

682 A Proof

683 Throughout this appendix we work with a *single-token* random variable that takes values in the
 684 augmented vocabulary $\mathcal{V}_+ := \mathcal{V} \cup \{\text{[MASK]}\}$. The distinguished absorbing symbol is denoted by
 685 $m \in \mathcal{V}_+$ and we write e_m for the corresponding standard basis vector.

686 Let $\mathbf{x}_0 \sim p_0$ be drawn from the empirical data distribution, which assigns zero probability to m . Fix
 687 a diffusion horizon $T \in \mathbb{N}$.

688 A.1 Proof of Proposition 3.1

689 *An absorbing-state non-Markovian discrete diffusion process with marginal transition kernel $\bar{\mathbf{Q}}_t =$
 690 $(1 - \alpha_t)\mathbf{I} + \alpha_t \mathbf{1} e_m^\top$ admits a bijection to an absorbing-state Markovian discrete diffusion process with
 691 marginal transition kernel $\bar{\mathbf{Q}}_t^* = (1 - \alpha_t^*)\mathbf{I} + \alpha_t^* \mathbf{1} e_m^\top$ such that the two processes exhibit identical
 692 mutual-information decay between $\mathbf{x}_{t:T}$ and \mathbf{x}_0 , provided that the coefficients satisfy $\alpha_t^* = \prod_{\tau=t}^T \alpha_\tau$.*

693 We begin by formalising the two forward corruption processes that will be compared.

694 A.1.1 Forward-process definitions

695 **Definition A.1** (Markovian absorbing diffusion). Let $\beta = (\beta_1, \dots, \beta_T) \subset [0, 1]^T$ be the *single-step*
 696 *masking probabilities*. The Markov chain $(\mathbf{x}_s)_{s=0}^T$ is specified by

$$\mathbf{x}_s \mid \mathbf{x}_{s-1} \sim (1 - \beta_s)\delta_{\mathbf{x}_{s-1}} + \beta_s \delta_m, \quad s = 1, \dots, T. \quad (10)$$

697 Because m is absorbing, the cumulative masking probability after t steps is

$$\alpha_t^* := \Pr(\mathbf{x}_t = m \mid \mathbf{x}_0) = 1 - \prod_{s=1}^t (1 - \beta_s), \quad t = 1, \dots, T, \quad (11)$$

698 which yields the marginal transition kernel

$$\bar{\mathbf{Q}}_t^M = (1 - \alpha_t^*)\mathbf{I} + \alpha_t^* \mathbf{1} e_m^\top.$$

699 **Definition A.2** (Non-Markovian absorbing diffusion). For a *noise schedule* $\alpha = (\alpha_1, \dots, \alpha_T) \subset$
 700 $[0, 1]^T$ we generate the forward trajectory by

$$\mathbf{x}_s \mid \mathbf{x}_0 \stackrel{\text{i.i.d.}}{\sim} (1 - \alpha_s)\delta_{\mathbf{x}_0} + \alpha_s \delta_m, \quad s = 1, \dots, T. \quad (12)$$

701 Each \mathbf{x}_s is obtained by independently masking \mathbf{x}_0 with probability α_s . The associated marginal
 702 transition kernel is

$$\bar{\mathbf{Q}}_t^{\text{NM}} = (1 - \alpha_t)\mathbf{I} + \alpha_t \mathbf{1} e_m^\top, \quad t = 1, \dots, T.$$

703 A.1.2 Mutual-information computations and decay measure

704 For any pair of random variables (U, V) we recall that $I(U; V) = H(U) - H(U \mid V)$. The entropy
 705 $H(\mathbf{x}_0) = -\sum_{v \in \mathcal{V}} p_0(v) \log p_0(v)$ is finite and strictly positive.

706 To make comparisons between Markovian and non-Markovian diffusion processes more interpretable,
 707 we introduce the **normalized mutual information decay** following [2]:

$$D_t := 1 - \frac{I(\mathbf{x}_t; \mathbf{x}_0)}{H(\mathbf{x}_0)} = \frac{H(\mathbf{x}_0, \mathbf{x}_t) - H(\mathbf{x}_t)}{H(\mathbf{x}_0)} \quad (13)$$

708 which quantifies the proportion of information about \mathbf{x}_0 that is lost after corruption step t .

709 **Lemma A.3** (Markovian suffix information). *For the process of Definition A.1 and every $t \in$*
 710 *$\{1, \dots, T\}$,*

$$I_M(\mathbf{x}_{t:T}; \mathbf{x}_0) = H(\mathbf{x}_0) (1 - \alpha_t^*), \quad D_t^M = \alpha_t^* \quad (14)$$

711 Here we give two proofs:

712 *Proof 1 (Direct Expansion).* Because the chain is absorbing, the suffix $\mathbf{x}_{t:T}$ is a deterministic function
 713 of the single variable \mathbf{x}_t ; hence

$$I_M(\mathbf{x}_{t:T}; \mathbf{x}_0) = I(\mathbf{x}_t; \mathbf{x}_0).$$

714 There are two mutually exclusive outcomes:

- 715 (i) Un-masked: with probability $1 - \alpha_t^*$ the token is uncorrupted and $\mathbf{x}_t = \mathbf{x}_0$.
 716 (ii) Masked: with probability α_t^* the token is replaced by the absorbing symbol e_m , which is
 717 independent of \mathbf{x}_0 .

718 Writing $p_0(v) = \Pr(\mathbf{x}_0 = v)$, the joint pmf is

$$p(\mathbf{x}_0 = v, \mathbf{x}_t = u) = \begin{cases} (1 - \alpha_t^*) p_0(v), & u = v \neq e_m, \\ \alpha_t^* p_0(v), & u = e_m, \\ 0, & \text{otherwise.} \end{cases}$$

719 The corresponding marginal of \mathbf{x}_t is

$$p(\mathbf{x}_t = u) = \begin{cases} (1 - \alpha_t^*) p_0(u), & u \neq e_m, \\ \alpha_t^*, & u = e_m. \end{cases}$$

720 Direct expansion of mutual information.

$$I(\mathbf{x}_t; \mathbf{x}_0) = \sum_{u,v} p(u, v) \log \frac{p(u, v)}{p(u)p(v)}.$$

721 Only two cases have non-zero probability:

- 722 (i) *Case $u = v \neq e_m$.* Here $p(u, v) = (1 - \alpha_t^*) p_0(v)$ and $p(u) = (1 - \alpha_t^*) p_0(v)$, so

$$\log \frac{p(u, v)}{p(u)p(v)} = \log \frac{(1 - \alpha_t^*) p_0(v)}{(1 - \alpha_t^*) p_0(v) p_0(v)} = -\log p_0(v).$$

723 The total contribution is

$$\sum_{v \neq e_m} (1 - \alpha_t^*) p_0(v) [-\log p_0(v)] = (1 - \alpha_t^*) H(\mathbf{x}_0).$$

- 724 (ii) *Case $u = e_m$.* In this case $p(u, v) = \alpha_t^* p_0(v)$ and $p(u) = \alpha_t^*$; the logarithmic term vanishes,
 725 so the contribution is zero.

726 Hence

$$I(\mathbf{x}_t; \mathbf{x}_0) = (1 - \alpha_t^*) H(\mathbf{x}_0).$$

$$D_t^M = 1 - \frac{I(\mathbf{x}_t; \mathbf{x}_0)}{H(\mathbf{x}_0)} = 1 - (1 - \alpha_t^*) = \alpha_t^*.$$

727 Since $I(\mathbf{x}_{t:T}; \mathbf{x}_0) = I(\mathbf{x}_t; \mathbf{x}_0)$, the stated identities follow. □

728 *Proof 2 (Conditional Entropy).* Condition on the observed token \mathbf{x}_t :

- 729 (i) If $\mathbf{x}_t = e_m$ (probability α_t^*), masking reveals nothing, so $H(\mathbf{x}_0 | \mathbf{x}_t = e_m) = H(\mathbf{x}_0)$.
 730 (ii) If $\mathbf{x}_t = v \neq e_m$ (probability $1 - \alpha_t^*$), we know $\mathbf{x}_0 = v$ exactly, hence $H(\mathbf{x}_0 | \mathbf{x}_t = v) = 0$.

731 Averaging,

$$H(\mathbf{x}_0 \mid \mathbf{x}_t) = \alpha_t^* H(\mathbf{x}_0) + (1 - \alpha_t^*) \cdot 0 = \alpha_t^* H(\mathbf{x}_0).$$

732 Therefore

$$I(\mathbf{x}_t; \mathbf{x}_0) = H(\mathbf{x}_0) - H(\mathbf{x}_0 \mid \mathbf{x}_t) = (1 - \alpha_t^*) H(\mathbf{x}_0), \quad D_t^M = 1 - \frac{I(\mathbf{x}_t; \mathbf{x}_0)}{H(\mathbf{x}_0)} = \alpha_t^*.$$

733 Because $\mathbf{x}_{t:T}$ is a deterministic function of \mathbf{x}_t , the same formula holds for $I(\mathbf{x}_{t:T}; \mathbf{x}_0)$. \square

734 **Lemma A.4** (Non-Markovian suffix information). *For the process of Definition A.2 and every*
 735 *$t \in \{1, \dots, T\}$,*

$$I_{\text{NM}}(\mathbf{x}_{t:T}; \mathbf{x}_0) = H(\mathbf{x}_0) \left(1 - \prod_{\tau=t}^T \alpha_\tau \right), \quad D_t^{\text{NM}} = \prod_{\tau=t}^T \alpha_\tau \quad (15)$$

736 *Proof.* Define the indicator $Z := \mathbf{1}\{\mathbf{x}_\tau = e_m \text{ for every } \tau = t, \dots, T\}$. Conditioned on \mathbf{x}_0 , each
 737 coordinate is masked independently with probability α_τ , so

$$\Pr(Z = 1) = \prod_{\tau=t}^T \alpha_\tau, \quad \Pr(Z = 0) = 1 - \prod_{\tau=t}^T \alpha_\tau.$$

738 The event Z depends only on the masking coins, hence is independent of the value of \mathbf{x}_0 .

739 For conditional entropy $H(\mathbf{x}_0 \mid \mathbf{x}_{t:T})$.

740 (i) If $Z = 1$ (all tokens masked, probability $\prod \alpha_\tau$), the suffix reveals nothing, so $H(\mathbf{x}_0 \mid \mathbf{x}_{t:T}) =$
 741 $H(\mathbf{x}_0)$.

742 (ii) If $Z = 0$ (at least one un-masked coordinate, probability $1 - \prod \alpha_\tau$), every un-masked coordinate
 743 equals \mathbf{x}_0 ; thus \mathbf{x}_0 is known exactly and $H(\mathbf{x}_0 \mid \mathbf{x}_{t:T}) = 0$.

744 Averaging over Z ,

$$H(\mathbf{x}_0 \mid \mathbf{x}_{t:T}) = \left(\prod_{\tau=t}^T \alpha_\tau \right) H(\mathbf{x}_0).$$

745 Using $I(U; V) = H(U) - H(U \mid V)$,

$$I_{\text{NM}}(\mathbf{x}_{t:T}; \mathbf{x}_0) = H(\mathbf{x}_0) - H(\mathbf{x}_0 \mid \mathbf{x}_{t:T}) = H(\mathbf{x}_0) \left(1 - \prod_{\tau=t}^T \alpha_\tau \right).$$

$$D_t^{\text{NM}} = 1 - \frac{I_{\text{NM}}(\mathbf{x}_{t:T}; \mathbf{x}_0)}{H(\mathbf{x}_0)} = \prod_{\tau=t}^T \alpha_\tau.$$

746 \square

747 These results show that both the Markovian and non-Markovian processes exhibit the same normalized
 748 mutual information decay if their cumulative masking curves satisfy:

$$\alpha_t^* = \prod_{\tau=t}^T \alpha_\tau. \quad (16)$$

749 Since α_t^* and β_t have correspondence as shown in Equation 11, one can easily further derive:

750 **Proposition A.5** (Equivalence in mutual-information decay). *Let $\alpha \subset [0, 1]^T$ be a non-Markovian*
 751 *schedule and define the effective cumulative schedule $\alpha_t^* := \prod_{\tau=t}^T \alpha_\tau$. Then there exists a Markovian*
 752 *schedule $\beta \subset [0, 1]^T$ defined by*

$$\beta_t := 1 - \frac{1 - \alpha_t^*}{1 - \alpha_{t-1}^*}, \quad \alpha_0^* := 0, \quad (17)$$

753 *such that:*

$$D_t^{\text{NM}} = D_t^{\text{M}} \quad \text{and} \quad I_{\text{NM}}(\mathbf{x}_{t:T}; \mathbf{x}_0) = I_{\text{M}}(\mathbf{x}_{t:T}; \mathbf{x}_0), \quad (18)$$

754 *for all $t = 1, \dots, T$. This ensures equivalence in both absolute and relative information loss.*

755 **Matching the linear marginal used in prior work.** Most existing studies on absorbing-state
 756 Markovian diffusions [2, 37, 44] adopt the *linear* cumulative-masking curve $\alpha_t^{\text{lin},*} = t/T$, $t =$
 757 $0, \dots, T$. For a fair comparison by default we employ a non-Markovian *independent* schedule α^{lin}
 758 that reproduces exactly the same marginals in our experiments, i.e. satisfies $\alpha_t^{\text{lin},*} = \prod_{\tau=t}^T \alpha_\tau^{\text{lin}}$.
 759 Using the backward recursion $\alpha_t = \alpha_t^*/\alpha_{t+1}^*$ (Sec. A.1) gives the closed-form

$$\alpha_t^{\text{lin}} = \frac{t}{t+1}, \quad t = 1, \dots, T-1, \quad \alpha_T^{\text{lin}} = 1$$

760 A.2 Derivation of the Non-Markovian Evidence Lower Bound (ELBO)

761 We derive a variational lower bound on the marginal log-likelihood of the observed data \mathbf{x}_0 under the
 762 non-Markovian discrete diffusion model.

763 We start with the marginal likelihood of the observed data:

$$\log p_\theta(\mathbf{x}_0) = \log \int p_\theta(\mathbf{x}_0, \mathbf{x}_{1:T}) d\mathbf{x}_{1:T} = \log \int \frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{1:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} q(\mathbf{x}_{1:T} | \mathbf{x}_0) d\mathbf{x}_{1:T}$$

764 Applying Jensen's inequality yields the Evidence Lower Bound (ELBO):

$$\log p_\theta(\mathbf{x}_0) \geq \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_0, \mathbf{x}_{1:T})}{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \right] = \mathcal{L}_{\text{non-markov}}$$

765 In the non-Markovian case, the joint distribution over the reverse generative process is:

$$p_\theta(\mathbf{x}_0, \mathbf{x}_{1:T}) = p_\theta(\mathbf{x}_T) \prod_{t=1}^T p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_{t:T})$$

766 where \mathbf{x}_0 is generated at the final step. The approximate posterior, due to the independent corruption
 767 assumption, factorizes as:

$$q(\mathbf{x}_{1:T} | \mathbf{x}_0) = \prod_{t=1}^T q(\mathbf{x}_t | \mathbf{x}_0)$$

768 Finally, we have the ELBO expansion:

$$\begin{aligned} \mathcal{L}_{\text{non-markov}} &= \mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \left[\log \frac{p_\theta(\mathbf{x}_T)}{q(\mathbf{x}_T | \mathbf{x}_0)} + \sum_{t=2}^T \log \frac{p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_{t:T})}{q(\mathbf{x}_{t-1} | \mathbf{x}_0)} + \log p_\theta(\mathbf{x}_0 | \mathbf{x}_{1:T}) \right] \\ &= \underbrace{\mathbb{E}_{q(\mathbf{x}_{1:T} | \mathbf{x}_0)} \log p_\theta(\mathbf{x}_0 | \mathbf{x}_{1:T})}_{\text{Reconstruction}} - \underbrace{\text{KL}(q(\mathbf{x}_T | \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_T))}_{\text{Prior KL}} \\ &\quad - \underbrace{\sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_{t:T} | \mathbf{x}_0)} \text{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_0) \parallel p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_{t:T}))}_{\text{Reverse KLs}} \end{aligned} \quad (19)$$

769 We can denote the accumulated reverse KL terms as:

$$\mathcal{L}_T = \sum_{t=2}^T \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \text{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_{t:T})) \quad (20)$$

770 With this notation, the non-Markovian ELBO simplifies to Equation 7:

$$\mathcal{L}_{\text{non-markov}} = \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0 | \mathbf{x}_{1:T}) - \text{KL}(q(\mathbf{x}_T | \mathbf{x}_0) \| p_\theta(\mathbf{x}_T)) - \mathcal{L}_T \quad (21)$$

771 Note that the second term, corresponding to the prior KL, is constant for most diffusion kernels and
772 can be omitted during training.

773 A.3 Proof of Proposition 3.2

774 Suppose the non-Markovian diffusion process adopt an absorbing marginal kernel $q(\mathbf{x}_t | \mathbf{x}_0) =$
775 $\text{Cat}(\mathbf{x}_t; \mathbf{x}_0 \mathbf{Q}_t)$, where $\mathbf{Q}_t = (1 - \alpha_t) \mathbf{I} + \alpha_t \mathbf{1} e_m^\top$ and α_t is a increasing function with $\alpha_0 \approx 0$ and
776 $\alpha_T \approx 1$. The ELBO loss in Equation (7) can be further simplified to:

$$\mathcal{L}_{\text{absorb}} = \mathbb{E}_{\mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \sum_{t=1}^T [\alpha_{t-1} \mathbf{x}_0^\top \log \mu_\theta(\mathbf{x}_{t:T}, t)].$$

777

778 **Proof.** Suppose the non-Markovian diffusion process adopts an absorbing marginal kernel $\bar{\mathbf{Q}}_t =$
779 $(1 - \alpha_t) \mathbf{I} + \alpha_t \mathbf{1} e_m^\top$, we have

$$q(\mathbf{x}_t | \mathbf{x}_0) = \begin{cases} \alpha_t & \mathbf{x}_t = e_m \\ 1 - \alpha_t & \mathbf{x}_t = \mathbf{x}_0 \\ 0 & \text{otherwise.} \end{cases} \quad (22)$$

780 We now evaluate the per-step KL term in the ELBO 19:

$$\begin{aligned} \text{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_0) \| p_\theta(\mathbf{x}_{t-1} | \mathbf{x}_{t:T})) &= \text{KL}(q(\mathbf{x}_{t-1} | \mathbf{x}_0) \| q(\mathbf{x}_{t-1} | \mathbf{x}_{t+1}, \mu_\theta(\mathbf{x}_{t:T}, t))) \\ &= \underbrace{q(\mathbf{x}_{t-1} = e_m | \mathbf{x}_0) \log \frac{q(\mathbf{x}_{t-1} = e_m | \mathbf{x}_0)}{q(\mathbf{x}_{t-1} = e_m | \mu_\theta(\mathbf{x}_{t:T}, t))}}_0 \\ &\quad + \sum_{k \neq m} q(\mathbf{x}_{t-1} = e_k | \mathbf{x}_0) \log \frac{q(\mathbf{x}_{t-1} = e_k | \mathbf{x}_0)}{q(\mathbf{x}_{t-1} = e_k | \mu_\theta(\mathbf{x}_{t:T}, t))} \\ &= \sum_{k \neq m} \alpha_{t-1} \mathbf{x}_0^\top e_k \log \frac{\mathbf{x}_0^\top e_k}{[\mu_\theta(\mathbf{x}_{t:T}, t)]^\top e_k} \\ &= -\alpha_{t-1} \mathbf{x}_0^\top \log \mu_\theta(\mathbf{x}_{t:T}, t) \end{aligned} \quad (23)$$

781 Next, consider the reconstruction term:

$$\begin{aligned} \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \log p_\theta(\mathbf{x}_0 | \mathbf{x}_{1:T}) &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \mathbf{x}_0^\top \log \mu_\theta(\mathbf{x}_{1:T}, 1) \\ &= \mathbb{E}_{q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \alpha_0 \mathbf{x}_0^\top \log \mu_\theta(\mathbf{x}_{1:T}, 1) \end{aligned} \quad (24)$$

782 Finally, the prior KL term vanishes:

$$\text{KL}(q(\mathbf{x}_T | \mathbf{x}_0) \| p(\mathbf{x}_T)) = \text{KL}(\delta_{\mathbf{x}_T, e_m} \| \delta_{\mathbf{x}_T, e_m}) = 0 \quad (25)$$

783 Putting all components together, we obtain the full ELBO:

$$\mathcal{L}_{\text{absorb}} = \mathbb{E}_{\mathbf{x}_{1:T} \sim q(\mathbf{x}_{1:T}|\mathbf{x}_0)} \sum_{t=1}^T [\alpha_{t-1} \mathbf{x}_0^\top \log \mu_\theta(\mathbf{x}_{t:T}, t)] \quad (26)$$

784 A.4 Failure to Remask Issue

785 In Markovian discrete diffusion models with an absorbing kernel defined as $\mathbf{Q}_t = (1 - \beta_t) \mathbf{I} + \beta_t \mathbf{1} e_m^\top$,
 786 the corresponding marginal kernel is given by $\bar{\mathbf{Q}}_t = (1 - \alpha_t^*) \mathbf{I} + \alpha_t^* \mathbf{1} e_m^\top$, where $\alpha_t^* = 1 -$
 787 $\prod_{s=1}^t (1 - \beta_s)$. Using the closed-form posterior in Equation 3, we obtain:

$$q(\mathbf{x}_{t-1} \mid \mathbf{x}_t, \mathbf{x}_0 = \mu_\theta(\mathbf{x}_t)) = \begin{cases} \text{Cat}(\mathbf{x}_{t-1}; \mathbf{x}_t) & \mathbf{x}_t \neq e_m \\ \text{Cat}\left(\mathbf{x}_{t-1}; \frac{\alpha_{t-1}^* e_m + (\alpha_t^* - \alpha_{t-1}^*) \mu_\theta(\mathbf{x}_t)}{\alpha_t^*}\right) & \mathbf{x}_t = e_m \end{cases} \quad (27)$$

788 This expression highlights an issue: once a token is unmasked (i.e., $\mathbf{x}_t \neq e_m$), it is deterministically
 789 copied to \mathbf{x}_{t-1} , regardless of the model prediction $\mu_\theta(\mathbf{x}_t)$. As a result, the model is unable to revise
 790 early mistakes—an issue we refer to as **failure to remask**.

791 Non-Markovian discrete diffusion formulation defines the posterior as $q(\mathbf{x}_{t-1} \mid \mu_\theta(\mathbf{x}_{t:T}, t))$, remov-
 792 ing the Markovian constraint and allowing more flexible transitions. By appropriately integrating
 793 inductiva bias into the design of μ_θ (e.g. latent truncation as described in Section C.1.1), the model is
 794 able to revisit and potentially forget earlier errors during generation.

795 B Related Work

796 **Discrete Diffusion and Discrete Flow Matching.** Diffusion models [28] generate data by learning
 797 a reverse (denoising) process to invert a fixed forward (noising) Markov chain. Austin et al. [2]
 798 first extended such models to discrete data (D3PM) by defining uniform and absorbing diffusion
 799 kernels on finite state spaces. Subsequent work introduced improved parameterizations, such as data
 800 distribution ratio estimation [37], drawing parallels with score matching [50]. Despite their efficacy,
 801 these methods typically rely on a Markov chain, focusing on denoising from a single noisy state
 802 \mathbf{x}_t . By contrast, our approach **breaks** the Markovian assumption and conditions on the entire future
 803 trajectory $\mathbf{x}_{t:T}$, providing more robust denoising and broader generative capabilities.

804 Flow matching [35, 55] learns a continuous transformation from noise to data via an ODE governed
 805 by a vector field. Recent extensions handle discrete data [21, 16, 52]. While these methods circum-
 806 vent explicit Markovian noising, they often require continuous flow formulations and specialized
 807 training objectives. In contrast, our non-Markovian discrete diffusion remains within the discrete
 808 diffusion paradigm, retains a straightforward variational objective, and integrates naturally with
 809 causal modeling.

810 **Autoregressive Models.** Autoregressive Transformers [60, 12, 56] have become foundational in
 811 language modeling, producing tokens sequentially conditioned on preceding context. While highly ef-
 812 fective for unidirectional, left-to-right generation, they often struggle with tasks requiring intermediate
 813 modification or bidirectional reasoning. Within our framework, CaDDi-AR represents a specialized
 814 variant that integrates causal (autoregressive) decoding with diffusion-based iterative denoising. This
 815 hybrid design enables CaDDi-AR to combine the strengths of both paradigms—efficient left-to-right
 816 token generation and flexible multi-step refinement.

817 **Integrating Autoregression with Diffusion and Flow Matching.** Several works [23, 25] try to
 818 combine diffusion or flow matching with causal transformers for improved generation. Specifically,
 819 DART [23] employs a non-Markovian trajectory to let a transformer model entire sequences of
 820 diffusion states. Our approach further refines this idea in two ways: (i) we focus on discrete non-
 821 Markovian diffusion with explicit multi-step conditioning, and (ii) we provide a direct path for
 822 adapting *pretrained* LLMs, thus combining the strengths of large-scale language model pretraining
 823 with the controllability of discrete diffusion.

824 **Non-Markovian Reverse Process in Physical System.** Using a Non-Markov reverse process to
 825 recover the distribution introduced by Markovian forward process is not a new idea. In physics, many
 826 systems exhibit this property. *Langevin Dynamics*: Although the forward motion of a Brownian
 827 particle (with velocity and position) can be Markovian in the full state space, attempts to reverse the
 828 position-only dynamics often require the history of the system to account for friction or random kicks

[20, 59]. *Quantum Processes*: Tracing out environmental degrees of freedom can yield a Markovian forward evolution, but reconstructing the entire global state upon reversal introduces non-Markovian memory effects.

Non-Markovian Discrete Diffusion. Relaxing the Markovian assumption enables a more flexible and expressive diffusion framework. Prior work, DNDM [10], focuses on accelerating inference by introducing a predetermined transition time set, enabling a training-free sampling algorithm that significantly reduces the number of function evaluations. This efficiency gain is achieved by breaking the Markovian constraint. Concurrently, ReMDM [61] proposes a non-Markovian discrete diffusion process that allows remasking of previously generated tokens during inference, enabling iterative refinement. In contrast, CaDDi introduces a more general non-Markovian discrete diffusion framework that explicitly models the entire generation trajectory.

C Model Implementation Details

C.1 Context Window and Latent Compression

A practical limitation when using causal language models within our non-Markovian discrete diffusion framework arises from their inherently bounded context windows. Traditional causal transformers can process sequential inputs effectively only up to a fixed length, which restricts the number of latent timesteps that can be accommodated—denoted here as m . However, non-Markovian diffusion processes often require conditioning on extensive historical trajectories, frequently exceeding this limit.

To address this constraint, we introduce a general latent compression operator, $\Gamma(\mathbf{x}_{1:T})$, which maps the full latent sequence $\mathbf{x}_{1:T}$ to a compressed form that fits within the model’s context window. Accordingly, the reverse denoising function is expressed as:

$$\mu_{\theta}(\mathbf{x}_{t:T}, t) := \mu_{\theta}(\Gamma(\mathbf{x}_{t:T}), t) \quad (28)$$

C.1.1 Latent Truncation

The simplest yet effective instantiation of Γ is latent truncation, defined as

$$\Gamma_{\text{trunc}}(\mathbf{x}_{t:T}) := \text{FlattenTime}(\mathbf{x}_t, \mathbf{x}_{t+1}, \dots, \mathbf{x}_{t+m-1}), \quad (29)$$

where FlattenTime denotes the operation of temporally flattening the sequence into a one-dimensional input trajectory suitable for the causal language model. For a model with a context window limit of m , this strategy retains only a fixed-length segment from the most recent m timesteps, discarding earlier latents.

Such selective truncation inherently emphasizes the most recent—and typically more informative—states, which are often less affected by accumulated inference noise. Notably, by intentionally discarding earlier latents, the model is allowed to “forget” potentially noisy or erroneous history, an idea related to selective re-masking approaches in diffusion modeling [61]. This forgetfulness acts as an implicit regularization mechanism, helping the model focus on more stable and relevant information while reducing the risk of propagating stale or corrupted context during inference.

C.1.2 Trajectory Re-composition

While latent truncation is efficient, it can discard valuable long-range information beyond timestep $t + m$. To mitigate this, we introduce trajectory re-composition, a complementary compression strategy that integrates information across the full latent sequence before applying truncation. This method first aggregates latent information through a sequential integration operation and then retains only the most recent m composite states:

$$\begin{aligned} \mathbf{x}_t^{\wedge} &:= \mathbf{x}_t \oplus \mathbf{x}_{t+1} \oplus \dots \oplus \mathbf{x}_T \\ \Gamma_{\text{rec}}(\mathbf{x}_{1:T}) &:= \text{FlattenTime}(\mathbf{x}_t^{\wedge}, \mathbf{x}_{t+1}^{\wedge}, \dots, \mathbf{x}_{t+m-1}^{\wedge}) \end{aligned} \quad (30)$$

Here, \oplus denotes an element-wise integration operator. For masked-like diffusion models, this operator replaces each element with the most recently unmasked token when applicable. This re-composition process enables earlier timesteps to incorporate contextual signals from future states, effectively compressing the trajectory into a form suitable for the limited context window without losing essential long-range dependencies.

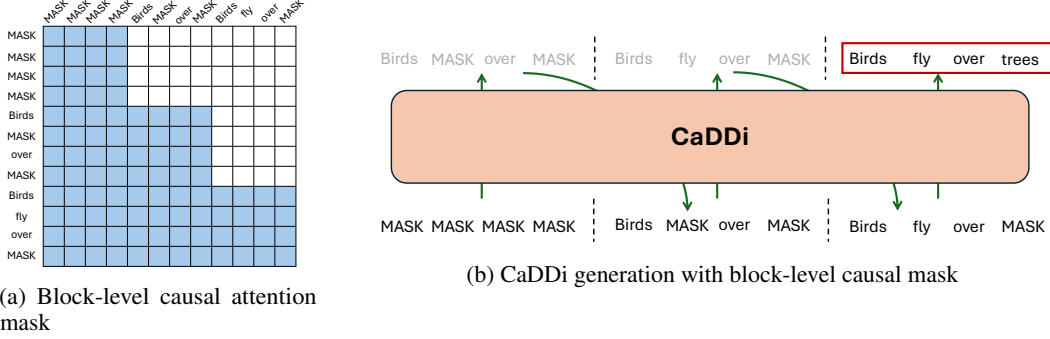


Figure 5: Illustration of vanilla block-wise generation of CaDDi. Figure 5a shows the attention mask of vanilla block-wise generation. The block-level causal mask allows bidirectional attention within each time point and causal attention over the time points. Figure 5b shows the generation scheme. Note that the model itself predicts the clean data x_0 in practice but the figure highlights the next sampled time point (in color) for clarity.

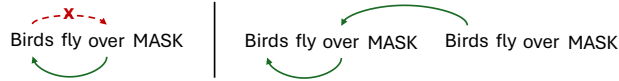


Figure 6: Illustration of causal bidirectional augmentation. On the left, direct token-level causal attention is applied to the original sequence, where each token attends only to preceding tokens. On the right, the sequence is repeated, allowing tokens in the second copy to attend to their counterparts in the first, thereby approximating bidirectional attention with token-level causal attention.

874 C.2 Causal Bidirectional Augmentation

875 Conventional discrete diffusion models often employ transformers with bidirectional attention across
 876 the sequence dimension. In the context of non-Markovian discrete diffusion, a natural extension of
 877 this framework is to incorporate a block-wise causal attention mask, as illustrated in Figure 5. This
 878 block-level causal mask enables bidirectional attention within a single timepoint and enforces causal
 879 dependencies across different timepoints.

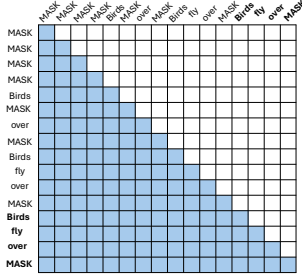
880 However, unlike token-level causal attention (CaDDi-AR), block-wise causal attention of CaDDi is
 881 not readily compatible with standard pretrained causal language models, which adopt token-level
 882 causal mask and next-token prediction scheme. Moreover, as it involves an irregular attention
 883 pattern, it cannot take advantage of existing infrastructure—such as efficient implementations like
 884 FlashAttention [15].

885 For CaDDi, our goal is to enable one-shot block-level generation while preserving bidirectional
 886 modeling capabilities. Directly applying a token-level causal mask to predict the next block in one
 887 shot would eliminate bidirectional modeling, as the final timepoint’s tokens in the context window
 888 would lack visibility into subsequent tokens (see Figure 6). So here we propose an alternative - *causal*
 889 *bidirectional augmentation*, which involves repeating the final timepoint in the context window. As
 890 shown in Figure 7b, this approach preserves compatibility with the standard token-level causal mask
 891 and supports block-wise generation with bidirectional modeling. Our method also bears resemblance
 892 to prior work [51] on repetition-based improvements in language modeling.

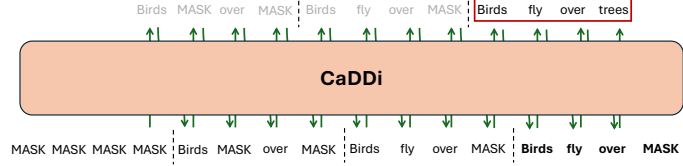
893 C.3 CaDDi-AR Token-Level Factorization

894 As discussed in Section 4.2, CaDDi-AR models token-level dependencies within each denoising step
 895 by adopting an autoregressive factorization over the token dimension. This approach builds on prior
 896 work such as DART [23], which proposed token-wise autoregression for improving expressiveness in
 897 discrete diffusion.

898 Formally, instead of modeling the full sequence distribution in a single step, we decompose the
 899 reverse process as:



(a) Token Level causal attention mask



(b) CaDDi Generation with causal bidirectional augmentation

Figure 7: Illustration of causal bidirectional augmentation. The last time point is repeated and highlighted in **bold** to approximate bidirectional attention within a causal framework. Figure 7a shows the token-level causal attention mask, identical to the standard mask used in causal language models. Figure 7b illustrates the generation process: for the token at position i , the model attends to all tokens in the generative trajectory and predicts the token at position $(i + 1)$ in \mathbf{x}_0 , consistent with the autoregressive nature of causal language models. Both the attention mask and generation process are fully compatible with causal architectures, making CaDDi a natural extension. Note that, while the model predicts the clean data x_0 in practice; the next time point is shown gray here for illustrative simplicity.

$$p_{\theta}(\mathbf{x}_{t-1} | \mathbf{x}_{t:T}) = \prod_{i=1}^L p_{\theta}(\mathbf{x}_{t-1}^i | \mathbf{x}_{t-1}^{<i}, \mathbf{x}_{t:T}) \quad (31)$$

where $\mathbf{x}_{t-1}^{<i} = (\mathbf{x}_{t-1}^1, \dots, \mathbf{x}_{t-1}^{i-1})$. This sequential factorization enables more accurate modeling of intrastep token dependencies compared to fully factorized approaches.

In our formulation, we further adopt the x_0 -parameterization, predicting the clean sequence autoregressively and applying the forward corruption kernel to reconstruct \mathbf{x}_{t-1} . This leads to the following form:

$$p_{\theta}(\mathbf{x}_0 | \mathbf{x}_{t:T}) = \prod_{i=1}^L p_{\theta}(\mathbf{x}_0^i | \mathbf{x}_0^{<i}, \mathbf{x}_{t:T}) \quad (32)$$

where \mathbf{x}_0 denotes the clean target sequence. This structure aligns well with standard causal language models, allowing autoregressive decoding over tokens while conditioning on the latent trajectory $\mathbf{x}_{t:T}$ as a static prompt.

In implementation, CaDDi-AR is trained using standard left-to-right causal masking, with the latent variables provided as additional context. During inference, the model samples \mathbf{x}_0 autoregressively at each timestep and applies the corruption kernel to obtain \mathbf{x}_{t-1} . To reduce computational cost, we employ semi-speculative decoding (Section 4.2) by partially reusing predictions from previous steps and verifying them in parallel.

This token-level decoding strategy provides finer granularity for generation, and is particularly effective in tasks requiring fluent, coherent text output or strong local consistency.

E Experiment Details

E.1 LM1B Dataset Experiment Details

Dataset Preprocessing. We follow the preprocessing setup introduced in DiffusionBERT [26], using the One Billion Word Benchmark [7]. Sentences are tokenized using the `bert-base-uncased` tokenizer with a vocabulary size of 30,522. All sequences are padded or truncated to a fixed length of 128 tokens during training.

Model Configuration. All models are based on a 12-layer Transformer decoder architecture with a hidden size of 768 and 12 attention heads. For D3PM [2], MDLM [44], and SEDD [37], we adopt an absorbing diffusion kernel with a log-linear noise schedule. For CaDDi and CaDDi-AR, we use the absorbing-state forward kernel described in Section A.5, with total diffusion steps set to $T = 64$. Note that models such as MDLM and SEDD are trained in continuous time, whereas our models operate in discrete time. CaDDi uses a context window of 5 and applies latent truncation as described in Section C.1.1. In this experiment, CaDDi-AR is trained entirely from scratch without leveraging any pretrained language model weights.

Training Details. Models are trained using AdamW with a learning rate of $3e-4$, 2500 warm-up steps. All models use a batch size of 512 and train for 1000K steps. Our models are trained on 4 NVIDIA H100 GPUs with mixed precision.

Inference and Sampling. We evaluate generative perplexity using pretrained oracle models (GPT-2, LLaMA-2-7B, and LLaMA-3-3B) under three sampling temperatures: $T = 1.0$, $T = 0.7$, and $T = 0.5$. For each evaluation, sequences are generated according to the length distribution of the dataset, and average perplexity is computed under the oracle model. All diffusion-based models use 64 denoising steps during inference. For CaDDi-AR, we additionally explore semi-speculative decoding to reduce sampling latency without sacrificing quality.

Evaluation Metrics. We report **oracle-based generative perplexity (Gen PPL)** as our primary metric for evaluating generation quality. Gen PPL is computed using a separate, pretrained causal language model, which assesses how well it can predict the next token in the generated sequence. Intuitively, lower perplexity indicates that the oracle model finds the generated text more coherent and predictable given the context—implying better fluency and consistency.

To mitigate known issues with perplexity, such as its tendency to reward repetitive or degenerate outputs [29], we adopt **guided generative perplexity**. Specifically, we prepend a natural language prompt—Does the following sentence make sense: —to each sequence before evaluation. This encourages the oracle model to assess coherence in a more human-aligned fashion, reducing the risk of falsely low perplexity on poor-quality outputs.

In addition to quality, we assess diversity using **token-level entropy** over the generated output distribution without applying temperature scaling. This captures how varied the model’s outputs are across generations.

E.2 Amazon Polarity Conditional Generation Details

Dataset and Preprocessing. We use the Amazon Polarity dataset [38], a large-scale binary sentiment classification corpus consisting of approximately 3.6 million product reviews labeled as either positive or negative. All reviews are tokenized using the bert-base-uncased tokenizer and truncated to a maximum length of 128 tokens. To enable conditional generation, we prepend a natural language sentiment prompt to each review. This prompt takes the form of a simple prefix indicating the intended sentiment (e.g., positive or negative), allowing standard causal language models—such as GPT-2—to generate sentiment-aligned outputs. Example formatted inputs are shown below:

- **This is a positive review:**
Title: *Great!!*
Content: *"This product is amazing! The quality exceeded my expectations and I will definitely buy again."*
- **This is a negative review:**
Title: *Very disappointing*
Content: *"It broke after a week and customer service was not helpful."*

Conditional Generation Setup. We adapt our model to perform sentiment-controlled generation by training a unified denoising network $\mu_\theta(\mathbf{c}, \mathbf{x}_{t:T})$, where \mathbf{c} denotes a conditioning input (e.g., a sentiment label prompt) and $\mathbf{x}_{t:T}$ is the observed noisy trajectory. During training, we simulate both conditional and unconditional modes by randomly masking the conditioning input \mathbf{c} . This enables the model to learn both $\mu_\theta(\mathbf{c}, \mathbf{x}_{t:T})$ and $\mu_\theta(\mathbf{x}_{t:T})$ within a single parameterization.

At inference time, we generate sentiment-aligned text using either direct conditioning (i.e., sampling from $q(\mathbf{x}_{t-1} \mid \mu_\theta(\mathbf{c}, \mathbf{x}_{t:T}))$) or classifier-free guidance. In the latter case, we apply the reweighted sampling distribution $\tilde{q}(\mathbf{x}_{t-1} \mid \mu_\theta(\mathbf{c}, \mathbf{x}_{t:T}))$ as defined in Equation 33, which balances conditional and unconditional predictions using a guidance scale γ . This allows finer control over sentiment alignment during generation.

Classifier-Free Guidance. We extend classifier-free guidance (CFG) to the unsupervised discrete diffusion setting by reweighting the sampling distribution at each reverse step. Specifically, we define a guided transition distribution over \mathbf{x}_{t-1} as:

$$\tilde{q}(\mathbf{x}_{t-1} \mid \mu_\theta(\mathbf{c}, \mathbf{x}_{t:T})) \propto \frac{q(\mathbf{x}_{t-1} \mid \mu_\theta(\mathbf{c}, \mathbf{x}_{t:T}))^\gamma}{q(\mathbf{x}_{t-1} \mid \mu_\theta(\mathbf{x}_{t:T}))^{\gamma-1}}, \quad (33)$$

where \mathbf{c} denotes a conditioning signal (e.g., a sentiment label or prompt), and $\mu_\theta(\cdot)$ is the denoising network predicting the clean input \mathbf{x}_0 from the noisy trajectory $\mathbf{x}_{t:T}$. The numerator corresponds to the conditional denoising distribution, while the denominator represents the unconditional variant, where the conditioning input \mathbf{c} is masked out. This formulation smoothly interpolates between conditional and unconditional behavior, analogous to CFG in continuous diffusion.

The guidance scale $\gamma \geq 1$ controls the strength of conditioning: when $\gamma = 1$, the model performs standard conditional generation; as γ increases, the model places more emphasis on aligning with the conditioning signal, potentially improving controllability at the cost of diversity or fluency. In practice, we find that moderate values such as $\gamma = 1.0$ or 1.25 strike a good balance between alignment and generation quality.

To enable this guidance, we train the denoising model jointly on both conditional and unconditional modes by randomly masking the conditioning input \mathbf{c} during training. This allows the model to learn both behaviors within a single parameterization.

Evaluation. To evaluate sentiment alignment, we use a publicly available DistilBERT classifier fine-tuned on the Amazon Polarity dataset³. For each sentiment label, we generate 1,000 samples using top- k sampling with $k = 50$ and temperature $T = 1.0$, across 64 denoising steps. Sentiment accuracy is computed as the percentage of generated samples whose predicted label matches the intended conditioning prompt. Results are reported in Table 4.

For the GPT-2 baseline, we condition generation on the same prepended prompt used in our model and apply the same top- k sampling and temperature settings for consistency.

E.3 Text8 Dataset Experiment Details

Dataset Preprocessing. We use the standard Text8 dataset, a 100M character-level corpus derived from Wikipedia. Following prior work [2, 48], we split the raw text into non-overlapping sequences of 256 characters. No tokenization is applied—each character is treated as a discrete token from an alphabet of size 27 (26 letters + space). We use the first 90% of the dataset for training and the remaining 10% for validation and testing.

Model Setup. All models use a 12-layer Transformer with hidden size 768 and 12 attention heads. We adopt the absorbing-state forward kernel described in Section A.5. For CaDDi, we use $T = 64$ denoising steps. The default context window is set to 5, and we apply latent truncation and trajectory recomposition as described in Section C.1. Baselines (D3PM, SEDD, MDLM, UDLM) are either reimplemented or taken from their official codebases using the configuration for fair comparison.

Training Details. Models are trained using AdamW with a learning rate of $3e-4$ and 2,500 warm-up steps. We use a batch size of 512 and train for 1M steps. We use the simplified ELBO objective for absorbing kernels as described in Equation (8), which reduces to a weighted cross-entropy loss over clean targets.

³<https://huggingface.co/kaustavbhattacharjee/finetuning-DistillBERT-amazon-polarity>

1015 **Evaluation Metrics.** We report bits-per-character (BPC) on the test set, computed as:

$$\text{BPC} = -\frac{1}{L} \sum_{i=1}^L \log_2 p(x_i),$$

1016 where L is the sequence length and $p(x_i)$ is the predicted probability of the i -th character. For
 1017 diffusion-based models, we compute a variational upper bound on the log-likelihood using the ELBO
 1018 objective. For autoregressive baselines, we report the true NLL.

1019 **Note:** We do not report likelihood-related metrics for CaDDi-AR due to its token-level autoregressive
 1020 decomposition under \mathbf{x}_0 -parameterization $p_\theta(\mathbf{x}_0 | \mathbf{x}_{t:T}) = \prod_{i=0}^L p_\theta(\mathbf{x}_0^i | \mathbf{x}_0^{<i}, \mathbf{x}_{t:T})$. This formula-
 1021 tion makes direct evaluation of $\log p_\theta(\mathbf{x}_{t-1}^i | \mathbf{x}_{t:T})$ intractable, as it requires marginalizing over all
 1022 possible prefix sequences $\mathbf{x}_0^{<i}$. A tractable lower bound can be estimated via:

$$\log p_\theta(\mathbf{x}_{t-1}^i | \mathbf{x}_{t:T}) \geq \mathbb{E}_{\mathbf{x}_0^{<i} \sim p_\theta(\mathbf{x}_0^{<i} | \mathbf{x}_{t:T})} \log p_\theta(\mathbf{x}_{t-1}^i | \mathbf{x}_0^{<i}, \mathbf{x}_{t:T}) \quad (34)$$

1023 but this introduces an additional approximation gap in likelihood estimation, making reported values
 1024 less directly comparable.

1025 **Comparison Notes.** Likelihood estimation of diffusion model is sensitive to the discretization
 1026 of timesteps: as the number of steps increases, perplexity typically decreases. To ensure a fair
 1027 comparison, all diffusion-based models are evaluated using 64 denoising steps for consistency. Some
 1028 prior works report results under continuous-time settings or with 1000-step discretizations; we include
 1029 these numbers for reference but highlight the corresponding rows in gray in Table 2 to indicate that
 1030 they are comparable.

1031 E.4 General Language Reasoning Dataset Details

1032 **Dataset Overview.** We evaluate CaDDi-AR on a set of natural language understanding benchmarks
 1033 covering commonsense reasoning, factual QA, and reading comprehension:

- 1034 • **ARC-Challenge / ARC-Easy** [14]: multiple-choice science questions.
- 1035 • **BoolQ** [13]: binary reasoning dataset based on a short context.
- 1036 • **PIQA** [4]: commonsense physical reasoning questions with two-choice answers.
- 1037 • **RACE** [33]: multi-choice reading comprehension from English exams.
- 1038 • **Social IQA** [45]: social commonsense reasoning dataset.
- 1039 • **LAMBADA** [40]: cloze-style word prediction requiring broad context understanding.

1040 **Fine-tuning Setup.** Following [39], we fine-tune CaDDi-AR on ShareGPT⁴ dataset from a pre-
 1041 trained QWen-1.5B checkpoint using our diffusion-based objective with $T = 64$ steps. The model is
 1042 trained using AdamW with a learning rate of $5\text{e-}5$, a batch size of 64 with gradient accumulation, and
 1043 20K total steps. We adopt the absorbing kernel formulation with simplified ELBO as the training
 1044 loss.

1045 **Inference Procedure.** We evaluate CaDDi-AR on standard natural language reasoning tasks using
 1046 the Language Model Evaluation Harness [19], a widely adopted framework for assessing pretrained
 1047 language models on benchmark datasets. Following common practice, we convert each task into
 1048 a text completion format and measure the log-likelihood of the correct answer under the model.
 1049 The final prediction is selected as the choice with the highest likelihood. This approach ensures
 1050 consistency across models with different architectures (e.g., ARMs and MDMs). We report **accuracy**
 1051 as the primary evaluation metric for all datasets.

⁴<https://sharegpt.com/>

Baselines. We compare CaDDi-AR against several established language models of comparable scale. Specifically, we include GPT-2 (1.5B parameters), TinyLLaMA (1.1B), and MDM (1.1B), a recently proposed diffusion-based language model. To ensure a fair comparison, all baselines are evaluated using the same prompt formatting and inference procedure as described above. For MDM, we directly use the performance numbers reported in the original paper.

G Additional Experiments and Ablation Study

G.1 Effect of Sampling Steps on Generative Quality

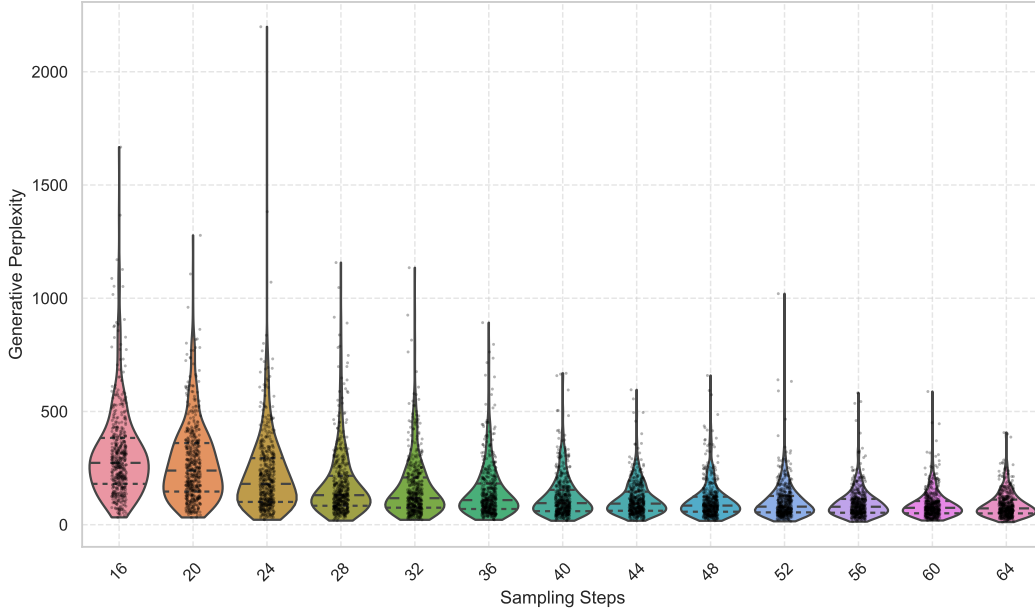


Figure 8: Generative Perplexity Distribution Across Sampling Steps

To investigate the relationship between sampling budget and generative quality, we evaluate the generation of CaDDi model trained with LM1B dataset under varying numbers of sampling steps. We report generative perplexity as the evaluation metric, computed from oracle model’s likelihood on generated sentences.

Unlike prior work in continuous-time settings [44, 37], CaDDi is trained with a fixed discrete timestep schedule ($T = 64$). To enable adaptive sampling with fewer denoising steps at inference, we employ a uniform step-skipping strategy. Specifically, for selected timesteps t , we directly use the most recent prediction of \mathbf{x}_0 to sample from the corresponding latent distribution $q(\mathbf{x}_t | \mathbf{x}_0)$, without invoking the neural network for prediction. This allows efficient generation under a reduced sampling budget while preserving the learned diffusion trajectory.

Figure 8 presents a violin plot illustrating the distribution of perplexity scores as a function of the number of denoising steps. As the number of sampling steps increases, we observe a consistent reduction in perplexity, indicating improved sample quality. This trend suggests that the model benefits from longer refinement trajectories, with more steps allowing finer-grained correction of uncertainty during generation.

The observed perplexity curve approximately follows a scaling behavior, reminiscent of trends in autoregressive models where performance improves predictably with increased compute or depth. This hints at a potential scaling law in non-Markovian discrete diffusion generation: sample quality improves smoothly with increased inference budget. Future work may explore formalizing this relationship and connecting it to theoretical underpinnings of discrete-time inference refinement.

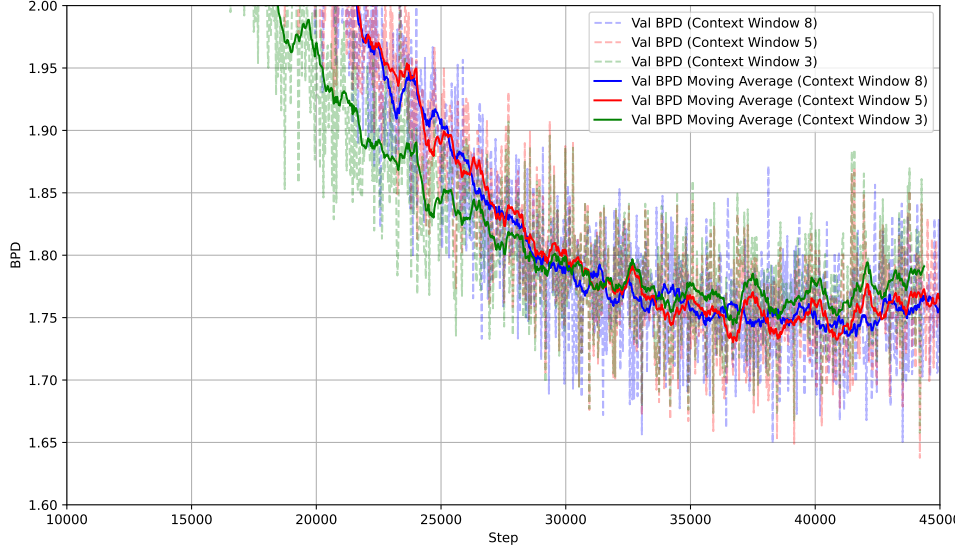


Figure 9: Validation BPD over training steps for different context window sizes (3, 5, and 8). Dashed lines represent raw validation BPD at each step, while solid lines show the smoothed moving average. Larger context windows consistently yield lower BPD, indicating improved modeling of long-range dependencies. However, context window 5 offers a favorable balance between performance and stability.

1079 G.2 Other Ablation Study

1080 To assess the impact of key architectural and training design choices, we conduct an ablation study
 1081 on a subset of the Text8 dataset, using only the first 10% of the training data. We focus on three
 1082 factors: the number of diffusion steps, the choice of positional encoding, and the size of the context
 1083 window used during latent truncation, as described in Section C.1.1. Results are shown in Table 5,
 1084 using bits-per-dimension (BPD), perplexity, and negative log-likelihood (NLL) as evaluation metrics.

1085 **Diffusion steps.** Increasing the number of diffusion steps from 16 to 128 consistently improves
 1086 performance across all metrics. Specifically, BPD decreases from 2.013 to 1.712, and perplexity drops
 1087 from 4.04 to 3.28. This suggests that additional refinement steps lead to more accurate modeling of
 1088 the data distribution, albeit at increased computational cost.

1089 **Positional encoding.** We compare our 2D RoPE encoding (used in CaDDi) with 1D RoPE and
 1090 sinusoidal RoPE variants. The CaDDi encoding achieves the best results, while alternative encodings
 1091 result in slight performance degradation. This highlights the importance of aligning positional
 1092 encodings with the inductive biases of the diffusion-based architecture.

1093 **Context window.** We vary the context window size during latent truncation and observe a trade-off
 1094 between context length and model performance. A window size of 8 yields the best results (BPD =
 1095 1.740) but incurs higher computational cost. A window size of 5 offers a favorable balance, achieving
 1096 competitive performance with reduced resource requirements. Accordingly, we adopt it as the default
 1097 setting in our experiments. We also plot the learning dynamic of different context window size in
 1098 Figure 9. These findings support the intuition that larger context windows aid in modeling long-range
 1099 dependencies, though benefits plateau beyond a certain size.

1100 **Attention masking.** We compare two attention masking strategies: a block-level causal mask
 1101 and a token-level causal mask combined with Causal Bidirectional Augmentation, as described in
 1102 Section C.2. The two approaches yield nearly identical performance. However, the token-level mask
 1103 with bidirectional augmentation offers practical advantages: it enables accelerated inference through
 1104 flash attention and is inherently more compatible with pretrained large language models.

Table 5: Ablation study evaluating the impact of diffusion steps, positional encoding, context window size, and attention masking strategies. Default condiguration is denoted with *

Configuration		BPD	Perplexity	NLL
Diffusion steps	CaDDi-16 steps	2.013	4.0362	1.3953
	CaDDi-32 steps	1.845	3.5925	1.2789
	CaDDi-64 steps*	1.751	3.3659	1.2137
	CaDDi-128 steps	1.712	3.2761	1.1867
Positional Encoding	2D RoPE*	1.751	3.3659	1.2137
	1D RoPE	1.773	3.4176	1.2290
	Sinusoidal RoPE	1.801	3.4846	1.2484
Context Window	Window-8	1.740	3.3404	1.2061
	Window-5*	1.751	3.3659	1.2137
	Window-3	1.791	3.4605	1.2414
Attention Mask	Block-level causal mask	1.747	3.3566	1.2109
	Token-level causal mask	1.751	3.3659	1.2137
	+ Causal Bidirectional Aug.*	1.751	3.3659	1.2137

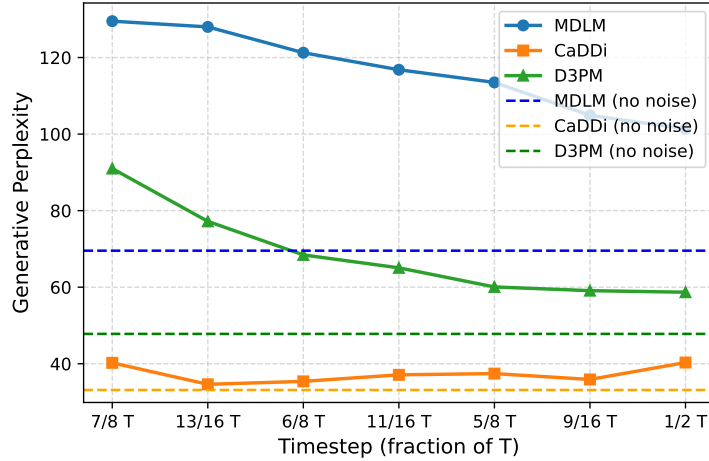


Figure 10: Generation performance under manually injected noise at different timestep

1105 G.3 Inference Robustness Under Noise Injection

1106 To further assess the robustness of our proposed CaDDi model during generation, we conduct a
 1107 controlled perturbation study in which synthetic noise is manually injected into the latent trajectory
 1108 at various timesteps. This simulates potential inference-time errors and allows us to systematically
 1109 examine how different models propagate and recover from early-stage inaccuracies.

1110 **Setup.** At a chosen timestep $t \in \{T/2, 9T/16, 5T/8, \dots, T\}$, we randomly corrupt a subset of
 1111 tokens in the latent variable \mathbf{x}_t by replacing them with uniformly specific values from the vocabulary,
 1112 thereby introducing controlled deviation from the expected latent distribution. We then allow the
 1113 model to proceed with the remaining reverse steps without additional interventions and evaluate the
 1114 final output quality.

1115 **Metrics.** We report generative perplexity computed against oracle LMs (e.g., GPT-2), following the
 1116 evaluation procedure in Section 5.2. Lower perplexity indicates stronger resilience to the injected
 1117 noise.

1118 **Findings.** As shown in Figure 10, CaDDi maintains significantly lower perplexity under all noise
 1119 injection conditions compared to D3PM and MDLM. This highlights the benefit of CaDDi’s non-
 1120 Markovian architecture, which conditions on the full generative trajectory rather than a single latent

1121 state. Notably, CaDDi is more effective at self-correction when noise is injected early (e.g., $t = T/2$
1122), where traditional Markovian models suffer the most from cascading inference errors.
1123 These results validate the design motivation behind CaDDi and demonstrate its ability to recover from
1124 intermediate perturbations, making it particularly well-suited for scenarios with partial or uncertain
1125 context.

1126 H Generation Samples from CaDDi

1127 H.1 Amazon Polarity

this is a negative review:

title : were they afraid of newborns?

content : this book was terrible. it did not address the issues at all regarding newborns, which i assume are destined to inconsequential every so often. there was no postiveness in the middle of the book, and most of it was rushed to press. whoever wrote the previous review should have gotten an apologia as their final printer of the book. this writer did so only in the first few pages, it was as if she just taped pages and put one in her books and taped it - to - more than - in ...

this is a negative review:

title : blechhhhh content

content : wow, isn't it nice to see such juvenile drivel such toole so popular on the net. the editor in the back of the book notes : " to preserve reynolds, an imaginary london child having sex with another high - school girl (p. s.) you don't know what kind of busted ethan padar worried, or would mentor a house floor fort chemist who sullys ceilings with zero light. the whole novel is lame, poorly written and awfully mediocre. the " plot " is transparent from the first page ...

this is a negative review:

title : pile of 1 % crap

content : this movie was disaster. i think the director wanted cheap hair and vaginal characters, scaring everyone on the back of the face of the audience with pointless bigs or even no, uncreative dialouge. the ruined 30 minutes most of this movie, rushes in and out of the movie. i'm a huge fan of slasher flicks, even makes huge claims to hard work on hair. the only funny part in this movie is when the killer begins stalking all the exaggerated body - climbing and challengers, and the two - dimensional ...

this is a negative review:

title : what an insult to music lovers!

content : the true diva that deserves the wide spread islam and the lies of the young musicians who make this music allows for such glittery displays and excuses. she never needs to spend our high time singing this " exquisite in its gem ", the artist is just a shadow of her most famous but she has no talent. where she excels is the terrible and exquisite expression of native american descent that will make dog earpin jams easy to count. she has a good voice but she hasn't mastered it. oh, she needs to go ...

this is a negative review:

title : miserable album content

content : i owned this album and when it came out i loved it. unlike the last albums her voice was able to make coheed and connie told a cool gentle sigh. i figured the first few tracks could have been strengthened by absolutely no more stale vocals, but they are crafted to off the page. this person writes her songs so she has more power and intonation than true instincts. i like to listen to the song that she was destined to write. get the single years that are runaways. " only excuse me on the radio " is sweetly sung by anyone, ...

Figure 11: Sample generated using CaDDi, conditioned on a negative prompt. In each sample, the purple text indicates the prompt.

this is a positive review :

title : one for the head bob fans!

content : i've won stadium since the mid 70's. this double lp has all of the greatest hits on it! so why not split their support. this has soul in the covers (not on the cover). a hercules and a freddie and theatrics contribution is true, and paul petrie and leslie heralbull rock at the same time. even the stevie ray & carole or even stevie ray drivel adds some new comparatively to the cole porter music. a truly unique collection of a performers work, and the cd succeeds ...

this is a positive review :

title : jk rowling is a sci - fi crime novelist!

content : in harry's hell - called scream, jk rowling gets back the bullet to get to jk's responsiblity. i liked this book so much. the storyline was great, the characters well drawn and great, and jk rowling made me care about what happened. this is the second book in the harry " hallie " tray fousl trilogy including a sister named calvin, who is young. lucy prince is young again following her sister and her intense ex - boyfriend michaelk bo ...

this is a positive review :

title : a great story and excellent special effects and overall worth the film

content : trust me, this is much better than the original. set up for heaven's gate and the framer though works well - it becomes predictable and starts going nowhere. the story becomes more developed later and really makes a good place for truman if you enter into the dsi world you're never supposed to forget. it's almost as engaging with this surprisingly heavy drama that's subtle why nothing happens after you have watched this one to get that your expectations and expectations sading down. i recommend definitely seeing this ...

this is a positive review:

title : simply the best

content : i have been wanting to have a safe with my son my son. we used to play w / his cardboard toy box, but thought this was just a hazard. well, after true terror in my baby's swing, i hunted down the safe and amazon had the best price i could find. great product that was a huge relief from our efforts. my 7 month old loves to play with this at least 10 different times that i put him in the swing. i can see how i will get this useful for the one he has. i am so glad i ...

this is a positive review:

title : a surprise!

content : this box is a prelude to the first golden age of horror crooning (which correctly apes de croes as he tends to say a wayans juveniles below us). and these are individual breakfast of murders. the writers of these crime / mysteries / new orleans exhibit that encompasses the catholic church in their b & w documents. best blurb of all those who would read about the errors they witness and don't believe. an interesting twist ending to the loose ends. hercule poirot is cast as a smart cop undercover. the two big funny g ...

Figure 12: Sample generated using CaDDi, conditioned on a positive prompt. In each sample, the purple text indicates the prompt.

1128 References

- 1129 [1] Qwen2 technical report. 2024.
- 1130 [2] Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg.
1131 Structured denoising diffusion models in discrete state-spaces, 2023. URL <https://arxiv.org/abs/2107.03006>.
1132
- 1133 [3] Stella Biderman, Hailey Schoelkopf, Quentin Gregory Anthony, Herbie Bradley, Kyle O'Brien,
1134 Eric Hallahan, Mohammad Aflah Khan, Shivanshu Purohit, USVSN Sai Prashanth, Edward
1135 Raff, et al. Pythia: A suite for analyzing large language models across training and scaling. In
1136 *International Conference on Machine Learning*, pages 2397–2430. PMLR, 2023.
- 1137 [4] Yonatan Bisk, Rowan Zellers, Ronan Le Bras, Jianfeng Gao, and Yejin Choi. Piqa: Reasoning
1138 about physical commonsense in natural language. In *Thirty-Fourth AAAI Conference on*
1139 *Artificial Intelligence*, 2020.

- 1140 [5] Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis,
1141 and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in*
1142 *Neural Information Processing Systems*, 35:28266–28279, 2022.
- 1143 [6] Edoardo Cetin, Tianyu Zhao, and Yujin Tang. Large language models to diffusion finetuning,
1144 2025. URL <https://arxiv.org/abs/2501.15781>.
- 1145 [7] Ciprian Chelba, Tomas Mikolov, Mike Schuster, Qi Ge, Thorsten Brants, Phillipp Koehn, and
1146 Tony Robinson. One billion word benchmark for measuring progress in statistical language
1147 modeling, 2014. URL <https://arxiv.org/abs/1312.3005>.
- 1148 [8] Charlie Chen, Sebastian Borgeaud, Geoffrey Irving, Jean-Baptiste Lespiau, Laurent Sifre, and
1149 John Jumper. Accelerating large language model decoding with speculative sampling. *arXiv*
1150 *preprint arXiv:2302.01318*, 2023.
- 1151 [9] Mark Chen, Alec Radford, Rewon Child, Jeffrey Wu, Heewoo Jun, David Luan, and Ilya
1152 Sutskever. Generative pretraining from pixels. In *International conference on machine learning*,
1153 pages 1691–1703. PMLR, 2020.
- 1154 [10] Zixiang Chen, Huizhuo Yuan, Yongqian Li, Yiwen Kou, Junkai Zhang, and Quanquan Gu. Fast
1155 sampling via discrete non-markov diffusion models with predetermined transition time. In *The*
1156 *Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024.
- 1157 [11] Rewon Child, Scott Gray, Alec Radford, and Ilya Sutskever. Generating long sequences with
1158 sparse transformers, 2019. URL <https://arxiv.org/abs/1904.10509>.
- 1159 [12] Aakanksha Chowdhery, Sharan Narang, Jacob Devlin, Maarten Bosma, Gaurav Mishra, Adam
1160 Roberts, Paul Barham, Hyung Won Chung, Charles Sutton, Sebastian Gehrmann, et al. Palm:
1161 Scaling language modeling with pathways. *Journal of Machine Learning Research*, 24(240):
1162 1–113, 2023.
- 1163 [13] Christopher Clark, Kenton Lee, Ming-Wei Chang, Tom Kwiatkowski, Michael Collins, and
1164 Kristina Toutanova. Boolq: Exploring the surprising difficulty of natural yes/no questions.
1165 *arXiv preprint arXiv:1905.10044*, 2019.
- 1166 [14] Peter Clark, Isaac Cowhey, Oren Etzioni, Tushar Khot, Ashish Sabharwal, Carissa Schoenick,
1167 and Oyvind Tafjord. Think you have solved question answering? try arc, the ai2 reasoning
1168 challenge. *arXiv preprint arXiv:1803.05457*, 2018.
- 1169 [15] Tri Dao, Daniel Y. Fu, Stefano Ermon, Atri Rudra, and Christopher Ré. FlashAttention: Fast
1170 and memory-efficient exact attention with IO-awareness. In *Advances in Neural Information*
1171 *Processing Systems (NeurIPS)*, 2022.
- 1172 [16] Oscar Davis, Samuel Kessler, Mircea Petrache, İsmail İlkan Ceylan, Michael Bronstein, and
1173 Avishek Joey Bose. Fisher flow matching for generative modeling over discrete data, 2024.
1174 URL <https://arxiv.org/abs/2405.14664>.
- 1175 [17] Sander Dieleman, Laurent Sartran, Arman Roshannai, Nikolay Savinov, Yaroslav Ganin,
1176 Pierre H Richemond, Arnaud Doucet, Robin Strudel, Chris Dyer, Conor Durkan, et al. Continu-
1177 ous diffusion for categorical data. *arXiv preprint arXiv:2211.15089*, 2022.
- 1178 [18] Abhimanyu Dubey, Abhinav Jauhri, Abhinav Pandey, Abhishek Kadian, Ahmad Al-Dahle,
1179 Aiesha Letman, Akhil Mathur, Alan Schelten, Amy Yang, Angela Fan, et al. The llama 3 herd
1180 of models. *arXiv preprint arXiv:2407.21783*, 2024.
- 1181 [19] Leo Gao, Jonathan Tow, Baber Abbasi, Stella Biderman, Sid Black, Anthony DiPofi, Charles
1182 Foster, Laurence Golding, Jeffrey Hsu, Alain Le Noac’h, Haonan Li, Kyle McDonell, Niklas
1183 Muennighoff, Chris Ociepa, Jason Phang, Laria Reynolds, Hailey Schoelkopf, Aviya Skowron,
1184 Lintang Sutawika, Eric Tang, Anish Thite, Ben Wang, Kevin Wang, and Andy Zou. The language
1185 model evaluation harness, 07 2024. URL <https://zenodo.org/records/12608602>.
- 1186 [20] C Gardiner. Stochastic methods: A handbook for the natural and social sciences 2009.

- 1187 [21] Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi,
1188 and Yaron Lipman. Discrete flow matching, 2024. URL <https://arxiv.org/abs/2407.15595>.
1189
- 1190 [22] Shansan Gong, Shivam Agarwal, Yizhe Zhang, Jiacheng Ye, Lin Zheng, Mukai Li, Chenxin An,
1191 Peilin Zhao, Wei Bi, Jiawei Han, et al. Scaling diffusion language models via adaptation from
1192 autoregressive models. *arXiv preprint arXiv:2410.17891*, 2024.
- 1193 [23] Jiatao Gu, Yuyang Wang, Yizhe Zhang, Qihang Zhang, Dinghuai Zhang, Navdeep Jaitly,
1194 Josh Susskind, and Shuangfei Zhai. Dart: Denoising autoregressive transformer for scalable
1195 text-to-image generation, 2024. URL <https://arxiv.org/abs/2410.08159>.
- 1196 [24] Ishaan Gulrajani and Tatsunori B Hashimoto. Likelihood-based diffusion language models.
1197 *Advances in Neural Information Processing Systems*, 36, 2024.
- 1198 [25] Sizhuang He, Daniel Levine, Ivan Vrkic, Marco Francesco Bressana, David Zhang, Syed Asad
1199 Rizvi, Yangtian Zhang, Emanuele Zappala, and David van Dijk. Calmflow: Volterra flow
1200 matching using causal language models, 2024. URL <https://arxiv.org/abs/2410.05292>.
- 1201 [26] Zhengfu He, Tianxiang Sun, Kuanning Wang, Xuanjing Huang, and Xipeng Qiu. Diffusionbert:
1202 Improving generative masked language models with diffusion models, 2022. URL <https://arxiv.org/abs/2211.15029>.
1203
- 1204 [27] Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance. *arXiv preprint*
1205 *arXiv:2207.12598*, 2022.
- 1206 [28] Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models, 2020.
1207 URL <https://arxiv.org/abs/2006.11239>.
- 1208 [29] Ari Holtzman, Jan Buys, Li Du, Maxwell Forbes, and Yejin Choi. The curious case of neural
1209 text degeneration, 2020. URL <https://arxiv.org/abs/1904.09751>.
- 1210 [30] Emiel Hoogeboom, Alexey A Gritsenko, Jasmijn Bastings, Ben Poole, Rianne van den Berg,
1211 and Tim Salimans. Autoregressive diffusion models. *arXiv preprint arXiv:2110.02037*, 2021.
- 1212 [31] Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax
1213 flows and multinomial diffusion: Learning categorical distributions. *Advances in neural*
1214 *information processing systems*, 34:12454–12465, 2021.
- 1215 [32] Vincent Tao Hu and Björn Ommer. [mask] is all you need. *arXiv preprint arXiv:2412.06787*,
1216 2024.
- 1217 [33] Guokun Lai, Qizhe Xie, Hanxiao Liu, Yiming Yang, and Eduard Hovy. RACE: Large-scale
1218 ReAding comprehension dataset from examinations. In *Proceedings of the 2017 Conference on*
1219 *Empirical Methods in Natural Language Processing*, pages 785–794, Copenhagen, Denmark,
1220 September 2017. Association for Computational Linguistics. doi: 10.18653/v1/D17-1082. URL
1221 <https://aclanthology.org/D17-1082>.
- 1222 [34] Yaniv Leviathan, Matan Kalman, and Yossi Matias. Fast inference from transformers via
1223 speculative decoding. In *International Conference on Machine Learning*, pages 19274–19286.
1224 PMLR, 2023.
- 1225 [35] Yaron Lipman, Ricky T. Q. Chen, Heli Ben-Hamu, Maximilian Nickel, and Matt Le. Flow
1226 matching for generative modeling, 2023. URL <https://arxiv.org/abs/2210.02747>.
- 1227 [36] Anji Liu, Oliver Broadrick, Mathias Niepert, and Guy Van den Broeck. Discrete copula diffusion.
1228 *arXiv preprint arXiv:2410.01949*, 2024.
- 1229 [37] Aaron Lou, Chenlin Meng, and Stefano Ermon. Discrete diffusion modeling by estimating the
1230 ratios of the data distribution, 2024. URL <https://arxiv.org/abs/2310.16834>.
- 1231 [38] Julian McAuley and Jure Leskovec. Hidden factors and hidden topics: understanding rating
1232 dimensions with review text. In *Proceedings of the 7th ACM conference on Recommender*
1233 *systems*, pages 165–172, 2013.

- [39] Shen Nie, Fengqi Zhu, Chao Du, Tianyu Pang, Qian Liu, Guangtao Zeng, Min Lin, and Chongxuan Li. Scaling up masked diffusion models on text, 2025. URL <https://arxiv.org/abs/2410.18514>.
- [40] 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)*, pages 1525–1534, Berlin, Germany, August 2016. Association for Computational Linguistics. URL <http://www.aclweb.org/anthology/P16-1144>.
- [41] Yong-Hyun Park, Chieh-Hsin Lai, Satoshi Hayakawa, Yuhta Takida, and Yuki Mitsufuji. Jump your steps: Optimizing sampling schedule of discrete diffusion models. *arXiv preprint arXiv:2410.07761*, 2024.
- [42] Alec Radford, Jeff Wu, Rewon Child, David Luan, Dario Amodei, and Ilya Sutskever. Language models are unsupervised multitask learners. 2019.
- [43] Aditya Ramesh, Mikhail Pavlov, Gabriel Goh, Scott Gray, Chelsea Voss, Alec Radford, Mark Chen, and Ilya Sutskever. Zero-shot text-to-image generation. In *International conference on machine learning*, pages 8821–8831. Pmlr, 2021.
- [44] 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.
- [45] Maarten Sap, Hannah Rashkin, Derek Chen, Ronan LeBras, and Yejin Choi. Socialliqa: Commonsense reasoning about social interactions, 2019. URL <https://arxiv.org/abs/1904.09728>.
- [46] Yair Schiff, Subham Sekhar Sahoo, Hao Phung, Guanghan Wang, Sam Boshar, Hugo Dallarrea, Bernardo P de Almeida, Alexander Rush, Thomas Pierrot, and Volodymyr Kuleshov. Simple guidance mechanisms for discrete diffusion models. *arXiv preprint arXiv:2412.10193*, 2024.
- [47] Tianxiao Shen, Hao Peng, Ruoqi Shen, Yao Fu, Zaid Harchaoui, and Yejin Choi. Film: Fill-in language models for any-order generation. *arXiv preprint arXiv:2310.09930*, 2023.
- [48] Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis K Titsias. Simplified and generalized masked diffusion for discrete data. *arXiv preprint arXiv:2406.04329*, 2024.
- [49] Andy Shih, Dorsa Sadigh, and Stefano Ermon. Training and inference on any-order autoregressive models the right way. *Advances in Neural Information Processing Systems*, 35:2762–2775, 2022.
- [50] Yang Song, Jascha Sohl-Dickstein, Diederik P. Kingma, Abhishek Kumar, Stefano Ermon, and Ben Poole. Score-based generative modeling through stochastic differential equations, 2021. URL <https://arxiv.org/abs/2011.13456>.
- [51] Jacob Mitchell Springer, Suhas Kotha, Daniel Fried, Graham Neubig, and Aditi Raghunathan. Repetition improves language model embeddings, 2024. URL <https://arxiv.org/abs/2402.15449>.
- [52] Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design, 2024. URL <https://arxiv.org/abs/2402.05841>.
- [53] Jianlin Su, Murtadha Ahmed, Yu Lu, Shengfeng Pan, Wen Bo, and Yunfeng Liu. Roformer: Enhanced transformer with rotary position embedding. *Neurocomputing*, 568:127063, 2024.
- [54] Jaesung Tae, Hamish Ivison, Sachin Kumar, and Arman Cohan. Tess 2: A large-scale generalist diffusion language model, 2025. URL <https://arxiv.org/abs/2502.13917>.

- 1281 [55] Alexander Tong, Kilian Fatras, Nikolay Malkin, Guillaume Huguët, Yanlei Zhang, Jarrid Rector-
1282 Brooks, Guy Wolf, and Yoshua Bengio. Improving and generalizing flow-based generative mod-
1283 els with minibatch optimal transport, 2024. URL <https://arxiv.org/abs/2302.00482>.
- 1284 [56] Hugo Touvron, Thibaut Lavril, Gautier Izacard, Xavier Martinet, Marie-Anne Lachaux, Timo-
1285 thée Lacroix, Baptiste Rozière, Naman Goyal, Eric Hambro, Faisal Azhar, et al. Llama: Open
1286 and efficient foundation language models. *arXiv preprint arXiv:2302.13971*, 2023.
- 1287 [57] Hugo Touvron, Louis Martin, Kevin Stone, Peter Albert, Amjad Almahairi, Yasmine Babaei,
1288 Nikolay Bashlykov, Soumya Batra, Prajjwal Bhargava, Shruti Bhosale, Dan Bikel, Lukas
1289 Blecher, Cristian Canton Ferrer, Moya Chen, Guillem Cucurull, David Esiobu, Jude Fernandes,
1290 Jeremy Fu, Wenyin Fu, Brian Fuller, Cynthia Gao, Vedanuj Goswami, Naman Goyal, Anthony
1291 Hartshorn, Saghar Hosseini, Rui Hou, Hakan Inan, Marcin Kardas, Viktor Kerkez, Madian
1292 Khabsa, Isabel Kloumann, Artem Korenev, Punit Singh Koura, Marie-Anne Lachaux, Thibaut
1293 Lavril, Jenya Lee, Diana Liskovich, Yinghai Lu, Yuning Mao, Xavier Martinet, Todor Mihaylov,
1294 Pushkar Mishra, Igor Molybog, Yixin Nie, Andrew Poulton, Jeremy Reizenstein, Rashi Rungta,
1295 Kalyan Saladi, Alan Schelten, Ruan Silva, Eric Michael Smith, Ranjan Subramanian, Xiao-
1296 qing Ellen Tan, Binh Tang, Ross Taylor, Adina Williams, Jian Xiang Kuan, Puxin Xu, Zheng
1297 Yan, Iliyan Zarov, Yuchen Zhang, Angela Fan, Melanie Kambadur, Sharan Narang, Aurelien
1298 Rodriguez, Robert Stojnic, Sergey Edunov, and Thomas Scialom. Llama 2: Open foundation
1299 and fine-tuned chat models, 2023. URL <https://arxiv.org/abs/2307.09288>.
- 1300 [58] Dustin Tran, Keyon Vafa, Kumar Agrawal, Laurent Dinh, and Ben Poole. Discrete flows:
1301 Invertible generative models of discrete data. *Advances in Neural Information Processing*
1302 *Systems*, 32, 2019.
- 1303 [59] Nicolaas Godfried Van Kampen. *Stochastic processes in physics and chemistry*, volume 1.
1304 Elsevier, 1992.
- 1305 [60] A Vaswani. Attention is all you need. *Advances in Neural Information Processing Systems*,
1306 2017.
- 1307 [61] Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking
1308 discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.
- 1309 [62] Zhisheng Xiao, Karsten Kreis, and Arash Vahdat. Tackling the generative learning trilemma
1310 with denoising diffusion gans. *arXiv preprint arXiv:2112.07804*, 2021.
- 1311 [63] Minkai Xu, Tomas Geffner, Karsten Kreis, Weili Nie, Yilun Xu, Jure Leskovec, Stefano Ermon,
1312 and Arash Vahdat. Energy-based diffusion language models for text generation. *arXiv preprint*
1313 *arXiv:2410.21357*, 2024.
- 1314 [64] Jiacheng Ye, Zhihui Xie, Lin Zheng, Jiahui Gao, Zirui Wu, Xin Jiang, Zhenguo Li, and Lingpeng
1315 Kong. Dream 7b, 2025. URL <https://hkunlp.github.io/blog/2025/dream>.
- 1316 [65] Peiyuan Zhang, Guangtao Zeng, Tianduo Wang, and Wei Lu. Tynllama: An open-source small
1317 language model, 2024. URL <https://arxiv.org/abs/2401.02385>.
- 1318 [66] Kaiwen Zheng, Yongxin Chen, Hanzi Mao, Ming-Yu Liu, Jun Zhu, and Qinsheng Zhang.
1319 Masked diffusion models are secretly time-agnostic masked models and exploit inaccurate
1320 categorical sampling. *arXiv preprint arXiv:2409.02908*, 2024.
- 1321 [67] Zachary Ziegler and Alexander Rush. Latent normalizing flows for discrete sequences. In
1322 *International Conference on Machine Learning*, pages 7673–7682. PMLR, 2019.