DISCRETE DIFFUSION MODELS WITH INTERPOLATION-BASED CONTROLLABLE RESAMPLING

Anonymous authors
Paper under double-blind review

ABSTRACT

Discrete diffusion models form a powerful class of generative models across diverse data modalities, including graphs and text. However, existing approaches face fundamental limitations. Masked diffusion models suffer from irreversible errors if tokens are unmasked too early in the reverse process, while uniform diffusion models, despite enabling self-correction, often yield low-quality samples due to their strong reliance on intermediate latent states. To address these limitations, we introduce *IDDM*, an Interpolation-Based Discrete Diffusion Model that improves diffusion by reducing dependence on intermediate latent states. Central to *IDDM* is a controllable resampling mechanism that partially resets probability mass to the prior distribution, mitigating error accumulation and enabling token corrections. *IDDM* defines a generative process whose transitions interpolate between staying at the current state, resampling from a prior, and flipping toward the target state, while enforcing marginal consistency and fully decoupling training from inference. We benchmark our model against state-of-the-art discrete diffusion models on molecular graph and text generation tasks, showing competitive performance.

1 INTRODUCTION

Discrete diffusion models (Austin et al., 2021) have emerged as a powerful approach for generating structures across domains such as proteins (Gruver et al., 2023; Nisonoff et al., 2024), DNA sequences (Sarkar et al., 2024), and molecules (Vignac et al., 2022; Campbell et al., 2022), and are closing the gap to autoregressive models in language generation (Lou et al., 2023; Sahoo et al., 2025; Schiff et al., 2024). Similar to their continuous counterparts, the core idea is to define Markov transitions that gradually corrupt the data, and then learn a model to denoise it by parameterizing the reverse process. Common forward processes include uniform transitions (Austin et al., 2021; Schiff et al., 2024; Lou et al., 2023) and marginal transitions (Vignac et al., 2022; Qin et al., 2024) over the state space or masking, also known as absorbing-state diffusion, in which the state either remains unchanged or transitions to a mask token (Austin et al., 2021; Sahoo et al., 2024; Shi et al., 2024).

A fundamental limitation shared by standard discrete diffusion models (Austin et al., 2021; Vignac et al., 2022; Sahoo et al., 2024) is that specifying the forward process induces a *single, fixed* reverse kernel via Bayes's rule. This limits the flexibility to modify the generative process independently of the noising mechanism without re-training. For example, the forward process in masked diffusion either preserves a token or irreversibly replaces it with an absorbing mask state. Consequently, the corresponding reverse process does not support refinement: once a token is unmasked, it remains fixed throughout the entire generation, restricting the model from correcting earlier mistakes and leading to error accumulation (Sahoo et al., 2024; Wang et al., 2025; Shi et al., 2024). Uniform-state diffusion, in contrast, corrupts tokens by resampling from the vocabulary, and its reverse process, therefore, permits repeated token revisions. Although this process enables self-correction, uniform diffusion models still suffer from error accumulation (Boget, 2025), and are typically outperformed by absorbing-state diffusion approaches on language modeling tasks (Austin et al., 2021; Lou et al., 2023).

Recent remasking methods enable correction in masked diffusion by replacing previously predicted tokens with a dedicated mask token (Wang et al., 2025; Boget, 2025), allowing the model to revise them in subsequent denoising steps. However, extending this correction mechanism beyond absorbing priors is not trivial, since corrupted tokens may occupy ordinary data states without an explicit

Reviewers: please read the Reviewer Guidelines (iclr.cc/Conferences/2027/ReviewerGuidelines) and the AI Policy for Reviewers (iclr.cc/Conferences/2027/AIPolicyForReviewers). If you used AI to expand, edit, or polish your review, please provide the input text to the LLM. Better still, consider skipping the LLM and submitting your original text: we, and the authors, are much more interested in your unedited thoughts than in what an LLM has to say. AI-assisted or not, you are putting your name and reputation behind your review: LLM-generated falsehoods, hallucinations or misrepresentations are subject to disciplinary action, which may include desk-rejecting all papers you have authored.

indicator of corruption. Supporting arbitrary priors is particularly relevant for graph generation, where diffusion models are trained using empirical prior distributions over node and edge attributes to preserve graph sparsity (Vignac et al., 2022). In molecular graphs, a single incorrect atom or bond prediction can violate chemical constraints and invalidate the molecule, making the ability to revise earlier predictions especially valuable. Similarly, enabling controllable token revisions in uniform diffusion language models could improve their self-correction and help narrow the sample-quality gap to masked diffusion language models.

In this work, we propose *the Interpolation-Based Discrete Diffusion Model (IDDM)*, a novel generative model for discrete data that enables controllable revisions under arbitrary categorical priors. We derive transition probabilities from a different perspective: instead of defining a Markovian forward transition to reach a terminal distribution, we directly define the reverse, i.e., the generative distribution. To this end, our transitions make use of three different actions: *(i)* staying in the current state, *(ii)* transitioning to the goal state, and *(iii)* resampling from the prior distribution, see Fig. 1. By constraining the marginal distribution to match an interpolation from prior to the data distribution, we obtain *a family of valid posterior transitions*. The posteriors interpolate between a fully local update without random flips and a state-independent update that discards the current state, enabling us to trade off preservation of the current knowledge and stochasticity.

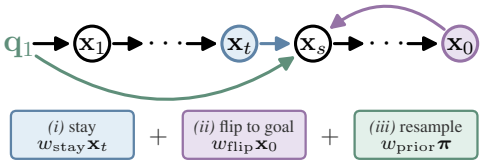


Figure 1: Graphical model of IDDM. In each step, our model consists of three actions: *(i)* stay in the current state, *(ii)* flip to the goal state, and *(iii)* sample from the prior. The model is trained without defining a forward process.

Contributions. Our contributions can be summarized as:

- **Interpolating reverse dynamics.** We propose the *Interpolation-Based Discrete Diffusion Model (IDDM)*, a simple discrete diffusion model with controllable resampling consisting of three different actions, i.e., stay in the current state, transition to the goal state, or resample from the prior. IDDM enables a principled notion of correction under arbitrary priors. We show that the resulting transitions, in general, correspond to a non-Markovian forward process, meaning their dynamics cannot be derived from a standard diffusion perspective.
- **Continuous-time ELBO.** We derive a continuous-time ELBO for IDDM under arbitrary categorical priors that explicitly accounts for both preserved and corrupted states. We further show that, under an absorbing prior without correction, our objective reduces to the weighted masked-token cross-entropy objective of standard masked diffusion language models.
- **Decoupled training and inference.** By enforcing the training marginal to adhere to an interpolation between the prior and the data distribution, IDDM enables training a single model, and deriving a family of valid marginal-consistent posterior transitions parametrized by a correction parameter that trades off preserving the current state against injecting stochasticity.
- **Empirical evaluation.** IDDM achieves state-of-the-art performance on molecule generation: it attains strong validity and FCD on GuacaMol, and outperforms DeFoG (Qin et al., 2024) on both MOSES and QM9 while only using 10% of the generation steps. In language generation, IDDM substantially narrows the generation-quality gap between uniform-state and masked-state diffusion language models and performs competitively with recent remasking and corrector-sampler approaches.

2 Discrete Diffusion Models

Notations. For clarity and ease of presentation, we use x to denote a single discrete random variable taking values in $\{1, \dots, K\}$. In sequence modeling, x represents a single token, while in graph modeling, it can represent a single node or a single edge in the graph. We denote the corresponding one-hot encoding by $\mathbf{x} \in \{0, 1\}^K$. We denote by $\mathrm{Cat}(\cdot; \mathbf{y})$ the categorical distribution with probability vector $\mathbf{y} \in \Delta^{K-1}$, where Δ^{K-1} is the K-simplex.

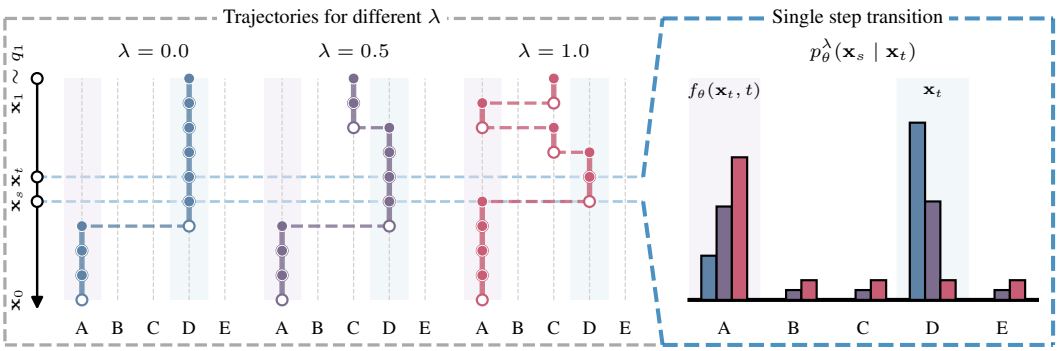


Figure 2: Overview of the generative process of IDDM. Initial samples are drawn from the prior $\boldsymbol{\pi}$. In each denoising step $\mathbf{x}_s \mid \mathbf{x}_t$, our model constructs the posterior $p_\theta^\lambda(\mathbf{x}_s \mid \mathbf{x}_t)$. Increasing λ increases the stochasticity and state transitions by shifting probability mass from the current to the predicted and random states, as visualized in the bar plot.

Diffusion models learn a Markov chain that involves a forward process to perturb the data $\mathbf{x} \sim q_0(\mathbf{x})$ to increasingly noisy latent variables $\mathbf{x}_t$, for $t \in [0, 1]$, and a trained reverse process. The forward process $q(\mathbf{x}_t \mid \mathbf{x}_s)$ is expressed under the Markovian assumption, and the induced marginal for every latent variable $\mathbf{x}_t$ can be defined as an interpolation between the clean data and a noisy prior $q_1 = \mathrm{Cat}(\cdot; \boldsymbol{\pi})$ (Sahoo et al., 2024):

$$q(\mathbf{x}_t \mid \mathbf{x}) \;=\; \mathrm{Cat}(\mathbf{x}_t; (1 \;-\; \gamma_t)\boldsymbol{\pi} \;+\; \gamma_t \mathbf{x}) \;, \tag{1}$$

where $\gamma_t \in [0, 1]$ is a strictly decreasing function in t, and unless stated otherwise, $\mathbf{x}_t$ is represented in one-hot form. The parametrized reverse process $p_\theta(\mathbf{x}_s \mid \mathbf{x}_t), s < t$ is trained to optimize a variational lower bound on the log-likelihood:

$$\log\; p_\theta(\mathbf{x}) \geq \mathbb{E}_{q(\mathbf{x}_{t(1)}, \ldots, \mathbf{x}_{t(T)} \mid \mathbf{x})} \left[\log \frac{p_\theta(\mathbf{x}, \mathbf{x}_{t(1)}, \ldots, \mathbf{x}_{t(T)})}{q(\mathbf{x}_{t(1)}, \ldots, \mathbf{x}_{t(T)} \mid \mathbf{x})} \right], \tag{2}$$

where $t(i) \;=\; i/T$, T is the number of diffusion steps, and the reverse Markov process $p_\theta(\mathbf{x}, \mathbf{x}_{t(1:T)}) = p(\mathbf{x}_{t(T)}) \big(\prod_{i=2}^{T} p_\theta(\mathbf{x}_{t(i-1)} \mid \mathbf{x}_{t(i)}) \big) p_\theta(\mathbf{x} \mid \mathbf{x}_{t(1)})$, gradually denoises the latent states towards the data distribution. There exist two popular variants for interpolating discrete diffusion models (Sahoo et al., 2024): uniform diffusion models with a stationary distribution $\boldsymbol{\pi} = \mathbf{1}/K$, where $\mathbf{1}$ is a K-dimensional column vector of all ones, and masked diffusion models, where the space is augmented with a special mask token, and the prior distribution is set to $\boldsymbol{\pi} = \mathbf{m}$. Marginal distributions over node and edge attributes, computed empirically from the data distribution by aggregating attribute frequencies across all nodes and edges, respectively, have proven to be more suitable for graph generation than uniform noise (Vignac et al., 2022).

Standard discrete diffusion models tightly couple the reverse process to a fixed forward corruption mechanism. The forward process is defined as a Markov chain $q(\mathbf{x}_t \mid \mathbf{x}_s)$, and the reverse transitions can be determined via Bayes' rule: $q(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \frac{q(\mathbf{x}_t \mid \mathbf{x}_s) q(\mathbf{x}_s \mid \mathbf{x})}{q(\mathbf{x}_t \mid \mathbf{x})}$, where $q(\mathbf{x}_t \mid \mathbf{x}_s) = q(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x})$ follows from the Markov property. As a result, the reverse dynamics are determined once the forward kernel and the noise schedule are fixed. This limits the extent to which reverse updates can be modified independently of the noising process and, consequently, can degrade the sample quality. For example, in masked diffusion models, the forward process replaces tokens with a mask token such that, once masked, they remain masked for all subsequent steps. This constrains the corresponding reverse process: once a token is unmasked, it cannot be updated in later steps. This leads to systematic error accumulation as early mistakes cannot be recovered. Uniform diffusion models, on the other hand, inherently enable self-correction since the latent states $\mathbf{x}_t$ can be continuously updated throughout the reverse process. However, this mechanism can introduce errors, as correctly predicted tokens can be randomly resampled from the uniform prior.

3 Methodology

In this section, we propose the *Interpolation-Based Discrete Diffusion Model (IDDM)*, a novel generative model for discrete data that enables controlled resampling during inference and improves

token correction. We provide an overview of IDDM in Fig. 2. Our goal is to learn the target distribution q_0. Similar to diffusion models, we define a sequence of latent variables $\mathbf{x}_t$ that interpolate between the data at $t = 0$ and a prior distribution at $t = 1$. However, unlike diffusion models, we do not explicitly define forward, i.e., noising process $q(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x})$, for $s < t$. Instead, we define the reverse-time generative process directly.

Standard discrete diffusion models (Sahoo et al., 2024; Schiff et al., 2024) induce a single reverse transition via Bayes' rule, leaving no freedom to control the stochasticity of the reverse process independently of the forward kernel. To overcome this limitation, we directly parametrize the reverse transition as a mixture of three interpretable actions: (i) *stay* in the current state, (ii) *flip* to the clean data $\mathbf{x}$, or (iii) *resample* from the prior $\boldsymbol{\pi}$. This yields an inference-time mechanism that flexibly controls how much we preserve the state $\mathbf{x}_t$ and resample from the prior. More specifically:

$$q(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{x}_s; w_{\text{stay}}\mathbf{x}_t + w_{\text{prior}}\boldsymbol{\pi} + w_{\text{flip}}\mathbf{x}\Big), \tag{3}$$

with $w_{\text{stay}}, w_{\text{flip}}, w_{\text{prior}} \geq 0$ and $w_{\text{stay}} + w_{\text{flip}} + w_{\text{prior}} = 1$. The three weights determine how much probability mass is assigned to preserving the current state, forgetting the current state by resampling from the prior, or moving directly toward the clean data token. See Fig. 1 for an illustration.

Marginal distribution. We constrain the marginal distribution to interpolate between the prior distribution and the goal state:

$$q(\mathbf{x}_t \mid \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{x}_t;\ (1-\gamma_t)\boldsymbol{\pi} + \gamma_t\mathbf{x}\Big), \tag{4}$$

with $\gamma_t \in [0, 1]$, $\gamma_1 = 0$ and $\gamma_0 = 1$. First, this form generalizes the marginals of both masked and uniform diffusion models as special cases (Sahoo et al., 2024). Second, this choice enables direct sampling of $\mathbf{x}_t$ at arbitrary t without simulating a forward chain. Crucially, it provides a common marginal path that is independent of the inference-time parameters $w_{\text{stay}}, w_{\text{flip}}, w_{\text{prior}}$. Consequently, different reverse kernels can share the same training distribution over noisy latents, allowing us to train a denoising model while varying the amount of resampling at inference during sampling.

Parametrization of transition weights. Our goal is to select the transition weights in Eq. (3) such that the posterior preserves the marginal path in Eq. (4). That is, for every $s < t$, we require the marginal consistency condition

$$\sum_{\mathbf{x}_t} q(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) q(\mathbf{x}_t \mid \mathbf{x}) = q(\mathbf{x}_s \mid \mathbf{x}). \tag{5}$$

Substituting into the marginal consistency condition yields:

$$\mathrm{Cat}\big(\mathbf{x}_s;\ w_{\text{stay}}\,\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})}[\mathbf{x}_t] + w_{\text{prior}}\boldsymbol{\pi} + w_{\text{flip}}\mathbf{x}\big) = \mathrm{Cat}(\mathbf{x}_s; (1-\gamma_s)\boldsymbol{\pi} + \gamma_s\mathbf{x}), \tag{6}$$

where $\mathbb{E}[\mathbf{x}_t] = (1-\gamma_t)\boldsymbol{\pi} + \gamma_t\mathbf{x}$. This yields the following constraints:

$$\begin{aligned} &(1)\ w_{\text{stay}}(1-\gamma_t) + w_{\text{prior}} = 1 - \gamma_s, \quad \text{and} \\ &(2)\ w_{\text{stay}}\gamma_t + w_{\text{flip}} = \gamma_s. \end{aligned} \tag{7}$$

These constraints provide a family of valid posteriors that are consistent with the same marginal path, which we parametrize with a forgetting parameter $\lambda \in [0, 1]$ that interpolates between a purely local update ($\lambda = 0$) and a global reset to the marginal at time s ($\lambda = 1$), as we show in Theorem 3.1.

Theorem 3.1. *Let $\gamma : [0, 1] \to [0, 1]$ satisfy $\gamma_0 = 1$ and $\gamma_1 = 0$, and assume γ is monotonically decreasing. For any $0 \leq s < t \leq 1$, define*

$$\gamma_{s|t} := \frac{1-\gamma_s}{1-\gamma_t} \in [0, 1]. \tag{8}$$

Then, for every $\lambda \in [0, 1]$, the weights

$$w_{\text{stay}} := (1-\lambda)\,\gamma_{s|t}, \tag{9}$$

$$w_{\text{prior}} := \lambda\,(1-\gamma_s), \tag{10}$$

$$w_{\text{flip}} := (1-\lambda)\,(1-\gamma_{s|t}) + \lambda\,\gamma_s \tag{11}$$

satisfy the marginal constraints in Eq. (7). Therefore, the reverse transition in Eq. (3) can be equivalently written as

$$q^{\lambda}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{x}_s; (1-\lambda)\gamma_{s|t}\,\mathbf{x}_t + \lambda(1-\gamma_s)\boldsymbol{\pi} + \Big(1 - \big((1-\lambda)\gamma_{s|t} + \lambda(1-\gamma_s)\big)\Big)\mathbf{x}\Big)$$

and its marginal satisfies $\sum_{\mathbf{x}_t} q^{\lambda}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x})\, q(\mathbf{x}_t \mid \mathbf{x}) = q(\mathbf{x}_s \mid \mathbf{x}), \quad$ *for all* $s < t$.

We provide a proof for Theorem 3.1 in Section A.

Special cases. The parameter λ acts as a forget mechanism controlling how much probability mass is taken from the current state $\mathbf{x}_t$, and how much is resampled independently from the marginal. Fig. 3 visualizes the transition distributions $q^{\lambda}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x})$ for increasing values of λ. At $\lambda = 0$, our model reduces to two choices, i.e., either remain in the current state, or flip to the data token. More specifically,

$$q^{\lambda=0}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{x}_s; \gamma_{s|t}\,\mathbf{x}_t + \Big(1 - \gamma_{s|t}\Big)\mathbf{x}\Big).$$

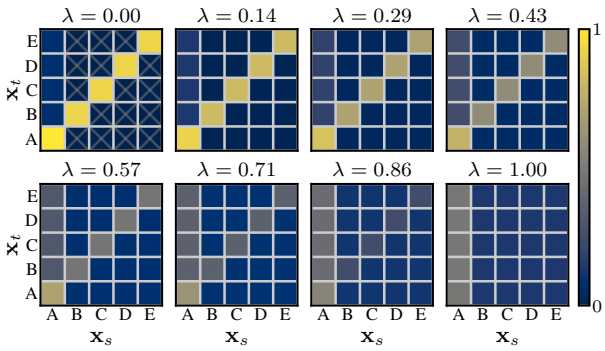


Figure 3: Visualization of $q^{\lambda}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x} = \mathrm{A})$ for different values of λ. Low λ preserves the current state with high probability. As λ increases, the mass progressively shifts toward resampling from the prior (uniform in this example).

These dynamics define an absorbing diffusion process directed toward the data token $\mathbf{x}$. In particular, when $\mathbf{x}_t = \mathbf{x}$, the transition distribution collapses to $q^{\lambda=0}(\mathbf{x}_s \mid \mathbf{x}_t = \mathbf{x}, \mathbf{x}) = \mathrm{Cat}(\mathbf{x}_s; \mathbf{x})$ indicating that $\mathbf{x}$ is absorbing and every other state $\mathbf{x}_t \neq \mathbf{x}$ remains non-absorbing, with a nonzero probability of transitioning to $\mathbf{x}$. This behavior is reflected in Fig. 3, where probability mass is concentrated along the diagonal and the column corresponding to the data point $\mathbf{x}$. For $\lambda = 1$, the posterior reduces to:

$$q^{\lambda=1}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \quad \mathrm{Cat}\Big(\mathbf{x}_s; (1-\gamma_s)\boldsymbol{\pi} + \gamma_s\mathbf{x}\Big) = q(\mathbf{x}_s \mid \mathbf{x}).$$

In this case, the latent variables $\mathbf{x}_t$ and $\mathbf{x}_s$ are conditionally independent given $\mathbf{x}$, i.e., the state $\mathbf{x}_t$ is entirely discarded, and the transitions reset to the marginal distribution. This is illustrated in Fig. 3 where all rows collapse to the same distribution, reflecting complete forgetting of the current state $\mathbf{x}_t$.

Remark on the forward process. Our approach does not require instantiating a Markovian noising process $q(\mathbf{x}_t \mid \mathbf{x}_s)$, for $s < t$, following standard diffusion methods. Instead, we directly sample the noisy latents from the marginal distribution in equation 4. For completeness, however, we derive the underlying forward transition step of IDDM via Bayes' rule, $q^{\lambda}(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x}) \propto q^{\lambda}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) q(\mathbf{x}_t \mid \mathbf{x})$. This induces a non-Markovian forward kernel of the form:

$$q^{\lambda}(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{x}_t; \alpha_{\mathbf{x}_s}\mathbf{m}_t + (1-\alpha_{\mathbf{x}_s})\mathbf{x}_s\Big)\,, \tag{12}$$

where $\mathbf{m}_t = (1-\gamma_t)\boldsymbol{\pi} + \gamma_t\mathbf{x}$ is the marginal probability vector, i.e., $q(\mathbf{x}_t \mid \mathbf{x}) = \mathrm{Cat}(\mathbf{x}_t; \mathbf{m}_t)$, and $\alpha_{\mathbf{x}_s} \in [0, 1]$ is a state-dependent resampling probability, see Section B for details. This transition admits an intuitive interpretation: with probability $\alpha_{\mathbf{x}_s}$, the latent $\mathbf{x}_t$ is resampled from the marginal distribution, otherwise it retains the previous state $\mathbf{x}_s$ with probability $1-\alpha_{\mathbf{x}_s}$. Crucially, our marginal is independent of the resampling scheduler λ, allowing decoupled training and sampling.

Expected total number of state transitions. In the following, we analyze the expected number of state transitions induced by IDDM, i.e., $\mathbf{x}_{t(k-1)} \neq \mathbf{x}_{t(k)}$, for $k \in \{1, \cdots, T\}$, along a generation trajectory of T steps. Let

$$C_k = \mathbf{1}\Big[\mathbf{x}_{t(k-1)} \neq \mathbf{x}_{t(k)}\Big]$$

Algorithm 1 Training IDDM

Inputs: Data distribution $\mathbf{q}_0(\mathbf{x})$; denoiser f_θ; noise schedule $\gamma(\cdot)$; steps T; learning rate η
Output: trained parameters θ
$\mathrm{d}t = 1/T$
While not converged **do**
  $\mathbf{x} \sim \mathbf{q}_0(\mathbf{x})$
  $t \sim \mathcal{U}(\mathrm{d}t, 1), \quad s = t - \mathrm{d}t$
  $\mathbf{x}_t \sim q(\mathbf{x}_t \mid \mathbf{x})$
  $q(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \gamma_{s|t}\,\mathbf{x}_t + (1-\gamma_{s|t})\,\mathbf{x}$
  $p_\theta(\mathbf{x}_s \mid \mathbf{x}_t) = \gamma_{s|t}\,\mathbf{x}_t + (1-\gamma_{s|t})\,f_\theta(\mathbf{x}_t, t)$
  $\theta = \theta - \eta \nabla_\theta\, \mathcal{L}_{\text{diff}}$
end while

Algorithm 2 Sampling with IDDM

Inputs: Denoiser f_θ; noising scheduler $\gamma(\cdot)$; forgetting scheduler $\lambda(\cdot)$; steps T; prior $\boldsymbol{\pi}$
Output: $\mathbf{x}_0$
$\mathbf{x}_T \sim q_1(\cdot; \boldsymbol{\pi})$
for $i = T, T-1, \ldots, 1$ **do**
  $t = i/T, \quad s = (i-1)/T$
  $\gamma_{s|t} = (1-\gamma_s)/(1-\gamma_t)$, $\lambda_t = \lambda(t)$
  $w_{\text{stay}} = (1-\lambda_t)\gamma_{s|t}, \quad w_{\text{prior}} = \lambda_t(1-\gamma_s)$
  $w_{\text{flip}} = 1 - (w_{\text{stay}} + w_{\text{prior}})$
  $\mathbf{x}_s \sim \mathrm{Cat}(\mathbf{x}_s;\, w_{\text{stay}}\mathbf{x}_t + w_{\text{prior}}\boldsymbol{\pi}$
    $+\; w_{\text{flip}} f_\theta(\mathbf{x}_t, t))$
end for

denote the indicator of a state change at step k. The expected total number of transitions is given by:

$$\mathbb{E}_\lambda\big[N_T\big] = \mathbb{E}_\lambda\left[\sum_{k=1}^{T} C_k\right] = \sum_{k=1}^{T} \mathbb{E}_\lambda\Big[q^\lambda(C_k = 1 \mid \mathbf{x}_{t(k)})\Big].$$

In Section E, we show that this expectation admits the form $\mathbb{E}_\lambda\big[N_T\big] = A\lambda + B$, where $A, B \geq 0$. This implies that the expected number of state transitions grows linearly with the resampling coefficient λ. In practice, however, the model $f_\theta(\mathbf{x}_t, t)$ does not perfectly preserve the marginal, therefore the distribution of $\mathbf{x}_{t(k)}$ can drift with λ. We investigate this effect in Section 4.3.

3.1 Generative Distribution and Optimization

As the true sample $\mathbf{x}$ is not known during inference, we train a neural network as a surrogate. We define the generative distribution as follows:

$$p_\theta^\lambda(\mathbf{x}_s \mid \mathbf{x}_t) = \mathrm{Cat}\Big(\mathbf{x}_s; (1-\lambda)\big[\gamma_{s|t}\,\mathbf{x}_t + (1-\gamma_{s|t})\mathbf{x}_\theta\big] + \lambda\big[(1-\gamma_s)\boldsymbol{\pi} + \gamma_s \mathbf{x}_\theta\big]\Big), \tag{13}$$

where $\mathbf{x}_\theta := f_\theta(\mathbf{x}_t, t)$ is a neural network prediction. As with standard diffusion models, we aim to optimize the log-likelihood $\log p_\theta^\lambda(\mathbf{x})$. However, due to the intractability of the marginalization, we derive the evidence lower bound (ELBO):

$$\underbrace{\mathbb{E}_{q^\lambda(\mathbf{x}_{t(1)}|\mathbf{x})}\big[\log p_\theta^\lambda(\mathbf{x} \mid \mathbf{x}_{t(1)})\big]}_{-\mathcal{L}_{\text{rec}}} - \underbrace{\sum_{i=2}^{T} \mathbb{E}_{q^\lambda(\mathbf{x}_{t(i)}|\mathbf{x})}\Big[\mathcal{L}_{\text{diff}}^{(i)}\Big]}_{\mathcal{L}_{\text{diff}}} - \underbrace{\mathbb{E}_{q_0(\mathbf{x})}\Big[\mathrm{KL}\Big(q(\mathbf{x}_{t(T)} \mid \mathbf{x}) \,\Big\|\, p(\mathbf{x}_{t(T)})\Big)\Big]}_{\mathcal{L}_{\text{prior}}},$$

where

$$\mathcal{L}_{\text{diff}}^{(i)} = \mathrm{KL}\big(q^\lambda(\mathbf{x}_{t(i-1)} \mid \mathbf{x}_{t(i)}, \mathbf{x}) \,\big\|\, p_\theta^\lambda(\mathbf{x}_{t(i-1)} \mid \mathbf{x}_{t(i)})\big)\,.$$

The ELBO is maximized when the KL divergences are minimized, i.e., when the neural network f_θ predicts $\mathbf{x}$. As the ELBOs share the same optimum across all λ when the model perfectly predicts $\mathbf{x}$, we only need to train one model, allowing us to change λ only during inference (see Table 9). We provide the training and sampling algorithms of IDDM in Algorithm 1 and Algorithm 2, respectively.

3.2 Continuous-Time ELBO

In the limit $T \to \infty$, the ELBO reduces to the following continuous-time objective:

$$\mathcal{L}_{\text{CT}} = \int_0^1 \beta_t\, \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})}\left[\mathbf{1}\{\mathbf{x}_t = \mathbf{x}\}\big(1 - \mathbf{x}^\top \mathbf{x}_\theta\big) + \mathbf{1}\{\mathbf{x}_t \neq \mathbf{x}\}\big(-\mathbf{x}_t^\top \mathbf{x}_\theta - \log(\mathbf{x}^\top \mathbf{x}_\theta)\big)\right] dt \tag{14}$$

with $\beta_t = -\frac{\dot{\gamma}_t}{1-\gamma_t}$. A full derivation is provided in Section C. We note that MDLM (Sahoo et al., 2024) arises as a special case of IDDM. Specifically, it corresponds to the reconstruction setting where $\lambda = 0$, prior $\boldsymbol{\pi} = \mathbf{m}$, and using the same noising schedule γ_t. Under these settings, we show in Section D that the reverse kernel of IDDM coincides exactly with that of MDLM, and that the continuous-time ELBO in Eq. (14) reduces to a weighted cross-entropy loss over the masked tokens.

Table 2: Large molecule generation performance. Best scores in **bold**, second best underlined. Baseline results are taken from the corresponding papers.

	GuacaMol				MOSES						
Model	Val. ↑	V.U. ↑	V.U.N. ↑	FCD Score ↑	Val. ↑	Unique. ↑	Novelty ↑	Filters ↑	FCD ↓	SNN ↑	Scaf ↑
Training set	100.0	100.0	0.0	92.8	100.0	100.0	0.0	100.0	0.01	0.64	99.1
DiGress (Vignac et al., 2022)	85.2	85.2	85.1	68.0	85.7	**100.0**	95.0	97.1	**1.19**	0.52	14.8
DisCo (Xu et al., 2024)	86.6	86.6	86.5	59.7	88.3	**100.0**	**97.7**	95.6	1.44	0.50	15.1
Cometh (Siraudin et al., 2024)	98.9	98.9	97.6	72.7	90.5	99.9	92.6	99.1	1.27	0.54	**16.0**
DeFoG (Qin et al., 2024)	99.0	99.0	**97.9**	73.8	**92.8**	99.9	92.1	98.9	1.95	0.55	14.4
IDDM (Ours)	**99.6**	**99.5**	97.8	**78.3**	90.7	99.9	90.2	**99.4**	1.58	**0.56**	13.2
DeFoG (Qin et al., 2024) (10% steps)	91.7	91.7	91.2	57.9	83.9	**99.9**	**96.9**	96.5	1.87	0.50	**23.5**
IDDM (Ours) (10% steps)	**98.1**	**98.0**	**97.5**	**77.6**	**87.1**	**99.9**	94.3	**98.9**	**1.14**	**0.53**	18.5

Extension to graphs and sequences. So far, we have considered a single-token data point $\mathbf{x}$. IDDM can be readily applied to arbitrary discrete domains and accommodates arbitrary prior distributions. In this paper, we apply IDDM to both graph and sequence generation tasks. We represent a graph $G = (\mathbf{X}, \mathbf{E})$ with N nodes, where $\mathbf{X} \in \mathbb{R}^{N \times K_1}$ is a matrix of one-hot encoded representations of each node $\mathbf{x}_i \in \{0,1\}^{K_1}$, and $\mathbf{E} \in \mathbb{R}^{N \times N \times K_2}$ groups the one-hot encoding of each edge $\mathbf{e}_{ij} \in \{0,1\}^{K_2}$. Here, K_1 and K_2 denote the number of possible node and edge attributes, respectively. We adopt empirical marginal prior distributions over nodes and edges, as they have proved effective in prior graph diffusion approaches (Vignac et al., 2022; Qin et al., 2024). For sequence generation, a sequence is represented as $\mathbf{S} \in \mathbb{R}^{L \times K}$, where L is the sequence length and K is the vocabulary size.

4 Experiments

In this section, we discuss our empirical evaluation across various benchmarks. Our goal is to compare our generative model to previous discrete diffusion approaches and investigate the effect of random flips to the prior distribution. We benchmark on two tasks: graph generation in Section 4.1 and text generation in Section 4.2. Additional experimental details are provided in Section F.

4.1 Graph Generation

Experimental setup. For graph generation, we parametrize our network with the graph transformer of Vignac et al. (2022), using Relative Random Walk Probabilities (RRWP) (Ma et al., 2023; Qin et al., 2024) to encode structural properties. The network predicts and generates node and edge labels (see Section 3.1). We provide additional details and hyperparameters in Section F.3.

Table 1: QM9 without hydrogen generation performance. Best results in **bold**, second best underlined. Baseline results are taken from the original papers.

Model	Val. ↑	Relaxed Val. ↑	Unique ↑	FCD ↓
Training set	99.3	99.5	99.2	0.03
DiGress (Vignac et al., 2022)	99.0	-	96.2	-
DisCo (Xu et al., 2024)	99.3	-	-	-
Cometh (Siraudin et al., 2024)	**99.6**	-	**96.8**	0.25
DeFoG (Qin et al., 2024)	99.3	99.4	96.3	**0.12**
IDDM (Ours)	99.3	**99.5**	96.3	0.13
DeFoG (Qin et al., 2024) (10% steps)	98.9	99.2	96.2	0.26
IDDM (Ours) (10% steps)	**99.3**	**99.5**	**96.5**	**0.10**

Small molecules. We evaluate IDDM on the QM9 dataset (Wu et al., 2018) (without hydrogen) and report results in Table 1. Overall, IDDM faithfully learns the data distribution, achieving a validity of 99.3%, matching the training distribution, and state-of-the-art FCD results with $\lambda = 0$ and a time distortion scheduler with $\rho = 4$, see Section G.2.2 for more details.

Large molecules. We evaluate IDDM on GuacaMol (Brown et al., 2019) and MOSES (Polykovskiy et al., 2020). We follow the standard data splits and provide our experimental results in Table 2. We observe that our model excels in FCD, SNN, and Filters while maintaining a high validity. Notably, our model with reduced steps (10% of steps) outperforms DeFoG with reduced steps across most metrics. Also, the full-step model achieves a higher validity than DeFoG. For these results, we used $\rho = 4$ for both datasets, and $\lambda = 0.4$ for MOSES and $\lambda = 0.2$ for GuacaMol.

4.2 Text Generation

Experimental setting. We train IDDM on the OpenWebText (OWT) dataset (Gokaslan & Cohen, 2019), using the same data splits, tokenization, and model configurations as Wang et al. (2025). We

train two variants, IDDM (Mask) and IDDM (Uniform), both with $\lambda = 0$. At sampling, we vary the λ scheduler according to either a linear decay scheduler, $\lambda(t) = \lambda_{\max}(1-t)$, a cosine annealing scheduler, $\lambda(t) = \frac{\lambda_{\max}}{2}[1+\cos(\pi t)]$ or the loop schedule as proposed in Wang et al. (2025). We evaluate generation quality using generative perplexity (Gen PPL), computed with GPT-2 Large, and measure sentence entropy to assess output diversity.

Sample quality. In Table 3, we report generative perplexity and entropy on OWT. We include $\lambda = 0$ as the no-correction baseline. Across both priors, structured correction λ schedules consistently outperform the no-correction baseline while maintaining comparable entropy. Under absorbing priors, IDDM (Absorb) achieves the lowest perplexity at 128 and 256 steps and remains competitive at 512 steps. The improvements are particularly pronounced under a uniform prior, where IDDM (Uniform) substantially outperforms SEDD Uniform and GIDD across different sampling budgets. We note that GIDD (Von Rütte et al., 2025) combines masking and uniform noise through a time-dependent mixing distribution to improve text generation. By adjusting λ_t during generation, IDDM enables effective inference-time correction under both masking and uniform priors, substantially narrowing the gap between uniform and masked diffusion language models.

Table 3: OWT sample quality.

	Gen PPL. ↓			Entropy ↑		
Model	128	256	512	128	256	512
SEDD Absorb (Lou et al., 2023)	119.2	110.1	107.2	**5.7**	**5.6**	**5.6**
D3PM Absorb (Austin et al., 2021)	–	–	72.9	–	–	5.4
MDLM (Sahoo et al., 2024)	61.5	55.8	53.0	5.5	5.5	5.5
MDLM+FB (Campbell et al., 2022)	42.8	39.6	37.1	5.4	5.4	5.4
MDLM+DFM (Gat et al., 2024)	37.9	26.5	23.3	5.3	5.3	5.2
ReMDM (Wang et al., 2025)	42.5	30.5	**21.1**	5.4	5.3	5.2
KLASS (Kim et al., 2026)	–	–	45.5	–	–	5.4
IDDM Absorb ($\lambda = 0$)	46.5	42.8	42.5	5.4	5.3	5.3
IDDM Absorb (cosine)	37.2	39.9	22.8	5.3	5.3	5.1
IDDM Absorb (linear)	35.0	**23.8**	24.8	5.3	5.1	5.2
IDDM Absorb (loop)	**32.0**	32.7	22.7	5.4	5.3	5.2
SEDD Uniform (Lou et al., 2023)	–	103.4	100.4	–	**5.6**	**5.6**
GIDD (Von Rütte et al., 2025)	95.1	80.5	76.9	5.0	4.9	4.9
IDDM Uniform ($\lambda = 0$)	82.7	81.0	80.4	**5.5**	5.5	5.5
IDDM Uniform (cosine)	**53.0**	**51.3**	**41.7**	5.3	5.3	5.3
IDDM Uniform (linear)	57.2	56.1	46.2	5.4	5.3	5.3
IDDM Uniform (loop)	73.9	59.4	49.9	**5.5**	5.4	5.4

4.3 Effect of λ on the state transitions

Intuitively, increasing λ introduces more transitions, i.e., $\mathbf{x}_t \neq \mathbf{x}_s$, during sampling as the weights w_{flip} and w_{prior} increase. In the extreme case $\lambda = 0$, the transition in equation 3 has at most one transition to the goal state. In practice, however, our trained network is not guaranteed to consistently predict the same class, which can lead to additional transitions. To quantify this effect, we measure the average number of node and edge transitions over 50 sampling steps on QM9, see Fig. 4. We find that the number of transitions increases linearly with λ, starting close to 0. Note that the average number of transitions for $\lambda = 0$ is well below 1, as we have a highly imbalanced class distribution and our sampling algorithm starts from the marginal distribution. We show its empirical effect in Fig. 5.

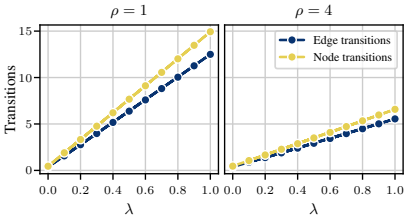


Figure 4: Number of transitions across different λ.

5 Discussion

Existing absorbing-state diffusion models provide a natural correction mechanism: previously revealed tokens can be returned to a dedicated mask state, allowing the model to revise them in subsequent steps (Wang et al., 2025; Gat et al., 2024; Campbell et al., 2022). Such a notion of correction is not directly defined for arbitrary categorical priors, as corrupted tokens may occupy ordinary data states without an explicit indicator of corruption.

In this work, we propose IDDM, a framework for inference-time correction under arbitrary categorical priors. We derive a parameterized family of reverse kernels that preserve the prescribed diffusion marginals while sharing a single trained denoiser. This construction supports correction under both absorbing and non-absorbing priors without changing the training marginals or retraining the model. Its correction behavior can be adapted to the task and computational budget through inexpensive inference-time tuning. Our experiments demonstrate effective correction under uniform and empirical marginal priors: IDDM substantially narrows the generation-quality gap between uniform and

masked diffusion language models and achieves state-of-the-art molecular generation performance, particularly in the low-step regime.

We limit our experiments to molecule and text generation; however, IDDM can be applied to other discrete domains, including protein (Campbell et al., 2024) and DNA sequence generation (Stark et al., 2024). Also, we limit the language experiments to absorbing and uniform priors; structured priors can be explored to further improve sample quality (Von Rütte et al., 2025). Future work could investigate guidance and other conditioning techniques. Finally, it would be promising to explore alternative training techniques based on variance reduction or distillation (Sahoo et al., 2025).

6 Related Work

Discrete Diffusion models. Diffusion models were first introduced for continuous data (Sohl-Dickstein et al., 2015) and have demonstrated strong performance across images (Ho et al., 2020; Nichol & Dhariwal, 2021), audio (Kong et al., 2020), and video (Ho et al., 2022). The extension to discrete data was first explored by Sohl-Dickstein et al. (2015) for binary variables and generalized to categorical variables in Hoogeboom et al. (2021); Austin et al. (2021), where structured corruption processes, including absorbing-state and discretized Gaussian diffusion, were introduced. Subsequent work reformulated these processes as interpolations between clean data and a prior distribution: Sahoo et al. (2024) derived a simplified ELBO for masked diffusion as weighted cross-entropy losses, while Schiff et al. (2024) extended this framework to uniform noise with a continuous-time NELBO, thereby improving performance. Alternative perspectives leverage continuous-time Markov chains (Campbell et al., 2022; 2024) or concrete score matching (Lou et al., 2023).

Graph generation. Graph generation approaches can be categorized into two classes: autoregressive approaches, which generate nodes and edges sequentially, and one-shot approaches, which rely on diffusion-based methods. Diffusion models leverage the order-agnostic nature of denoising processes to maintain permutation equivariance, and many state-of-the-art approaches incorporate marginal distributions over node and edge attributes as priors (Vignac et al., 2022; Qin et al., 2024). Recent advances in graph generation have demonstrated that both discrete diffusion models, implemented in either discrete-time settings (Vignac et al., 2022) or continuous-time formulations (Xu et al., 2024), as well as discrete flow-matching approaches (Qin et al., 2024), achieve strong performance in different domains, including molecular and synthetic graph generation. Boget (2025) propose a graph diffusion model in which intermediate noisy states depend only on clean data, removing dependence on partially denoised states; this is closely related to the $\lambda = 1$ extreme of IDDM.

Text generation. Autoregressive transformers (Radford et al., 2019; Vaswani et al., 2017) achieved impressive results in language modeling but suffer from slow sequential sampling and limited controllability. To address this, various works explore discrete diffusion models. Absorbing-state variants (Shi et al., 2024; Sahoo et al., 2024) achieve strong empirical results; however, once a token is unmasked, it cannot be revised, which can lead to error accumulation and degrade sample quality. Several predictor-corrector samplers have been proposed (Campbell et al., 2022; Gat et al., 2024; Wang et al., 2025) to mitigate this issue, but at the cost of increased computation. In contrast, uniform diffusion models (Austin et al., 2021; Schiff et al., 2024) are naturally self-correcting, but remain less competitive empirically. We note that IDDM naturally enables resampling from arbitrary priors, thereby allowing refinement of tokens for both absorbing-state and uniform diffusion.

7 Conclusion

We introduce a simple and flexible discrete diffusion model whose reverse process is expressed by three actions: *(i)* staying in the current state, *(ii)* transitioning to the goal state, and *(iii)* resampling from the prior distribution. We derive a family of marginal-consistent posteriors that interpolates between a local and a state-independent update, enabling us to directly control the transition frequency while training only a single model. Further, we show that our reverse process cannot be recovered from a standard Markov noising process, as it corresponds to a non-Markovian forward process. Finally, we demonstrate that IDDM achieves competitive performance on graph and language generation tasks under different priors.

AI USE STATEMENT

In this work, we used generative AI tools for language editing, including grammar, spelling, and word choice, as well as for assisting in polishing mathematical proofs and formulating ideas. We have not used generative AI tools to generate synthetic datasets, develop theoretical models or conceptual frameworks, provide critical ingredients for proving mathematical claims, propose or refine hypotheses, design or provide feedback on research methodology or experiments, implement methods, assist with translation, clean or reformat datasets, support qualitative or thematic data analysis, or interpret results. We have reviewed all AI-assisted work. We take responsibility for the final content of this work, including text, claims, or artifacts produced with the aid of generative AI.

ETHICS STATEMENT

In this work, we propose a generative model for discrete data and apply it to molecule and text generation. As discrete generative models can be applied in various fields, we see potential positive and negative societal impacts. Generative modeling can accelerate scientific discovery and enhance natural language generation capabilities. However, there are potential societal consequences, specifically those related to the generation of harmful text or designing unsafe drugs.

REPRODUCIBILITY STATEMENT

The setup of all experimental evaluations is described in Section 4, Section F, and Section G.

REFERENCES

Jacob Austin, Daniel D Johnson, Jonathan Ho, Daniel Tarlow, and Rianne Van Den Berg. Structured denoising diffusion models in discrete state-spaces. *Advances in neural information processing systems*, 34:17981–17993, 2021.

Yoann Boget. Simple and critical iterative denoising: A recasting of discrete diffusion in graph generation. *arXiv preprint arXiv:2503.21592*, 2025.

Nathan Brown, Marco Fiscato, Marwin HS Segler, and Alain C Vaucher. Guacamol: benchmarking models for de novo molecular design. *Journal of chemical information and modeling*, 59(3): 1096–1108, 2019.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. *Advances in Neural Information Processing Systems*, 35:28266–28279, 2022.

Andrew Campbell, Jason Yim, Regina Barzilay, Tom Rainforth, and Tommi Jaakkola. Generative flows on discrete state-spaces: Enabling multimodal flows with applications to protein co-design. *arXiv preprint arXiv:2402.04997*, 2024.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky TQ Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. *Advances in Neural Information Processing Systems*, 37: 133345–133385, 2024.

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

Nate Gruver, Samuel Stanton, Nathan Frey, Tim GJ Rudner, Isidro Hotzel, Julien Lafrance-Vanasse, Arvind Rajpal, Kyunghyun Cho, and Andrew G Wilson. Protein design with guided discrete diffusion. *Advances in neural information processing systems*, 36:12489–12517, 2023.

Zhengfu He, Tianxiang Sun, Qiong Tang, Kuanning Wang, Xuan-Jing Huang, and Xipeng Qiu. Diffusionbert: Improving generative masked language models with diffusion models. In *Proceedings of the 61st annual meeting of the association for computational linguistics (volume 1: Long papers)*, pp. 4521–4534, 2023.

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. *Advances in neural information processing systems*, 35:8633–8646, 2022.

Emiel Hoogeboom, Didrik Nielsen, Priyank Jaini, Patrick Forré, and Max Welling. Argmax flows and multinomial diffusion: Learning categorical distributions. *Advances in neural information processing systems*, 34:12454–12465, 2021.

Tero Karras, Miika Aittala, Timo Aila, and Samuli Laine. Elucidating the design space of diffusion-based generative models. *Advances in neural information processing systems*, 35:26565–26577, 2022.

Seo Hyun Kim, Sunwoo Hong, Hojung Jung, Youngrok Park, and Se-Young Yun. Klass: Kl-guided fast inference in masked diffusion models. *Advances in Neural Information Processing Systems*, 38:92267–92301, 2026.

Zhifeng Kong, Wei Ping, Jiaji Huang, Kexin Zhao, and Bryan Catanzaro. Diffwave: A versatile diffusion model for audio synthesis. *arXiv preprint arXiv:2009.09761*, 2020.

Xiang Li, John Thickstun, Ishaan Gulrajani, Percy S Liang, and Tatsunori B Hashimoto. Diffusion-lm improves controllable text generation. *Advances in neural information processing systems*, 35: 4328–4343, 2022.

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

Liheng Ma, Chen Lin, Derek Lim, Adriana Romero-Soriano, Puneet K Dokania, Mark Coates, Philip Torr, and Ser-Nam Lim. Graph inductive biases in transformers without message passing. In *International Conference on Machine Learning*, pp. 23321–23337. PMLR, 2023.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In *International conference on machine learning*, pp. 8162–8171. PMLR, 2021.

Hunter Nisonoff, Junhao Xiong, Stephan Allenspach, and Jennifer Listgarten. Unlocking guidance for discrete state-space diffusion and flow models. *arXiv preprint arXiv:2406.01572*, 2024.

William Peebles and Saining Xie. Scalable diffusion models with transformers. In *Proceedings of the IEEE/CVF international conference on computer vision*, pp. 4195–4205, 2023.

Daniil Polykovskiy, Alexander Zhebrak, Benjamin Sanchez-Lengeling, Sergey Golovanov, Oktai Tatanov, Stanislav Belyaev, Rauf Kurbanov, Aleksey Artamonov, Vladimir Aladinskiy, Mark Veselov, et al. Molecular sets (moses): a benchmarking platform for molecular generation models. *Frontiers in pharmacology*, 11:565644, 2020.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Gunter Klambauer. Fréchet chemnet distance: a metric for generative models for molecules in drug discovery. *Journal of chemical information and modeling*, 58(9):1736–1741, 2018.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. Defog: Discrete flow matching for graph generation. *arXiv preprint arXiv:2410.04263*, 2024.

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

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, 2024.

Subham Sekhar Sahoo, Justin Deschenaux, Aaron Gokaslan, Guanghan Wang, Justin Chiu, and Volodymyr Kuleshov. The diffusion duality. *arXiv preprint arXiv:2506.10892*, 2025.

Anirban Sarkar, Yijie Kang, Nirali Somia, Pablo Mantilla, Jessica Lu Zhou, Masayuki Nagai, Ziqi Tang, Chris Zhao, and Peter Koo. Designing dna with tunable regulatory activity using score-entropy discrete diffusion. *bioRxiv*, pp. 2024–05, 2024.

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.

Jiaxin Shi, Kehang Han, Zhe Wang, Arnaud Doucet, and Michalis Titsias. Simplified and generalized masked diffusion for discrete data. *Advances in neural information processing systems*, 37: 103131–103167, 2024.

Antoine Siraudin, Fragkiskos D Malliaros, and Christopher Morris. Cometh: A continuous-time discrete-state graph diffusion model. *arXiv preprint arXiv:2406.06449*, 2024.

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

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. *arXiv preprint arXiv:2402.05841*, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Attention is all you need. *Advances in neural information processing systems*, 30, 2017.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. *arXiv preprint arXiv:2209.14734*, 2022.

Dimitri Von Rütte, Janis Fluri, Yuhui Ding, Antonio Orvieto, Bernhard Schölkopf, and Thomas Hofmann. Generalized interpolating discrete diffusion. *arXiv preprint arXiv:2503.04482*, 2025.

Alex Wang and Kyunghyun Cho. Bert has a mouth, and it must speak: Bert as a markov random field language model. *arXiv preprint arXiv:1902.04094*, 2019.

Guanghan Wang, Yair Schiff, Subham Sekhar Sahoo, and Volodymyr Kuleshov. Remasking discrete diffusion models with inference-time scaling. *arXiv preprint arXiv:2503.00307*, 2025.

Zhenqin Wu, Bharath Ramsundar, Evan N Feinberg, Joseph Gomes, Caleb Geniesse, Aneesh S Pappu, Karl Leswing, and Vijay Pande. Moleculenet: a benchmark for molecular machine learning. *Chemical science*, 9(2):513–530, 2018.

Zhe Xu, Ruizhong Qiu, Yuzhong Chen, Huiyuan Chen, Xiran Fan, Menghai Pan, Zhichen Zeng, Mahashweta Das, and Hanghang Tong. Discrete-state continuous-time diffusion for graph generation. *Advances in Neural Information Processing Systems*, 37:79704–79740, 2024.

A Proof of Theorem 3.1

Theorem A.1. *Let $\gamma : [0,1] \to [0,1]$ satisfy $\gamma_0 = 1$ and $\gamma_1 = 0$, and assume γ is monotone decreasing. For any $0 < t \leq 1$ and $0 \leq s < t$, define*

$$\gamma_{s|t} := \frac{1-\gamma_s}{1-\gamma_t} \in [0,1]. \tag{15}$$

Then for every $\lambda \in [0,1]$, the weights

$$w_{\text{stay}} := (1-\lambda)\,\gamma_{s|t}, \tag{16}$$
$$w_{\text{prior}} := \lambda\,(1-\gamma_s), \tag{17}$$
$$w_{\text{flip}} := (1-\lambda)\,(1-\gamma_{s|t}) + \lambda\,\gamma_s \tag{18}$$

satisfy the marginal constraints equation 7. Therefore, the reverse transition can be written equivalently as

$$\begin{aligned} q^{\lambda}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{x}_s; (1-\lambda)\gamma_{s|t}\,\mathbf{x}_t + \lambda(1-\gamma_s)\boldsymbol{\pi} \\ + \Big(1 - \big((1-\lambda)\gamma_{s|t} + \lambda(1-\gamma_s)\big)\Big)\mathbf{x}\Big) \end{aligned} \tag{19}$$

and its marginal satisfies

$$\sum_{\mathbf{x}_t} q^{\lambda}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x})\, q(\mathbf{x}_t \mid \mathbf{x}) = q(\mathbf{x}_s \mid \mathbf{x}), \qquad \textit{for all } s < t, \tag{20}$$

where $q(\mathbf{x}_t \mid \mathbf{x}) = \mathrm{Cat}(\mathbf{x}_t; (1-\gamma_t)\boldsymbol{\pi} + \gamma_t \mathbf{x})$.

Proof. We provide a proof in the following.

As a starting point, we have $q(\mathbf{x}_{t_1} \mid \mathbf{x}) = \mathrm{Cat}((1-\gamma_{t_1})\boldsymbol{\pi} + \gamma_{t_1}\mathbf{x})$ by definition. For the inductive hypothesis, we assume that for some $t \in [0,1]$, the marginal satisfies $q(\mathbf{x}_t \mid \mathbf{x}) = \mathrm{Cat}((1-\gamma_t)\boldsymbol{\pi} + \gamma_t\mathbf{x})$. The goal is to show that $q(\mathbf{x}_s \mid \mathbf{x}) = \mathrm{Cat}((1-\gamma_s)\boldsymbol{\pi} + \gamma_s\mathbf{x})$, for $s < t$.

Let $\mathbf{m}_t = (1-\gamma_t)\boldsymbol{\pi} + \gamma_t\mathbf{x}$. By marginalizing out $\mathbf{x}_t$ and using the linearity of the expectation, we have:

$$\begin{aligned} q(\mathbf{x}_s \mid \mathbf{x}) &= \sum_{\mathbf{x}_t} q^{\lambda}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x})\, q(\mathbf{x}_t \mid \mathbf{x}) \\ &= \sum_{\mathbf{x}_t} \mathrm{Cat}(\mathbf{x}_s; w_{\text{stay}}\mathbf{x}_t + w_{\text{prior}}\boldsymbol{\pi} + w_{\text{flip}}\mathbf{x})\, \mathrm{Cat}(\mathbf{x}_t; \mathbf{m}_t) \\ &= \mathrm{Cat}\Big(\mathbf{x}_s; \mathbb{E}_{\mathbf{x}_t \sim q(\cdot|\mathbf{x})}[w_{\text{stay}}\mathbf{x}_t + w_{\text{prior}}\boldsymbol{\pi} + w_{\text{flip}}\mathbf{x}]\Big) \\ &= \mathrm{Cat}\Big(\mathbf{x}_s; w_{\text{stay}}\mathbf{m}_t + w_{\text{prior}}\boldsymbol{\pi} + w_{\text{flip}}\mathbf{x}\Big) \end{aligned} \tag{21}$$

Now we simplify the expressions using the coefficients in equation 16 and equation 18:

$$\begin{aligned} w_{\text{stay}}\mathbf{m}_t &= (1-\lambda)\,\gamma_{s|t}\big((1-\gamma_t)\boldsymbol{\pi} + \gamma_t\mathbf{x}\big) \\ &= (1-\lambda)\frac{1-\gamma_s}{1-\gamma_t}(1-\gamma_t)\boldsymbol{\pi} + (1-\lambda)\gamma_{s|t}\gamma_t\mathbf{x} \\ &= (1-\lambda)(1-\gamma_s)\boldsymbol{\pi} + (1-\lambda)\gamma_{s|t}\gamma_t\mathbf{x} \end{aligned} \tag{22}$$

$$\begin{aligned} w_{\text{flip}}\mathbf{x} &= \big((1-\lambda)(1-\gamma_{s|t}) + \lambda\gamma_s\big)\mathbf{x} \\ &= (1-\lambda)(1-\gamma_{s|t})\mathbf{x} + \lambda\gamma_s\mathbf{x} \end{aligned} \tag{23}$$

Substituting equation 22 and equation 23 into equation 21 yields:

$$
\begin{aligned}
q(\mathbf{x}_s \mid \mathbf{x}) &= \mathrm{Cat}\Big(\mathbf{x}_s; w_{\text{stay}}\mathbf{m}_t + w_{\text{prior}}\boldsymbol{\pi} + w_{\text{flip}}\mathbf{x}\Big) \\
&= \mathrm{Cat}\Big(\mathbf{x}_s; (1-\lambda)(1-\gamma_s)\boldsymbol{\pi} + (1-\lambda)\gamma_{s|t}\gamma_t\mathbf{x} + \lambda(\mathbf{1}-\gamma_s)\boldsymbol{\pi} + (1-\lambda)(1-\gamma_{s|t})\mathbf{x} + \lambda\gamma_s\mathbf{x}\Big) \\
&= \mathrm{Cat}\Big(\mathbf{x}_s; (1-\gamma_s)\boldsymbol{\pi} + \big((1-\lambda)(1-\gamma_{s|t}(1-\gamma_t)) + \lambda\gamma_s\big)\mathbf{x}\Big) \\
&= \mathrm{Cat}\Big(\mathbf{x}_s; (1-\gamma_s)\boldsymbol{\pi} + \big((1-\lambda)(1-\frac{1-\gamma_s}{1-\gamma_t}(1-\gamma_t)) + \lambda\gamma_s\big)\mathbf{x}\Big) \\
&= \mathrm{Cat}\Big(\mathbf{x}_s; (1-\gamma_s)\boldsymbol{\pi} + \big((1-\lambda)\gamma_s + \lambda\gamma_s\big)\mathbf{x}\Big) \\
&= \mathrm{Cat}\Big(\mathbf{x}_s; (1-\gamma_s)\boldsymbol{\pi} + \gamma_s\mathbf{x}\Big)
\end{aligned}
$$

This completes the induction. □

B Derivation of the forward transition $q^\lambda(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x})$

In the following, we derive the forward transition step $q^\lambda(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x})$. Recall our reverse process defined for $\lambda \in [0, 1]$:

$$
\begin{aligned}
q^\lambda(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) &= \mathrm{Cat}\Big(\mathbf{x}_s; (1-\lambda)\gamma_{s|t}\,\mathbf{x}_t + \lambda(1-\gamma_s)\boldsymbol{\pi} && (24)\\
&\quad + \Big(1 - \big((1-\lambda)\gamma_{s|t} + \lambda(1-\gamma_s)\big)\Big)\mathbf{x}\Big) && (25)\\
&= \mathrm{Cat}\Big(\mathbf{x}_s; w_{\text{stay}}\mathbf{x}_t + w_{\text{prior}}\boldsymbol{\pi} + w_{\text{flip}}\mathbf{x}\Big) && (26)
\end{aligned}
$$

Let $\mathbf{x}_t$ and $\mathbf{x}$ be single tokens. We write the categorical parameters in scalar form. The reverse kernel implies that, for any $i, j \in \{1, \dots K\}$,

$$
q^\lambda(\mathbf{x}_s = i \mid \mathbf{x}_t = j, \mathbf{x}) = w_{\text{stay}}\mathbf{1}[i = j] + w_{\text{prior}}\boldsymbol{\pi}(i) + w_{\text{flip}}\mathbf{1}[i = \mathbf{x}].
$$

Let $\mathbf{m}_t(j) = q(\mathbf{x}_t = j \mid \mathbf{x})$, where $\mathbf{m}_t(\cdot) = (1-\gamma_t)\boldsymbol{\pi}(\cdot) + \gamma_t\mathbf{1}[. = \mathbf{x}]$.

We first derive the unnormalized forward probabilities: We fix $\mathbf{x}_s = i$. Then, for each j, we have:

$$
\begin{aligned}
q^\lambda(\mathbf{x}_t = j \mid \mathbf{x}_s = i, \mathbf{x}) &\propto q^\lambda(\mathbf{x}_s = i \mid \mathbf{x}_t = j, \mathbf{x})p(\mathbf{x}_t = j \mid \mathbf{x}) && (27)\\
&\propto \big(w_{\text{stay}}\mathbf{1}[i = j] + w_{\text{prior}}\boldsymbol{\pi}(i) + w_{\text{flip}}\mathbf{1}[i = \mathbf{x}]\big)\mathbf{m}_t(j) && (28)\\
&\propto \big(w_{\text{stay}}\mathbf{1}[i = j] + D_i\big)\mathbf{m}_t(j), && (29)
\end{aligned}
$$

where $D_i = w_{\text{prior}}\boldsymbol{\pi}(i) + w_{\text{flip}}\mathbf{1}[i = \mathbf{x}]$.

The normalizing constant is $q(\mathbf{x}_s = i \mid \mathbf{x}) = \mathbf{m}_s(i)$. We can also derive it as follows:

$$
\begin{aligned}
q(\mathbf{x}_s = i \mid \mathbf{x}) &= \sum_{j'} q^\lambda(\mathbf{x}_s = i \mid \mathbf{x}_t = j', \mathbf{x})q(\mathbf{x}_t = j' \mid \mathbf{x}) && (30)\\
&= \sum_{j'} (w_{\text{stay}}\mathbf{1}[i = j'] + D_i)\mathbf{m}_t(j') && (31)\\
&= w_{\text{stay}}\mathbf{m}_t(i) + D_i \sum_{j'} \mathbf{m}_t(j') && (32)\\
&= w_{\text{stay}}\mathbf{m}_t(i) + D_i\ , && (33)
\end{aligned}
$$

since $\sum_{j'} \mathbf{m}_t(j') = 1$. The normalized forward transition of equation 29 is:

$$
q^\lambda(\mathbf{x}_t = j \mid \mathbf{x}_s = i, \mathbf{x}) = \frac{\big(w_{\text{stay}}\mathbf{1}[i = j] + D_i\big)\mathbf{m}_t(j)}{\mathbf{m}_s(i)}. \tag{34}
$$

If $i = j$, then:

$$
q^\lambda(\mathbf{x}_t = i \mid \mathbf{x}_s = i, \mathbf{x}) = \frac{(w_{\text{stay}} + D_i)\mathbf{m}_t(i)}{\mathbf{m}_s(i)}.
$$

For $i \neq j$:

$$q^{\lambda}(\mathbf{x}_t = j \mid \mathbf{x}_s = i, \mathbf{x}) = \frac{D_i \mathbf{m}_t(j)}{\mathbf{m}_s(i)}.$$

To further simplify equation 34, we define the coefficient:

$$\alpha_i = \frac{D_i}{\mathbf{m}_s(i)} = \frac{D_i}{w_{\text{stay}} \mathbf{m}_t(i) + D_i} \in [0, 1] \,. \tag{35}$$

Then, we can rewrite the distribution over $\mathbf{x}_t$ as a mixture:

$$q^{\lambda}(\mathbf{x}_t = j \mid \mathbf{x}_s = i, \mathbf{x}) = \alpha_i \mathbf{m}_t(j) + (1 - \alpha_i)\mathbf{1}[j = i] \,. \tag{36}$$

The final forward transition can be simplified to:

$$q^{\lambda}(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x}) = \text{Cat}(\mathbf{x}_t; \alpha_{\mathbf{x}_s} \mathbf{m}_t + (1 - \alpha_{\mathbf{x}_s})\mathbf{e}_{\mathbf{x}_s}) \,, \tag{37}$$

where $\mathbf{e}_{\mathbf{x}_s}$ is the one-hot representation of $\mathbf{x}_s$. This means that in the forward transition, $\mathbf{x}_t = \mathbf{x}_s$ with probability $(1 - \alpha_{\mathbf{x}_s})$, otherwise we sample from the marginal $\mathbf{x}_t \sim \mathbf{m}_t = q(\mathbf{x}_t \mid \mathbf{x})$. The underlying forward process of IDDM is therefore non-Markovian as it requires access to the clean state $\mathbf{x}$.

Instantiating the forward transitions for the special cases.

For $\lambda = 0$: The coefficients in the reverse process are simplified to:

$$w_{\text{stay}} = \gamma_{s|t}, \quad w_{\text{prior}} = 0, \quad \text{and } w_{\text{flip}} = 1 - \gamma_{s|t}.$$

Therefore, $D_i = (1 - \gamma_{s|t})\mathbf{1}[i = x]$. The forward transition is:

- **Case 1:** if $\mathbf{x}_s \neq \mathbf{x}$ i.e., $(i \neq x)$, then $D_i = 0$ and the mixing coefficients $\alpha_i = 0$ according to equation 35. If we plug in these terms to equation 37, we get

$$q^{\lambda=0}(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x}) = \text{Cat}(\mathbf{x}_t; \mathbf{x}_s) \,.$$

- **Case 2:** if $\mathbf{x}_s = \mathbf{x}$, then $D_i = (1 - \gamma_{s|t})$, and the coefficient α in equation 37 simplifies to

$$\alpha = \frac{1 - \gamma_{s|t}}{(1 - \gamma_{s|t} + \gamma_{s|t} \mathbf{m}_t)} \,.$$

For $\lambda = 1$: The coefficients of the generative process are

$$w_{\text{stay}} = 0, \quad w_{\text{prior}} = (1 - \gamma_s), \quad \text{and } w_{\text{flip}} = \gamma_s$$

. Therefore, the Bayes expression in equation 34 collapses to:

$$q^{\lambda=1}(\mathbf{x}_t = j \mid \mathbf{x}_s = i, \mathbf{x}) = \frac{D_i \mathbf{m}_t(j)}{D_i} = \mathbf{m}_t(j) \,.$$

This implies that the forward transition is **conditionally independent of $\mathbf{x}_s$ given $\mathbf{x}$** and corresponds directly to the marginal

$$q^{\lambda=1}(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x}) = q(\mathbf{x}_t \mid \mathbf{x}) = \text{Cat}\Big((1 - \gamma_t)\boldsymbol{\pi} + \gamma_t \mathbf{x}\Big).$$

C CONTINUOUS-TIME ELBO FOR $\lambda = 0$.

We first derive the continuous-time diffusion loss, then identify the final continuous-time ELBO of IDDM. Setting $\lambda = 0$ gives the following simplified true reverse posterior and generative posterior:

$$q^{\lambda=0}(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \text{Cat}\Big(\mathbf{x}_s;\, \gamma_{s|t}\, \mathbf{x}_t + (1 - \gamma_{s|t})\, \mathbf{x}\Big), \tag{38}$$

$$p_\theta^{\lambda=0}(\mathbf{x}_s \mid \mathbf{x}_t) = \text{Cat}\Big(\mathbf{x}_s;\, \gamma_{s|t}\, \mathbf{x}_t + (1 - \gamma_{s|t})\, \mathbf{x}_\theta\Big), \tag{39}$$

where

$$\gamma_{s|t} = \frac{1-\gamma_s}{1-\gamma_t}, \qquad \mathbf{x}_\theta = f_\theta(\mathbf{x}_t, t). \tag{40}$$

For the discrete-time ELBO, we consider a uniform discretization $t^{(i)} = i/T$ with $\Delta_t = 1/T$, and write $s = t - \Delta_t$. The diffusion loss is

$$\mathcal{L}_{\text{diff}} := \sum_{i=2}^{T} \mathbb{E}_{q(\mathbf{x}_{t^{(i)}}|\mathbf{x})} \left[D_{\mathrm{KL}}\Big(q^{\lambda=0}(\mathbf{x}_{t^{(i-1)}} \mid \mathbf{x}_{t^{(i)}}, \mathbf{x}) \,\Big\|\, p_\theta^{\lambda=0}(\mathbf{x}_{t^{(i-1)}} \mid \mathbf{x}_{t^{(i)}}) \Big) \right]. \tag{41}$$

As $T \to \infty$, the sum converges to

$$\mathcal{L}_{\text{diff}} \to \int_0^1 \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[\lim_{\Delta_t \to 0} \frac{1}{\Delta_t} D_{\mathrm{KL}}\Big(q^{\lambda=0}(\mathbf{x}_{t-\Delta_t} \mid \mathbf{x}_t, \mathbf{x}) \,\Big\|\, p_\theta^{\lambda=0}(\mathbf{x}_{t-\Delta_t} \mid \mathbf{x}_t) \Big) \right] dt. \tag{42}$$

We therefore define the instantaneous loss

$$\ell(t, \mathbf{x}_t, \mathbf{x}) := \lim_{\Delta_t \to 0} \frac{1}{\Delta_t} D_{\mathrm{KL}}\Big(q^{\lambda=0}(\mathbf{x}_{t-\Delta_t} \mid \mathbf{x}_t, \mathbf{x}) \,\Big\|\, p_\theta^{\lambda=0}(\mathbf{x}_{t-\Delta_t} \mid \mathbf{x}_t) \Big). \tag{43}$$

We derive the first-order expansion of $\gamma_{s|t}$. Set $s = t - \Delta_t$. Since γ_t is continuously differentiable and monotone decreasing,

$$\gamma_s = \gamma_t - \dot{\gamma}_t \, \Delta_t + O(\Delta_t^2). \tag{44}$$

Therefore:

$$\gamma_{t-\Delta_t|t} = \frac{1-\gamma_{t-\Delta_t}}{1-\gamma_t} = 1 + \frac{\dot{\gamma}_t}{1-\gamma_t}\Delta_t + O(\Delta_t^2) = 1 - \beta_t \Delta_t + O(\Delta_t^2), \tag{45}$$

where we define the instantaneous rate:

$$\beta_t := -\frac{\dot{\gamma}_t}{1-\gamma_t} \geq 0. \tag{46}$$

Equivalently,

$$1 - \gamma_{t-\Delta_t|t} = \beta_t \Delta_t + O(\Delta_t^2). \tag{47}$$

Let $\mathbf{x}_t = \mathbf{e}_j$ and $\mathbf{x} = \mathbf{e}_c$. We re-write the infinitesimal form of the two categorical kernels:

$$q^{\lambda=0}(\mathbf{x}_{t-\Delta_t} \mid \mathbf{e}_j, \mathbf{e}_c) = (1 - \beta_t \Delta_t)\mathbf{e}_j + \beta_t \Delta_t \mathbf{e}_c + O(\Delta_t^2), \tag{48}$$

$$p_\theta^{\lambda=0}(\mathbf{x}_{t-\Delta_t} \mid \mathbf{e}_j) = (1 - \beta_t \Delta_t)\mathbf{e}_j + \beta_t \Delta_t \mathbf{x}_\theta + O(\Delta_t^2). \tag{49}$$

Case 1: $\mathbf{x}_t = \mathbf{x}$ (i.e. $j = c$)

If $j = c$, then the true posterior is a point mass at class c and

$$p_c^\theta = (1 - \beta_t \Delta_t) + \beta_t \Delta_t [\mathbf{x}_\theta]_c = 1 - \beta_t \Delta_t \big(1 - [\mathbf{x}_\theta]_c\big). \tag{50}$$

Hence

$$D_{\mathrm{KL}} = -\log\Big(1 - \beta_t \Delta_t (1 - [\mathbf{x}_\theta]_c)\Big) \tag{51}$$

$$= \beta_t \Delta_t \big(1 - [\mathbf{x}_\theta]_c\big) + O(\Delta_t^2) \tag{52}$$

$$= \beta_t \Delta_t \left(1 - [\mathbf{x}_\theta]_c\right) + O(\Delta_t^2). \tag{53}$$

Therefore,

$$\ell(t, \mathbf{e}_c, \mathbf{e}_c) = \beta_t \big(1 - [\mathbf{x}_\theta]_c\big) = \beta_t \big(1 - \mathbf{x}^\top \mathbf{x}_\theta\big). \tag{54}$$

Case 2: $\mathbf{x}_t \neq \mathbf{x}$ (i.e. $j \neq c$)

$$D_{\mathrm{KL}} = (1 - \beta_t \Delta_t) \log \frac{1 - \beta_t \Delta_t}{(1 - \beta_t \Delta_t) + \beta_t \Delta_t x_{\theta,j}} + \beta_t \Delta_t \log \frac{1}{x_{\theta,c}}$$

For small β_t:

$$\log \frac{1-\beta_t\Delta_t}{(1-\beta_t\Delta_t)+\beta_t\Delta_t x_{\theta,j}} = -\beta_t\Delta_t x_{\theta,j} + O(\Delta_t^2)$$

Hence,

$$D_{\mathrm{KL}} = \beta_t\Delta_t\Big(-\log x_{\theta,c} - x_{\theta,j}\Big)$$

Therefore,

$$\ell(t,\mathbf{x}_t,\mathbf{x}) = \beta_t(-\log \mathbf{x}_{\theta,c} - \mathbf{x}_{\theta,j})$$

Final continuous-time ELBO. We note that in the continuous-time, $\mathbf{x}_{\Delta_t} \sim q(\mathbf{x}_{\Delta_t} \mid \mathbf{x}) = \mathrm{Cat}(\mathbf{x}_{\Delta_t}; (1-\gamma_{\Delta_t})\boldsymbol{\pi} + \gamma_{\Delta_t}\mathbf{x}) = \mathrm{Cat}(\mathbf{x}_{\Delta_t}, \mathbf{x})$, as $\gamma_{\Delta_t} \to 1$. Consequently, the first latent vector is equal to the clean sample $\mathbf{x}$. We parametrize the model to copy the input at time $t(0)$ (Schiff et al., 2024), such that $f_\theta(\mathbf{x}_{t(0)}, t(0)) = \mathbf{x}_{t(0)}$. Therefore, in the limit, the reconstruction term $\mathcal{L}_{\text{rec}}$ is equal to 0. We further combine the two cases from above and take the expectation over $q(\mathbf{x}_t \mid \mathbf{x})$ in the following:

$$\lim_{\Delta_t\to 0}\frac{1}{\Delta_t}\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})}\left[D_{\mathrm{KL}}\right] = \beta_t\,\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})}\left[\mathbf{1}\{\mathbf{x}_t=\mathbf{x}\}\big(1-\mathbf{x}^\top\mathbf{x}_\theta\big) + \mathbf{1}\{\mathbf{x}_t\neq\mathbf{x}\}\big(-\mathbf{x}_t^\top\mathbf{x}_\theta - \log(\mathbf{x}^\top\mathbf{x}_\theta)\big)\right]. \tag{55}$$

Therefore, the continuous-time ELBO is

$$\mathcal{L}_{\mathrm{CT}} = \int_0^1 \beta_t\,\mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})}\left[\mathbf{1}\{\mathbf{x}_t=\mathbf{x}\}\big(1-\mathbf{x}^\top\mathbf{x}_\theta\big) + \mathbf{1}\{\mathbf{x}_t\neq\mathbf{x}\}\big(-\mathbf{x}_t^\top\mathbf{x}_\theta - \log(\mathbf{x}^\top\mathbf{x}_\theta)\big)\right] dt \tag{56}$$

with $\beta_t = -\frac{\dot\gamma_t}{1-\gamma_t}$.

D MDLM IS A SPECIAL CASE OF IDDM

MDLM matches IDDM's construction with:

$$\boldsymbol{\pi} := \mathbf{m}, \tag{57}$$
$$\gamma_t := \alpha_t, \tag{58}$$
$$\lambda := 0, \tag{59}$$

where α_t is the noise scheduler used in MDLM. In this case, the marginal corresponds exactly to MDLM's marginal:

$$q(\mathbf{x}_t \mid \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{x}_t;\,(1-\gamma_t)\mathbf{m} + \gamma_t\mathbf{x}\Big). \tag{60}$$

In addition, the coefficients of IDDM in the posterior in Eq. (3) simplify to:

$$w_{\text{stay}} := \frac{1-\gamma_s}{1-\gamma_t}, \tag{61}$$
$$w_{\text{prior}} := 0, \tag{62}$$
$$w_{\text{flip}} := \frac{\gamma_s-\gamma_t}{1-\gamma_t}. \tag{63}$$

The MDLM posterior is recovered as:

$$p(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \mathrm{Cat}\Big(\mathbf{x}_s; \frac{1-\gamma_s}{1-\gamma_t}\mathbf{x}_t + \frac{\gamma_s-\gamma_t}{1-\gamma_t}\mathbf{x}\Big), \tag{64}$$

where $\mathbf{x}_t \in \{\mathbf{x}, \mathbf{m}\}$. We can rewrite the posterior as:

$$q(\mathbf{x}_s \mid \mathbf{x}_t, \mathbf{x}) = \begin{cases} \mathrm{Cat}(\mathbf{x}_s; \mathbf{x}_t), & \mathbf{x}_t \neq \mathbf{m}, \\ \mathrm{Cat}\Big(\mathbf{x}_s; \frac{(1-\gamma_s)\mathbf{m} + (\gamma_s-\gamma_t)\mathbf{x}}{1-\gamma_t}\Big), & \mathbf{x}_t = \mathbf{m}, \end{cases} \tag{65}$$

which follows the reverse kernel in Sahoo et al. (2024).

To simplify the general ELBO in Eq. (56), we consider two cases:

Case 1: $\mathbf{x}_t = \mathbf{x}$ The first term in the ELBO $\mathbf{1}\{\mathbf{x}_t = \mathbf{x}\}\big(1 - \mathbf{x}^\top \mathbf{x}_\theta\big)$ can be written as $\mathbf{1}\{\mathbf{x}_t = \mathbf{x}\}\big(1 - [\mathbf{x}_\theta]_c\big)$, where $\mathbf{x} = \mathbf{e}_c$. If $\mathbf{x}_t$ is unmasked, then the unmasked latents are carried over, i.e., $\mathbf{x}_\theta(\mathbf{x}_t, t) = \mathbf{x}_t$, following (Sahoo et al., 2024). This is done by copying the network's output with the true token. In this case, $[\mathbf{x}_\theta]_c = 1$, and the first term in the ELBO vanishes:

$$\mathbf{1}\{\mathbf{x}_t = \mathbf{x}\}\big(1 - [\mathbf{x}_\theta]_c\big) = 0.$$

Case 2: $\mathbf{x}_t \neq \mathbf{x}$ This corresponds to the case where $\mathbf{x}_t = \mathbf{m}$. In this case, the term

$$\mathbf{1}\{\mathbf{x}_t \neq \mathbf{x}\}(-\mathbf{x}_t^\top \mathbf{x}_\theta) = \mathbf{1}\{\mathbf{x}_t = \mathbf{m}\}(-[\mathbf{x}_\theta]_\mathbf{m})$$

vanishes, since the model does not predict the mask token $\big([\mathbf{x}_\theta]_\mathbf{m} = 0\big)$.

Under these two cases, the final diffusion loss using the masking prior simplifies to:

$$\begin{aligned}
\mathcal{L}_{\text{diff}} &= \int_0^1 \beta_t \, \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[\mathbf{1}\{\mathbf{x}_t = \mathbf{x}\} \underbrace{\big(1 - \mathbf{x}^\top \mathbf{x}_\theta\big)}_{=0} + \mathbf{1}\{\mathbf{x}_t \neq \mathbf{x}\} \big(\underbrace{-\mathbf{x}_t^\top \mathbf{x}_\theta}_{=0} - \log(\mathbf{x}^\top \mathbf{x}_\theta) \big) \right] dt \\
&= \int_0^1 \beta_t \, \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[\mathbf{1}\{\mathbf{x}_t \neq \mathbf{x}\} \big(- \log(\mathbf{x}^\top \mathbf{x}_\theta) \big) \right] dt \\
&= - \int_0^1 \beta_t \, \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[\log\big(\mathbf{x}^\top \mathbf{x}_\theta(\mathbf{x}_t, t)\big) \right] dt \\
&= \int_0^1 \frac{\dot{\gamma}_t}{1 - \gamma_t} \, \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[\log\big(\mathbf{x}^\top \mathbf{x}_\theta(\mathbf{x}_t, t)\big) \right] dt \\
&= \mathbb{E}_{q(\mathbf{x}_t|\mathbf{x})} \left[\int_0^1 \frac{\dot{\gamma}_t}{1 - \gamma_t} \log\big(\mathbf{x}^\top \mathbf{x}_\theta(\mathbf{x}_t, t)\big) dt \right],
\end{aligned} \tag{66}$$

which corresponds to a weighted cross-entropy over the masked tokens, aligning exactly with MDLM's objective.

E Number of expected transitions

We can analytically derive the expected number of transitions, i.e., the number of transitions where $\mathbf{x}_s \neq \mathbf{x}_t$. First, we recall the update probability:

$$q^\lambda(\mathbf{x}_s = j \mid \mathbf{x}_t = i, \mathbf{x}) = (1 - \lambda)\gamma_{s|t} \mathbf{1}[i = j] + \lambda(1 - \gamma_s)\boldsymbol{\pi}(j) + w_{\text{flip}} \mathbf{1}[j = \mathbf{x}]. \tag{67}$$

A transition occurs when $j \neq i$. We define the indicator variable $C = \mathbf{1}[\mathbf{x}_t \neq \mathbf{x}_s]$. Now we want to derive the probability of a transition:

$$q^\lambda(C = 1 \mid \mathbf{x}_t = i, \mathbf{x}) = 1 - q^\lambda(\mathbf{x}_s = i \mid \mathbf{x}_t = i, \mathbf{x}). \tag{68}$$

Next, we condition the probability $q^\lambda(\mathbf{x}_s = j \mid \mathbf{x}_t = i, \mathbf{x})$ on two events:

Case 1: $\mathbf{x}_t \neq \mathbf{x}$ In this case, the last term drops out out and the transition simplifies further:

$$q^\lambda(\mathbf{x}_s = \mathbf{x}_t \mid \mathbf{x}_t \neq \mathbf{x}) = (1 - \lambda)\gamma_{s|t} + \lambda(1 - \gamma_s)\boldsymbol{\pi}(\mathbf{x}_t), \tag{69}$$

and therefore:

$$\begin{aligned}
q^\lambda(C = 1 \mid \mathbf{x}_t \neq \mathbf{x}) &= 1 - (1 - \lambda)\gamma_{s|t} - \lambda(1 - \gamma_s)\boldsymbol{\pi}(\mathbf{x}_t), && (70) \\
&= 1 - \gamma_{s|t} + \lambda\gamma_{s|t} - \lambda(1 - \gamma_s)\boldsymbol{\pi}(\mathbf{x}_t) && (71) \\
&= A + \lambda \left(\frac{1 - \gamma_s}{1 - \gamma_t} - (1 - \gamma_s)\boldsymbol{\pi}(\mathbf{x}_t) \right) && (72) \\
&= \underbrace{A}_{\geq 0} + \lambda \underbrace{\left((1 - \gamma_s) \left(\frac{1}{1 - \gamma_t} - \boldsymbol{\pi}(\mathbf{x}_t) \right) \right)}_{\geq 0} && (73)
\end{aligned}$$

Case 2: $\mathbf{x}_t = \mathbf{x}$ Here, we plug in the value of w_{flip}:

$$q^\lambda(\mathbf{x}_s = \mathbf{x}_t \mid \mathbf{x}_t = \mathbf{x}) = (1-\lambda)\gamma_{s|t} + \lambda(1-\gamma_s)\boldsymbol{\pi}(\mathbf{x}) + (1-\lambda)\,(1-\gamma_{s|t}) + \lambda\,\gamma_s \tag{74}$$

And therefore:

$$\begin{aligned} q^\lambda(C = 1 \mid \mathbf{x}_t = \mathbf{x}) &= 1 - (1-\lambda)\gamma_{s|t} - \lambda(1-\gamma_s)\boldsymbol{\pi}(\mathbf{x}) - (1-\lambda)\,(1-\gamma_{s|t}) - \lambda\,\gamma_s && (75)\\ &= 1 - (1-\lambda) - \lambda(1-\gamma_s)\boldsymbol{\pi}(\mathbf{x}) - \lambda\,\gamma_s && (76)\\ &= \lambda - \lambda(1-\gamma_s)\boldsymbol{\pi}(\mathbf{x}) - \lambda\,\gamma_s && (77)\\ &= \lambda\,(1 - (1-\gamma_s)\boldsymbol{\pi}(\mathbf{x}) - \gamma_s) && (78)\\ &= \lambda \underbrace{((1-\gamma_s)(1-\boldsymbol{\pi}(\mathbf{x}))}_{\geq 0} && (79)\\ & && (80) \end{aligned}$$

Finally, we write down the number of total expected transitions:

$$\begin{aligned} \mathbb{E}\left[\sum_{t=1}^{T} C_t\right] &= \sum_{t=1}^{T} \mathbb{E}\left[C_t\right] && (81)\\ &= \sum_{t=1}^{T} \mathbb{E}\left[C_t \mid \mathbf{x}_t = \mathbf{x}\right] q(\mathbf{x}_t = \mathbf{x}) + \mathbb{E}\left[C_t \mid \mathbf{x}_t \neq \mathbf{x}\right] q(\mathbf{x}_t \neq \mathbf{x}) && (82)\\ &= \sum_{t=1}^{T} q^\lambda(C_t = 1 \mid \mathbf{x}_t = \mathbf{x}) p(\mathbf{x}_t = \mathbf{x}) + q^\lambda(C_t = 1 \mid \mathbf{x}_t \neq \mathbf{x}) p(\mathbf{x}_t \neq \mathbf{x}), && (83) \end{aligned}$$

which is a mixture of two functions that are linear with respect to λ. As the mixture parameters are independent of λ, the total expected value is linear.

F Experimental Details

For all experiments we use a linear scheduler for γ_t, i.e., $\gamma_t = 1 - t$. In the following, we describe dataset specific details.

F.1 Datasets

Table 4: Additional details on the datasets.

Dataset	Tokenizer	Vocab. Size	Sequence Length
OWT	GPT2	50,257	1024
LM1B	bert-base-uncased	30,522	128

F.1.1 Text Datasets

We summarize the details of the text dataset in Table 4.

LM1B. We download the dataset from Hugging Face Datasets (LM1B), and use the bert-base-uncased tokenizer. We pad the sequences to a maximum length of 128. We use the official training, validation, and test splits provided in the dataset. The LM1B benchmark release is licensed under Apache 2.0.

OpenWebText. We tokenize using the GPT-2 tokenizer and wrap the sequences to a maximum length of 1024. We adopt the same train-validation splits from Sahoo et al. (2024), where the last 100,000 samples are used for validation.

F.1.2 Molecule datasets

QM9. The QM9 dataset (Wu et al., 2018) consists of molecules with up to 9 heavy atoms. We use the same split established by (Vignac et al., 2022), and 10000 graphs for evaluation. The dataset license is CC BY 4.0.

GuacaMol. Furthermore, we evaluate our model on the GuacaMol dataset (Brown et al., 2019), licensed under the MIT License. The molecules range from 2 to 88 heavy atoms.

MOSES. Finally, we benchmark our model on the MOSES dataset (Polykovskiy et al., 2020), licensed under the MIT License, spanning filtered molecules from 8 to 27 heavy atoms.

F.2 Model architectures

Table 5: Model architectures on various datasets.

Dataset	Architecture	Parameter Count
LM1B	Transformer	139 M
OWT	Transformer	169 M

Text datasets. For LM1B and OWT training, we use the modified Transformer (Peebles & Xie, 2023) architecture from Sahoo et al. (2024). The model consists of 12 layers, a hidden dimension of 768, and 12 attention heads.

Graph datasets. For the graph datasets, we use the graph transformer proposed by (Vignac et al., 2022) with the hyperparameters used by (Qin et al., 2024) and Relative Random Walk Probabilities (Ma et al., 2023) as node and edge features. We show our hyperparameters in Table 6. We used the GTX-1080, A100, H100, and H200 GPUs for the Molecule experiments in single-GPU training.

Table 6: Overview of the training hyperparameters for the different experiments.

	QM9	Guacamol	Moses
Epochs	1000	250	300
Learning rate	2×10^{-4}	2×10^{-4}	2×10^{-4}
Optimizer	ADAM $(0.9, 0.999)$	ADAM $(0.9, 0.999)$	ADAM $(0.9, 0.999)$
RRWP steps	12	12	20
Number of layers	9	10	12
Number of heads	8	8	8
ρ	4	4	4
λ	0.0	0.2	0.4

F.3 Training settings

Table 7: Overview of the training hyperparameters for the different experiments.

	LM1B	OWT
Train steps	1000K	300K
Context size	128	1024
Batch size	512	64
Learning rate	1×10^{-3}	3×10^{-4}
Optimizer	ADAM $(0.9, 0.999)$	ADAM $(0.9, 0.999)$
LR warmup steps	2.5K	2.5K
GPU count	1	1
GPU type	H200	H200

In Table 6 and Table 7, we report the training hyperparameters used in our experiments for the different datasets.

F.4 Evaluation metrics

F.4.1 Text Generation

Perplexity. Perplexity is computed as the exponential of the negative average token-level log-likelihood over the evaluation data. We report the perplexities on the test splits following Sahoo et al. (2024).

Generative perplexity. We report generative perplexity for unconditionally generated sequences, following (Sahoo et al., 2025). We measure it using GPT-2 large (Radford et al., 2019). We further report sample entropy to assess diversity.

F.4.2 Graph Generation

We evaluate the graph generation with standard molecule metrics. We report the validity, uniqueness, i.e., proportion of generated molecules with different SMILES string, and novelty, i.e., proportion of generated molecules not in the training set. Furthermore, we report the Frechet ChemNetDistance (FCD) (Preuer et al., 2018), which measures the embedding similarity of the generated molecules with respect to the training set. Finally, we report the Scaffold similarity and SNN.

G Additional Results

G.1 Text experiments

Likelihood evaluation. We compute the perplexity on LM1B and report results in Table 8. Our baselines include uniform discrete diffusion models as well as absorbing-state diffusion models (Wang & Cho, 2019; Austin et al., 2021; Sahoo et al., 2024). Compared to prior diffusion baselines, IDDM consistently achieves competitive performance across both families, remaining close to MDLM (Sahoo et al., 2024) in the absorbing-state setting.

Table 8: Test perplexities. Best results in **bold**, second best underlined.

Model	LM1B
Autoregressive	
Transformer (Sahoo et al., 2024)	22.3
Diffusion (Uniform)	
D3PM Uniform (Austin et al., 2021)	137.9
SEDD Uniform (Lou et al., 2023)	<u>40.3</u>
UDLM (Schiff et al., 2024)	**36.7**
IDDM Uniform (Ours)	40.7
Diffusion (Absorbing State / Gaussian)	
BERT-Mouth (Wang & Cho, 2019)	142.9
Diffusion-LM (Li et al., 2022)	118.6
D3PM Absorb (Austin et al., 2021)	76.9
DiffusionBert (He et al., 2023)	63.8
Duo (Sahoo et al., 2025)	33.7
SEDD Absorb (Lou et al., 2023)	32.7
MDLM (Sahoo et al., 2024)	**27.0**
IDDM Absorb (Ours)	<u>27.8</u>

G.2 Graph experiments

G.2.1 Training-sampling mismatch

As the marginal distribution of IDDM is shared across different λ, we can train a model and use different sampling strategies during inference. In the following, we investigate whether this induces a training-sampling mismatch. For this, we train and evaluate our QM9 setup across different λ combinations. We show the results in Table 9 As we observe, train-

Table 9: QM9 test performance under training–sampling mismatch. Bold denotes the best training λ for each evaluation λ.

FCD ↓

$\lambda_{\text{train}} \backslash \lambda_{\text{eval}}$	0.0	0.2	0.4	0.6	0.8	1.0
0.0	0.130	0.210	0.280	0.350	0.430	0.510
0.5	**0.083**	**0.174**	0.240	0.319	0.366	0.442
1.0	0.098	0.182	**0.203**	**0.241**	**0.262**	**0.283**

Validity (%) ↑

$\lambda_{\text{train}} \backslash \lambda_{\text{eval}}$	0.0	0.2	0.4	0.6	0.8	1.0
0.0	99.27	99.42	99.45	99.39	99.44	99.42
0.5	**99.49**	99.42	99.50	99.44	99.44	99.40
1.0	99.45	**99.51**	**99.63**	**99.60**	**99.62**	**99.59**

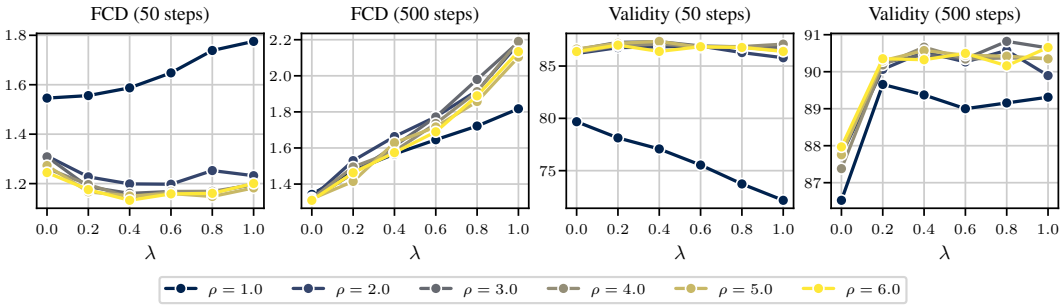


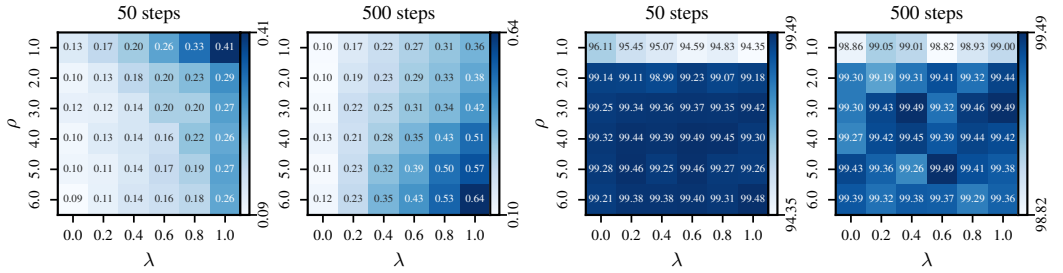
Figure 5: Different metrics on the MOSES dataset across different values of ρ and λ. For 50 steps, increasing ρ proves to be helpful but yields vanishing improvements after $\rho = 2$. Further, the FCD and validity achieve their optimum around 0.4 for 50 steps. Finally, for 500 steps, increasing λ worsens the FCD while increasing the validity.

ing with different λ can improve the performance. $\lambda = 0.5$ gives the best FCD at $\lambda_{\text{eval}} \in \{0, 0.2\}$, whereas training with $\lambda = 1$ performs best for $\lambda_{\text{eval}} \geq 0.4$. At $\lambda_{\text{eval}} = 1$, training with $\lambda_{\text{train}} = 1$ reduces FCD while maintaining comparable validity.

G.2.2 Time distortion

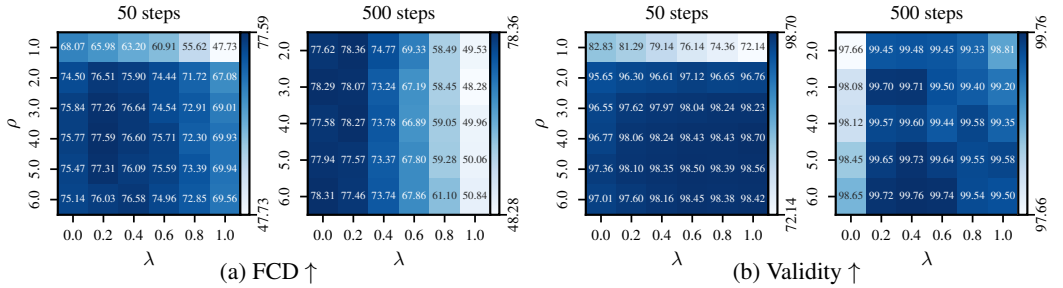
We distort time steps during sampling following Karras et al. (2022); Qin et al. (2024). Instead of uniformly spacing the time steps, we allocate more steps to later stages. We use a similar scheduler as Karras et al. (2022): $t(i)' = t(i)^{\rho}$, with $\rho > 1$. We analyze the effect of ρ in Fig. 5 and report the FCD and validity for MOSES across different settings. Increasing ρ above 1 yields a substantial improvement while stagnating from $\rho = 2$. We provide additional ablations in Section G.

In the following, we show hyperparameter ablations for ρ and λ across the datasets QM9, Moses, and GuacaMol in Figs. 6 to 8.



(a) FCD ↓ (b) Validity ↑

Figure 6: FCD and validity for QM9 across different parameters.



(a) FCD ↑ (b) Validity ↑

Figure 7: FCD and validity for Guacamol across different parameters.

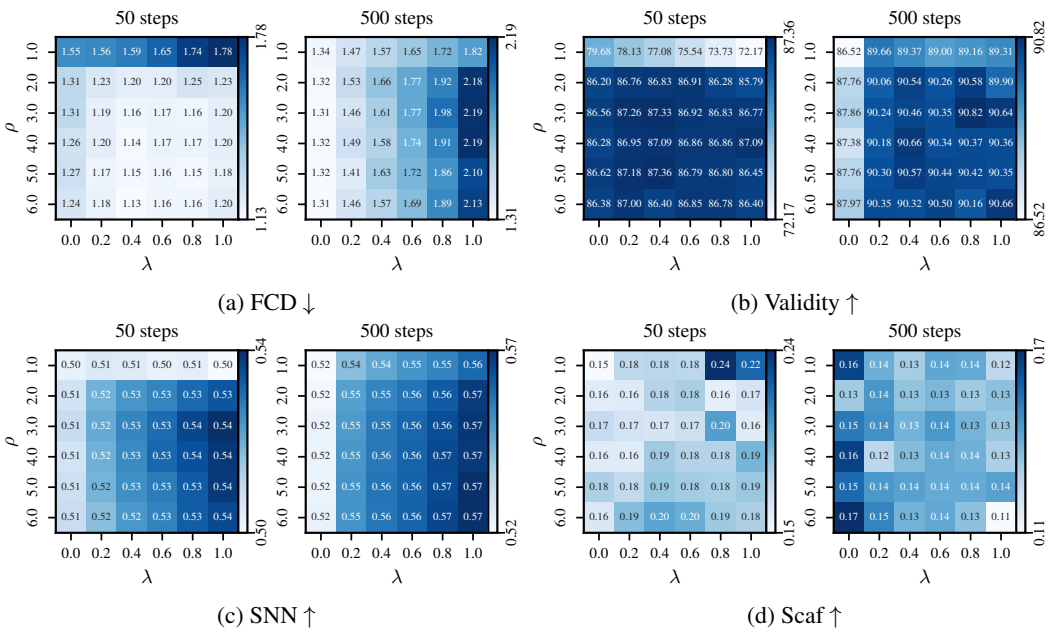


Figure 8: FCD, validity, SNN, and Scaffold similarity for MOSES across different parameters.