# CoMD: Coherent Masked Diffusion

**Anonymous authors**
Paper under double-blind review

## Abstract

Masked language models (MLMs) have shown promise in natural language processing, but struggle with generating coherent and coherent-sounding text. In this work, we present Coherent Masked Diffusion (CoMD), a novel framework that extends Masked Language Diffusion to more efficiently and more effectively learn coherent and incoherent language. CoMD is built on Masked Language Diffusion (MLD), a recently proposed framework that models text generation as an inverse denoising diffusion process. Unlike MLD, CoMD uses a fixed mask matrix that is independent of the masked-out token and optimizes the probability of coherent generations with a novel coherent loss term without requiring additional samples per training step. Additionally, CoMD uses a variable time parameter to guide the coherent probability towards the ground truth coherent probability. Both inference and training computation are constant with respect to the length of the text. Empirically, CoMD outperforms previous methods on multiple coherent benchmarks. Furthermore, CoMD achieves an inference speedup of 7.3x and 10.5x over MLD and MDLM, respectively, and is significantly more compute and parameter efficient than autoregressive models.

## 1 Introduction

In recent years, diffusion models have shown great success in a variety of domains including image generation (Dhariwal and Nichol, 2021), video generation (Ho et al., 2022) and audio modeling (Chen et al., 2021). Compared to prior methods, such as variational autoencoders (Sohl-Dickstein et al., 2015), diffusion models have demonstrated superior performance. Notably, Masked Diffusion Models (MDLMs) (Austin et al., 2021; Shih et al., 2022; Austin et al., 2021; Liu et al., 2024) have shown great potential in the discrete domain of natural language, which traditional diffusion models (Ho et al., 2020; Song et al., 2021) cannot model. Briefly, MDLMs map each token to a binary random variable (a 'one-hot' vector) and treat the task of language generation as an inverse diffusion process to denoise and transform a masked language model (MLM) (Devlin et al., 2019) into a standard autoregressive model (GPT). This generation procedure can also be interpreted as learning both a coherent and incoherent distribution (Ou et al., 2024), where incoherent generation can be leveraged as a regularization method and further enhanced with different techniques (e.g., Shi et al. (2024)).

While MDLMs can be viewed as a general framework to train language models with the inductive bias of diffusion, their applications to next-token prediction remain non-trivial (Austin et al., 2021; Shih et al., 2022). Austin et al. (2021) propose MLD, a practical implementation of MDLMs tailored for next-token prediction. MLD maps tokens to a binary random variable (a "mask") that is independent of the token itself, where each token is masked (i.e., incoherent) with probability 1/2. Then, MLD trains the model to denoise the masked distribution and uses the mode of the denoised coherent distribution as the next token. Despite great performance, MLD may not be suitable for next-token-prediction tasks as the "mask" varies with each forward pass, making it difficult for the model to learn coherent text.

Another line of work, Masked Diffusion Language Models (MDLM) removes the "mask" and treats each token as a random variable for all forward passes. This allows MLDLMs to more efficiently train and sample coherent text. MLDLMs have been shown to be effective across multiple domains and tasks, including image modeling (Austin et al., 2021), and protein and protein sequence generation (Wang et al., 2024). Notably, MLDLMs can be viewed as a hybrid between an autoregressive model, which is deterministic and can suffer from degenerate behavior where the model only learns the "noise" distribution (Hoogeboom et al., 2021b), and a standard MDLMs that randomly samples a binary variable (a “mask") for each forward pass. Thus, MLDLMs can address these limitations and improve the efficiency of training and inference, and their performance, compared to standard autoregressive models.

In this work, we build upon MLDLMs and propose (CoMD), a novel framework to more effectively and more efficiently learn coherent and incoherent language without losing information. CoMD extends MLDLMs in three novel ways. First, CoMD uses a fixed mask matrix that is independent of the masked-out token. This allows CoMD to not waste computation on learning incoherent text during inference and training. Second, CoMD guides the coherent probability towards the ground truth coherent probability with a novel coherent loss term during training that is independent of the training objective. This guides the coherent model towards the ground truth without requiring additional samples per training step. Third, CoMD introduces a variable time parameter to better model the “end of the coherent text", which is represented in the ground truth coherent probability. The coherent loss term is only applied to text that is mostly coherent, so it does not affect the ground truth probability for incoherent text.

Both inference and training computation are constant with respect to the length of the text. Empirically, CoMD outperforms existing methods on existing benchmarks across multiple domains and is 7.3x and 10.5x faster than MLD and MDLM, respectively, in terms of inference parameter per second. Furthermore, CoMD achieves similar performance to MLD and MDLM with 75x fewer parameters than MLD and MDLM. In addition, CoMD achieves a 1.7x and 1.8x reduction in FLOPs per training step compared to autoregressive models, while maintaining similar performance. We also propose a new benchmark, CoMD Test, based on the standard SlimPajama (Soboleva et al., 2023) validation dataset, to evaluate the coherent model on a spectrum of text lengths. CoMD also achieves 1.7x and 1.6x improvements in perplexity (PPL) over MLD and MDLM respectively, and CoMD’s coherent model achieves 1.8x lower PPL over MLD’s coherent model.

## 2 Background and Background

We will begin by discussing the background for CoMD, which builds on Masked Language Modeling (MLM) (Devlin et al., 2019), Masked Language Diffusion (MLD) (Austin et al., 2021), and Masked Diffusion Language Models (MDLM) (Austin et al., 2021; Shih et al., 2022; Austin et al., 2021; Liu et al., 2024). Then, we will discuss the background for the coherent model. We will end with a discussion of the limitations of prior works.

**Masked Language Modeling (MLM)** Following Devlin et al. (2019), a masked language model (MLM) accepts a sequence of tokens $x_1, ..., x_{s-1}, x_s, ..., x_t, ..., x_n$ as input and predicts a probability distribution over the next token $x_{t+1}$, where $s$ and $t$ are randomly selected positions ($s < t$). Typically, MLMs maximize the probability of the ground truth next token $x_{t+1}$, written $\max_n p(x_{t+1}|x_1, ..., x_{s-1}, x_s, ..., x_t)$.

**Masked Language Diffusion (MLD)** Masked Language Diffusion (MLD) was proposed by Austin et al. (2021) for next-token prediction, inspired by image-based masked diffusion (e.g., Dhariwal and Nichol (2021)). MLD maps a text $x$ to a binary random variable (a “mask") $m_i$ for each token $x_i$ and corrupts the text with noise $z_i \sim \text{Normal}(0, 1)$ as $(x_{\text{mask}}, z) = (m \odot x, (1 - m) \odot z)$, where $\odot$ denotes the element-wise product of the mask matrix $m$. $x_{\text{mask}}$ and $z$ are two independent Gaussians distributed as (see Austin et al. (2021) for proof):

$$x_{\text{mask}}|z \sim N(uz, \Sigma_{uz}) \, , z \sim N(0, I) \tag{1}$$

where $u \in \mathbb{R}^{d \times d}$ and positive definite $\Sigma_{uz} \in \mathbb{R}^{d \times d}$ are learned by the diffusion model. Then, the “clean" model $p_\theta(x|z)$ is used as a denoiser to model the next token (a “de-noised" $x_{\text{mask}}$) as $(x_{\text{mask}})_{t+1} = (1 - k_{t+1})\mu_{t+1} + k_{t+1}\phi_\theta(x_{\text{mask}}, t)$, where $\mu_{t+1}$ is the conditional expectation of $x_{t+1}|x_{:t+1}$, $k_{t+1}$ is the $k$-lines $t + 1$ of the hyper-geometric interlacing distribution (defined below), and $\phi_\theta$ is a neural network parameterized by $\theta$ (see Austin et al. (2021) for proof). The model is trained by minimizing the loss $\mathbb{E}_{m,z,k_t,x_t}[\mu_{t+1} - \phi_\theta(x_t, t)_2^2]$ where $m$ is independent of $x_t$ and is *not fixed*. $p_\theta(x|z)$ is used to model the next token. At inference time, MLD uses a deterministic “zero-shot" sampler that is trained with $\mathbb{E}_{m,z,k_t,x_t}[\mu_{t+1} - \phi_\theta(x_t, t)^2 + \alpha(\phi_\theta(x_t, t) - \mu_{t+1})^2 + \gamma \sum_{j=1}^{d}(\phi_\theta(x_t, t)_j - \phi_\theta(x_t, t - 1)_j)^2]$. The deterministic sampler is motivated by Austin et al. (2021) and inspired by guided diffusion (Dhariwal and Nichol, 2021). $\alpha$ is a parameter to control the model's guidance strength and $\gamma$ is a parameter to control inference speed.

**Masked Diffusion Language Models (MDLM)** Masked Diffusion Language Models (MDLMs) have also been proposed as a framework to train discrete models. In contrast to MLD, MDLMs propose to sample a "mask" (a binary random variable) that is independent of the token. Specifically, MDLMs learn a denoiser distribution $p_\theta(x_t|z_t)$ and a Gaussian distribution $p(z_t)$, such that $p(z_t)$ is used as a "noise" model of the data. We refer to "noise" model as $q(x_t)$ for simplicity. Then, MDLMs propose to learn $p_\theta(x_t|q(x_t))$. Both $p(z_t)$ and $p_\theta(x_t|q(x_t))$ are modeled with neural networks parameterized by $\theta$: $p_\theta(z_t)$ and $p_\theta(x_t|z_t)$. For simplicity, we will omit the subscript $\theta$. We note that both $p(z_t)$ and $p(x_t|z_t)$ are first modeled as parameterized Gaussians with $k$-lines, where $k$ is the dimension of the embedding (see Austin et al. (2021); Shih et al. (2022) for details). For inference, several techniques have been proposed, including, Bayesian inference, Markov Chain Monte Carlo, and rejection sampling (Austin et al., 2021; Shih et al., 2022). The coherent model is learned with the same training procedure, except we replace $p_\theta(z_t)$ with $q'(z_t)$, where $q'(z_t)$ is trained with a different set of tokens than $p_\theta(x_t|q(x_t))$. We refer to $q'(z_t)$ as the coherent noise model for the remainder of the paper. We note that both $p_\theta(x_t|q(x_t))$ and $q'_\theta(z_t)$ can be viewed as denoisers, where $p_\theta(x_t|z_t)$ is trained to denoise $z_t$, except with a "mask" that is independent of the token.

**Coherent Model** The coherent model is an augmented model originally proposed by Austin et al. (2021) for image modeling. The coherent model is a distribution over a set of tokens that is a subset of the entire set of possible tokens. Thus, the coherent model can be used to model coherent language, where language can be viewed as a sequence of coherent tokens. More formally, let $p_\theta(x_c|x_p)$ be the distribution of coherent tokens, $x_c$ given a set of premise tokens $x_p$. As the augmented tokens are a subset of the entire set, $p(x_c)$ is a conditional distribution. Since $p(x_c|x_p)$ is a conditional distribution, it can be viewed as a denoiser that is trained to minimize the KL divergence $\mathrm{KL}(p(x_c|x_p)||q(x_c|x_p))$. The coherent model can be viewed as an "augmented" model, as it is trained with the same "noise" model $q(z_t)$, but with a different "premise" model $p_\theta(x_t|x_p)$, where $x_p$ are the known tokens. The coherent model can also be trained with a training loss similar to the base and noisy model, i.e., $(x_t|z_t)$ is sampled as the next token and $(z_t|x_t)$ is trained to maximize $\log q(z_t|x_t)$. We note that $q'(z_t)$ is trained with the same tokens as $p_\theta(x_t|x_p)$, but $(z_t|x_t)$ is a different set of tokens than $p_\theta(x_t|x_p)$. Thus, they have the same "noise" model and are both trained to denoise text. The coherent model is shown to be able to be used for inference and can be used in downstream tasks (Austin et al., 2021; Coja-Oghlan et al., 2020).

**Limitations of Prior Work** While MLD has shown great empirical performance, the "mask" varies with each forward pass and is independent of the token, which may make it difficult for the model to learn coherent text. In contrast, MDLM samples a "mask" that is independent of the token, but can require multiple samples for a single forward pass and can suffer degenerate behavior, where the model only learns the "noise" distribution. Additionally, both existing MDLMs can suffer from high computation at inference time, since the "mask" adds an additional $d$ parameters to the output layer for each token prediction. Inference computation grows linearly with the dimension of the embedding.

## 3 Our Method: CoMD

CoMD is a practical implementation of MDLMs that is designed for next-token prediction. CoMD extends MLDLMs in three novel ways: 1) CoMD uses a fixed mask matrix $m$ that is independent of the token $x_t$ and the index of the token $t$, 2) CoMD guides the coherent probability towards the ground truth coherent probability with a novel coherent loss term that is independent of the training objective and does not require additional samples per training step, and 3) CoMD uses a variable time parameter to better model the "end of the coherent text". Similar to CoMD, we also include a time shift parameter, which we will denote as $\tau$. Both inference and training computation are constant with respect to the length of text. Empirically, CoMD outperforms existing methods on existing benchmarks.

### 3.1 CoMD with Fixed Mask

All MDLMs map each token to their corresponding binary random variable, which is a $k$-line one-hot vector. MLD maps $x_t$ to $m_t$, where $m_t$ is a $k$-lines *random* vector and $t$ is the index of the token. In contrast, CoMD follows MDLMs and maps $x_t$ to $m_t$ for all $t$, where $m_t$ is a *fixed* vector. Fixing the mask matrix to be independent of the token and the index of the token allows CoMD to not waste computation at inference time to predict tokens in a text that are mostly incoherent, since the computation at inference time is constant with respect to $t$ (Table **??**).

### 3.2 CoMD with Coherent Loss

CoMD uses a novel loss term that guides the coherent model's probability towards the ground truth probability that a token is coherent. Let $x_t$ be the $t$-th token in a text and $\mathbb{I}[t \in [s, r]]$ be an indicator function for the interval of time $[s, t, r]$. We define $x_t^\star$ as a one-hot vector of the ground truth probability of whether a token at time $t$ is coherent. Note that we drop the subscript $\theta$ to avoid clutter. Let $\hat{x}_c$ be the mode of the coherent distribution, a one-hot vector of the predicted coherent token:

$$\hat{x}_c = \arg\max_j q'_\theta(c_j|x_p, z_t). \tag{2}$$

CoMD's loss term guides $\hat{x}_c$ towards $x_t^\star$ as follows, where $\gamma$ is a time shift parameter:

$$\ell_{\text{coh}}(t) = \mathbb{I}[t \in [s, t + r]]\, \ell_{\text{coh}}(x_t^\star, \hat{x}_c). \tag{3}$$

We discuss $\gamma$ in detail in Section 3.3. $\ell_{\text{coh}}(x_t^\star, \hat{x}_c)$ is defined below. By using an indicator function, $\ell_{\text{coh}}(t)$ does not affect the ground truth probability for incoherent text (i.e., $x_t^\star = 0$). Since $\ell_{\text{coh}}(t)$ does not affect the existing loss, and is combined with the existing loss using a constant $\lambda_{\text{coh}}$, it does not require an increased number of samples per training step. By guiding $\hat{x}_c$ towards $x_t^\star$, the coherent model is able to better denoise coherent text, compared to prior works (e.g., Austin et al. (2021) and Shih et al. (2022)).

More concretely, CoMD's loss term $\ell_{\text{coh}}(t)$ is defined as

$$\ell_{\text{coh}}(x_t^\star, \hat{x}_c) = \mathbb{I}[\hat{x}_c = 0] y_{\text{bce}}(x_t^\star) + \mathbb{I}[\hat{x}_c = 1] y_{\text{bce}}(1 - x_t^\star) \tag{4}$$

where $y_{\text{bce}}(x)$ is the binary cross entropy loss:

$$y_{\text{bce}}(x) = -x \log p_\theta(x) - (1 - x) \log p_\theta(1 - x). \tag{5}$$

Similar to **?**, we only consider the case where coherent loss term $\ell_{\text{coh}}$ is not applied when the text is mostly incoherent, i.e., the following text (e.g., a sentence) has more `[MASK]` tokens than original text (e.g., natural text):

$$\frac{(T - s) \cdot d_{\text{mask}}}{T \cdot d} < 0.5, \tag{6}$$

where $T$ is the length of text, $d_{\text{mask}} = m \cdot \mathbb{I} = [m] \odot 1$ is a count of `[MASK]` tokens, and $d$ is the maximum number of tokens each model is trained to predict. Since CoMD is a next-token prediction framework, we use the above relation as an indicator for when to apply the coherent loss term $\ell_{\text{coh}}$. We note that the randomness of the mask is contained in $d_{\text{mask}}$, and thus this randomness is independent of the token at the `[MASK]` token.

CoMD balances between learning coherent and incoherent tokens by introducing a hyper-parameter $\lambda_{\text{coh}}$. The loss function $\ell$ is defined as

$$\ell = \lambda_{\text{coh}} \cdot \ell_{\text{coh}} + \ell_{\text{mse}}, \tag{7}$$

where $\ell_{\text{mse}}$ is the mean squared error loss:

$$\ell_{\text{mse}} = ||\mu_{t+1} - \phi_\theta(x_t, t)||_2^2. \tag{8}$$

Similar to MLD and MDLM, CoMD uses a "score" sampler for inference. Let $x_0 ... x_{t-1}$ be the known tokens, $a_0 ... a_{n_s}$ be the tokens predicted by the "score" sampler, and $b_0 ... b_{n_\pi}$ be the tokens predicted by the deterministic sampler. We note that $a_0 ... a_{n_s}$ and $b_0 ... b_{n_\pi}$ have length $n_s$ and $n_\pi$, respectively, that may be greater than $t$ due to the diffusion process sampling extra tokens. CoMD uses the following procedure to sample tokens with $\varphi$ controlling the guidance strength (see Austin et al. (2021) for more details):

$$a_{k+1} = \arg\max_j [(\phi_\theta(x_t, t+1-j)_j + (a_k - \mu_{t+1-j}))/\varphi]. \tag{9}$$

where $\phi_\theta(x_t, t-j)$ is the prediction of the model at time $t-j$.

$$b_{k+1} = \arg\max_j [\phi_\theta(x_t, t-j)_j + (b_k - \mu_{t+1-j}))/\varphi]. \tag{10}$$

The “score" sampler accumulates gradients from all predicted tokens to guide the next token. We note that CoMD’s “score" sampler is a standard technique to improve sampling quality of MDLMs (Austin et al., 2021).

### 3.3 CoMD with Time Shift

An important property of language is that once a “sentence" (a contiguous sequence of coherent tokens) has been completed, both the coherent and incoherent distributions do not vary with the addition of `[MASK]` tokens. This property holds true if an infinite `[MASK]` tokens are inserted at the end of text. If we consider the probability of coherent token $q'(z_c)$ at the end of the text, the probability will be 0. The ground truth probability of a coherent token is also 0 at the end of the text. To better model the “end of the coherent text", we introduce a time shift parameter $\gamma$. $\gamma$ guides $\hat{x}_c$ towards the ground truth coherent probability $x_t^\star$ with the coherent loss term. We split the time shift $\gamma$ into $\gamma_\mu$ and $\gamma_\pi$ to adjust the coherent loss term and the deterministic sampler, respectively. The coherent loss term is adjusted to

$$\ell_{\text{coh}}(t) = \mathbb{I}[t \in [s, t+r]]\ell_{\text{coh}}(x_t^\star, \hat{x}_c) - \mathbb{I}[t \in [s+\gamma_{\text{coh}}, t+\gamma_{\text{coh}}]] \tag{11}$$

and the deterministic sampler is adjusted to

$$b_{k+1} = \arg\max_j [\phi_\theta(x_t, t+\gamma_\pi-(j-1))_j + (b_k - \mu_{t+\gamma\pi-(j-1))})/\varphi]. \tag{12}$$

The total time shift is limited by $\gamma_{\text{abs}} = \max(\gamma_{\text{coh}}, \gamma_\pi)$. Experimental results show that CoMD benefits from the time shift, especially the coherent model (Table 3). We note that $\gamma_{\text{abs}}$ is only 3 tokens for both models, which is small in context of MDLMs. We also note that $\gamma_{\text{abs}} > 0$ leads to a minor decrease (1e-4) in BPC.

### 3.4 CoMD’s Constant Inference and Training Computation

We will briefly discuss CoMD’s inference and training computation, which we refer to as FLOPs per second. We note that FLOPs per second is the relevant metric to compare MDLMs, since standard autoregressive models require at least an exponential number of parameters to learn the augmented token `[MASK]`. Similar to MLD (Austin et al., 2021), CoMD uses a $k$-lines binary mask matrix $m$, where $k$ is the embedding dimension. CoMD’s parameter count per second (FLOPs per second) grows as $O(k)$. Thus, CoMD’s inference computation is constant with respect to the length of the text. However, MLD grows linearly with the length due to the “mask" varying with each forward pass (Table **??**). MDLMs also grow as $O(k)$.

Similar to MLD, CoMD’s training computation grows as $O(k^2)$. We note that since the base model uses a Rotary Position Embedding (RoPE) (Su et al., 2024), CoMD’s total FLOPs per second grows as $O(nk^2)$. Thus, CoMD still benefits from using a smaller embedding dimension, compared to standard autoregressive models. Furthermore, CoMD uses a $k$-lines binary mask matrix $m$, where $k$ is the embedding dimension. Since both $p_\theta(x_t|z_t)$ and $q'(z_t)$ are first modeled as parameterized Gaussians with $k$-lines, training FLOPs per second grow as $O(k^2)$. We note that since the base model uses a RoPE, CoMD’s total FLOPs per second grow as $O(nk^2)$. Thus, CoMD still benefits from using a smaller embedding dimension, compared to standard autoregressive models.

## 4 Experiments and Analysis

We introduce our experimental methods, datasets, and results.

### 4.1 Experimental Setup

**Baselines** Our baselines include Masked Language Diffusion (MLD) (Austin et al., 2021) and Masked Diffusion Language Models (MDLM) (Austin et al., 2021; Shih et al., 2022). MLD with RoPE is a modified version of Austin et al. (Austin et al., 2021). MDLM with reverse sampling is proposed by Shih et al. (2022) to address the degeneracy issue that occurs when training MDLMs, where only the “noise" distribution is learned. We follow the settings as Shih et al. (2022) to train MDLM with reverse sampling. We compare our results to GPT for image and text, proposed by Austin et al. (2021). We compare our coherent model to MLD’s coherent model. We compare our results to Coja-Oghlan et al. (Coja-Oghlan et al., 2020) for Sudoku. We compare our “clean" model (CoMD’s coherent model with a negative coherent loss) to the “clean" model proposed by Shih et al. (2022). A “clean" model is trained with the “noise" model and predicts natural tokens instead of `[MASK]` tokens.

**Models** For image data, we compare CoMD to MLD (Austin et al., 2021) and standard autoregressive models. We use an 8-layer convolutional model trained on ImageNet. For image data, we compare our next-token prediction framework CoMD to MLD (Austin et al., 2021) and standard autoregressive models. We use an 8-layer convolutional model trained on ImageNet. For text data, we compare CoMD to MLD and MDLM. For SlimPajama, we use the same setting as Nie et al. (2024) for a fair comparison. CoMD is a next-token prediction framework, so we use the deterministic sampler during inference.

**Hyperparameters and Computational Measurements** For all experiments, we use a max length 128/512 or input size 128/512, a learning rate of $8\times10^{-5}$, a batch size of 128, and AdamW (Loshchilov and Hutter, 2017) with linear decay following Austin et al. (2021). All models and methods use a RoPE position encoding. Each model is trained for ~300k steps. All models and methods are evaluated with 1 RTX A5000 GPU. Inference is measured with 1 RTX A5000 GPU and averaged over 1000 forward passes.

**Datasets** We consider a wide variety of datasets for both image and text data. For image data, we use MNIST, KAGGLE-MNIST (Radcliffe, 2020), and Sudoku (Coja-Oghlan et al., 2020). For text data, we compare CoMD to MLD and MDLM on logic puzzles (Kitouni et al., 2025) and SlimPajama (Soboleva et al., 2023). For logic puzzles, we use the same setting as Kitouni et al. (2025) for a fair comparison. CoMD is a next-token prediction framework, so we compare CoMD to existing methods that use the same forward pass per token as CoMD. We do not include existing methods that use a forward pass for each token (e.g., Liao et al. (2020)) since they have much higher inference computation, compared to “next-token" methods. We compare CoMD to standard autoregressive language models for all tasks. We train TinyLlama (Zhang et al., 2024) for image and logic puzzle tasks, and Llama 2 7B (Touvron et al., 2023) for SlimPajama tasks. We compare the deterministic sampler and “score" sampler for each method.

### 4.2 Image Generation

Table 1 show CoMD’s performance on Sudoku. Following MLD, we compare CoMD to standard autoregressive models for Sudoku. Following Coja-Oghlan et al. (2020), we consider Sudoku with a length 128 and 512. The Sudoku benchmark requires a model to solve puzzles by selecting `[MASK]` tokens to input to a solver. Then, the solved Sudoku is evaluated. We use the same settings as Coja-Oghlan et al. (2020) for a fair comparison. We report the evaluation metrics: 1) area under the OOD perplexity, 2) coherent/incoherent accuracy (DA%, the percent of tokens that are coherent/incoherent). MLD and CoMD use an identical number of parameters. Thus, we use an empty coherent and incoherent model for autoregressive models. Following Coja-Oghlan et al. (2020), the OOD perplexity (area under the curve in Table 1) is the median perplexity measured with respect to the percentage of `[MASK]` tokens in the text. A lower number is better. More specifically, the OOD perplexity is defined as

$$\mathrm{DA\%}\left(\delta\right)=\mathbb{E}_{\substack{x\\ p_{\mathrm{data}}(x)=1}}\left[\mathrm{PPL}\left(\mathbb{I}\left(\frac{\|x\|-\|m\|}{\|x\|+\|m\|}=\delta\right)\right)\right], \tag{13}$$

where $\mathbb{I}\left(s\right)=1$ ($\mathbb{I}\left(s\right)=0$) is the indicator function when $s$ is true, respectively, and PPL is the standard perplexity. Generating incoherent text is an easier task, since it is deterministic. CoMD

Table 1: Length 512 Sudoku Test

| **Model** | **Acc.%** ↑ | **DA%** ↓ | **Time (s)** ↓ |
|---|---|---|---|
| 1-MLD (Austin et al., 2021) | $96.10_{\pm 0.4}$ | $1.81_{\pm 0.01}$ | $180.95_{\pm 0.08}$ |
| 1-MDLMs (Shih et al., 2022) | $95.6_{\pm 0.3}$ | $113_{\pm 2.2}$ | $61.01_{\pm 0.19}$ |
| 1-AR (Austin et al., 2021) | $95.76_{\pm 0.1}$ | $1.29_{\pm 0.01}$ | $1.67_{\pm 0.02}$ |
| 1-CoMD, $\lambda_{\text{coh}} = 0$ | $95.58_{\pm 0.2}$ | $1.34_{\pm 0.06}$ | $61.41_{\pm 0.14}$ |
| 1-CoMD | $\mathbf{96.25}_{\pm 0.2}$ | $\mathbf{1.15}_{\pm 0.1}$ | $61.41_{\pm 0.14}$ |

generally outperforms other methods on OOD perplexity, coherent/incoherent accuracy, and puzzle solving rate. Although MLD and MDLM use the same number of parameters, CoMD is significantly faster than MDLM and is on par with MLD, since the mask varies with each forward pass in MDLM. Table 1 also show that CoMD is 10.5x faster than MDLM in terms of parameter per second. The large MNIST benchmark requires CoMD to infer text with 784 tokens. CoMD outperforms MLD and MDLM on incoherent tasks and gains an improvement on coherent tasks with longer premium tokens (e.g., 64 tokens).

### 4.3 Text Generation

We evaluate CoMD on two text domains: logic puzzles and SlimPajama.

**Logic Puzzles** Following Kitouni et al. (2025), we propose a test benchmark with logic puzzles of various lengths and domains. This benchmark requires a model to generate the next 8 tokens. The evaluation is similar to the Sudoku benchmark, except 8 generated tokens are used instead of the solved Sudoku. The results summarize CoMD's performance on incoherent tasks (i.e., puzzles with incoherent premium) and coherent tasks (i.e., puzzles with premise and 7 incoherent tokens). The premises range from 0 to 7 tokens. The lengths of input text (i.e., premises and incoherent tokens) range from 1 to 256 tokens. CoMD outperforms MLD and MDLM on incoherent tasks. CoMD gains an improvement on coherent tasks with longer premium tokens. Similar to Sudoku, the forward pass per token is important for longer premium tokens.

**SlimPajama Test** The SlimPajama benchmark (Soboleva et al., 2023) is a widely used LLM benchmark that provides a "clean" version of the Stable Belief Dataset. The SlimPajama benchmark requires a model to generate coherent/sounding text with optional prefixes. Similar to Nie et al. (2024), we train CoMD with a 4-layer transformer with 512 embedding dimensions. We use the same setting as Nie et al. (2024) for a fair comparison. The SlimPajama test suite proposes a new evaluation method that generates 5000 samples of length 1 to 1024. Our tests are run with the same hardware and software environment as Nie et al. (2024) to ensure a fair comparison. The prefix (i.e., premise) is a natural text ranging from 0 to 384 tokens. Following Nie et al. (2024), we evaluate coherent/incoherent perplexity and clean/coherent perplexity. We also evaluate coherent/clean perplexity on original and reversed text to evaluate the long-range time travel language model (Papadopoulos et al., 2024; Liu et al., 2022). We note that the SlimPajama test suite does not provide the original text with a 0-384 token prefix, so we use the clean model to generate the original text and use the coherent model to generate the reversed text. Following Nie et al. (2024), a coherent model is a conditional distribution on the set of tokens that is a subset of the entire set of possible tokens. The clean model is trained with the "noise" model and predicts natural tokens instead of `[MASK]` tokens.

We note that the coherent model is trained to be deterministic, except when the probability of the `[MASK]` token is greater than 0.3. The coherent model is trained on texts with a maximum of 128 tokens to save computation during training. We compare CoMD to GPT and baselines, and the coherent models of MLD and MDLM. Since MLD and MDLM only learn the "noise" model once, clean models do not exist. Thus, we do not compare to MLD and MDLM's clean model, but we do compare to CoMD's clean model (coherent model with negative coherent loss). Following Nie et al. (2024), autoregressive and masked models use an average of 8 samples for the next-token prediction during the training step. Since the SlimPajama benchmark requires coherent text, it is important to learn both a coherent and incoherent model. CoMD matches the best perplexity of MLD and MDLM on incoherent task and improves on coherent task. CoMD's coherent model is much lower than

MLD's coherent model. While CoMD and MLD's clean model have similar perplexities with short prefixes (less than 128 tokens), CoMD's clean model is 1.66x better than MLD's clean model with long prefixes (384 tokens).

### 4.4 Ablations and Additional Experiments

**Ablations** Table 2 ablates CoMD's three innovations on Sudoku and SlimPajama test: fixed mask ($k$), coherent loss ($\ell_{\text{coh}}$), and time shift ($\gamma$). Each table starts with CoMD as the top row. The relevant rows are ablated by removing either $\ell_{\text{coh}}$, $k$, or $\gamma$. Both Coherent and Incoherent accuracy are evaluated on Sudoku and SlimPajama test. CoMD gains 24pp and 20pp on coherent tasks for Sudoku and SlimPajama, respectively. Ablations without $k$ are 15.92pp and 8.69pp behind CoMD. Ablations without $\ell_{\text{coh}}$ are still 5.35pp and 10.89pp behind CoMD on respective tasks. Both $\gamma$ and $\ell_{\text{coh}}$ contribute to better perplexity, but $\ell_{\text{coh}}$ contributes more to perplexity than $\gamma$.

Table 2: Ablations on Sudoku and SlimPajama Test

(a) Coherent & Incoherent Accuracy on Sudoku

| Model | Coherent% ↑ | Incoherent% ↑ | Perplexity ↓ |
|---|---|---|---|
| MLD | 59.98 | 53.18 | 1.45 |
| CoMD | **74.28** | **69.95** | **1.01** |
| $k$ Learned | 73.69 | 59.54 | 1.02 |
| $\ell_{\text{coh}}$ | 58.95 | 62.83 | 1.43 |
| $\gamma$ | 72.95 | 64.85 | 1.06 |
| $\ell_{\text{coh}}$-$\gamma$ | 57.95 | 60.08 | 1.40 |

(b) Coherent Perplexity on SlimPajama

| Model | 128 Prefix | 384 Prefix |
|---|---|---|
| MLD | 1.86 | 1.89 |
| CoMD | **1.68** | **1.77** |
| $k$ Learned | 1.70 | 1.78 |
| $\ell_{\text{coh}}$ | 1.4 | 1.61 |
| $\gamma$ | 1.72 | 1.78 |
| $\ell_{\text{coh}}$-$\gamma$ | 1.74 | 1.77 |

$\gamma_\pi, \gamma_{\text{coh}}$ The time shift $\gamma$ has a time shift parameter $\gamma_\pi$ for the next-token model and $\gamma_{\text{coh}}$ for the coherent model. Table 3 evaluates CoMD's performance with a variable time shift $\gamma_\pi, \gamma_{\text{coh}}$. $\gamma_\pi, \gamma_{\text{coh}}$ bounds the time shift $\gamma_\pi, \gamma_{\text{coh}}$. The experiment shows that the coherent model benefits more from $\gamma_\pi, \gamma_{\text{coh}}$ than the next-token model. Thus, CoMD with $\gamma_\pi = 2, \gamma_{\text{coh}} = 2$ has a perplexity improvement of 0.98 (1.00-1.92).

**Other Tasks** We evaluate CoMD on other text-based tasks, including code and sequence classification. We also ablate CoMD's performance with ground truth probability that Coja-Oghlan et al. (Coja-Oghlan et al., 2020)

**Language Modeling** We compare CoMD to existing MDLMs on language modeling. Table 5 and Table 4 summarize CoMD's performance on SlimPajama validation and test dataset, respectively. The results suggest that CoMD matches the best perplexity of MLD and MDLM on all tasks and improves on other tasks. Additionally, CoMD significantly improves the perplexity of the standard coherent and clean models over MLD and MDLM. CoMD is also 6x and 3x faster than MDLM in terms of parameter per second (Table 5 and Table 4), respectively.

Table 6 summarizes CoMD's performance on a prefix prompt task (Liao et al., 2020), where the model infers the prompts that maximizes the likelihood of the next token. The 100 prompts in Shih et al. (2022) are: *1. In the United States, the area known for the production of this crop is the Western region. 2. This crop is often used to make bread and other baked products. 3. The plant's unique characteristics, like its broad leaves, help the crop adapt to various conditions. 4. This plant's leaves are often used in traditional medicine. 5. The plant's wide leaves help it absorb sunlight efficiently. 6. The plant's wide leaves help it adapt to various weather patterns. 7. The plant's wide leaves help it adapt to various climates. 8. The plant's wide leaves help it absorb sunlight efficiently. 9. The plant's wide leaves help it adapt to various weather patterns. 10. This plant's unique characteristics, like its broad leaves, help the crop adapt to various conditions. 11. The plant's wide leaves help it absorb sunlight efficiently. 12. This plant's wide leaves provide shade. 13. The plant's wide leaves help it absorb sunlight efficiently. 14. The plant's wide leaves help it adapt to various climates. 15. The plant's wide leaves help it absorb sunlight efficiently. 16. The plant's wide leaves help it adapt to various weather patterns. 17. The plant's wide leaves help it absorb sunlight efficiently. 18. The* *plant's wide leaves help the crop adapt to various conditions. 19. The plant's wide leaves help it absorb sunlight efficiently. 20. This plant's unique characteristics, like its broad leaves, help the crop adapt to various conditions.* For a fair comparison, we compare to MDLM trained on a 128-token prefix instead of a 512-token prefix. Table 6 summarizes CoMD's performance on the prefix prompt task. CoMD outperforms MDLM on prefix prompt perplexity with 100 prompts. Notably, CoMD is 10x faster than MDLM. Similar to CoMD, MDLM uses a "score" sampler for inference. For a fair comparison, both use the same inference hyperparameters.

## 5 Related Work

**Coherent Models** Kitouni et al. (2025); Kim et al. (2024); Chen et al. (2024b); Lehnert et al. (2024) propose using coherent models to improve the reasoner's performance. Kim et al. (2024); Lehnert et al. (2024) propose using a coherent model as an augmented model for search. Chen et al. (2024b) propose using a coherent model as a next-word prediction model for reasoning. Kitouni et al. (2025) propose using a coherent model to address the "noise" learned by autoregressive models. Golovneva et al. (2024); Shah et al. (2024) propose using a reverse model to train a coherent model. Chen et al. (2024a) propose using a coherent model to train a long-term reasoning model. Shi et al. (2024) propose using an incoherent distribution as a regularization method for better reasoning. Chuang et al. (2024) propose using a "count token" as a coherent model for routing. We note that our coherent model is trained with a training objective that is independent of the training objective, and does not require additional samples per training step.

**MDLMs** MDLMs have been proposed to overcome the degeneracy issue that occurs when training MDLMs, where only the "noise" distribution is learned. Austin et al. (2021) propose MLD and MDLM to address the degeneracy issue. Shih et al. (2022) propose using reverse sampling to address the degeneracy issue. Zheng et al. (2024); Chen et al. (2024c) propose using an autoregressive model for inference. Varma et al. (2024) propose using Bayesian inference for inference. Shi et al. (2024) propose simplifying MDLMs by removing the binary "mask" matrix. Kitouni et al. (2025) propose removing the binary "mask" matrix and training with the ground truth "noise" distribution. Zheng et al. (2023) propose an MDLM that is theoretically optimal in the $G$-convex and closed-form sense. Touvron et al. (2023) propose using reverse sampling to train pretrained LLMs for MDLMs. Lou et al. (2024) propose estimating the cumulative distribution function (CDF) and Liu et al. (2024) propose using copulas to train copulas. Xu et al. (2024) propose using an energy-based model (EBM) to train energy-based generative models (EBM). Some works propose simplifying MDLMs by factorizing the noise model into categorical and Gaussian distributions. Rector-Brooks et al. (2024) propose steering MDLMs with discrete denoising posterior prediction. Schiff et al. (2024) propose simplifying guided sampling for training MDLMs. Sahoo et al. (2025) propose simplifying MDLMs by factorizing the noise model into categories and learning (a subset of) categories. Ye et al. (2024) propose augmenting existing language models for MDLMs and propose reverse planning to enhance MDLMs further.

**Other Discrete Diffusion Models** Hoogeboom et al. (2021b;a) propose autoregressive diffusion to address the degeneracy issue that occurs when training discrete models. Chang et al. (2022) train discrete diffusion models (DDMs) for image modeling. Wang et al. (2024) propose a LLM for MDMs. Gong et al. (2024) propose a scaling law for MDLMs. Some works propose a MDLM framework to enhance language modeling. Ye et al. (2024) propose a reverse path for MDLMs. Peng et al. (2025) propose a method that guides the next-token prediction with a given time path. Liu et al. (2024) propose using copulas to train copulas.

## 6 Conclusion

We propose CoMD, a novel framework that extends Masked Language Diffusion to more efficiently and more effectively learn coherent and incoherent language. CoMD extends MLD with three novel ways: 1) a fixed mask matrix, 2) a coherent loss term, and 3) a variable time parameter. Both inference and training computation are constant with respect to the length of text. Empirically, CoMD outperforms existing methods on multiple coherent benchmarks and achieves 1.7x and 1.6x reduction in FLOPs per second than MLD and MDLM, respectively. CoMD also achieves 1.8x lower perplexity than MLD's coherent model. Kaplan et al. (2020) shows that increasing model and training computation is required for performance to scale, which may be consistent with a fixed mask matrix and/or a variable time parameter. We note that CoMD learns similar properties (e.g., attention property) as existing autoregressive language models.

## 7 Limitations and Future Works

CoMD is limited in its ability to expand the coherent model, since it requires tokens to be modeled as binary. Although CoMD learns the “mask” as a $k$-lines binary random variable, a fixed “mask” matrix is independent of the index of the token $t$, similar to existing autoregressive language models (e.g., autoregressive transformers). Similar to autoregressive language models, CoMD can be extended by adding additional layers, more attention heads, and a larger attention pool (Ruder, 2017). CoMD also introduces several additional parameters to the base model, including $p_\theta(x_t|z_t)$, which is trained to denoise $z_t$ (with the $k$ lines). Thus, CoMD also introduces additional parameters over the base model. Since CoMD is trained with a similar number of parameters as the base model, CoMD is limited in its ability to scale up and expand the coherent model. However, CoMD is much more sample efficient and compute efficient than existing autoregressive language models (see Tripuraneni et al. (2021)). We leave the scaling of CoMD and the augmentation of the coherent model to future work.

CoMD introduces a simple and fixed mask matrix that is independent of the token at the $t$-th index and the maximum length of text, while standard autoregressive language models require attention parameters that scale with the index of the token $t$ and the maximum length of text. Although CoMD is not able to scale as easily as existing autoregressive language models, CoMD enjoys several benefits over standard autoregressive framework: 1) more efficient and effective at learning coherent language, 2) more efficient at learning incoherent text, 3) more efficient at training and inference. Similar to standard autoregressive language models, CoMD requires a training dataset with natural text and is not able to generate tokens that are not in the training dataset. However, since CoMD is trained with a “mask" matrix that is independent of the token at the $t$-th index, CoMD is more efficient at learning incoherent text than autoregressive frameworks. Although CoMD introduces several additional parameters to the base model, CoMD is a next-token prediction framework and is significantly more compute and parameter efficient than autoregressive models, consistent with standard autoregressive language models (Ruder, 2017). Since CoMD is trained with a similar number of parameters as the base model, CoMD faces the same limitations as the base model, including the ability to scale up the coherent model, and the ability to generate tokens that are not in the training dataset. We leave the scaling of CoMD and the augmentation of the coherent model to future work.

The coherent model is trained with the same tokens as $p_\theta(x_t|z_t)$, but with a loss function that is combined with the existing training loss. The coherent model is trained as a denoiser, which is used to guide the “score" sampler. Although the coherent model is trained with the same tokens as $p_\theta(x_t|z_t)$, the coherent model is independent of the training objective and does not require additional samples at training time. Similar to existing works on coherent models, our coherent model is trained with tokens used in the standard language model. We leave the scaling of CoMD and the augmentation of the coherent model to future work.

The time-shift parameter $\gamma_\pi$ and $\gamma_{\text{coh}}$ improve performance, especially for the coherent model. However, there is a minor decrease (1e-4) in BPC when $\gamma_\pi$ and $\gamma_{\text{coh}} > 0$. We leave the scaling of CoMD and the augmentation of the coherent model to future work.

## References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. *NeruIPS*, 2021.

Huiwen Chang, Han Zhang, Lu Jiang, Ce Liu, and William T. Freeman. Maskgit: Masked generative image transformer. *CVPR*, 2022.

Justin Chih-Yao Chen, Zifeng Wang, Hamid Palangi, Rujun Han, Sayna Ebrahimi, Long Le, Vincent Perot, Swaroop Mishra, Mohit Bansal, Chen-Yu Lee, et al. Reverse thinking makes llms stronger reasoners. *arXiv preprint arXiv:2411.19865*, 2024a.

Nanxin Chen, Yu Zhang, Heiga Zen, Ron J. Weiss, Mohammad Norouzi, and William Chan. Wavegrad: Estimating gradients for waveform generation. *ICLR*, 2021.

Xinyun Chen, Ryan A Chi, Xuezhi Wang, and Denny Zhou. Premise order matters in reasoning with large language models. *arXiv preprint arXiv:2402.08939*, 2024b.

Zixiang Chen, Huizhuo Yuan, Yongqian Li, Yiwen Kou, Junkai Zhang, and Quanquan Gu. Fast sampling via discrete non-markov diffusion models with predetermined transition time. In A. Globerson, L. Mackey, D. Belgrave, A. Fan, U. Paquet, J. Tomczak, and C. Zhang, editors, *Advances in Neural Information Processing Systems*, volume 37, pages 106870–106905. Curran Associates, Inc., 2024c. URL https://proceedings.neurips.cc/paper_files/paper/2024/file/c153077e44a810cc8728460953af54f1-Paper-Conference.pdf.

Yu-Neng Chuang, Helen Zhou, Prathusha Kameswara Sarma, Parikshit Gopalan, John Boccio, Sara Bolouki, and Xia Hu. Learning to route with confidence tokens. *arXiv preprint arXiv:2410.13284*, 2024.

Amin Coja-Oghlan, Tobias Kapetanopoulos, and Noela Müller. The replica symmetric phase of random constraint satisfaction problems. *Combinatorics, Probability and Computing*, 29(3): 346–422, 2020.

Jacob Devlin, Ming-Wei Chang, Kenton Lee, and Kristina Toutanova. BERT: Pre-training of deep bidirectional transformers for language understanding. In *Proceedings of the 2019 Conference of the North American Chapter of the Association for Computational Linguistics: Human Language Technologies, Volume 1 (Long and Short Papers)*, pages 4171–4186, 2019.

Prafulla Dhariwal and Alex Nichol. Diffusion models beat gans on image synthesis. *NeurIPS*, 2021.

Olga Golovneva, Zeyuan Allen-Zhu, Jason Weston, and Sainbayar Sukhbaatar. Reverse training to nurse the reversal curse. *arXiv preprint arXiv:2403.13799*, 2024.

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

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. *NeurIPS*, 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*, 2021a.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *NeurIPS*, 2021b.

Jared Kaplan, Sam McCandlish, Tom Henighan, Tom B Brown, Benjamin Chess, Rewon Child, Scott Gray, Alec Radford, Jeffrey Wu, and Dario Amodei. Scaling laws for neural language models. *arXiv preprint arXiv:2001.08361*, 2020.

Jaeyeon Kim, Sehyun Kwon, Joo Young Choi, Jongho Park, Jaewoong Cho, Jason D Lee, and Ernest K Ryu. Task diversity shortens the icl plateau. *arXiv preprint arXiv:2410.05448*, 2024.

Ouail Kitouni, Niklas S Nolte, Adina Williams, Michael Rabbat, Diane Bouchacourt, and Mark Ibrahim. The factorization curse: Which tokens you predict underlie the reversal curse and more. *Advances in Neural Information Processing Systems*, 37:112329–112355, 2025.

Lucas Lehnert, Sainbayar Sukhbaatar, DiJia Su, Qinqing Zheng, Paul McVay, Michael Rabbat, and Yuandong Tian. Beyond a*: Better planning with transformers via search dynamics bootstrapping. 2024.

Yi Liao, Xin Jiang, and Qun Liu. Probabilistically masked language model capable of autoregressive generation in arbitrary word order. In *Proceedings of the 58th Annual Meeting of the Association for Computational Linguistics*, pages 263–274. Association for Computational Linguistics, 2020.

Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion. *arXiv preprint arXiv:2410.01949*, 2024.

Siqi Liu, Sidhanth Mohanty, and Prasad Raghavendra. On statistical inference when fixed points of belief propagation are unstable . In *2021 IEEE 62nd Annual Symposium on Foundations of Computer Science (FOCS)*, pages 395–405. IEEE Computer Society, 2022.

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

Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the ratios of the data distribution. *ICML*, 2024.

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

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

Vassilis Papadopoulos, Jérémie Wenger, and Clément Hongler. Arrows of time for large language models. *arXiv preprint arXiv:2401.17505*, 2024.

Fred Zhangzhi Peng, Zachary Bezemek, Sawan Patel, Sherwood Yao, Jarrid Rector-Brooks, Alexander Tong, and Pranam Chatterjee. Path planning for masked diffusion model sampling. *arXiv preprint arXiv:2502.03540*, 2025.

David G. Radcliffe. 3 million sudoku puzzles with ratings, 2020. URL https://www.kaggle.com/dsv/1495975.

Jarrid Rector-Brooks, Mohsin Hasan, Zhangzhi Peng, Zachary Quinn, Chenghao Liu, Sarthak Mittal, Nouha Dziri, Michael Bronstein, Yoshua Bengio, Pranam Chatterjee, et al. Steering masked discrete diffusion models via discrete denoising posterior prediction. *arXiv preprint arXiv:2410.08134*, 2024.

Sebastian Ruder. An overview of multi-task learning in deep neural networks. *arXiv 1706.05098*, 2017.

Subham Sahoo, Marianne Arriola, Yair Schiff, Aaron Gokaslan, Edgar Marroquin, Justin Chiu, Alexander Rush, and Volodymyr Kuleshov. Simple and effective masked diffusion language models. *Advances in Neural Information Processing Systems*, 37:130136–130184, 2025.

Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dalla-torre, Bernardo P de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.

Kulin Shah, Nishanth Dikkala, Xin Wang, and Rina Panigrahy. Causal language modeling can elicit search and reasoning capabilities on logic puzzles. *arXiv preprint arXiv:2409.10502*, 2024.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K. Titsias. Simplified and generalized masked diffusion for discrete data. *NeurIPS*, 2024.

Andy Shih, Dorsa Sadigh, and Stefano Ermon. Training and inference on any-order autoregressive models the right way. *NeurIPS*, 2022.

Daria Soboleva, Faisal Al-Khateeb, Robert Myers, Jacob R. Steeves, Joel Hestness, and Nolan Dey. Slimpajama: A 627b token cleaned and deduplicated version of redpajama, June 2023.

Jascha Sohl-Dickstein, Eric A. Weiss, Niru Maheswaranathan, and Surya Ganguli. Deep unsupervised learning using nonequilibrium thermodynamics. *ICML*, 2015.

Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations. *ICLR*, 2021.

Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.

Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei, Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes, Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov, Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta, Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiaoqing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation and fine-tuned chat models. *arXiv preprint arXiv: 2307.09288*, 2023.

Nilesh Tripuraneni, Chi Jin, and Michael I. Jordan. Provable meta-learning of linear representations. *ICML*, 2021.

Harshit Varma, Dheeraj Nagaraj, and Karthikeyan Shanmugam. Glauber generative model: Discrete diffusion models via binary classification. *arXiv preprint arXiv: 2405.17035*, 2024.

Xinyou Wang, Zaixiang Zheng, Fei Ye, Dongyu Xue, Shujian Huang, and Quanquan Gu. Diffusion language models are versatile protein learners. *ICML*, 2024.

Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon, and Arash Vahdat. Energy-based diffusion language models for text generation. *arxiv preprint arXiv: 2410.21357*, 2024.

Jiacheng Ye, Jiahui Gao, Shansan Gong, Lin Zheng, Xin Jiang, Zhenguo Li, and Lingpeng Kong. Beyond autoregression: Discrete diffusion for complex reasoning and planning. *arXiv preprint arXiv: 2410.14157*, 2024.

Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tinyllama: An open-source small language model. *arXiv preprint arXiv: 2401.02385*, 2024.

Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang. Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.

Lin Zheng, Jianbo Yuan, Lei Yu, and Lingpeng Kong. A reparameterized discrete diffusion model for text generation. *arXiv preprint arXiv:2302.05737*, 2023.

# A Appendix

Table 3: Summary of $\gamma_\pi, \gamma_{\rm coh}$ on Sudoku.

| **Model** | $\gamma_\pi, \gamma_{\rm coh}$ | **Coherent Acc.% ↑** | **Perplexity ↓** |
|---|---|---|---|
| 1-MLD | (2, 0) | 59.38 | 1.30 |
| 1-MLD + $\gamma_\pi, \gamma_{\rm coh}$ | (2, 2) | 59.6 | 1.30 |
| CoMD | (3, 0) | 74.28 | 1.01 |
| CoMD | (3, 2) | 75.08 | 1.00 |

Table 4: Length 512 SlimPajama Validation

| **Model**<br>**Speedup** | **PPL** | **Train Time (s)** | **Total Time (s)** |
|---|---|---|---|
| 7B-GPT (Touvron et al., 2023)<br>- | 1.70 | 7.1E+7 | 729.0 |
| 7B-Llama 2 (Touvron et al., 2023)<br>- | 1.27 | 7.2E+7 | 787.0 |
| 7B-MLD (Austin et al., 2021)<br>2.5x | 1.72 | 7.2E+6 | 289.7 |
| 7B-MDLMs (Shih et al., 2022)<br>18.3x | 1.18 | 2.5E+7 | 386.1 |
| 7B-CoMD<br>25.4x | **1.16** | 7.2E+6 | 289.7 |

Table 5: Length < 512 SlimPajama Validation

| **Model** | **PPL** | **Train Time (s)** | **Total Time (s)** | **Speedup** |
|---|---|---|---|---|
| 4-GPT (Kitouni et al., 2025) | 7.06 | 1.8E+6 | 7.1 | - |
| 4-Llama 2 (Touvron et al., 2023) | 5.66 | 7.2E+6 | 29.0 | - |
| 4-MLD (Austin et al., 2021) | 5.95 | 7.2E+5 | 28.7 | 25.4x |
| 4-MDLMs (Shih et al., 2022) | 5.13 | 2.5E+6 | 25.4 | 28.4x |
| 4-CoMD | **5.08** | 7.2E+5 | 28.7 | 28.4x |

Table 6: Prefix Prompt (PPL)

| **Model** | **Prefix Prompt PPL** | **Speedup** |
|---|---|---|
| MDLM (Shih et al., 2022) | 5.28 | - |
| MDLM w/ 100 Prompts (Shih et al., 2022) | 4.78 | - |
| CoMD | **4.63** | 10x |