of context tokens as the target tokens together with their two nearest neighboring tokens in relative position. Context tokens with repetition tend to exhibit high entropy in deeper layers.

Here, $\mathrm{Recompute}()$ denotes the recomputation of attention. $\mathcal{E}_p$ and $\mathcal{E}_s$ indicate the predefined update periods for prefix and suffix tokens, respectively. $\arg\min_\alpha$ selects the bottom α proportion of elements from the sequence, and $\mathrm{Sim}()$ refers to the cosine similarity function.[3]

3.2 Repeat Curse

Repetition Quantitative Metrics. As illustrated in Figure 1, we define the redundant repetition of tokens in model responses as the Repeat Curse. To quantitatively evaluate this phenomenon, we introduce the Adjacent Repetition Rate (ARR), which measures the proportion of repeated tokens within a response sequence $\{y_i\}_{i=1}^{M}$ of lengths M:

$$\text{ARR} = \frac{1}{M-1}\sum_{i=1}^{M} \mathbf{1}(y_i = y_{i-1})\,, \tag{4}$$

where $\mathbf{1}()$ is an indicator function that takes the value 1 when the condition is satisfied and 0 otherwise. In addition, we introduce sample repetition rate (SRR), maximum repetition length (MRL), average repetition length (ARL), and 95th-percentile repetition length (95pRL) to evaluate the severity of the Repeat Curse. (The detailed computation procedures are presented in Appendix J.)

Quantitative Experimental Results. Figure 2(a) summarizes the quantitative results of the Repeat Curse under both long- and short-response settings. The experiments are performed on the image captioning task with 500 randomly sampled MSCOCO images Lin et al. (2014). We observe that the baseline model suffers from severe token repetition when the cache is applied.

3.3 Information Flow Analysis

Motivated by the need to understand the underlying cause of the Repeat Curse, we analyze the information flow to contrast model behaviors with and without cache, and provide a mechanistic interpretation of the phenomenon. To this end, we visualize the model's attention matrices to analyze the information flow among tokens. As shown in Figure 3, we observe that context tokens consistently receive high attention throughout the decoding process. Furthermore, attention progressively converges toward the context tokens from shallow to deeper layers. This phenomenon is reminiscent of the 'attention sink' observed in autoregressive models Chen et al. (2024a); Wang et al. (2023); Zhang et al. (2024), where certain special tokens act as anchors that aggregate information and absorb attention. The information flow pattern in Figure 3 highlights the role of context tokens as anchors in dMLLMs.

Finding 1: In dMLLMs, context tokens serve as anchors that aggregate information across layers and guide the final prediction.

[3] We adopt dLLM-Cache Liu et al. (2025) as the baseline cache method, keeping all relevant settings strictly consistent to ensure experimental fairness. Hyperparameters are fixed as $\alpha = 25\%$, $\mathcal{E}_p = 25$, and $\mathcal{E}_s = 7$.

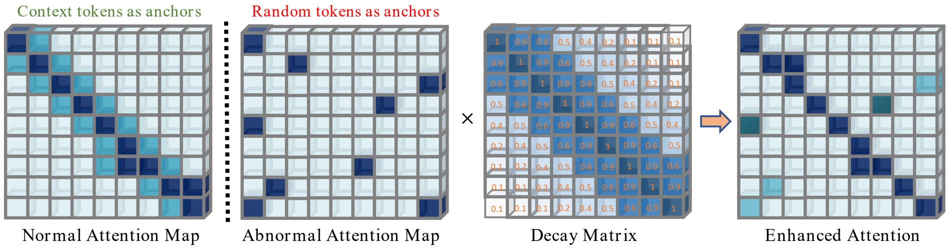


Figure 5: **Illustration of context tokens attention enhancement**. Example values are annotated on the decay matrix for clarity, while actual values are computed using Equation 5 and 6.

Subsequently, we plot the cross-layer entropy curves of context tokens, as shown in Figure 4. We find that the entropy of context tokens remains high in the shallow layers but gradually converges in the deeper layers, reflecting that as information accumulates layer by layer, the model's predictions become increasingly stable and certain.

Finding 2: For context tokens under normal decoding, the information entropy gradually converges in the deeper layers.

Figure 2.b shows that after applying cache, the model exhibits a randomized attention distribution, disrupting the original information flow from context tokens and weakening output stability. Figure 4.b further presents the cross-layer entropy curves of context tokens under repetition, revealing that repetition is accompanied by a failure of entropy convergence in the deeper layers.

Finding 3: Repetition is typically linked to disruptions in the information flow of context tokens and to the inability of their entropy to converge in deeper layers.

4 Method

Building on the observations in Section 3, we propose CoTA, a training-free method to mitigate the Repeat Curse, which has two key components: (1) Context Tokens Attention Enhancement, which preserves the intrinsic information flow pattern of context tokens, and (2) Context Tokens Entropy-Guided Voting, which prevents outputs driven by uncertain context tokens during decoding.

4.1 Context Tokens Attention Enhancement (CATE)

As analyzed in Section 3, context tokens in dMLLMs typically serve as anchors to aggregate information and absorb attention, but the use of cache disrupts this information flow pattern. We conjecture that tokens with shorter relative distances exhibit stronger semantic correlations, and thus the model's increased attention to context tokens facilitates local semantic coherence. Therefore, we propose Context Tokens Attention Enhancement (CTAE), a simple yet effective attention intervention. We formalized the complete algorithmic procedure in Algorithm 1. As illustrated in the figure 5, the core idea is to introduce a decay term and apply an element-wise multiplication with the attention matrix. Given a query token q_i at position i and a key token k_j at position j, the decay term $\mathcal{G}_{i,j}$ for each query–key pair is computed as follows:

$$\mathcal{G}_{i,j} = \gamma_{\min} + (1 - \gamma_{\min}) g_{i,j}, \quad \gamma_{\min} \in (0, 1], \tag{5}$$

$$g_{i,j} = \exp\left(- \left(\frac{|i-j|}{\tau} \right)^2 \right). \tag{6}$$

Here, $g_{i,j}$ denotes the Gaussian decay term computed using the exponential function $\exp()$ and the relative positional distance $|i - j|$ between query and key tokens. τ denotes the temperature factor, which is fixed to 5 in the experiments. We then introduce a lower-bound constant $\gamma_{\min}$ for stabilization, yielding the final decay term $\mathcal{G}_{i,j}$.

Assume the attention between a query token q_i and a key token k_j is denoted as $Attn_{i,j}$. We enhance the attention to context tokens by applying $Attn_{i,j} * \mathcal{G}_{i,j}$, thereby preserving the native information flow pattern in dMLLMs.

Algorithm 1 Context Tokens Attention Enhancement (CTAE)

```
Require: Attention weights $A \in \mathbb{R}^{L \times H \times T \times T}$, temperature $\tau > 0$, lower bound $\gamma_{\min} \in (0, 1]$
Ensure: Enhanced attention $\tilde{A} \in \mathbb{R}^{L \times H \times T \times T}$
$\tilde{A} \leftarrow A$
for $\ell \leftarrow 1$ to $L$ do                          ▷ layer index
    for $h \leftarrow 1$ to $H$ do                         ▷ head index
        $D_{i,j} \leftarrow |i - j| \quad \forall i, j \in \{1, \dots, T\}$
        $g_{i,j} \leftarrow \exp\bigl(-(D_{i,j}/\tau)^2\bigr)$
        $\mathcal{G}_{i,j} \leftarrow \gamma_{\min} + (1 - \gamma_{\min}) \cdot g_{i,j}$
        $\tilde{A}_{\ell,h} \leftarrow A_{\ell,h} \odot \mathcal{G}$
    end for
end for
return $\tilde{A}$
```

Algorithm 2 Context Tokens Entropy-Guided Voting (CTEV)

```
Input: Deep-layer logits $z^{(l)}_{i,:}$, base confidences $c_{(i)}$, coefficient $\alpha$
for $i \leftarrow 1$ to $M$ do
    $E_{\text{sum}}(i) \leftarrow 0$
    for $l \in \{26, \dots, 30\}$ do
        $p^{(l)}_v = \exp(z^{(l)}_{i,v}) / \sum_{u=1}^{V} \exp(z^{(l)}_{i,u})$
        $E_{\text{sum}}(i) \leftarrow E_{\text{sum}}(i) - \frac{1}{\log V} \sum_{v=1}^{V} p^{(l)}_v \log p^{(l)}_v$
    end for
end for
Build $\mathcal{C}(t) = \{t\} \cup \{\text{two nearest tokens of } t\}$
for $i \in \mathcal{C}(t)$ do
    $E^{ctx}_{\text{sum}}(i) \leftarrow \sum_{j \in \mathcal{C}(i)} E_{\text{sum}}(j)$
    $\text{Score}(i) \leftarrow c_{(i)} + \alpha \cdot E^{ctx}_{\text{sum}}(i)$
end for
```

Synergistic Workflow. CTAE applies a decay term to attention values based on relative distance, enhancing the attention to context tokens. CTEV introduces deep-layer context token entropy as a penalty to the voting scores. Together, they jointly alleviate the repeat curse.

4.2 Context Tokens Entropy-Guided Voting (CTEV)

As observed in Section 3, context tokens with repetition typically exhibit persistently high entropy in deeper layers, reflecting the model's uncertainty about context tokens during decoding. Yet baseline dMLLMs rely solely on confidence scores to vote for candidate decoding tokens, ignoring this uncertainty. To address this, we propose Context Tokens Entropy-Guided Voting (CTEV), which incorporates the aggregated deep-layer entropy of context tokens as a penalty term on the confidence score to prevent decoding under uncertain context tokens. The complete algorithm is formalized in Algorithm 2. First, we compute the entropy of each candidate token based on its softmax probability distribution p_v as follows:

$$E = \frac{-\sum_{v=1}^{V} p_v \, \log p_v}{\log V}, p_v = \frac{\exp(z_v)}{\sum_{u=1}^{V} \exp(z_u)}, \tag{7}$$

where z_v denotes the logits of the v-th word in the vocabulary, and V is the vocabulary size. On this basis, the entropy of tokens in deeper layers [4] is accumulated layer by layer as follows:

$$E_{\text{sum}} = \sum_{l=26}^{30} E^{(l)} = \sum_{l=26}^{30} \frac{-\sum_{v=1}^{V} p^{(l)}_v \, \log p^{(l)}_v}{\log V}. \tag{8}$$

Let $E_{\text{sum}}(i)$ denote the E_{sum} of token x_i at position i in a sequence of length M. The accumulated deep-layer entropy of the context tokens, $E^{ctx}_{\text{sum}}(i)$, is then computed as follows:

$$E^{ctx}_{\text{sum}}(i) \;=\; \sum_{j \in \mathcal{C}(i)} E_{\text{sum}}(i), \qquad \mathcal{C}(i) = \{\, x_i \,\} \cup \Big\{\, x_j \;\big|\; j \in \arg\min_{j \in [M] \setminus \{i\}}{}_2 \, |j - i| \Big\}. \tag{9}$$

Finally, the weighted $E^{ctx}_{\text{sum}}(i)$ with coefficient α is incorporated into the confidence score $c_{(i)}$ from Equation 1, resulting in the new voting score $\text{Score}(i)$ defined as follows:

$$\text{Score}(i) = c_{(i)} + \alpha E^{ctx}_{\text{sum}}(i). \tag{10}$$

[4] Here, we define layers 26–30 as the deep layers, and target tokens together with their two nearest tokens in relative position are defined as the context tokens.

Method	512				64			
	ARR↓	MRL↓	ARL↓	SRR↓	ARR↓	MRL↓	ARL↓	SRR↓
LLaDA-V	0.2	2.0	2.0	6.9	0.1	2.0	2.0	3.3
+ dllm-cache	14.3	11.0	3.8	82.3	7.1	6.1	2.7	65.6
+ dllm-cache + CTEV	3.2	2.2	2.0	10.6	2.5	2.3	2.1	5.6
+ dllm-cache + CTAE	2.9	1.9	1.4	8.0	1.8	1.6	1.4	4.6
+ dllm-cache + CTEV +CTAE	**1.2**	**1.3**	**1.2**	**6.3**	**1.0**	**1.1**	**1.1**	**3.0**

Table 1: **"Repeat Curse" Evaluation Results.** 512 and 64 denote the maximum generation length.

Model	Type	LLM Tower	DocVQA	ChartQA	MMStar	MME^p	Seed^I	MMBench
LLaVA1.5	AR	Vicuna-7B	-	-	-	1510	66.1	64.3
Qwen2-VL	AR	Qwen2-7B	-	83.0	60.7	-	-	-
DeepSeek-VL	AR	DeepSeek-7B	-	-	-	-	70.4	73.2
LLaVA-OV	AR	Qwen2-7B	-	80.0	61.7	1580	75.4	80.8
MetaMorph	AR+Diff.	LLaMA3.1-8B	-	37.1	-	-	71.8	75.2
JanusFlow	AR+Diff.	DeepSeek-1.3B	-	64.6	-	1333	70.5	74.9
LLaMA3-V	Diff.	LLaMA3-8B	86.2	77.8	56.5	1581	76.6	79.8
LLaDA-V	Diff.	LLaDA-8B	83.9	78.3	60.1	1507	74.8	82.9
LLaDA-V w/dllm-cache	Diff.	LLaDA-8B	82.1	78.1	58.3	1410	72.1	83.0
Ours	Diff.	LLaDA-8B	84.1	78.4	59.3	1523	73.9	83.1

Table 2: Benchmark results of different MLLMs on multiple multimodal evaluation datasets.

5 Experiments

5.1 Experimental Setup Details

Model. Following the baseline dMLLM LLaDA-V You et al. (2025), the language tower adopts LLaDA-8B-Instruct Nie et al. (2025), the vision tower employs siglip2-so400m-patch14-384 Tschannen et al. (2025), and the projector is a two-layer MLP.

Evaluation. We evaluate the effectiveness of our method on eight multimodal benchmarks: DocVQA Mathew et al. (2020), ChartQA Masry et al. (2022), MMStar Chen et al. (2024b), MME Fu et al. (2023), Seed Li et al. (2023a), LLaVA^w Liu et al. (2023a), MathVista Lu et al. (2023), and MMBench Liu et al. (2023c). The Appendix I reports more details about settings.

5.2 Results on "repeat curse" evaluation

We randomly select 500 samples from COCO2014 Lin et al. (2014) for the caption VQA task. The generated captions are then evaluated for mitigating the "Repeat Curse." The evaluation metrics include Adjacent Repetition Rate (ARR), Maximum Repetition Length (MRL), Average Repetition Length (ARL), and Sample Repetition Rate (SRR). Table 1 presents the quantitative results of our method in mitigating repetition, evaluated under both long-text and short-text response settings. Experimental results are obtained by aggregating all samples and taking the average. The baseline model shows only minimal repetition, with the maximum repetition length being 2. In contrast, applying cache causes a rapid degradation of the outputs. Moreover, longer responses are more susceptible to repetition; for instance, with an output length of 512, ARR rises by 14.1 and SRR by 75.4. Our method demonstrates effective repetition mitigation, yielding ARR improvements of 13.1 and 6.1 under the long-response and short-response settings, respectively. We further perform ablation studies on two key components of our method, CTEV and CTAE. The results indicate that each component alone can also effectively mitigate repetition, highlighting the flexibility and complementarity of the method.

5.3 Comparison with Other MLLMs

Table 2 presents a performance comparison with other paradigm MLLMs. Our goal is not absolute performance maximization, but rather to mitigate the output degradation of baseline models when using cache. Results across six benchmarks demonstrate the effectiveness of our method.

Method	LLaVA^w				MathVista			
	Score↑	ARR↓	TPS↑	FLOPs ↓	Score ↑	ARR ↓	TPS ↑	FLOPs ↓
LLaDA-V	70.1	1.3	7.3	16.1	59.7	0	8.3	14.1
Ours	70.5	1.1	5.1	17.8	59.8	0	7.5	15.3
LLaDA-V w/dllm-cache	63.2	7.3	23.1	3.2	54.9	6.9	31.0	3.9
Ours w/dllm-cache	69.9	1.4	20.3	5.1	59.7	1.3	28.7	5.0

Table 3: **Generalization evaluation results of the method**. TPS stands for Tokens Per Second, and FLOPs stands for Floating Point Operations.

5.4 Generality Study

We present the generalization evaluation of our method in Table 3. Our approach consistently lowers response repetition across both open-domain natural and mathematical tasks, highlighting its generalizability. While caching reduces computational cost, it inevitably causes performance degradation, which our method successfully alleviates. Concretely, it yields gains of +6.7 in Score and +5.9 in ARR on MathVerse, and +1.8 in ACC and +5.6 in ARR on MathVista.

Our method can be flexibly applied as a complement to caching. We further evaluate its computational overhead, with experiments on LLaVAW demonstrating that it incurs only a 2.8 TPS reduction and a 1.9 FLOPs increase.

5.5 Ablation Study

We conduct an ablation study on the hyperparameters involved in our method, including the weighting factor α in Equation 10, the minimum gain $\gamma_{\min}$ in Equation 5, and the number of context tokens. We report the results in Table 4: the best hyperparameter configuration is $\alpha = 0.75$, $\gamma_{\min} = 0.5$, and the number of context tokens set to 3, achieving an ARR of 1.2% and an ACC of 23.1. When the length of context tokens is set to 1, only the current target token is considered while surrounding tokens are ignored. We find that using longer context tokens (e.g., 5) does not lead to better results.

$\gamma_{\min}$	α	ARR↓	ACC↑
1	0.25	2.8	21.4
	0.5	2.2	21.9
	0.75	2.3	22.1
0.5	0.25	2.9	20.0
	0.5	2.5	20.2
	0.75	2.1	20.1

(a) Context tokens number = 1

$\gamma_{\min}$	α	ARR↓	ACC↑
1	0.25	2.1	22.1
	0.5	2.0	22.5
	0.75	1.7	22.3
0.5	0.25	1.6	22.2
	0.5	1.4	22.0
	0.75	1.2	23.1

(b) Context tokens number = 3

$\gamma_{\min}$	α	ARR↓	ACC↑
1	0.25	3.5	19.6
	0.5	3.8	19.2
	0.75	3.1	19.2
0.5	0.25	3.9	19.0
	0.5	3.4	18.2
	0.75	3.3	19.1

(c) Context tokens number = 5

Table 4: Ablation results on hyperparameters evaluated on the MathVerse benchmark.

6 Conclusion and Limitations

This paper investigates the phenomenon of repeated text generation in dMLLMs when using cache, which we term the "Repeat Curse." Through information flow analysis, we reveal that in baseline dMLLMs, context tokens act as anchors to aggregate information and guide predictions. Applying cache disrupts this pattern, leading to repetition. Moreover, repetitive context tokens exhibit persistently high entropy in deeper layers. Building on these insights, we propose CoTA, a plug-and-play approach to mitigate the Repeat Curse. Extensive experiments demonstrate the effectiveness of the CoTA design. Due to the limited research on baseline dMLLMs, CoTA has not yet been validated for generalizability across more open-source dMLLMs and base models of different scales. Moreover, as cache methods for dMLLMs are still scarce, CoTA cannot be tested on a wider range of cache approaches. Future work will address these limitations.

REFERENCES

Josh Achiam, Steven Adler, Sandhini Agarwal, Lama Ahmad, Ilge Akkaya, Florencia Leoni Aleman, Diogo Almeida, Janko Altenschmidt, Sam Altman, Shyamal Anadkat, et al. Gpt-4 technical report. *arXiv preprint arXiv:2303.08774*, 2023.

Shuai Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Sibo Song, Kai Dang, Peng Wang, Shijie Wang, Jun Tang, Humen Zhong, Yuanzhi Zhu, Mingkun Yang, Zhaohai Li, Jianqiang Wan, Pengfei Wang, Wei Ding, Zheren Fu, Yiheng Xu, Jiabo Ye, Xi Zhang, Tianbao Xie, Zesen Cheng, Hang Zhang, Zhibo Yang, Haiyang Xu, and Junyang Lin. Qwen2.5-vl technical report. *arXiv preprint arXiv:2502.13923*, 2025.

Fan Bao, Shen Nie, Kaiwen Xue, Chongxuan Li, Shi Pu, Yaole Wang, Gang Yue, Yue Cao, Hang Su, and Jun Zhu. One transformer fits all distributions in multi-modal diffusion at scale. In *Proceedings of the 40th International Conference on Machine Learning*, volume 202, pp. 1692–1717. PMLR, 23–29 Jul 2023.

Xiao Bi, Deli Chen, Guanting Chen, Shanhuang Chen, Damai Dai, Chengqi Deng, Honghui Ding, Kai Dong, Qiushi Du, Zhe Fu, Huazuo Gao, Kaige Gao, Wenjun Gao, Ruiqi Ge, Kang Guan, Daya Guo, Jianzhong Guo, Guangbo Hao, Zhewen Hao, Ying He, Wenjie Hu, Panpan Huang, Erhang Li, Guowei Li, Jiashi Li, Yao Li, Y. K. Li, Wenfeng Liang, Fangyun Lin, Alex X. Liu, Bo Liu, Wen Liu, Xiaodong Liu, Xin Liu, Yiyuan Liu, Haoyu Lu, Shanghao Lu, Fuli Luo, Shirong Ma, Xiaotao Nie, Tian Pei, Yishi Piao, Junjie Qiu, Hui Qu, Tongzheng Ren, Zehui Ren, Chong Ruan, Zhangli Sha, Zhihong Shao, Junxiao Song, Xuecheng Su, Jingxiang Sun, Yaofeng Sun, Minghui Tang, Bingxuan Wang, Peiyi Wang, Shiyu Wang, Yaohui Wang, Yongji Wang, Tong Wu, Y. Wu, Xin Xie, Zhenda Xie, Ziwei Xie, Yiliang Xiong, Hanwei Xu, R. X. Xu, Yanhong Xu, Dejian Yang, Yuxiang You, Shuiping Yu, Xingkai Yu, B. Zhang, Haowei Zhang, Lecong Zhang, Liyue Zhang, Mingchuan Zhang, Minghua Zhang, Wentao Zhang, Yichao Zhang, Chenggang Zhao, Yao Zhao, Shangyan Zhou, Shunfeng Zhou, Qihao Zhu, and Yuheng Zou. Deepseek llm: Scaling open-source language models with longtermism. *CoRR*, abs/2401.02954, 2024.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Tom Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In *Advances in Neural Information Processing Systems*, 2022.

Liang Chen, Haozhe Zhao, Tianyu Liu, Shuai Bai, Junyang Lin, Chang Zhou, and Baobao Chang. An image is worth 1/2 tokens after layer 2: Plug-and-play inference acceleration for large vision-language models. In *European Conference on Computer Vision*, 2024a.

Lin Chen, Jinsong Li, Xiao wen Dong, Pan Zhang, Yuhang Zang, Zehui Chen, Haodong Duan, Jiaqi Wang, Yu Qiao, Dahua Lin, and Feng Zhao. Are we on the right way for evaluating large vision-language models? *ArXiv*, abs/2403.20330, 2024b.

Zhe Chen, Jiannan Wu, Wenhai Wang, Weijie Su, Guo Chen, Sen Xing, Muyan Zhong, Qinglong Zhang, Xizhou Zhu, Lewei Lu, et al. Internvl: Scaling up vision foundation models and aligning for generic visual-linguistic tasks. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition*, pp. 24185–24198, 2024c.

Wei-Lin Chiang, Zhuohan Li, Zi Lin, Ying Sheng, Zhanghao Wu, Hao Zhang, Lianmin Zheng, Siyuan Zhuang, Yonghao Zhuang, Joseph E Gonzalez, et al. Vicuna: An open-source chatbot impressing gpt-4 with 90%* chatgpt quality. *See https://vicuna. lmsys. org (accessed 14 April 2023)*, 2(3):6, 2023.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. Bert: Pre-training of deep bidirectional transformers for language understanding. *arXiv preprint arXiv:1810.04805*, 2018.

Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, Anirudh Goyal, Anthony Hartshorn, Aobo Yang, Archi Mitra, Archie Sravankumar, Artem Korenev, Arthur Hinsvark, Arun Rao, Aston Zhang, Aurélien Rodriguez, Austen Gregerson, Ava Spataru, Baptiste Rozière, Bethany Biron, Binh Tang, Bobbie Chern, Charlotte Caucheteux, Chaya Nayak, Chloe Bi, Chris Marra, Chris McConnell, Christian Keller, Christophe Touret, Chunyang Wu, Corinne Wong, Cristian Canton Ferrer, Cyrus Nikolaidis, Damien Allonsius, Daniel Song, Danielle Pintz, Danny Livshits, David Esiobu, Dhruv Choudhary, Dhruv Mahajan, Diego Garcia-Olano, Diego Perino, Dieuwke Hupkes, Egor Lakomkin, Ehab AlBadawy, Elina Lobanova, Emily Dinan, Eric Michael Smith, Filip Radenovic, Frank Zhang, Gabriel Synnaeve, Gabrielle Lee, Georgia Lewis Anderson, Graeme Nail, Grégoire Mialon, Guan Pang, Guillem Cucurell, Hailey Nguyen, Hannah Korevaar, Hu Xu, Hugo Touvron, Iliyan Zarov, Imanol Arrieta Ibarra, Isabel M. Kloumann, Ishan Misra, Ivan Evtimov, Jade Copet, Jaewon Lee, Jan Geffert, Jana Vranes, Jason Park, Jay Mahadeokar, Jeet Shah, Jelmer van der Linde, Jennifer Billock, Jenny Hong, Jenya Lee, Jeremy Fu, Jianfeng Chi, Jianyu Huang, Jiawen Liu, Jie Wang, Jiecao Yu, Joanna Bitton, Joe Spisak, Jongsoo Park, Joseph Rocca, Joshua Johnstun, Joshua Saxe, Junteng Jia, Kalyan Vasuden Alwala, Kartikeya Upasani, Kate Plawiak, Ke Li, Kenneth Heafield, Kevin Stone, and et al. The llama 3 herd of models. *CoRR*, abs/2407.21783, 2024.

Patrick Esser, Sumith Kulal, Andreas Blattmann, Rahim Entezari, Jonas Müller, Harry Saini, Yam Levi, Dominik Lorenz, Axel Sauer, Frederic Boesel, Dustin Podell, Tim Dockhorn, Zion English, and Robin Rombach. Scaling rectified flow transformers for high-resolution image synthesis. In *Forty-first International Conference on Machine Learning*, 2024.

Chaoyou Fu, Peixian Chen, Yunhang Shen, Yulei Qin, Mengdan Zhang, Xu Lin, Zhenyu Qiu, Wei Lin, Jinrui Yang, Xiawu Zheng, Ke Li, Xing Sun, and Rongrong Ji. Mme: A comprehensive evaluation benchmark for multimodal large language models. *ArXiv*, abs/2306.13394, 2023.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Antonio A. Ginart, Naveen Kodali, Jason Lee, Caiming Xiong, Silvio Savarese, and John R. Emmons. Lz penalty: An information-theoretic repetition penalty for autoregressive language models, 2025.

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. Scaling diffusion language models via adaptation from autoregressive models. In *The Thirteenth International Conference on Learning Representations*, 2025.

Suriya Gunasekar, Yi Zhang, Jyoti Aneja, Caio César Teodoro Mendes, Allie Del Giorno, Sivakanth Gopi, Mojan Javaheripi, Piero Kauffmann, Gustavo de Rosa, Olli Saarikivi, Adil Salim, Shital Shah, Harkirat Singh Behl, Xin Wang, Sébastien Bubeck, Ronen Eldan, Adam Tauman Kalai, Yin Tat Lee, and Yuanzhi Li. Textbooks are all you need. *CoRR*, abs/2306.11644, 2023.

Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural text degeneration. In *International Conference on Learning Representations*, 2020.

Qidong Huang, Xiao wen Dong, Pan Zhang, Bin Wang, Conghui He, Jiaqi Wang, Dahua Lin, Weiming Zhang, and Neng H. Yu. Opera: Alleviating hallucination in multi-modal large language models via over-trust penalty and retrospection-allocation. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 13418–13427, 2023.

Mingyu Jin, Kai Mei, Wujiang Xu, Mingjie Sun, Ruixiang Tang, Mengnan Du, Zirui Liu, and Yongfeng Zhang. Massive values in self-attention modules are the key to contextual knowledge understanding. *ArXiv*, abs/2502.01563, 2025.

Woosuk Kwon, Zhuohan Li, Siyuan Zhuang, Ying Sheng, Lianmin Zheng, Cody Hao Yu, Joseph E. Gonzalez, Hao Zhang, and Ion Stoica. Efficient memory management for large language model serving with pagedattention. In *Proceedings of the ACM SIGOPS 29th Symposium on Operating Systems Principles*, 2023.

Evgeny Lagutin, Daniil Gavrilov, and Pavel Kalaidin. Implicit unlikelihood training: Improving neural text generation with reinforcement learning. In *Proceedings of the 16th Conference of the European Chapter of the Association for Computational Linguistics: Main Volume*, pp. 1432–1441, 2021.

Bo Li, Yuanhan Zhang, Dong Guo, Renrui Zhang, Feng Li, Hao Zhang, Kaichen Zhang, Peiyuan Zhang, Yanwei Li, Ziwei Liu, and Chunyuan Li. LLaVA-onevision: Easy visual task transfer. *Transactions on Machine Learning Research*, 2025a. ISSN 2835-8856.

Bohao Li, Rui Wang, Guangzhi Wang, Yuying Ge, Yixiao Ge, and Ying Shan. Seed-bench: Benchmarking multimodal llms with generative comprehension. *ArXiv*, abs/2307.16125, 2023a.

Huayang Li, Tian Lan, Zihao Fu, Deng Cai, Lemao Liu, Nigel Collier, Taro Watanabe, and Yixuan Su. Repetition in repetition out: Towards understanding neural text degeneration from the data perspective. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b.

Shufan Li, Konstantinos Kallidromitis, Hritik Bansal, Akash Gokul, Yusuke Kato, Kazuki Kozuka, Jason Kuen, Zhe Lin, Kai-Wei Chang, and Aditya Grover. Lavida: A large diffusion language model for multimodal understanding, 2025b.

Xiang Lisa Li, John Thickstun, Ishaan Gulrajani, Percy Liang, and Tatsunori Hashimoto. Diffusion-LM improves controllable text generation. In Alice H. Oh, Alekh Agarwal, Danielle Belgrave, and Kyunghyun Cho (eds.), *Advances in Neural Information Processing Systems*, 2022.

Zijie Li, Henry Li, Yichun Shi, Amir Barati Farimani, Yuval Kluger, Linjie Yang, and Peng Wang. Dual diffusion for unified image generation and understanding, 2025c.

Tsung-Yi Lin, Michael Maire, Serge J. Belongie, James Hays, Pietro Perona, Deva Ramanan, Piotr Dollár, and C. Lawrence Zitnick. Microsoft coco: Common objects in context. In *European Conference on Computer Vision*, 2014.

Zhenghao Lin, Yeyun Gong, Yelong Shen, Tong Wu, Zhihao Fan, Chen Lin, Nan Duan, and Weizhu Chen. Text generation with diffusion language models: A pre-training approach with continuous paragraph denoise. In *ICML*, pp. 21051–21064, 2023.

Haotian Liu, Chunyuan Li, Yuheng Li, and Yong Jae Lee. Improved baselines with visual instruction tuning. *2024 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26286–26296, 2023a.

Haotian Liu, Chunyuan Li, Qingyang Wu, and Yong Jae Lee. Visual instruction tuning. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023b.

Yuanzhan Liu, Haodong Duan, Yuanhan Zhang, Bo Li, Songyang Zhang, Wangbo Zhao, Yike Yuan, Jiaqi Wang, Conghui He, Ziwei Liu, Kai Chen, and Dahua Lin. Mmbench: Is your multi-modal model an all-around player? *ArXiv*, abs/2307.06281, 2023c.

Zhiyuan Liu, Yicun Yang, Yaojie Zhang, Junjie Chen, Chang Zou, Qingyuan Wei, Shaobo Wang, and Linfeng Zhang. dllm-cache: Accelerating diffusion large language models with adaptive caching. *ArXiv*, abs/2506.06295, 2025.

Justin Lovelace, Varsha Kishore, Chao Wan, Eliot Shekhtman, and Kilian Q. Weinberger. Latent diffusion for language generation. In *NeurIPS*, 2023.

Pan Lu, Hritik Bansal, Tony Xia, Jiacheng Liu, Chun yue Li, Hannaneh Hajishirzi, Hao Cheng, Kai-Wei Chang, Michel Galley, and Jianfeng Gao. Mathvista: Evaluating math reasoning in visual contexts with gpt-4v, bard, and other large multimodal models. *ArXiv*, abs/2310.02255, 2023.

Yiyang Ma, Xingchao Liu, Xiaokang Chen, Wen Liu, Chengyue Wu, Zhiyu Wu, Zizheng Pan, Zhenda Xie, Haowei Zhang, Xingkai Yu, Liang Zhao, Yisong Wang, Jiaying Liu, and Chong Ruan. Janusflow: Harmonizing autoregression and rectified flow for unified multimodal understanding and generation. *CoRR*, abs/2411.07975, 2024.

Ahmed Masry, Do Xuan Long, Jia Qing Tan, Shafiq R. Joty, and Enamul Hoque. Chartqa: A benchmark for question answering about charts with visual and logical reasoning. *ArXiv*, abs/2203.10244, 2022.

Minesh Mathew, Dimosthenis Karatzas, R. Manmatha, and C. V. Jawahar. Docvqa: A dataset for vqa on document images. *2021 IEEE Winter Conference on Applications of Computer Vision (WACV)*, pp. 2199–2208, 2020.

Shen Nie, Fengqi Zhu, Zebin You, Xiaolu Zhang, Jingyang Ou, Jun Hu, Jun Zhou, Yankai Lin, Ji-Rong Wen, and Chongxuan Li. Large language diffusion models. *CoRR*, abs/2502.09992, February 2025.

Dustin Podell, Zion English, Kyle Lacey, Andreas Blattmann, Tim Dockhorn, Jonas Müller, Joe Penna, and Robin Rombach. SDXL: Improving latent diffusion models for high-resolution image synthesis. In *The Twelfth International Conference on Learning Representations*, 2024.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *2022 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 10674–10685, 2022.

Litu Rout, Constantine Caramanis, and Sanjay Shakkottai. Anchored diffusion language model. In *The Thirty-ninth Annual Conference on Neural Information Processing Systems*, 2025.

Alexander Swerdlow, Mihir Prabhudesai, Siddharth Gandhi, Deepak Pathak, and Katerina Fragkiadaki. Unified multimodal discrete diffusion. *arXiv preprint arXiv:2503.20853*, 2025. doi: 10.48550/arXiv.2503.20853.

Feilong Tang, Zile Huang, Chengzhi Liu, Qiang Sun, Harry Yang, and Ser-Nam Lim. Intervening anchor token: Decoding strategy in alleviating hallucinations for MLLMs. In *The Thirteenth International Conference on Learning Representations*, 2025a.

Feilong Tang, Chengzhi Liu, Zhongxing Xu, Ming Hu, Zile Huang, Haochen Xue, Ziyang Chen, Zelin Peng, Zhiwei Yang, Sijin Zhou, Wenxue Li, Yulong Li, Wenxuan Song, Shiyan Su, Wei Feng, Jionglong Su, Mingquan Lin, Yifan Peng, Xuelian Cheng, Imran Razzak, and Zongyuan Ge. Seeing far and clearly: Mitigating hallucinations in mllms with attention causal decoding. In *Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26147–26159, June 2025b.

Feilong Tang, Chengzhi Liu, Zhongxing Xu, Ming Hu, Zelin Peng, Zhiwei Yang, Jionglong Su, Minquan Lin, Yifan Peng, Xuelian Cheng, Imran Razzak, and Zongyuan Ge. Seeing far and clearly: Mitigating hallucinations in mllms with attention causal decoding. *2025 IEEE/CVF Conference on Computer Vision and Pattern Recognition (CVPR)*, pp. 26147–26159, 2025c.

Bingkui Tong, Jiaer Xia, and Kaiyang Zhou. Mitigating hallucination in multimodal llms with layer contrastive decoding, 2025.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, Aurélien Rodriguez, Armand Joulin, Edouard Grave, and Guillaume Lample. Llama: Open and efficient foundation language models. *CoRR*, abs/2302.13971, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton-Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurélien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *CoRR*, abs/2307.09288, 2023b.

Michael Tschannen, Alexey Gritsenko, Xiao Wang, Muhammad Ferjad Naeem, Ibrahim M. Alabdulmohsin, Nikhil Parthasarathy, Talfan Evans, Lucas Beyer, Ye Xia, Basil Mustafa, Olivier H'enaff, Jeremiah Harmsen, Andreas Steiner, and Xiao-Qi Zhai. Siglip 2: Multilingual vision-language encoders with improved semantic understanding, localization, and dense features. *ArXiv*, abs/2502.14786, 2025.

Lean Wang, Lei Li, Damai Dai, Deli Chen, Hao Zhou, Fandong Meng, Jie Zhou, and Xu Sun. Label words are anchors: An information flow perspective for understanding in-context learning. *ArXiv*, abs/2305.14160, 2023.

Peng Wang, Shuai Bai, Sinan Tan, Shijie Wang, Zhihao Fan, Jinze Bai, Keqin Chen, Xuejing Liu, Jialin Wang, Wenbin Ge, Yang Fan, Kai Dang, Mengfei Du, Xuancheng Ren, Rui Men, Dayiheng Liu, Chang Zhou, Jingren Zhou, and Junyang Lin. Qwen2-vl: Enhancing vision-language model's perception of the world at any resolution. *arXiv preprint arXiv:2409.12191*, 2024a.

Weichuan Wang, Zhaoyi Li, Defu Lian, Chen Ma, Linqi Song, and Ying Wei. Mitigating the language mismatch and repetition issues in llm-based machine translation via model editing. In *EMNLP*, pp. 15681–15700, 2024b.

Sean Welleck, Ilia Kulikov, Stephen Roller, Emily Dinan, Kyunghyun Cho, and Jason Weston. Neural text generation with unlikelihood training. In *International Conference on Learning Representations*, 2020.

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

Guangxuan Xiao and et al. Efficient streaming language models with attention sinks. In *International Conference on Learning Representations (ICLR)*, 2024.

Guangxuan Xiao, Yuandong Tian, Beidi Chen, Song Han, and Mike Lewis. Efficient streaming language models with attention sinks. *ArXiv*, abs/2309.17453, 2023.

Jinheng Xie, Weijia Mao, Zechen Bai, David Junhao Zhang, Weihao Wang, Kevin Qinghong Lin, Yuchao Gu, Zhijie Chen, Zhenheng Yang, and Mike Zheng Shou. Show-o: One single transformer to unify multimodal understanding and generation. In *The Thirteenth International Conference on Learning Representations*, 2025.

Jin Xu, Xiaojiang Liu, Jianhao Yan, Deng Cai, Huayang Li, and Jian Li. Learning to break the loop: Analyzing and mitigating repetitions for neural text generation. In *NeurIPS*, 2022.

Kaiwen Xue, Yuhao Zhou, Shen Nie, Xu Min, Xiaolu Zhang, JUN ZHOU, and Chongxuan Li. Unifying bayesian flow networks and diffusion models through stochastic differential equations. In *Forty-first International Conference on Machine Learning*, 2024.

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, Junyang Lin, Kai Dang, Keming Lu, Keqin Bao, Kexin Yang, Le Yu, Mei Li, Mingfeng Xue, Pei Zhang, Qin Zhu, Rui Men, Runji Lin, Tianhao Li, Tingyu Xia, Xingzhang Ren, Xuancheng Ren, Yang Fan, Yang Su, Yichang Zhang, Yu Wan, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zihan Qiu. Qwen2.5 technical report. *CoRR*, abs/2412.15115, 2024a.

Haoran Yang, Deng Cai, Huayang Li, Wei Bi, Wai Lam, and Shuming Shi. A frustratingly simple decoding method for neural text generation. In *LREC/COLING*, pp. 536–557, 2024b.

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

Junchi Yao, Shu Yang, Jianhua Xu, Lijie Hu, Mengdi Li, and Di Wang. Understanding the repeat curse in large language models from a feature perspective. In *Annual Meeting of the Association for Computational Linguistics*, 2025.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b, 2025. URL `https://hkunlp.github.io/blog/2025/dream`.

Jiasheng Ye, Zaixiang Zheng, Yu Bao, Lihua Qian, and Quanquan Gu. Diffusion language models can perform many tasks with scaling and instruction-finetuning. *ArXiv*, abs/2308.12219, 2023.

Itay Yona, Ilia Shumailov, Jamie Hayes, Federico Barbero, and Yossi Gandelsman. Interpreting the repeated token phenomenon in large language models. *ArXiv*, abs/2503.08908, 2025.

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

Runpeng Yu, Xinyin Ma, and Xinchao Wang. Dimple: Discrete diffusion multimodal large language model with parallel decoding, 2025.

Zhongzhi Yu, Zheng Wang, Yonggan Fu, Huihong Shi, Khalid Shaikh, and Yingyan Celine Lin. Unveiling and harnessing hidden attention sinks: Enhancing large language models without training through attention calibration. *ArXiv*, abs/2406.15765, 2024.

Ruixiang ZHANG, Shuangfei Zhai, Yizhe Zhang, James Thornton, Zijing Ou, Joshua M. Susskind, and Navdeep Jaitly. Target concrete score matching: A holistic framework for discrete diffusion. In *Forty-second International Conference on Machine Learning*, 2025.

Xiaofeng Zhang, Yihao Quan, Chen Shen, Xiaosong Yuan, Shaotian Yan, Liang Xie, Wenxiao Wang, Chaochen Gu, Hao Tang, and Jieping Ye. From redundancy to relevance: Information flow in lvlms across reasoning tasks. In *North American Chapter of the Association for Computational Linguistics*, 2024.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. In *The Thirteenth International Conference on Learning Representations*, 2025.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *ArXiv*, abs/2302.05737, 2023.

Chunting Zhou, LILI YU, Arun Babu, Kushal Tirumala, Michihiro Yasunaga, Leonid Shamis, Jacob Kahn, Xuezhe Ma, Luke Zettlemoyer, and Omer Levy. Transfusion: Predict the next token and diffuse images with one multi-modal model. In *The Thirteenth International Conference on Learning Representations*, 2025.

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

Jinguo Zhu, Weiyun Wang, Zhe Chen, Zhaoyang Liu, Shenglong Ye, Lixin Gu, Hao Tian, Yuchen Duan, Weijie Su, Jie Shao, Zhangwei Gao, Erfei Cui, Xuehui Wang, Yue Cao, Yangzhou Liu, Xingguang Wei, Hongjie Zhang, Haomin Wang, Weiye Xu, Hao Li, Jiahao Wang, Nianchen Deng, Songze Li, Yinan He, Tan Jiang, Jiapeng Luo, Yi Wang, Conghui He, Botian Shi, Xingcheng Zhang, Wenqi Shao, Junjun He, Yingtong Xiong, Wenwen Qu, Peng Sun, Penglong Jiao, Han Lv, Lijun Wu, Kaipeng Zhang, Huipeng Deng, Jiaye Ge, Kai Chen, Limin Wang, Min Dou, Lewei Lu, Xizhou Zhu, Tong Lu, Dahua Lin, Yu Qiao, Jifeng Dai, and Wenhai Wang. Internvl3: Exploring advanced training and test-time recipes for open-source multimodal models, 2025b.

Wenhong Zhu, Hongkun Hao, and Rui Wang. Penalty decoding: Well suppress the self-reinforcement effect in open-ended text generation. In *The 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

(a) Effect of prompt token recompute interval.

Interval	1	5	15	25
SRR↓	0	0	0	0

(b) Effect of output token recompute interval.

Interval	1	3	5	7
SRR↓	0	79.9	87.4	89.7

(c) Effect of similarity thresholds.

Threshold	0	0.25	0.5	0.75	1
SRR ↓	89.7	75.0	69.8	29.7	0

(d) Comparison of reuse policies.

Policy	prefix KV cache	dllms-cache
SRR↓	0	75.0

Table 5: Ablation study on cache mechanism design components and their effects on repetition.

Method	ARR↓	SRR↓
MMaDA	0.7	6.0
+dllm-cache	4.4	55.0
+dllm-cache+CTAE	2.4	29.0
+dllm-cache+CTEV	0.8	35.0
+dllm-cache+CoTA	0.6	30.0

(a) Repetition curse across different dMLLMs.

Method	ARR↓	SRR↓
LLaDA-V	0	0
+dllm-cache	9.3	75.0
+prefix KV cache	0	0
LaViDa	0	0
+prefix KV cache	0	0

(b) Repetition curse under different cache methods.

Table 6: Generalization experiments.

A Analysis of Cache Mechanism and Repetition Curse

We conduct a series of ablation studies on LLaDA-V to analyze how different components of the cache mechanism influence repetition curse.

(a) We begin by examining the prompt token recomputation interval, while fixing the output token recomputation interval to 1 and setting the similarity threshold to 0. As shown in Table 5a, periodic caching of prompt tokens has negligible impact on repetition.

(b) We then fix the prompt token recomputation interval to 25 and set the similarity threshold to 0, varying only the output token recomputation interval. As reported in Table 5b, we observe that periodic caching of output tokens induces repetition, and longer recomputation intervals lead to higher repetition rates.

(c) In the dLLM-Cache framework, a subset of output tokens is adaptively recomputed at each step based on a similarity threshold. To analyze this further, we fix the prompt token recomputation interval to 25 and the output token recomputation interval to 7, while varying the similarity threshold. As shown in Table 5c, the threshold plays a critical role: lower thresholds correspond to higher repetition rates.

(d) Finally, we compare dLLM-Cache with prefix KV cache Kwon et al. (2023); Li et al. (2025b) to analyze the effect of different reuse policies. We fix the prompt token recomputation interval to 25, the output token recomputation interval to 7, and the similarity threshold to 0.25. As shown in Table 5d, prefix KV cache, which only reuses cached states for prefix tokens, does not trigger repetition.

These four experiments reveal that the prompt recomputation interval has little effect on repetition behavior, while the output token recomputation interval, similarity threshold, and caching policy exert a substantial influence. In general, the fewer output tokens that are recomputed at each decoding step, the more likely the model is to fall into repetition.

We hypothesize that this is because the token states within the prompt sequence remain relatively static, so caching them has minimal impact on future predictions. In contrast, output tokens evolve across decoding steps. Caching these dynamic tokens forces the model to rely on outdated states, reducing uncertainty during prediction—ultimately contributing to the non-convergence of context-token entropy in deeper layers.

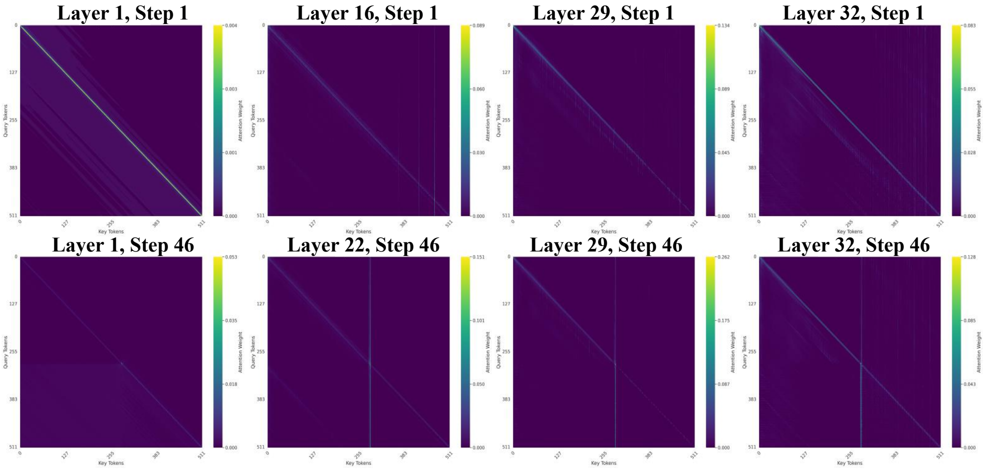


Figure 6: Information flow visualization for MMaDA-8B without using cache.

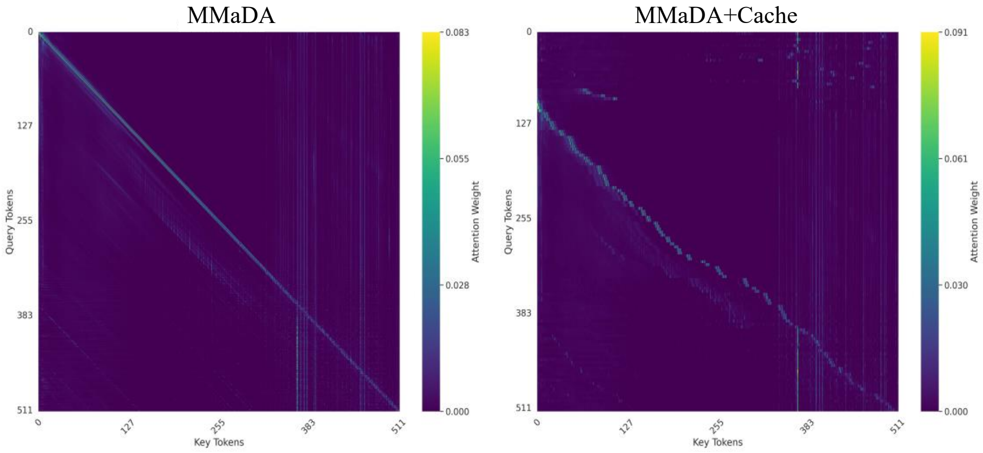


Figure 7: Comparison of MMaDA's information flow patterns with and without cache.

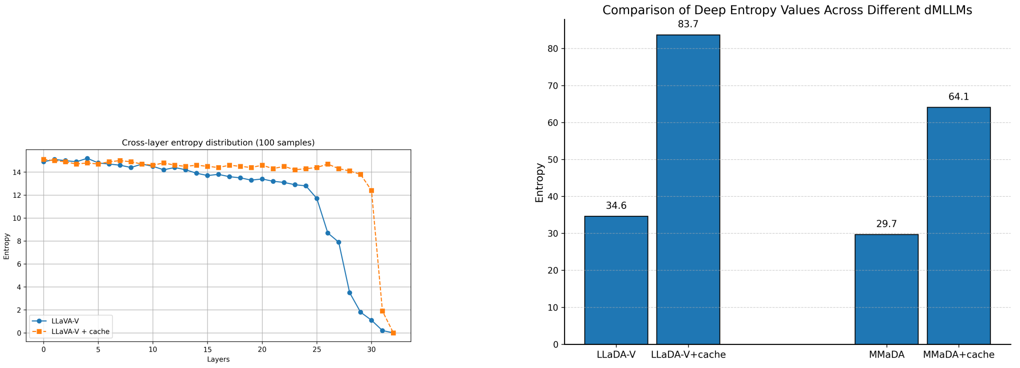
B More Results

B.1 Repetition Curse Across Different dMLLMs

We present the repetition analysis of applying the cache mechanism to MMaDA in Table 6a. Notably, we find that MMaDA also suffers from the repetition curse when dLLM-Cache is enabled. Specifically, the adjacent repetition rate increases by +3.7%, and the sample repetition rate increases by +49%. These findings highlight that the repetition curse is not limited to a single architecture, but is a pervasive issue across dMLLMs.

B.2 Information Flow Across Different dMLLMs

As illustrated in Figure 6, we visualize the attention maps of MMaDA to analyze the flow of information between tokens. We observe a similar pattern to LLaDA-V, where context tokens function as anchors, continually attracting attention across layers. Interestingly, we also detect vertical bands of concentrated attention in Layer 22 at Step 46, resembling the "attention sink" Huang et al. (2023) phenomenon commonly seen in autoregressive MLLMs. However, after introducing the cache, the original attention pattern of MMaDA is also disrupted (as shown in Figure 7).



(a) Cross-layer entropy distribution across 100 samples.

(b) Comparison of total deep-layer entropy across different dMLLMs.

Figure 8: Deep-layer entropy analysis experiment.

B.3 Generalization Experiments of CoTA

We apply CoTA to the MMaDA model, which exhibits repetition after enabling cache, and present the results in Table 6a. The results show that CoTA and its components (CTAE and CTEV) effectively suppress repetitive text generation in MMaDA as well, demonstrating the generalizability of our approach.

B.4 Repetition Curse Under Different Cache Methods

As shown in Table 6b, we further assess the repetition behavior when applying the prefix-KV cache method Kwon et al. (2023) to both LLaDA-V and LaViDa Li et al. (2025b). Notably, the prefix-KV cache does not trigger repetition in either model. We speculate that this is because prefix-KV cache only reuses the cached states of prefix tokens (prompt and image tokens), without affecting suffix tokens (output text tokens). This finding is consistent with the conclusions presented in Section A.

C More experimental details of CTEV

In CTEV, a central design choice is the selection of layers 26–30 for entropy estimation. This choice is informed by both multi-sample statistics and detailed single-sample analyses. Concretely, we compute per-layer entropy over 100 samples, and then aggregate and average the entropy values for each layer across samples. The resulting entropy distribution across layers is shown in Figure 8a. A comparison of the entropy trajectories before and after enabling the cache reveals a clear pattern: cache usage induces pronounced entropy deviations specifically in layers 26–30. As discussed in Section 3.3, these layers consistently exhibit the strongest sensitivity to cache-induced shifts, which directly motivates their adoption in CTEV.

To further validate this choice, we examine the cumulative entropy over layers 26–30 across different dMLLM architectures. As shown in Figure 8b, all models display a consistent trend: caching systematically disrupts entropy convergence in deep layers. This cross-model agreement provides strong empirical support for focusing on layers 26–30 within the CTEV framework.

Methods	**16**	**32**	**64**	**128**
LLaDA-V	2%	1%	1%	1%
LLaDA-V+cache	88%	85%	79%	75%
LLaDA-V+CoTA	17%	14%	10%	8%

Table 7: Repeat curse comparison under different block lengths. The evaluation metric is the sample repetition rate.

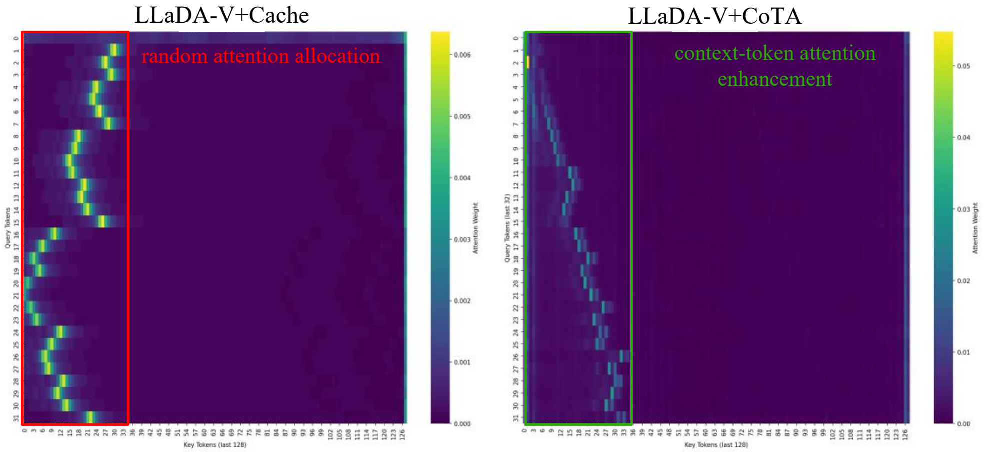


Figure 9: Attention map visualizations of LLaDA-V+Cache and LLaDA-V+CoTA.

We further analyze the sensitivity of CTEV through additional experiments. As shown in Table 7, we evaluate the CoTA performance under different block lengths during decoding. We observe that our method consistently reduces the repetition rate across all tested block configurations, demonstrating its robustness to different window sizes.

Layer range	**512**		**128**	
	ARR↓	**SRR↓**	**ARR↓**	**SRR↓**
LLaDA-V+cache	13.9%	85%	9.3%	75%
1–10	12.1%	81%	8.4%	73%
11–20	12.4%	83%	9.1%	74%
21–25	11.3%	73%	7.9%	65%
25–30	5.6%	23%	4.3%	19%
26–30	2.9%	10%	1.8%	8%
26–31	3.2%	12%	2.1%	10%
26–32	3.3%	12%	2.1%	9%

Table 8: Performance comparison under different layer ranges.

We further present results across different layer ranges in Table 8. We find that computing the cumulative entropy using layers 26–30 yields the best performance. We speculate that this is because the entropy values in this depth range exhibit the largest separation between normal and abnormal modes, while incorporating additional layers introduces unnecessary noise, leading to diminished performance.

D Additional Analyses of CoTA

As shown in Figure 9, we visualize the attention maps to analyze the changes in the model's internal information-flow patterns. After applying Cache, the model exhibits a more random attention distribution, whereas CoTA restores the "context tokens as anchors" information-flow pattern by strengthening attention toward contextual tokens.

E Linguistic Attribute Analysis of Repeated Tokens

In this section, we analyze the linguistic attributes of repeated tokens by computing their frequency statistics over 200 samples that exhibit repetition. As shown in Table 9, the top-3 most frequently repeated words are "the," "of," and "a." We observe that repeated tokens are predominantly function

	LLaDA-V+cache		
Repeated word	**the**	**of**	**a**
Repetition ratio	98%	76%	56%

Table 9: Statistics of repeated words generated by LLaDA-V with cache.

Methods	**ACC↑**	**ARR↓**
LLaDA-V+cache	54.9	6.9
n-gram (n=2)	54.0	2.4
n-gram (n=3)	54.3	3.2
CoTA	59.7	1.3

(a) Comparison between CoTA and n-gram penalties.

Methods	**ARR↓**	**SRR↓**
Decay Matrix	1.8%	8%
AliBI Bias Matrix	3.9%	10%

(b) Comparison of different attention biasing methods.

Table 10: Experimental results of CoTA and related techniques.

words carrying low semantic content, which is also consistent with the case study presented in Figure 1. This indicates a common tendency of the model when generating uncertain or repetitive text.

F Discussion of CoTA and Related Techniques

N-gram penalties Holtzman et al. (2020), as a simple and widely used decoding-time technique, are commonly applied to control repetition. We further compare the performance of CoTA and n-gram penalties in Table 10a. We find that although n-gram penalties can reduce repetition, their effectiveness is inferior to that of CoTA. Moreover, applying n-gram penalties often leads to a degradation in model accuracy, likely due to forced token substitution, which disrupts semantic coherence and lowers output quality. In addition, we replace the decay matrix used in CTAE's attention biasing design with an Alibi bias penalty matrix Tang et al. (2025b)to evaluate its effect. As shown in Table 10b, the results confirm that using the Decay Matrix for attention intervention in CTAE is more effective.

G More Efficiency Analysis

We further perform a comprehensive efficiency analysis on DocVQA, covering prefix-KV, dLLM-cache, CoTA, and its two components (CTAE and CTEV). As shown in Table F, integrating CoTA introduces only a modest and acceptable increase in latency, along with a slight reduction in throughput, yet it still delivers substantial improvements over the baseline. Moreover, to provide a more holistic assessment of efficiency, we additionally report latency and throughput results on both LLaDA-V and MMaDA.

H Related Work

Diffusion-based Multimodal Large Language Models (dMLLMs). The latest diffusion-based large language models (dLLMs) Nie et al. (2025); Zhu et al. (2025a); Ye et al. (2025); Gong et al. (2025) have been successfully scaled to 8B parameters, achieving performance comparable to state-of-the-art autoregressive large language models Touvron et al. (2023a;b); Devlin et al. (2018); Dubey et al. (2024); Bi et al. (2024); Yang et al. (2024a); Gunasekar et al. (2023); Chiang et al. (2023). Furthermore, by integrating visual instruction tuning and architectural extensions, dLLMs have recently been extended to dMLLMs Nie et al. (2025); Zhu et al. (2025a); Ye et al. (2025); Gong et al. (2025), demonstrating promising multimodal capabilities.

Diffusion Language Models. Motivated by the remarkable success of diffusion models in image generation Rombach et al. (2022); Podell et al. (2024); Esser et al. (2024), recent studies have introduced the diffusion process into language modeling, including both continuous diffusion mod-

Method	throughput (tok/s) ↑	latency (s/sample) ↓
LLaDA-V	1.8	17.4
+prefix KV cache	2.7	13.3
+dllm-cache	4.9	6.1
+dllm-cache+CTEV	4.1	7.2
+dllm-cache+CTAE	4.6	6.4
+dllm-cache+CoTA	3.8	7.7
MMaDA	0.5	32.1
+dllm-cache	2.0	19.6
+dllm-cache+CTEV	1.5	22.0
+dllm-cache+CTAE	1.7	21.4
+dllm-cache+CoTA	1.3	24.0

Table 11: Throughput and latency comparison on DocVQA.

els Lovelace et al. (2023); Li et al. (2022); ZHANG et al. (2025); Xue et al. (2024); Lin et al. (2023) and discrete diffusion models Campbell et al. (2022); Zheng et al. (2025); Gat et al. (2024); Ye et al. (2023); Zheng et al. (2023). The latest masked diffusion models Nie et al. (2025); Zhu et al. (2025a); Ye et al. (2025); Gong et al. (2025) have successfully scaled dLLMs to 8B parameters and demonstrated performance comparable to autoregressive large language models Yang et al. (2024a).

Multimodal Large Language Models (MLLMs). Building upon the unprecedented generative capabilities of large language models (LLMs)Touvron et al. (2023a;b); Devlin et al. (2018); Dubey et al. (2024); Bi et al. (2024); Yang et al. (2024a); Gunasekar et al. (2023); Chiang et al. (2023), MLLMs have emerged as powerful systems that extend LLM architectures to process over multimodal inputs. Probabilistic modeling approaches for MLLMs generally fall into three paradigms: (i) autoregressiveLiu et al. (2023b;a); Bai et al. (2025); Wang et al. (2024a); Zhu et al. (2025b); Chen et al. (2024c); Achiam et al. (2023); Li et al. (2025a), (ii) autoregressive–diffusion hybrid Bao et al. (2023); Xie et al. (2025); Zhou et al. (2025); Ma et al. (2024); Yu et al. (2025), and (iii) pure diffusion Swerdlow et al. (2025); Li et al. (2025c). Building on the recent breakthroughs in dLLMs, the latest dMLLMs You et al. (2025); Yang et al. (2025); Li et al. (2025b) exploit the language modeling capabilities of dLLMs, coupled with effective training pipelines, to deliver performance on par with leading autoregressive and hybrid counterparts.

Repeated Token Generation. Early research on text repetition and degeneration in model predictions primarily focused on improving sampling strategies for language models Holtzman et al. (2020), such as nucleus sampling and top-k sampling. Several studies treat repetition control as a training objective and introduce various training-time strategies to mitigate repetitive generation Welleck et al. (2020); Lagutin et al. (2021); Xu et al. (2022). Other methods focus on incorporating repetition-aware penalties during the sampling stage Zhu et al. (2023).

In recent years, research on repetitive text generation has expanded from conventional language generation models to modern LLMs. Yang et al. (2024b) estimates the likelihood of future repetition using an n-gram LM constructed from the generated prefix and imposes penalties based on this estimate. Ginart et al. (2025) introduces a repetition penalty from a compression-based perspective, while Li et al. (2023b) highlights how repeated tokens in training corpora can exacerbate repetition during inference.

Some recent work further investigates the underlying mechanisms behind repetition. DUC Yao et al. (2025) employs Sparse Autoencoders (SAEs) to identify latent features that become highly activated when a model produces repeated tokens, referred to as repetition features. They attribute repetition to the activation of these features at specific layers and mitigate repetition by suppressing them. Wang et al. (2024b) employs model-editing techniques to locate FFN neurons strongly associated with repetition, as well as neurons strongly related to the main task. By intersecting these neuron sets, they filter out neurons purely tied to repetition and perform targeted edits.

Distinct from the above approaches, our work examines repetition from an information-flow perspective. By comparing the information-flow patterns that emerge during repetitive generation with those observed under normal conditions, we identify the underlying mechanisms that lead to repetition. Our analysis shows that in dMLLMs, context tokens act as anchors that aggregate and

Evaluation data	DocVQA	ChartQA	MMStar	MME	SEED
Max generation length	32	16	2	2	2
Block length	32	16	2	2	2
Decode steps	16	16	2	2	2
Batchsize	1	1	1	1	1

Table 12: Evaluation configuration details of different datasets.

propagate information. However, the introduction of cache disrupts this aggregation pattern and consequently triggers repetitive outputs. In addition, our layer-wise entropy analysis reveals that context tokens involved in local repetition exhibit a failure of entropy convergence in deeper layers. It is important to note that existing research on repetitive outputs in MLLMs and dMLLMs remains limited. Prior studies on MLLMs discuss repetition primarily in the context of hallucination and demonstrate that certain decoding strategies can reduce this effect Tong et al. (2025); Huang et al. (2023). Recent dMLLM work has reported the presence of repetition but has not conducted deeper investigation into its causes Yu et al. (2025).

I Evaluation configuration details of different datasets

We report detailed information about the evaluation setup in Table 12, including the maximum generation length, block length, decoding steps, and batch size for different benchmarks.

J Maximum Repetition Length, Average Repetition Length, 95th-percentile Repetition Length, and Sample Repetition Rate

Given an input sequence of length M, $\{\mathcal{T}_i\}_{i=1}^{M}$, we record the length of each consecutive identical segment r_k as follows:

$$r_k = \left|\{\, \mathcal{T}_i \;\mid\; \mathcal{T}_i = \mathcal{T}_{i+1} = \cdots = \mathcal{T}_{i+r_k-1} \,\}\right|, \quad k = 1, 2, \ldots, K, \tag{11}$$

where K denotes the total number of segments. Furthermore, we obtain all token repetition segments rep_runs as follows:

$$rep\_runs = \{\, r_k \;\mid\; r_k \geq 2,\; k = 1, 2, \ldots, K \,\}. \tag{12}$$

Based on rep_runs, the maximum repetition length (MRL), average repetition length (ARL), and 95th-percentile repetition length (95pRL) can be obtained from Equations 13, 14, and 15, respectively.

$$\text{MRL} = \max(\text{rep\_runs}). \tag{13}$$

$$\text{ARL} = \frac{1}{|\text{rep\_runs}|} \sum_{r \in \text{rep\_runs}} r. \tag{14}$$

$$\text{95pRL} = \text{Quantile}_{0.95}(\text{rep\_runs}). \tag{15}$$

Let the total number of generated results be N, among which the number of results containing repetition is N_{dup}. The Sample Repetition Rate is defined as:

$$\text{SRR} = \frac{N_{\text{dup}}}{N} \tag{16}$$

K Case Study

As illustrated in the Figure 1, incorporating dllm-cache into LLaDA-V leading to generated responses with excessive repetition of words (e.g., "the") and punctuation marks (e.g., ","). In contrast, our mitigation strategy substantially reduces such repetition. Notably, it also encourages the model to attend more effectively to the information contained in surrounding tokens, enabling the generation of image descriptions that are both more detailed and more coherent.

Instruction: What preparations should be made before cooking?

LLaDA-8B-Instruct:

Before cooking, it is important to gather all the ingredients ingredients needed for the recipe and it is also important to have all the necessary tools and equipment ready. It is also helpful to idea the cooking methods and time required before cooking.

Figure 10: Repetition phenomena in dLLMs.

L Repetition Phenomena in dLLMs

As illustrated in Figure 10, we find that diffusion-based large language models likewise suffer from repetition in generated text. This finding motivates future research into the internal mechanisms of both dLLMs and dMLLMs. Furthermore, our study reveals that the repeated words are predominantly low-semantic terms (e.g., "the," "is"), while the example in the Figure 10 suggests that repetitions in dLLMs may involve a broader variety of words.

M Declaration of the Use of Generative AI (Large Language Models)

Generative AI tools, including Grammarly and ChatGPT, were used solely for grammar checking and language polishing. All technical content, experimental design, data analysis, and conclusions were generated and verified exclusively by the human authors. The use of AI tools does not affect the originality or authorship of this work.

N Ethics Statement

This work seeks to uncover the underlying causes of the Repeat Curse in dMLLMs with cache from an information-flow perspective and to propose mitigation strategies that enhance the output performance of diffusion-based multimodal large language models (dMLLMs). The proposed methods, CoTA, is developed using only publicly available dMLLMs (LLaDA-V), caching approaches (dLLM-Cache), and benchmark datasets (e.g., MME, MMBench, SEED), without relying on any private, sensitive, or human-subject data. These techniques do not introduce biases beyond those inherent to the base models, and are designed to supplement rather than replace human supervision in critical applications. Nevertheless, while our methods can effectively mitigate response repetition, they cannot guarantee the elimination of errors or misleading outputs. Hence, caution is advised in practical deployments.

O Reproducibility Statement

To ensure full reproducibility, we provide the following resources: (1) Code: The complete implementation of CoTA, including Context Tokens Attention Enhancement (CTAE) and Context Tokens Entropy-Guided Voting (CTEV), will be publicly released on GitHub upon publication. (2) Hyperparameters: For dLLM-Cache, all experiments fix the hyperparameters at $\alpha = 25\%$, $\mathcal{E}_p = 25$, and $\mathcal{E}_s = 7$. The parameter details for CoTA are described in Section 5.5, while those for the evaluation setup are provided in Section I. (3) Evaluation: We use standard, publicly available benchmarks (e.g., MME, MMBench, SEED) along with their official evaluation scripts. (4) Compute: Experiments are conducted on NVIDIA A800 80GB GPUs. Inference latency is reported using TPS and FLOPs (Table 3). (5) Models and Cache Methods: We evaluate open-source dMLLMs, specifically LLaDA-V (8B). The cache method is based on the official implementation of dLLM-Cache for LLaDA-V. No proprietary data or models are used in this work.