# DIFFUMAMBA: HIGH-THROUGHPUT DIFFUSION LMS WITH MAMBA BACKBONE

**Vaibhav Singh**[12]*, **Oleksiy Ostapenko**[3], **Pierre-André Noel**[3], **Eugene Belilovsky**[12], **and Torsten Scholak**[3]
[1]Concordia University, [2]Mila, [3]ServiceNow Research, Montreal

## ABSTRACT

Diffusion language models (DLMs) have emerged as a promising alternative to autoregressive (AR) generation, yet their reliance on Transformer backbones limits inference efficiency due to quadratic attention or KV-cache overhead. We introduce ***DiffuMamba***, a masked diffusion language model built on a bidirectional Mamba backbone that combines the diffusion objective with linear-time sequence modeling, and ***DiffuMamba-H***, a hybrid variant with interleaved attention. Across scales up to 1.3B parameters, our models match Transformer-based diffusion in downstream performance while achieving up to **8.2**× and **4.3**× higher inference throughput, respectively, on long sequences. We further present a systematic analysis of inference efficiency across modern DLM variants combining asymptotic complexity with empirical measurements. Notably, cache-efficient block diffusion with Mamba mixers emerges as the only strategy that scales linearly with sequence length and achieves the strongest performance across all baselines, suggesting a promising direction for future diffusion-based generation systems.

## 1 INTRODUCTION

Most state-of-the-art Large Language Models (LLMs) (Brown et al., 2020; Achiam et al., 2023; Chowdhery et al., 2023; Touvron et al., 2023a) rely on full multi-head attention (MHA) (Bahdanau et al., 2014; Vaswani et al., 2017) and are trained with an *autoregressive* (AR) objective, predicting each token conditioned on all previous tokens. Despite its simplicity and effectiveness, this paradigm exhibits several fundamental limitations. AR decoding is inherently sequential, causing inference latency to grow linearly with output length. Moreover, AR training is data-inefficient, as each example provides supervision for only a single next-token prediction, limiting the learning signal available from finite datasets (Ni et al., 2025a). This issue is increasingly critical as high-quality data, rather than compute, becomes a primary scaling bottleneck. In addition, AR Transformers face significant memory and compute constraints: the KV cache grows linearly with context length, while attention incurs quadratic complexity in sequence length, restricting long-context training and inference (Zhou et al., 2024; Tay et al., 2020). Together, these limitations reduce throughput and test-time scalability, motivating the exploration of alternative architectures.

Diffusion Language Models (DLMs) (Li et al., 2025) provide a flexible alternative to autoregressive generation by iteratively denoising entire corrupted sequences in parallel, enabling non-sequential multi-token generation, partial infilling, and self-correction, and potentially faster generation when few denoising steps suffice. However, *existing DLMs rely on Transformer backbones*, making iterative denoising compute- and/or memory-intensive and yielding substantially lower throughput than AR models, particularly at long sequence lengths. In standard DLMs, each denoising step re-encodes the full sequence using bidirectional attention, as token representations evolve across steps and cannot be incrementally reused, resulting in per-step costs that scale quadratically with sequence length. While several approaches introduce KV caching (Wu et al., 2025; Ma et al., 2025; Nguyen-Tri et al., 2025; Fathi et al., 2025), the core limitation remains: *cache growth with sequence length induces increasing memory traffic, becoming the dominant inference bottleneck.* Consequently, per-token latency grows with both sequence length and the number of denoising steps due to attention and cache-management

*Work done during research internship at ServiceNow Research, Montreal, Canada. Correspondence: vaibhav.singh@mila.quebec, oleksiy.ostapenko@servicenow.com

overheads, leading to a paradox: *DLMs promise flexible generation yet remain constrained by memory overhead and periodic cache recomputation inherent to Transformer backbones* (Wu et al., 2025).

In parallel with advances in DLMs, state-space models (SSMs) have emerged as powerful sequence mixers with linear-time complexity (Gu et al., 2022; Poli et al., 2023; Fu et al., 2023; Gu & Goel, 2023; Dao & Gu, 2024). Recent work shows that SSMs and SSM–Transformer hybrids can match or outperform Transformers while achieving substantially higher inference throughput (Gu & Goel, 2023; Somvanshi et al., 2025; Wang et al., 2025a). Despite this promise, SSMs remain largely unexplored in the context of DLMs, motivating a key question: *can structured recurrence serve as an effective language denoiser while enabling faster inference?* Our work investigates this intersection by combining efficient SSM backbones with masked discrete diffusion for language modeling.

The main contributions of this work are:

- **New architectural direction.** We propose ***DiffuMamba***, which replaces Transformer denoisers with bidirectional Mamba-2 mixers for discrete masked diffusion language modeling. We further introduce ***DiffuMamba-H***, a sparse hybrid variant that interleaves attention by inserting one Transformer block every five Mamba blocks ($\approx 20\%$ attention), starting with an attention block. Together, these models demonstrate that iterative denoising does not inherently require dense attention, positioning linear-time backbones as a scalable alternative for diffusion language models.
- **Controlled evaluation across scales.** We conduct a systematic comparison between DiffuMamba and Transformer-based diffusion models (`DiffuTran`) under identical training data, tokenization, noise schedules, and decoding steps at three parameter budgets (240M, 0.5B, and 1.3B). Our experiments demonstrate that ***DiffuMamba*** and ***DiffuMamba-H*** can match the modeling quality of the full transformer.
- **Comprehensive throughput benchmarking.** We present an extensive asymptotic and empirical analysis of modern DLM inference strategies, scaling generation length beyond 100k tokens, a regime particularly relevant for complex reasoning workloads. Across all evaluated inference algorithms, Mamba-backed DLMs consistently outperform Transformer-based counterparts: by up to **8.2×** in full-sequence denoising (Figures 2a and 2b) and by **2.3×** in the most efficient block-wise autoregressive denoising setting (Figures 2c and 2d). Notably, the latter is the only inference strategy that scales linearly with sequence length, and when combined with Mamba-based denoisers, it outperforms all other baselines, outlining a promising path for future diffusion-based generation systems.

## 2 Related Work

**Diffusion Language Models (DLMs)** Diffusion modeling replaces left-to-right decoding with iterative denoising. While early work focused on continuous domains (Ho et al., 2020; Song & Ermon, 2021; Rombach et al., 2022; Peebles & Xie, 2023), recent research has adapted diffusion to *discrete* text. Austin et al. (2021) introduced structured transition matrices with absorbing states, and DiffusionBERT (He et al., 2023) employed a masked corruption process paired with a BERT-style denoiser. Subsequent studies have further refined masking strategies, noise schedules, and sampling procedures (Chen et al., 2023; Lou et al., 2023; Varma et al., 2025; Fathi et al., 2025). Masked diffusion approaching autoregressive quality (Sahoo et al., 2024; 2025) enabled scalable models (Ye et al., 2025; Nie et al., 2025) competitive with LLaMA (Touvron et al., 2023b), with further gains from AR adaptation and instruction tuning (Gong et al., 2025; Zhu et al., 2025). To enhance inference efficiency, recent acceleration techniques (Ma et al., 2025; Wu et al., 2025; Liu et al., 2025b; Israel et al., 2025; Wang et al., 2025b) use approximate KV-caching mechanisms to speed up DLM inference. In contrast, `DiffuMamba-H` achieves its speedups without relying on KV caching, though its attention layers remain compatible with these optimizations.

**State-Space Models (SSMs) for Sequence Modeling** SSMs introduce structured recurrences computable via fast convolutions or scans, achieving linear-time scaling with sequence length. Early models like S4 (Gu et al., 2022) demonstrated strong long-context capacity, while Hyena (Poli et al., 2023) and H3 (Fu et al., 2023) explored alternative structured operators for language modeling. Mamba (Gu & Goel, 2023) extended this line by introducing input-conditioned selective state spaces,

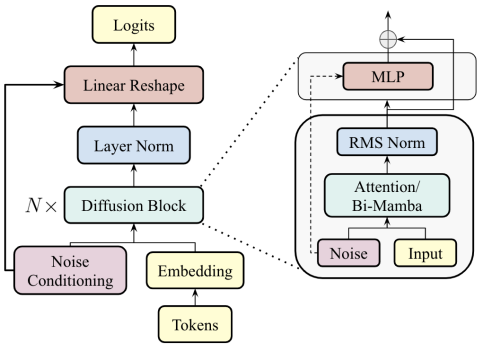



Figure 1: Schematic diagram of our proposed `DiffuMamba` architecture where mixer blocks replaces attention layers with bidirectional Mamba layers. For `DiffuMamba-H` we have interleaved attention layers after every $N$ Mamba layers. Attention provides global token interactions while Mamba enables efficient state space sequence modeling, allowing the hybrid denoiser to capture both long-range dependencies and local temporal dynamics with significantly improved efficiency. In our experiments, we fix $N = 5$.

matching or surpassing Transformers in accuracy with lower latency and memory use. Somvanshi et al. (2025) further survey the breadth of SSM architectures and highlight their efficiency advantages. In autoregressive language modeling, SSMs rival Transformers at comparable scales while offering higher tokens-per-second throughput. More recent works have evolved various linear-complexity LMs and hybrids (Wang et al., 2025a; Ostapenko et al., 2025), ranging from vector recurrences (De et al., 2024) to advanced gating mechanisms (Yang et al., 2024). Although SSM backbones have been used in diffusion models for non-textual domains such as generative modelling in biological sequences (Sahoo et al., 2024), text diffusion LMs continue to rely on full-attention denoisers, leaving open whether SSM recurrences can replace attention for iterative denoising.

**Diffusion with Mamba/SSMs in Vision.** In vision, several studies have replaced Transformer-based DiT backbones with state-space models (SSMs), particularly Mamba variants (Gu & Goel, 2023; Ergasti et al., 2025; Dang et al., 2024), to improve diffusion efficiency. The Diffusion Mamba (DiM) family (Teng et al., 2024; Mo, 2025) achieves high-resolution image synthesis with multi-directional scanning and local feature enhancement, yielding notable throughput gains over attention-based models. VM-DDPM (Ju & Zhou, 2024) fuses convolutional locality with SSM-based global modeling to enhance structural fidelity in medical image generation. ZigMa (Hu et al., 2024) introduces zigzag scanning for faster, more memory-efficient diffusion while maintaining competitive quality, and Phung et al. (2024) incorporates wavelet transforms to strengthen local inductive biases, achieving faster convergence and favorable quality–efficiency trade-offs. Beyond diffusion, Zhu et al. (2024) generalizes Mamba as a vision backbone, and recent surveys (Xu et al., 2024; Liu et al., 2025a; Ergasti et al., 2025; Wang et al., 2024; Dang et al., 2024) map its rapid adoption across segmentation, restoration, and dense prediction. Collectively, these works show that replacing attention with Mamba in multi-step image diffusion preserves or improves quality while substantially lowering per-step compute.

## 3 METHOD

### 3.1 DIFFUSION MAMBA LANGUAGE MODELS

We propose `DiffuMamba`, an MDM whose denoiser replaces the Transformer encoder with a *bidirectional state-space Mamba (BiMamba)* backbone (Zhu et al., 2024; Sahoo et al., 2024; Schiff et al., 2024), preserving the probabilistic structure of masked discrete diffusion while enabling *linear-time* inference and substantially reduced memory overhead during multi-step denoising. We refer to transformer based DLM in our work as `DiffuTran` for simplicity.

**Architecture details.** We now describe the BiMamba denoiser architecture used in `DiffuMamba` also shown in Figure 1. Let $\mathbf{x} \in \mathbb{R}^{B\times L\times d}$ denote a batch of token embeddings, where $B$ is the batch size, $L$ the sequence length, and $d$ the hidden dimension. Each block is composed of two independent Mamba layers: one processes the sequence in the forward direction, and the other processes the

sequence in the reverse direction

$$\mathbf{h}_i^{\rightarrow} = A_f * \mathbf{h}_{i-1}^{\rightarrow} + B_f * \mathbf{x}_i, \quad i = 1, \ldots, L, \tag{1}$$

$$\mathbf{h}_i^{\leftarrow} = A_b * \mathbf{h}_{i+1}^{\leftarrow} + B_b * \mathbf{x}_i, \quad i = L, \ldots, 1. \tag{2}$$

Here $A_f, B_f, A_b, B_b \in \mathbb{R}^{k \times d}$ are learnable state-transition kernels implemented as 1D causal and anti-causal convolutions or efficient scan operations (Gu & Goel, 2023). The two directional feature streams are fused through simple *additive integration*

$$\text{Mamba}(x_i) = \mathbf{h}_i = \mathbf{h}_i^{\rightarrow} + \mathbf{h}_i^{\leftarrow}, \tag{3}$$

providing a symmetric context representation while maintaining numerical stability. The resulting hidden sequence $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_L)$ is then normalized and passed through a lightweight feed-forward projection before residual addition.

Each diffusion block applies noise-conditioned Mamba mixing followed by an MLP refinement with residual connections, and the resulting hidden states are propagated through a stack of such blocks to predict token logits at each denoising step. The model defines a categorical distribution over clean tokens and is trained using the standard masked diffusion objective, preserving probabilistic consistency with absorbing-state discrete diffusion. We defer the diffusion objective, full architectural formulation, conditioning mechanism, and training details to Appendix A and B.1. By replacing attention with bidirectional state-space dynamics, `DiffuMamba` achieves efficient and expressive denoising with linear scaling in both sequence length and memory.

Recent results in large-scale autoregressive modeling show that hybrid architectures (Lieber et al., 2024; Bae et al., 2025; Zuo et al., 2025; Wang et al., 2025a) combining attention and state-space layers can outperform both pure Transformer and pure Mamba models under similar compute budgets, leveraging complementary strengths in local recurrence and global dependencies. Motivated by these findings, we investigate hybrid DLMs by interleaving bidirectional Mamba and Transformer blocks, inserting one attention mixer every $N$ Mamba mixer blocks[1] (with $N = 5$ in all experiments). Extending this paradigm from AR to diffusion, we examine whether similar complementarities hold under masked denoising objectives.

Table 1: Zeroshot Perplexities (PPL ↓) across Benchmarks for Different Model Sizes and Configurations. Best PPL are highlighted in blue and second best are underlined. We observe that ***DiffuMamba-H*** (1.3B) achieves the best overall performance across all configurations, with ***DiffuMamba*** as the next strongest model. At smaller scales (240M), however, ***DiffuMamba*** delivers performance comparable to its attention-based diffusion counterpart, indicating that the advantages of hybridization become more pronounced at larger model sizes.

| Model Size | Configuration | PTB | WikiText | LM1B | Lambada | AG News | PubMed | ArXiv | Avg. |
|---|---|---|---|---|---|---|---|---|---|
| 240M | `DiffuTran` | 153.47 | 49.06 | 142.25 | 49.06 | 105.06 | 29.83 | 43.79 | 81.79 |
| | `DiffuMamba` | 99.04 | 51.35 | 169.92 | 46.66 | 112.59 | 33.98 | 33.20 | **78.11** |
| | `DiffuMamba-H` | 147.97 | 50.96 | 145.25 | 46.43 | 114.20 | 31.94 | 30.99 | 81.11 |
| 0.5B | `DiffuTran` | 112.69 | 40.76 | 103.51 | 39.73 | 81.49 | 25.51 | 25.37 | 61.15 |
| | `DiffuMamba` | 116.29 | 42.56 | 117.42 | 40.31 | 90.10 | 27.60 | 27.79 | 65.87 |
| | `DiffuMamba-H` | 110.42 | 40.25 | 100.59 | 39.30 | 78.38 | 26.12 | 25.84 | **60.13** |
| 1.3B | `DiffuTran` | 101.14 | 36.63 | 99.72 | 36.73 | 67.53 | 23.03 | 23.25 | 55.29 |
| | `DiffuMamba` | 99.04 | 34.74 | 102.01 | 36.04 | 68.75 | 22.35 | 22.98 | 55.13 |
| | `DiffuMamba-H` | 96.51 | 31.92 | 92.83 | 34.04 | 63.22 | 19.05 | 20.67 | **51.18** |

## 4 EXPERIMENTS

We evaluate three architectures under an identical diffusion objective, masking process, and noise schedule to ensure a controlled comparison. The first configuration, `DiffuTran`, employs an MHA-based mixer in every diffusion block; the second, `DiffuMamba`, replaces all such mixers with bidirectional Mamba blocks; and the third, `DiffuMamba-H`, adopts a hybrid design that interleaves MHA and Mamba blocks. Section 3.1 details the corresponding design principles. Across all variants, the overall architecture, data flow, and conditioning mechanisms remain identical, with only the internal mixer type varying, as illustrated in Figure 1. To maintain comparable parameter budgets, the MLP expansion ratio in `DiffuMamba` and hybrid configurations is set to half that of the MHA-only

[1] $N$ is a hyperparameter that can be tuned to balance model performance and inference throughput.

model, while other hyperparameters, such as the number of blocks, hidden size, etc are kept constant as detailed in Table 5. This design ensures a fair, architecture-isolated evaluation of modeling quality, efficiency, and inference throughput. Our goals are to answer:

1. **Performance & Scaling:** Can Mamba-based DLMs reach or even surpass the performance of standard Transformer DLMs at different parameter scales and training tokens scales?
2. **Throughput and Latency:** Does an SSM backbone yield higher tokens/sec and lower per-step latency with identical inference settings?

### 4.1 Data, Tokenization, and Pretraining Setup

We evaluate `DiffuMamba` and `DiffuMamba-H` against `DiffuTran` at three parameter scales: 240M, 0.5B, and 1.3B, and additionally compare throughput and wall-clock latency at the 1.3B scale. All models are trained on the DCLM dataset (Li et al., 2024) using the GPT2 tokenizer (Radford et al., 2019), with a fixed context length of 1,024 tokens. We train all models with, under approximate Quokka optimal compute budget (Ni et al., 2025b) as given in the Appendix, Table 5 and Table 6. For measuring inference throughput, we record the wall-clock model latency and tokens-per-second throughput on a single NVIDIA H100 GPU using bf16 precision and PyTorch backend using CUDA-graphs. Each measurement averages over 5 full diffusion decoding runs following 3 warm-up iterations.

### 4.2 Modeling Quality

Table 2: Validation Perplexity (Val. PPL ↓) under Chinchilla (Hoffmann et al., 2022) and Quokka (Ni et al., 2025b) Compute Budgets. Best PPL are highlighted in blue and second best are underlined. For the 1.3B model, `DiffuMamba-H` improves over `DiffuTran` by **2%** and yields a **4.3**× inference speedup (see Figure 2)

| Config | 240M | | 0.5B | | 1.3B | |
|---|---|---|---|---|---|---|
| | Chinchilla | Quokka | Chinchilla | Quokka | Chinchilla | Quokka |
| `DiffuTran` | <u>43.82</u> | 32.11 | 31.03 | 25.07 | 25.01 | 22.72 |
| `DiffuMamba` | 44.09 | 33.01 | <u>30.78</u> | 25.46 | <u>23.96</u> | <u>21.41</u> |
| `DiffuMamba-H` | 42.67 | <u>32.49</u> | 29.06 | <u>25.14</u> | 22.89 | 20.17 |

We first measure validation perplexity (Val. PPL) for different configuration of models as shown in Table 2. Across all model sizes, hybrid `DiffuMamba-H` consistently outperform their pure-attention counterparts under both Chinchilla and Quokka compute budgets. At the 1.3B scale, `DiffuTran` attains 25.01 and 22.72 PPL, but `DiffuMamba-H` achieves the best results with 22.89 and 20.17, delivering roughly a 2% perplexity reduction. At smaller scales (240M and 0.5B), `DiffuMamba-H` remain competitive, often securing the best or second-best perplexities, demonstrating that hybridization provides consistent gains across pre-training regimes.

We next assess models' ability to generalize zero-shot to held-out datasets. The results are presented in Table 1. Following Sahoo et al. (2024), we evaluate PPL upper bounds on Penn Treebank (PTB; Marcus et al., 1993), WikiText (Merity et al., 2017), LM1B (Chelba et al., 2013), Lambada (Paperno et al., 2016), AG News (Zhang et al., 2015), and Pubmed/Arxiv subsets of Scientific Papers (Cohan et al., 2018). All models are evaluated without fine-tuning.

At the 240M scale, `DiffuTran` outperforms `DiffuMamba` and `DiffuMamba-H` on 4 of 7 datasets, suggesting that Mamba-based models struggle to generalize at smaller scales, a trend also reflected in Table 2 where `DiffuTran` attains lower validation perplexity. At 0.5B, the benefits of hybridization become evident, with `DiffuMamba-H` leading on 5 of 7 datasets. This advantage further strengthens at 1.3B, where `DiffuMamba-H` outperforms `DiffuTran` across all datasets, while `DiffuMamba` ranks second in most cases (5 of 7), indicating that Mamba's sequence-modeling inductive bias scales more effectively than pure attention for diffusion denoising. These results reveal a clear trend: while Mamba enables efficient long-range token mixing, interleaving attention layers in `DiffuMamba-H` captures complementary global dependencies, consistent with findings in AR linear attention (Wang et al., 2025a), and consistently improves generalization at larger scales.

Table 3: Asymptotic inference efficiency at batch size $B$=1. We report FLOPs ($F$), memory operations ($M$), arithmetic intensity (AI= $F/M$), and throughput ($T$). $L$ is sequence length, $d$ hidden dimension, $K$ diffusion steps, $A$ Mamba state size, $G$ block size, and $p$ the step scaling factor. †Cost includes block diffusion steps and cache recomputation.

| Model | FLOPs ($F$) | Memory ($M$) | AI ($F/M$) | Throughput ($T$) |
|---|---|---|---|---|
| AR | $\mathcal{O}(Ld^2 + L^2d)$ | $\mathcal{O}(Ld^2 + L^2d)$ | $\mathcal{O}(1)$ | $\mathcal{O}\left(\frac{1}{Ld+d^2}\right)$ |
| DiffuTran | $\mathcal{O}(KLd^2 + KL^2d)$ | $\mathcal{O}(KLd + Kd^2)$ | $\mathcal{O}(L)$ | $\mathcal{O}\left(\frac{C_{\max}}{KLd+Kd^2}\right)$ |
| DiffuMamba | $\mathcal{O}(KLd^2 + KLdA)$ | $\mathcal{O}(KLd + Kd^2 + KLA)$ | $\mathcal{O}\left(\frac{dL}{d+L}\right)$ | $\mathcal{O}\left(\frac{L}{KLd+Kd^2}\right)$ |
| DiffuMamba-H | $\mathcal{O}(KLd^2 + KL^2d)$ | $\mathcal{O}(KLd + Kd^2)$ | $\mathcal{O}(L)$ | $\mathcal{O}\left(\frac{C_{\max}}{KLd+Kd^2}\right)$ |
| DiffuMamba-H + Fast-dLLM | $\mathcal{O}\left(KGd^2 + KGLd + \frac{L}{G}(Ld^2 + L^2d)\right)^{\dagger}$ | $\mathcal{O}(KLd + Kd^2)$ | $\mathcal{O}\left(\frac{pL}{G}\right)$ | $\mathcal{O}\left(\frac{C_{\max}}{KLd+Kd^2}\right)$ |

We also report downstream eval results for 1.3B models in Table 4. As expected, the absolute scores remain modest at this scale, reflecting the difficulty of these reasoning and knowledge-intensive for base models at 1.3B scale. Nonetheless, a consistent trend emerges across all tasks: DiffuMamba and DiffuMamba-H clearly outperform DiffuTran by $\approx 4\%$ on average, indicating that linear-time state space modeling provides a stronger denoising backbone for diffusion-based LMs. Further hybridizing with attention layers, as in DiffuMamba-H produces the strongest results, suggesting that a small degree of explicit cross-token interaction complements the Mamba backbone. Even at low model capacity where downstream metrics are weak, Mamba based DLM clearly outperform attention based DiffuTran, demonstrating the effectiveness of state space driven denoising and its superior inference throughput.

**Takeaway.** From 240M to 1.3B models Mamba-based DLMs match or improve upon MHA-based ones in validation and zero-shot PPL, with hybrid DiffuMamba-H yielding the most consistent gains at larger scale.

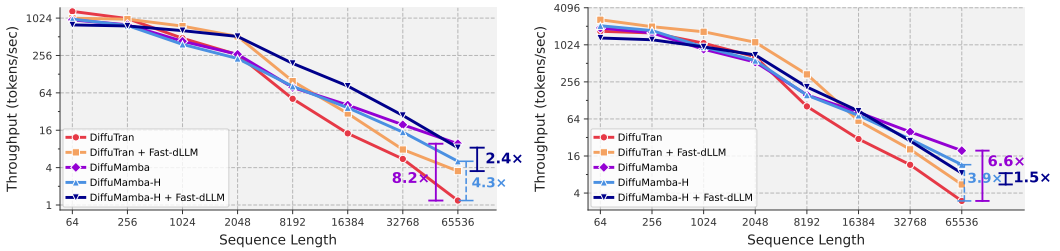
Table 4: Zero-shot downstream accuracy (↑) on reasoning and commonsense benchmarks for 1.3B models. At this scale, overall performance is modest, as 1.3B models struggle on these challenging benchmarks. Both DiffuMamba and DiffuMamba-H consistently outperform the attention-based diffusion baseline (DiffuTran), with DiffuMamba-H achieving the strongest results. Best results are shown in blue; second-best are underlined.

| Config (1.3B) | OBQA | HS | PIQA | LOQA | ARC-C | Avg. |
|---|---|---|---|---|---|---|
| DiffuTran | 26.61 | 34.97 | 60.64 | 21.86 | 24.98 | 33.81 |
| DiffuMamba | 32.12 | 37.74 | 62.56 | 29.87 | 28.33 | 38.12 |
| DiffuMamba-H | 33.87 | 38.02 | 59.13 | 32.35 | 27.83 | 38.24 |

### 4.3 Inference Throughput

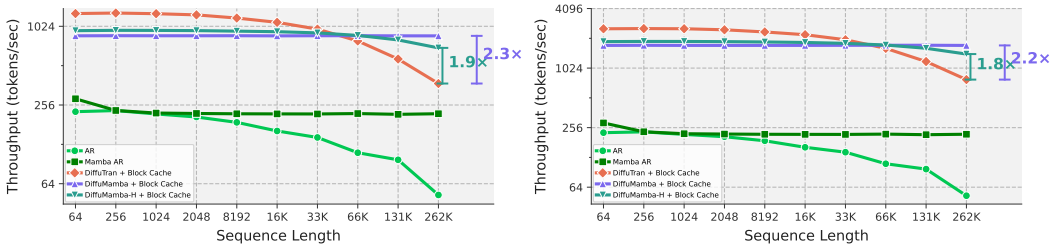
Figures 2a and 2b compare inference throughput between the attention-based diffusion model, DiffuTran and our proposed DiffuMamba and DiffuMamba-H as sequence length ($L$) increases from 64 to 65536. Following Nie et al. (2025); Wu et al. (2025), we fix batch size $B = 1$. We also evaluate Fast-dLLM (Wu et al., 2025), which is a training free block diffusion decoding paradigm that relies on KV-cache for inter-block diffusion and *recomputes* KV-cache after each generation block. Throughput ($T$) is measured in tokens per second as the number of generated tokens divided by total wall-clock decoding time. Unlike prior works that use a fixed number of denoising steps (e.g., $K = 128$ in Nie et al. (2025)), we scale the number of steps with the sequence length as $K = L/p$ to enable long-context evaluation, and report results for $p \in \{8, 16\}$. For Fast-dLLM we fix the block size to $G = 32$ following Wu et al. (2025), yielding denoising steps per block, $k = 32/p$.

It can be observed that for moderate sequence lengths ($L \leq$ 2K), DiffuTran+Fast-dLLM achieves the highest throughput among all models. In this regime, DiffuMamba and DiffuTran exhibit competitive and nearly identical performance. As the sequence length increases, throughput gradually decreases for all methods, consistent with the asymptotic analysis in Table 3. In particular,




(a) Throughput at $K = L/8$ denoising steps.

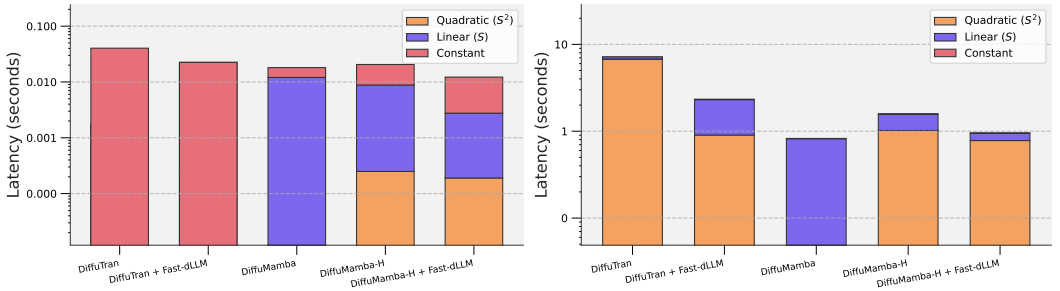
(b) Throughput at $K = L/16$ denoising steps.




(c) Throughput at $K = L/8$ denoising steps.

(d) Throughput at $K = L/16$ denoising steps.

Figure 2: Inference throughput for 1.3B models vs. sequence length ($L$) at batch size 1 with $L/p$ decoding steps ($p \in 8, 16$). In (a,b), `DiffuMamba` and `DiffuMamba-H` achieve 8.2× and 4.3× higher throughput than `DiffuTran`, respectively; `DiffuMamba-H+Fast-dLLM` (block size 32) further improves long-sequence throughput, yielding 2.4× gains over `DiffuTran+Fast-dLLM` (Wu et al., 2025). In (c,d), `DiffuMamba` and `DiffuMamba-H` attain 2.3× and 1.9× improvements over `DiffuTran` with simple block caching (Wang et al., 2025b), which also consistently outperforms AR baselines.




(a) Latency Components at $L = 1024$

(b) Latency Components at $L = 65K$

Figure 3: **Latency decomposition per denoising step** from a single forward pass, obtained by fitting a quadratic model $t(L) = aL^2 + bL + c$ to latency curves. The quadratic term $a$ captures attention-like interactions, the linear term $b$ reflects token-wise computation (e.g., MLP/SSM), and $c$ denotes constant overhead. For intermediate sequence lengths, linear and constant terms dominate across models; at large lengths, `DiffuMamba` benefits from a negligible quadratic component, while `DiffuTran` becomes quadratic-dominated. `DiffuMamba-H` reduces latency relative to `DiffuTran` but remains governed by attention scaling.

throughput becomes dominated by the hidden dimension $d$ rather than by $L$, staying approximately constant in this regime.

Beyond $2K$ tokens in Figures 2a and 2b, the throughput of `DiffuTran` degrades sharply, while `DiffuMamba` (memory bandwidth bound (Baruah et al., 2025)) and `DiffuMamba-H` (reduces FLOPs per forward) experience a substantially slower decline. This trend aligns with the asymptotic throughput analysis in Table 3, which predicts more favorable scaling behavior for `DiffuMamba` and `DiffuMamba-H`. When $L > d$ (1920) (approximately at $L = 2048$), the diffusion model enters a compute-saturated regime. In this regime, the FLOPs per token grow with $L$, while the sustained FLOPs/s remain bounded by the peak hardware capacity $C_{\max}$, resulting in a decline in throughput. Beyond this point, `DiffuTran` exhibits the steepest throughput collapse due to

the combined effects of quadratic attention cost and linearly increasing denoising steps, resulting in $T = \mathcal{O}(\frac{C_{max}}{KLd+Kd^2}) \approx \mathcal{O}(\frac{1}{L^2})$, where $K = L/p$, scaling at long context lengths. In contrast, `DiffuMamba` is memory bound. As a result, for $L > d$, throughput scales as $\mathcal{O}(\frac{L}{Kd^2+KLd})$, where $K = L/p$, thus $T = \mathcal{O}(\frac{p}{d^2+Ld}) = O(\frac{1}{L})$. This gives slower throughput degradation, retaining a $\mathbf{8.2}\times$ and $\mathbf{4.3}\times$ advantage at $K = L/16$ denoising steps for `DiffuMamba` and `DiffuMamba-H` respectively over `DiffuTran` at 65K tokens.

In Figures 2c and 2d, we evaluate all DLMs using a block-autoregressive inference that reuses cached representationsacross successive generation blocks (Wang et al., 2025b; Arriola et al., 2025). Concretely, once a block of tokens is denoised, the corresponding cache is retained and directly reused when processing the next block, avoiding repeated forward passes over previously generated tokens. As generation proceeds, caches accumulate incrementally across blocks, enabling efficient long-context inference. Under this setting, all DLMs recover a clear advantage over autoregressive baselines. Further, at long generation length ($L = 260$K), `DiffuMamba` achieves a **2.3**$\times$ throughput improvement and `DiffuMamba-H` a **1.9**$\times$ improvement over `DiffuTran`. This inference paradigm is particularly well suited to Mamba-based diffusion: *state updates can be constructed locally within each block and reused across blocks in an autoregressive fashion.* In our bidirectional Mamba design, right-to-left state updates are restricted to operate only within the current block, while cached states are propagated forward across blocks. These results highlight that eliminating cache recomputation is key to outperforming autoregressive baselines at long contexts.

To further understand the throughput trends in Figure 2, we analyze the *per-forward-pass latency* as a function of sequence length $L$. From each benchmark record, we compute the mean latency per denoising step as $t(L) = \frac{\texttt{Model Total Time}}{\texttt{steps}}$ which isolates the cost of a single model forward pass at sequence length $L$. We model the measured latency using a non-negative quadratic decomposition, $t(L) = aL^2 + bL + c$ s.t. $a, b, c \geq 0$, where the quadratic term $aL^2$ captures attention-like pairwise token interactions, the linear term $bL$ captures token-wise computations (e.g., MLP/SSM/projections), and the constant term $c$ captures sequence-length–independent overheads.

The fitted equations make the scaling behavior explicit as demonstrated in Figure 3. Attention-based diffusion (`DiffuTran`) has a comparatively larger quadratic coefficient, so the quadratic component $aL^2$ becomes dominant as $L$ grows. In contrast, Mamba-based diffusion (`DiffuMamba`) exhibits a much smaller quadratic coefficient; over the evaluated range, latency is largely explained by the linear and constant components. Hybrid interleavings (`DiffuMamba-H` variants) sit between these regimes, retaining a controlled quadratic contribution due to periodic attention insertion.

Taken together, this latency decomposition clarifies the fundamental source of the throughput advantages observed in Figure 2. By isolating the role of quadratic attention costs, our analysis explains why Mamba-based diffusion exhibits more favorable scaling and why hybrid designs interpolate smoothly between the two regimes. These insights motivate the architectural choices we summarize next and provide a principled foundation for the conclusions that follow.

**Takeaway** Mamba-based DLMs achieve higher throughput than MHA-based ones under all inference algorithms. Block caching removes quadratic recomputation in Transformers pushing DLMs' throughput far beyond AR. Mamba-based block-AR DLMs with constant memory requirement achieve best throughput in this memory-bound setting.

## 5 Conclusion

We demonstrate the feasibility of Mamba-based DLMs by introducing `DiffuMamba`, the first diffusion LM built solely on linear state-space mixers, and `DiffuMamba-H`, a hybrid that interleaves Mamba-2 and attention to capture global context. Across 240M–1.3B parameters, both models match or outperform the Transformer-based baseline `DiffuTran`.

Our throughput analysis demonstrates that Mamba-based DLMs scale more favorably than MHA-based DLMs across a range of inference algorithms. In particular, DLMs relying on block cache reuse yield the strongest performance across the entire context length spectrum efficiently combining the multi-token generation advantages of DLMs and preventing quadratic scaling by fixing the representations of the past tokens (e.g. using KV-cache). While `DiffuMamba + Block Cache`

emerges as the most efficient design for long sequence generation, achieving optimal performance in this regime requires training with block-diffusion inductive biases or incorporating distillation objectives as in Wang et al. (2025b). As the primary goal of this work is to establish feasibility and characterize the efficiency advantages of DLMs with linear mixers, we leave the training of block-cached hybrid models at useful scales and the exploration of alternative linear mixers to future work. We summarize the main limitations of this work in Appendix C.

## 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.

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.

Sangmin Bae, Bilge Acun, Haroun Habeeb, Seungyeon Kim, Chien-Yu Lin, Liang Luo, Junjie Wang, and Carole-Jean Wu. Hybrid architectures for language models: Systematic analysis and design insights. *arXiv preprint arXiv:2510.04800*, 2025.

Dzmitry Bahdanau, Kyunghyun Cho, and Yoshua Bengio. Neural machine translation by jointly learning to align and translate. *arXiv preprint arXiv:1409.0473*, 2014.

Trinayan Baruah, Kaustubh Shivdikar, Sara Prescott, and David Kaeli. Characterizing the behavior of training mamba-based state space models on gpus. *arXiv preprint arXiv:2508.17679*, 2025.

Tom Brown, Benjamin Mann, Nick Ryder, Melanie Subbiah, Jared D Kaplan, Prafulla Dhariwal, Arvind Neelakantan, Pranav Shyam, Girish Sastry, Amanda Askell, et al. Language models are few-shot learners. *Advances in neural information processing systems*, 33:1877–1901, 2020.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, T. Brants, Phillip Todd Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling. In *Interspeech*, 2013.

Jiaao Chen, Aston Zhang, Mu Li, Alex Smola, and Diyi Yang. A cheaper and better diffusion language model with soft-masked noise. In *Proceedings of the 2023 Conference on Empirical Methods in Natural Language Processing*, 2023.

Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm: Scaling language modeling with pathways. *Journal of Machine Learning Research*, 2023.

Arman Cohan, Franck Dernoncourt, Doo Soon Kim, Trung Bui, Seokhwan Kim, Walter Chang, and Nazli Goharian. A discourse-aware attention model for abstractive summarization of long documents. In *Proceedings of NAACL-HLT*, pp. 615–624, 2018.

Trung Dinh Quoc Dang, Huy Hoang Nguyen, and Aleksei Tiulpin. Log-vmamba: local-global vision mamba for medical image segmentation. In *Proceedings of the Asian Conference on Computer Vision*, 2024.

Tri Dao and Albert Gu. Transformers are ssms: Generalized models and efficient algorithms through structured state space duality. *arXiv preprint arXiv:2405.21060*, 2024.

Soham De, Samuel L Smith, Anushan Fernando, Aleksandar Botev, George Cristian-Muraru, Albert Gu, Ruba Haroun, Leonard Berrada, Yutian Chen, Srivatsan Srinivasan, et al. Griffin: Mixing gated linear recurrences with local attention for efficient language models. *arXiv preprint arXiv:2402.19427*, 2024.

Alex Ergasti, Filippo Botti, Tomaso Fontanini, Claudio Ferrari, Massimo Bertozzi, and Andrea Prati. U-shape mamba: State space model for faster diffusion. In *Proceedings of the Computer Vision and Pattern Recognition Conference*, 2025.

Nima Fathi, Torsten Scholak, and Pierre-André Noël. Unifying autoregressive and diffusion-based sequence generation. *COLM*, 2025.

Daniel Y Fu, Tri Dao, Khaled K Saab, Armin W Thomas, Atri Rudra, and Christopher Ré. Hungry hungry hippos: Towards language modeling with state space models. *ICLR*, 2023.

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. *ICLR*, 2025.

Albert Gu and Tri Dao Goel. Mamba: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. In *ICLR*, 2022.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics*, 2023.

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

Jordan Hoffmann, Sebastian Borgeaud, Arthur Mensch, Elena Buchatskaya, Trevor Cai, Eliza Rutherford, Diego de Las Casas, Lisa Anne Hendricks, Johannes Welbl, Aidan Clark, et al. Training compute-optimal large language models. *arXiv preprint arXiv:2203.15556*, 2022.

Vincent Tao Hu, Stefan Andreas Baumann, Ming Gui, Olga Grebenkova, Pingchuan Ma, Johannes S. Fischer, and Bjorn Ommer. Zigma: A dit-style zigzag mamba diffusion model. In *European Conference on Computer Vision*, 2024.

Daniel Israel, Guy Van den Broeck, and Aditya Grover. Accelerating diffusion llms via adaptive parallel decoding. *Advances in neural information processing systems*, 2025.

Zhihan Ju and Wanting Zhou. Vm-ddpm: Vision mamba diffusion for medical image synthesis. *arXiv preprint arXiv:2405.05667*, 2024.

Jeffrey Li, Alex Fang, Georgios Smyrnis, Maor Ivgi, Matt Jordan, Samir Yitzhak Gadre, Hritik Bansal, Etash Guha, Sedrick Scott Keh, Kushal Arora, et al. Datacomp-lm: In search of the next generation of training sets for language models. *Advances in Neural Information Processing Systems*, 37:14200–14282, 2024.

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

Opher Lieber, Barak Lenz, Hofit Bata, Gal Cohen, Jhonathan Osin, Itay Dalmedigos, Erez Safahi, Shaked Meirom, Yonatan Belinkov, Shai Shalev-Shwartz, et al. Jamba: A hybrid transformer-mamba language model. *arXiv preprint arXiv:2403.19887*, 2024.

Xiao Liu, Chenxu Zhang, Fuxiang Huang, Shuyin Xia, Guoyin Wang, and Lei Zhang. Vision mamba: A comprehensive survey and taxonomy. *IEEE Transactions on Neural Networks and Learning Systems*, 2025a.

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 preprint arXiv:2506.06295*, 2025b.

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. In *International Conference on Machine Learning*, 2023.

Xinyin Ma, Runpeng Yu, Gongfan Fang, and Xinchao Wang. dkv-cache: The cache for diffusion language models. *arXiv preprint arXiv:2505.15781*, 2025.

Mitchell P. Marcus, Mary Ann Marcinkiewicz, and Beatrice Santorini. Building a large annotated corpus of english: The penn treebank. *Computational linguistics*, 19(2):313–330, 1993.

Stephen Merity, Caiming Xiong, James Bradbury, and Richard Socher. Pointer sentinel mixture models. In *Proceedings of the International Conference on Learning Representations (ICLR)*, 2017.

Shentong Mo. Scaling diffusion mamba with bidirectional ssms for efficient 3d shape generation. In *Proceedings of the Thirty-Ninth AAAI Conference on Artificial Intelligence and Thirty-Seventh Conference on Innovative Applications of Artificial Intelligence and Fifteenth Symposium on Educational Advances in Artificial Intelligence*, 2025.

Quan Nguyen-Tri, Mukul Ranjan, and Zhiqiang Shen. Attention is all you need for kv cache in diffusion llms. *arXiv preprint*, 2025.

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*, 2025a.

Jinjie Ni, Qian Liu, Chao Du, Longxu Dou, Hang Yan, Zili Wang, Tianyu Pang, and Michael Qizhe Shieh. Training optimal large diffusion language models. *arXiv preprint arXiv:2510.03280*, 2025b.

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.

Oleksiy Ostapenko, Luke Kumar, Raymond Li, Denis Kocetkov, Joel Lamy-Poirier, Shruthan Radhakrishna, Soham Parikh, Shambhavi Mishra, Sebastien Paquet, Srinivas Sunkara, et al. Apriel-h1: Towards efficient enterprise reasoning models. *arXiv preprint arXiv:2511.02651*, 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.

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Nghia The Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernández. The lambada dataset: Word prediction requiring a broad discourse context. In *Proceedings of ACL*, 2016.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *IEEE/CVF International Conference on Computer Vision, ICCV*, 2023.

Hao Phung, Quan Dao, Trung Dao, Hoang Phan, Dimitris Metaxas, and Anh Tran. Dimsum: Diffusion mamba - a scalable and unified spatial-frequency method for image generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.

Michael Poli, Stefano Massaroli, Eric Nguyen, Daniel Y Fu, Tri Dao, Stephen Baccus, Yoshua Bengio, Stefano Ermon, and Christopher Ré. Hyena hierarchy: Towards larger convolutional language models. In *International Conference on Machine Learning*, pp. 28043–28078. PMLR, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 2019.

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

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin T. Chiu, and Volodymyr Kuleshov. The diffusion duality. In *Proceedings of the 42nd International Conference on Machine Learning (ICML)*, 2025.

Yair Schiff, Chia-Hsiang Kao, Aaron Gokaslan, Tri Dao, Albert Gu, and Volodymyr Kuleshov. Caduceus: Bi-directional equivariant long-range dna sequence modeling. *Proceedings of machine learning research*, 235:43632, 2024.

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.

Shriyank Somvanshi, Md Monzurul Islam, Mahmuda Sultana Mimi, Sazzad Bin Bashar Polock, Gaurab Chhetri, and Subasish Das. From s4 to mamba: A comprehensive survey on structured state space models. In *arXiv:2503.18970*, 2025.

Yang Song and Stefano Ermon. Score-based generative modeling through stochastic differential equations. In *ICLR*, 2021.

Yi Tay, Mostafa Dehghani, Samira Abnar, Yikang Shen, Dara Bahri, Philip Pham, Jinfeng Rao, Liu Yang, Sebastian Ruder, and Donald Metzler. Long range arena: A benchmark for efficient transformers. In *ICLR*, 2020.

Yao Teng, Yue Wu, Han Shi, Xuefei Ning, Guohao Dai, Yu Wang, Zhenguo Li, and Xihui Liu. Dim: Diffusion mamba for efficient high-resolution image synthesis. *arXiv preprint arXiv:2405.14224*, 2024.

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.

Harshit Varma, Dheeraj Nagaraj, and Karthikeyan Shanmugam. Glauber generative model: Discrete diffusion models via binary classification. *ICLR*, 2025.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Dustin Wang, Rui-Jie Zhu, Steven Abreu, Yong Shan, Taylor Kergan, Yuqi Pan, Yuhong Chou, Zheng Li, Ge Zhang, Wenhao Huang, et al. A systematic analysis of hybrid linear attention. *arXiv preprint arXiv:2507.06457*, 2025a.

Jinhong Wang, Jintai Chen, Danny Chen, and Jian Wu. Lkm-unet: Large kernel vision mamba unet for medical image segmentation. In Marius George Linguraru, Qi Dou, Aasa Feragen, Stamatia Giannarou, Ben Glocker, Karim Lekadir, and Julia A. Schnabel (eds.), *Medical Image Computing and Computer Assisted Intervention – MICCAI 2024*, 2024.

Xu Wang, Chenkai Xu, Yijie Jin, Jiachun Jin, Hao Zhang, and Zhijie Deng. Diffusion llms can do faster-than-ar inference via discrete diffusion forcing. *arXiv preprint arXiv:2508.09192*, 2025b.

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*, 2025.

Rui Xu, Shu Yang, Yihui Wang, Yu Cai, Bo Du, and Hao Chen. Visual mamba: A survey and new outlooks. *arXiv preprint arXiv:2404.18861*, 2024.

Songlin Yang, Jan Kautz, and Ali Hatamizadeh. Gated delta networks: Improving mamba2 with delta rule. *arXiv preprint arXiv:2412.06464*, 2024.

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.

Xiang Zhang, Junbo Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *Advances in Neural Information Processing Systems (NeurIPS)*, 2015.

Zixuan Zhou, Xuefei Ning, Ke Hong, Tianyu Fu, Jiaming Xu, Shiyao Li, Yuming Lou, Luning Wang, Zhihang Yuan, Xiuhong Li, et al. A survey on efficient inference for large language models. *arXiv preprint arXiv:2404.14294*, 2024.

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

Lianghui Zhu, Bencheng Liao, Qian Zhang, Xinlong Wang, Wenyu Liu, and Xinggang Wang. Vision mamba: efficient visual representation learning with bidirectional state space model. In *ICML*, 2024.

Jingwei Zuo, Maksim Velikanov, Ilyas Chahed, Younes Belkada, Dhia Eddine Rhayem, Guillaume Kunsch, Hakim Hacid, Hamza Yous, Brahim Farhat, Ibrahim Khadraoui, et al. Falcon-h1: A family of hybrid-head language models redefining efficiency and performance. *arXiv preprint arXiv:2507.22448*, 2025.

## A Discrete Diffusion Modeling Preliminaries

Masked Diffusion Models (MDMs), employ a forward noising process where tokens are progressively replaced by a special [MASK] token (Sahoo et al., 2024; Shi et al., 2024). This process is defined by the transition probability

$$q_{t|0}(\mathbf{x}_t \mid \mathbf{x}_0) = \prod_{i=1}^{L} q_{t|0}(x_t^i \mid x_0^i)$$
$$= \prod_{i=1}^{L} \mathrm{Cat}\Big(x_t^i;\ (1-t)\delta_{x_0^i} + t\,\delta_{\text{MASK}}\Big), \tag{4}$$

where $t \in [0, 1]$ controls interpolation between the original data $\mathbf{x}_0$ (at $t = 0$) and a fully masked sequence (at $t = 1$). $\mathrm{Cat}(\cdot)$ denotes the categorical distribution. A parametric model $p_\theta$ learns the reverse denoising process, and generation starts from all `[MASK]` and iteratively unmasks by sampling $p_\theta(x_0^i|\mathbf{x}_t)$.

Recent theory (MDM (Shi et al., 2024; Sahoo et al., 2024), RADD (Ou et al., 2024)) simplifies training from a variational bound to a reweighted cross-entropy over masked positions

$$\mathcal{L}_{\text{MDM}} = \int_0^1 \frac{1}{t} \mathbb{E}_{q_{t|0}(\mathbf{x}_t|\mathbf{x}_0)} \left[ \sum_{i:x_t^i=\text{MASK}} -\log p_\theta(x_0^i|\mathbf{x}_t) \right] dt. \tag{5}$$

This formulation scales to LLMs as DLMs, with LLaDA (Nie et al., 2025) and Dream-7B (Ye et al., 2025) matching autoregressive performance while enabling parallel decoding and flexible infilling.

The analytical reverse of the forward process defined in Equation 4 is computationally inefficient for generation, as it typically modifies only a single token at each step (Campbell et al., 2022; Lou et al., 2023). A common strategy to accelerate sampling is to employ an *MCMC-based approximation of the reverse process* (Nie et al., 2025), enabling the model to update multiple masked tokens in a single step when transitioning from a noise level $t$ to an earlier level $s < t$. This yields the following factorized form:

$$q_{s|t} = \prod_{i=1}^{L} q_{s|t}(\boldsymbol{x}_s^i \mid \boldsymbol{x}_t).$$

$$= \begin{cases} 1 & \text{if } \boldsymbol{x}_s^i = \boldsymbol{x}_t^i \neq [\text{MASK}], \\ \dfrac{s}{t} & \text{if } \boldsymbol{x}_s^i = \boldsymbol{x}_t^i = [\text{MASK}], \\ \dfrac{t-s}{t}\, q_{0|t}(\boldsymbol{x}_s^i \mid \boldsymbol{x}_t) & \text{if } \boldsymbol{x}_t^i = [\text{MASK}] \neq \boldsymbol{x}_s^i. \end{cases} \tag{6}$$

Here, $q_{0|t}(\boldsymbol{x}_s^i \mid \boldsymbol{x}_t)$ denotes the model-provided distribution over the vocabulary for predicting a non-[MASK] token when $\boldsymbol{x}_t^i$ is masked. In conditional generation settings e.g., generating a response $\boldsymbol{x}_0$ given a prompt $p$, the reverse diffusion process in Equation 6 must be adapted. In this case, the model's predictive distribution for unmasking a token becomes *prompt-conditioned*: $q_{0|t}(\boldsymbol{x}_s^i \mid \boldsymbol{x}_t, p)$, reflecting that token predictions now depend on both the intermediate noised sequence and the conditioning prompt.

## B Model Configurations

Model configurations for `DiffuTran`, `DiffuMamba` and `DiffuMamba-H` in our experiments. All variants are trained under the same diffusion objective, noise schedule, and tokenization. The only difference lies in the internal mixer architecture. The MLP expansion ratio in `DiffuMamba` and `DiffuMamba-H` configurations is reduced by half for comparable parameter budgets. In the hybrid architecture, an attention layer is inserted every $N$ layers; we set $N = 5$ in all experiments. In Table 6, we list down all the training and inference hyperparamters.

Table 5: Model configurations for `DiffuTran`, `DiffuMamba` and `DiffuMamba-H`.

| Model | Params (B) | Layers | $d_{model}$ | $d_{mlp}$ | $d_{head}$ | $d_{state}$ | Context Len. | # Tokens |
|---|---|---|---|---|---|---|---|---|
| `DiffuTran`-0.24B | 0.24 | 24 | 960 | 960*4 | 32 | - | 1024 | 25B |
| `DiffuMamba`-0.29B | 0.29 | 24 | 960 | 960*2 | 32 | 128 | 1024 | 25B |
| `DiffuMamba-H`-0.26B (K=5) | 0.26 | 24 | 960 | 960*2 | 32 | 128 | 1024 | 25B |
| `DiffuTran`-0.5B | 0.52 | 36 | 960 | 960*4 | 32 | - | 1024 | 50B |
| `DiffuMamba`-0.6B | 0.67 | 36 | 960 | 960*2 | 32 | 128 | 1024 | 50B |
| `DiffuMamba-H`-0.6B (K=5) | 0.61 | 36 | 960 | 960*2 | 32 | 128 | 1024 | 50B |
| `DiffuTran`-1.3B | 1.3 | 24 | 1920 | 1920*4 | 32 | - | 1024 | 120B |
| `DiffuMamba`-1.6B | 1.6 | 24 | 1920 | 1920*2 | 32 | 128 | 1024 | 120B |
| `DiffuMamba-H`-1.5B (K=5) | 1.5 | 24 | 1920 | 1920*2 | 32 | 128 | 1024 | 120B |

### B.1 Architecture Details of DiffuMamba

We present here the architectural details of `DiffuMamba` an MDM whose denoiser is built on a *bidirectional state-space Mamba (BiMamba)* backbone. DiffuMamba preserves the probabilistic structure of masked discrete diffusion while replacing the Transformer encoder with a bidirectional Mamba mixer, enabling *linear-time* inference and substantially lower memory overhead during multi-step denoising.

Let $\mathbf{x} \in \mathbb{R}^{B\times L\times d}$ denote a batch of token embeddings, where $B$ is the batch size, $L$ the sequence length, and $d$ the hidden dimension. Each block is composed of two independent Mamba layers: one processes the sequence in the forward direction, and the other processes the sequence in the reverse direction

$$\mathbf{h}_i^{\rightarrow} = A_f * \mathbf{h}_{i-1}^{\rightarrow} + B_f * \mathbf{x}_i, \quad i = 1, \ldots, L, \tag{7}$$

$$\mathbf{h}_i^{\leftarrow} = A_b * \mathbf{h}_{i+1}^{\leftarrow} + B_b * \mathbf{x}_i, \quad i = L, \ldots, 1. \tag{8}$$

Here $A_f, B_f, A_b, B_b \in \mathbb{R}^{k\times d}$ are learnable state-transition kernels implemented as 1D causal and anti-causal convolutions or efficient scan operations (Gu & Goel, 2023). The two directional feature streams are fused through simple *additive integration*

$$\text{Mamba}(x_i) = \mathbf{h}_i = \mathbf{h}_i^{\rightarrow} + \mathbf{h}_i^{\leftarrow}, \tag{9}$$

providing a symmetric context representation while maintaining numerical stability. The resulting hidden sequence $\mathbf{H} = (\mathbf{h}_1, \ldots, \mathbf{h}_L)$ is then normalized and passed through a lightweight feed-forward projection before residual addition.

Each Diffusion Block employs timestep-conditioned adaptive layer normalization (AdaLN) to inject the diffusion noise level into the hidden activations. A small MLP maps the scalar timestep $t$ to a continuous embedding $\tau_t = \text{MLP}(t) \in \mathbb{R}^{d_c}$, which modulates both the mixer and MLP sublayers:

$$\text{AdaLN}(\mathbf{x}; \tau_t) = \gamma_t \cdot \frac{\mathbf{x} - \mu(\mathbf{x})}{\sigma(\mathbf{x})} + \beta_t, \quad (\gamma_t, \beta_t) = W_{\text{cond}}\tau_t. \tag{10}$$

This conditioning allows the denoiser to adapt its internal recurrence dynamics to varying noise levels across diffusion steps.

Each block then applies Mamba mixing followed by an MLP refinement as shown in Figure 1:

$$\mathbf{y}_{\text{mixer}} = \text{Mamba}(\text{AdaLN}(\mathbf{x}; \tau_t)) + \mathbf{x}, \tag{11}$$

$$\mathbf{y}_{\text{mlp}} = \text{MLP}(\text{AdaLN}(\mathbf{y}_{\text{mixer}}; \tau_t)) + \mathbf{y}_{\text{mixer}}. \tag{12}$$

The final output $\mathbf{y}_{\text{mlp}}$ is passed to the next block (B) in the stack. During denoising, given masked input embeddings $\mathbf{x}_t$ and timestep embedding $\tau_t$, the model predicts token logits via the full diffusion stack:

$$\mathbf{z} = f_\theta(\mathbf{x}_t, \tau_t) = \text{OutputProj}(\text{B}_N(\cdots \text{B}_1(E(\mathbf{x}_t), \tau_t) \cdots)). \tag{13}$$

The conditional distribution over clean tokens is

$$p_\theta(\mathbf{x}_0|\mathbf{x}_t, t) = \prod_{i=1}^{L} \text{Cat}(x_0^i; \text{softmax}(z_i)). \tag{14}$$

Training minimizes the masked diffusion objective (Eq. 5), preserving probabilistic consistency with absorbing-state discrete diffusion.

Table 6: Key training and inference hyperparameters used across all experiments unless stated otherwise.

| **Category** | **Setting** |
|---|---|
| Backbone | Transformer (`DiffuTran`), Mamba (`DiffuMamba`), Hybrid (`DiffuMamba-H`) |
| Diffusion Type | Absorbing-state masked diffusion |
| Parameterization | Substitution (subs) |
| Noise Conditioning | Log Linear |
| Global Batch Size | 512 |
| Precision | bf16 |
| EMA Decay | 0.9999 |
| Antithetic Sampling | Enabled |
| Sampling $\epsilon$ | $10^{-3}$ |
| Gradient Clipping | 1.0 |
| Optimizer | Adam |
| Max Learning Rate | $1 \times 10^{-4}$ |
| Min Learning Rate | $1 \times 10^{-6}$ |
| Weight Decay | 0.1 |
| Adam Betas | (0.9, 0.95) |
| Adam $\epsilon$ | $10^{-8}$ |
| LR scheduler | Cosine |
| Denoising Steps factor (p) | $\{8, 16\}$ |
| Block Length | 32 |
| Inference Batch Size | 1 |
| hidden dimension | 960 |
| # blocks | 24 |
| # heads | 24 |
| bidirectional weight tie | False |
| mamba state dimension | 128 |
| mamba conv dimension | 4 |
| expansion factor | 2 |
| interleaving attention (N) | 5 |

## C LIMITATIONS

While our results demonstrate strong inference gains, the joint training of diffusion language models with Mamba backbones and block cache reuse remains unexplored. Our evaluation therefore isolates the effect of backbone choice and inference algorithm on throughput and latency, but does not reflect potential gains from training models explicitly for the fastest decoding regimes. In particular, training models explicitly optimized for the fastest inference algorithms such as block diffusion with cache reuse alongside the exploration of alternative linear-time token mixers, remains a natural and immediate next step.

Further our throughput measurements assume fixed acceptance rates through the parameter $p$, which controls the number of denoising steps $K = L/p$. Although diffusion models achieve their best quality when $p = 1$, this setting effectively reduces diffusion decoding to autoregressive-style computation and defeats the primary motivation of diffusion-based generation. Our choice of moderate $p$ values reflects the intended efficiency regime of DLMs, but alternative acceptance behaviors may lead to different trade-offs.

Finally, all experiments are conducted at batch size $B = 1$. This setting is consistent with prior work (Nie et al., 2025; Wu et al., 2025) on long-context decoding and reflects the practical constraints of GPU memory at extreme sequence lengths. Evaluating larger batch sizes in the long-context regime remains an important direction for future study.

**Algorithm 1** LT-MELBO: Latent-Token Marginal ELBO Estimator (Self-Conditioned Variational Tightening)

**Require:** Completion tokens $y \in \mathcal{V}^L$, prompt/context $c$ (kept unmasked), mask token id $m$, embedding matrix $E \in \mathbb{R}^{|\mathcal{V}| \times d}$
**Require:** Masking distribution $\pi_{t|0}$: each completion token masked i.i.d. with probability $t$
**Require:** Model $p_\theta(\cdot \mid \cdot)$ returning logits for all positions, and supporting `inputs_embeds`
**Require:** Temperature $\tau > 0$, KL weight $\lambda \geq 0$, number of time samples $S \geq 1$
**Ensure:** Estimated log-likelihood proxy $\widehat{\mathcal{L}}_{\mathrm{LT}}(y \mid c)$

$\widehat{\mathcal{L}} \leftarrow 0$
**for** $s = 1$ to $S$ **do**
  Sample $t \sim \mathrm{Unif}(0, 1)$
  Sample corrupted sequence $y_t \sim \pi_{t|0}(\cdot \mid y)$ ▷ i.i.d. masking on completion only
  Let $M \leftarrow \{i : (y_t)_i = m\}$ be masked positions
  ▷ **Pass 1 (latent-token proposal)**
  Compute logits $\ell^{(0)} \leftarrow f_\theta(c, y_t)$
  Define factorized proposal on masked positions:

$$q_\theta^{(0)}(z_i = v \mid c, y_t) \;\propto\; \exp\Big(\ell_{i,v}^{(0)}/\tau\Big), \quad i \in M$$

  Construct *soft latent token embeddings* for $i \in M$:

$$e_i^{(0)} \;\leftarrow\; \sum_{v \in \mathcal{V}} q_\theta^{(0)}(z_i = v \mid c, y_t)\, E[v]$$

  Build embedded input $\tilde{x}$ by replacing masked token embeddings with $e_i^{(0)}$ and keeping unmasked token embeddings unchanged
  ▷ **Pass 2 (self-conditioned scoring)**
  Compute logits $\ell^{(1)} \leftarrow f_\theta(\tilde{x})$ ▷ via `inputs_embeds`
  Define self-conditioned predictive distribution on masked positions:

$$p_\theta^{(1)}(y_i = v \mid c, \tilde{x}) \;\propto\; \exp\Big(\ell_{i,v}^{(1)}\Big), \quad i \in M$$

  Compute the *self-conditioned masked log-score*:

$$Z_\theta^{\mathrm{SC}}(t, y_t) \;\leftarrow\; \frac{1}{t} \sum_{i \in M} \log p_\theta^{(1)}(y_i \mid c, \tilde{x})$$

  Compute the *latent-consistency penalty*:

$$R_\theta(t, y_t) \;\leftarrow\; \frac{1}{t} \sum_{i \in M} \mathrm{KL}\Big(q_\theta^{(0)}(z_i \mid c, y_t) \,\|\, p_\theta^{(1)}(\cdot \mid c, \tilde{x})\Big)$$

  Update estimate:

$$\widehat{\mathcal{L}} \;\leftarrow\; \widehat{\mathcal{L}} + \big(Z_\theta^{\mathrm{SC}}(t, y_t) - \lambda\, R_\theta(t, y_t)\big)$$

**end for**
**return** $\widehat{\mathcal{L}}_{\mathrm{LT}}(y \mid c) \leftarrow \widehat{\mathcal{L}}/S$

**Algorithm 2 LT-MELBO-CV** — Latent-Token Marginal ELBO with Control Variates

**Require:** Policy $p_\theta$, old policy $p_{\theta_{\text{old}}}$, sequence $\boldsymbol{x}_0 = (c, y)$ with completion length $L$, mask token `[MASK]`, KL weight $\lambda_{\text{KL}}$, CV weights $a_1, a_2$, temperature $\tau$

**Ensure:** Log-ratio proxy $\Delta \approx \log \frac{\pi_\theta(y|c)}{\pi_{\theta_{\text{old}}}(y|c)}$

Sample $t \sim \text{Uniform}(0, 1)$; sample mask $\boldsymbol{M} \in \{0, 1\}^L$ with $M_i \sim \text{Bernoulli}(t)$
Ensure $\|\boldsymbol{M}\|_1 \geq 1$ and $\|\mathbf{1} - \boldsymbol{M}\|_1 \geq 1$; construct $\boldsymbol{x}_{\text{partial}} = \textsc{ApplyMask}(c, y, \boldsymbol{M})$

**Pass 1: Proposal Generation**

$\boldsymbol{\ell}_\theta^{(0)} \leftarrow f_\theta(\boldsymbol{x}_{\text{partial}})$, $\boldsymbol{\ell}_{\text{old}}^{(0)} \leftarrow f_{\theta_{\text{old}}}(\boldsymbol{x}_{\text{partial}})$
**for** each masked position $i$ with $M_i = 1$ **do**
    $q_{\theta,i}^{(0)}(v) \leftarrow \text{softmax}(\boldsymbol{\ell}_{\theta,i}^{(0)}/\tau)_v$; $\tilde{e}_i \leftarrow \sum_{v \in \text{Top-}K} q_{\theta,i}^{(0)}(v)\, E[v]$
**end for**
Build $\tilde{\boldsymbol{x}}$: unmasked use $E[y_i]$, masked use $\tilde{e}_i$

**Pass 2: Self-Conditioned Scoring**

$\boldsymbol{\ell}_\theta^{(1)} \leftarrow f_\theta(\tilde{\boldsymbol{x}})$, $\boldsymbol{\ell}_{\text{old}}^{(1)} \leftarrow f_{\theta_{\text{old}}}(\tilde{\boldsymbol{x}})$

**Masked ELBO Terms**

$Z_\theta \leftarrow \frac{1}{t} \sum_{i:M_i=1} \log p_{\theta,i}^{(1)}(y_i \mid \tilde{\boldsymbol{x}})$, $R_{\text{KL},\theta} \leftarrow \frac{1}{t} \sum_{i:M_i=1} D_{\text{KL}}(q_{\theta,i}^{(0)} \| p_{\theta,i}^{(1)})$
$\ell_{\text{LT}}(\theta) \leftarrow Z_\theta - \lambda_{\text{KL}} R_{\text{KL},\theta}$; compute $\ell_{\text{LT}}(\theta_{\text{old}})$ analogously

**Control Variates**

$C_{\text{coh}}(\theta) \leftarrow \frac{1}{|\{i:M_i=0\}|} \sum_{i:M_i=0} -\log p_{\theta,i}^{(1)}(y_i \mid \tilde{\boldsymbol{x}})$, $C_{\text{ent}}(\theta) \leftarrow \frac{1}{L} \sum_{i=1}^{L} \text{H}(p_{\theta,i}^{(1)})$
$\Delta C_{\text{coh}} = C_{\text{coh}}(\theta) - C_{\text{coh}}(\theta_{\text{old}})$, $\Delta C_{\text{ent}} = C_{\text{ent}}(\theta) - C_{\text{ent}}(\theta_{\text{old}})$

**Final Log-Ratio**

$\Delta_{\text{LT}} = \ell_{\text{LT}}(\theta) - \ell_{\text{LT}}(\theta_{\text{old}})$
**return** $\Delta \leftarrow \Delta_{\text{LT}} + a_1\, \Delta C_{\text{coh}} + a_2\, \Delta C_{\text{ent}}$