INTERPOLATING AUTOREGRESSIVE AND DISCRETE DENOISING DIFFUSION LANGUAGE MODELS

Anonymous authors
Paper under double-blind review

ABSTRACT

Diffusion language models offer unique benefits over autoregressive (AR) models due to their potential for parallelized generation and controllability, yet they lag in likelihood modeling and are limited to fixed-length generation. In this work, we introduce a class of semi-autoregressive (SAR) diffusion models that interpolate between discrete denoising diffusion and autoregressive models. We propose a recipe for building effective SAR models that includes an efficient training algorithm, estimators of gradient variance, and data-driven noise schedules to minimize the variance. SAR models overcome key limitations of diffusion language models, setting a new state-of-the-art performance on language modeling benchmarks and enabling generation of arbitrary-length sequences.

1 INTRODUCTION

Diffusion models are widely used to generate images (Ho et al., 2020; Dhariwal & Nichol, 2021) and videos (Ho et al., 2022; Gupta et al., 2023), and are becoming increasingly effective at generating discrete data such as text (Lou et al., 2023; Sahoo et al., 2024) or biological sequences (Avdeyev et al., 2023). Compared to autoregressive models, diffusion models have the potential to improve the controllability of model outputs and to accelerate generation.

However, discrete diffusion models currently face at least two limitations. First, in applications such as chat systems, models must generate output sequences of arbitrary length (e.g., a response to a user's question). However, most recent diffusion architectures only generate fixed-length vectors (Austin et al., 2021; Lou et al., 2023). Second, the quality of discrete diffusion models, as measured by standard metrics such as perplexity, lags behind autoregressive approaches and further limits their applicability (Gulrajani & Hashimoto, 2024; Sahoo et al., 2024).

This paper makes progress towards addressing both limitations by introducing semi-autoregressive denoising discrete diffusion language models (SAD3-LMs), which interpolate between diffusion and autoregressive modeling. Specifically, SAD3-LMs define an autoregressive probability distribution over blocks of discrete random variables; the conditional probability of a block given previous blocks is specified by a denoising discrete diffusion model (Austin et al., 2021; Sahoo et al., 2024).

Developing effective SAD3-LMs involves two challenges. First, efficiently computing the training objective for a semi-autoregressive model is not possible using one standard forward pass of a neural network and requires developing specialized algorithms. Second, training is hampered by the high variance of the gradients of the diffusion objective, causing SAD3-LMs to under-perform autoregression even with a block size of one (when both models should be equivalent). We derive estimators of gradient variance, and demonstrate that it is a key contributor to the gap in perplexity between autoregression and diffusion. We then propose custom noise processes that minimize gradient variance and make progress towards closing the perplexity gap.

We evaluate SAD3-LMs on language modeling benchmarks, and demonstrate that they are able to generate sequences of arbitrary length, including lengths that exceed their training context. In addition, SAD3-LMs achieve new state-of-the-art perplexities among discrete diffusion models. Compared to alternative semi-autogressive formulations that perform Gaussian diffusion over embeddings (Han et al., 2022; 2023), our discrete approach features tractable likelihood estimates and yields samples with improved generative perplexity using an order magnitude fewer generation steps. In summary, our work makes the following contributions:

- We introduce semi-autoregressive denoising discrete diffusion models, which are autoregressive over blocks of tokens; conditionals over each block are based on discrete diffusion.
- We introduce custom training algorithms for semi-autoregressive models that enable efficiently leveraging the entire batch of tokens provided to the model.
- We identify gradient variance as a limiting factor of the performance of diffusion models, and we propose custom data-driven noise schedules that reduce gradient variance.
- Our results set a new state-of-the-art in perplexity for discrete diffusion and make progress towards reducing the gap to autoregressive models.

2 Background: Language Modeling Paradigms

2.1 Autoregressive Models

Consider a sequence of L tokens $\mathbf{x} = \left(x^1, \ldots, x^L\right)$ drawn from the data distribution $q(\mathbf{x})$. We aim to fit a model $p_\theta(\mathbf{x})$ of q. Autoregressive (AR) models define a factorized distribution of the form

$$\log p_\theta(\mathbf{x}) = \sum_{i=1}^{L} \log p_\theta(\mathbf{x}^i \mid \mathbf{x}^{<i}), \tag{1}$$

where each $p_\theta(\mathbf{x}^i \mid \mathbf{x}^{<i})$ is parameterized directly with a neural network. As a result, AR models may be trained efficiently via next token prediction. However, AR models take L steps to generate L tokens due to the sequential dependencies.

2.2 Discrete Denoising Diffusion Probabilistic Models

Diffusion models fit a model $p_\theta(\mathbf{x})$ to undo a forward corruption process q (Sohl-Dickstein et al., 2015; Ho et al., 2020). This process starts with clean data $\mathbf{x}$ drawn from the data distribution $q(\mathbf{x})$ and defines latent variables $\mathbf{x}_t$ for $t \in [0, 1]$ that represent progressively noisy versions of $\mathbf{x}$. In the discrete-time setting, the D3PM framework (Austin et al., 2021) defines q to be a Markov forward process $q(\mathbf{x}_t \mid \mathbf{x}_{t-1}) = \text{Cat}(\mathbf{x}_t; Q_t \mathbf{x}_{t-1})$ defined by the multiplication of matrices Q_t over T discrete time steps. The Q_t can encode masking, random token changes, related word substitutions, and more.

An ideal diffusion model p_θ is the reverse of the process q. The D3PM framework defines p_θ as

$$p_\theta(\mathbf{x}_s \mid \mathbf{x}_t) = \sum_{\mathbf{x}} q(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) p_\theta(\mathbf{x} \mid \mathbf{x}_t), \tag{2}$$

where the denoising base model $p_\theta(\mathbf{x} \mid \mathbf{x}_t)$ predicts clean tokens $\mathbf{x}$ given noised tokens $\mathbf{x}_t$. The marginalization is tractable due to the independent noising process over tokens in q, as well as the independence assumptions commonly made in the base model: the clean tokens are modeled independently as $\prod_i p(\mathbf{x}^i \mid \mathbf{x}_t)$.

The diffusion model p_θ is trained using variational inference. Given a number of discretization steps T, defining $s(j) = (j-1)/T$ and $t(j) = j/T$, and using $D_{\text{KL}}[\cdot]$ to denote the Kullback–Leibler divergence, the Negative ELBO (NELBO) equals (Sohl-Dickstein et al., 2015):

$$\mathcal{L}(\mathbf{x}) = \mathbb{E}_q \left[-\log p_\theta(\mathbf{x} | \mathbf{x}_{t(0)}) + \sum_{j=1}^{T} D_{\text{KL}}[q(\mathbf{x}_{s(j)} | \mathbf{x}_{t(j)}, \mathbf{x}) \| p_\theta(\mathbf{x}_{s(j)} | \mathbf{x}_{t(j)})] + D_{\text{KL}}[q(\mathbf{x}_{t(T)} | \mathbf{x}) \| p_\theta(\mathbf{x}_{t(T)})] \right] \tag{3}$$

For brevity, we drop j from $t(j)$ and $s(j)$ below; in general, s will denote the time step before t. This formalism extends to continuous time via Markov chain (CTMC) theory, and admits score-based generalizations (Song & Ermon, 2019; Lou et al., 2023; Sun et al., 2022) and simplifications (Sahoo et al., 2024) that tighten the ELBO and improve performance.

3 Semi-Autoregressive Denoising Discrete Diffusion (SAD3-LM)

We explore a class of semi-autoregressive denoising discrete diffusion language models, SAD3-LMs, that interpolate between autoregressive and diffusion models by defining an autoregressive distribution

over blocks of tokens. We provide a semi-autoregressive objective for maximum likelihood estimation and efficient training and sampling algorithms. We show that for a block size of one, the diffusion objective suffers from high variance despite being equivalent to the autoregressive likelihood in expectation. We identify high training variance as a limitation of diffusion models and propose data-driven noise schedules that reduce the variance of the gradient updates during training.

Notation Consider a sequence of L tokens $\mathbf{x} = \left[x^1, \dots, x^L\right]$ drawn from the data distribution $q(\mathbf{x})$. We group tokens in $\mathbf{x}$ into B blocks of length L' with $B = L/L'$ (we assume that B is an integer). We denote each block $\mathbf{x}^{(b-1)L':bL'}$ from token at positions $(b-1)L'$ to bL' for blocks $b \in \{1, \dots, B\}$ as $\mathbf{x}^b$ for simplicity.

3.1 Semi-Autoregressive Distributions and Model Architectures

We propose to combine the language modeling paradigms in Sec. 2 by autoregressively modeling blocks of tokens and performing diffusion within each block. Our likelihood factorizes over blocks as

$$\log p_\theta(\mathbf{x}) = \sum_{b=1}^{B} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}^{<b}), \tag{4}$$

and each $p_\theta(\mathbf{x}^b \mid \mathbf{x}^{<b})$ is modeled using discrete diffusion over a block of L' tokens. Specifically, we define a reverse diffusion process

$$p_\theta(\mathbf{x}_s^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b}) = \sum_{\mathbf{x}^b} q(\mathbf{x}_s^b \mid \mathbf{x}_t^b, \mathbf{x}^b) p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b})$$

as in (2), but restricted to block b.

We obtain a principled learning objective by applying the NELBO in (3) to each term in (4) to obtain

$$-\log p_\theta(\mathbf{x}) \le \mathcal{L}_{\text{SAR}}(\mathbf{x}, \theta) := \sum_{b=1}^{B} \mathcal{L}(\mathbf{x}^b, \mathbf{x}^{<b}, \theta), \tag{5}$$

where each $\mathcal{L}(\mathbf{x}^b, \mathbf{x}^{<b}, \theta)$ is an instance of (3) applied to $\log p_\theta(\mathbf{x}^b \mid \mathbf{x}^{<b})$. Since the model is defined by $\mathbf{x}_\theta$ conditioned on $\mathbf{x}^{<b}$, we make the dependence on $\mathbf{x}^{<b}, \theta$ explicit in $\mathcal{L}$. We denote the sum of these terms $\mathcal{L}_{\text{SAR}}(\mathbf{x}, \theta)$ (itself a valid NELBO).

Model Architecture Crucially, we parameterize the B base denoiser models $p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b})$ using a single neural network $\mathbf{x}_\theta$. The neural network $\mathbf{x}_\theta$ outputs not only the probabilities $p_\theta(\mathbf{x}^b \mid \mathbf{x}_b^t, \mathbf{x}^{<b})$, but also computational artifacts for efficient training. This will enable us to compute the loss $\mathcal{L}_{\text{SAR}}(\mathbf{x}, \theta)$ in parallel for all B blocks in a memory-efficient manner. Specifically, we parameterize $\mathbf{x}_\theta$ using a transformer (Vaswani et al., 2017) with a block causal attention mask. The transformer $\mathbf{x}_\theta$ is applied to L tokens, and tokens in block b attend to tokens in blocks 1 to b. When $\mathbf{x}_\theta$ is trained, $\mathbf{x}_\theta^b(\mathbf{x}_t^b, \mathbf{x}^{<b})$ yields L' predictions for denoised tokens in block b based on noised $\mathbf{x}_t^b$ and clean $\mathbf{x}^{<b}$.

In autoregressive generation, it is normal to cache keys and values for previously generated tokens to avoid recomputing them at each step. Similarly, we use $\mathbf{K}^b, \mathbf{V}^b$ to denote the keys and values at block b, and we define $\mathbf{x}_\theta$ to support these as input and output. The full signature of $\mathbf{x}_\theta$ is

$$\mathbf{x}_{\text{logits}}^b, \mathbf{K}^b, \mathbf{V}^b \leftarrow \mathbf{x}_\theta^b(\mathbf{x}_t^b, \mathbf{K}^{1:b-1}, \mathbf{V}^{1:b-1}) := \mathbf{x}_\theta^b(\mathbf{x}_t^b, \mathbf{x}^{<b}), \tag{6}$$

where $\mathbf{x}_{\text{logits}}^b$ are the predictions for the clean $\mathbf{x}^b$, and $\mathbf{K}^b, \mathbf{V}^b$ is the key-value cache in the forward pass of x_θ, and $\mathbf{K}^{1:b-1}, \mathbf{V}^{1:b-1}$ are keys and values cached on a forward pass of $\mathbf{x}_\theta$ over $\mathbf{x}^{<b}$ (hence the inputs $\mathbf{x}^{<b}$ and $\mathbf{K}^{1:b-1}, \mathbf{V}^{1:b-1}$ are equivalent).

3.2 Efficient Training and Sampling Algorithms

Ideally, we wish to compute the loss $\mathcal{L}_{\text{SAR}}(\mathbf{x}, \theta)$ in one forward pass of $\mathbf{x}_\theta$. However, observe that denoising $\mathbf{x}_t^b$ requires a forward pass on this noisy input, while denoising the next blocks requires running $\mathbf{x}_\theta$ on the clean version $\mathbf{x}^b$. Thus every block has to go through the model at least twice.

Training Based on this observation, we propose a training algorithm with these minimal computational requirements (Alg. 1). Specifically, we precompute keys and values $\mathbf{K}^{1:B}, \mathbf{V}^{1:B}$ for the full sequence $\mathbf{x}$ in a first forward pass $(\emptyset, \mathbf{K}^{1:B}, \mathbf{V}^{1:B}) \leftarrow \mathbf{x}_\theta(\mathbf{x})$. We then compute denoised predictions for each block using $\mathbf{x}_\theta^b(\mathbf{x}_t^b, \mathbf{K}^{1:b\text{-}1}, \mathbf{V}^{1:b\text{-}1})$. Each token passes through $\mathbf{x}_\theta$ twice.

Vectorized Training Naively, Alg. 1 would apply $\mathbf{x}_\theta^b(\mathbf{x}_{t_b}^b, \mathbf{K}^{1:b\text{-}1}, \mathbf{V}^{1:b\text{-}1})$ in a loop B times. We propose a vectorized implementation that computes $\mathcal{L}_{\text{SAR}}(\mathbf{x})$ in one forward pass on the concatenation $\mathbf{x}_{\text{noisy}} \oplus \mathbf{x}$ of clean data $\mathbf{x}$ with noisy data $\mathbf{x}_{\text{noisy}} = \mathbf{x}_{t_1}^1 \oplus \cdots \oplus \mathbf{x}_{t_B}^B$ obtained by applying a noise level t_b to each block $\mathbf{x}^b$. We mask $\mathbf{x}_{\text{noisy}} \oplus \mathbf{x}$ such that noisy tokens attend to other noisy tokens in their block and to all clean tokens in preceding blocks (see Suppl. D). Our method keeps the overhead of training SAD3-LMs tractable and combines with pretraining to further reduce costs.

Algorithm 1 SAR training

Input: datapoint $\mathbf{x}_0$, # of blocks B, forward noise process $q_t(\cdot|\mathbf{x}_0)$, model $\mathbf{x}_\theta$, loss $\mathcal{L}_{SAR}$
repeat
  Sample $t_1, \ldots, t_B \sim \mathcal{U}(0, 1)$
  $\forall b \in \{1, ..., B\} : \mathbf{x}_{t_b}^b \sim q_{t_b}(\cdot|\mathbf{x}^b)$
  $\emptyset, \mathbf{K}^{1:B}, \mathbf{V}^{1:B} \leftarrow \mathbf{x}_\theta(\mathbf{x})$ ▷ KV cache
  $\forall b$: $\mathbf{x}_{\text{logit}}^b, \emptyset, \emptyset \leftarrow \mathbf{x}_\theta^b(\mathbf{x}_{t_b}^b, \mathbf{K}^{1:b\text{-}1}, \mathbf{V}^{1:b\text{-}1})$
  Let $\mathbf{x}_{\text{logit}} \leftarrow \mathbf{x}_{\text{logit}}^1 \oplus \cdots \oplus \mathbf{x}_{\text{logit}}^B$
  Take gradient step on $\nabla_\theta \mathcal{L}_{SAR}(\mathbf{x}_{\text{logit}})$
until converged

Algorithm 2 SAR Sampling

Input: # blocks B, model $\mathbf{x}_\theta$, diffusion sampling algorithm SAMPLE
$\mathbf{x}, \mathbf{K}, \mathbf{V} \leftarrow \emptyset$ ▷ output & KV cache
for $b = 1$ to B **do**
  $\mathbf{x}^b \leftarrow \text{SAMPLE}(\mathbf{x}_\theta^b, \mathbf{K}^{1:b\text{-}1}, \mathbf{V}^{1:b\text{-}1})$
  $\emptyset, \mathbf{K}^b, \mathbf{V}^b \leftarrow \mathbf{x}_\theta^b(\mathbf{x}^b)$
  $\mathbf{x} \leftarrow \mathbf{x}^{1:b-1} \oplus \mathbf{x}^b$
  $(\mathbf{K}, \mathbf{V}) \leftarrow (\mathbf{K}^{1:b-1} \oplus \mathbf{K}^b, \mathbf{V}^{1:b-1} \oplus \mathbf{V}^b)$
end for
return $\mathbf{x}$

Sampling. We sample one block at a time, conditioned on previously sampled blocks (Alg 2). We may use any sampling procedure $\text{SAMPLE}(\mathbf{x}_\theta^b, \mathbf{K}^{1:b\text{-}1}, \mathbf{V}^{1:b\text{-}1})$ to sample from the conditional distribution $p_\theta(\mathbf{x}_s^b|\mathbf{x}_t^b, \mathbf{x}^{<b})$, where the context conditioning is generated using cross-attention with pre-computed keys and values $\mathbf{K}^{<b}, \mathbf{V}^{<b}$. Similar to AR models, caching the keys and values saves computation instead of recalculating them when sampling a new block.

Notably, our SAR decoding algorithm enables us to sample sequences of arbitrary length, whereas diffusion models are restricted to fixed-length generation. Further, our sampler admits parallel generation within each block, whereas AR samplers are constrained to generate token-by-token.

4 Understanding Likelihood Gaps Between Diffusion & AR Models

4.1 Masked SAD3-LMs

The most effective diffusion models leverage a masking noise process (Austin et al., 2021; Lou et al., 2023; Sahoo et al., 2024), where tokens are gradually replaced with a special mask token. Here, we introduce masked SAD3-LMs, a special class of SAR models based on the masked diffusion language modeling framework (Sahoo et al., 2024).

More formally, we adopt a per-token noise process $q(x_t|x_0) = \text{Cat}(x_t; \alpha_t x_0 + (1 - \alpha_t)m)$ where m is a one-hot encoding of the mask token, and $\alpha_t \in [0, 1]$ is a strictly decreasing function in t, with $\alpha_0 \approx 1$ and $\alpha_1 \approx 0$. Intuitively, the probability of masking a token at time t is $1 - \alpha_t$. We adopt the simplified objective from Sahoo et al. (2024) (the full derivation is provided in Suppl. A):

$$-\log p_\theta(\mathbf{x}) \leq \mathcal{L}_{\text{MDLM}}(\mathbf{x}, \theta) := \sum_{b=1}^{B} \mathbb{E}_{t_b \sim (0,1]} \mathbb{E}_q \frac{\alpha_t'}{1 - \alpha_t} \log p_\theta(\mathbf{x}^b|\mathbf{x}_{t_b}^b, \mathbf{x}^{<b}) \tag{7}$$

where α_t' is the instantenous rate of change of α_t under the continuous-time extension of (3) that takes $T \to \infty$. Under the linear schedule, $\alpha_t' = -1$.

The NELBO is tight for $L' = 1$ but becomes a looser approximation of the true negative log-likelihood for $L' > 1$, $L' \to L$ (see Suppl. C).

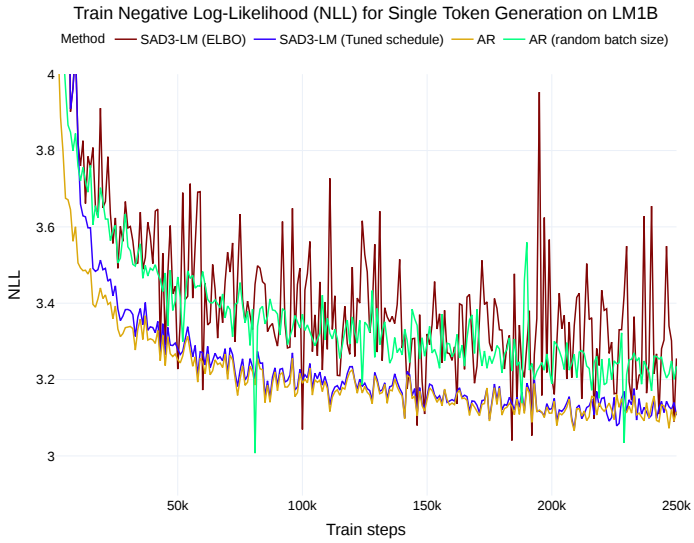

Figure 1: Train NLLs for modeling the per-token likelihood on LM1B. Training under the discrete diffusion ELBO, where half of tokens are masked on average, has similar training variance to an AR model with a random batch size.

4.2 Case study: single token generation

Our SAR parameterization (Eq. 7) is equivalent in expectation to the autoregressive NLL (Eq. 1) in the limiting case where $L' = 1$. Surprisingly, we find a two point perplexity gap between our SAR model for $L' = 1$ and AR when training both models on the LM1B dataset.

Although the objectives are equivalent in expectation, we show that the remaining perplexity gap is a result of high training variance. Whereas an AR model is trained using the cross-entropy of L tokens, our SAR model for $L' = 1$ only computes the cross-entropy for masked tokens $\mathbf{x}_t = \mathbf{m}$, so that $\mathbb{E}_{t\sim\mathcal{U}[0,1]} q(\mathbf{x}_t = \mathbf{m}|\mathbf{x}) = 0.5$. As a result, training on the diffusion objective involves estimating loss gradients with 2x fewer tokens and is responsible for higher variance during training compared to AR.

Table 1: Test perplexities for single-token generation (PPL; ↓) across 16B tokens on LM1B. We also report the variance of the parameter gradient updates over 525M tokens.

	PPL (↓)
AR	**22.88**
+ random batch size	24.37
SAD3-LM $L' = 1$	$\leq$ 25.56
+ tuned schedule	**22.88**

To close the likelihood gap, we train our SAR model for $L' = 1$ by designing the forward process to fully mask tokens, i.e. $q(\mathbf{x}_t = \mathbf{m}|\mathbf{x}) = 1$. Under this schedule, training under the AR objective is *equivalent* to training under the SAR objective (Suppl. B). In Table 1, we show that training under the SAR objective yields the same perplexity as training under the AR objective. Empirically, we see that this reduces the variance of the training loss in Figure 1. We verify that tuning the noise schedule reduces the variance of the gradient updates by measuring it over 525M tokens: while training on the ELBO results in a gradient variance $\text{Var}_{\mathbf{x},t(i)}\left[\nabla_\theta \mathcal{L}_{\text{MDLM}}(\mathbf{x}, \theta)\right] = 0.92$, training under full masking reduces the gradient variance to 0.53.

4.3 Diffusion gap from high variance training

Next, we formally describe the issue of gradient variance in training diffusion models. Given our empirical observations for single-token generation, we propose an estimator for gradient variance that we use to minimize the variance of diffusion model training for $L' \geq 1$.

While the ELBO is invariant to the choice of noise schedule (Suppl. A), this invariance does not hold for our Monte Carlo estimator of the loss used during training. As a result, the variance of the estimator and its gradients are dependent on the schedule. First, we express the estimator of the NELBO with a batch size K. We denote a batch of sequences as $\mathbf{X} = \left[\mathbf{x}^{(1)}, \mathbf{x}^{(2)}, \ldots, \mathbf{x}^{(K)}\right]$, with each $\mathbf{x}^{(k)} \overset{\text{iid}}{\sim} q(\mathbf{x})$. The batch NELBO estimator is given by

$$\mathcal{L}_{\text{MDLM}}(\mathbf{X}) := l_\theta(\mathbf{X}) = \frac{1}{K} \sum_{k=1}^{K} \sum_{b=1}^{B} \frac{\alpha'_t}{1 - \alpha_{t_{(k),b}}} \log p_\theta \left(\mathbf{x}^{(k),b} \mid \mathbf{x}^{(k),b}_{t_{(k),b}}, \mathbf{x}^{(k),<b} \right), \tag{8}$$

where under the linear schedule, α'_t is constant across $t \in [0, 1]$.

We derive the variance of the gradient estimator over M batches $\mathbf{X}^m$ consisting of K sequences each as:

$$\text{Var}_{\mathbf{X},t} \left[\nabla_\theta l(\mathbf{X}) \right] \approx \frac{1}{M-1} \sum_{m=1}^{M} \left\| \nabla_\theta l_\theta(\mathbf{X}^m) - \frac{1}{M} \sum_{m=1}^{M} \nabla_\theta l_\theta(\mathbf{X}^m) \right\|_2^2, \tag{9}$$

where $t \sim \mathcal{U}[0, 1]$.

5 Low-Variance Noise Schedules for Masked SAD3-LMs

5.1 Intuition: Avoid Extreme Mask Rates

We aim to identify schedules that minimize the variance of the gradient estimator and make training most efficient. In a masked setting, we want to mask random numbers of tokens, so that the model learns to undo varying levels of noise, which is important during sampling. However, if we mask very few tokens, reconstructing them is easy, and does not provide useful learning signal. If we mask everything, the optimal reconstruction are the marginals of each token in the data distribution, which is easy to learn, and again is not useful. These extreme masking rates lead to poor high-variance gradients: we want to learn how to clip them via a simple and effective new class of schedules.

5.2 Clipped Schedules for Low-Variance Gradients

We propose a class of "clipped" noise schedules that sample mask rates $1 - \alpha_t \sim \mathcal{U}[\beta, \omega]$ for $0 \leq \beta, \omega \leq 1$. We argue that from the perspective of deriving Monte Carlo gradient estimates, these schedules are equivalent to a continuous schedule where the mask probability is approximately 0 outside of the specified range such that $1 - \alpha_{<\beta}, 1 - \alpha_{>\omega} \approx \epsilon$ and α'_t is linear within the range: $\alpha'_t \approx \beta - \omega$.

5.3 Data-Driven Clipped Schedules Across Block Sizes

As the optimal mask rates may differ depending on the block size L', we adaptively learn the schedule during training. While Kingma et al. (2021) perform variance minimization by isolating a variance term using their squared diffusion loss, this strategy is not directly applicable to our variance estimator in Equation 9 since we seek to reduce variance across random batches in addition to random t_b.

Instead, we optimize parameters β, ω to directly minimize training variance. To limit the computational burden of the optimization, we use the variance of the estimator of the diffusion ELBO as a proxy for the gradient estimator to optimize β, ω: $\min_{\beta,\omega} \text{Var}_{\mathbf{X},t} \left[\mathcal{L}(\theta, \beta, \omega; \mathbf{X}) \right]$.

In Table 2, we show that variance of the diffusion ELBO is correlated with test perplexity. Under a range of "clipped" noise rate distributions, we find that there exists a unique distribution for each block size $L' \in \{4, 16, 128\}$ that minimizes both the variance of the NLL and the test perplexity. We provide experimental details on the optimization procedure in Sec. 6.4.

Table 2: Perplexities (PPLs) and variances of the SAR ELBO $\text{Var}_{\mathbf{X},t}\left[\mathcal{L}_{\text{SAR}}(\mathbf{X}, \theta)\right]$ from training on 60M tokens on LM1B across SAD3-LMs.

L'	$\mathcal{U}[0, .5]$		$\mathcal{U}[.3, .8]$		$\mathcal{U}[.5, 1]$		$\mathcal{U}[0, 1]$	
	PPL	Var. ELBO	PPL	Var. ELBO	PPL	Var. ELBO	PPL	Var. ELBO
128	**31.72**	**1.03**	31.78	1.35	31.92	1.83	31.78	3.80
16	31.27	7.90	**31.19**	**3.62**	31.29	3.63	31.33	7.39
4	29.23	32.68	29.37	10.39	**29.16**	**8.28**	29.23	23.65

6 Experiments

We evaluate SAD3-LMs across standard language modeling benchmarks and demonstrate their ability to generate arbitrary-length sequences on unconditional generation tasks. We train a base SAD3-LM using the maximum context length $L' = L$ for 850K gradient steps and fine-tune under varying L' for 150K gradient steps on the One Billion Words dataset (LM1B) and OpenWebText (OWT). Further details are provided in Suppl F.

To reduce the variance of training on the diffusion ELBO, we adaptively learn the range of masking rates by optimizing parameters β, ω as described in Section 5.3. In practice, we do so using a grid search during every validation step (after $\sim$5K gradient updates) to identify β, ω: $\min_{\beta,\omega} \text{Var}_{\mathbf{X},t}\left[\mathcal{L}(\theta, \beta, \omega; \mathbf{X})\right]$.

During evaluation, we report likelihood under uniformly sampled mask rates (Eq. 7) as in Austin et al. (2021); Sahoo et al. (2024).

6.1 Likelihood Evaluation

On LM1B, SAD3-LMs outperform all prior diffusion methods in Table 3. Compared to MDLM, SAD3-LM achieves up to a 13% improvement in perplexity. We observe a similar trend on OpenWebText, as shown in Table 4.

We also evaluate the ability of SAD3-LMs to generalize to unseen datasets in a zero-shot setting, following the benchmark from (Radford et al., 2019). We evaluate the likelihood of SAD3-LMs trained with OWT on the following benchmark datasets (Radford et al., 2019): LM1B, Lambada (Paperno et al., 2016), AG News (Zhang et al., 2015), and Scientific Papers (Pubmed and Arxiv subsets; (Cohan et al., 2018)). In Table 5, SAD3-LM achieves the best zero-shot perplexity on Pubmed, surpassing AR, and the best perplexity among diffusion models on LM1B.

Table 3: Test perplexities (PPL; $\downarrow$) of models trained for 1M steps on LM1B. †Reported in He et al. (2022). Best diffusion value is bolded.

	Model	Parameters	PPL ($\downarrow$)
Autoregressive	Transformer-X Base (Dai et al., 2019)	0.46B	23.5
	Transformer (Sahoo et al., 2024)	110M	22.83
Diffusion	D3PM (absorb) (Austin et al., 2021)	70M	$\leq$ 82.34
	SEDD (Lou et al., 2023)	110M	$\leq$ 32.71
	MDLM (Sahoo et al., 2024)	110M	$\leq$ 32.03
Semi-autoregressive (Ours)	SAD3-LMs $L' = 16$	110M	$\leq$ 30.67
	$L' = 8$	110M	$\leq$ 29.99
	$L' = 4$	110M	$\leq$ **28.33**

6.2 Qualitative Analysis and Variable-Length Sequence Generation

We also examine qualitatively samples taken from the SAD3-LM model and baselines (AR, MDLM) trained on the OWT dataset; we report samples in E. We observed similar levels of coherence and

Table 4: Test perplexities (PPL; ↓) on OWT for models trained for 262B tokens.

	PPL (↓)
AR Transformer (Sahoo et al., 2024)	17.54
SEDD (Lou et al., 2023)	$\leq$ 24.10
MDLM (Sahoo et al., 2024)	$\leq$ 23.21
SAD3-LMs $L' = 16$	$\leq$ 22.45
$L' = 8$	$\leq$ 21.57
$L' = 4$	$\leq$ **20.65**

Table 5: Zero-shot validation perplexities (↓) of models trained for 524B tokens on OWT. All perplexities for diffusion models are upper bounds.

	LM1B	Lambada	AG News	Pubmed	Arxiv
AR	**51.25**	51.28	**52.09**	49.01	41.73
SEDD	68.20	49.86	62.09	44.53	38.48
MDLM	67.01	**47.52**	61.15	41.89	**37.37**
SAD3-LM $L' = 4$	64.53	49.45	68.19	**41.32**	37.46

diversity across samples from all models; we were not able to distinguish samples coming from different models in a blind test, despite the AR model achieving significantly lower perplexity values.

One key drawback of many existing diffusion language models (e.g,. Austin et al. (2021); Lou et al. (2023)) is that they cannot generate full-length sequences that are longer than the length of the output context chosen at training time. The OWT dataset is useful for examining this limitation, as it contains many documents that are longer than 1,024 tokens in length. Accordingly, while our SEDD samples were clipped at 1,024 tokens, several of our AR and SAD3-LM samples exceed this threshold (Suppl E).

6.3 Comparing to Continuous-State Semi-Autoregressive Discrete Diffusion (SSD-LM)

Han et al. (2022) introduced SSD-LM, an alternative semi-autoregressive formulation that performs Gaussian diffusion over word embeddings, as in Li et al. (2022). Our approach instead applies discrete noise and can be seen as the analogous extension of Austin et al. (2021).

Unlike SAD3-LMs, SSD-LM does not support likelihood estimation. Thus, to compare the models, we generated unconditionally sequences of length 1024 from both models as well as well as from the AR model, all trained on OWT. We measured generative PPL compared against SSD-LM and AR as well as the number of calls to the models during sampling. Compared to alternative semi-autogressive formulations that perform Gaussian diffusion over embeddings (Han et al., 2022), our discrete approach yields samples with improved generative perplexity using an order magnitude fewer generation steps.

Table 6: Semi-AR generative perplexity (Gen. PPL; ↓) and sampling a sequences of $L = 1024$. All models are trained using a context length of 1024 tokens. For SEDD, MDLM, and SAR we set the number of diffusion steps $T = 5000$. SSD-LM uses a block size of 25 with $T = 40K$ diffusion steps ($T = 1000$ per block). However, SAD3-LM ($L' = 16$) uses fewer NFEs by caching the predictions for $\mathbf{x}$ after sampling (see Sahoo et al. (2024)).

	Gen. PPL	NFEs
AR	34.04	1024
SSD-LM	36.54	40K
SAD3-LM	33.56	1015

SAD3-LMs also enjoy efficiency improvements over AR models by using fewer functional evaluations (NFEs), whereas AR generation is fixed to L NFEs. We evaluate our proposed SAR decoding algorithm (Alg. 2) in unconditional generation generate 10 sequences of lengths $L = 1024$ using SAD3-LMs. We show that SAR approaches support generating sequences of arbitrary length, overcoming a key limitation of sampling with diffusion models. In generating 1024 tokens (Table 6), SAD3-LM achieves comparable results while using much fewer NFEs.

6.4 Ablations

We assess the impact of the design choices in our proposed SAR recipes, namely 1) selection of the noise schedule and 2) the efficiency improvement of the proposed training and sampling algorithms relative to a naive implementation.

Selecting Noise Schedules to Reduce Training Variance

Our class of "clipped" masking is the most effective in reducing the perplexity gap compared to linear, cosine, and logarithmic schedules. In Table 7, we train SAD3-LMs under a variety of different schedules for $L' = 4, 16$ and show their impact on test perplexity on LM1B. We find that "clipping" the masking rates during training is the most effective for reducing the variance of the ELBO, which correlates with the perplexity. The ideal "clipped" masking rates, which are learned during training, are specific to the block size and further motivate our optimization.

Table 7: Effect of training under different noise schedules on perplexity on LM1B. All models are finetuned for 50K steps and are evaluated under the linear schedule where $t \sim \mathcal{U}[0, 1]$. For our clipped schedules, we compare the optimized clipping rates for $L' = 4, 16$.

SAD3-LMs	Noise schedule	PPL	Var. ELBO
$L' = 4$	Linear $t \sim \mathcal{U}[0, 1]$	30.18	23.45
	Clipped $t \sim \mathcal{U}[0.45, 0.95]$	**29.21**	**6.24**
	Clipped $t \sim \mathcal{U}[0.3, 0.8]$	29.38	10.33
	Logarithmic	30.36	23.53
	Square root	31.41	26.43
$L' = 16$	Linear $t \sim \mathcal{U}[0, 1]$	31.72	7.62
	Clipped $t \sim \mathcal{U}[0.45, 0.95]$	31.42	3.60
	Clipped linear $t \sim \mathcal{U}[0.3, 0.8]$	**31.12**	**3.58**
	Square	31.43	13.03
	Cosine	31.41	13.00

Efficiency of Training Algorithm

In the training algorithm presented in Section 3.2, we compute $\mathbf{x}_{\text{logit}}$ using two options. We may perform two forward passes through the network (precomputing keys and values for the full sequence $\mathbf{x}$, then computing denoised predictions), or combine these passes by concatenating the two inputs into the same attention kernel.

We find that performing this operation in a single forward pass is often more efficient as we reduce memory bandwidth bottlenecks by leveraging efficient, pre-exisiting flash attention kernels Dao et al. (2022). Instead of paying the cost of 2 passes through the network, we only pay the cost of a more expensive attention operation. Empirically we see that this approach has >2x speed-up relative to performing two forward passes.

7 Discussion, Prior Work, and Conclusion

Comparison to D3PM Semi-autoregressive diffusion builds off D3PM (Austin et al., 2021) and applies it to each auto-regressive conditional. We improve over D3PM in three ways: (1) we provide a way of extending D3PM beyond fixed sequence lengths; (2) we study the perplexity gap of D3PM and AR models, identify gradient variance as a contributor, and design variance-minimizing schedules;

(3) we improve over the perplexity of D3PM models. While (1) involves modifying D3PM to make it semi-autoregressive, (2) is applicable to vanilla D3PM.

Comparison to MDLM Masked SAD3-LMs further make use of the perplexity-enhancing improvements in MDLM (Sahoo et al., 2024). We also build upon MDLM: (1) while Sahoo et al. (2024) points out that their ELBO is invariant to the noise schedule, we show that the noise schedule has a significant effect on gradient variance; (2) we push the state-of-the-art in perplexity beyond MDLM. Note that our perplexity improvements stem not only from semi-autoregression, but also from optimized schedules, and could enhance standard MDLM models.

Comparison to Autoregressive-Diffusion Models Han et al. (2022) introduced an alternative semi-autoregressive formulation that performs Gaussian diffusion over word embeddings, as in Li et al. (2022). Our approach instead applies discrete noise as in Austin et al. (2021), and features notable improvements: (1) tractable likelihood estimates enabling principled evaluation; (2) faster generation, as our number of model calls is bounded by the number of generated tokens, while SSD-LM performs orders of magnitude more calls; (3) significantly improved performance, measured by perplexity relative to existing models as well as generative perplexity relative to samples from both SSD-LM.

AR-Diffusion (Wu et al., 2023) is a variant of SSD-LM that uses a noise schedule which encourages sampling in a left-to-right manner. However, they sacrifice parallelism by assigning unique, per-token timesteps. Autoregressive Diffusion Models (Hoogeboom et al., 2021) generalize order-agnostic autoregressive models and discrete diffusion. In contrast, our approach is not order-agnostic as blocks are modeled autoregressively. TimeGrad (Rasul et al., 2021) performs time series forecasting under a diffusion objective conditioned on observations at the previous timestep. However, their framework is autoregressive so it is not amenable to parallel sampling or controllability.

Comparison to Jacobi Decoding Jacobi decoding (Santilli et al., 2023) is an AR inference technique that iteratively refines a random sequence and supports parallel generation of token blocks. However, Santilli et al. (2023) preserve causal masking from AR whereas SAD3-LMs may leverage more context by attending to tokens within a block and in previous blocks. Whereas Jacobi decoding uses uniform noise, SAD3-LMs use masking which has been shown to be superior in language modeling (Austin et al., 2021; Lou et al., 2023). Consistency LLMs (Kou et al., 2024) extend Jacobi decoding to include a fine-tuning objective. In contrast, SAD3-LMs may leverage clean conditional context $\mathbf{x}^{<b}$ to enhance predictions.

Limitations Training SAD3-LMs requires a custom procedure that is more expensive than regular diffusion training. We propose a vectorized implementation that keeps training speed within <2x of diffusion training speed; in our experiments, we also pre-train with a standard diffusion loss to further reduce the speed gap. Additionally, SAD3-LMs generate blocks sequentially, hence may face the same speed and controllability constraints as AR, especially when blocks are small. The optimal block size for control is task specific, and optimal blocks for speed depend on the parallelization capabilities of inferencing hardware (e.g,. flops vs. memory throughput) and serving batch size.

Conclusion This work explores semi-autoregressive diffusion and is motivated by two problems with existing discrete diffusion: the need to generate arbitrary-length sequences and the perplexity gap to discrete models. We introduce SAD3-LMs, which represent a semi-autoregressive extension of the D3PM framework (Austin et al., 2021), and leverage a specialized training algorithm and custom noise schedules that further improve performance. We observe that in addition to being able to generate long-form documents, these models also improve perplexity, setting a new state-of-the-art among discrete diffusion models.

REFERENCES

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.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In *International Conference on Machine Learning*, pp. 1276–1301. PMLR, 2023.

Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and Tony Robinson. One billion word benchmark for measuring progress in statistical language modeling, 2014.

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. *Proceedings of the 2018 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 2 (Short Papers)*, 2018. doi: 10.18653/v1/n18-2097. URL http://dx.doi.org/10.18653/v1/n18-2097.

Zihang Dai, Zhilin Yang, Yiming Yang, Jaime Carbonell, Quoc V Le, and Ruslan Salakhutdinov. Transformer-xl: Attentive language models beyond a fixed-length context. *arXiv preprint arXiv:1901.02860*, 2019.

Tri Dao, Dan Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. Flashattention: Fast and memory-efficient exact attention with io-awareness. *Advances in Neural Information Processing Systems*, 35:16344–16359, 2022.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis, 2021. URL https://arxiv.org/abs/2105.05233.

Aaron Gokaslan, Vanya Cohen, Ellie Pavlick, and Stefanie Tellex. Openwebtext corpus. http://Skylion007.github.io/OpenWebTextCorpus, 2019.

Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models. *Advances in Neural Information Processing Systems*, 36, 2024.

Agrim Gupta, Lijun Yu, Kihyuk Sohn, Xiuye Gu, Meera Hahn, Li Fei-Fei, Irfan Essa, Lu Jiang, and José Lezama. Photorealistic video generation with diffusion models, 2023. URL https://arxiv.org/abs/2312.06662.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. Ssd-lm: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. *arXiv preprint arXiv:2210.17432*, 2022.

Xiaochuang Han, Sachin Kumar, Yulia Tsvetkov, and Marjan Ghazvininejad. David helps goliath: Inference-time collaboration between small specialized and large general diffusion lms. *arXiv preprint arXiv:2305.14771*, 2023.

Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. *arXiv preprint arXiv:2211.15029*, 2022.

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

Jonathan Ho, Tim Salimans, Alexey Gritsenko, William Chan, Mohammad Norouzi, and David J Fleet. Video diffusion models. *arXiv:2204.03458*, 2022.

Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg, and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.

Diederik Kingma, Tim Salimans, Ben Poole, and Jonathan Ho. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Siqi Kou, Lanxiang Hu, Zhezhi He, Zhijie Deng, and Hao Zhang. CLLMs: Consistency large language models. In *Forty-first International Conference on Machine Learning*, 2024. URL https://openreview.net/forum?id=8uzBOVmh8H.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in Neural Information Processing Systems*, 35: 4328–4343, 2022.

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

Denis Paperno, Germán Kruszewski, Angeliki Lazaridou, Ngoc Quan Pham, Raffaella Bernardi, Sandro Pezzelle, Marco Baroni, Gemma Boleda, and Raquel Fernandez. The LAMBADA dataset: Word prediction requiring a broad discourse context. In *Proceedings of the 54th Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. URL http://www.aclweb.org/anthology/P16-1144.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF International Conference on Computer Vision*, pp. 4195–4205, 2023.

Alec Radford, Jeffrey Wu, Rewon Child, David Luan, Dario Amodei, Ilya Sutskever, et al. Language models are unsupervised multitask learners. *OpenAI blog*, 1(8):9, 2019.

Kashif Rasul, Calvin Seward, Ingmar Schuster, and Roland Vollgraf. Autoregressive denoising diffusion models for multivariate probabilistic time series forecasting. In Marina Meila and Tong Zhang (eds.), *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pp. 8857–8868. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/rasul21a.html.

Subham Sekhar Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin T Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *arXiv preprint arXiv:2406.07524*, 2024.

Andrea Santilli, Silvio Severino, Emilian Postolache, Valentino Maiorca, Michele Mancusi, Riccardo Marin, and Emanuele Rodola. Accelerating transformer inference for translation via parallel decoding. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki (eds.), *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pp. 12336–12355, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.689. URL https://aclanthology.org/2023.acl-long.689.

Jascha Sohl-Dickstein, Eric Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. In *International conference on machine learning*, pp. 2256–2265. PMLR, 2015.

Yang Song and Stefano Ermon. Generative modeling by estimating gradients of the data distribution. *Advances in neural information processing systems*, 32, 2019.

Jianlin Su, Yu Lu, Shengfeng Pan, Ahmed Murtadha, Bo Wen, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *arXiv preprint arXiv:2104.09864*, 2021.

Haoran Sun, Lijun Yu, Bo Dai, Dale Schuurmans, and Hanjun Dai. Score-based continuous-time discrete diffusion models. *arXiv preprint arXiv:2211.16750*, 2022.

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.

Tong Wu, Zhihao Fan, Xiao Liu, Hai-Tao Zheng, Yeyun Gong, yelong shen, Jian Jiao, Juntao Li, zhongyu wei, Jian Guo, Nan Duan, and Weizhu Chen. AR-diffusion: Auto-regressive diffusion model for text generation. In *Thirty-seventh Conference on Neural Information Processing Systems*, 2023. URL https://openreview.net/forum?id=0EG6qUQ4xE.

Xiang Zhang, Junbo Jake Zhao, and Yann LeCun. Character-level convolutional networks for text classification. In *NIPS*, 2015.

CONTENTS

A SAR ELBO Derivation

Below, we provide the ELBO for the SAR parameterization.

$$
\begin{aligned}
\log p(\mathbf{x}) &= \sum_{b=1}^{L/L'} \log p(\mathbf{x}^b|\mathbf{x}^{<b}) \\
&= \sum_{b=1}^{L/L'} \log \mathbb{E}_q \frac{p(\mathbf{x}^b_{0:T}|\mathbf{x}^{<b})}{q(\mathbf{x}^b_{1:T}|\mathbf{x}^b_0)} \\
&= \sum_{b=1}^{L/L'} \log \mathbb{E}_q \frac{p(\mathbf{x}^b_T|\mathbf{x}^{<b}) \prod_{t=1}^T p(\mathbf{x}^b_{t-1}|\mathbf{x}^b_t, \mathbf{x}^{<b})}{\prod_{t=1}^T q(\mathbf{x}^b_t|, x^b_{t-1})} \\
&\geq \sum_{b=1}^{L/L'} \Bigg[\mathbb{E}_q \log p_\theta(\mathbf{x}^b|\mathbf{x}^b_1, \mathbf{x}^{<b}) \\
&\quad - \mathbb{E}_{t\in\{2,\dots T\}} \mathbb{E}_q T \mathrm{D}_{KL}\left(q(\mathbf{x}^b_{t-1}|, \mathbf{x}^b_t, \mathbf{x}^b_0) \parallel p(\mathbf{x}^b_{t-1}|\mathbf{x}^b_t, \mathbf{x}^{<b})\right) \\
&\quad - \mathrm{D}_{KL}\left(q(\mathbf{x}^b_T|\mathbf{x}^b) \parallel p(\mathbf{x}^b_T)\right) \Bigg] \\
&= -\sum_{b=1}^{L/L'} \mathbb{E}_{t\in\{2,\dots T\}} \mathbb{E}_q \left[T \log \frac{q(\mathbf{x}^b_{t-1}|, \mathbf{x}^b_t, \mathbf{x}^b)}{p(\mathbf{x}^b_{t-1}|\mathbf{x}^b_t, \mathbf{x}^{<b})} \right]
\end{aligned}
$$

Following Sahoo et al. (2024), we simplify the expression for the ELBO in the case of absorbing state diffusion. In particular, we leverage the fact that $\mathbf{x}_s$ the reverse posterior $q(\mathbf{x}_s|\mathbf{x}_t, \mathbf{x})$ in Eq. 2 may take on two states: $\mathbf{x}_s \in \{\mathbf{x}, \mathbf{m}\}$. See the full derivation in Sahoo et al. (2024).

$$
\begin{aligned}
&= q(\mathbf{x}_s = \mathbf{x}|\mathbf{x}_t = \mathbf{m}, \mathbf{x}) \log \frac{q(\mathbf{x}_s = \mathbf{x}|\mathbf{x}_t = \mathbf{m}, \mathbf{x})}{p_\theta(\mathbf{x}_s = \mathbf{x}|\mathbf{x}_t = \mathbf{m}, \mathbf{x}^{<b})} \\
&+ q(\mathbf{x}_s = \mathbf{m}|\mathbf{x}_t = \mathbf{m}, \mathbf{x}) \log \frac{q(\mathbf{x}_s = \mathbf{m}|\mathbf{x}_t = \mathbf{m}, \mathbf{x})}{p_\theta(\mathbf{x}_s = \mathbf{m}|\mathbf{x}_t = \mathbf{m}, \mathbf{x}^{<b})} \\
&= T \Bigg[\frac{\alpha_s - \alpha_t}{1 - \alpha_t} \log \frac{\alpha_t p_\theta(\mathbf{x}^b = \mathbf{m} \mid \mathbf{x}^b_t, \mathbf{x}^{<b}) + (1 - \alpha_t)}{(1 - \alpha_t) p_\theta(\mathbf{x}^b = \mathbf{x} \mid \mathbf{x}^b_t, \mathbf{x}^{<b})} \\
&+ \frac{1 - \alpha_s}{1 - \alpha_t} \log \frac{(1 - \alpha_s)(\alpha_t p_\theta(\mathbf{x}^b = \mathbf{m} \mid \mathbf{x}^b_t, \mathbf{x}^{<b}) + (1 - \alpha_t))}{(1 - \alpha_t)(\alpha_s p_\theta(\mathbf{x}^b = \mathbf{m} \mid \mathbf{x}^b_t, \mathbf{x}^{<b}) + (1 - \alpha_s))} \Bigg] \\
&= \sum_{b=1}^{L/L'} \mathbb{E}_{t\sim(0,1]} \mathbb{E}_q T \left[\frac{\alpha_s - \alpha_t}{1 - \alpha_t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}^b_t, \mathbf{x}^{<b}) \right] \\
&= \sum_{b=1}^{L/L'} \mathbb{E}_{t\sim(0,1]} \mathbb{E}_q \left[\frac{\alpha'_t}{1 - \alpha_t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}^b_t, \mathbf{x}^{<b}) \right]
\end{aligned}
$$

B Recovering the NLL from the SAR NELBO

The SAR NELBO is equivalent to the AR NLL when modeling a single token:

$$
-\log p(\mathbf{x}) \leq \sum_{b=1}^{L} \mathbb{E}_{t\sim[0,1]} \mathbb{E}_q \left[\frac{\alpha'_t}{1 - \alpha_t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}^b_t, \mathbf{x}^{<b}) \right] \tag{10}
$$

$$
= -\sum_{b=1}^{L} \mathbb{E}_{t\sim[0,1]} \mathbb{E}_q \left[\frac{1}{t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}^b_t, \mathbf{x}^{<b}) \right] \qquad \text{since } \alpha'_t = -1 \text{ and } \alpha_t = 1 - t \tag{11}
$$

$$= -\sum_{b=1}^{L} \mathbb{E}_{t\sim[0,1]} \left[\frac{q(\mathbf{x}_t^b = \mathbf{m}|\mathbf{x}^b)}{t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b = \mathbf{m}, \mathbf{x}^{<b}) \right. \tag{12}$$

$$\left. + \frac{q(\mathbf{x}_t^b = \mathbf{x}|\mathbf{x}^b)}{t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b = \mathbf{x}, \mathbf{x}^{<b}) \right] \tag{13}$$

We follow the parameterization from Sahoo et al. (2024) for masked diffusion and set the posterior $\log p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b = \mathbf{x}, \mathbf{x}^{<b}) = 0$, since an unmasked token will never transition to a different state in the reverse process.

$$-\log p(\mathbf{x}) \leq -\sum_{b=1}^{L} \mathbb{E}_{t\sim[0,1]} \frac{q(\mathbf{x}_t^b = \mathbf{m}|\mathbf{x}^b)}{t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b = \mathbf{m}, \mathbf{x}^{<b}) \tag{14}$$

$$= -\sum_{b=1}^{L} \mathbb{E}_{t\sim[0,1]} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b = \mathbf{m}, \mathbf{x}^{<b}) \qquad \text{since } q(\mathbf{x}_t = \mathbf{m}|\mathbf{x}) = t \tag{15}$$

$$= -\sum_{b=1}^{L} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}^b = \mathbf{m}, \mathbf{x}^{<b}) \tag{16}$$

For single-token generation ($L' = 1$) and under sampling $t \sim \mathcal{U}[1, 1]$, we recover the autoregressive NLL:

$$-\log p(\mathbf{x}) \leq \sum_{b=1}^{L} \mathbb{E}_{t\sim[1,1]} \mathbb{E}_q \left[\frac{\alpha_t'}{1-\alpha_t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b}) \right] \tag{17}$$

$$= \sum_{b=1}^{L} \mathbb{E}_{t\sim[1,1]} \mathbb{E}_q \left[\frac{-1}{1-\alpha_t} \log p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b}) \right] \quad \text{since } \alpha_t' = -1 \text{ for } \alpha_t = 1 - t \tag{18}$$

$$= \sum_{b=1}^{L} -\log p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b}) \qquad \text{since } \mathbb{E}_{t\sim[1,1]} \frac{1}{1-\alpha_t} = 1 \tag{19}$$

C TIGHTNESS OF THE SAR ELBO

For block sizes $L \leq K \leq 1$, we show that -$\log p(\mathbf{x}) \leq \mathcal{L}^{K+1} \leq \mathcal{L}^K$. Consider $K = 1$, where we recover the autoregressive NLL (Suppl B):

$$\mathcal{L}^1 = \sum_{b=1}^{L} -\log \mathbb{E}_{t\sim[0,1]} \mathbb{E}_q \frac{\alpha_t'}{1-\alpha_t} p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b}) \tag{20}$$

$$= \sum_{b=1}^{L} -\log p_\theta(\mathbf{x}^b \mid x_t^b = \mathbf{m}, \mathbf{x}^{<b}) \tag{21}$$

$$\tag{22}$$

Consider the ELBO for $K = 2$ blocks:

$$\mathcal{L}^2 = \sum_{b=1}^{L/2} \log \mathbb{E}_{t\sim[0,1]} \mathbb{E}_q \frac{\alpha_t'}{1-\alpha_t} p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b}) \tag{23}$$

We show that $\mathcal{L}^2 \geq \mathcal{L}^1$, and this holds for all $L \leq K \leq 1$ by induction.

$$\sum_{b=1}^{L} -\log\langle p_\theta(\mathbf{x}_t^b = \mathbf{m}|\mathbf{x}^{<b}), \mathbf{x}^b\rangle \tag{24}$$

$$= \sum_{b=1}^{L/2} -\log \mathbb{E}_{t\sim[0,1]} \mathbb{E}_q \frac{1}{1-\alpha_t} p_\theta(\mathbf{x}^b \mid \mathbf{x}_t^b, \mathbf{x}^{<b}) \tag{25}$$

$$
\begin{aligned}
&= \sum_{b=1}^{L/2} -\log \mathbb{E}_{t\sim[0,1]} \mathbb{E}_q \prod_{i=1}^{2} \frac{1}{1-\alpha_t} p_\theta(\mathbf{x}^{i,b} \mid \mathbf{x}_t^{i,b}, \mathbf{x}^{<b}) \\
&= \sum_{b=1}^{L/2} -\log \prod_{i=1}^{2} \mathbb{E}_{t\sim[0,1]} \mathbb{E}_q \frac{1}{1-\alpha_t} p_\theta(\mathbf{x}^{i,b} \mid \mathbf{x}_t^{i,b}, \mathbf{x}^{<b}) \\
&\leq \sum_{b=1}^{L/2} \sum_{i=1}^{2} -\log \mathbb{E}_{t\sim[0,1]} \mathbb{E}_q \frac{1}{1-\alpha_t} p_\theta(\mathbf{x}^{i,b} \mid \mathbf{x}_t^{i,b}, \mathbf{x}^{<b}) \qquad (26)
\end{aligned}
$$

D SPECIALIZED ATTENTION MASKS FOR SAR MODELING

We aim to model conditional probabilities $\log p(\mathbf{x}^b|\mathbf{x}^{<b})$ for $b \in [1, B]$ simultaneously by designing an efficient training algorithm with our transformer backbone. However, modeling all B conditonal terms requires processing both the noised sequence $\mathbf{x}_t$ and the clean sequence $\mathbf{x}$ (which captures conditional context).

Rather than calling the denoising network B times, we process both sequences simultaneously by concatenating them $\mathbf{x}_{full} \leftarrow \mathbf{x}_t \oplus \mathbf{x}$ as input to a transformer. We update this sequence $\mathbf{x}_{full}$ of length $2l$ using a custom attention mask $\mathcal{M}(L, B) \in \{0, 1\}^{2L\times 2L}$ for efficient training.

This attention mask is comprised of 4 $L \times L$ smaller attention masks:

$$
\textsc{Mask}(L, B) = \begin{bmatrix} \mathcal{M}_{BD} & \mathcal{M}_{OBC} \\ \mathbf{0} & \mathcal{M}_{BC} \end{bmatrix}
$$

where $\mathcal{M}_{BD}$ and $\mathcal{M}_{OBC}$ are used to update the representation of $\mathbf{x}_t$ and $\mathcal{M}_{BC}$ is used to update the representation of $\mathbf{x}$. We define these masks as follows:

- $\mathcal{M}_{BD}$ (Block-diagonal mask): Self-attention mask within noised blocks $\mathbf{x}_t^b$

$$
\mathcal{M}_{BD} = (m_{i,j})_{L\times L} = \begin{cases} 1 & \text{if } i, j \text{ are in the same block} \\ 0 & \text{otherwise} \end{cases}
$$

- $\mathcal{M}_{OBC}$ (Offest block-causal mask): Cross-attention mask for conditional context $\mathbf{x}^{<b}$

$$
\mathcal{M}_{OBC} = (m_{i,j})_{L\times L} = \begin{cases} 1 & \text{if } i \text{ belongs in a block after } j \\ 0 & \text{otherwise} \end{cases}
$$

- $\mathcal{M}_{BC}$ (Block-causal mask): Attention mask for updating $\mathbf{x}^b$

$$
\mathcal{M}_{BC} = (m_{i,j})_{L'\times L} = \begin{cases} 1 & \text{if } j \text{ is not in a block after } i \\ 0 & \text{otherwise} \end{cases}
$$

We visualize an example attention mask for $L = 6$ and $L' = 2$:

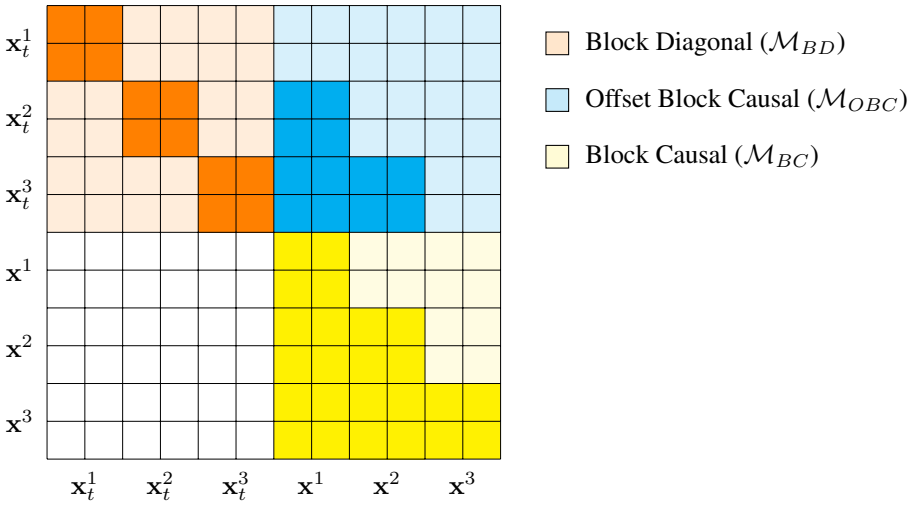

Figure 2: Example of Specialized Attention Mask

E SAMPLES

<|endoftext|>“unacceptable.” Citing none of the conflicts in terms of military casualties, Trump said he was confident that no other country needs weapons such as bombs. “We also know we can kill and save lives,” he said in his remarks in Havana.
“And those are where we protect our citizens from terror attacks.”
“My real concern is not just that capability of the people. It is where we train our fighters and it would be unacceptable to eliminate that.”
Trump, who says he will put his proposals to voters at a later date, also wants a “clean” veto in Congress — a proposal he is unlikely to be able to unilaterally veto.
“We seem to have taken a step back in the number of attacks,” Trump said of Obama’s eight years in office.
“I think the attacks have gone down further, as well. You think of those two words. But, once you put up something like that, I think the answer is that.”
Repeting Hillary’s claim that “Putin’s God really wants us to attack ISIS,” Trump said, “They use tunnels to get their oil.”
“It’s because ISIS had oil. You know, Iraq. They had these tunnels up in Iraq, and here’s the man said ISIS was, ‘You get their oil, and then you get their oil.’ For people who don’t believe that the president could actually use that account, and to have flown out over a 4,000 mile — or how many of them? Well, if you look at membership of the Islamic State. You know Syria, if you put their Syria fighting back in 2016, or if they were today, it would be a million or so, but ISIS was prior to 10. They have experienced 500. And who are they here? What are they doing?”
But Trump said the answer is, “The death of ISIS.”
“That’s why I said that. I think you understand it. I know, I don’t know, you look at their own war numbers. It doesn’t tell them much, because they’re not afraid. They certainly know.”
“At some point, there’s no,” Trump said to applause.
He added, “Let’s say it true. Give it a thought to it. Now let’s tear it down a little more sharply. And we want to think about it — tear it up and leave a little more room.”
Trump tried to tone down the question, adding that he would have to say more to describe it.
“For me, I think what we have said is ‘moderate the opposition.’ Look, and I say, no we can’t. And this is a civil opposition,” Trump said. “Let’s see how we do it. How will it be. And will it be OK?”
What Trump did not say, he said he already knew what he would do.
RELATED WATCH
“If you got the proper thing, you see Russia going away, and you win your transition,” he said, during a speech in the Valley.
In fact, what Trump did not address said was the immediate danger of disruption going overseas.
“The easy part to feel is,” Trump noted. “I’ve heard people destroy things abroad to make some of it worse. That scares me.”
…Then they, they took over this country, the people loved this. What will it look like, says Trump, and while he plans for it all to be OK, duh. Under the Russian attacks, his nature to their establishments isn’t already ruined.
“I’ve just said. I’m sure you’ve agreed to be a part of the solution of this, man. And some of you would like to do so,” he added
“The people love it because they say you can’t deal with something like that.”
“Nice, peaceful,” he said. “It all is working that way.”
“And I want you to be watching this again,” Putin noted.
“You all see this part. If you were surprised by the decision that Donald Trump has made this last week, I ask you to know that my desire and my support to continue to fight against terrorism from you. I’m pleased all of you would gladly accept our support,” said Putin, in his Mr. Trump stance. He said this ISIS war is important to him and now he is telling you young men to “go forward – that is a great bit of advice, I think you will love, and as you know, you’re going to be in every aspect of the US end of this operation.”
Putin and Trump added, “I believe you’re giving me the best of fighters and I know you personally very well, I am going to make a better into a better.”
“This, be good for you. Good for them. Good for them.” world leader said
Many praised, good faith, and thanks to Mr.Trump’s plans. European leaders were asked what would be nice, Trump is doing if he got the next post. He said what he does is good, but the act of doing what he’s not concerned about is whatwhat he’s [very well is] doing.” Interestingly, Trump responded to that that he’s really worried only because it’s working so well.
Everybody’s looking at fellow colleagues, it only goes to see who’s the president, how we’ll be going, is whathe is well, because it’s obvious how wellwill only make worse thing that’s happening, he said. And he is doing well and we are too well, he is pretty just he’s like, let him go of the turmoil and let it go fine with the two, let these things fine with the two.’ ‘They’re good. They’re good,’ he said ‘find he’s good and of course – be the – and deal with it, take it, will kill the.<|endoftext|>

Figure 3: Samples from SAD3-LM for $L' = 16$ sample length $L = 1408$ and $T = 5000$ diffusion steps. (trained with a context length of $L = 1024$). The generative perplexity under GPT2-Large is 22.50.

<|endoftext|> one or more of the votes in the top two of the popular vote in each state? The fewer the more votes the better. Although Republican Trump votes aren't exactly dead, they got worse fast.
Two-candidate Dead Favorites in 2016
Matt Latimer turned out to be sorely underestimated on Tuesday as he discovered that there was another candidate in the race that not only angered his party, but that it was running a double-standard for voters. John Kasich is far from a perfect Ryan-Ryan pick, but he is the clear, true vehicle for Trump's message and, indeed, supported the Republican nominee.
Alec Baldwin's Day With Trump's Friend Alec Baldwin's Day With Trump got him to voice what he does get to express himself through a two limb swing next to Hillary Clinton.
Trump's praise of Gillian Flynn has propelled him to his third nominee in a row. Flynn has made calls for "tough" action to curb the mass illegal immigration of illegal aliens. But Democrats know that Flynn comes to Trump with his own baggage. They know that Flynn supports the president while remaining silent about Trump's many bad deeds.
Gracie Brzezinski, in Maddow's article, offered a great opportunity to answer some of those questions that Hillary and her future secretary of State may have to answer. By the way, Maddow took up a legal question to her from an Obama supporter a few weeks ago, an incident that resulted in Brzezinski being asked a second time today. Let me introduce you to Gracie. . . ..and one last bit before you ask my own question:
"Gracie Brzezinski, in You Led the Future to v Outrage Trump, Trump to Sit Up on First Hype? Brynn Tannehill as Donald Trump – It's come to the end of the Anthony Scaramucci era for tin sake of a bottle of wine and a kickass commentary program on FOX that as I address you from Miami ahead of tomorrow's historic loss in Wisconsin it bears repeating: Tariffs Will Go Up On Your Generals And Presidents Citizens.
And how much one dollar 7/12/2015, if not 10/6/2015, of the next ten years will Americans' corn and soy subsidies be subject to Trump shock-art? Clinton was on a whole lot more taxing when Obama ran and the I Regulatory Tax Relief Act of 2005 is a complete wind up on the brakes on illegal immigration. But normal Americans would forever be looking to thee for proof that decent businesspeople are still very much endorsing Trump.
Keep fighting. We want to hear from you in the farm-to-fork area (right outside Congress!) on Thursday afternoon – Brzezinski would be digging in her grave- Kathryn. But keep voting for Trump, seriously Hillary and the return of Chuck Schumer to Washington D.C. – and let's just say you've been unwatchable by the last week of the ill-fated, mean-spirited 1960's!
Today, I may be getting a little too fuming, but I'll just make sure to never miss a beat for Adele Tucker in that bright light that is reality television! Thank you for inspiring me this holiday season and come back in some years for more bonkers ingratiating tales of "if you don't rock the boat coming back, never again."
And look what sports nuts happen to be like when Donald Trump is president!
– Andrew Hageldon and Sam Koshan
Advertisements
Dr Ben Carson is accusing vice presidential candidates on the Republican presidential ticket of having "no concept of value to the American people." In the days after the first debate, Trump, the first Presbyterian candidate to win in the 2016 general election, faced an energized firestorm of criticism over how economic inequality is impacting the health of the middle class and making America less safe. Carson also tangled with Ted Cruz, the Texas senator who was viewed by almost all of those listening to the GOP debate as the only honest candidate on stage.
But as the dust settled, Carson was pushed out of the race by hand, and Trump has fired questions at one of the more accessible Republican candidates, the head of a state earmarked heavily for the Republican nomination and Fox News host John Kasich, who is also mentioned in the polls as a probable Republican primary nominee.
Read more
But as the dust settled, Carson was pushed out of the race by hand, and Trump has fired questions at one of the most accessible Republican candidates, the head of a state earmarked heavily for the Republican nomination and Fox News host John Kasich, who is also mentioned in the polls as a probable Republican primary nominee.
Carson is accusing vice presidential candidates on the Republican presidential ticket of having <|endoftext|>

Figure 4: Samples from an AR model (Sahoo et al., 2024) with sample length $L = 4540$ (trained with a context length of $L = 1024$). The generative perplexity of this sequence is 30.42.

<|endoftext|> Marx cannot say, aside from his philosophy of time, that the class structure in the economic organization he says, is a contradiction. Those two things are progressive forces, which exist beyond the contradictions. Is this a serious statement about that?
Grier: Yes. Really because the system today, the system that creates the contradictions is the capitalist system. And we are not going to discuss these old contradictions.
When it comes to production, the conditions for people only get a pay are where to pay and how to produce them. So that's why the contradictions are so progressive, one of which is there is this contradiction. So how can people get an income? why can we have people get an income? There's nobody who can say, if you have the proletariat, you can speak of the working class, so we are only discussing a working-class or a minority.
In fact, when Karl Marx developed the idea of communism, he developed the revolutionary idea that revolution would oppress people and take power to the oppressed people. The idea is that the capitalist class would be an oppressed class, like the Marx and Lenin, who say this is the take of an enslaved class, and they would be 'made out of abstract labour.'
If you say, Marx uses this as an example, that the process for a revolutionary is not that of a capitalist. There's no way that you're going to go without Marx. You can still go without Marx and be part of society that develops, and a communist party who is going to enrich the experience of the old or the new. So that's really revolution. No matter how different things are in different ways.
The way to say this is because money is the easiest thing that capitalism allows you to work. But one day you give up this power of money and so to start in the field, and to work to it to sell the goods that the masses of people demands of that money. And money gives the power to sell it to a wage worker over that, and so with the rule of class solidarity.
This is why Marx and Lenin and such else expressed why there is a working class that the working class makes money from, and another very important fact, it gives people the power to work off of this money... So you can work out, not only for the capitalists, but for all people.
But the other side, that when you start with the power of money, money dominates, and that's part of the socialist argument that there's a market along. But under socialism, we'll be in the free market, and the voice, "I am going to be used to where I have to keep all of my money." So, "Under Labour, I have to collect all of my money and work based on the power of money." This is what a socialist does. That is, it takes money from the authority of money. So you can collect it, and there's no bureaucracy or money power.
You also get some benefit of spontaneous, spontaneous mobilization without revolution if people who, for example, they want to borrow money because it's hard work. They get free money, by the poor and by the economy and those who are willing, who say, "Oh, I've started a movement, the Party, and I'm a socialist. I can get some money from that movement, but I'm paying a tax, and I want to get free money, too." All of these things happen.
This is why particularly, government goes to this defensive position where you have a specialist who speaks out at a public demonstration and argues among other things, that while socialism is good, there's no market for both economies here and elsewhere. And this means we have a private exchange. Where you have to enter into a human exchange where people get free money.
And as long as they make money, and work, they're going to get some money too. And if you have a human exchange, people, everybody get free money. So everyone has got a market. The production of all this free money is being paid for by others. This is the real society, and let's say the bourgeoisie gives the market what it wants in clear up the wrong problem with free money, money surplus, so that production would flow in free money. And this shows us that that process of world revolution is completely influenced by individual desires. I mean, you want to find a private pool of money and at the moment you want to go to some kind of country. But then you try to justify the existence of the society, because it's the product of a social demand. Then the demand for free money, there is a social demand for a socialist society, and that is really important to overcome. <|endoftext|>

Figure 5: Sample from MDLM (Sahoo et al., 2024) of length $L = 1024$ and $T = 5000$ diffusion steps. (trained with a context length of $L = 1024$). The generative perplexity under GPT2-Large is 24.95.

F Experimental details

We closely follow the same training and evaluation setup as used by Sahoo et al. (2024). We conduct experiments on two datasets: The One Billion Word Dataset (LM1B; Chelba et al. (2014)) and OpenWebText(OWT; Gokaslan et al. (2019)). Models trained on LM1B use the `bert-base-uncased` tokenizer and a context length of 128. We report perplexities on the test split of LM1B. Models trained on OWT use the `GPT2` tokenizer Radford et al. (2019) and a context length of 1024.

However, in preparing LM1B examples, Sahoo et al. (2024) pad each example to fit in the context length. Since most examples consist of only a single sentence, semi-autoregressive modeling for larger block sizes $L' > 4$ would not be useful for training. Instead, we concatenate and wrap sequences to a length of 128. As a result, we retrain our autoregressive baseline, SEDD, and MDLM on LM1B with wrapping. Similarly for OWT, we do not pad or truncate sequences, but concatenate them and wrap them to a length of 1024 similar to LM1B. Since OWT does not have a validation split, we leave the last 100k documents for validation.

The model architecture augments the diffusion transformer (Peebles & Xie, 2023) with rotary positional embeddings (Su et al., 2021). We parameterize our autoregressive baselines, SEDD, MDLM, and SAD3-LMs with a transformer architecutre from Sahoo et al. (2024) that use 12 layers, a hidden dimension of 768, and 128 attention heads. We do not include timestep conditioning as Sahoo et al. (2024) show it does not affect performance.

We train a base SAD3-LM using the maximum context length $L' = L$ for 850K gradient steps and fine-tune under varying L' for 150K gradient steps on the One Billion Words dataset (LM1B) and OpenWebText (OWT). This translates to 33B tokens on LM1B and 262B tokens on OWT.

We use 3090, A5000, A6000, and A100 GPUs. We train for 73 epochs on LM1B and 59 epochs on OWT (all correspond to 1M gradient updates with a batch size of 512). Training SAD3-LMs on LM1B takes 1.5 days on 4 A5000s and OWT takes 4.5 days on 8 A100s.