# DIMBA: REVOLUTIONIZING THEORETICAL ULTRA-FAST INFERENCE AND ADVANCED REASONING WITH MAMBA-BASED DIFFUSION

A PREPRINT

**Faris Allafi**
Independent Researcher
faris.qais.allafi@gmail.com

April 26, 2025

**ABSTRACT**

Standard autoregressive language models incur high inference latency due to sequential token-by-token decoding, limiting their applicability in real-time scenarios and long-context tasks. Existing non-autoregressive methods often trade generation quality for speed, failing to match transformer-based models in coherence and fluency. We introduce **DIMBA**, a non-autoregressive architecture that fuses a cosine-scheduled diffusion process with a Mamba-2 state-space model (SSM) denoiser to generate entire token sequences in parallel. DIMBA begins with noisy token embeddings and iteratively refines them over $T$ diffusion steps, conditioned on a prompt and timestep embeddings. Leveraging Mamba-2's efficient long-range dependency modeling, DIMBA achieves controllable trade-offs between inference speed and output fidelity by adjusting $T$(steps). We hypothesize that DIMBA can deliver significant latency reduction, maintain semantic coherence, and scale gracefully to long contexts due to the SSM backbone. **Disclaimer:** This work is architectural; implementation and empirical evaluation are future work due to current compute constraints.

## 1 Introduction

Modern large language models (LLMs) such as GPT and PaLM achieve state-of-the-art performance across a variety of NLP tasks but suffer from high inference latency caused by autoregressive decoding, where each token is generated sequentially. This latency grows quadratically with sequence length $L$, posing challenges for interactive systems and latency-sensitive applications.

Non-autoregressive generation strategies—including mask-predict methods Gu et al. [2018], insertion-based decoding Lee et al. [2019], and iterative refinement Ghazvininejad et al. [2019]—attempt to parallelize text generation but often degrade fluency and coherence compared to autoregressive baselines. Diffusion models have demonstrated powerful generative capabilities in vision and audio by reversing a structured noise corruption process Ho et al. [2020]. Adapting diffusion to text requires operating in continuous embedding spaces to handle the discrete nature of language.

State-space models (SSMs), particularly the Mamba-2 architecture Dao et al. [2023], provide linear-time sequence modeling with selective recurrence and parallel scan operations. They offer a compelling alternative to Transformers for long-context tasks. We propose **DIMBA**, which integrates diffusion-based generation with Mamba-2 SSM layers to enable fast, parallel, high-quality text generation.

**Contributions:**

- A novel diffusion–SSM fusion architecture combining a cosine-scheduled diffusion process with Mamba-2 denoising.
- A comprehensive description of DIMBA's components, data flow, and theoretical rationale.
- Conceptual training and inference procedures, plus a roadmap for future implementation and evaluation.

## 2 Related Work

### 2.1 Autoregressive Models

Transformer architectures Vaswani et al. [2017] have dominated the landscape of modern large language models, establishing state-of-the-art performance across numerous natural language processing tasks. However, these models inherently suffer from $O(L)$ decoding latency due to their autoregressive nature, where each token must be generated sequentially. This limitation becomes particularly problematic for real-time applications and long-context scenarios. More recently, state-space models such as Mamba-2 Dao et al. [2023] have emerged as promising alternatives, offering linear-time sequence processing capabilities without incurring the quadratic computational cost associated with attention mechanisms.

### 2.2 Non-Autoregressive Generation

Several approaches have been proposed to overcome the sequential generation bottleneck. Mask-Predict Gu et al. [2018] introduced a parallel decoding strategy that iteratively refines token predictions. Similarly, insertion-based decoding methods Lee et al. [2019] and conditional masked language models Ghazvininejad et al. [2019] attempt to parallelize the generation process. While these techniques offer theoretical speedups, they frequently compromise on output fluency and coherence when compared to their autoregressive counterparts, particularly for longer sequences where maintaining global consistency becomes challenging.

### 2.3 Diffusion Models

Denoising diffusion probabilistic models (DDPM) Ho et al. [2020] and denoising diffusion implicit models (DDIM) Song et al. [2021] have demonstrated remarkable capabilities in generating high-fidelity samples across continuous domains, particularly in computer vision and audio synthesis. These models operate by learning to reverse a gradual noise corruption process. The introduction of cosine noise schedules Nichol and Dhariwal [2021] has further enhanced sample quality while reducing the need for extensive hyperparameter tuning. Adapting diffusion frameworks to discrete domains like text generation presents unique challenges but offers promising avenues for parallel decoding.

### 2.4 State-Space Models

State-space models (SSMs) represent a class of sequence modeling architectures that leverage structured recurrence patterns. Models such as S4 Gu et al. [2021] and the more recent Mamba-2 Dao et al. [2023] have demonstrated efficient long-range dependency modeling capabilities through selective state-space parameterizations. These models achieve linear scaling with sequence length while maintaining competitive performance on tasks requiring long-context understanding. Their ability to process information in parallel during training while selectively applying recurrence during inference makes them particularly suitable for integration with other generative paradigms.

## 3 The DIMBA Architecture

### 3.1 High-Level Data Flow

DIMBA frames text generation as a parallel denoising process, fundamentally departing from the sequential nature of autoregressive models. Figure 1 illustrates the end-to-end data flow of our proposed architecture. The process begins with token embeddings derived from prompt tokens, which are then processed through a prompt encoder to generate conditioning information. Simultaneously, these embeddings undergo a controlled noise corruption process according to a cosine schedule. The resulting noisy embeddings are iteratively refined through a diffusion-SSM fusion denoiser, conditioned on both the prompt representation and timestep embeddings. Finally, the denoised embeddings are projected to token logits and decoded to produce the generated text sequence.

### 3.2 Components

#### 3.2.1 Token Embeddings

The initial token embeddings $X_0 \in \mathbb{R}^{L \times d}$ are obtained by mapping discrete tokens to a continuous embedding space via a learned embedding matrix $E$. These embeddings serve as the foundation for both the conditioning pathway and the diffusion process.

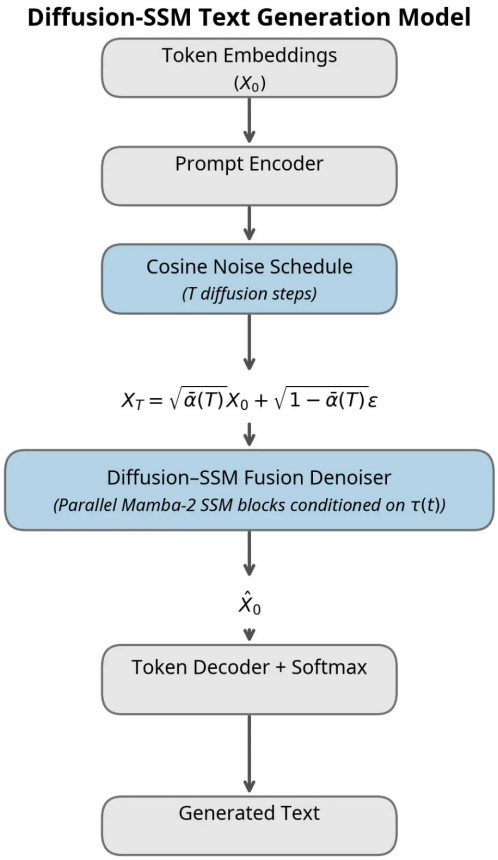


Figure 1: The DIMBA architecture for non-autoregressive text generation. The model fuses a cosine-scheduled diffusion process with Mamba-2 SSM blocks to enable parallel generation of entire token sequences.

#### 3.2.2 Prompt Encoder

A lightweight MLP or frozen encoder processes the token embeddings to produce conditioning information $C \in \mathbb{R}^{L \times d_c}$. This conditioning vector encapsulates the semantic content of the prompt and guides the denoising process toward contextually appropriate generations.

#### 3.2.3 Cosine Noise Schedule

We adopt a cosine noise schedule following Nichol and Dhariwal [2021], defined as:

$$\bar{\alpha}(t) = \cos^2\left(\frac{t/T + s}{1+s} \cdot \frac{\pi}{2}\right), \quad s = 0.008$$

$$\beta_t = 1 - \frac{\bar{\alpha}(t)}{\bar{\alpha}(t-1)}$$

The noise injection process is formulated as:

$$X_T = \sqrt{\bar{\alpha}(T)}X_0 + \sqrt{1-\bar{\alpha}(T)}\epsilon$$

where $\epsilon \sim \mathcal{N}(0, I)$ represents Gaussian noise. This schedule provides a smooth transition from clean to noisy embeddings, facilitating stable training and inference.

#### 3.2.4 Timestep Embedding

To inform the denoiser about the current noise level, we employ sinusoidal positional encodings processed through an MLP to yield timestep embeddings $\tau(t) \in \mathbb{R}^d$. These embeddings enable the model to adapt its denoising strategy based on the specific diffusion step.

#### 3.2.5 Mamba-Denoiser

The core component of DIMBA is the diffusion-SSM fused denoiser, which takes the tuple $(X_t, C, \tau(t))$ as input. The denoiser applies either additive or Feature-wise Linear Modulation (FiLM) conditioning and processes the resulting representation through $N$ Mamba-2 blocks to predict $\hat{X}_{t-1}$. The selective state-space mechanism of Mamba-2 enables efficient modeling of long-range dependencies while maintaining linear computational complexity.

#### 3.2.6 Output Projection

A linear layer, optionally weight-tied with the embedding matrix, maps the denoised embeddings $\hat{X}_0$ to token logits. These logits are then processed through a softmax function to obtain the final token distribution.

### 3.3 Conceptual Training Procedure

The training procedure for DIMBA follows the standard diffusion model training paradigm, adapted for the text domain:

for each batch:
$t \sim \text{Uniform}(1, T)$
$X_t = \sqrt{\bar{\alpha}(t)} X_0 + \sqrt{1 - \bar{\alpha}(t)} \epsilon$
$C = \text{PromptEncoder}(X_0)$
$\tau = \text{MLP}(t)$
$X_{\text{pred}} = \text{Denoiser}(X_t, C, \tau)$
$\mathcal{L} = \|X_{\text{pred}} - X_0\|^2$
update parameters

This approach allows the model to learn the reverse diffusion process at arbitrary timesteps, enabling efficient training through random sampling of diffusion steps.

### 3.4 Conceptual Inference Procedure

During inference, DIMBA generates text by iteratively denoising from random noise, conditioned on the prompt embeddings $X_{\text{prompt}}$. Let $L_{\text{gen}}$ be the desired length of the generated sequence.

The procedure is as follows:

1. **Compute prompt conditioning:** Obtain the conditioning vector $C$ from the input prompt embeddings $X_{\text{prompt}}$:
$$C = \text{PromptEncoder}(X_{\text{prompt}})$$
2. **Initialize with random noise:** Sample the initial noisy state $X_T$ from a standard Gaussian distribution, matching the desired output length $L_{\text{gen}}$ and embedding dimension $d$:
$$X_T \sim \mathcal{N}(0, I) \in \mathbb{R}^{L_{\text{gen}} \times d}$$
3. **Iterative denoising loop:** Iterate backwards from the maximum diffusion step $T$ down to 1:
for $t = T \ldots 1$ :
Compute timestep embedding: $\tau = \text{MLP}(t)$
Predict previous state using the denoiser:
$X_{t-1} = \text{Denoiser}(X_t, C, \tau)$
end

4. **Final projection:** Map the final denoised embeddings $X_0$ to token logits using a linear layer and apply softmax to get the token probabilities:

$$X_0 \rightarrow \text{linear layer} \rightarrow \text{softmax} \rightarrow \text{output tokens}$$

This procedure enables the parallel generation of entire token sequences of length $L_{\text{gen}}$. The number of diffusion steps $T$ controls the trade-off between generation speed (lower $T$ is faster) and output quality (higher $T$ may be required for better results).

# 4 Discussion & Future Work

## 4.1 Hypothesized Advantages

The DIMBA architecture offers several theoretical advantages over existing text generation approaches:

- **Latency Reduction:** By enabling parallel denoising across the entire sequence, DIMBA achieves constant-time per-step complexity, potentially reducing inference latency by orders of magnitude compared to autoregressive models for long sequences.
- **Long-Range Coherence:** Leveraging Mamba-2's efficient long-range dependency modeling capabilities, DIMBA can maintain semantic coherence across extended contexts without incurring the quadratic computational cost associated with attention mechanisms.
- **Controllability:** The diffusion step count $T$ provides a natural mechanism for balancing generation speed against output fidelity, allowing deployment-time adjustments based on specific application requirements.
- **Theoretical Enhanced Reasoning Coherence:**
  DIMBA's parallel generation framework offers a potential benefit for test-time reasoning. Unlike autoregressive approaches that build solutions token-by-token, DIMBA processes the entire sequence simultaneously during its iterative refinement. This allows the model to potentially enforce global semantic coherence and structural consistency required for complex reasoning tasks more effectively, as it operates with a complete view of the intended output throughout generation.
- **Extensibility:** The architecture readily supports extensions such as learned noise schedules, adaptive sampling strategies, and prompt-guided generation techniques, offering a flexible framework for future enhancements.

## 4.2 Potential Challenges

Despite its promising characteristics, several challenges must be addressed in the implementation and evaluation of DIMBA:

- **Training Cost:** Diffusion models with large step counts $T$ may require substantial computational resources during training, potentially limiting accessibility for researchers with constrained compute budgets.
- **Discrete–Continuous Gap:** The mapping between discrete tokens and continuous embedding spaces introduces complexities that may affect generation quality, particularly for rare tokens or specialized vocabularies.
- **Conditioning Robustness:** The efficacy of FiLM or additive fusion mechanisms for incorporating conditioning information requires empirical validation across diverse prompting scenarios.
- **Hyperparameter Sensitivity:** The performance of diffusion-based models often exhibits sensitivity to hyperparameters such as step count $T$, denoiser architecture depth, and learning rate schedules, necessitating careful tuning for a preformant model.

## 4.3 Future Work

We outline a roadmap for future research and development of the DIMBA architecture:

- **Prototype Implementation:** Develop a reference implementation to validate the architectural concepts presented in this work.
- **Proof-of-Concept Training:** Conduct initial training experiments on controlled datasets to establish baseline performance metrics and identify optimization opportunities.

- **Comprehensive Benchmarking:** Evaluate DIMBA against both autoregressive and non-autoregressive baselines across dimensions including generation speed, perplexity, and human evaluations of fluency and coherence.
- **Ablation Studies:** Systematically investigate the impact of various design choices, including noise schedules, conditioning mechanisms, DDIM sampling strategies, and classifier-free guidance techniques.

## 5 Conclusion

This paper introduces DIMBA, a novel non-autoregressive architecture that fuses diffusion-based denoising with Mamba-2 state-space modeling to enable fast, parallel text generation. By framing text generation as an iterative denoising process operating on continuous token embeddings, DIMBA offers a promising approach to overcoming the sequential generation bottleneck inherent in autoregressive models.

The proposed architecture leverages a cosine-scheduled diffusion process combined with efficient SSM-based denoising to achieve controllable trade-offs between inference speed and output fidelity. While this work remains conceptual due to current compute constraints, we provide a comprehensive blueprint for implementation, training, and evaluation.

We believe that DIMBA represents a significant step toward addressing the latency challenges in modern language generation systems while maintaining the quality standards established by state-of-the-art autoregressive models. We encourage the research community to build upon this foundation, implementing and extending DIMBA to advance the frontier of efficient text generation.

## References

Tri Nguyen Dao, Albert Gu, and Jianfeng Ma. Mamba-2: Linear-time sequence modeling with selective state spaces. *arXiv preprint arXiv:2312.00752*, 2023.

Marjan Ghazvininejad, Omer Levy, Yinhan Liu, and Luke Zettlemoyer. Mask-predict: Parallel decoding of conditional masked language models. *Proceedings of the 2019 Conference on Empirical Methods in Natural Language Processing*, pages 6112–6121, 2019.

Albert Gu, Karan Goel, and Christopher Ré. Efficiently modeling long sequences with structured state spaces. *arXiv preprint arXiv:2111.00396*, 2021.

Jiatao Gu, James Bradbury, Caiming Xiong, Victor O.K. Li, and Richard Socher. Non-autoregressive neural machine translation. *arXiv preprint arXiv:1711.02281*, 2018.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. *Advances in Neural Information Processing Systems*, 33:6840–6851, 2020.

Jason Lee, Elman Mansimov, and Kyunghyun Cho. Insertion transformer: Flexible sequence generation via insertion operations. *arXiv preprint arXiv:1902.03249*, 2019.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. *International Conference on Machine Learning*, pages 8162–8171, 2021.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. *International Conference on Learning Representations*, 2021.

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.