Differences in Text Generated by Diffusion and Autoregressive Language Models

Zeyang Zhang[1,2*] **Chengwei Liang**[2*] **Xingyan Chen**[2*] **Meiqi Gu**[2*]
Minrui Luo[2] **Jingzhao Zhang**[2,1,3†] **Tianxing He**[2,3,1†]
[1]Shanghai Qi Zhi Institute
[2]Institute for Interdisciplinary Information Sciences, Tsinghua University
[3]Xiongan AI Institute
{zhangzey24,lcw24,chenxy24,gumq24,luomr22}@mails.tsinghua.edu.cn
{jingzhaoz,hetianxing}@mail.tsinghua.edu.cn

Abstract

Diffusion language models (DLMs) are promising alternatives to autoregressive language models (ARMs), yet the intrinsic differences in their generated text remain underexplored. We first find empirically that off-the-shelf DLMs exhibit lower n-gram entropy, higher semantic coherence, and higher semantic diversity. To understand the cause, we conduct controlled experiments that decouple the effects of training objectives and decoding algorithms. Results suggest that the DLM training objective contributes to the increases in semantic coherence and semantic diversity, but has a minor influence on entropy. These differences are primarily driven by the bidirectional context; other components in the training objective, such as input masking, label masking, and the weighting function, have a much weaker influence. Further, our experiments demonstrate that the reduction in entropy stems from DLMs' decoding algorithms, particularly confidence-based remasking strategies. We provide a theoretical understanding for this entropy reduction phenomenon. Together, our work uncovers key mechanisms underlying the differences between DLMs and ARMs in text generation, and informs future design of training objectives and decoding algorithms in DLMs.[1]

1 Introduction

Diffusion language models (DLMs) have recently emerged as promising alternatives to autoregressive language models (ARMs) for natural language generation (Li et al., 2025b; Tseng et al., 2025). They offer distinct advantages, such as higher data efficiency and faster text generation (Prabhudesai et al., 2025; Khanna et al., 2025; Liu et al., 2025). With comparable model sizes and training budgets, DLMs achieve performance competitive with ARMs across various tasks, such as language understanding, mathematical reasoning, and code generation (Song et al., 2025; Khanna et al., 2025; Zhu et al., 2025).

However, beyond comparisons on efficiency and benchmark performance, the differences in the intrinsic properties of text generated by DLMs and ARMs remain underexplored. In this work, we evaluate the generated texts using three metrics: entropy, semantic coherence, and semantic diversity (defined in §3.2), which serve as distributional probes capturing how the generated corpus disperses at the token, sentence, and document levels, respectively.

As shown in Figure 1, our evaluation of 20 off-the-shelf DLMs and ARMs reveals that DLMs tend to produce texts with lower n-gram entropy, higher semantic coherence, and higher semantic diversity (n-gram entropy serves as a proxy for full entropy; see §3.3 for details).

[*]Equal contribution.
[†]Corresponding authors.
[1]Our code is available at https://github.com/imzzy1201/DifferenceDiffusionAutoregressive.

These findings provide insights into how the generation behaviors of DLMs and ARMs differ.

We conduct controlled experiments to isolate and analyze the mechanisms underlying these differences. Unlike ARMs, DLMs model the language distribution through mask prediction from noised inputs $p_\theta(x|x_{\text{noised}})$ (Austin et al., 2021; Shi et al., 2024; Sahoo et al., 2024). Moreover, DLMs employ a variety of flexible decoding strategies (Nie et al., 2025; Cheng et al., 2025; Wu et al., 2025a; Bie et al., 2025; Ye et al., 2025). Consequently, both the training objective and the decoding algorithm could contribute to the observed metric differences.

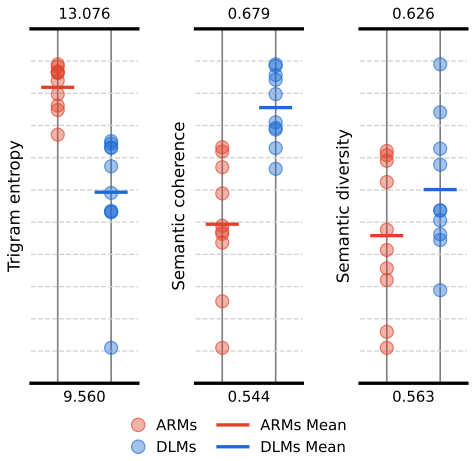


Figure 1: Comparison of three metrics between 20 off-the-shelf DLMs and ARMs. DLMs tend to exhibit lower trigram entropy alongside higher semantic coherence and semantic diversity compared to ARMs.

To decouple their effects, we perform two sets of experiments: (1) on the training objectives, comparing DLMs and ARMs under a shared decoding setup; and (2) on the decoding algorithms, comparing various decoding strategies within DLMs. To summarize, our contributions are as follows:

- We evaluate the generation behaviors of 20 off-the-shelf DLMs and ARMs on n-gram entropy, semantic coherence, and semantic diversity, providing insights into the differences between their generated corpora (§3.3).
- Through controlled experiments on training objectives, we find that the DLM training objective contributes to the increase in semantic coherence and semantic diversity, but minimally affects entropy. By interpolating the objectives between DLMs and ARMs, we identify bidirectional context as the dominant factor responsible for these increments, whereas other components, such as input masking, label masking, or weighting functions, have a much smaller effect (§4).
- Through detailed experiments and analyses on decoding algorithms, we demonstrate that the choice of remasking strategy is critical. Specifically, DLMs' entropy drops under confidence-based remasking strategies, resulting in lower entropy than ARMs; under other strategies, DLMs' entropy remains approximately unchanged and stays at ARMs' level. We also provide theoretical understanding for this effect (§5).

2 Related work

Diffusion language models. Inspired by the success of diffusion models in image generation (Rombach et al., 2022; Ho et al., 2020), early work establishes a viable path from continuous to discrete domains, introducing diffusion over discrete tokens for language modeling (Austin et al., 2021; Lou et al., 2023). Building on subsequent theoretical developments (Shi et al., 2024; Ou et al., 2024; Sahoo et al., 2024), recent efforts scale these methods to large language models, such as LLaDA (Nie et al., 2025) and Dream (Ye et al., 2025), demonstrating strong scalability and promising reasoning capabilities.

To further improve efficiency, various approaches are proposed, including autoregressive initialization and block-wise diffusion (Wu et al., 2025b; Ye et al., 2025; Gong et al., 2024; Cheng et al., 2025; Wu et al., 2025a). Concurrently, the flexible decoding order of DLMs motivates specialized decoding strategies, including random selection, confidence-based or entropy-based remasking (Chang et al., 2022; Nie et al., 2025; Kim et al., 2025; Ye et al., 2025; Ben-Hamu et al., 2025), dynamic scheduling (Wu et al., 2025b; Cheng et al., 2025; Wu et al., 2025a; Bie et al., 2025), and other related techniques.

Comparison between DLMs and ARMs. Prior works comparing DLMs and ARMs can be organized into three aspects: training objective and dynamics, decoding mechanism, and downstream task performance.

In terms of training objective and dynamics, prior works examine how different training formulations shape model behavior, such as in representation structure, generalization, and robustness. Specifically, DLMs may exhibit early-layer redundancy and reduced recency bias (Goel et al., 2026), achieve higher data efficiency (Prabhudesai et al., 2025; Gao et al., 2025; Ni et al., 2025), and mitigate the reversal curse (Kitouni et al., 2024; Shin et al., 2026).

Regarding decoding mechanism, DLMs enable flexible, any-order generation and theoretically support more efficient parallel sampling (Li & Cai, 2025; Jiang et al., 2025). However, empirical results show that these advantages are not always realized in practice (Yang et al., 2025b; Ni et al., 2026), as performance could be influenced by decoding strategies. Other works discover that DLMs often determine their answers at early decoding stages (Li et al., 2025a), and can be halted early (Vaina et al., 2023). Diffusion-specific safety issues are also explored (Rahimi et al., 2026; Zhang et al., 2025c; Wen et al., 2025).

For downstream task performance, prior works generally find that DLMs are competitive with similarly sized ARMs (Nie et al., 2024; Li et al., 2025b), and that DLMs show advantages in tasks requiring bidirectional reasoning or global context understanding (Zhang et al., 2025a; Wang et al., 2026; Xiong et al., 2025). Further, some studies also demonstrate trade-offs between generation quality and generation speed (Feng et al., 2025; Dikov et al., 2025).

Overall, existing studies primarily characterize differences in efficiency and downstream task performance, leaving the intrinsic properties of the generated text itself underexplored.

3 Preliminaries and results for off-the-shelf models

3.1 Diffusion language models

Let $\mathcal{V}$ be the vocabulary. A text sequence of length L is denoted by $x \in \mathcal{V}^L$, where x^i is the i-th token. We denote by $\mathcal{D}_{\text{data}}$ the data distribution over $\mathcal{V}^L$.

DLMs learn to reconstruct clean text from noised inputs by predicting the ground-truth tokens for randomly masked positions (Austin et al., 2021; Nie et al., 2025):

$$\mathcal{L}(\theta) = -\mathbb{E}_{x_0,t,x_t}\left[\omega_t \sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_t)\right]. \tag{1}$$

Here, a clean sequence $x_0 \sim \mathcal{D}_{\text{data}}$ is noised into x_t by converting each token to `MASK` token with probability t, where $t \sim \text{Uniform}(0,1)$ represents the noise level. A weighting function ω_t, commonly set to $1/t$, balances the loss across noise levels. The objective trains the model to recover the original token x_0^i at each masked position in x_t.

Unlike the left-to-right sequential decoding of ARMs, DLMs start generation with a sequence of `MASK` tokens, iteratively refining it into actual text. A common practice is block-wise decoding (Arriola et al., 2025), where the sequence is divided into equal-length blocks that are refined consecutively. We focus on this setting.

Within each block, the model performs multiple denoising steps to gradually decode all masked positions before moving to the next block. At each step, the model predicts a token for every masked position in the block, then remasks a subset of them, i.e., reverting them to `MASK`, before proceeding to the next step. The choice of which positions to remask is determined by a *remasking strategy*. Assume a block length of B and a target number of N denoising steps. In this work, we investigate four commonly used remasking strategies:

- **Low-confidence remasking** (Chang et al., 2022; Nie et al., 2025): This strategy remasks tokens with the lowest *confidence* values, i.e., their predicted probabilities, leaving exactly B/N most confident tokens to be decoded at each step.
- **Dynamic low-confidence remasking** (Cheng et al., 2025; Wu et al., 2025a; Bie et al., 2025): At each step, this strategy first attempts to remask tokens with confidence lower

than τ. If the number of decoded tokens does not reach the target B/N, it falls back to low-confidence remasking. Consequently, the total number of decoding steps for a block may be fewer than N, since more than B/N tokens may be decoded in a single step.

- **High-entropy remasking** (Ye et al., 2025): This strategy evaluates the entropy of the predicted token distribution at each position, and remasks positions with the highest entropy values so that exactly B/N tokens are decoded at each step.
- **Random remasking** (Nie et al., 2025): This strategy uniformly randomly selects tokens to remask so that exactly B/N tokens are decoded at each step.

We refer to the first two approaches collectively as *confidence-based remasking strategies*, as they select remasking positions based on the confidence values of the predicted tokens.

3.2 Metrics for comparing generated text

We aim to understand the different generation behaviors of DLMs and ARMs across multiple granularities, progressing from the smallest token level, to the intermediate sentence level, and finally to the overall document level. Accordingly, we adopt three metrics, entropy, semantic coherence, and semantic diversity.

The underlying intuition motivating this choice is that each metric captures how the generated corpus "disperses" at its corresponding granularity. Specifically, higher entropy, lower semantic coherence, and higher semantic diversity reflect more dispersed generation behaviors at the token, sentence, and document levels, respectively.

Given $p_{\mathcal{G}}$ as the generation distribution over the tuple (c, x), where c is the prompt and x is the response, the three metrics are defined as follows:

- **Entropy** of the generation distribution represents the average uncertainty per token:

$$\mathcal{H}(p_{\mathcal{G}}) \coloneqq -\mathbb{E}_{(c,x)\sim p_{\mathcal{G}}}\left[\frac{1}{L}\log p_{\mathcal{G}}(x \mid c)\right]. \tag{2}$$

  When $p_{\mathcal{G}}(x \mid c)$ is intractable to evaluate, following Shannon (1951), we use *n-gram entropy*, i.e., the entropy of the empirical n-gram distribution of the corpus, as a proxy for the full entropy.

- **Semantic coherence** characterizes the degree of local semantic continuity between adjacent sentences. Following Hansen et al. (2023), we evaluate this metric by calculating the average cosine similarity between embeddings of adjacent sentences. Given a normalized text embedding model $\mathrm{emb}(\cdot)$[2], and denoting each sampled response as K consecutive sentences $x = s^1 s^2 \cdots s^K$, we define semantic coherence as:

$$\mathrm{Coh}(p_{\mathcal{G}}) \coloneqq \mathbb{E}_{(c,x)\sim p_{\mathcal{G}}}\left[\frac{1}{K-1}\sum_{i=1}^{K-1}\langle \mathrm{emb}(s^i), \mathrm{emb}(s^{i+1})\rangle \;\middle|\; K \geq 2\right]. \tag{3}$$

  A higher score indicates greater semantic similarity between adjacent sentences, which we interpret as stronger local semantic continuity or topic persistence.

- **Semantic diversity** is quantified, in line with Guo et al. (2025b), as one minus the average pairwise cosine similarity of the generated sequences in the semantic embedding space. Using the same normalized embedding model, it is computed as:

$$\mathrm{Div}(p_{\mathcal{G}}) \coloneqq 1 - \mathbb{E}_{(c,x),(c',x')\sim p_{\mathcal{G}}}[\langle \mathrm{emb}(x), \mathrm{emb}(x')\rangle]. \tag{4}$$

  This is equivalent to the trace of the covariance matrix of the embeddings, i.e., $\mathrm{tr}\left(\mathrm{Cov}_{(c,x)\sim p_{\mathcal{G}}}(\mathrm{emb}(x))\right)$.[3]

  It measures whether a model possesses rich language patterns to adapt to a variety of prompts, rather than collapsing into homogeneous responses irrespective of the input.

[2] All sentence embeddings satisfy $\|\mathrm{emb}(\cdot)\|_2 = 1$.

[3] Let $z = \mathrm{emb}(x)$ with $\|z\|_2 = 1$. Then $\mathbb{E}[\langle z, z'\rangle] = \|\mathbb{E}[z]\|_2^2$ for independent z, z', and $\mathrm{tr}(\mathrm{Cov}(z)) = \mathbb{E}[\|z\|_2^2] - \|\mathbb{E}[z]\|_2^2 = 1 - \|\mathbb{E}[z]\|_2^2$, yielding the equivalence.

These metrics serve as neutral distributional probes rather than quality judgments, with no inherently preferable directionality.

3.3 Evaluation on off-the-shelf models

To establish a practical foundation for our work and motivate our subsequent analysis, we begin by evaluating the generation behaviors of off-the-shelf DLMs and ARMs across the three metrics in §3.2. We compare 10 DLMs and 10 ARMs from diverse model families, with comparable model sizes ranging from 3B to 8B (listed in Appendix A). To estimate these metrics, we sample texts from these models on 1000 different prompts from the FineWeb dataset (Penedo et al., 2024), using their default sampling configurations.

Regarding entropy, the non-sequential generation nature of DLMs prevents computing the log probability of a sequence; thus, we adopt n-gram entropy as a proxy (Shannon, 1951). To evaluate semantic coherence and semantic diversity, we extract embeddings using `bge-m3` (Chen et al., 2024). As shown in Figure 1, DLMs on average exhibit lower trigram entropy, higher semantic coherence, and higher semantic diversity than ARMs. Unigram and bigram entropy exhibit the same trends as trigram entropy. See Appendix A for more details.

To control for model architecture and training data, we additionally evaluate the 1.1B-parameter pretrained ARM and DLM checkpoints from SMDM (Nie et al., 2024). As shown in Table 3 (Appendix A.3), the results are consistent with the trends observed above.

While the results on off-the-shelf models reveal clear trends, there is no strict control over their training setups, which may compromise the robustness of the observed phenomena and hinder rigorous analysis. Therefore, in §4 and §5, we train models from scratch and conduct controlled experiments to isolate and analyze the mechanisms underlying these differences. Since DLMs and ARMs differ in both their training objectives and decoding algorithms, we decouple these effects through two sets of experiments: in §4, we compare the training objectives under a shared decoding scheme; in §5, we evaluate various decoding strategies on the same model.

4 Controlled experiments on training objectives

As initially observed in off-the-shelf models (Figure 1), DLMs and ARMs exhibit distinct generation behaviors. We first investigate whether these differences are intrinsically driven by their training objectives.

4.1 Decomposition of objective components

To analyze the intrinsic effects of training objectives, we begin by decomposing each objective into its constituent components. Recall from §3.1 that the objective of DLMs is

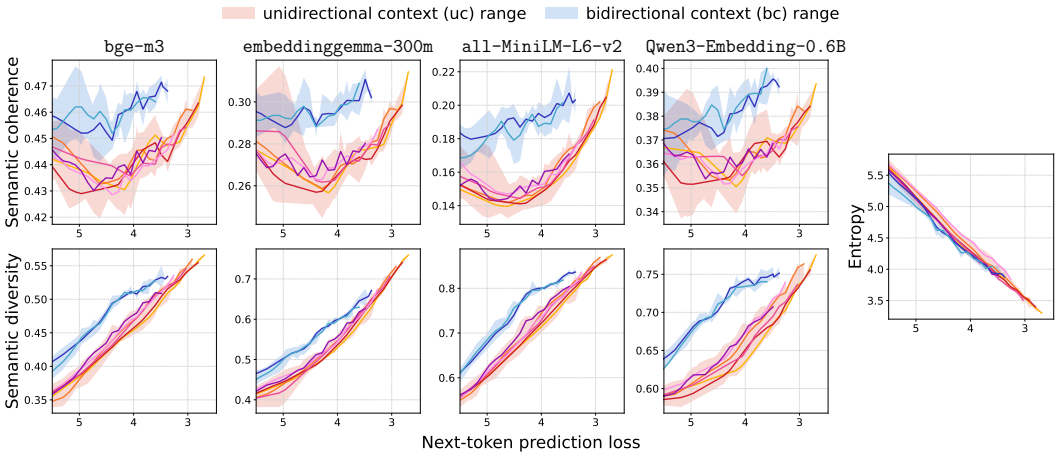
$$\mathcal{L}_{\text{DLM}}(\theta) := -\mathbb{E}_{x_0,t,x_t}\left[\omega_t \sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_t)\right], \quad \text{where } \omega_t = 1/t. \tag{5}$$

We can express ARM's training objective in a similar format:

$$\mathcal{L}_{\text{ARM}}(\theta) := -\mathbb{E}_{x_0}\left[\sum_{i=1}^{L} \log p_\theta(x_0^i \mid x_0^{<i})\right]. \tag{6}$$

The differences between these two objectives can be decomposed into four components:

- **Input masking**. DLMs take the noised sequence $x_{\text{input}} = x_t$ as input, while ARMs take the raw sequence $x_{\text{input}} = x_0$.
- **Label masking**. DLMs accumulate the loss only over masked tokens ($\mathbf{1}[x_t^i = \texttt{MASK}]$), while ARMs accumulate it over all tokens.



Legend color	Components	Training objective
	uc	$-\mathbb{E}_{x_0}\left[\sum_{i=1}^{L} \log p_\theta(x_0^i \mid x_0^{<i})\right] = \mathcal{L}_{\text{ARM}}$
	uc+im	$-\mathbb{E}_{x_0,t,x_t}\left[\sum_{i=1}^{L} \log p_\theta(x_0^i \mid x_t^{<i})\right]$
	uc+lm	$-\mathbb{E}_{x_0,t,x_t}\left[\sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_0^{<i})\right]$
	uc+lm+wf	$-\mathbb{E}_{x_0,t,x_t}\left[\frac{1}{t}\sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_0^{<i})\right]$
	uc+im+lm	$-\mathbb{E}_{x_0,t,x_t}\left[\sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_t^{<i})\right]$
	uc+im+lm+wf	$-\mathbb{E}_{x_0,t,x_t}\left[\frac{1}{t}\sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_t^{<i})\right]$
	bc+im+lm	$-\mathbb{E}_{x_0,t,x_t}\left[\sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_t)\right]$
	bc+im+lm+wf	$-\mathbb{E}_{x_0,t,x_t}\left[\frac{1}{t}\sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_t)\right] = \mathcal{L}_{\text{DLM}}$

Figure 2: Definition and evaluation of eight interpolated training objectives. Abbreviations denote the incorporated components: unidirectional context (uc), bidirectional context (bc), input masking (im), label masking (lm), and weighting function (wf). Curves are averaged over random seeds. A clear clustering emerges, demonstrating that the bidirectional context in the DLM training objective markedly increases semantic coherence and semantic diversity, while entropy stays approximately the same across all eight models.

- **Weighting function**. The diffusion process of DLMs yields a time-dependent weight ($\omega_t = 1/t$), whereas ARMs use a constant weight.

- **Context scope**. DLMs condition on bidirectional context $p_\theta(x_0^i \mid x_{\text{input}})$, whereas ARMs condition on unidirectional context $p_\theta(x_0^i \mid x_{\text{input}}^{<i})$.

To investigate how each individual component affects the resulting model, we design six intermediate training objectives, as detailed in Figure 2. See Appendix B.1 for more details on the choice of training objectives. These objectives serve as an interpolation between the ARM objective and the DLM objective. Starting from the ARM objective, we gradually change each component until we obtain the DLM objective. Consequently, by observing at which step the differences in metrics emerge, we can isolate the influence of specific components.

For instance, the uc+im objective in Figure 2 takes a randomly masked prefix $x_t^{<i}$ as input and is trained to predict the ground truth x_0^i at all positions, using the loss $-\mathbb{E}_{x_0,t,x_t}[\sum_{i=1}^{L} \log p_\theta(x_0^i \mid x_t^{<i})]$. Building on this, the uc+im+lm objective trains on x_0^i only for $x_t^i = \texttt{MASK}$, utilizing the loss $-\mathbb{E}_{x_0,t,x_t}[\sum_{i=1}^{L} \mathbf{1}[x_t^i = \texttt{MASK}] \log p_\theta(x_0^i \mid x_t^{<i})]$.

4.2 Experimental setup

We train eight models from scratch, each optimized with one of the training objectives defined in Figure 2. All models use an identical 120M-parameter LLaMA architecture (Touvron et al., 2023a). Each model is trained on the FineWeb (Penedo et al., 2024) dataset for 2B tokens using a context length of 512. Each experiment is repeated with three different random seeds.

For evaluation, to isolate the effect of decoding algorithms, we adopt a unified decoding procedure across all models. Specifically, we evaluate every model using standard sequential autoregressive decoding. For DLMs, this setup is equivalent to setting the decoding block length to one. At each checkpoint, we evaluate the models by generating 512 sequences, each 512 tokens long, conditioned on distinct 32-token prompts sampled from the evaluation split of the dataset.

Due to the sequential decoding, we can directly estimate entropy by autoregressive factorization:

$$\mathcal{H}(p_{\mathcal{G}}) = -\mathbb{E}_{(c,x)\sim p_{\mathcal{G}}}\left[\frac{1}{L}\sum_{i=1}^{L}\log p_{\mathcal{G}}(x^i|x^{<i},c)\right]. \tag{7}$$

To ensure robustness, we evaluate both semantic coherence and semantic diversity using four different embedding models: `bge-m3` (Chen et al., 2024), `embeddinggemma-300m` (Vera et al., 2025), `all-MiniLM-L6-v2` (Reimers & Gurevych, 2019), and `Qwen3-Embedding-0.6B` (Zhang et al., 2025b). See Appendix B.2 for more details on experimental setups.

4.3 Results

To ensure a fair comparison, we evaluate the models at checkpoints that achieve the same *next-token prediction loss* on the validation set (see Appendix B.2 for details). Since all models are restricted to sequential generation, this loss is exactly the cross-entropy between each model's generation distribution and the data distribution, and thus reflects generation quality. Consequently, this alignment provides evidence that models trained with different objectives have broadly comparable capabilities. We further verify, using LLM-as-a-judge, MAUVE (Pillutla et al., 2021), and generative perplexity, that generation quality is controlled across all eight models. See Appendix B.3 for details.

Under this controlled setup, we observe a clear clustering effect across the metrics (Figure 2). The eight models are divided into two distinct groups. The cluster containing the DLM objective consistently achieves **higher semantic coherence and higher semantic diversity** than the cluster containing the ARM objective. **This matches the trends in Figure 1.** Meanwhile, we observe **no significant difference in entropy** across the evaluated models.

We attribute this two-cluster separation to the difference in *context scope*: those utilizing bidirectional context form the cluster with higher semantic coherence and higher semantic diversity, while those using unidirectional context form the other. This separation suggests that bidirectional context is the dominant factor driving these differences, whereas input masking, label masking, and the weighting function exert a much weaker influence.

Specifically, we emphasize two pairs of experiments in Figure 2, where each pair differs only in context scope: (`uc+im+lm`, `bc+im+lm`) and (`uc+im+lm+wf`, `bc+im+lm+wf`). For each pair, the two experiments show a large gap in semantic coherence and semantic diversity, indicating the dominant effect of bidirectional context.

We verify that all phenomena in this section are robust across different datasets (FineWeb and TinyStories) and architectures (LLaMA and Qwen2). See Appendix B.5 for details.

5 Analysis of decoding algorithms

Apart from left-to-right decoding, DLMs can generate tokens in the order guided by an arbitrary remasking strategy, as mentioned in §3.1. At the same time, the experiments in

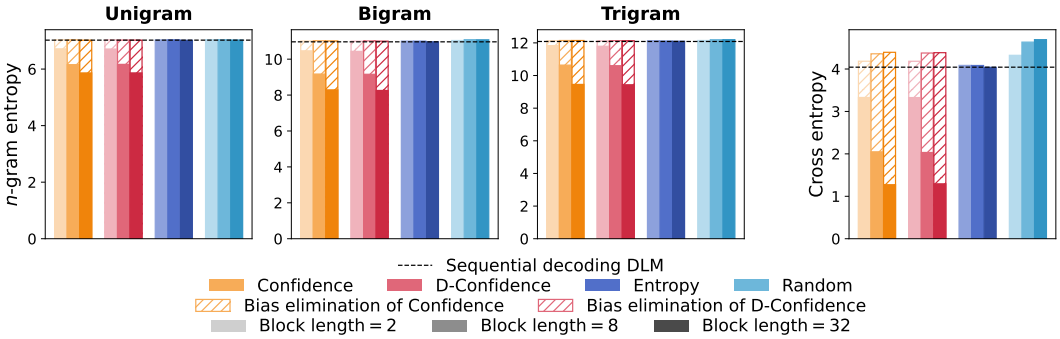


Figure 3: n-gram entropy and cross-entropy (defined in Eq.(9)) across different DLM remasking strategies and block lengths. The labels denote low-confidence remasking (Confidence), dynamic low-confidence remasking (D-Confidence), high-entropy remasking (Entropy), random remasking (Random). The horizontal dashed line in the last subplot indicates $\mathcal{H}(p_{\text{seq}})$. The combined height of the bars for the confidence-based strategies, including the hatched areas stacked on top, represents the results of the bias elimination experiments (refer to §5.2). This figure shows results for a single seed; see Appendix C.3 for other seeds.

§4 show that the training objective alone does not lead to an entropy difference between DLMs and ARMs, while Figure 1 points to potentially lower entropy in DLMs. We therefore investigate the influence of remasking strategies on the generation behaviors of DLMs.

5.1 Experimental setup

We use sequential decoding as a baseline to evaluate the influence of these remasking strategies. Our evaluation builds on the configuration from §4.2, using the final checkpoints of models trained in §4 with the $\mathcal{L}_{\text{DLM}}$ objective, modifying only the decoding algorithm. For all strategies, we set the target number of denoising steps per block N equal to the block length B, which targets decoding one token per step. For the dynamic low-confidence remasking strategy, following Cheng et al. (2025), we set the confidence threshold to $\tau = 0.9$. We evaluate three block lengths: 2, 8, and 32. Following §4, we conduct ablations across seeds, datasets and architectures. They are detailed in Appendix C.3.

5.2 Results

Entropy reduction phenomenon. Following §3.3, we use n-gram entropy to approximate full entropy. As shown in Figure 3, confidence-based strategies consistently lower n-gram entropy compared to sequential decoding, whereas the other strategies have a comparatively minor influence. This suggests that **confidence-based decoding reduces entropy**.

To verify this, we establish the following inequality chain, comparing the full entropy under a confidence-based decoding strategy $\mathcal{H}(p_{\text{con}})$ with that under sequential decoding $\mathcal{H}(p_{\text{seq}})$:

$$\mathcal{H}(p_{\text{con}}) \overset{\text{(i)}}{\leq} \mathcal{H}(p_{\text{con}}, p_{\text{seq}}) \overset{\text{(ii)}}{\leq} \mathcal{H}(p_{\text{seq}}) - \delta, \tag{8}$$

for some non-negligible $\delta > 0$, where $\mathcal{H}(p_{\text{con}})$ and $\mathcal{H}(p_{\text{seq}})$ are defined in Eq.(2), and the intermediate term denotes the cross-entropy:

$$\mathcal{H}(p_{\text{con}}, p_{\text{seq}}) \coloneqq -\mathbb{E}_{(c,x)\sim p_{\text{con}}} \left[\frac{1}{L} \log p_{\text{seq}}(x \mid c) \right]. \tag{9}$$

Inequality (i) holds by Gibbs' inequality (Brémaud, 2012). We empirically evaluate inequality (ii) by sampling x from p_{con} (or p_{seq}) for each prompt c, and computing $\log p_{\text{seq}}(x)$ using autoregressive factorization similar to Eq.(7).

As shown in Figure 3, for confidence-based strategies, the cross-entropy $\mathcal{H}(p_{\text{con}}, p_{\text{seq}})$ consistently falls far below the sequential decoding entropy $\mathcal{H}(p_{\text{seq}})$. This validates inequality (ii),

completing the inequality chain. Conversely, inequality (ii) fails when replacing p_{con} by other non-confidence-based strategies, thereby breaking the chain.[4]

Understanding of entropy reduction. One possible explanation for this entropy reduction is a *distributional bias*. In theory, even for a perfectly trained DLM, confidence-based decoding yields a biased generation distribution.

As mentioned in §3.1, confidence-based strategies first sample from the predicted distribution, then select remasking positions based on the confidence values of these sampling outcomes. By contrast, other strategies select positions based solely on the predicted distribution (i.e., by the distribution entropy or at random) before actual sampling occurs. Consequently, the outcome-dependence in confidence-based strategies introduces bias, whereas other strategies remain unbiased.

To further explore this explanation, we provide two supporting observations:

- **Theoretical illustration under factorization assumption.** To understand how this bias connects to entropy reduction, we first examine a single denoising step. Confidence-based strategies prefer positions with higher confidence values, i.e., predicted probabilities. As a result, they sharpen the predicted distribution by amplifying high probabilities and suppressing low ones, thereby reducing per-step entropy.

  Building upon this, we then formally prove that entropy reduction holds for the final generation under a simplified factorization assumption (Theorem 1).

  Theorem 1 (Informal)**.** *Assume that the data distribution at each position is independent of the others, so that the full distribution factorizes as $p_{\text{data}}(x) = \prod_{i=1}^{L} p^i_{\text{data}}(x^i)$, and that the DLM p_θ is optimally trained, i.e., for any masked position i of a noised input x_t, $p_\theta(x_0^i \mid x_t) = p^i_{\text{data}}(x_0^i)$. Let p_{lcr} and p_{dlcr} denote the generation distributions produced by applying low-confidence remasking and dynamic low-confidence remasking to the model, respectively. Then,*

  $$\mathcal{H}(p_{\text{lcr}}) \leq \mathcal{H}(p_{\text{data}}), \qquad \mathcal{H}(p_{\text{dlcr}}) \leq \mathcal{H}(p_{\text{data}}). \tag{10}$$

  See Appendix C.1 for the full statement and the proof.

- **Bias elimination experiments.** Theorem 1 serves as a conceptual illustration under simplified assumptions. To provide the primary empirical validation, we apply a simple fix to the confidence-based strategy to eliminate this bias. After the strategy determines the positions to decode, we *resample* the tokens at these positions. This modification breaks the correlation between the remasking position selection and the sampled token outcomes, thereby eliminating the bias. As shown in Figure 3, this fix effectively restores the n-gram entropy to a level similar to that of sequential decoding and invalidates inequality (ii). Details can be found in Appendix C.2.

We do not treat this confidence-based bias as a flaw, as it is mathematically undesirable but empirically useful. We further discuss its role and desirability in Appendix C.4.

Other influences. We further evaluate how remasking strategies affect semantic coherence and semantic diversity. Our results show that confidence-based strategies are relatively unstable: while they consistently increase semantic coherence across our multiple experiments, their influence on semantic diversity is highly dataset-dependent. Specifically, in our main experiments on the FineWeb dataset, confidence-based strategies increase semantic diversity, but ablation experiments on the TinyStories dataset show decreases in semantic diversity, suggesting a relatively unpredictable behavior.

In contrast, other strategies have comparatively negligible effects on both metrics. For confidence-based strategies, applying bias elimination also removes their effects. These results suggest that the bias-based distinction may also determine whether a strategy affects semantic coherence and semantic diversity. See Appendix C.5 for more details.

[4]Note that the failure of inequality (ii) does not imply an increase in entropy, since Eq.(8) does not hold in the reverse direction.

6 Conclusion and discussion

In this work, we investigate the different generation behaviors of DLMs and ARMs by decoupling the effects of their training objectives and decoding procedures. Our control experiments identify two key mechanisms: First, the DLM training objective, particularly through the bidirectional context, drives the model toward higher semantic coherence and higher semantic diversity, while maintaining an entropy similar to that of ARMs. Second, the DLM decoding algorithms, specifically confidence-based remasking strategies, reduce entropy relative to autoregressive decoding.

It is also worth noting that our observations imply that DLMs can achieve higher semantic diversity at the same or lower entropy levels than ARMs. Under the evaluation settings considered in our work, this suggests that DLMs may relax the conventional text generation trade-off between diversity and certainty, pointing to their potential for diverse yet controllable text generation. The extent to which this observation generalizes may nevertheless depend on the choice of metrics, decoding strategies, tasks, and evaluation protocols (Wiher et al., 2022; Arias et al., 2025; Ding et al., 2025).

Together, our findings offer insights into the selection between DLMs and ARMs for different scenarios and may inform the design of training objectives and decoding algorithms in DLMs.

7 Limitations and future work

We conclude by outlining several limitations and directions for future work. Our experiments are limited to relatively small models; it remains to be verified that our findings generalize to larger models and datasets. Our theoretical analysis of the decoding procedure relies on several simplifying assumptions. Future work may relax these assumptions and extend the study to broader DLM architectures and decoding algorithms. In addition, human evaluation and broader corpus-level distributional analyses remain valuable complementary directions for validating and contextualizing our distributional findings.

Acknowledgements

We thank Haoming Wen from Tsinghua University for his valuable feedback and for his constructive suggestions on writing.

References

Abdelrahman Abouelenin, Atabak Ashfaq, Adam Atkinson, Hany Awadalla, Nguyen Bach, Jianmin Bao, Alon Benhaim, Martin Cai, Vishrav Chaudhary, Congcong Chen, et al. Phi-4-mini technical report: Compact yet powerful multimodal language models via mixture-of-loras. *arXiv preprint arXiv:2503.01743*, 2025.

Esteban Garces Arias, Meimingwei Li, Christian Heumann, and Matthias Aßenmacher. Decoding decoded: Understanding hyperparameter effects in open-ended text generation. In *Proceedings of the 31st International Conference on Computational Linguistics*, pp. 9992–10020, 2025.

Marianne Arriola, Aaron Gokaslan, Justin T Chiu, Zhihan Yang, Zhixuan Qi, Jiaqi Han, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Block diffusion: Interpolating between autoregressive and diffusion language models. *arXiv preprint arXiv:2503.09573*, 2025.

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

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

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

Pierre Brémaud. *An introduction to probabilistic modeling*. Springer Science & Business Media, 2012.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T Freeman. Maskgit: Masked generative image transformer. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 11315–11325, 2022.

Jianlv Chen, Shitao Xiao, Peitian Zhang, Kun Luo, Defu Lian, and Zheng Liu. Bge m3-embedding: Multi-lingual, multi-functionality, multi-granularity text embeddings through self-knowledge distillation. *arXiv preprint arXiv:2402.03216*, 4(5), 2024.

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

Alexander Dikov, Vladimir Zvorygin, and Valentin Malykh. Diffusion vs autoregression: An empirical study on code comment translation. In *2025 5th International Conference on Code Quality (ICCQ)*, pp. 55–63. IEEE, 2025.

Yuanhao Ding, Esteban Garces Arias, Meimingwei Li, Julian Rodemann, Matthias Aßenmacher, Danlu Chen, Gaojuan Fan, Christian Heumann, and Chongsheng Zhang. Guard: Glocal uncertainty-aware robust decoding for effective and efficient open-ended text generation. *Findings of the Association for Computational Linguistics: EMNLP*, 2025.

Ronen Eldan and Yuanzhi Li. Tinystories: How small can language models be and still speak coherent english? *arXiv preprint arXiv:2305.07759*, 2023.

Guhao Feng, Yihan Geng, Jian Guan, Wei Wu, Liwei Wang, and Di He. Theoretical benefit and limitation of diffusion language model. *arXiv preprint arXiv:2502.09622*, 2025.

Zitian Gao, Haoming Luo, Lynx Chen, Jason Klein Liu, Ran Tao, Joey Zhou, and Bryan Dai. What makes diffusion language models super data learners? *arXiv preprint arXiv:2510.04071*, 2025.

Raghavv Goel, Risheek Garrepalli, Sudhanshu Agrawal, Chris Lott, Mingu Lee, and Fatih Porikli. Skip to the good part: Representation structure & inference-time layer skipping in diffusion vs. autoregressive llms. *arXiv preprint arXiv:2603.07475*, 2026.

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

Aaron Grattafiori, Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle, Aiesha Letman, Akhil Mathur, Alan Schelten, Alex Vaughan, et al. The llama 3 herd of models. *arXiv preprint arXiv:2407.21783*, 2024.

Daya Guo, Dejian Yang, Haowei Zhang, Junxiao Song, Peiyi Wang, Qihao Zhu, Runxin Xu, Ruoyu Zhang, Shirong Ma, Xiao Bi, et al. Deepseek-r1: Incentivizing reasoning capability in llms via reinforcement learning. *arXiv preprint arXiv:2501.12948*, 2025a.

Yanzhu Guo, Guokan Shang, and Chloé Clavel. Benchmarking linguistic diversity of large language models. *Transactions of the Association for Computational Linguistics*, 13:1507–1526, 2025b.

Lasse Hansen, Ludvig Renbo Olsen, and Kenneth Enevoldsen. Textdescriptives: A python package for calculating a large variety of metrics from text. *arXiv preprint arXiv:2301.02057*, 2023.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Matthew Honnibal, Ines Montani, Sofie Van Landeghem, Adriane Boyd, et al. spacy: Industrial-strength natural language processing in python. 2020.

Albert Qiaochu Jiang, Alexandre Sablayrolles, Arthur Mensch, Chris Bamford, Devendra Singh Chaplot, Diego de Las Casas, Florian Bressand, Gianna Lengyel, Guillaume Lample, Lucile Saulnier, Lélio Renard Lavaud, Marie-Anne Lachaux, Pierre Stock, Teven Le Scao, Thibaut Lavril, Thomas Wang, Timothée Lacroix, and William El Sayed. Mistral 7b. *ArXiv*, abs/2310.06825, 2023. URL https://api.semanticscholar.org/CorpusID:263830494.

Haozhe Jiang, Nika Haghtalab, and Lijie Chen. Diffusion language models are provably optimal parallel samplers. *arXiv preprint arXiv:2512.25014*, 2025.

Samar Khanna, Siddhant Kharbanda, Shufan Li, Harshit Varma, Eric Wang, Sawyer Birnbaum, Ziyang Luo, Yanis Miraoui, Akash Palrecha, Stefano Ermon, et al. Mercury: Ultra-fast language models based on diffusion. *arXiv e-prints*, pp. arXiv–2506, 2025.

Jaeyeon Kim, Kulin Shah, Vasilis Kontonis, Sham Kakade, and Sitan Chen. Train for the worst, plan for the best: Understanding token ordering in masked diffusions. *arXiv preprint arXiv:2502.06768*, 2025.

Robert Kirk, Ishita Mediratta, Christoforos Nalmpantis, Jelena Luketina, Eric Hambro, Edward Grefenstette, and Roberta Raileanu. Understanding the effects of rlhf on llm generalisation and diversity. *arXiv preprint arXiv:2310.06452*, 2023.

Ouail Kitouni, Niklas Nolte, Diane Bouchacourt, Adina Williams, Mike Rabbat, and Mark Ibrahim. The factorization curse: Which tokens you predict underlie the reversal curse and more. *Advances in Neural Information Processing Systems*, 37:112329–112355, 2024.

Gen Li and Changxiao Cai. Breaking ar's sampling bottleneck: Provable acceleration via diffusion language models. *arXiv preprint arXiv:2505.21400*, 2025.

Pengxiang Li, Yefan Zhou, Dilxat Muhtar, Lu Yin, Shilin Yan, Li Shen, Soroush Vosoughi, and Shiwei Liu. Diffusion language models know the answer before decoding. *arXiv preprint arXiv:2508.19982*, 2025a.

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

Yangzhou Liu, Yue Cao, Hao Li, Gen Luo, Zhe Chen, Weiyun Wang, Xiaobo Liang, Biqing Qi, Lijun Wu, Changyao Tian, Yanting Zhang, Yuqiang Li, Tong Lu, Yu Qiao, Jifeng Dai, and Wenhai Wang. Sequential diffusion language models. *arXiv preprint arXiv:2509.24007*, 2025.

Ilya Loshchilov and Frank Hutter. Decoupled weight decay regularization. *arXiv preprint arXiv:1711.05101*, 2017.

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

Albert W Marshall, Ingram Olkin, and Barry C Arnold. Inequalities: theory of majorization and its applications. 1979.

Jinjie Ni, Qian Liu, Longxu Dou, Chao Du, Zili Wang, Hang Yan, Tianyu Pang, and Michael Qizhe Shieh. Diffusion language models are super data learners. *arXiv preprint arXiv:2511.03276*, 2025.

Zanlin Ni, Shenzhi Wang, Yang Yue, Tianyu Yu, Weilin Zhao, Yeguo Hua, Tianyi Chen, Jun Song, Cheng Yu, Bo Zheng, et al. The flexibility trap: Why arbitrary order limits reasoning potential in diffusion language models. *arXiv preprint arXiv:2601.15165*, 2026.

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

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. *arXiv preprint arXiv:2502.09992*, 2025.

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

Guilherme Penedo, Hynek Kydlíček, Anton Lozhkov, Margaret Mitchell, Colin Raffel, Leandro Von Werra, Thomas Wolf, et al. The fineweb datasets: Decanting the web for the finest text data at scale. *Advances in Neural Information Processing Systems*, 37:30811–30849, 2024.

Krishna Pillutla, Swabha Swayamdipta, Rowan Zellers, John Thickstun, Sean Welleck, Yejin Choi, and Zaid Harchaoui. Mauve: Measuring the gap between neural text and human text using divergence frontiers. *Advances in Neural Information Processing Systems*, 34: 4816–4828, 2021.

Mihir Prabhudesai, Mengning Wu, Amir Zadeh, Katerina Fragkiadaki, and Deepak Pathak. Diffusion beats autoregressive in data-constrained settings. *arXiv preprint arXiv:2507.15857*, 2025.

Eliron Rahimi, Elad Hirshel, Rom Himelstein, Amit LeVi, Avi Mendelson, and Chaim Baskin. Step-wise refusal dynamics in autoregressive and diffusion language models. *arXiv preprint arXiv:2602.02600*, 2026.

Nils Reimers and Iryna Gurevych. Sentence-bert: Sentence embeddings using siamese bert-networks. In *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*. Association for Computational Linguistics, 11 2019. URL https://arxiv.org/abs/1908.10084.

Robin Rombach, Andreas Blattmann, Dominik Lorenz, Patrick Esser, and Björn Ommer. High-resolution image synthesis with latent diffusion models. In *Proceedings of the IEEE/CVF conference on computer vision and pattern recognition*, pp. 10684–10695, 2022.

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

Claude E Shannon. Prediction and entropy of printed english. *Bell system technical journal*, 30(1):50–64, 1951.

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

Sangwoo Shin, BumJun Kim, Kyelim Lee, Moongyu Jeon, and Albert No. Understanding the reversal curse mitigation in masked diffusion models through attention and training dynamics. *arXiv preprint arXiv:2602.02133*, 2026.

Yuxuan Song, Zheng Zhang, Cheng Luo, Pengyang Gao, Fan Xia, Hao Luo, Zheng Li, Yuehang Yang, Hongli Yu, Xingwei Qu, et al. Seed diffusion: A large-scale diffusion language model with high-speed inference. *arXiv preprint arXiv:2508.02193*, 2025.

Gemma Team, Aishwarya Kamath, Johan Ferret, Shreya Pathak, Nino Vieillard, Ramona Merhej, Sarah Perrin, Tatiana Matejovicova, Alexandre Ramé, Morgane Rivière, Louis Rouillard, Thomas Mesnard, Geoffrey Cideron, Jean bastien Grill, Sabela Ramos, Edouard

Yvinec, Michelle Casbon, Etienne Pot, Ivo Penchev, Gaël Liu, Francesco Visin, Kathleen Kenealy, Lucas Beyer, Xiaohai Zhai, Anton Tsitsulin, Robert Busa-Fekete, Alex Feng, Noveen Sachdeva, Benjamin Coleman, Yi Gao, Basil Mustafa, Iain Barr, Emilio Parisotto, David Tian, Matan Eyal, Colin Cherry, Jan-Thorsten Peter, Danila Sinopalnikov, Surya Bhupatiraju, Rishabh Agarwal, Mehran Kazemi, Dan Malkin, Ravin Kumar, David Vilar, Idan Brusilovsky, Jiaming Luo, Andreas Steiner, Abe Friesen, Abhanshu Sharma, Abheesht Sharma, Adi Mayrav Gilady, Adrian Goedeckemeyer, Alaa Saade, Alex Feng, Alexander Kolesnikov, Alexei Bendebury, Alvin Abdagic, Amit Vadi, András György, André Susano Pinto, Anil Das, Ankur Bapna, Antoine Miech, Antoine Yang, Antonia Paterson, Ashish Shenoy, Ayan Chakrabarti, Bilal Piot, Bo Wu, Bobak Shahriari, Bryce Petrini, Charlie Chen, Charline Le Lan, Christopher A. Choquette-Choo, CJ Carey, Cormac Brick, Daniel Deutsch, Danielle Eisenbud, Dee Cattle, Derek Cheng, Dimitris Paparas, Divyashree Shivakumar Sreepathihalli, Doug Reid, Dustin Tran, Dustin Zelle, Eric Noland, Erwin Huizenga, Eugene Kharitonov, Frederick Liu, Gagik Amirkhanyan, Glenn Cameron, Hadi Hashemi, Hanna Klimczak-Plucińska, Harman Singh, Harsh Mehta, Harshal Tushar Lehri, Hussein Hazimeh, Ian Ballantyne, Idan Szpektor, Ivan Nardini, Jean Pouget-Abadie, Jetha Chan, Joe Stanton, John Wieting, Jonathan Lai, Jordi Orbay, Joseph Fernandez, Josh Newlan, Ju yeong Ji, Jyotinder Singh, Kat Black, Kathy Yu, Kevin Hui, Kiran Vodrahalli, Klaus Greff, Linhai Qiu, Marcella Valentine, Marina Coelho, Marvin Ritter, Matt Hoffman, Matthew Watson, Mayank Chaturvedi, Michael Moynihan, Min Ma, Nabila Babar, Natasha Noy, Nathan Byrd, Nick Roy, Nikola Momchev, Nilay Chauhan, Noveen Sachdeva, Oskar Bunyan, Pankil Botarda, Paul Caron, Paul Kishan Rubenstein, Phil Culliton, Philipp Schmid, Pier Giuseppe Sessa, Pingmei Xu, Piotr Stanczyk, Pouya Tafti, Rakesh Shivanna, Renjie Wu, Renke Pan, Reza Rokni, Rob Willoughby, Rohith Vallu, Ryan Mullins, Sammy Jerome, Sara Smoot, Sertan Girgin, Shariq Iqbal, Shashir Reddy, Shruti Sheth, Siim Põder, Sijal Bhatnagar, Sindhu Raghuram Panyam, Sivan Eiger, Susan Zhang, Tianqi Liu, Trevor Yacovone, Tyler Liechty, Uday Kalra, Utku Evci, Vedant Misra, Vincent Roseberry, Vlad Feinberg, Vlad Kolesnikov, Woohyun Han, Woosuk Kwon, Xi Chen, Yinlam Chow, Yuvein Zhu, Zichuan Wei, Zoltan Egyed, Victor Cotruta, Minh Giang, Phoebe Kirk, Anand Rao, Kat Black, Nabila Babar, Jessica Lo, Erica Moreira, Luiz Gustavo Martins, Omar Sanseviero, Lucas Gonzalez, Zach Gleicher, Tris Warkentin, Vahab Mirrokni, Evan Senter, Eli Collins, Joelle Barral, Zoubin Ghahramani, Raia Hadsell, Yossi Matias, D. Sculley, Slav Petrov, Noah Fiedel, Noam Shazeer, Oriol Vinyals, Jeff Dean, Demis Hassabis, Koray Kavukcuoglu, Clement Farabet, Elena Buchatskaya, Jean-Baptiste Alayrac, Rohan Anil, Dmitry, Lepikhin, Sebastian Borgeaud, Olivier Bachem, Armand Joulin, Alek Andreev, Cassidy Hardin, Robert Dadashi, and Léonard Hussenot. Gemma 3 technical report, 2025. URL https://arxiv.org/abs/2503.19786.

Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timothée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023a.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, et al. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv:2307.09288*, 2023b.

Chiung-Yi Tseng, Danyang Zhang, Junhao Song, and Ziqian Bi. Diffusion-based large language models survey. *Authorea Preprints*, 2025.

Sofia Maria Lo Cicero Vaina, Nikita Balagansky, and Daniil Gavrilov. Diffusion language models generation can be halted early. *arXiv preprint arXiv:2305.10818*, 2023.

Henrique Schechter Vera, Sahil Dua, Biao Zhang, Daniel Salz, Ryan Mullins, Sindhu Raghuram Panyam, Sara Smoot, Iftekhar Naim, Joe Zou, Feiyang Chen, et al. Embeddinggemma: Powerful and lightweight text representations. *arXiv preprint arXiv:2509.20354*, 2025.

Zihang Wang, Siyue Zhang, Yilun Zhao, Jingyi Yang, Tingyu Song, Anh Tuan Luu, and Chen Zhao. Analyzing diffusion and autoregressive vision language models in multimodal embedding space. *arXiv preprint arXiv:2602.06056*, 2026.

Zichen Wen, Jiashu Qu, Zhaorun Chen, Xiaoya Lu, Dongrui Liu, Zhiyuan Liu, Ruixi Wu, Yicun Yang, Xiangqi Jin, Haoyun Xu, et al. The devil behind the mask: An emergent safety vulnerability of diffusion llms. *arXiv preprint arXiv:2507.11097*, 2025.

Gian Wiher, Clara Meister, and Ryan Cotterell. On decoding strategies for neural text generators. *Transactions of the Association for Computational Linguistics*, 10:997–1012, 2022.

Chengyue Wu, Hao Zhang, Shuchen Xue, Shizhe Diao, Yonggan Fu, Zhijian Liu, Pavlo Molchanov, Ping Luo, Song Han, and Enze Xie. Fast-dllm v2: Efficient block-diffusion llm. *arXiv preprint arXiv:2509.26328*, 2025a.

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 preprint arXiv:2505.22618*, 2025b.

Zhihui Xie, Jiacheng Ye, Lin Zheng, Jiahui Gao, Jingwei Dong, Zirui Wu, Xueliang Zhao, Shansan Gong, Xin Jiang, Zhenguo Li, et al. Dream-coder 7b: An open diffusion language model for code. *arXiv preprint arXiv:2509.01142*, 2025.

Zhen Xiong, Yujun Cai, Zhecheng Li, and Yiwei Wang. Unveiling the potential of diffusion large language model in controllable generation. *arXiv preprint arXiv:2507.04504*, 2025.

An Yang, Baosong Yang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Zhou, Chengpeng Li, Chengyuan Li, Dayiheng Liu, Fei Huang, Guanting Dong, Haoran Wei, Huan Lin, Jialong Tang, Jialin Wang, Jian Yang, Jianhong Tu, Jianwei Zhang, Jianxin Ma, Jin Xu, Jingren Zhou, Jinze Bai, Jinzheng He, Junyang Lin, Kai Dang, Keming Lu, Keqin Chen, Kexin Yang, Mei Li, Mingfeng Xue, Na Ni, Pei Zhang, Peng Wang, Ru Peng, Rui Men, Ruize Gao, Runji Lin, Shijie Wang, Shuai Bai, Sinan Tan, Tianhang Zhu, Tianhao Li, Tianyu Liu, Wenbin Ge, Xiaodong Deng, Xiaohuan Zhou, Xingzhang Ren, Xinyu Zhang, Xipin Wei, Xuancheng Ren, Yang Fan, Yang Yao, Yichang Zhang, Yu Wan, Yunfei Chu, Yuqiong Liu, Zeyu Cui, Zhenru Zhang, and Zhihao Fan. Qwen2 technical report. *arXiv preprint arXiv:2407.10671*, 2024a.

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. *arXiv preprint arXiv:2412.15115*, 2024b.

An Yang, Anfeng Li, Baosong Yang, Beichen Zhang, Binyuan Hui, Bo Zheng, Bowen Yu, Chang Gao, Chengen Huang, Chenxu Lv, et al. Qwen3 technical report. *arXiv preprint arXiv:2505.09388*, 2025a.

Chenxiao Yang, Cai Zhou, David Wipf, and Zhiyuan Li. On powerful ways to generate: Autoregression, diffusion, and beyond. *arXiv preprint arXiv:2510.06190*, 2025b.

Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Dream 7b: Diffusion large language models. *arXiv preprint arXiv:2508.15487*, 2025.

Siyue Zhang, Yilun Zhao, Liyuan Geng, Arman Cohan, Luu Anh Tuan, and Chen Zhao. Diffusion vs. autoregressive language models: A text embedding perspective. In *Proceedings of the 2025 Conference on Empirical Methods in Natural Language Processing*, pp. 4273–4303, 2025a.

Yanzhao Zhang, Mingxin Li, Dingkun Long, Xin Zhang, Huan Lin, Baosong Yang, Pengjun Xie, An Yang, Dayiheng Liu, Junyang Lin, et al. Qwen3 embedding: Advancing text embedding and reranking through foundation models. *arXiv preprint arXiv:2506.05176*, 2025b.

Yuanhe Zhang, Fangzhou Xie, Zhenhong Zhou, Zherui Li, Hao Chen, Kun Wang, and Yufei Guo. Jailbreaking large language diffusion models: Revealing hidden safety flaws in diffusion-based text generation. *arXiv preprint arXiv:2507.19227*, 2025c.

Fengqi Zhu, Zebin You, Yipeng Xing, Zenan Huang, Lin Liu, Yihong Zhuang, Guoshan Lu, Kangyu Wang, Xudong Wang, Lanning Wei, Hongrui Guo, Jiaqi Hu, Wentao Ye, Tieyuan Chen, Chenchen Li, Chengfu Tang, Haibo Feng, Jun Hu, Jun Zhou, Xiaolu Zhang, Zhenzhong Lan, Junbo Zhao, Da Zheng, Chongxuan Li, Jianguo Li, and Ji-Rong Wen. Llada-moe: A sparse moe diffusion language model. *arXiv preprint arXiv:2509.24389*, 2025.

A Evaluation details of off-the-shelf models

We collect a dataset of texts generated by off-the-shelf DLMs and ARMs for evaluation.

A.1 Generation configuration

We randomly sample 1000 examples from the FineWeb dataset (Penedo et al., 2024). For each example, we use its first 30 tokens as a prompt and generate 20 continuations using 10 ARMs and 10 DLMs, with a maximum length of 512 tokens for each continuation. Table 1 lists the employed models.

We use the prompt template "`Continue the following text in approximately 500 words: [PROMPT]`", along with model-specific chat templates. For generation hyperparameters, we adopt the default settings for each model, and we disable thinking mode for all models. These models all have different architectures and training procedures, so adopting the default generation setting is a fair basis for comparison since it demonstrates how each model is "intended" to be used.

A.2 Data filtering and preprocessing

Since degenerate data severely interfere with our analysis, we filter out degenerate samples with extremely short length or excessively repeated content. Specifically, a sample is removed if it is shorter than 500 characters or contains a 30-character sequence repeated consecutively at least 3 times. The number of samples remaining after filtering is reported in Table 1.

Controlled experiments in §4 do not require filtering, since all models generate sequences of a fixed length, eliminating the short-length issue, and our repetition detection confirms the absence of highly repetitive text (see Appendix B.4).

Type	Model	Count
ARM	DeepSeek-R1-0528-Qwen3-8B (Guo et al., 2025a)	946
	Gemma-3-4B-it (Team et al., 2025)	996
	Llama-2-7B-chat (Touvron et al., 2023b)	997
	Llama-3.1-8B-Instruct (Grattafiori et al., 2024)	999
	Llama-3.2-3B-Instruct (Grattafiori et al., 2024)	996
	Mistral-7B-Instruct-v0.3 (Jiang et al., 2023)	995
	Phi-4-mini-Instruct (Abouelenin et al., 2025)	996
	Qwen2-7B-Instruct (Yang et al., 2024a)	993
	Qwen2.5-7B-Instruct (Yang et al., 2024b)	991
	Qwen3-8B (Yang et al., 2025a)	993
DLM	Dream-Coder-v0-Instruct-7B (Xie et al., 2025)	694
	Dream-v0-Instruct-7B (Ye et al., 2025)	215
	Fast-dLLM-v2-7B (Wu et al., 2025a)	982
	LLaDA-8B-Instruct (Nie et al., 2025)	902
	LLaDA-MoE-7B-A1B-Instruct (Zhu et al., 2025)	731
	SDAR-4B-Chat (Cheng et al., 2025)	854
	SDAR-4B-Chat-b8 (Cheng et al., 2025)	861
	SDAR-8B-Chat (Cheng et al., 2025)	871
	SDLM-3B-D4 (Liu et al., 2025)	652
	SDLM-3B-D8 (Liu et al., 2025)	689

Table 1: List of evaluated models and the number of remaining samples for each model.

A.3 Evaluation

The per-model evaluation results are shown in Table 2, and a scatter plot visualization of the same results is provided in Figure 4. For n-gram entropy, we use the Qwen3 tokenizer (Yang et al., 2025a) for tokenization. For semantic coherence and semantic diversity, we use `bge-m3` (Chen et al., 2024) as the embedding model. Off-the-shelf DLMs on average exhibit lower n-gram entropy alongside higher semantic coherence and higher semantic diversity compared to ARMs. This lower n-gram entropy does not imply mode collapse, as higher semantic diversity is observed alongside it.

To control for model architecture and training data, we additionally evaluate the 1.1B-parameter pretrained ARM and DLM checkpoints from SMDM (Nie et al., 2024). As shown in Table 3, the results are consistent with the trends observed above.

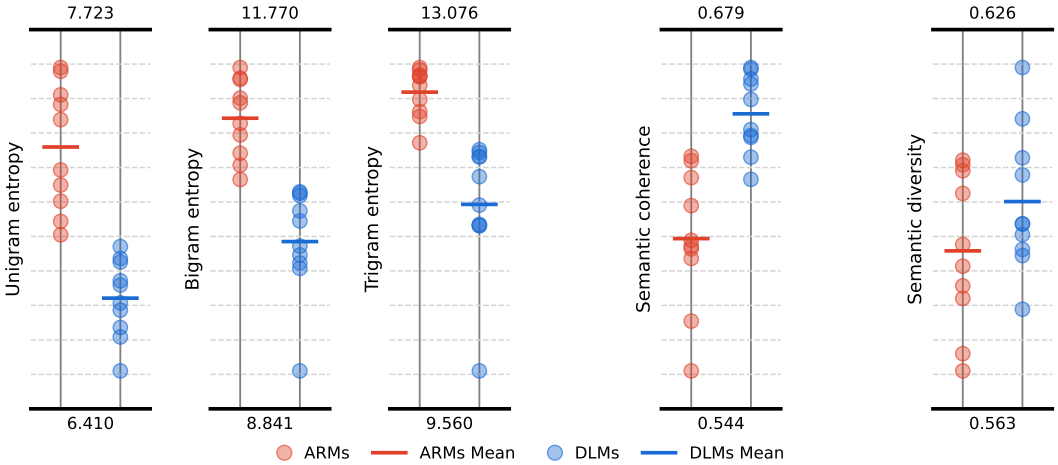


Figure 4: Scatter plot visualization of the off-the-shelf evaluation results in Table 2. Each point denotes one model, and the horizontal bars indicate the mean over models within each type.

A.4 Robustness checks

We conduct three robustness checks for the off-the-shelf evaluation:

- **Data filtering.** To ablate data filtering, we re-evaluate the metrics on the unfiltered data.
- **Response length.** The off-the-shelf generations have variable lengths, which may confound the metrics. We retain 100 generations per model with at least 200 tokens, truncate each to 200 tokens, and recompute the metrics.
- **Sentence segmentation.** The semantic coherence metric relies on segmenting each response into sentences, which we perform by default using a punctuation-based heuristic. As an alternative, we replace the original heuristic with spaCy (Honnibal et al., 2020).

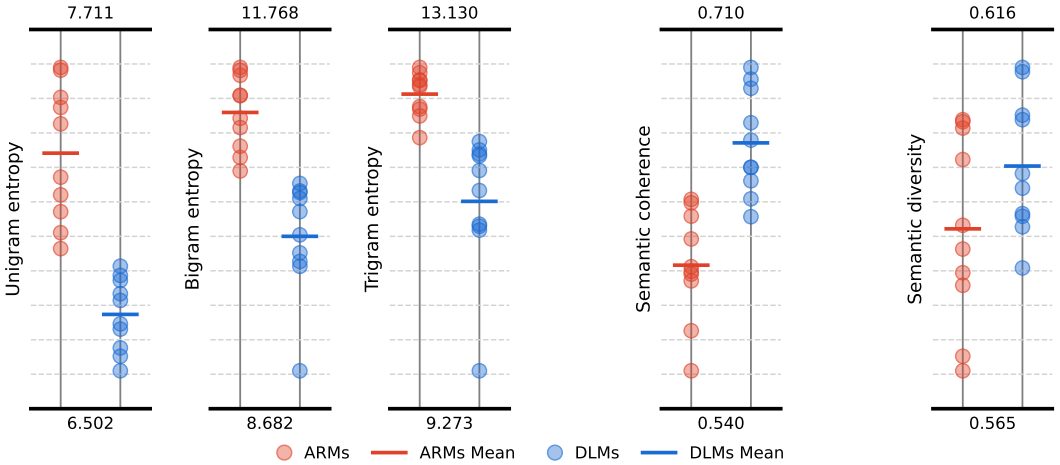
As shown in Figure 5, the results of all three ablations are consistent with the original findings.

Model	n-gram entropy			Semantic coherence	Semantic diversity
	Unigram	Bigram	Trigram		
DeepSeek-R1-0528-Qwen3-8B	7.579	11.385	12.655	0.575	0.569
Gemma-3-4B-it	7.411	11.241	12.639	0.557	0.572
Llama-2-7B-chat	7.013	10.612	12.027	0.634	0.604
Llama-3.1-8B-Instruct	7.129	10.816	12.316	0.626	0.604
Llama-3.2-3B-Instruct	7.059	10.723	12.271	0.632	0.599
Mistral-7B-Instruct-v0.3	7.237	10.957	12.428	0.616	0.603
Phi-4-mini-Instruct	7.592	11.477	12.725	0.597	0.587
Qwen2-7B-Instruct	7.497	11.395	12.702	0.602	0.581
Qwen2.5-7B-Instruct	7.464	11.205	12.641	0.604	0.590
Qwen3-8B	7.185	11.046	12.560	0.601	0.583
ARM Avg.	**7.316**	**11.086**	**12.497**	**0.604**	**0.589**
Dream-Coder-v0-Instruct-7B	6.777	10.030	11.262	0.625	0.592
Dream-v0-Instruct-7B	6.541	9.134	9.912	0.654	0.620
Fast-dLLM-v2-7B	6.838	10.370	11.896	0.633	0.590
LLaDA-8B-Instruct	6.853	10.292	11.713	0.661	0.594
LLaDA-MoE-7B-A1B-Instruct	6.752	10.098	11.449	0.665	0.580
SDAR-4B-Chat	6.972	10.506	11.898	0.641	0.611
SDAR-4B-Chat-b8	6.931	10.488	11.937	0.643	0.602
SDAR-8B-Chat	6.918	10.518	11.963	0.640	0.605
SDLM-3B-D4	6.692	9.967	11.270	0.659	0.594
SDLM-3B-D8	6.659	9.926	11.257	0.665	0.589
DLM Avg.	**6.793**	**10.133**	**11.456**	**0.649**	**0.597**

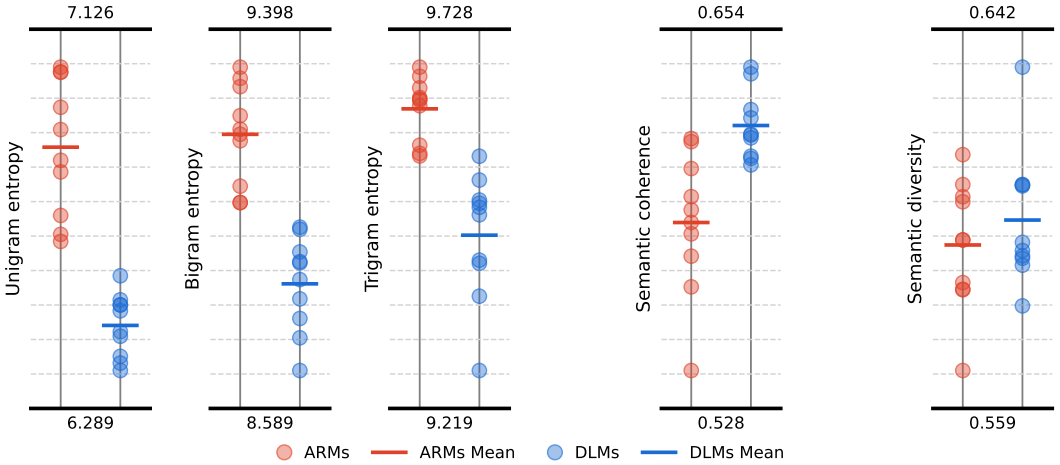
Table 2: Evaluation results for 20 off-the-shelf ARMs (top) and DLMs (bottom). "Avg." denotes the average over models within each type.

Type	n-gram entropy			Semantic coherence	Semantic diversity
	Unigram	Bigram	Trigram		
ARM	7.381	9.902	10.338	0.468	0.629
DLM	6.344	8.892	9.684	0.602	0.635

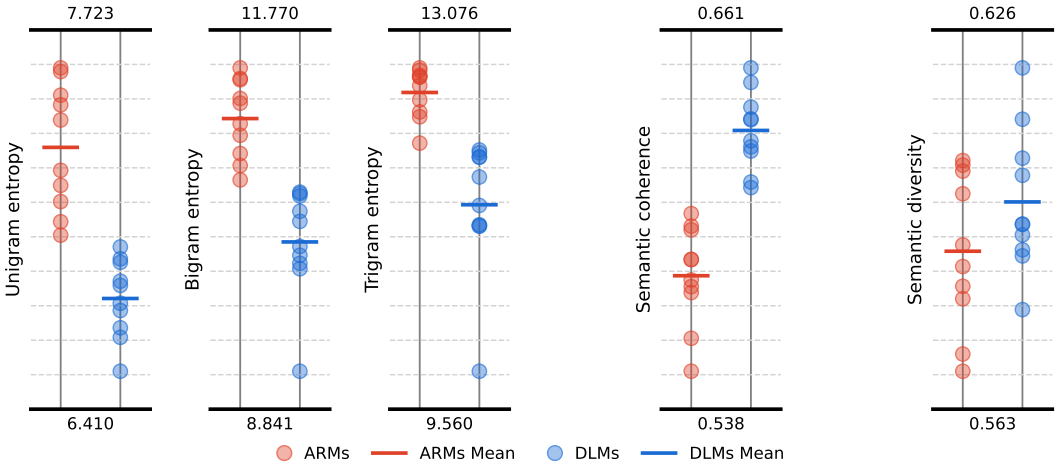
Table 3: Evaluation of 1.1B-parameter ARM and DLM checkpoints from SMDM.



(a) Data filtering ablation.



(b) Response length ablation.



(c) Sentence segmentation ablation.

Figure 5: Robustness checks for the off-the-shelf evaluation, presented in the same format as Figure 4.

B Additional details for controlled experiments on training objectives

B.1 Training objectives selection

We select all possible combinations of the four decomposed components listed in §4.1 subject to two rules. First, when bidirectional context is used, both input masking and label masking are necessarily included: without either of them, the task for some tokens reduces to identity mapping rather than prediction. Second, the weighting function is only varied when label masking is present, since $\omega_t = 1/t$ serves to normalize the loss magnitude. Specifically, it compensates for the fact that the number of loss terms (i.e. masked positions) is proportional to t.

B.2 Experimental setup details

For the main experiments, we use a 120M-parameter LLaMA architecture (Touvron et al., 2023a), configured with a hidden size of 768, 12 attention heads, and 12 layers. We optimize all models using AdamW (Loshchilov & Hutter, 2017) optimizer with a global batch size of 1024 and a peak learning rate of 1e-3, which warms up over the first 100 steps and remains constant thereafter. Training does not incorporate block attention; rather, the attention mask is governed solely by the loss function's context scope (unidirectional or bidirectional).

Text generation is performed using standard sequential autoregressive decoding with a temperature of 1.0, without top-p or top-k sampling. This setup ensures that the generated outputs faithfully reflect the learned distribution, without introducing additional distortion.

As a control variable, we also evaluate the next-token prediction loss at each checkpoint. We compute it using a uniform procedure: sample $x \sim \mathcal{D}_{\text{data}}$; for each $i \in \{2, \ldots, L\}$, mask all tokens x^j with $j \geq i$ to obtain $\tilde{x}$; then feed $\tilde{x}$ into the model and compute the negative log likelihood of predicting x^i from $\tilde{x}$. Averaging this quantity over i and then taking expectation over x gives the next-token prediction loss used in evaluation.

B.3 Quality control metrics

In controlled experiments, we intentionally control generation quality to isolate the influence of the base model's capability. We use next-token prediction loss to measure generation quality, and supplement the evaluation with additional quality metrics:

- **LLM-as-a-judge.** We prompt `qwen3.5-flash` to assign each generated response a quality score from 1 to 10.
- **MAUVE.** MAUVE (Pillutla et al., 2021) measures the gap between model-generated text and human text using divergence frontiers.
- **Generative perplexity.** Generative perplexity evaluates the likelihood of generated text under an external reference model.

As shown in Figure 6 and Figure 7, the LLM-as-a-judge scores, MAUVE, and generative perplexity are all comparable across the eight training objectives. This confirms that our control setup is robust and ensures that the distributional differences are driven by training objectives rather than model capability gaps.

B.4 Semantic coherence validity

Semantic coherence can have two different interpretations depending on the regime. In the normal non-degenerate range, it measures the smoothness of semantic transitions between adjacent sentences. In extreme cases, it can be inflated by trivial repetitive generation. To rule out the latter explanation and validate the metric in our setting, we add repetition diagnostics and unique n-gram analyses.

Repetition detection. We evaluate the passing rate of samples generated in the controlled experiments using off-the-shelf repetition detection criteria (i.e., a 30-character sequence

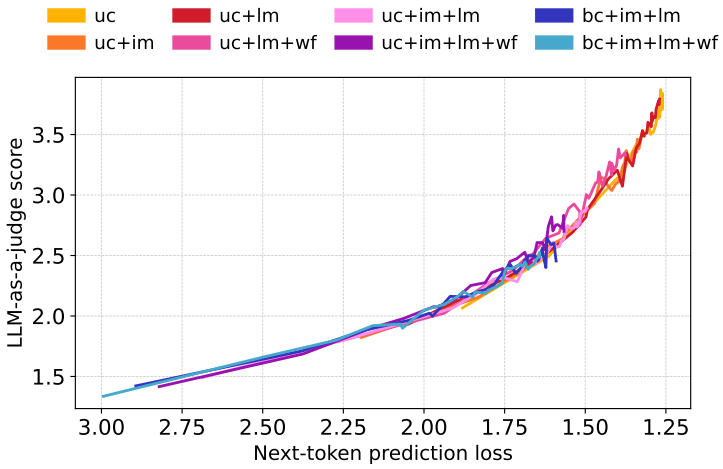


Figure 6: LLM-as-a-judge scores across interpolated training objectives, using the TinyStories dataset, presented in the same format as Figure 2.

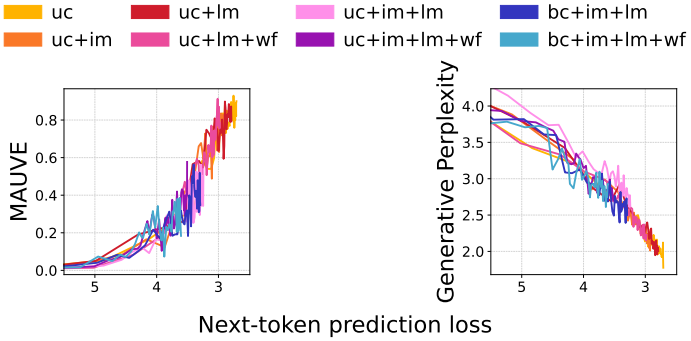


Figure 7: MAUVE and generative perplexity across interpolated training objectives in §4 (seed = 0), presented in the same format as Figure 2. We employ GPT-2 Large as the embedding model for MAUVE, following its original paper, and also use it as the reference model for generation perplexity.

repeated consecutively at least 3 times). As shown in Figure 8, both ARMs and DLMs achieve a passing rate of approximately 99%, confirming the absence of highly repetitive text.

Unique n-gram analysis. We compute the rate of unique n-grams in the generated text. As shown in Figure 9, DLMs and ARMs fall well within normal and healthy ranges and exhibit similar unique n-gram rates, verifying that the higher DLM coherence is not due to degenerate repetition.

B.5 Robustness across seeds, architectures, and datasets

As an additional robustness check, we repeat the main controlled experiments in §4 with three random seeds. Apart from the main experiments, we also verify the results across different model architectures and datasets. For the architecture ablation, we train another set of models using a 30M-parameter Qwen2 architecture (Yang et al., 2024a), configured with a hidden size of 512, 8 attention heads, and 8 layers. Other hyperparameters remain the same as those in §4. For the dataset ablation, we train a set of models on the TinyStories dataset (Eldan & Li, 2023) for 5 epochs each. Because the texts in the TinyStories dataset are shorter, we reduce the context length to 128 and, accordingly, the learning rate to 5e-4.

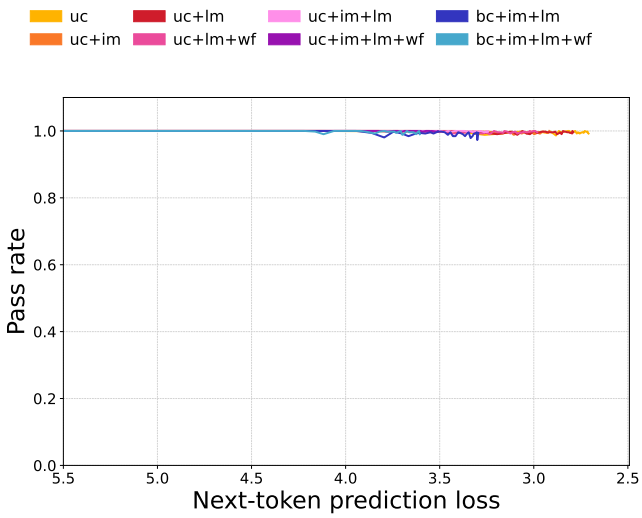


Figure 8: Repetition detection pass rates across interpolated training objectives in §4 (seed = 0), presented in the same format as Figure 2. The pass rate is calculated as the percentage of samples that contain no repetitive text.

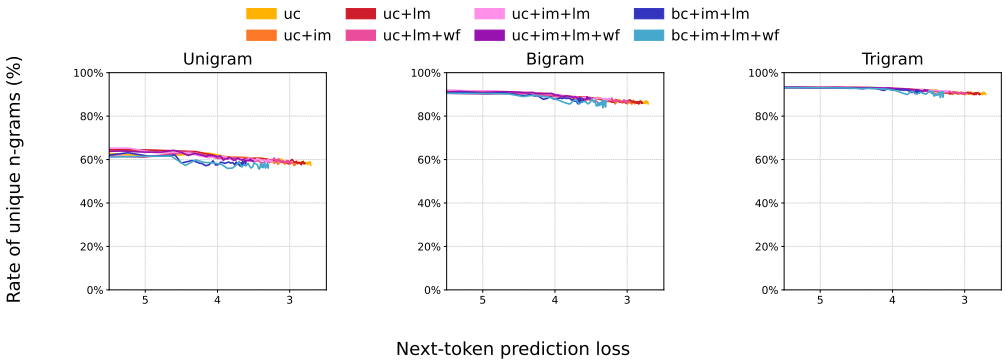


Figure 9: Unique n-gram rates across interpolated training objectives in §4 (seed = 0), presented in the same format as Figure 2.

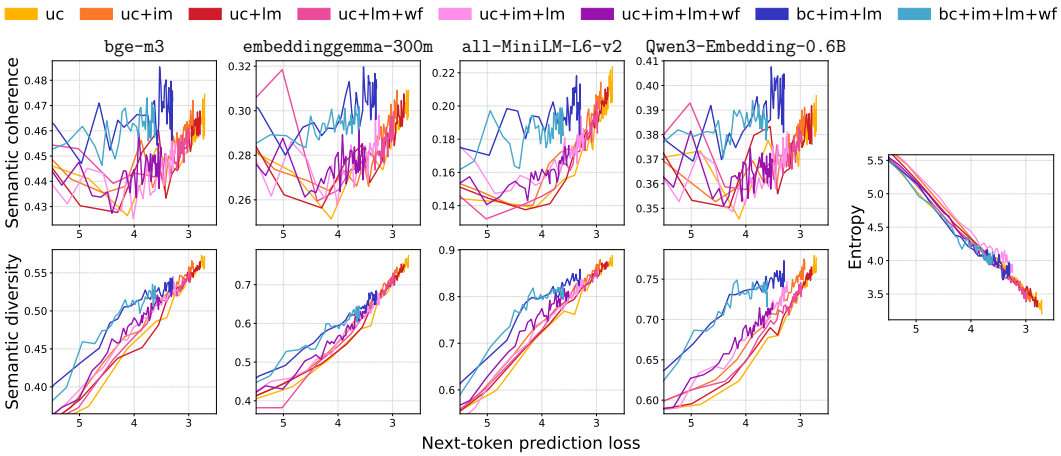
The results for the random seed, architecture, and dataset ablation are shown in Figure 10, Figure 11, and Figure 12, respectively. All additional experiments exhibit the same phenomena as in §4: clustering effects are observed in semantic coherence and semantic diversity, while entropy shows no significant differences, and context scope remains the primary driver. These results further confirm the robustness of our conclusion.

B.6 Within-prompt diversity

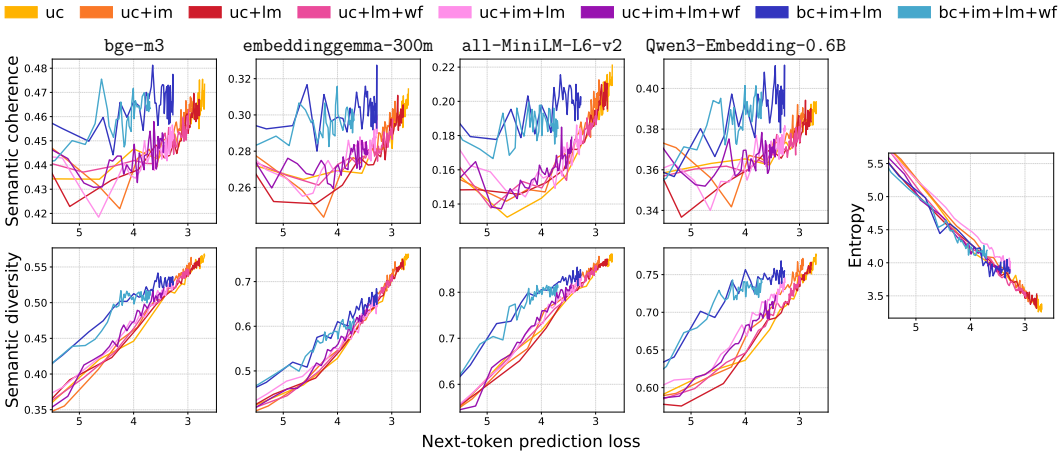
Across-prompt diversity, which we adopt as semantic diversity in our main analysis, measures whether a model possesses rich language patterns to adapt to a variety of prompts, rather than collapsing into homogeneous responses irrespective of the input. In contrast, within-prompt diversity captures the semantic variance generated from a single prompt (Kirk et al., 2023). Formally, it is defined as

$$\mathrm{Div}_{\mathrm{within}}(p_{\mathcal{G}}) := 1 - \mathbb{E}_c \left[\mathbb{E}_{x,x' \sim p_{\mathcal{G}}(\cdot|c)} \langle \mathrm{emb}(x), \mathrm{emb}(x') \rangle \right]. \tag{11}$$

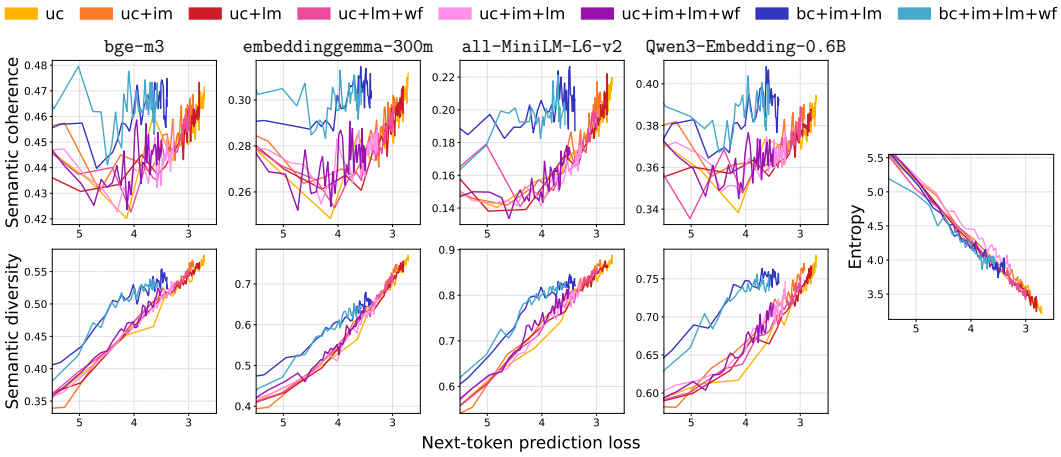
As a complementary analysis, we additionally evaluate within-prompt diversity. We compute it by sampling 64 prompts, generating 8 sequences for each prompt, computing one minus the pairwise cosine similarity among the 8 semantic embeddings, and averaging over prompts. As shown in Figure 13, DLMs still exhibit higher or on-par within-prompt diversity compared to ARMs, though the difference between DLMs and ARMs in within-prompt diversity is less significant than the difference in across-prompt diversity.



(a) seed = 0



(b) seed = 1



(c) seed = 2

Figure 10: Controlled experiments on interpolated training objectives with three different random seeds, presented in the same format as Figure 2.

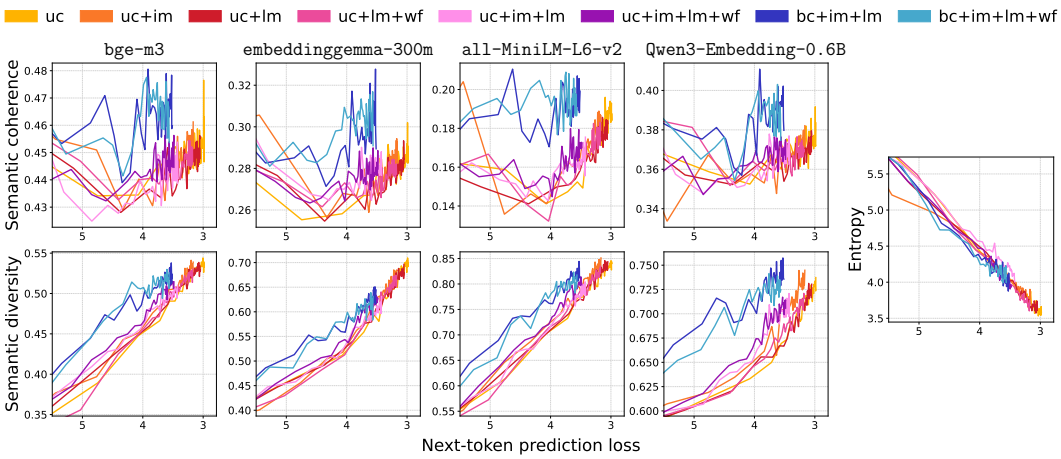


Figure 11: Controlled experiments on interpolated training objectives using the Qwen2 architecture, presented in the same format as Figure 2.

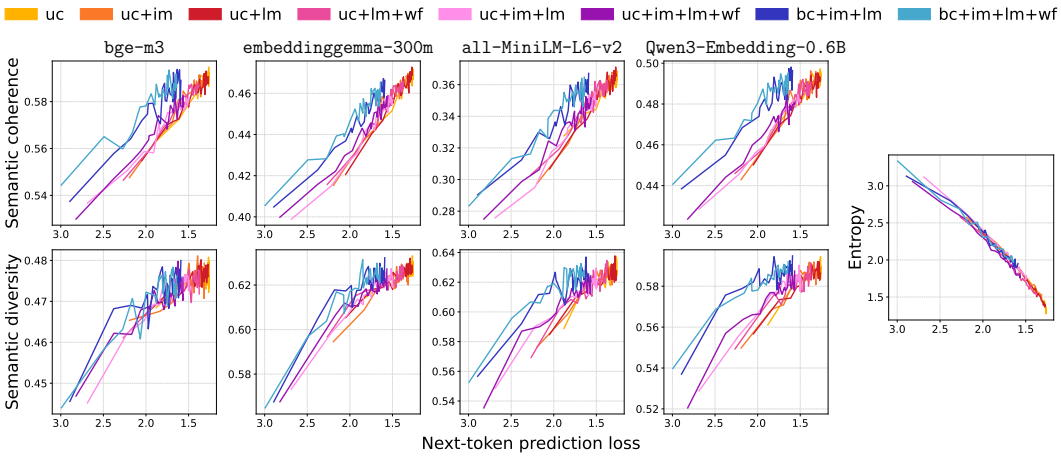


Figure 12: Controlled experiments on interpolated training objectives using the TinyStories dataset, presented in the same format as Figure 2.

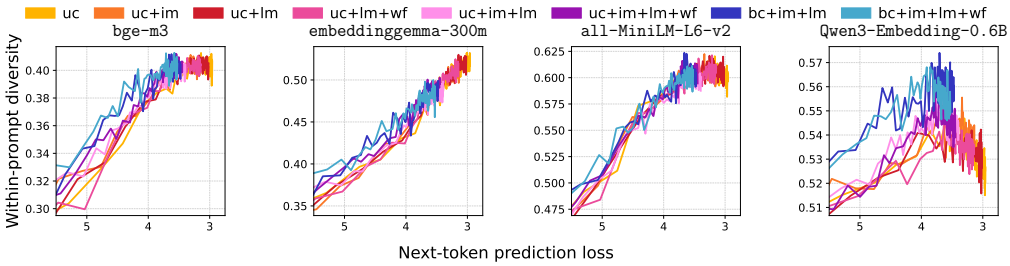


Figure 13: Within-prompt diversity across interpolated training objectives, using the Qwen2 architecture, presented in the same format as Figure 2.

C Supplementary analyses of decoding algorithms

C.1 Proof of entropy reduction

We formally prove that confidence-based strategies reduce full entropy, assuming the data distribution factorizes into independent position-wise distributions, and the DLM is optimally trained.

Throughout this section, we use superscripts to index positions within a sequence. To contextualize the proof, we briefly recall the decoding mechanisms of confidence-based strategies. We assume a block length B and a target number of denoising steps N. Let p_θ be the trained DLM. At each denoising step, given current partially decoded sequence x_t, the model samples a token from $p_\theta(x_0^i \mid x_t)$ for each masked position i within the block.

Low-confidence remasking decodes B/N tokens within this block that have the highest prediction probabilities $p_\theta(x_0^i \mid x_t)$ (i.e. the confidence values), remasking the others. Dynamic low-confidence remasking extends this by additionally decoding any token in the block whose confidence value strictly exceeds a predefined threshold τ, meaning that the number of tokens decoded per step can dynamically increase. Consistent with our experimental design, we set $N = B$, which targets decoding one token per step. Once all tokens within the current block are decoded, the generation process proceeds to the next block.

Theorem 1. *Assume that the data distribution at each position is independent of the others, so that the full distribution factorizes as*

$$p_{\text{data}}(x) = \prod_{i=1}^{L} p_{\text{data}}^i(x^i) =: \prod_{i=1}^{L} q_{x^i}^i \,, \tag{12}$$

and that the DLM is optimally trained, i.e., for any masked position i of a noised input x_t,

$$p_\theta(x_0^i \mid x_t) = p_{\text{data}}^i(x_0^i) = q_{x_0^i}^i. \tag{13}$$

Let p_{lcr} and p_{dlcr} denote the generation distributions induced by low-confidence remasking and dynamic low-confidence remasking, respectively. Then

$$\mathcal{H}(p_{\text{lcr}}) \le \mathcal{H}(p_{\text{data}}), \qquad \mathcal{H}(p_{\text{dlcr}}) \le \mathcal{H}(p_{\text{data}}). \tag{14}$$

Proof. Without loss of generality, we assume that $q_1^i \ge q_2^i \ge \cdots \ge q_{|\mathcal{V}|}^i$ for all i. By definition, low-confidence remasking is a special case of dynamic low-confidence remasking when $\tau = 1$. Thus, it suffices to prove $\mathcal{H}(p_{\text{dlcr}}) \le \mathcal{H}(p_{\text{data}})$.

We assume the model starts generating from an empty prompt, as our proof can generalize the starting state to any sequence.

Let $X \sim p_{\text{dlcr}}$, $X = (X^1, X^2, ..., X^L)$. By the chain rule of entropy,

$$\mathcal{H}(p_{\text{dlcr}}) = \mathcal{H}(X^1) + \mathcal{H}(X^2 \mid X^1) + \cdots + \mathcal{H}(X^L \mid X^{1:L-1}). \tag{15}$$

It is sufficient to show that for every position i and possible prefix sequence $x^{1:i-1}$, we have

$$\forall k,\ \sum_{c=1}^{k} p_{\text{dlcr}}(X^i = c \mid X^{1:i-1} = x^{1:i-1}) \ge \sum_{c=1}^{k} q_c^i. \tag{16}$$

This is a majorization relation (Marshall et al., 1979). Since the entropy function is Schur-concave, the majorization implies

$$\mathcal{H}(X^i \mid X^{1:i-1} = x^{1:i-1}) \le \mathcal{H}(q^i) = \mathcal{H}(p_{\text{data}}^i), \tag{17}$$

and taking expectation over $x^{1:i-1}$ gives

$$\mathcal{H}(X^i \mid X^{1:i-1}) \le \mathcal{H}(p_{\text{data}}^i). \tag{18}$$

Summing over i then yields our desired inequality

$$\mathcal{H}(p_{\text{dlcr}}) \leq \mathcal{H}(p_{\text{data}}). \tag{19}$$

We prove Eq.(16) by introducing latent variable Z, where Z^i denotes the decoding step at which X^i is generated. It is sufficient to show that for any configuration $z^1, \ldots, z^L$ and any realization of the other tokens $x^1, \ldots, x^{i-1}, x^{i+1}, \ldots, x^L$ such that

$$p_{\text{dlcr}}(X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}) > 0, \tag{20}$$

we have for all k,

$$\sum_{c=1}^{k} p_{\text{dlcr}}\Big(X^i = c \,\Big|\, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big) \geq \sum_{c=1}^{k} q_c^i. \tag{21}$$

This is sufficient because the conditional distribution given $X^{1:i-1}$ can be expressed as a mixture over the remaining variables, and majorization is preserved under such mixtures.

Let

$$S = \{j : z^j = z^i\}, \qquad T = \{j : (z^j \geq z^i) \wedge (j \text{ and } i \text{ are in the same block})\}, \tag{22}$$

be the set of positions decoded at step z^i and the set of positions within the same block as i that are not decoded before step z^i. We consider the following two cases.

Case 1: $|S| = 1$. In this case, the decoding step of X^i decodes exactly one position, namely i. The event that X^i is the only token that is decoded at this step occurs if and only if all positions in $T \setminus \{i\}$ are remasked at the same step. Under the optimal model assumption, decoding $X^i = c$ requires that for every $j \in T \setminus \{i\}$, the sampled token at position j has confidence no larger than q_c^i and does not exceed the threshold τ. We claim that

$$\begin{aligned} &p_{\text{dlcr}}\Big(X^i = c \,\Big|\, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big) \\ &\qquad \propto q_c^i \prod_{j \in T \setminus \{i\}} \sum_{r:\, q_r^j \leq \min(\tau, q_c^i)} q_r^j. \end{aligned} \tag{23}$$

To understand this proportionality, note that by Bayes' theorem, the conditional probability is proportional to the joint probability:

$$\begin{aligned} &p_{\text{dlcr}}\Big(X^i = c \,\Big|\, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big) \\ &\propto p_{\text{dlcr}}\Big(X^i = c, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big). \end{aligned} \tag{24}$$

Expanding this joint probability into a product of step-wise probabilities reveals that most of the factors are independent of c and can be absorbed into the normalization constant. The only terms dependent on c occur at the specific step where X^i is decoded: (1) the probability q_c^i of sampling $X^i = c$, and (2) the probability that tokens sampled at the remaining positions $T \setminus \{i\}$ fall below $\min(\tau, q_c^i)$ and are thus remasked.

From Eq.(23), define

$$w_c := \prod_{j \in T \setminus \{i\}} \sum_{r:\, q_r^j \leq \min(\tau, q_c^i)} q_r^j. \tag{25}$$

Since q_c^i is non-increasing in c, the weight w_c is also non-increasing in c. Therefore, after normalization, the distribution

$$p_{\text{dlcr}}\Big(X^i = c \,\Big|\, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big) \propto q_c^i w_c \tag{26}$$

majorizes q^i, i.e.,

$$\forall k,\ \sum_{c=1}^{k} p_{\text{dlcr}}\Big(X^i = c \,\Big|\, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big) \geq \sum_{c=1}^{k} q_c^i. \tag{27}$$

Case 2: $|S| > 1$**.** In this case, the decoding step of X^i simultaneously decodes multiple positions. Thus X^i is decoded to c if and only if its confidence q_c^i exceeds the threshold τ. Hence,

$$\begin{aligned} p_{\text{dlcr}}\Big(X^i = c \,\Big|\, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big) \\ \propto q_c^i \mathbf{1}[q_c^i > \tau]. \end{aligned} \tag{28}$$

As before, this proportionality holds because expanding the joint probability reveals that most step-wise factors are independent of c and absorbed into the normalization constant. The only terms dependent on c occur at the specific step where X^i is decoded: (1) the probability q_c^i of sampling c, and (2) the threshold constraint $\mathbf{1}[q_c^i > \tau]$.

Let

$$w_c := \mathbf{1}[q_c^i > \tau]. \tag{29}$$

Since q_c^i is non-increasing in c, the weight w_c is also non-increasing in c. Therefore, after normalization, the distribution

$$p_{\text{dlcr}}\Big(X^i = c \,\Big|\, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big) \propto q_c^i w_c \tag{30}$$

majorizes q^i, i.e.,

$$\forall k,\ \sum_{c=1}^{k} p_{\text{dlcr}}\Big(X^i = c \,\Big|\, X^{1:i-1} = x^{1:i-1}, X^{i+1:L} = x^{i+1:L}, Z^{1:L} = z^{1:L}\Big) \geq \sum_{c=1}^{k} q_c^i. \tag{31}$$

Combining the two cases, we obtain Eq.(21), completing the proof. □

Remark. Ambiguity may arise when there exist i, j and i', j' such that $q_j^i = q_{j'}^{i'}$, as it is not defined which token takes priority for remasking. However, this is not a significant issue: first, such exact equality is extremely rare in practice; second, even if it occurs, as long as we assign an oracle order to these tokens, Eq.(25) remains non-increasing in c, and the rest of the proof remains unaffected and valid.

C.2 Bias elimination experiments

In §5.2, we suggest that the entropy reduction observed in confidence-based remasking strategies stems from a theoretical distributional bias. To support this explanation, we introduce a simple modification to the confidence-based strategy: after selecting the decoding positions based on confidence values, we *resample* the decoded tokens at these positions. This breaks the correlation between position selection and the sampled tokens, thereby eliminating the bias.

As shown in Figure 3, this modification restores n-gram entropy to the baseline level and invalidates inequality (ii) in Eq.(8). This confirms that the bias causes the n-gram entropy reduction. These phenomena are consistent across seeds, datasets and architectures (Figure 15, Figure 16 and Figure 17).

Beyond entropy, bias elimination also removes the influences of confidence-based strategies on semantic coherence and semantic diversity (Figure 18, Figure 19, and Figure 20). This suggests that the bias-based distinction could be generalized to these metrics as well.

C.3 Ablations across seeds, datasets and architectures

Consistent with Appendix B.5, we further validate our main findings by repeating the main experiments with two additional random seeds, using the Qwen2 architecture, and on the TinyStories dataset. As shown in Figure 15, Figure 16, and Figure 17, we consistently observe entropy reduction, including decreases in n-gram entropy and the validity of inequality (ii). We also observe that bias elimination remains effective. These results confirm the robustness of our findings.

C.4 Role and desirability of confidence-based bias

We do not treat confidence-based bias as a flaw, as it is mathematically undesirable but empirically useful. Mathematically, the confidence-based bias deviates from the true optimal data distribution. Empirically, however, it suppresses the long tail of uncertain tokens to ensure local fluency.

We additionally conduct LLM-as-a-judge evaluations on the decoding strategies (Figure 14). Confidence-based remasking slightly increases quality, while bias elimination returns the score to the baseline level of other strategies. Thus, the confidence-based strategies aid perceived quality, at the cost of introducing a bias that deviates from the true data distribution and potentially reducing diversity on certain datasets, as detailed in Appendix C.5.

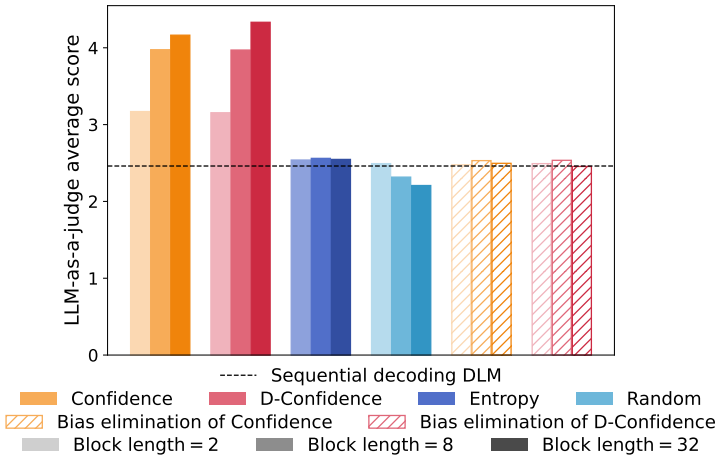
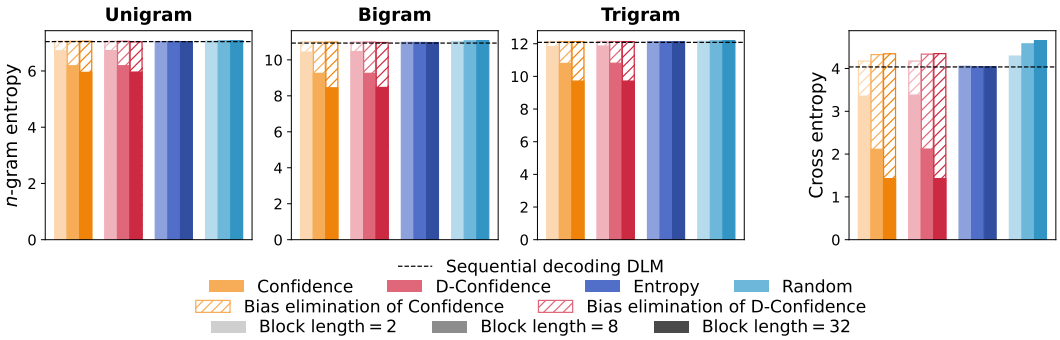


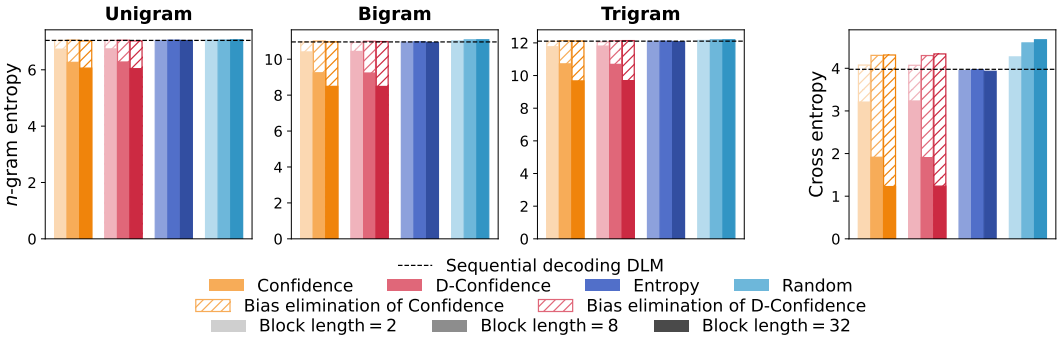
Figure 14: LLM-as-a-judge scores across different remasking strategies for the TinyStories dataset experiment, presented in the same format as Figure 3.

C.5 Effects on semantic coherence and semantic diversity

We also evaluate semantic coherence and semantic diversity across all remasking strategies and configurations. As shown in Figure 18, Figure 19, and Figure 20, non-confidence-based strategies have much smaller effects on both metrics compared to confidence-based strategies. Confidence-based strategies consistently increase semantic coherence, while their effect on semantic diversity depends highly on the dataset. As shown in Figure 18 and Figure 19, they increase semantic diversity on the FineWeb dataset, consistent across seeds and architectures. However, when we switch to the TinyStories dataset (Figure 20), confidence-based strategies reduce semantic diversity.



(a) seed = 1



(b) seed = 2

Figure 15: n-gram entropy and cross-entropy (defined in Eq.(9)) across different remasking strategies with other seeds. The format follows Figure 3, which displays the seed = 0 results.

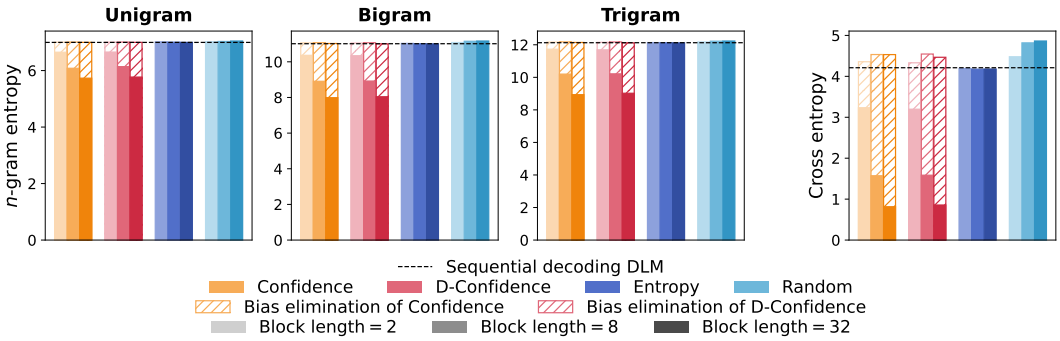


Figure 16: n-gram entropy and cross-entropy (defined in Eq.(9)) across different remasking strategies for the Qwen2 architecture experiment, presented in the same format as Figure 3.

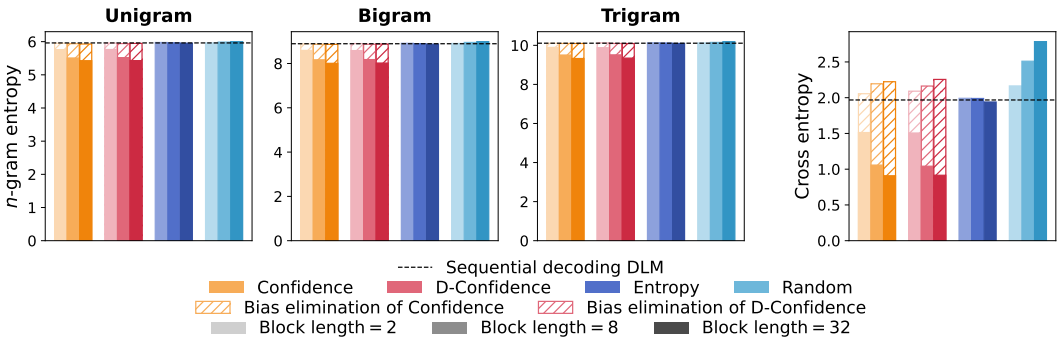
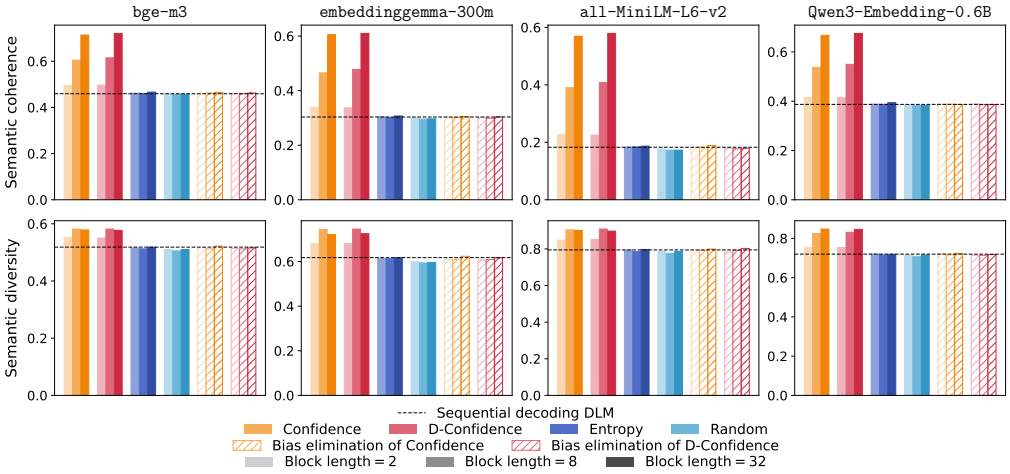
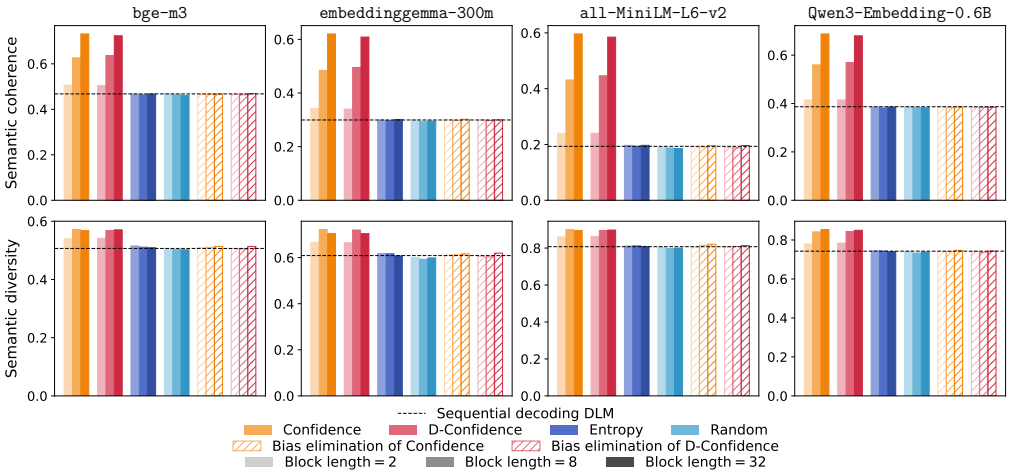


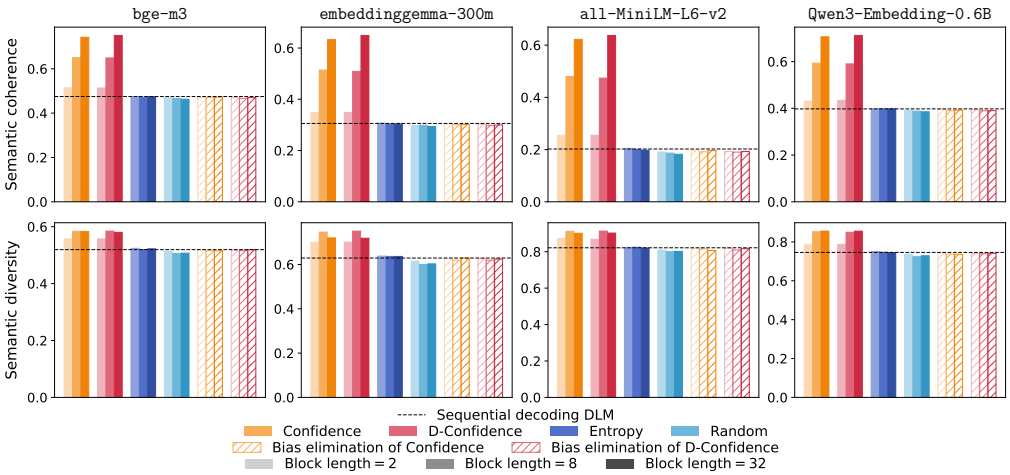
Figure 17: n-gram entropy and cross-entropy (defined in Eq.(9)) across different remasking strategies for the TinyStories dataset experiment, presented in the same format as Figure 3.



(a) seed = 0



(b) seed = 1



(c) seed = 2

Figure 18: Semantic coherence and semantic diversity across different remasking strategies with three random seeds.

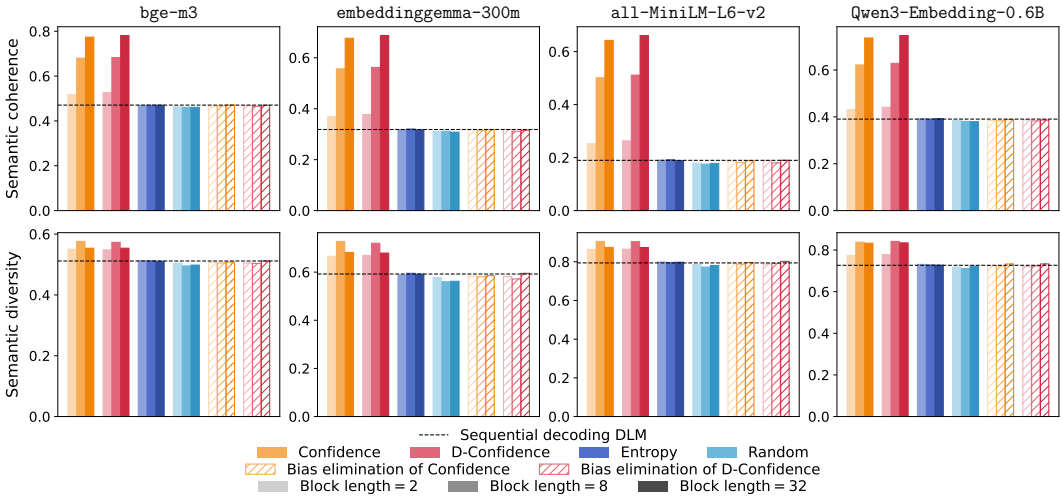


Figure 19: Semantic coherence and semantic diversity across different remasking strategies for the Qwen2 architecture experiment.

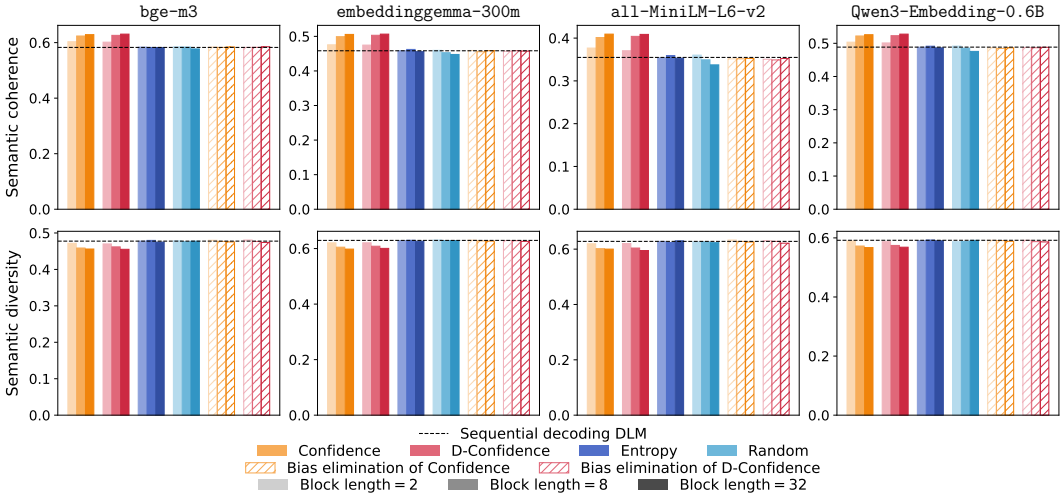


Figure 20: Semantic coherence and semantic diversity across different remasking strategies for the TinyStories dataset experiment.