Simple and Effective Masked Diffusion Language Models

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, Volodymyr Kulesho - NeurIPS 2024

Presented by: Armin Saghafian, Rui Guo

EECE 571F - Advanced Deep Learning

Autoregressive VS Diffusion

Autoregressive (AR) Models

Models like GPT are constrained to generate text strictly sequentially (left-to-right). Currently, they dominate language modeling due to their superior log-likelihood and perplexity on discrete data.

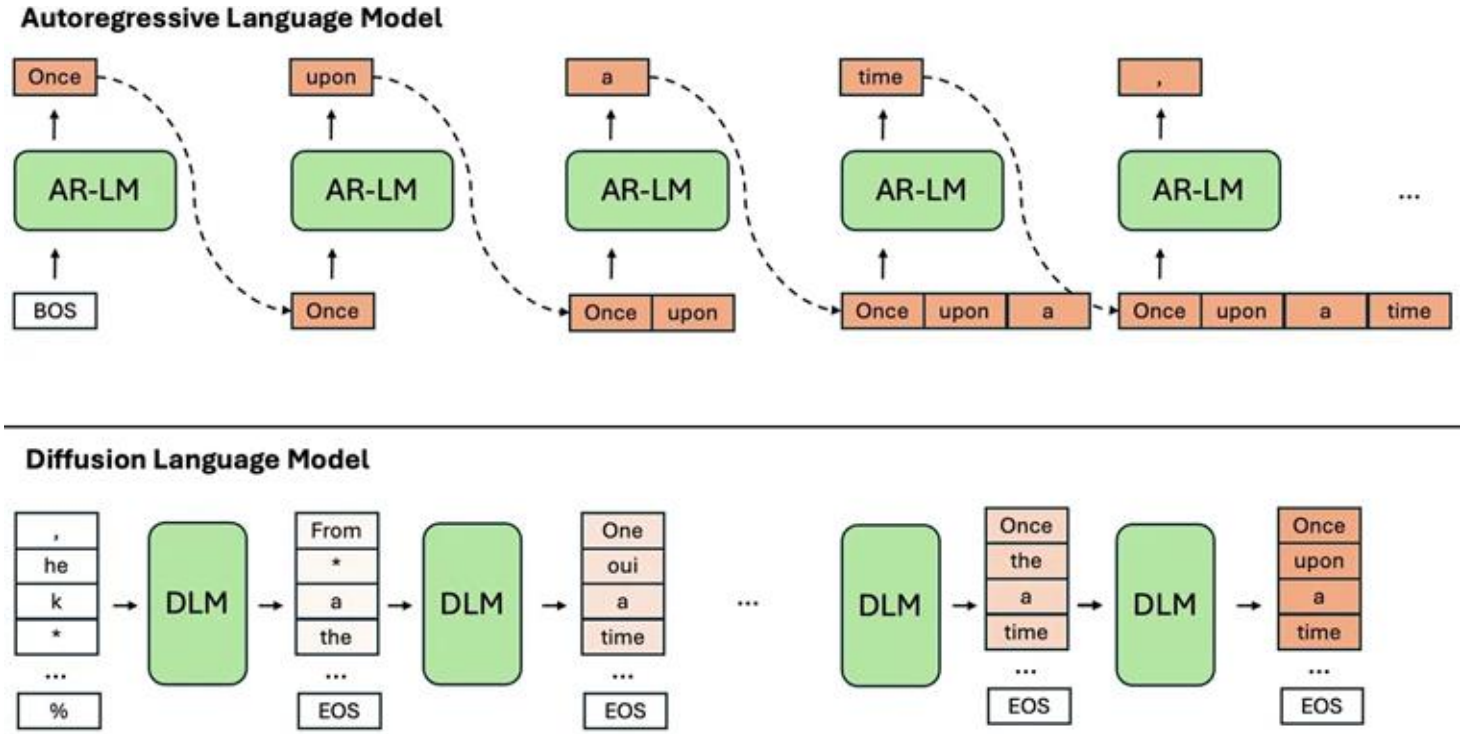
Diffusion Models

Diffusion excels in generating continuous data (like images). However, there has been a significant performance gap when applying these models to discrete data generation (text).

Why Diffusion?

- **Error Propagation:** Future predictions are based on that flawed token, causing errors to propagate and accumulate over time.
- **Indirect Generation Control:** Controlling AR generation is tricky. For example, if you want to generate a passage of a certain length, you either have to train a separate length predictor or do prompting.
- **Long-Term Planning:** By resolving the sequence globally rather than step-by-step, diffusion models can better maintain global consistency.
- **Flexible Sampling:** Potential for much faster decoding speeds, particularly for arbitrary lengths of text via semi-autoregressive generation.

Why Diffusion?



* https://spacehunterinf.github.io/blog/2025/diffusion-language-models/

Previous works

D3PM

A foundational framework for discrete diffusion. Allowed complex state transitions, but its general nature was previously thought to yield mediocre empirical performance in language modeling.

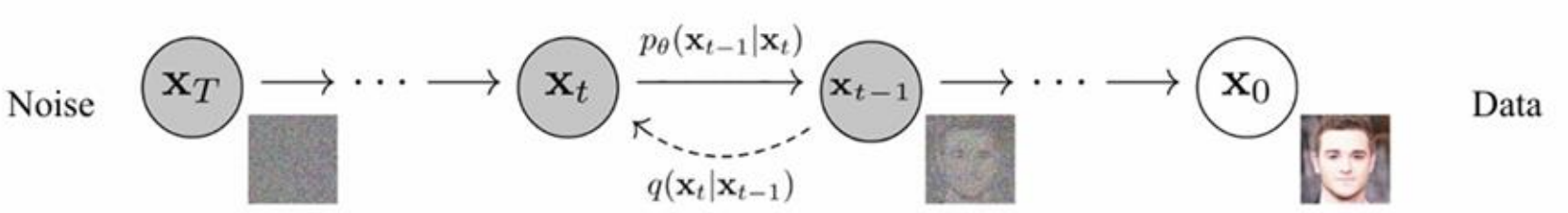
SEDD

Connected score-based diffusion with discrete data to achieve strong results. However, it relies heavily on highly complex Continuous-Time Markov Chain (CTMC) theory, making it difficult to optimize.

The Paper's Contributions

- **Masked Diffusion Language Models (MDLM):** Bypasses complex CTMC theory by focusing entirely on a simple "absorbing state" (masking) process.
- **Fast Samplers:** Introduces highly efficient ancestral and semi-autoregressive decoders.
- **State-of-the-Art Results:** Establishes a new SOTA for discrete diffusion models, finally approaching AR model performance levels.

Preliminaries : Diffusion Models



Forward process: The process of adding noise to the original image, following a pre-defined distribution q

Reverse process: The process of removing the noise added at each step, consisting of the learned target distribution p

Input data : Continuous image data

Noise data: Fully noised data, serving as the initial input to the generative model

Preliminaries： Discrete Diffusion Models

Diffusion Forward Process

$$q(\mathbf{z}_t|\mathbf{x}) = \mathcal{N}\left(\mathbf{z}_t; \sqrt{\alpha_t}\mathbf{x}, (1-\alpha_t)\mathbf{I}\right)$$

Interpolated Discrete Diffusion

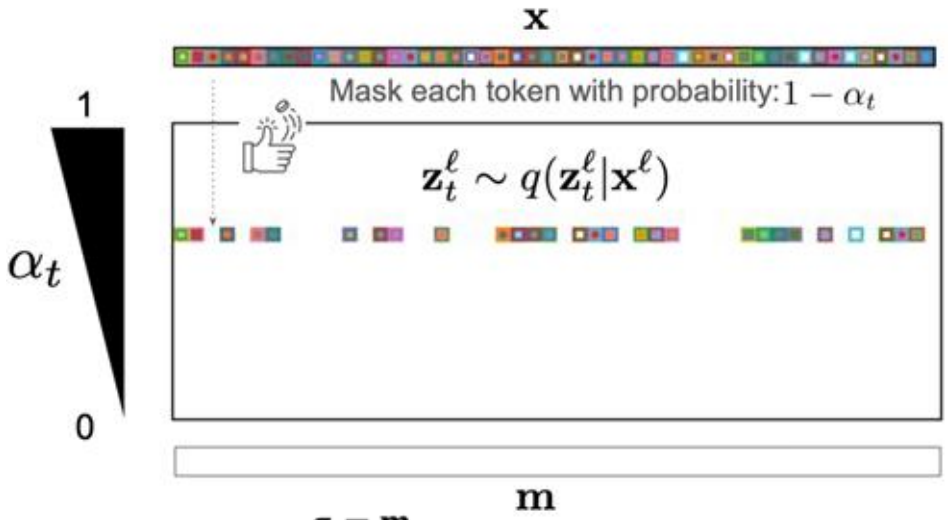
$$q(\mathbf{z}_t|\mathbf{x}) = Cat(\mathbf{z}_t; Q_t^T\mathbf{x}) \xrightarrow{Q_t = \alpha_t\mathbf{I} + (1-\alpha_t)1\pi^T} q(\mathbf{z}_t|\mathbf{x}) = \mathrm{Cat}(\mathbf{z}_t; \alpha_t\mathbf{x} + (1-\alpha_t)\boldsymbol{\pi})$$

Reverse Posterior

$$q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}) = \mathrm{Cat}\left(\mathbf{z}_s; \frac{[\alpha_{t|s}\mathbf{z}_t + (1-\alpha_{t|s})\mathbf{1}\boldsymbol{\pi}^\top\mathbf{z}_t] \odot [\alpha_s\mathbf{x} + (1-\alpha_s)\boldsymbol{\pi}]}{\alpha_t\mathbf{z}_t^\top\mathbf{x} + (1-\alpha_t)\mathbf{z}_t^\top\boldsymbol{\pi}}\right)$$

Masked Diffusion

Forward Masking Process

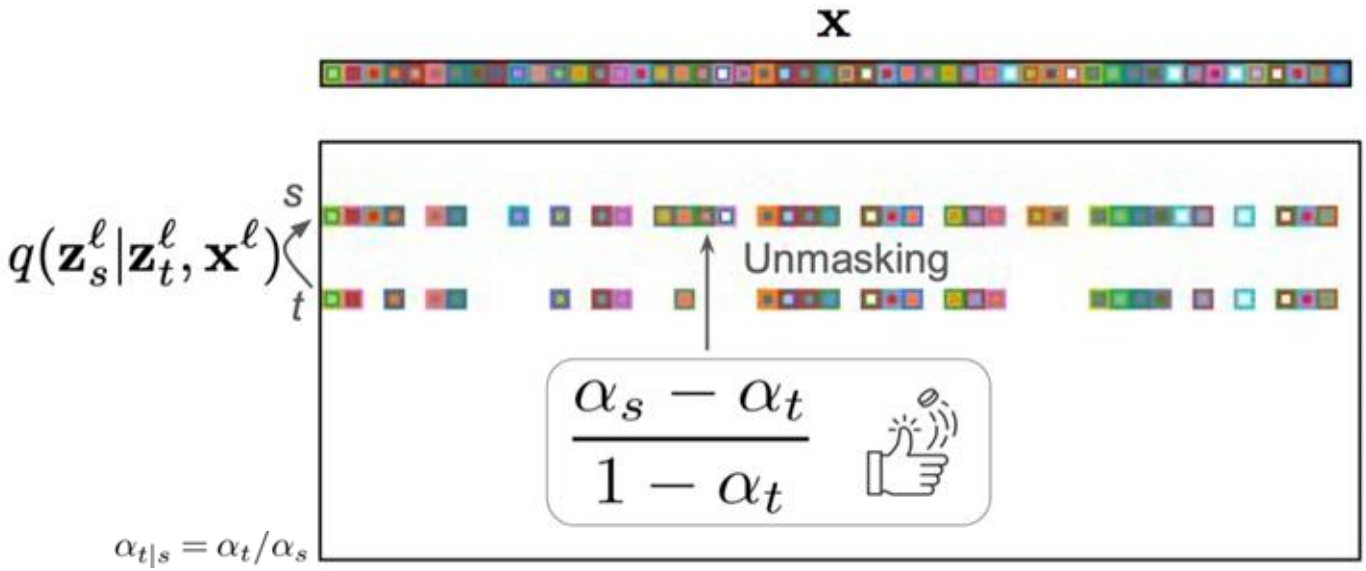


$$q(\mathbf{z}_t|\mathbf{x}) = \text{Cat}(\mathbf{z}_t; \alpha_t \mathbf{x} + (1-\alpha_t)\boldsymbol{\pi}) \xrightarrow{\boldsymbol{\pi} = \mathbf{m}} q(\mathbf{z}_t|\mathbf{x}) = \text{Cat}(\mathbf{z}_t; \alpha_t \mathbf{x} + (1-\alpha_t)\mathbf{m})$$

$$\begin{matrix} a \\ \vdots \\ \text{zoo} \\ [\text{MASK}] \end{matrix} \begin{bmatrix} 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad 0.7 \begin{bmatrix} 1 \\ \vdots \\ 0 \\ 0 \end{bmatrix} \quad 0.3 \begin{bmatrix} 0 \\ \vdots \\ 0 \\ 1 \end{bmatrix} = \begin{matrix} 0.7 \\ \vdots \\ 0 \\ 0.3 \end{matrix} \begin{matrix} a \\ \\ \text{zoo} \\ [\text{MASK}] \end{matrix}$$

Masked Diffusion

Reverse Unmasking Process



$$q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}) = \mathrm{Cat}\left(\mathbf{z}_s; \frac{[\alpha_{t|s}\mathbf{z}_t + (1-\alpha_{t|s})\mathbf{1}\boldsymbol{\pi}^\top \mathbf{z}_t] \odot [\alpha_s \mathbf{x} + (1-\alpha_s)\boldsymbol{\pi}]}{\alpha_t \mathbf{z}_t^\top \mathbf{x} + (1-\alpha_t)\mathbf{z}_t^\top \boldsymbol{\pi}}\right) \xrightarrow{\boldsymbol{\pi} = \mathbf{m}} q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s; \mathbf{z}_t) & \mathbf{z}_t \neq \mathbf{m}, \\ \mathrm{Cat}\left(\mathbf{z}_s; \frac{(1-\alpha_s)\mathbf{m} + (\alpha_s - \alpha_t)\mathbf{x}}{1-\alpha_t}\right) & \mathbf{z}_t = \mathbf{m}. \end{cases}$$

Masked Diffusion

SUBS Parameterization

Masked Diffusion Posterior: The process of removing noise added to the raw token during the training process

$$q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s;\mathbf{z}_t) & \mathbf{z}_t \neq \mathbf{m}, \\ \mathrm{Cat}\left(\mathbf{z}_s;\frac{(1-\alpha_s)\mathbf{m}+(\alpha_s-\alpha_t)\mathbf{x}}{1-\alpha_t}\right) & \mathbf{z}_t = \mathbf{m}. \end{cases}$$

SUBS Parameterization: The process of using the model to remove noise during the inference phase

Parameterization：

$$p_\theta(\mathbf{z}_s|\mathbf{z}_t) = q(\mathbf{z}_s|\mathbf{z}_t,\mathbf{x}=\mathbf{x}_\theta(\mathbf{z}_t,t)) = \begin{cases} \mathrm{Cat}(\mathbf{z}_s;\mathbf{z}_t), & \mathbf{z}_t \neq \mathbf{m}, \\ \mathrm{Cat}\left(\mathbf{z}_s;\frac{(1-\alpha_s)\mathbf{m}+(\alpha_s-\alpha_t)\mathbf{x}_\theta(\mathbf{z}_t,t)}{1-\alpha_t}\right). & \mathbf{z}_t = \mathbf{m}, \end{cases}$$

Replace the raw token that cannot be obtained with predicted token derived from the model

- **Traditional Posterior：** Utilizing raw data to remove noise → Cannot be used during the inference (testing) phase.
- **SUBS Parameterization：** Instead of utilizing the raw data, denoising using data estimated by the model.

Introducing two assumptions applicable to language modeling:

1. **Zero Masking Probabilities：** The model does not predict the [MASK] token (i.e., the model predicts only specific words)
2. **Carry-Over Masking：** The tokens restored in the previous step will remain unchanged in subsequent steps。

Masked Diffusion

Training Objective

$$\mathbb{E}_q \left[\underbrace{D_{KL}(q(\mathbf{x}_T|\mathbf{x}_0) \parallel p(\mathbf{x}_T))}_{L_T} + \sum_{t>1} \underbrace{D_{KL}(q(\mathbf{x}_{t-1}|\mathbf{x}_t, \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1}|\mathbf{x}_t))}_{L_{t-1}} - \underbrace{\log p_\theta(\mathbf{x}_0|\mathbf{x}_1)}_{L_0} \right]$$

Run as the actual training objective function

Objective function when using SUBS parameterization

$$\mathscr{L}_{\text{diffusion}} = \sum_{i=1}^{T} \mathbb{E}_q \left[D_{KL}\Big(q(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}, \mathbf{x}) \parallel p_\theta(\mathbf{z}_{s(i)}|\mathbf{z}_{t(i)}) \Big) \right] = \sum_{i=1}^{T} \mathbb{E}_q \left[\frac{\alpha_{t(i)} - \alpha_{s(i)}}{1 - \alpha_{t(i)}} \log \left\langle \mathbf{x}_\theta(\mathbf{z}_{t(i)}), \mathbf{x} \right\rangle \right]$$

Rao-Blackwellization:

By analytically computing expectations (using Zero Masking & Carry-Over properties), we simplify the general D3PM objective to a cleaner form. This reduces gradient variance during training.

Masked Diffusion Language Model

vocab size{ $\begin{vmatrix} 0.6 \\ \vdots \\ 0.01 \end{vmatrix} \times \begin{vmatrix} 1 \\ \vdots \\ 0 \end{vmatrix} = 0.6$

$$\mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_q \int_{t=0}^{t=1} \frac{\alpha_t'}{1-\alpha_t} \log \langle \mathbf{x}_\theta(\mathbf{z}_t, t), \mathbf{x} \rangle \mathrm{d}t \longrightarrow \mathcal{L}_{\text{NELBO}}^{\infty} = \mathbb{E}_q \int_{t=0}^{t=1} \frac{\alpha_t'}{1-\alpha_t} \sum_{\ell} \log \langle \mathbf{x}_\theta^{\ell}(\mathbf{z}_t^{1:L}, t), \mathbf{x}^{\ell} \rangle \mathrm{d}t$$

Performance Improvement: The final objective function is derived by applying the continuous-time likelihood bound to optimize model performance.

Definition of $\mathbf{z}_t^{1:L}$: The state of a sentence of length L at time step t → i.e., the state where some tokens are replaced by [MASK].

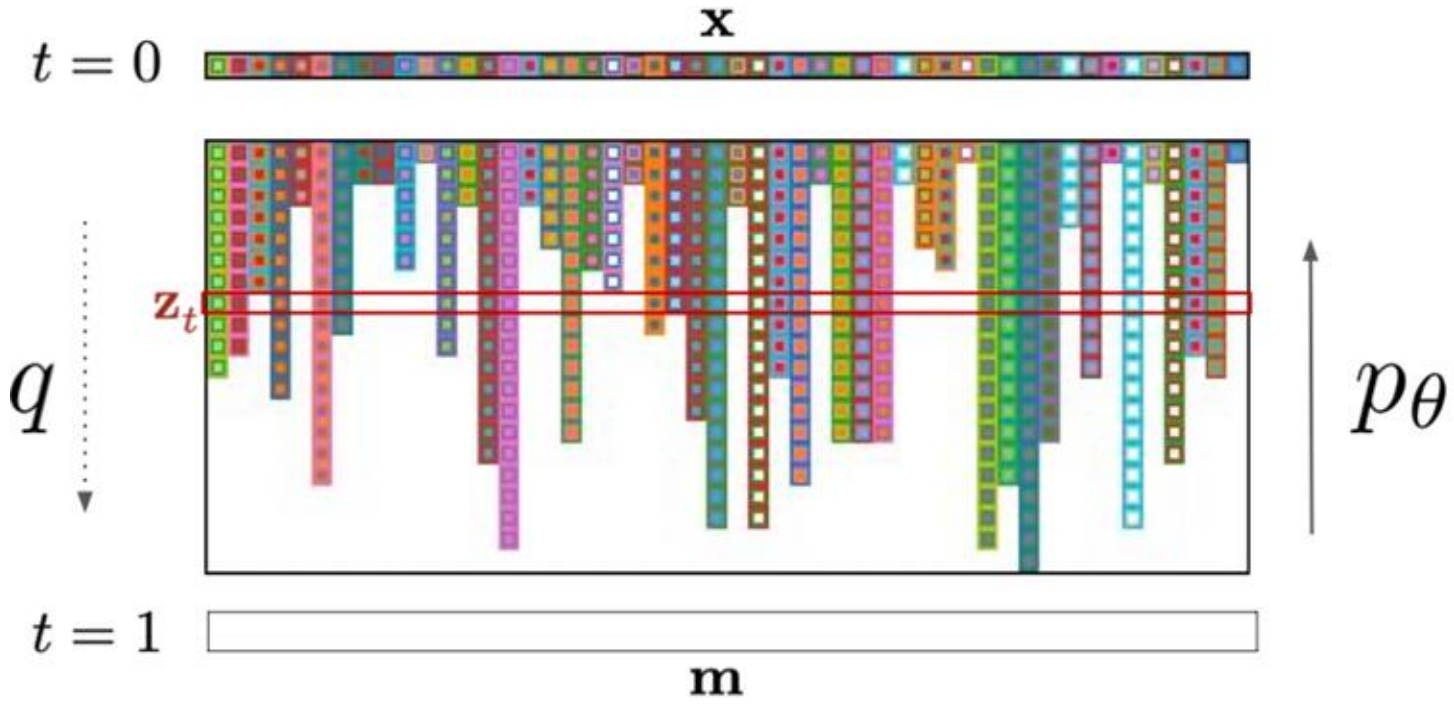
Role of the inner term log: To extract the likelihood of the actual tokens.

- **Objective Function:** At each time step, maximize the log-likelihood of the reconstructed [MASK] token.
- The form of this objective function is almost identical to BERT's Masked Language Model (MLM).

Weight ($\frac{\alpha_t'}{1-\alpha_t}$**):** Represents the proportion of the [MASK] token that is reconstructed at the current time step.

When t is large: Most tokens consist of [MASK] tokens → is lower, accounting for a higher proportion of the overall objective function

When t is small: Most tokens consist of real tokens → is higher, accounting for a lower proportion of the overall objective function


x
$t = 0$
$\mathbf{z}_t$
q
p_θ
$t = 1$
m

Training

Algorithm 1 Training MDLM

1: **repeat**
2: $\mathbf{x}^{1:L} \sim q(\mathbf{x})$ ▷ Sample a sentence.
3: $t \sim \mathcal{U}[0,1]$ ▷ Sample a time step.
4: $\mathbf{z}_t^\ell \sim \text{Cat}(\mathbf{z}_t^\ell; \alpha_t \mathbf{x}^\ell + (1-\alpha_t)\mathbf{m}) \; \forall \, 1 \le \ell \le L$ ▷ Mask Each token $\mathbf{x}^\ell$ independently to obtain the latent $\mathbf{z}_t^{1:L}$.
5: Take gradient descent step on

$$\nabla_\theta \frac{\alpha_t'}{1-\alpha_t} \sum_\ell \log \langle \mathbf{x}_\theta^\ell(\mathbf{z}_t^{1:L}, t), \mathbf{x}^\ell \rangle$$

6: **until** converged

1. **Data Sampling:** Corresponds to the construction of mini-batches in general deep learning.

2. **Step Sampling:** A specific step (time step t) is set to determine the actual [MASK] ratio.

3. **Random Token [MASK] Conversion:** Tokens with a ratio of is converted to [MASK].

4. **Model Training:** Training is performed using a weighted MLM (Masked Language Model) loss.

Although the mathematical derivation is highly complex, the final implementation is remarkably simple —it essentially involves applying a weighting to BERT's MLM loss function.

Inference and Sampling

Efficient Ancestral Sampling

- Unmasked tokens remain unchanged during the reverse process.
- If no new tokens are unmasked and the network is not conditioned on time, reuse the cached output.

Semi-Autoregressive (SAR) Decoding

- Generate sequences of arbitrary length.
- Use the latter portion of a generated sequence as a fixed prefix for the next round.

Experimental Setup

- Tasks:
  1. Generative Model (Log Likelihood):
     a. One Billion Words Dataset (LM1B)
     b. OpenWebText
  2. Downstream Tasks
     a. General Language Understanding Evaluation (GLUE) benchmark

- Model:
  bert-base-uncased

Generative Model

Table 1: Test perplexities (PPL; ↓) on LM1B. †Reported in He et al. [26]. Best diffusion value is bolded.

		Parameters	PPL (↓)
Autoregressive	Transformer-X Base [13]	0.46B	23.5
	OmniNet$_T$ [61]	100M	21.5
Diffusion	BERT-Mouth [64]†	110M	$\leq$142.89
	D3PM (absorb) [1]	70M	$\leq$76.90
	Diffusion-LM [30]†	80M	$\leq$118.62
	DiffusionBert [26]	110M	$\leq$63.78
	SEDD [33] (33B tokens)	110M	$\leq$ 32.79
Autoregressive (Retrained)	Transformer (33B tokens)	110M	22.32
	Transformer (327B tokens)		20.86
Diffusion (Ours)	MDLM (33B tokens)	110M	$\leq$27.04
	MDLM (327B tokens)		$\leq$**23.00**

Downstream Tasks

Table 4: GLUE evaluation results. Evaluation measures (↑) are F1 score for QQP and MRPC, Spearman correlations for STS-B, and accuracy for the rest. For MNLI, we report match/mismatch accuracies.

	MNLI (m/mm)	QQP	QNLI	SST-2	COLA	STS-B	MRPC	RTE	Avg
AR	80.94/80.78	86.98	86.16	90.14	33.43	84.32	83.88	47.29	74.88
BERT	84.43/85.35	88.41	**90.46**	**92.20**	54.81	**88.41**	89.16	61.37	81.62
+MDLM-FT	**84.76/85.07**	**88.49**	90.30	**92.20**	**57.69**	87.48	**90.53**	**62.09**	**82.06**

Semi-Autoregressive Modeling

Compared to SSD-LM

Generates 25 token block

Their model generates 512 tokens at each block (25 to 30 times faster)

Table 5: Semi-AR generative perplexity (Gen. PPL; ↓) for sequences of 2048 tokens.

	Gen. PPL (↓)	Sec/Seq (↓)
SSD-LM	35.43	2473.9
MDLM (Ours)	**27.18**	**89.3**

DNA Models

Base Model:

State Space Models (Caduceus DNA)

Dataset:

HG38 human reference genome

		Params	PPL (↓)
Autoregressive (Retrained)	Mamba	465K	$3.067 \pm .010$
	HyenaDNA	433K	$3.153 \pm .001$
Diffusion (Retrained)	Plaid	507K	$\leq 3.240 \pm .005$
	SEDD	467K	$\leq 3.216 \pm .003$
Diffusion (Ours)	MDLM	467K	$\leq \mathbf{3.199} \pm .010$

Ablation Analysis

Modern Engineering: Simply applying modern practices (Diffusion Transformers with rotary embeddings, better tokenization, low-discrepancy samplers) massively improves the performance of even older baselines like D3PM.

Continuous-Time Formulation: Moving from discrete steps to a continuous-time integral formulation results in lower perplexity.

"**Carry-Over" Unmasking:** The SUBS parameterization ensures unmasked tokens aren't pointlessly re-predicted. Removing this single mathematical property worsens perplexity by over 1.5 points.

Table 8: Test perplexities (PPL; ↓) for MDLM ablations on LM1B. For the discrete-time models, we use $T = 1000$. Standard deviation is measured over 5 seeds during evaluation.

	PPL (≤)
MDLM (47)	**27.04**±.01
w/o continuous time (43)	27.19±.07
& w/o carry-over (41)	28.56±.15
& w/o zero masking (39)	28.51±.15

Comments

Notes

- **Still 10–15% behind AR perplexity**
- **Engineering vs. Theory:** A significant portion of the performance gain over prior baselines (like D3PM) is a result of modern engineering practices
- **Generalizes beyond NLP — DNA results**

- **BERT training is a special case of masked diffusion:** The complex diffusion objective mathematically reduces to a weighted average of standard Masked Language Modeling (MLM) losses

Speed

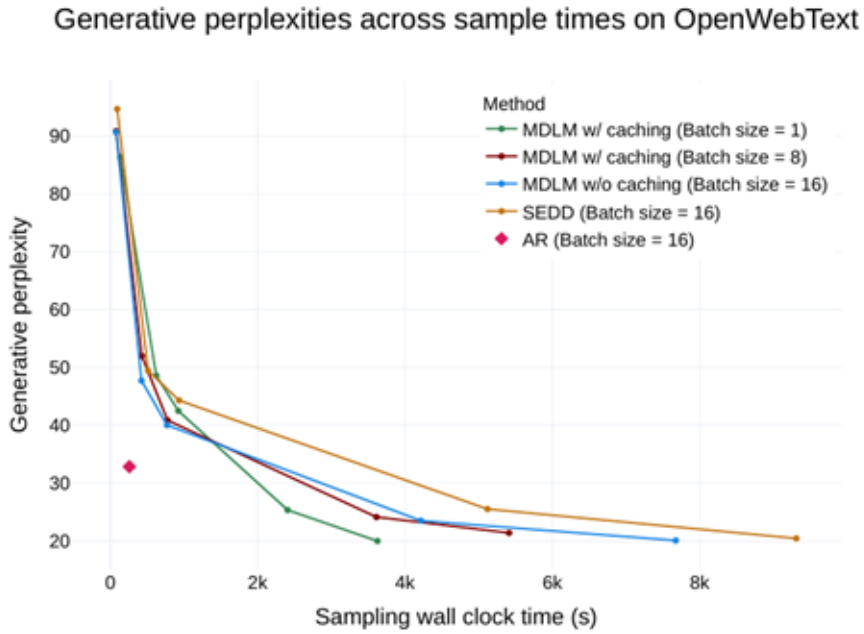


Figure 2: Generative perplexities across wall clock time for generating 64 samples on OWT using a single 32GB A5000 GPU are compared by varying $T \in \{100, 500, 1000, 5000, 10000\}$ in the reverse diffusion process. The samples are generated in mini-batches with a batch size of 16 for AR, SEDD, and MDLM without caching, as it is the largest batch size that fits on this GPU. For MDLM with caching, we vary the batch size.

What I wanted to see

- Scaling?
- controllability and planning advantages over AR, but never actually demonstrate these advantages empirically

Summary

- Shifted focus strictly to absorbing state (Masking) diffusion, bypassing complex CTMC math.
- Derived a clean, continuous-time ELBO that reduces to a weighted Masked Language Modeling (MLM) loss.
- Enabled fast sampling and arbitrary-length semi-autoregressive (SAR) generation.
- Endowed BERT-style encoders with generative capabilities, achieving state-of-the-art results among diffusion language models.

Pros & Cons

Pros:

- Mathematically elegant and highly stable to train.
- Retains strong downstream representation learning (GLUE scores) while acting as a generative model.
- Versatile: Effective not just on text, but also on biological sequences like DNA.

Cons:

- Despite being the best diffusion model, it still trails behind traditional Autoregressive (AR) models in perplexity.
- Semi-autoregressive generation, while faster than older diffusion models, still cannot beat the raw speed of highly optimized AR decoding.

Future Work

THE UNIVERSITY OF BRITISH COLUMBIA