Unrestrained Simplex Denoising for Discrete Data A Generative Method Applied to Graphs

Yoann Boget
University of Applied Sciences
Western Switzerland (HES-SO)
Department of Computer Science
University of Geneva
yoann.boget@hes-so.ch

Alexandros Kalousis
University of Applied Sciences
Western Switzerland (HES-SO)

Abstract

Denoising models such as Diffusion or Flow Matching have recently advanced generative modeling for discrete structures, yet most approaches operate directly in the discrete state space, leading to abrupt state changes. We introduce simplex denoising, a simple yet effective generative framework that operates on the probability simplex. The key idea is a non-Markovian noising scheme in which, for a given clean data point, noisy representations at different times are conditionally independent. While preserving the theoretical guarantees of denoising-based generative models, our method removes unnecessary constraints, thereby improving performance and simplifying the formulation. Empirically, *unrestrained simplex denoising* surpasses strong discrete diffusion and flow-matching baselines across synthetic and real-world graph benchmarks. These results highlight the probability simplex as an effective framework for discrete generative modeling.

1 Introduction

Denoising models, such as discrete diffusion and discrete flow matching, have substantially advanced generative modeling for discrete structures [Austin et al., 2021, Campbell et al., 2022, 2024, Gat et al., 2024], including graphs [Haefeli et al., 2022, Vignac et al., 2023, Qin et al., 2025, Boget, 2025]. Yet, surprisingly few works study denoising on simplices for modeling discrete data [Richemond et al., 2022, Avdeyev et al., 2023, Stark et al., 2024, Liu et al., 2025]. This may result from the pathological behavior of interpolant approaches on the simplex [Stark et al., 2024], which creates abrupt discontinuities in the support of the noisy distributions (Section 3).

Moreover, standard approaches such as diffusion or flow matching strongly constrain the reverse dynamics by enforcing a continuous denoising trajectory in the continuous-time limit, i.e., $\lim_{h\to 0} p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) = \delta(\mathbf{x}_{t+h} - \mathbf{x}_t)$. In discrete diffusion, this constraint has been linked to compounding denoising errors [Lezama et al., 2023, Boget, 2025, Zhang et al., 2025]. We argue that a similar issue arises in the continuous-state setting.

To overcome these issues, we introduce *Unrestrained Simplex Denoising (UNSID)*, a simple yet effective generative model for discrete data that operates directly on the probability simplex. Our model leverages a non-Markovian noising process in which noisy states are conditionally independent given the clean input. By removing dependence on the previous noisy state, our formulation eliminates unnecessary constraints and simplifies the denoising process.

Our key contributions are as follows:

- We introduce a new non-Markovian noising approach for categorical data that operates on the probability simplex (Section 4) and apply it to graph generation. The formulation

40th Conference on Neural Information Processing Systems (NeurIPS 2026).

substantially simplifies standard denoising paradigms such as diffusion or flow matching and addresses the issue of compounding denoising errors.

- We provide theoretical guarantees, proving convergence of the proposed procedure assuming access to the true denoising distribution $p_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$ (Section 4).
- Leveraging explicit probability paths, we propose new theoretical tools for the analysis of the noising dynamics on the probability simplex. Specifically, we introduce the Voronoi probability and derive a closed-form expression for it under the Dirichlet probability path (Section 3).
- Empirically, our unrestrained simplex denoising method surpasses strong discrete diffusion and flow-matching baselines on synthetic and real-world benchmarks for graph generation, advancing the state of the art on three of the five datasets considered while remaining competitive with the strongest baselines on the other two (Section 5).

2 Background

In this paper, we focus on modeling multivariate categorical data, with applications to graphs that may have discrete node and edge attributes. In this section, we first review the related literature on denoising models for discrete data, focusing on graph generative models, and then present the non-Markovian noising processes relevant to our method.

2.1 Related Work

Recently, *denoising models* for discrete data have attracted considerable attention. Coupled with equivariant Graph Neural Networks (GNNs), they have become the dominant approach to graph generation, as they have proven highly effective (see Appendix C.1 for a comprehensive review of generative methods for graphs). We group existing denoising approaches for discrete data in general, and for graphs in particular, into three broad families: (1) continuous relaxation, (2) discrete diffusion, and (3) simplex-based denoising.

Approaches based on *continuous relaxation* embed discrete variables into the unconstrained Euclidean space $\mathbb{R}^K$ and train conventional continuous diffusion models with a standard Gaussian prior [Jo et al., 2022, 2024, Niu et al., 2020]. A related class of methods similarly injects continuous noise, but trains the denoiser to predict the categorical distribution directly [Eijkelboom et al., 2024, Song et al., 2025]. These approaches have two important shortcomings. First, points on the simplex have $K - 1$ degrees of freedom; embedding them in $\mathbb{R}^K$ adds a redundant dimension that carries no information and introduces unnecessary noise. Second, simplex-based noising naturally defines explicit and interpretable probability paths over categorical distributions (Section 3), whereas Euclidean noising requires nonlinear projection or renormalization steps whose behavior is harder to characterize.

Initiated by D3PM [Austin et al., 2021], *discrete diffusion* corrupts data by injecting discrete noise and models transitions directly in the discrete domain. Subsequent work extends this idea to continuous time [Campbell et al., 2022] and to discrete flow matching [Campbell et al., 2024, Gat et al., 2024]. In graph generation, this approach has been widely adopted [Haefeli et al., 2022, Xu et al., 2024, Boget, 2025], with DiGress [Vignac et al., 2023] and DeFoG [Qin et al., 2025] standing as two prominent representatives. However, transitioning abruptly between discrete states during denoising can drastically alter the structural properties of the instance; removing a single edge, for example, may disconnect the graph. Furthermore, GNNs employed as denoisers exhibit limited expressivity when operating on graphs with discrete node and/or edge attributes (see Appendix C.2). To address this limitation, discrete graph denoising methods typically require costly, handcrafted structural features.

In *simplex-based denoising*, categorical variables are represented as points on the probability simplex, with one-hot encodings corresponding to its vertices. One strategy diffuses in $\mathbb{R}^K$ and leverages distributional correspondences, such as gamma–Dirichlet or normal–logistic-normal, to map noisy samples back onto the simplex [Richemond et al., 2022, Floto et al., 2023]. A second strategy generates noise *directly* on the simplex [Avdeyev et al., 2023, Stark et al., 2024, Han et al., 2023]. In this work, we adopt the latter approach.

In graph generation, Beta diffusion [Liu et al., 2025] operates on binary data, and Eijkelboom et al. [2024] implement the Dirichlet flow-matching formulation as a baseline but report poor performance.

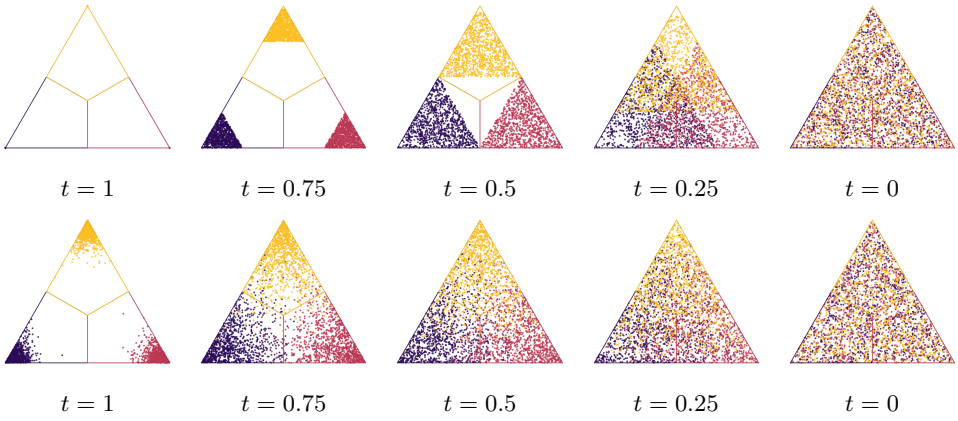


Figure 1: Upper row: Noising with a linear interpolant creates discontinuities and, for $t > 0.5$, all points remain in the Voronoi region of their original vertex. Lower row: Noising with an explicit parametric probability path avoids discontinuities.

No model has yet generalized successfully to the full probability simplex, suggesting that simplex-based graph generation is more challenging than it appears.

Yet, noise on the simplex presents notable advantages over discrete noise. (i) In general, continuous vectors contain more information than one-hot representations of the same dimension. (ii) Continuous noise on the simplex avoids discontinuities, most notably graph disconnection, and yields edge weights that naturally encode proximity, providing each node with information about its position relative to all others. (iii) Continuous vectors act as unique identifiers for nodes and edges, thereby alleviating the expressivity limitations of GNNs [Loukas, 2020] and limiting the need for additional handcrafted features.

2.2 Non-Markovian Noising Processes

Standard formulations of diffusion and flow-matching models are Markovian, such that $q_{t|t+h,1}(\mathbf{x}_t \mid \mathbf{x}_{t+h}, \mathbf{x}_1) = q_{t|t+h}(\mathbf{x}_t \mid \mathbf{x}_{t+h})$, where $h > 0$ denotes a small but not necessarily infinitesimal time interval such that $t+h \leq 1$. To our knowledge, DDIM [Song et al., 2021] was the first diffusion model to adopt a non-Markovian noising scheme, defining the forward kernel $q_{t|t+h,1}$ to depend on both the preceding noisy state and the clean sample. Recently, a family of non-Markovian formulations defines the discrete noising process solely as a function of the noise level α_t and the clean data $\mathbf{x}_1$, such that $q_{t|t+h,1}(\mathbf{x}_t \mid \mathbf{x}_{t+h}, \mathbf{x}_1) = q_{t|1}(\mathbf{x}_t \mid \mathbf{x}_1)$. During denoising, this independence assumption removes direct dependencies on previous noisy states, thereby mitigating error accumulation ("compounding denoising errors") [Lezama et al., 2023]. To the best of our knowledge, our work is the first to extend this class of non-Markovian methods to simplex-based denoising; it can be interpreted as the simplex counterpart of Simple Iterative Denoising (SID) [Boget, 2025].

3 Noising on the Simplex

Noising processes defined on the probability simplex pose specific challenges. In particular, as shown by Stark et al. [2024], specifying q_t as an *interpolant* between endpoint distributions q_1 and q_0, as is common in discrete diffusion and flow matching, restricts the support of q_t, leaving large regions of the simplex out of distribution and creating jump discontinuities in q_t. In this section, we formalize the setting, introduce an explicit, well-behaved parameterization of the noising path, and propose new theoretical tools for analyzing the noising dynamics.

3.1 Problem Setting

Let $\mathbb{S}_K$ denote the $(K-1)$-dimensional probability simplex, $\mathbb{S}_K = \left\{ \boldsymbol{x} \in \mathbb{R}^K \,\middle|\, \mathbf{1}_K^\top \boldsymbol{x} = 1,\ x_i > 0\ \forall i \right\}$. We represent categorical node/edge attributes as vertices of the simplex, i.e., $\mathbf{x}_1 \in \left\{ \boldsymbol{e}_i \right\}_{i \in [K]}$, where $\boldsymbol{e}_i$ denotes the ith standard basis vector.

A noising process on the simplex is a family of conditional distributions $(q_t)_{t \in [0,1)}$ with $q_t(\cdot \mid \mathbf{x}_1)$ supported on $\mathbb{S}_K$ for all $t \in [0,1)$, and boundary conditions $\lim_{t \to 1} q_t(\mathbf{x}_t \mid \mathbf{x}_1) = \delta(\mathbf{x}_t - \mathbf{x}_1), \quad q_0(\mathbf{x}_0 \mid \mathbf{x}_1) = q_0(\mathbf{x}_0)$, i.e., the terminal distribution collapses to the clean vertex and the initial distribution is independent of $\mathbf{x}_1$. Such processes can be specified either implicitly via a stochastic interpolant or explicitly by choosing a parametric family on the simplex.

The interpolant construction is standard in discrete diffusion and flow matching, and also underlies SID [Boget, 2025], which can be viewed as the discrete analogue of our approach. This construction sets $\mathbf{x}_t = \alpha_t \mathbf{x}_1 + (1 - \alpha_t)\mathbf{x}_0$, with $\mathbf{x}_1 \sim p_{\text{data}}$, $\mathbf{x}_0 \sim q_0$, and $\alpha_t \in [0,1]$. However, this formulation exhibits what Stark et al. [2024] call pathological properties. In particular, within the interpolant framework, the posterior $P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) \propto (\mathbf{x}_1) q_t(\mathbf{x}_t \mid \mathbf{x}_1) p_{\text{data}}$ has support on at most $K-1$ vertices for all noise levels $\alpha_t > 1/K$, and becomes trivial, i.e., concentrated on a single vertex, for $\alpha_t > 0.5$. For all t, $q_t(\mathbf{x}_t \mid \mathbf{x}_1)$ is a uniform distribution over a subset of $\mathbb{S}_K$, and zero elsewhere, producing support discontinuities. The upper row of Figure 1 illustrates such a process on the 2-simplex. Such piecewise-constant behavior is ill-suited for approximating posteriors with differentiable neural networks. We avoid these pathologies by specifying an explicit, parametric probability path on the simplex.

Dirichlet distributions (and their mixtures) offer a natural, tractable choice on $\mathbb{S}_K$. Recall the Dirichlet density function: $\mathrm{Dir}(\boldsymbol{x}; \boldsymbol{\alpha}) = \mathrm{B}(\boldsymbol{\alpha})^{-1} \prod_{i=1}^K x_i^{\alpha_i - 1}$, where B is the multivariate beta function (see Equation 12). For instance, a simple yet useful noising path is defined as $q_t(\mathbf{x}_t \mid \mathbf{x}_1) = \mathrm{Dir}\big(\mathbf{x}_t; \mathbf{1} + \alpha_t \mathbf{x}_1\big)$, with a non-decreasing noise schedule $\alpha_t \in \mathbb{R}_{\geq 0}$ such that $\alpha_0 = 0$ and $\lim_{t \to 1} \alpha_t = \infty$, e.g., $\alpha_t = -a \log(1-t)$, where $a > 0$ is a hyperparameter. This way, q_t has full support on $\mathbb{S}_K$ for any $t \in [0,1)$, thereby avoiding the support restrictions of interpolant-based methods. The lower row of Figure 1 illustrates noising based on a parametric probability path.

3.2 Voronoi Probabilities

The Voronoi regions of the simplex vertices provide an intuitive lens on how a noising process behaves. Let $\mathbb{V}_k \subseteq \mathbb{S}_K$ denote the Voronoi region of vertex k, defined as $\mathbb{V}_k = \{\boldsymbol{x} \in \mathbb{S}_K \mid d(\boldsymbol{x}, \boldsymbol{e}_k) \leq d(\boldsymbol{x}, \boldsymbol{e}_j)\ \forall j \neq k\}$, with $d(\cdot,\cdot)$ the Euclidean distance. Geometrically, $\mathbb{V}_k$ is the polytope of points whose nearest simplex vertex is $\boldsymbol{e}_k$. Equivalently, we have $\boldsymbol{x} \in \mathbb{V}_k \iff x_k = \max_j x_j$.

Given the conditional noising distribution $q_t(\mathbf{x}_t \mid \mathbf{x}_1)$, we define the *Voronoi probability* for vertex k as the conditional probability $P_{v_k}(\mathbf{x}_t \in \mathbb{V}_k \mid \mathbf{x}_1 = \boldsymbol{e}_k)$, i.e., the probability that the closest vertex to $\mathbf{x}_t$ is still its origin $\mathbf{x}_1$. Intuitively, this quantity summarizes how quickly the unconditional distribution $q_t(\mathbf{x}_t)$ "mixes". For the stochastic-interpolant construction, for instance, one has $P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) = 1$ for all t such that $\alpha_t > 0.5$; consequently, the posterior over vertices collapses to the Voronoi indicator $P_{1|t}(\mathbf{x}_1 = \boldsymbol{e}_k \mid \mathbf{x}_t) = [\mathbf{x}_t \in \mathbb{V}_k]$.

For the probability path $\mathrm{Dir}(\mathbf{x}_t; \mathbf{1} + \alpha_t \mathbf{x}_1)$ introduced above, the Voronoi probability admits a closed-form expression:

Proposition 3.1 *Assume* $\mathbf{x}_t \mid \mathbf{x}_1 = \boldsymbol{e}_i \sim \mathrm{Dir}(\mathbf{1} + \alpha_t \boldsymbol{e}_i)$. *Then,*

$$P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) = \sum_{k=0}^{K-1} \frac{(-1)^k \binom{K-1}{k}}{(k+1)^{\alpha_t + 1}}. \tag{1}$$

We present all proofs in Appendix A.

We leverage Voronoi probabilities to design and calibrate explicit probability paths. In Appendix F.3.2, we evaluate our model across multiple values of the schedule hyperparameter a, demonstrating the usefulness of Voronoi probabilities for calibrating the noise schedule.

4 Method

In this section, we present our UNSID method. As with any other denoising model, it includes a noising process that progressively corrupts the data and a learned denoising process that operates backward.

We denote by $p := p_{\text{data}}$ an *arbitrary* data distribution on $\mathcal{S}^L$, where $\mathcal{S}$ is a finite alphabet of size K and L is a finite number of dimensions, and by $\mathcal{X} := \{\mathbf{x}_1 \in \mathcal{S}^L : p(\mathbf{x}_1) > 0\}$ its support. Following Section 3, we identify each symbol of $\mathcal{S}$ with its one-hot encoding, i.e., $\mathcal{S} = \{\boldsymbol{e}_1, \dots, \boldsymbol{e}_K\}$, so that clean instances $\mathbf{x}_1 \in \mathcal{S}^L$ are vertices of the multi-simplex $\mathbb{S}_K^L$. Throughout, $\mathbf{x} = \mathbf{x}^{(1:L)}$ denotes a full instance and $\mathbf{x}^{(i)}$ its ith dimension. The multivariate forward (noising) process acts independently on each dimension $\mathbf{x}^{(i)}$, $i \in [L]$, noising instances on $\mathbb{S}_K^L$. In contrast, during the reverse process used at inference, dependencies across dimensions are captured by a learned denoiser that outputs element-wise logits conditioned on all dimensions $\mathbf{x}_t^{(1:L)}$. This setup follows prior work on discrete diffusion and flow matching [Austin et al., 2021, Vignac et al., 2023, Campbell et al., 2024, Stark et al., 2024, Boget, 2025].

Our UNSID framework can be viewed as a simplex analogue of SID [Boget, 2025]. Extending SID to the continuous simplex setting is non-trivial and constitutes a significant methodological contribution of this work. We refer the reader to Appendix C.3 for a detailed comparison and a precise account of the contributions over SID.

4.1 Noising

Our noising mechanism relies on two key components: (i) a probability path that governs corruption at any continuous time $t \in [0, 1)$, and (ii) a conditional-independence assumption across different times.

Probability path Since the forward process operates independently across dimensions, the distribution at time t factorizes as

$$q_t(\mathbf{x} \mid \mathbf{x}_1) = \prod_{i=1}^{L} q_t\big(\mathbf{x}^{(i)} \mid \mathbf{x}_1^{(i)}\big). \tag{2}$$

In principle, our method accommodates any path $(q_t)_{t\in[0,1)}$. In practice, however, we require (a) the ability to easily sample from q_t for all t, and (b) well-behaved noise distributions as described in Section 3, in particular satisfying the overlap condition of Proposition 4.3. For instance, Dirichlet distributions (or mixtures thereof) represent a suitable choice, e.g., $q_t(\mathbf{x}^{(i)} \mid \mathbf{x}_1^{(i)}) = \mathrm{Dir}(\mathbf{x}^{(i)}; \mathbf{1} + \alpha_t \mathbf{x}_1^{(i)})$, where $\alpha_t \in \mathbb{R}_{\geq 0}$ is a non-decreasing noise schedule with $\alpha_0 = 0$ and $\lim_{t\to 1} \alpha_t = \infty$. We further discuss the choice of the probability path in the context of graph generation in Appendix B and leave the exploration of alternative probability paths for future work.

Independence across time Unlike standard diffusion models such as DDPM [Ho et al., 2020], DDIM [Song et al., 2021], or D3PM [Austin et al., 2021], our model does not define direct dependencies between $\mathbf{x}_t$ and $\mathbf{x}_{t+h}$. Instead, we assume conditional independence of intermediate noisy states given the clean input, as formalized in the following assumption:

Assumption 4.1 *For any $s \neq t$, the noising process satisfies the following conditional-independence property:*

$$q_t(\mathbf{x}_t \mid \mathbf{x}_s, \mathbf{x}_1) = q_t(\mathbf{x}_t \mid \mathbf{x}_1). \tag{3}$$

Because training typically draws a single noisy sample at a randomly chosen time t, this assumption has no practical effect on the training objective. It does, however, have major implications for the denoising process.

4.2 Denoising

Let $P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$ denote the posterior distribution of the clean instance $\mathbf{x}_1 \in \mathcal{X}$ given a noisy instance $\mathbf{x}_t$ at time t. We denote by $p_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t)$ a single denoising transition and obtain the following result.

Proposition 4.2 *Given $P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$ and Assumption 4.1, the one-step denoising kernel satisfies*

$$\begin{aligned} p_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) &= \sum_{\mathbf{x}_1 \in \mathcal{X}} q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1)\, P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) \\ &= \mathbb{E}_{\mathbf{x}_1 \sim P_{1|t}(\cdot|\mathbf{x}_t)} \left[q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1) \right]. \end{aligned} \tag{4}$$

Thus, each denoising step can be interpreted as (i) sampling a clean instance $\mathbf{x}_1$ from the posterior $P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$, followed by (ii) re-noising this sample via $q_{t+h}(\cdot \mid \mathbf{x}_1)$. Equivalently, we sample from the mixture of noising distributions weighted by the posterior. In practice, the posterior is approximated by a product of per-dimension distributions (Section 4.5), so both steps are performed independently for each dimension. We emphasize that the simplicity of these formulations relies on the fact that $q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1)$ remains on the simplex. At inference time, we fix the number of function evaluations (NFE) to T and use a uniform step size $h = 1/T$. The full reverse procedure consists of iterating the transition in Equation 4 T times.

4.3 Convergence and Corrector Sampling

Fixing the time index in Equation 4 (i.e., setting $h = 0$) yields a Markov kernel at time t, whose convergence is characterized by the following proposition.

Proposition 4.3 *Let $t \in [0, 1)$ and let $P_{1|t}$ be the true posterior. Assume the* overlap condition

$$\varepsilon_t \;:=\; \int_{\mathbb{S}_K^L} \min_{\mathbf{x}_1 \in \mathcal{X}} q_t(\mathbf{x} \mid \mathbf{x}_1)\, \mathrm{d}\mathbf{x} \;>\; 0. \tag{5}$$

Then, for any initial distribution, the Markov chain with kernel $p_{t'}(\mathbf{x}'_t \mid \mathbf{x}_t) = \sum_{\mathbf{x}_1 \in \mathcal{X}} P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, q_t(\mathbf{x}'_t \mid \mathbf{x}_1)$ converges in total variation to its unique stationary distribution

$$\pi_t(\mathbf{x}_t) = \sum_{\mathbf{x}_1 \in \mathcal{X}} p(\mathbf{x}_1)\, q_t(\mathbf{x}_t \mid \mathbf{x}_1). \tag{6}$$

The overlap condition requires the noisy distributions $q_t(\cdot \mid \mathbf{x}_1)$ to share a common component of total mass ε_t. Since, for every k, the density of $\mathrm{Dir}(\mathbf{1} + \alpha_t \boldsymbol{e}_k)$ vanishes only on the boundary of the simplex, and the path factorizes across dimensions, we have $\varepsilon_t > 0$ for all $t < 1$.

Applying this kernel at a fixed time during sampling therefore yields a simple *corrector* sampler, analogous in spirit to the corrector step of Gat et al. [2024] for discrete flow matching. Thus, iterating denoising and corrector steps to stationarity recovers samples from $p(\mathbf{x}_1)$. In practice, the true posterior $P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$ is not available and must be approximated by a parameterized model $P^{\theta}_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$. The resulting approximation error induces an error in the denoising transition, which is bounded as follows:

Proposition 4.4 *The KL divergence between the true and approximate one-step denoising distributions is upper-bounded by the KL divergence between the true and approximate posterior distributions:*

$$D_{\mathrm{KL}}\left(p_{t+h}(\cdot \mid \mathbf{x}_t) \,\|\, p^{\theta}_{t+h}(\cdot \mid \mathbf{x}_t)\right) \le D_{\mathrm{KL}}\left(P_{1|t}(\cdot \mid \mathbf{x}_t) \,\|\, P^{\theta}_{1|t}(\cdot \mid \mathbf{x}_t)\right). \tag{7}$$

This proposition guarantees that the posterior approximation error is not amplified in the induced one-step denoising transition.

4.4 Differences from and Advantages over Diffusion and Flow Matching

While our procedure resembles standard diffusion and flow matching, there is a key distinction. In classical reverse processes, one typically enforces $\lim_{h \to 0} p_{t+h}(\mathbf{x} \mid \mathbf{x}_t) = \delta(\mathbf{x} - \mathbf{x}_t)$. By Assumption 4.1, our model does not impose this constraint. Intuitively, diffusion and flow matching update the sample by interpolating between the current noisy state and the denoiser's prediction, whereas our method directly resamples a less noisy state conditioned on the denoiser's prediction.

Unlike flow matching, our method contracts the error accumulated during sampling, as formalized below. We denote by Q the Markov operator associated with the denoising kernel of Equation 4 under the true posterior, acting on a distribution μ as $\mu Q := \int p_{t+h}(\cdot \mid \mathbf{x}_t)\, \mu(\mathbf{x}_t)\, \mathrm{d}\mathbf{x}_t$.

Proposition 4.5 *Let $t + h < 1$ with $\varepsilon_{t+h} > 0$ (Equation 5), and let $\tilde{\pi}_t$ be any distribution on $\mathbb{S}_K^L$. Let Φ be the flow map from t to $t + h$ of an exact flow-matching ODE with marginals $(\pi_s)_s$, assumed to be a diffeomorphism. Then*

$$\|\tilde{\pi}_t Q - \pi_{t+h}\|_{\mathrm{TV}} \leq (1 - \varepsilon_{t+h}) \, \|\tilde{\pi}_t - \pi_t\|_{\mathrm{TV}}, \tag{8}$$

$$\|\Phi_{\#}\tilde{\pi}_t - \pi_{t+h}\|_{\mathrm{TV}} = \|\tilde{\pi}_t - \pi_t\|_{\mathrm{TV}}. \tag{9}$$

Here, $\tilde{\pi}_t$ represents the law of the sampler at time t, including the error accumulated so far: a flow-matching step transports this error unchanged, whereas a UNSID step shrinks it by a factor of at least $1 - \varepsilon_{t+h}$.

Moreover, in diffusion and flow matching, if $\mathbf{x}_t$ lies in a region where the learned denoiser $P^{\theta}_{1|t}$ poorly approximates the true posterior $P_{1|t}$, or in a region highly sensitive to small perturbations (e.g., near equilibrium points), then the next iterate $\mathbf{x}_{t+h}$ remains close to $\mathbf{x}_t$, causing approximation errors to compound over time. In contrast, our method resamples from $P^{\theta}_{1|t}$ at each step, so $\mathbf{x}_{t+h}$ need not stay in the same local region. This reduces error compounding by allowing the trajectory to escape regions that are poorly approximated and/or sensitive to small perturbations. Our experiments demonstrate that UNSID consistently outperforms diffusion-based models and Dirichlet Flow Matching by a large margin.

In addition, our formulation is a substantial simplification compared with standard denoising methods such as diffusion and flow matching. Compared to diffusion, Assumption 4.1 yields the simplification $q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1, \mathbf{x}_t) = q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1)$. Compared to flow matching, it removes the need to construct a vector field that realizes the chosen probability path and to evaluate its derivatives at every denoising step, operations that are computationally expensive (see the sampling-time comparison in Appendix F.1). In contrast, our method is both simple and computationally efficient. We therefore regard this simplification as an important contribution.

4.5 Parameterization and Learning

As in discrete diffusion, the posterior $P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$ is intractable. We approximate it by a product of per-dimension categorical distributions, $P^{\theta}_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) = \prod_{i=1}^{L} \mathrm{Cat}(\mathbf{x}_1^{(i)}; \boldsymbol{\omega}^{(i)})$, with $\boldsymbol{\omega}^{(1:L)} = \mathrm{softmax}\big(f_\theta(\mathbf{x}_t, t)\big)$, where the neural network f_θ, referred to as the *denoiser*, outputs element-wise logits. Our training objective matches the x-prediction of discrete diffusion and discrete flow matching methods, so any of their criteria can be adopted. To respect the symmetries of the graph, we instantiate f_θ as a GNN that is equivariant with respect to node permutations.

On graphs, in place of generic vectors $\mathbf{x}^{(i)} \in \{0, 1\}^K$, we write $\mathbf{v}^{(i)} \in \{0, 1\}^{d_v}$ for vectors encoding the categorical attributes of node i (d_v categories), and $\mathbf{a}^{(i,j)} \in \{0, 1\}^{d_a+1}$ for those encoding the categorical attributes of the pair (i, j) (d_a categories), with an additional category accounting for the absence of an edge.

Following Vignac et al. [2023], we train our model by minimizing the following weighted negative log-likelihood:

$$\begin{aligned} \mathcal{L}_\theta = \mathbb{E}\Bigg[& \frac{\gamma}{n} \sum_i \Big[-\log(P^{\theta}_{1|t}(\mathbf{v}_1^{(i)} \mid \mathcal{G}_t)) \Big] \\ & + \frac{1-\gamma}{n(n-1)} \sum_{i \neq j} \Big[-\log(P^{\theta}_{1|t}(\mathbf{a}_1^{(i,j)} \mid \mathcal{G}_t)) \Big] \Bigg], \end{aligned} \tag{10}$$

where the expectation is taken over $t \sim \mathcal{U}(0, 1)$, $\mathcal{G}_1 \sim p_{\text{data}}$, and $\mathcal{G}_t \sim q_t(\mathcal{G}_t \mid \mathcal{G}_1)$, n is the number of nodes, and γ is a weighting factor between nodes and edges.

4.6 Guidance

While high-quality unconditional generation is a prerequisite, conditioning on graph-level properties is necessary for many downstream tasks. In drug discovery, for example, one often seeks molecules that are both synthetically accessible and highly active against specified targets. Denoising-based models typically steer samples via either *classifier guidance* [Dhariwal and Nichol, 2021] or *classifier-free guidance* [Ho and Salimans, 2022]. Our framework supports both through direct adaptations of

established techniques. Due to space limitations, we provide implementation details in Appendix D. We demonstrate the effectiveness of our approach for property-conditioned molecular generation in Section 5.

5 Evaluation and Results

We evaluate our model on both molecular and unattributed graph datasets. For molecular data, we adopt two standard benchmark datasets: `QM9` and `ZINC250K`. For `QM9`, we use the more challenging version *with explicit hydrogens* (`QM9H`), since performance on the hydrogen-free version is close to saturation. The `QM9H` dataset contains 133,885 molecules with up to 29 atoms of 5 types. The `ZINC250K` dataset contains 249,455 molecular graphs with up to 38 heavy atoms of 9 types. For unattributed graphs, we evaluate on the synthetic `Planar` and Stochastic Block Model (`SBM`) datasets, comprising graphs of exactly 64 nodes and of up to 187 nodes, respectively, as well as on the real-world `Enzymes` dataset, with graphs of up to 125 nodes. Visualizations of generated molecules and graphs are available in Appendix G. Our (anonymous) code is available at `https://github.com/yoboget/UNSID`.

5.1 Baselines

We compare against two categories of baselines: (i) published models representing different denoising approaches, and (ii) our own implementations for controlled comparisons. Due to space limitations, we describe the baselines in Appendix E.7. In general, we report results from each baseline's original paper, except on `Planar` and `SBM`, where we reran the experiments for baselines that reported results from a *single* sampling run (see Appendix E.7 for details and further justification).

To isolate the effect of our sampling formulation, we train three models with identical architectures and training hyperparameters for each experiment: a discrete denoiser, a simplex-based denoiser, and a continuous denoiser. The only notable difference is that the discrete denoiser includes the extra input features used by DIGRESS. For the discrete and the simplex-based denoisers, we evaluate two sampling procedures each. For the discrete denoiser, we use standard discrete diffusion (DISCDIFF, as in DIGRESS) and its non-Markovian variant (NM-DD, as in SID). For the simplex-based denoiser, we implement the Dirichlet flow-matching sampler of Stark et al. [2024] (DIRIFM), as well as our unrestrained simplex denoising (UNSID) sampler. The continuous denoiser is identical to the simplex-based denoiser, but its inputs lie in $\mathbb{R}^{K\times L}$ instead of $\mathbb{S}^L_K$. We refer to it as non-Markovian continuous diffusion (NM-CD (see Appendix E.6 for details). Experimental and implementation details are provided in Appendix E.5.

5.2 Results

We report means over five independent sampling runs. Appendices E.5 and F provide the evaluation protocol, additional metrics, and standard deviations. We highlight in bold all results that fall within the training-set error margin. If none do, we bold the best result.

Molecule Generation We evaluate generative performance on `QM9H` (Table 6 in Appendix E.5) and `ZINC250K` (Table 1) by measuring distributional similarity in chemical space via the Fréchet ChemNet Distance (FCD) [Preuer et al., 2018], in subgraph structures via the neighborhood subgraph pairwise distance kernel MMD (NSPDK) [Costa and Grave, 2010], reported $\times 10^3$), and chemical validity without correction. Our UNSID model surpasses all baselines, often by a large margin, on all metrics. In particular, it generates very few invalid molecules.

Table 1: Results on ZINC250K.

	VALID (%)↑	FCD↓	NSPDK↓
TRAIN.	100.00	1.13	0.10
DIGRESS	94.99	3.48	2.1
GRUM	98.65	2.26	1.5
GRABFN	99.22	2.12	1.3
SID	99.50	2.06	2.01
DISCDIFF	74.17	4.78	4.08
NM-DD	99.92	2.65	3.48
NM-CD	57.96	9.51	13.65
DIRIFM	97.32	2.79	1.92
UNSID	**99.98**	**1.79**	**1.00**

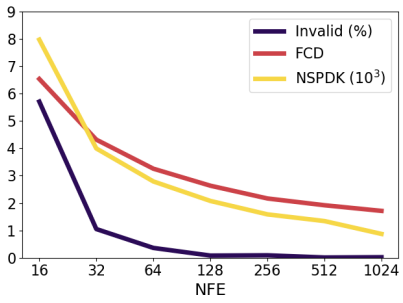


Figure 2: ZINC250K results for various NFE.

Unattributed Graph Generation For unattributed graphs, we measure distributional similarity via Maximum Mean Discrepancy (MMD) over degree distributions, clustering coefficients, orbit counts, and spectral densities. For Planar and SBM, we follow the procedure of Martinkus et al. [2022]: we report MMDs $\times 10^3$, as well as validity, i.e., planarity for Planar and statistical consistency with the true stochastic block model for SBM. On SBM, we also report novelty, i.e., the fraction of unique generated graphs that are not isomorphic to those in the training set. On Enzymes, we follow Jo et al. [2022]. In particular, we compute MMDs using the Earth Mover's Distance rather than the total variation distance.

On Planar (Table 2), our model achieves a perfect validity rate and results comparable to the best baselines on the other metrics. On SBM, we reach the highest validity after DEFOG, whose results, however, strongly suggest overfitting and dataset memorization and should be taken with caution (see Appendix C.4). On the remaining metrics, our results are comparable to those of the strongest baselines. On Enzymes (Table 3), our model achieves the best orbit MMD and the best the mean ratio between the model's MMDs to the training-set MMDs, while remaining competitive on the other metrics.

Table 2: Results on Planar and SBM.

	PLANAR					SBM					
	VALID↑	DEG.↓	CLUST.↓	ORBIT↓	SPECT.↓	VALID↑	DEG.↓	CLUST.↓	ORBIT↓	SPECT.↓	NOVEL ↑
TRAIN.	100.0	0.25	36.6	0.8	6.6	93.5	1.57	50.1	37.0	4.6	100
DIGRESS	45.5	0.71	45.4	1.31	8.4	51.0	**1.28**	51.5	**39.6**	**5.0**	100
GRUM	91.0	**0.38**	40.7	6.42	**7.9**	67.5	2.20	**49.9**	**40.4**	**5.1**	100
SID	91.3	5.9	163.4	19.1	**7.6**	63.5	11.5	**51.4**	123.1	5.9	100
DEFOG	99.5	0.5	50.1	0.6	**7.2**	**90.0**	0.6*	51.7	55.6	**5.4**	90
DISCDIFF	0.0	56.1	294.0	1410.0	85.1	0.0	**1.82**	86.4	125.7	11.3	100
NM-DD	98.0	14.1	363.0	27.2	**6.9**	60.5	4.38	**50.9**	52.9	5.7	100
NM-CD	0.0	1.2	226.3	45.8	22.0	57.5	**1.89**	**50.3**	**41.1**	**5.4**	100
DIRIFM	0.0	6.4	196.6	45.0	21.8	46.0	4.26	53.0	53.0	**5.0**	100
UNSID	**100.0**	**0.36**	**39.9**	**0.78**	**7.1**	78.5	**1.74**	**49.9**	52.1	5.9	100

* Significantly below the reference value.

5.3 Ablations and Efficiency

We study the impact of key design choices, namely the unrestrained denoising formulation, the NFE, the noise schedule, and the initialization of the sampling noise, and we assess sampling efficiency. First, DIRIFM and UNSID use the same denoiser. The only difference lies in the sampling strategy. On almost all experiments and metrics, our unrestrained sampler outperforms Dirichlet flow matching, often by a large margin, demonstrating the empirical superiority of our sampling method. Second, we ablate NFE on ZINC250K (Figure 2 and Table 7 in Appendix E.5). Our model attains state-of-the-art validity in as few as 64 steps, with competitive FCD and NSPDK. As expected, all metrics improve steadily with larger NFE. Third, Appendix F.1 shows that our model is highly efficient at sampling: it is the fastest among all compared methods and achieves these results with fewer parameters than the strongest baselines. Fourth, Appendix F.3.2 demonstrates that Voronoi probabilities provide an

effective tool for calibrating our noise schedule, and that our model remains robust under substantial variations of the schedule hyperparameter a. Finally, we compare two noise initializations: the uniform prior $q_0 = \mathrm{Dir}(\mathbf{1})$ and the marginal mixture q_0^{marg} with $\kappa = 2$ (see Appendix B for definitions). Marginal initialization yields a small but consistent improvement on all metrics (see Figure 5 in Appendix E.5).

5.4 Guidance

We evaluate conditional generation toward a target molecular property using the Quantitative Estimate of Drug-likeness (QED) score, as implemented in `RDKit`. On `QM9H`, we apply classifier guidance to steer samples toward prescribed standardized QED values. Concretely, we draw 1,000 target values from the test set, generate 1,000 molecules conditioned on these targets, and compute the QED of each generated molecule with `RDKit`. We quantify guidance effectiveness by the mean absolute error (MAE) between the target and generated standardized QED values. Results are reported in Table 9 in Appendix E.5. Classifier guidance effectively steers generation toward the prescribed values, reducing the MAE from 1.06 (unconditional) to 0.42, while preserving the model's ability to generate valid molecules.

Table 3: Results on `Enzymes`.

	DEG.↓	CLUST.↓	ORBIT↓	SPECT.↓	RATIO↓
TRAIN.	0.003	0.019	0.001	0.0189	1
GRAPHVAE	1.369	0.629	0.191	-	-
GRAPHRNN	0.017	0.062	0.046	-	-
GDSS	0.026	0.061	0.009	-	-
DISCDIFF	0.212	0.631	0.515	0.0196	155.0
NM-CD	0.084	0.308	0.125	0.0202	42.6
NM-DD	0.014	**0.040**	0.010	0.0174	4.3
DIRIFM	**0.007**	0.101	0.006	0.0198	3.7
UNSID	0.010	0.052	**0.003**	**0.0195**	**2.3**

6 Conclusion

We introduced *Unrestrained Simplex Denoising* (UNSID), a new denoising paradigm for categorical data that operates directly on the probability simplex and relaxes the Markovian constraints commonly imposed by diffusion and flow-matching methods. Theoretically, we established convergence and error-contraction guarantees and introduced Voronoi probabilities as a tool for analyzing and calibrating noising paths on the simplex. On molecular and unattributed graph benchmarks, UNSID matches or surpasses strong discrete diffusion and flow-matching baselines, reaches state-of-the-art validity with few denoising steps, and enables effective property-conditioned generation. Ablations demonstrate the benefits of the non-Markovian formulation, a favorable NFE–quality trade-off, and the utility of marginal-based initialization on the simplex.

Limitations Although UNSID could in principle extend to other forms of discrete data, we validate it only on graph generation in the present work. Its applicability to other discrete modalities, in particular sequences such as text, proteins, or DNA, remains to be established empirically and is left for future research.

Future work This limitation also points to opportunities. As a new class of generative models, UNSID opens promising directions, such as exploring alternative probability paths and integrating hybrid discrete–continuous variables.

Acknowledgments and Disclosure of Funding

We acknowledge the financial support of the Swiss National Science Foundation within the LegoMol project (grant no. 207428). The computations were performed at the University of Geneva on "Baobab" and "Bamboo" HPC clusters.

References

Jacob Austin, Daniel D. Johnson, Jonathan Ho, Daniel Tarlow, and Rianne van den Berg. Structured denoising diffusion models in discrete state-spaces. In M. Ranzato, A. Beygelzimer, Y. Dauphin, P.S. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, volume 34, pages 17981–17993. Curran Associates, Inc., 2021. URL https://proceedings.neurips.cc/paper_files/paper/2021/file/958c530554f78bcd8e97125b70e6973d-Paper.pdf.

Pavel Avdeyev, Chenlai Shi, Yuhao Tan, Kseniia Dudnyk, and Jian Zhou. Dirichlet diffusion score model for biological sequence generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 1276–1301. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/avdeyev23a.html.

Andreas Bergmeister, Karolis Martinkus, Nathanaël Perraudin, and Roger Wattenhofer. Efficient and scalable graph generation through iterative local expansion. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=2XkTz7gdpc.

Yoann Boget. Simple and critical iterative denoising: A recasting of discrete diffusion in graph generation. In *Proceedings of the 42th International Conference on Machine Learning*, Proceedings of Machine Learning Research. PMLR, July 2025.

Yoann Boget, Magda Gregorova, and Alexandros Kalousis. Discrete graph auto-encoder. *Transactions on Machine Learning Research*, 2024. ISSN 2835-8856. URL https://openreview.net/forum?id=bZ80b0wb9d.

Yoann Boget, Pablo Strasser, and Alexandros Kalousis. Hierarchical equivariant graph generation, 2025. URL https://openreview.net/forum?id=uEqOYXtn7f.

Andrew Campbell, Joe Benton, Valentin De Bortoli, Thomas Rainforth, George Deligiannidis, and Arnaud Doucet. A continuous time framework for discrete denoising models. In S. Koyejo, S. Mohamed, A. Agarwal, D. Belgrave, K. Cho, and A. Oh, editors, *Advances in Neural Information Processing Systems*, volume 35, pages 28266–28279. Curran Associates, Inc., 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. In *Proceedings of the 41st International Conference on Machine Learning*, pages 5453–5512, 2024.

Xiaohui Chen, Jiaxing He, Xu Han, and Liping Liu. Efficient and degree-guided graph generation via discrete diffusion modeling. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 4585–4610. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/chen23k.html.

Fabrizio Costa and Kurt De Grave. Fast neighborhood subgraph pairwise distance kernel. In Johannes Fürnkranz and Thorsten Joachims, editors, *Proceedings of the 27th International Conference on Machine Learning (ICML-10), June 21-24, 2010, Haifa, Israel*, pages 255–262. Omnipress, 2010. URL https://icml.cc/Conferences/2010/papers/347.pdf.

Nicola De Cao and Thomas Kipf. MolGAN: An implicit generative model for small molecular graphs. *arXiv:1805.11973 [cs, stat]*, May 2018. URL http://arxiv.org/abs/1805.11973. arXiv: 1805.11973.

Prafulla Dhariwal and Alexander Quinn Nichol. Diffusion models beat GANs on image synthesis. In A. Beygelzimer, Y. Dauphin, P. Liang, and J. Wortman Vaughan, editors, *Advances in Neural Information Processing Systems*, 2021. URL https://openreview.net/forum?id=AAWuCvzaVt.

Floor Eijkelboom, Grigory Bartosh, Christian A. Naesseth, Max Welling, and Jan-Willem van de Meent. Variational flow matching for graph generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=UahrHR5HQh.

Griffin Floto, Thorsteinn Jonsson, Mihai Nica, Scott Sanner, and Eric Zhengyu Zhu. Diffusion on the probability simplex, 2023. URL https://arxiv.org/abs/2309.02530.

Itai Gat, Tal Remez, Neta Shaul, Felix Kreuk, Ricky T. Q. Chen, Gabriel Synnaeve, Yossi Adi, and Yaron Lipman. Discrete flow matching. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=GTDKo3Sv9p.

Nikhil Goyal, Harsh Vardhan Jain, and Sayan Ranu. Graphgen: A scalable approach to domain-agnostic labeled graph generation. In Yennun Huang, Irwin King, Tie-Yan Liu, and Maarten van Steen, editors, *WWW '20: The Web Conference 2020, Taipei, Taiwan, April 20-24, 2020*, pages 1253–1263. ACM / IW3C2, 2020. doi: 10.1145/3366423.3380201. URL https://doi.org/10.1145/3366423.3380201.

Kilian Konstantin Haefeli, Karolis Martinkus, Nathanaël Perraudin, and Roger Wattenhofer. Diffusion models for graphs benefit from discrete state spaces. In *The First Learning on Graphs Conference*, 2022. URL https://openreview.net/forum?id=CtsKBwhTMKg.

Xiaochuang Han, Sachin Kumar, and Yulia Tsvetkov. SSD-LM: Semi-autoregressive simplex-based diffusion language model for text generation and modular control. In Anna Rogers, Jordan Boyd-Graber, and Naoaki Okazaki, editors, *Proceedings of the 61st Annual Meeting of the Association for Computational Linguistics (Volume 1: Long Papers)*, pages 11575–11596, Toronto, Canada, July 2023. Association for Computational Linguistics. doi: 10.18653/v1/2023.acl-long.647. URL https://aclanthology.org/2023.acl-long.647/.

Jonathan Ho and Tim Salimans. Classifier-free diffusion guidance, 2022. URL https://arxiv.org/abs/2207.12598.

Jonathan Ho, Ajay Jain, and Pieter Abbeel. Denoising diffusion probabilistic models. In H. Larochelle, M. Ranzato, R. Hadsell, M.F. Balcan, and H. Lin, editors, *Advances in Neural Information Processing Systems*, volume 33, pages 6840–6851. Curran Associates, Inc., 2020. URL https://proceedings.neurips.cc/paper_files/paper/2020/file/4c5bcfec8584af0d967f1ab10179ca4b-Paper.pdf.

Jaehyeong Jo, Seul Lee, and Sung Ju Hwang. Score-based Generative Modeling of Graphs via the System of Stochastic Differential Equations. *Proceedings of the 39th International Conference on Machine Learning*, 162:10362–10383, 2022. URL https://github.com/harryjo97/GDSS. http://arxiv.org/abs/2202.02514.

Jaehyeong Jo, Dongki Kim, and Sung Ju Hwang. Graph generation with diffusion mixture. In Ruslan Salakhutdinov, Zico Kolter, Katherine Heller, Adrian Weller, Nuria Oliver, Jonathan Scarlett, and Felix Berkenkamp, editors, *Proceedings of the 41st International Conference on Machine Learning*, volume 235 of *Proceedings of Machine Learning Research*, pages 22371–22405. PMLR, 21–27 Jul 2024. URL https://proceedings.mlr.press/v235/jo24b.html.

Mahdi Karami. Higen: Hierarchical graph generative networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=KNvubydSB5.

Lingkai Kong, Jiaming Cui, Haotian Sun, Yuchen Zhuang, B. Aditya Prakash, and Chao Zhang. Autoregressive diffusion model for graph generation. In Andreas Krause, Emma Brunskill, Kyunghyun Cho, Barbara Engelhardt, Sivan Sabato, and Jonathan Scarlett, editors, *Proceedings of the 40th International Conference on Machine Learning*, volume 202 of *Proceedings of Machine Learning Research*, pages 17391–17408. PMLR, 23–29 Jul 2023. URL https://proceedings.mlr.press/v202/kong23b.html.

Igor Krawczuk, Pedro Abranches, Andreas Loukas, and Volkan Cevher. {GG}-{gan}: A geometric graph generative adversarial network, 2021. URL https://openreview.net/forum?id=qiAxL3Xqx1o.

Jose Lezama, Tim Salimans, Lu Jiang, Huiwen Chang, Jonathan Ho, and Irfan Essa. Discrete predictor-corrector diffusion models for image synthesis. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=VM8batVBWvg.

Renjie Liao, Yujia Li, Yang Song, Shenlong Wang, Will Hamilton, David K Duvenaud, Raquel Urtasun, and Richard Zemel. Efficient Graph Generation with Graph Recurrent Attention Networks. In *Advances in Neural Information Processing Systems*, volume 32. Curran Associates, Inc., 2019.

Jenny Liu, Aviral Kumar, Jimmy Ba, Jamie Kiros, and Kevin Swersky. Graph normalizing flows. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Xinyang Liu, Yilin He, Bo Chen, and Mingyuan Zhou. Advancing graph generation through beta diffusion. In *13th International Conference on Learning Representations (ICLR 2025)*, 2025.

Andreas Loukas. What graph neural networks cannot learn: depth vs width. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=B1l2bp4YwS.

Youzhi Luo, Keqiang Yan, and Shuiwang Ji. GraphDF: A Discrete Flow Model for Molecular Graph Generation. *Proceedings of the 38th International Conference on Machine Learning*, 139: 7192–7203, 2021. URL http://arxiv.org/abs/2102.01189.

Kaushalya Madhawa, Katushiko Ishiguro, Kosuke Nakago, and Motoki Abe. Graphnvp: An invertible flow model for generating molecular graphs, 2019.

Karolis Martinkus, Andreas Loukas, Nathanaël Perraudin, and Roger Wattenhofer. SPECTRE: Spectral conditioning helps to overcome the expressivity limits of one-shot graph generators. In Kamalika Chaudhuri, Stefanie Jegelka, Le Song, Csaba Szepesvari, Gang Niu, and Sivan Sabato, editors, *Proceedings of the 39th International Conference on Machine Learning*, volume 162 of *Proceedings of Machine Learning Research*, pages 15159–15179. PMLR, 17–23 Jul 2022. URL https://proceedings.mlr.press/v162/martinkus22a.html.

Christopher Morris, Martin Ritzert, Matthias Fey, William L. Hamilton, Jan Eric Lenssen, Gaurav Rattan, and Martin Grohe. Weisfeiler and leman go neural: Higher-order graph neural networks. In *33rd AAAI Conference on Artificial Intelligence, AAAI 2019, 31st Innovative Applications of Artificial Intelligence Conference, IAAI 2019 and the 9th AAAI Symposium on Educational Advances in Artificial Intelligence, EAAI 2019*, volume 33, pages 4602–4609. AAAI Press, jul 2019. ISBN 9781577358091. doi: 10.1609/aaai.v33i01.33014602. URL www.aaai.org.

Van Khoa Nguyen, Yoann Boget, Frantzeska Lavda, and Alexandros Kalousis. GLAD: Improving latent graph generative modeling with simple quantization. In *ICML 2024 Workshop on Structured Probabilistic Inference & Generative Modeling*, 2024. URL https://openreview.net/forum?id=aY1gdSolIv.

Alexander Quinn Nichol and Prafulla Dhariwal. Improved denoising diffusion probabilistic models. In Marina Meila and Tong Zhang, editors, *Proceedings of the 38th International Conference on Machine Learning*, volume 139 of *Proceedings of Machine Learning Research*, pages 8162–8171. PMLR, 18–24 Jul 2021. URL https://proceedings.mlr.press/v139/nichol21a.html.

Chenhao Niu, Yang Song, Jiaming Song, Shengjia Zhao, Aditya Grover, and Stefano Ermon. Permutation Invariant Graph Generation via Score-Based Generative Modeling. In *Proceedings of Machine Learning Research*, volume 108, pages 4474–4484. PMLR, jun 2020. URL http://proceedings.mlr.press/v108/niu20a.html.

Kristina Preuer, Philipp Renz, Thomas Unterthiner, Sepp Hochreiter, and Günter 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. doi: 10.1021/acs.jcim.8b00234. PMID: 30118593.

Yiming Qin, Clement Vignac, and Pascal Frossard. Sparse training of discrete diffusion models for graph generation, 2024. URL https://openreview.net/forum?id=oTRekADULK.

Yiming Qin, Manuel Madeira, Dorina Thanou, and Pascal Frossard. Defog: Discrete flow matching for graph generation. In *Forty-second International Conference on Machine Learning*, 2025. URL https://openreview.net/forum?id=KPRIwWhqAZ.

Pierre H. Richemond, Sander Dieleman, and Arnaud Doucet. Categorical sdes with simplex diffusion, 2022. URL https://arxiv.org/abs/2210.14784.

Chence Shi, Minkai Xu, Zhaocheng Zhu, Weinan Zhang, Ming Zhang, and Jian Tang. Graphaf: a flow-based autoregressive model for molecular graph generation. In *International Conference on Learning Representations*, 2020. URL https://openreview.net/forum?id=S1esMkHYPr.

Martin Simonovsky and Nikos Komodakis. GraphVAE: Towards Generation of Small Graphs Using Variational Autoencoders. *arXiv:1802.03480 [cs]*, February 2018. URL http://arxiv.org/abs/1802.03480. arXiv: 1802.03480.

Jiaming Song, Chenlin Meng, and Stefano Ermon. Denoising diffusion implicit models. In *International Conference on Learning Representations*, 2021. URL https://openreview.net/forum?id=St1giarCHLP.

Yuxuan Song, Juntong Shi, Jingjing Gong, Minkai Xu, Stefano Ermon, Hao Zhou, and Wei-Ying Ma. Smooth interpolation for improved discrete graph generative models. In Aarti Singh, Maryam Fazel, Daniel Hsu, Simon Lacoste-Julien, Felix Berkenkamp, Tegan Maharaj, Kiri Wagstaff, and Jerry Zhu, editors, *Proceedings of the 42nd International Conference on Machine Learning*, volume 267 of *Proceedings of Machine Learning Research*, pages 56363–56388. PMLR, 13–19 Jul 2025. URL https://proceedings.mlr.press/v267/song25f.html.

Hannes Stark, Bowen Jing, Chenyu Wang, Gabriele Corso, Bonnie Berger, Regina Barzilay, and Tommi Jaakkola. Dirichlet flow matching with applications to dna sequence design. In *Proceedings of the 41st International Conference on Machine Learning*, ICML'24. JMLR.org, 2024.

Ashish Vaswani, Noam Shazeer, Niki Parmar, Jakob Uszkoreit, Llion Jones, Aidan N Gomez, Łukasz Kaiser, and Illia Polosukhin. Transformer: Attention is all you need. *Advances in Neural Information Processing Systems 30*, pages 5998–6008, 2017. ISSN 10495258.

Clement Vignac, Igor Krawczuk, Antoine Siraudin, Bohan Wang, Volkan Cevher, and Pascal Frossard. Digress: Discrete denoising diffusion for graph generation. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=UaAD-Nu86WX.

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. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=YkSKZEhIYt.

Carl Yang, Peiye Zhuang, Wenhan Shi, Alan Luu, and Pan Li. Conditional Structure Generation through Graph Variational Generative Adversarial Nets. In *Advances in Neural Information Processing Systems*, volume 32, 2019.

Jiaxuan You, Rex Ying, Xiang Ren, William Hamilton, and Jure Leskovec. GraphRNN: Generating Realistic Graphs with Deep Auto-regressive Models. In *Proceedings of the 35th International Conference on Machine Learning*, pages 5708–5717. PMLR, July 2018. URL https://proceedings.mlr.press/v80/you18a.html. ISSN: 2640-3498.

Chengxi Zang and Fei Wang. MoFlow: An Invertible Flow Model for Generating Molecular Graphs. *Proceedings of the ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, 10:617–626, aug 2020. doi: 10.1145/3394486.3403104. URL https://dl.acm.org/doi/10.1145/3394486.3403104.

Yangtian Zhang, Sizhuang He, Daniel Levine, Lawrence Zhao, David Zhang, Syed A Rizvi, Emanuele Zappala, Rex Ying, and David van Dijk. Non-markovian discrete diffusion with causal language models, 2025. URL https://arxiv.org/abs/2502.09767.

Lingxiao Zhao, Xueying Ding, and Leman Akoglu. Pard: Permutation-invariant autoregressive diffusion for graph generation. In *The Thirty-eighth Annual Conference on Neural Information Processing Systems*, 2024. URL https://openreview.net/forum?id=x4Kk4FxLs3.

A Proofs

A.1 Preliminaries

We recall the gamma function,

$$\Gamma(a) = \int_0^\infty s^{a-1} e^{-s}\,\mathrm{d}s, \quad a \in \mathbb{R}_{>0}, \tag{11}$$

the multivariate beta function,

$$\mathrm{B}(\boldsymbol{\alpha}) = \frac{\prod_{i=1}^K \Gamma(\alpha_i)}{\Gamma\big(\sum_{i=1}^K \alpha_i\big)}, \tag{12}$$

the density of the Dirichlet distribution at a point $\boldsymbol{x} \in \mathbb{S}_K$,

$$\mathrm{Dir}(\boldsymbol{x}; \boldsymbol{\alpha}) = \frac{1}{\mathrm{B}(\boldsymbol{\alpha})} \prod_{i=1}^K x_i^{\alpha_i - 1}, \tag{13}$$

and the density of the $\mathrm{Gamma}(a, b)$ distribution,

$$f(x; a, b) = \frac{b^a}{\Gamma(a)} x^{a-1} e^{-bx}, \quad x > 0,\ a > 0,\ b > 0. \tag{14}$$

A.2 Proposition 3.1: Voronoi Probability

Let $\mathbf{x} \sim \mathrm{Dir}(\mathbf{1} + \alpha_t \boldsymbol{e}_i)$ denote the noisy sample $\mathbf{x}_t$ given $\mathbf{x}_1 = \boldsymbol{e}_i$, and let $a := \alpha_t + 1$; i.e., all Dirichlet parameters equal one except the ith, which equals a. We show that

$$P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) = \sum_{k=0}^{K-1} \frac{(-1)^k \binom{K-1}{k}}{(k+1)^a}. \tag{15}$$

Recall that the Voronoi region of vertex i is defined as

$$\mathbb{V}_i = \{\boldsymbol{x} \in \mathbb{S}_K \mid d(\boldsymbol{x}, \boldsymbol{e}_i) \le d(\boldsymbol{x}, \boldsymbol{e}_j)\ \forall j \ne i\}, \tag{16}$$

with $d(\cdot, \cdot)$ the Euclidean distance. Since $d(\boldsymbol{x}, \boldsymbol{e}_j)^2 = \|\boldsymbol{x}\|^2 - 2x_j + 1$, we have $d(\boldsymbol{x}, \boldsymbol{e}_i) \le d(\boldsymbol{x}, \boldsymbol{e}_j) \iff x_i \ge x_j$, hence

$$\boldsymbol{x} \in \mathbb{V}_i \iff x_i = \max_{j \in [K]} x_j, \tag{17}$$

so that

$$P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) = P\Big(x_i = \max_{j \in [K]} x_j\Big). \tag{18}$$

It is well known that, for any $b > 0$, if $g_1, \ldots, g_K$ are independent with $g_j \sim \mathrm{Gamma}(\alpha_j, b)$, then

$$[x_1, \ldots, x_K]^\top \sim \mathrm{Dir}(\alpha_1, \ldots, \alpha_K), \qquad \text{where } x_j = \frac{g_j}{\sum_{l=1}^K g_l}. \tag{19}$$

Since all coordinates share the same positive denominator, the normalization preserves their ordering:

$$g_i = \max_{j \in [K]} g_j \iff x_i = \max_{j \in [K]} x_j. \tag{20}$$

In the following, we take $b = 1$, with $g_i \sim \mathrm{Gamma}(a, 1)$ and $g_j \sim \mathrm{Gamma}(1, 1)$ for all $j \ne i$, all independent.

From Equations 18 and 20, we get

$$P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) = P\big(g_i \ge g_j\ \ \forall j \ne i\big). \tag{21}$$

The cumulative distribution function of $\mathrm{Gamma}(1, 1)$, i.e., the exponential distribution with unit rate, is $F(s) = 1 - e^{-s}$ for $s \ge 0$. Conditioning on $g_i = s$ and using the independence of the g_j, we have

$$P\big(g_j \le s\ \ \forall j \ne i \,\big|\, g_i = s\big) = (1 - e^{-s})^{K-1}. \tag{22}$$

Integrating against the density of g_i (Equation 14 with $b = 1$) yields

$$P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) = \frac{1}{\Gamma(a)} \int_0^\infty s^{a-1} e^{-s} (1 - e^{-s})^{K-1} \, \mathrm{d}s. \tag{23}$$

By the binomial theorem,

$$(1 - e^{-s})^{K-1} = \sum_{k=0}^{K-1} (-1)^k \binom{K-1}{k} e^{-ks}. \tag{24}$$

Substituting into Equation 23 and exchanging the finite sum and the integral, we obtain

$$P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) = \frac{1}{\Gamma(a)} \sum_{k=0}^{K-1} (-1)^k \binom{K-1}{k} \int_0^\infty s^{a-1} e^{-(k+1)s} \, \mathrm{d}s. \tag{25}$$

With the change of variables $u = (k+1)s$, i.e., $s = u/(k+1)$ and $\mathrm{d}s = \mathrm{d}u/(k+1)$,

$$\begin{aligned}
P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) &= \frac{1}{\Gamma(a)} \sum_{k=0}^{K-1} (-1)^k \binom{K-1}{k} \int_0^\infty \frac{u^{a-1}}{(k+1)^{a-1}} e^{-u} \frac{1}{k+1} du && (26) \\
&= \frac{1}{\Gamma(a)} \sum_{k=0}^{K-1} \frac{(-1)^k \binom{K-1}{k}}{(k+1)^a} \int_0^\infty u^{a-1} e^{-u} du && (27) \\
&= \frac{1}{\Gamma(a)} \sum_{k=0}^{K-1} \frac{(-1)^k \binom{K-1}{k}}{(k+1)^a} \Gamma(a) && (28) \\
&= \sum_{k=0}^{K-1} \frac{(-1)^k \binom{K-1}{k}}{(k+1)^a} && (29)
\end{aligned}$$

Hence,

$$P_{v_i}(\mathbf{x}_t \in \mathbb{V}_i \mid \mathbf{x}_1 = \boldsymbol{e}_i) = \sum_{k=0}^{K-1} \frac{(-1)^k \binom{K-1}{k}}{(k+1)^a}, \tag{30}$$

which concludes the proof.

A.3 Proposition 4.2: Denoising Kernel

Given $P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$ and Assumption 4.1, the one-step denoising kernel satisfies

$$\begin{aligned}
p_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) &= \sum_{\mathbf{x}_1 \in \mathcal{X}} q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1) \, P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) \\
&= \mathbb{E}_{\mathbf{x}_1 \sim P_{1|t}(\cdot|\mathbf{x}_t)} \left[q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1) \right].
\end{aligned} \tag{31}$$

By the law of total probability, we have

$$p_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) = \sum_{\mathbf{x}_1 \in \mathcal{X}} p_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, \mathbf{x}_1) \, P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t). \tag{32}$$

The true denoising process is given by the conditionals of the joint law of the noising process; in particular, $p_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, \mathbf{x}_1) = q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, \mathbf{x}_1)$. By Assumption 4.1, applied at time $t + h$ with $s = t$, this equals $q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1)$, which yields the first equality. The second equality rewrites the sum as an expectation over $\mathbf{x}_1 \sim P_{1|t}(\cdot \mid \mathbf{x}_t)$.

A.4 Proposition 4.4: Approximation Error Bound

The KL divergence between the true and approximate one-step denoising distributions is upper-bounded by the KL divergence between the true and approximate posterior distributions:

$$D_{\mathrm{KL}}\big(p_{t+h}(\cdot \mid \mathbf{x}_t) \,\|\, p^{\theta}_{t+h}(\cdot \mid \mathbf{x}_t)\big) \le D_{\mathrm{KL}}\big(P_{1|t}(\cdot \mid \mathbf{x}_t) \,\|\, P^{\theta}_{1|t}(\cdot \mid \mathbf{x}_t)\big). \tag{33}$$

Here, p^{θ}_{t+h} denotes the denoising kernel of Equation 4 with $P_{1|t}$ replaced by $P^{\theta}_{1|t}$. We first define the joint laws on $\mathcal{X} \times \mathbb{S}_K^L$

$$\rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t) = P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1) \tag{34}$$

and

$$\rho^{\theta}(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t) = P^{\theta}_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1). \tag{35}$$

By Proposition 4.2, their marginals in $\mathbf{x}_{t+h}$ are $p_{t+h}(\cdot \mid \mathbf{x}_t)$ and $p^{\theta}_{t+h}(\cdot \mid \mathbf{x}_t)$, respectively. Moreover, since both joint laws share the same kernel $q_{t+h}(\cdot \mid \mathbf{x}_1)$, it cancels in the log-ratio:

$$\begin{aligned}
D_{\mathrm{KL}}(\rho \,\|\, \rho^{\theta}) &= \sum_{\mathbf{x}_1 \in \mathcal{X}} \int \rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t) \log \frac{\rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)}{\rho^{\theta}(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)} \,\mathrm{d}\mathbf{x}_{t+h} && (36)\\
&= \sum_{\mathbf{x}_1 \in \mathcal{X}} \int P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1) \log \frac{P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)}{P^{\theta}_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)} \,\mathrm{d}\mathbf{x}_{t+h} && (37)\\
&= \sum_{\mathbf{x}_1 \in \mathcal{X}} P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) \log \frac{P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)}{P^{\theta}_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)} \int q_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_1) \,\mathrm{d}\mathbf{x}_{t+h} && (38)\\
&= D_{\mathrm{KL}}\big(P_{1|t}(\cdot \mid \mathbf{x}_t) \,\|\, P^{\theta}_{1|t}(\cdot \mid \mathbf{x}_t)\big), && (39)
\end{aligned}$$

since $q_{t+h}(\cdot \mid \mathbf{x}_1)$ integrates to 1 for every $\mathbf{x}_1$.

The log-sum inequality yields, for each fixed $\mathbf{x}_{t+h}$,

$$\begin{aligned}
&\Big(\sum_{\mathbf{x}_1 \in \mathcal{X}} \rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)\Big) \log \frac{\sum_{\mathbf{x}_1 \in \mathcal{X}} \rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)}{\sum_{\mathbf{x}_1 \in \mathcal{X}} \rho^{\theta}(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)} \\
&\qquad \le \sum_{\mathbf{x}_1 \in \mathcal{X}} \rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t) \log \frac{\rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)}{\rho^{\theta}(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)}.
\end{aligned} \tag{40}$$

Using the definition of the marginals, this becomes

$$p_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) \log \frac{p_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t)}{p^{\theta}_{t+h}(\mathbf{x}_{t+h} \mid \mathbf{x}_t)} \le \sum_{\mathbf{x}_1 \in \mathcal{X}} \rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t) \log \frac{\rho(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)}{\rho^{\theta}(\mathbf{x}_1, \mathbf{x}_{t+h} \mid \mathbf{x}_t)}. \tag{41}$$

Integrating over $\mathbf{x}_{t+h}$ gives

$$D_{\mathrm{KL}}\big(p_{t+h}(\cdot \mid \mathbf{x}_t) \,\|\, p^{\theta}_{t+h}(\cdot \mid \mathbf{x}_t)\big) \le D_{\mathrm{KL}}(\rho \,\|\, \rho^{\theta}). \tag{42}$$

Combining this with the previous identity yields

$$D_{\mathrm{KL}}\big(p_{t+h}(\cdot \mid \mathbf{x}_t) \,\|\, p^{\theta}_{t+h}(\cdot \mid \mathbf{x}_t)\big) \le D_{\mathrm{KL}}\big(P_{1|t}(\cdot \mid \mathbf{x}_t) \,\|\, P^{\theta}_{1|t}(\cdot \mid \mathbf{x}_t)\big).$$

A.5 Proposition 4.3: Convergence

Fix $t \in [0, 1)$. Throughout this proof, we write $q := q_t$, $\varepsilon := \varepsilon_t$, and $\pi := \pi_t$, where $\pi(\mathbf{x}) = \sum_{\mathbf{x}_1 \in \mathcal{X}} p(\mathbf{x}_1)\, q(\mathbf{x} \mid \mathbf{x}_1)$ is the marginal of the noising process at time t. Assume:

(A1) **(Overlapping noising densities)** The noising densities share a positive amount of probability mass:

$$\varepsilon \;:=\; \int_{\mathbb{S}_K^L} \min_{\mathbf{x}_1 \in \mathcal{X}} q(\mathbf{x} \mid \mathbf{x}_1) \,\mathrm{d}\mathbf{x} \;>\; 0. \tag{43}$$

(A2) **(Exact posterior)** $P_{1|t}$ is the Bayes posterior of the noising process: for every $\mathbf{x}_1 \in \mathcal{X}$ and every $\mathbf{x}_t$ with $\pi(\mathbf{x}_t) > 0$,

$$P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) \;=\; \frac{p(\mathbf{x}_1)\, q(\mathbf{x}_t \mid \mathbf{x}_1)}{\pi(\mathbf{x}_t)}. \tag{44}$$

Where $\pi(\mathbf{x}_t) = 0$, $P_{1|t}(\cdot \mid \mathbf{x}_t)$ may be any distribution on $\mathcal{X}$.

Define the Markov kernel on $\mathbb{S}_K^L$ that denoises $\mathbf{x}_t$ and then renoises the prediction at the same noise level t, independently of $\mathbf{x}_t$:

$$p_{t'|t}(\mathbf{x}'_t \mid \mathbf{x}_t) \;=\; \sum_{\mathbf{x}_1 \in \mathcal{X}} P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, q(\mathbf{x}'_t \mid \mathbf{x}_1), \tag{45}$$

and let Q_t denote the associated Markov operator, acting on a distribution μ as $(\mu Q_t)(\mathbf{x}'_t) := \int \mu(\mathbf{x}_t)\, p_{t'|t}(\mathbf{x}'_t \mid \mathbf{x}_t)\, \mathrm{d}\mathbf{x}_t$. Then π is the unique stationary distribution of $p_{t'|t}$, and for every initial distribution μ_0 on $\mathbb{S}_K^L$,[1]

$$\left\|\mu_0 Q_t^n - \pi\right\|_{\mathrm{TV}} \;\le\; (1-\varepsilon)^n, \qquad n \ge 0. \tag{46}$$

In particular, the law of the iterates converges to π in total variation from any initialization, at a geometric rate.

We (1) check that Equation 45 is a valid Markov kernel, (2) show that π is stationary, (3) show that every transition lies above a common floor of mass ε, and (4) deduce geometric convergence and uniqueness.

1. $p_{t'|t}$ is a valid Markov kernel. For fixed $\mathbf{x}_t$, the posterior weights $P_{1|t}(\cdot \mid \mathbf{x}_t)$ are nonnegative and sum to one over $\mathcal{X}$. The kernel is therefore nonnegative, and

$$\int_{\mathbb{S}_K^L} p_{t'|t}(\mathbf{x}'_t \mid \mathbf{x}_t)\, \mathrm{d}\mathbf{x}'_t \;=\; \sum_{\mathbf{x}_1 \in \mathcal{X}} P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) \int_{\mathbb{S}_K^L} q(\mathbf{x}'_t \mid \mathbf{x}_1)\, \mathrm{d}\mathbf{x}'_t \;=\; \sum_{\mathbf{x}_1 \in \mathcal{X}} P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) \;=\; 1. \tag{47}$$

2. π is stationary. We first note the Bayes identity

$$\pi(\mathbf{x}_t)\, P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t) \;=\; p(\mathbf{x}_1)\, q(\mathbf{x}_t \mid \mathbf{x}_1) \qquad \text{for all } \mathbf{x}_t \in \mathbb{S}_K^L,\ \mathbf{x}_1 \in \mathcal{X}. \tag{48}$$

Where $\pi(\mathbf{x}_t) > 0$, this is Equation 44 multiplied by $\pi(\mathbf{x}_t)$. Where $\pi(\mathbf{x}_t) = 0$, both sides vanish, since $\pi(\mathbf{x}_t) = 0$ implies $p(\mathbf{x}_1)\, q(\mathbf{x}_t \mid \mathbf{x}_1) = 0$ for all $\mathbf{x}_1 \in \mathcal{X}$.

Using Equation 45, exchanging the finite sum and the integral, and applying Equation 48,

$$(\pi Q_t)(\mathbf{x}'_t) = \int_{\mathbb{S}_K^L} \pi(\mathbf{x}_t)\, p_{t'|t}(\mathbf{x}'_t \mid \mathbf{x}_t)\, \mathrm{d}\mathbf{x}_t \tag{49}$$

$$= \sum_{\mathbf{x}_1 \in \mathcal{X}} q(\mathbf{x}'_t \mid \mathbf{x}_1) \int_{\mathbb{S}_K^L} \pi(\mathbf{x}_t)\, P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, \mathrm{d}\mathbf{x}_t \tag{50}$$

$$= \sum_{\mathbf{x}_1 \in \mathcal{X}} q(\mathbf{x}'_t \mid \mathbf{x}_1)\, p(\mathbf{x}_1) \int_{\mathbb{S}_K^L} q(\mathbf{x}_t \mid \mathbf{x}_1)\, \mathrm{d}\mathbf{x}_t \tag{51}$$

$$= \sum_{\mathbf{x}_1 \in \mathcal{X}} p(\mathbf{x}_1)\, q(\mathbf{x}'_t \mid \mathbf{x}_1) \;=\; \pi(\mathbf{x}'_t). \tag{52}$$

Hence $\pi Q_t = \pi$: the marginal π is stationary for $p_{t'|t}$.

3. Doeblin minorization. Let

$$g(\mathbf{x}) \;:=\; \min_{\mathbf{x}_1 \in \mathcal{X}} q(\mathbf{x} \mid \mathbf{x}_1), \qquad \text{so that} \qquad \varepsilon \;=\; \int_{\mathbb{S}_K^L} g(\mathbf{x})\, \mathrm{d}\mathbf{x} \tag{53}$$

[1] We use the convention $\|\mu - \mu'\|_{\mathrm{TV}} = \sup_A |\mu(A) - \mu'(A)|$, which lies in $[0, 1]$.

by Equation 5. The function g is the pointwise lower envelope of the noising densities, and ε is the probability mass they all share. We have $\varepsilon > 0$ by (A1), and $\varepsilon \le 1$ since $g \le q(\cdot \mid \mathbf{x}_1)$.

Whichever $\mathbf{x}_1$ is sampled, the renoising density lies above g, and since the posterior weights sum to one (Step 1), so does their mixture: for all $\mathbf{x}_t, \mathbf{x}'_t$,

$$p_{t'|t}(\mathbf{x}'_t \mid \mathbf{x}_t) \;=\; \sum_{\mathbf{x}_1 \in \mathcal{X}} P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, q(\mathbf{x}'_t \mid \mathbf{x}_1) \;\ge\; g(\mathbf{x}'_t). \tag{54}$$

Writing $\nu := g/\varepsilon$ for the normalized floor, this is the uniform Doeblin minorization

$$p_{t'|t}(\mathbf{x}'_t \mid \mathbf{x}_t) \;\ge\; \varepsilon\, \nu(\mathbf{x}'_t) \qquad \text{for all } \mathbf{x}_t, \mathbf{x}'_t. \tag{55}$$

4. Geometric convergence and uniqueness. If $\varepsilon = 1$, then each $q(\cdot \mid \mathbf{x}_1)$ coincides with g, so $p_{t'|t}(\cdot \mid \mathbf{x}_t) = \pi$ for every $\mathbf{x}_t$ and the chain reaches π in one step. Otherwise, by Equation 55, each transition can be written as a mixture

$$p_{t'|t}(\cdot \mid \mathbf{x}_t) \;=\; \varepsilon\, \nu \;+\; (1-\varepsilon)\, r(\cdot \mid \mathbf{x}_t), \qquad r(\cdot \mid \mathbf{x}_t) := \frac{p_{t'|t}(\cdot \mid \mathbf{x}_t) - \varepsilon\, \nu}{1-\varepsilon}, \tag{56}$$

where r is again a transition density: it is nonnegative by Equation 55 and integrates to one. In words, at every step the chain ignores its current state with probability ε and draws a fresh sample from ν.

Now compare the chain started from two distributions μ and μ', and let R denote the Markov operator of r. The fresh-sample component $\varepsilon\nu$ is the same for both, so it cancels after one step:

$$\mu Q_t - \mu' Q_t \;=\; (1-\varepsilon)\,(\mu R - \mu' R). \tag{57}$$

Since applying a Markov kernel never increases the total variation distance,

$$\|\mu Q_t - \mu' Q_t\|_{\mathrm{TV}} \;\le\; (1-\varepsilon)\, \|\mu - \mu'\|_{\mathrm{TV}}. \tag{58}$$

That is, each step shrinks the distance between any two distributions by a factor of at least $1-\varepsilon$.

Taking $\mu = \mu_0$ and $\mu' = \pi$, using $\pi Q_t = \pi$ (Step 2), and iterating n times gives

$$\|\mu_0 Q_t^n - \pi\|_{\mathrm{TV}} \;\le\; (1-\varepsilon)^n\, \|\mu_0 - \pi\|_{\mathrm{TV}} \;\le\; (1-\varepsilon)^n, \tag{59}$$

which is Equation 46.

Finally, if π' is another stationary distribution, then $\|\pi' - \pi\|_{\mathrm{TV}} = \|\pi' Q_t^n - \pi Q_t^n\|_{\mathrm{TV}} \le (1-\varepsilon)^n \to 0$, so $\pi' = \pi$, which proves uniqueness.

A.6 Proposition 4.5: UNSID Contracts Errors, Flow Matching Preserves Them

Let p and $\mathcal{X}$ be as in Section 4. For each $t \in [0,1)$ and $\mathbf{x}_1 \in \mathcal{X}$, let $q_t(\cdot \mid \mathbf{x}_1)$ be the noising density at time t on $\mathbb{S}_K^L$, with marginal

$$\pi_t(\mathbf{x}) \;:=\; \sum_{\mathbf{x}_1 \in \mathcal{X}} p(\mathbf{x}_1)\, q_t(\mathbf{x} \mid \mathbf{x}_1). \tag{60}$$

Fix a time t and a step size $h > 0$ with $t + h < 1$, and let $\tilde{\pi}_t$ be *any* distribution on $\mathbb{S}_K^L$, representing the law of the sampler at time t, including the error accumulated so far.

(a) **UNSID contracts the error.** Let $P_{1|t}$ be the true posterior, let

$$p_{t+h}(\mathbf{x} \mid \mathbf{x}_t) \;=\; \sum_{\mathbf{x}_1 \in \mathcal{X}} P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, q_{t+h}(\mathbf{x} \mid \mathbf{x}_1) \tag{61}$$

be the UNSID denoising kernel (Equation 4), and let Q be the associated Markov operator. Assume that the noising densities at time $t+h$ overlap:

$$\varepsilon_{t+h} \;:=\; \int_{\mathbb{S}_K^L} \min_{\mathbf{x}_1 \in \mathcal{X}} q_{t+h}(\mathbf{x} \mid \mathbf{x}_1)\, \mathrm{d}\mathbf{x} \;>\; 0. \tag{62}$$

Then

$$\|\tilde{\pi}_t Q - \pi_{t+h}\|_{\mathrm{TV}} \;\le\; (1-\varepsilon_{t+h})\, \|\tilde{\pi}_t - \pi_t\|_{\mathrm{TV}}. \tag{63}$$

(b) **Flow matching preserves the error.** Let $\Phi : \mathbb{S}_K^L \to \mathbb{S}_K^L$ be the flow-matching step from t to $t+h$. Assume that Φ is a measurable bijection with measurable inverse and that it transports the true marginals, $\Phi_{\#}\pi_t = \pi_{t+h}$. Then

$$\|\Phi_{\#}\tilde{\pi}_t - \pi_{t+h}\|_{\mathrm{TV}} \;=\; \|\tilde{\pi}_t - \pi_t\|_{\mathrm{TV}}. \tag{64}$$

(a) UNSID contracts the error. We first show that $\pi_t Q = \pi_{t+h}$. By the Bayes identity at time t (Equation 48), $\int_{\mathbb{S}_K^L} \pi_t(\mathbf{x}_t)\, P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, \mathrm{d}\mathbf{x}_t = p(\mathbf{x}_1)$, hence

$$(\pi_t Q)(\mathbf{x}) = \sum_{\mathbf{x}_1 \in \mathcal{X}} q_{t+h}(\mathbf{x} \mid \mathbf{x}_1) \int_{\mathbb{S}_K^L} \pi_t(\mathbf{x}_t)\, P_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)\, \mathrm{d}\mathbf{x}_t \tag{65}$$

$$= \sum_{\mathbf{x}_1 \in \mathcal{X}} p(\mathbf{x}_1)\, q_{t+h}(\mathbf{x} \mid \mathbf{x}_1) \tag{66}$$

$$= \pi_{t+h}(\mathbf{x}). \tag{67}$$

Next, Steps 3 and 4 of the proof of Proposition 4.3 only use the facts that the posterior weights sum to one and that the renoising densities overlap. Applied with q_{t+h} in place of q_t, they yield the minorization $p_{t+h}(\cdot \mid \mathbf{x}_t) \geq \varepsilon_{t+h}\, \nu_{t+h}$, where $\nu_{t+h} := \min_{\mathbf{x}_1 \in \mathcal{X}} q_{t+h}(\cdot \mid \mathbf{x}_1)/\varepsilon_{t+h}$. They therefore also yield the contraction of Equation 58, with Q and ε_{t+h} in place of Q_t and ε. Taking $\mu = \tilde{\pi}_t$ and $\mu' = \pi_t$,

$$\|\tilde{\pi}_t Q - \pi_{t+h}\|_{\mathrm{TV}} \;=\; \|\tilde{\pi}_t Q - \pi_t Q\|_{\mathrm{TV}} \;\leq\; (1 - \varepsilon_{t+h})\, \|\tilde{\pi}_t - \pi_t\|_{\mathrm{TV}}. \tag{68}$$

(b) Flow matching preserves the error. For any measurable set A, $(\Phi_{\#}\tilde{\pi}_t)(A) - (\Phi_{\#}\pi_t)(A) = \tilde{\pi}_t(B) - \pi_t(B)$ with $B = \Phi^{-1}(A)$. Because Φ is a measurable bijection with measurable inverse, $A \mapsto \Phi^{-1}(A)$ is a one-to-one correspondence between measurable sets: every measurable B arises this way, from $A = \Phi(B)$. Taking the supremum over A is therefore the same as taking it over B:

$$\|\Phi_{\#}\tilde{\pi}_t - \Phi_{\#}\pi_t\|_{\mathrm{TV}} \;=\; \sup_A \big|\tilde{\pi}_t(\Phi^{-1}(A)) - \pi_t(\Phi^{-1}(A))\big| \;=\; \sup_B \big|\tilde{\pi}_t(B) - \pi_t(B)\big| \;=\; \|\tilde{\pi}_t - \pi_t\|_{\mathrm{TV}}. \tag{69}$$

Using $\Phi_{\#}\pi_t = \pi_{t+h}$ gives Equation 64.

B Probability Path and Prior Choices

A natural choice for the noise prior is the uniform distribution over the simplex, $q_0(\mathbf{x}) = \mathrm{Dir}(\mathbf{1})$. However, graphs are typically sparse, with highly imbalanced edge distributions. In discrete diffusion, Vignac et al. [2023] has shown that the empirical marginal distribution is theoretically optimal and empirically effective.

Motivated by this result, we consider a marginal-weighted Dirichlet mixture prior: $q_0^{\mathrm{marg}}(\mathbf{x}) = \sum_{k=1}^K m_k \mathrm{Dir}(\mathbf{x}; \mathbf{1} + \kappa \boldsymbol{e}_k)$, where $\boldsymbol{m} = (m_1, \ldots, m_K)$ denotes the empirical marginal over categories, $\boldsymbol{e}_k$ is the kth standard basis vector, and $\kappa > 0$ is a user-specified concentration parameter.

We can directly parametrize the prior as such, but we can also approximate this prior using a model trained with the probability path $q_t^\star(\mathbf{x}_t \mid \mathbf{x}_1) := \mathrm{Dir}(\mathbf{x}_t; \mathbf{1} + \alpha_t \mathbf{x}_1)$. In that case, we have dimension-wise: $\mathbb{E}_{\mathbf{x}_1 \sim p_{\text{data}}}[q_t^\star(\mathbf{x}_t \mid \mathbf{x}_1)] = q_0^{\mathrm{marg}}(\mathbf{x}_0) \quad \Longleftrightarrow \quad \alpha_t = \kappa$. This identity does not directly extend to the L-dimensional joint, since in general the L dimensions are not independent. Nevertheless, when $q_t^\star$ is sufficiently close to the stationary distribution $q_0^\star$, we can leverage the following approximation:

$$q_t^\star\big(\mathbf{x}_t^{(1:L)} \mid \mathbf{x}_1^{(1:L)}\big) \approx \prod_{l=1}^L q_t^\star\big(\mathbf{x}_t^{(l)} \mid \mathbf{x}_1^{(l)}\big) = q_0^{\mathrm{marg}}(\mathbf{x}^{(1:L)}), \tag{70}$$

with $\alpha_t = \kappa$.

Operationally, we can thus initialize inference with $\mathbf{x}_0^{(1:L)} \sim q_0^{\mathrm{marg}}$, and then run the reverse process along the path $q_t^\star$ with $\alpha_0 = \kappa$. We show in Section 5 that this initialization improves generative performance.

C Supplementary Discussion

C.1 Extended Related Work on Graph Generation

A central challenge in generative graph modeling arises from node permutations: a graph with n nodes can be represented by up to $n!$ different adjacency matrices. This has motivated two dominant families of methods: *sequential* (autoregressive) approaches, which generate graphs step by step by adding nodes, edges, or higher-order motifs according to a fixed or learned ordering [You et al., 2018, Shi et al., 2020, Luo et al., 2021, Liao et al., 2019, Goyal et al., 2020, Kong et al., 2023, Zhao et al., 2024], and *permutation-equivariant* models, which enforce equivariance by design. Permutation-equivariant architectures have been instantiated within several generative frameworks, including variational autoencoders [Simonovsky and Komodakis, 2018], generative adversarial networks [De Cao and Kipf, 2018, Krawczuk et al., 2021, Martinkus et al., 2022], normalizing flows [Madhawa et al., 2019, Zang and Wang, 2020, Liu et al., 2019], and vector-quantized autoencoders [Boget et al., 2024, Nguyen et al., 2024].

Recently, *denoising models* coupled with equivariant GNNs have become the dominant approach, as they have proven highly effective. Continuous-space methods include score-based diffusion [Yang et al., 2019, Jo et al., 2022], diffusion bridges [Jo et al., 2024], and flow matching [Eijkelboom et al., 2024]. Discrete formulations, such as diffusion in discrete time [Haefeli et al., 2022, Vignac et al., 2023] or in continuous time [Xu et al., 2024], flow matching [Qin et al., 2025], and non-Markovian discrete diffusion [Boget, 2025], also report strong results. However, these approaches do not perform *smooth* diffusion on the probability simplex: they either add noise to graphs in an unconstrained Euclidean space or operate as jump processes over discrete states.

Beta diffusion [Liu et al., 2025] applies continuous noise on the 1-simplex via the beta distribution and handles K-categorical variables by encoding them as K binary variables. Eijkelboom et al. [2024] implement Dirichlet flow matching for graphs as a baseline but report poor performance (consistent with our DIRIFM results), which leads them to relax the noising process to $\mathbb{R}^K$. More recently, the Graph Bayesian Flow Network [Song et al., 2025] proposes a diffusion mechanism that adds Gaussian noise in $\mathbb{R}^K$, followed by a projection onto the simplex.

Orthogonally, a recent line of work targets scalability. A key limitation of standard models is their reliance on dense representations and all-pairs computations, which impedes scaling to large graphs. To address this, prior work has proposed scalable denoising architectures and sampling strategies [Qin et al., 2024, Chen et al., 2023, Karami, 2024, Bergmeister et al., 2024, Boget et al., 2025]. Notably, Qin et al. [2024] and Boget et al. [2025] introduce methods that enable permutation-equivariant models to scale to large graphs. These techniques are complementary to our approach and could be used to scale UNSID; we leave this exploration to future work.

C.2 Denoiser Expressivity in Discrete and Continuous Settings

GNNs have limited expressivity because of inherent graph symmetries [Morris et al., 2019]. These limitations, however, disappear once each node has a unique identifier. In particular, Loukas [2020] demonstrates that sufficiently deep message-passing architectures become universal when nodes can be uniquely distinguished. Continuous noise naturally provides such distinguishing information, effectively acting as a unique identifier for each node. Consequently, injecting continuous noise enhances the expressive power of GNN-based denoisers.

In practice, discrete denoisers often require additional handcrafted features to overcome symmetry-induced limitations. Common examples include spectral embeddings derived from Laplacian eigenvectors [Vignac et al., 2023] and relative random-walk probability features [Qin et al., 2025]. These features must be recomputed for every noisy graph, during both training and sampling, adding non-negligible computational overhead.

A typical criticism of continuous models is that adding continuous noise to edge attributes may collapse the graph into a fully connected weighted structure, ostensibly erasing the discrete adjacency information. However, the graph operations underlying the aforementioned extra features—such as spectral decompositions and random-walk statistics—remain well defined on weighted graphs. Thus, continuous perturbations do not destroy the structure required for these computations. Moreover, since continuous noise serves as an intrinsic node identifier, it eliminates the need for such auxiliary features altogether.

C.3 Comparison with Simple Iterative Denoising

Our UNSID framework can be viewed as a simplex analogue of Simple Iterative Denoising (SID) [Boget, 2025]. Below, we outline the key similarities and differences.

Noising. In SID, the corruption process follows discrete diffusion for categorical data [Austin et al., 2021], where intermediate noisy distributions linearly interpolate between a point mass at the clean data and a stationary prior:

$$q_t(\mathbf{x} \mid \mathbf{x}_1) = \alpha_t \delta_{\mathbf{x}_1}(\mathbf{x}) + (1 - \alpha_t) q_0(\mathbf{x}). \tag{71}$$

As discussed in Section 3, interpolation-based constructions are ill-suited to the continuous simplex setting. Accordingly, UNSID adopts an explicit probability path on the simplex.

Independence Assumption. Both SID and UNSID rely on the same conditional-independence assumption (Assumption 4.1), which is central to the formulation.

Denoising. The denoising rules in UNSID follow directly from the conditional-independence assumption and are closely related in spirit to SID. A notable distinction is our original hybrid construction, which couples a *continuous* kernel q_t on the simplex with a *discrete* posterior $P_{1|t}$.

Convergence. Proposition 4.3 and its proof are original to this work. SID provides no analogous convergence result. We regard this result as a core theoretical contribution.

Advantages over Diffusion and Flow Matching. Although there are conceptual similarities at a high level, including the connection to compounding denoising errors, the underlying intuition differs substantially between the continuous and discrete settings. In particular, the observation that $\mathbf{x}_{t+h}$ remains close to $\mathbf{x}_t$, and the reasoning that follows from it, hold only in the continuous case and do not translate directly to discrete models.

Parameterization and Learning. Both approaches employ standard parameterizations and training objectives common to discrete diffusion and flow-matching methods.

Probability Paths and Priors. Our choices and analysis of probability paths and priors are specific to continuous simplex noise and have no direct counterpart in SID.

Guidance. As explained in the main text, guidance in UNSID is obtained via direct adaptations of standard diffusion techniques (classifier and classifier-free guidance). SID does not include a guidance mechanism.

Beyond this point-by-point comparison, our primary contribution is integrating these components into a simple, efficient framework for graph generation that achieves state-of-the-art performance on three of the five datasets considered, while remaining competitive with the strongest baselines on the other two.

C.4 Comments on the `SBM` Experiments

MMD-based metrics do not penalize dataset memorization: a model that reproduces training graphs obtains near-perfect validity and MMDs close to the training-set reference, and a model that reproduces test graphs obtains near-zero MMDs. On `SBM`, two independent observations point in this direction for DeFoG:

- **Novelty.** `SBM` graphs have on the order of 100 nodes, i.e., thousands of node pairs, so the probability that two independently sampled graphs are isomorphic is negligible. A novelty of 90% therefore cannot be explained by chance and indicates that about 10% of the samples reproduce training graphs.
- **Reference MMD.** The MMD between the training and test splits is the natural reference level for a model that generalizes. DeFoG reports a degree MMD of 0.6, less than half of this reference (1.57), which is difficult to reconcile with sampling from the data distribution and may indicate overfitting to the evaluation split.

Taken together, these observations suggest that the validity and MMD results of DEFOG on SBM should be interpreted with caution.

C.5 Use of LLMs

We used large language models (LLMs) for editing, consistency checks, identifying typographical errors, and formatting tables. LLMs were also used to check and correct the proofs and to improve their readability. No code, analyses, or results were generated by LLMs.

D Guidance

Classifier guidance and classifier-free guidance are two standard mechanisms for steering generation toward desired properties. Classifier guidance augments an *unconditional* generative model with the gradient of a separately trained classifier (or regressor) for the target property. Classifier-free guidance does not require an auxiliary classifier, but it does require both a *conditional* and an *unconditional* generative model, typically obtained by training a single network with randomly masked conditioning vectors.

Below, we first present the key ideas and then provide the derivations.

D.1 Key Ideas

Let y denote the target property, let $p(y \mid \mathbf{x})$ be the conditional distribution of y given a noisy state $\mathbf{x}$, and let $p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t)$ denote the unconditional reverse transition from time t to $t+h$. We modulate the strength of conditioning with a scale $\omega > 0$ via

$$p^{\omega}_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) \;\propto\; p(y \mid \mathbf{x}_{t+h})^{\omega}\, p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t). \tag{72}$$

Classifier guidance. Here, $p(y \mid \mathbf{x})$ is approximated by a classifier $p_\phi(y \mid \mathbf{x})$ trained at all noise levels. During sampling, one combines a reverse step with an update in the direction of the classifier gradient $\nabla_{\mathbf{x}} \log p_\phi(y \mid \mathbf{x})|_{\mathbf{x}=\mathbf{x}_t}$.

Classifier-free guidance. In this approach, no external classifier is needed; instead, one uses a conditional reverse model $p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y)$ and an unconditional one $p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t)$, typically produced by a single network via masking. A standard identity yields the log-linear interpolation

$$\log p^{\omega}_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) = \omega \log p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) + (1-\omega) \log p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) + C, \tag{73}$$

where C does not depend on $\mathbf{x}_{t+h}$.

D.2 Derivations

Assume that, given $\mathbf{x}_{t+h}$, the noisier state $\mathbf{x}_t$ carries no additional information about y, i.e., $q(\mathbf{x}_t \mid \mathbf{x}_{t+h}, y) = q(\mathbf{x}_t \mid \mathbf{x}_{t+h})$. By Bayes' rule and this assumption,

$$p(y \mid \mathbf{x}_t, \mathbf{x}_{t+h}) = \frac{q(\mathbf{x}_t \mid \mathbf{x}_{t+h}, y)\, p(y \mid \mathbf{x}_{t+h})}{q(\mathbf{x}_t \mid \mathbf{x}_{t+h})} = p(y \mid \mathbf{x}_{t+h}). \tag{74}$$

Consequently,

$$p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) = \frac{p(y \mid \mathbf{x}_t, \mathbf{x}_{t+h})\, p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t)}{p(y \mid \mathbf{x}_t)} \;\propto\; p(y \mid \mathbf{x}_{t+h})\, p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t). \tag{75}$$

Introducing the scale $\omega > 0$ controls the strength of the guidance, with $\omega > 1$ sharpening it:

$$p^{\omega}_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) \;\propto\; p(y \mid \mathbf{x}_{t+h})^{\omega}\, p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t). \tag{76}$$

The above assumption holds exactly for Markovian noising processes. Under Assumption 4.1, however, $\mathbf{x}_t$ and $\mathbf{x}_{t+h}$ are conditionally independent noisy views of $\mathbf{x}_1$, so $\mathbf{x}_t$ may carry additional information about y. In other words, Equation 76 is exact for Markovian models and serves as an approximation for UNSID.

D.2.1 Classifier-Free Guidance

Applying Bayes' rule to $p(y \mid \mathbf{x}_{t+h}) = p(y \mid \mathbf{x}_t, \mathbf{x}_{t+h})$ in Equation 76, we obtain

$$p^{\omega}_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) \propto \left(\frac{p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y)\, p(y \mid \mathbf{x}_t)}{p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t)} \right)^{\omega} p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) \tag{77}$$

$$= p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y)^{\omega}\, p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t)^{1-\omega}\, p(y \mid \mathbf{x}_t)^{\omega}. \tag{78}$$

Since $p(y \mid \mathbf{x}_t)^{\omega}$ does not depend on $\mathbf{x}_{t+h}$, taking the logarithm and absorbing this term into the constant yields

$$\log p^{\omega}_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) = \omega \log p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) + (1-\omega) \log p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) + C. \tag{79}$$

D.2.2 Classifier Guidance

For classifier guidance, we start from Equation 76, approximate $p(y \mid \mathbf{x}_{t+h})$ by the classifier $p_\phi(y \mid \mathbf{x}_{t+h})$, and linearize its logarithm around $\mathbf{x}_t$ with a first-order Taylor expansion:

$$\log p_\phi(y \mid \mathbf{x}_{t+h}) \approx \log p_\phi(y \mid \mathbf{x}_t) + (\mathbf{x}_{t+h} - \mathbf{x}_t)^\top \boldsymbol{g}_t = \mathbf{x}_{t+h}^\top \boldsymbol{g}_t + C = \sum_{i=1}^{L} \big(\mathbf{x}^{(i)}_{t+h}\big)^\top \boldsymbol{g}^{(i)}_t + C, \tag{80}$$

where $\boldsymbol{g}_t := \nabla_{\mathbf{x}} \log p_\phi(y \mid \mathbf{x})|_{\mathbf{x}=\mathbf{x}_t}$, $\boldsymbol{g}^{(i)}_t$ is its block for dimension i, and C does not depend on $\mathbf{x}_{t+h}$. Substituting into Equation 76 gives

$$\log p^{\omega}_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t, y) \approx \log p_{t+h|t}(\mathbf{x}_{t+h} \mid \mathbf{x}_t) + \omega\, \mathbf{x}_{t+h}^\top \boldsymbol{g}_t + C. \tag{81}$$

This shows that classifier guidance tilts the unconditional reverse kernel in the direction of the classifier gradient, toward states with a higher likelihood of the target property.

E Technical Report

E.1 GNN Architecture

Our denoiser uses the same architecture as SID [Boget, 2025]; we reproduce its description below for completeness. The denoisers are GNNs inspired by the general, powerful, scalable (GPS) graph Transformer. A single layer is described as

$$\tilde{\boldsymbol{X}}^\ell, \tilde{\boldsymbol{E}}^\ell = \text{MPNN}(\boldsymbol{X}^\ell, \boldsymbol{E}^\ell), \tag{82}$$

$$\boldsymbol{X}^{\ell+1} = \text{MultiheadAttention}(\tilde{\boldsymbol{X}}^\ell + \boldsymbol{X}^\ell) + \tilde{\boldsymbol{X}}^\ell, \tag{83}$$

$$\boldsymbol{E}^{\ell+1} = \tilde{\boldsymbol{E}}^\ell + \boldsymbol{E}^\ell, \tag{84}$$

where $\boldsymbol{X}^\ell$ and $\boldsymbol{E}^\ell$ are the node and edge hidden representations at the input of the lth layer. The MultiheadAttention layer is the classical multi-head attention layer of Vaswani et al. [2017], and MPNN is a message-passing neural network layer described below.

The MPNN operates on node and edge representations as follows:

$$\boldsymbol{h}^\ell_{i,j} = \text{ReLU}\big(\boldsymbol{W}^\ell_{\text{src}} \boldsymbol{x}^\ell_i + \boldsymbol{W}^\ell_{\text{trg}} \boldsymbol{x}^\ell_j + \boldsymbol{W}^\ell_{\text{edge}} \boldsymbol{a}^\ell_{i,j}\big), \tag{85}$$

$$\boldsymbol{a}^{\ell+1}_{i,j} = \text{LayerNorm}\big(f_{\text{edge}}(\boldsymbol{h}^\ell_{i,j})\big), \tag{86}$$

$$\boldsymbol{x}^{\ell+1}_i = \text{LayerNorm}\Big(\boldsymbol{x}^\ell_i + \sum_{j \in \mathcal{N}(i)} f_{\text{node}}(\boldsymbol{h}^\ell_{i,j})\Big), \tag{87}$$

with $\boldsymbol{W}^\ell_{\text{src}}$, $\boldsymbol{W}^\ell_{\text{trg}}$, and $\boldsymbol{W}^\ell_{\text{edge}}$ denoting trainable weight matrices, and f_{node} and f_{edge} being small neural networks.

The node hidden states $\boldsymbol{x}_i$ and the outputs of f_{node} are of dimension d_h, a tunable hyperparameter (see Table 4). The edge hidden states $\boldsymbol{a}_{i,j}$, intermediate messages $\boldsymbol{h}^\ell_{i,j}$, and the outputs of f_{edge} have dimension $d_h/2$ or $d_h/4$.

Inputs and Outputs At the input, we concatenate the node attributes, extra features, and time step into node features, copying graph-level information (e.g., the time step or graph size) to each node. The node and edge input vectors are then projected to their respective hidden dimensions, d_h for nodes and $d_h/4$ for edges.

Similarly, the outputs of the final layer are projected to their respective dimensions, d_v for nodes and $d_a + 1$ for edges. To enforce edge symmetry, we replace each edge output $\boldsymbol{a}_{i,j}$ by $(\boldsymbol{a}_{i,j} + \boldsymbol{a}_{j,i})/2$. Finally, we ensure that the outputs can be interpreted as probabilities by applying a softmax or sigmoid function, as appropriate.

E.2 Probability Path and Noise Schedule

For all experiments, we use a probability path parameterized via the Dirichlet distribution, $\mathrm{Dir}(\mathbf{x}_t; \mathbf{1} + \alpha_t \mathbf{x}_1)$, with $\alpha_t = -a \log(1-t)$. The hyperparameter a controls the noising dynamics (see Figure 3). We use $a = 3$ in all our experiments, except on `SBM`, where we use $a = 2$.

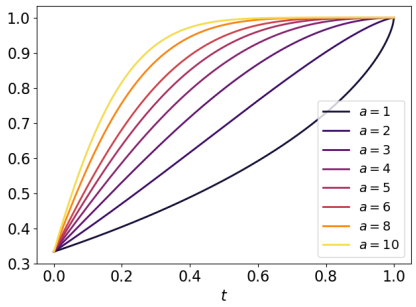


Figure 3: Voronoi probabilities over time for $\mathbf{x}_t \sim \mathrm{Dir}(\mathbf{1} + \alpha_t \mathbf{x}_1)$ on $\mathbb{S}_3$, with $\alpha_t = -a \log(1-t)$, for various values of a. For $a = 1$, P_{v_k} remains low for most of the trajectory and only increases sharply as $t \to 1$; for $a = 10$, P_{v_k} is already close to 1 by $t = 0.6$. Suitable choices of a therefore likely lie between 1 and 10.

E.3 Hyperparameters

Table 4: Hyperparameters.

	`Planar`	`SBM`	`QM9H`	`ZINC250K`
GNN layers	8	8	8	8
Hidden layers in MLPs	2	2	2	2
Node representation size	256	256	256	256
Edge representation size	64	128	128	128
Denoising steps	128	512	1024	1024
Learning rate	0.0002	0.0005	0.0005	0.0005
Optimizer	AdamW	AdamW	AdamW	AdamW

For discrete diffusion (DISCDIFF and NM-DD), we use the cosine schedule [Nichol and Dhariwal, 2021] and the marginal noise distribution.

E.4 Extra Features

For discrete diffusion (Markovian and non-Markovian, i.e., DISCDIFF and NM-DD), we follow common practice [Vignac et al., 2023, Qin et al., 2025, Boget et al., 2025] and enhance the discrete graph representation with synthetic extra node features. For UNSID and Dirichlet flow matching, we add only the graph size.

We use the following extra features: spectral features (`ZINC250K`, `Planar`, `SBM`), graph size, molecular features (`ZINC250K`), cycle information (`Planar`), and relative random-walk probabilities

(RRWPs; QM9H). All these features are concatenated to the input node attributes; the RRWPs are additionally concatenated to the edge attributes.

Spectral features We use the eigenvectors associated with the k lowest eigenvalues of the graph Laplacian. Additionally, we concatenate the corresponding k lowest eigenvalues to each node.

Graph size encoding The graph size is encoded as the ratio between the size of the current graph and that of the largest graph in the dataset, n/n_{max}. This value is concatenated to all nodes in the graph.

Molecular features For ZINC250K, we use the charge and valency of each atom as additional features.

Cycles Following Vignac et al. [2023], we count the number of cycles of size 3, 4, and 5 that each node is part of, and use these counts as features.

Relative random-walk probabilities RRWPs are the probabilities of reaching a target node from a source node in a given number of steps. We compute them for all numbers of steps from 1 to k.

E.5 Evaluation

E.5.1 Molecule Generation

For ZINC250K, we adopt the evaluation procedure of Jo et al. [2024] (from which we also take baseline results), originally established by Jo et al. [2022]. We assess performance using three standard metrics: (i) the Fréchet ChemNet Distance (FCD) [Preuer et al., 2018], which measures similarity in chemical feature space; (ii) the Neighborhood Subgraph Pairwise Distance Kernel (NSPDK) [Costa and Grave, 2010], which evaluates graph-structural similarity; and (iii) validity, the fraction of chemically valid molecules without any post hoc correction or resampling. We report means over five sampling runs, each generating 10,000 molecules. For completeness, we also provide standard deviations, as well as uniqueness and novelty statistics, in Appendix F.

QM9H For QM9H, we generally follow the above procedure. For evaluation, we rely on SMILES with explicit hydrogen atoms to compute validity, uniqueness, and FCD. Under this evaluation procedure, a small fraction of the molecules in the dataset are themselves invalid. We use the kekulized version of the dataset (i.e., with three bond types: single, double, and triple). Some concurrent implementations use a variant of the dataset with explicit aromatic bonds; in such cases, evaluation metrics, in particular validity and FCD, are not directly comparable to ours.

E.5.2 Unattributed Synthetic Graph Generation

For unattributed graphs, we follow the procedure of Martinkus et al. [2022]: an 80/20 train–test split, with 20% of the training set used for validation. We measure distributional similarity via MMD over degree distributions, clustering coefficients, orbit counts, and spectral densities, and we additionally report validity. For Planar, a valid graph must be planar and connected. For SBM, validity indicates consistency with the data-generating stochastic block model (intra-community edge density 0.3, inter-community edge density 0.005). We omit uniqueness because all models reach 100% on both datasets, and we report novelty only for SBM, since it is 100% for all models on Planar. Uniqueness is the fraction of distinct graphs among the generated samples; novelty is the fraction of unique graphs not present in the training set. We report means over five runs, each generating 40 graphs.

E.5.3 Real-World Graph Generation

Enzymes is a dataset of 587 protein graphs representing the tertiary structures of enzymes from the BRENDA database. The graphs have between 10 and 125 nodes. For consistency, we follow Jo et al. [2022] in using the Earth Mover's Distance (EMD) to compute the MMDs. We use 20% of the data as the test set and the remaining 80% as the training set, from which we additionally reserve 20% for validation. MMDs are computed between the test set and a batch of generated graphs of the same size.

E.6 Non-Markovian Continuous Diffusion

To further assess the role of the simplex parameterization itself, we implemented an additional baseline that uses the same non-Markovian resampling rule, but with a parameterization in unconstrained Euclidean space rather than on the simplex. Concretely, we define the probability path as

$$\mathbf{x}_t = \alpha_t(2\mathbf{x}_1 - \mathbf{1}) + (1 - \alpha_t)\mathbf{x}_0, \tag{88}$$

with $\mathbf{x}_1 \sim p_{\text{data}}$ and $\mathbf{x}_0 \sim \mathcal{N}(\mathbf{0}, \boldsymbol{I})$. We use a linear schedule, $\alpha_t = t$, which we found more effective. As in UNSID, we learn $P^{\theta}_{1|t}(\mathbf{x}_1 \mid \mathbf{x}_t)$ and denoise with Equation 4, where $q_{t+h}(\cdot \mid \mathbf{x}_1)$ is now the Gaussian path above.

E.7 Baselines

We report results for five representative methods: DiGress [Vignac et al., 2023] implements standard discrete diffusion; GruM [Jo et al., 2024] performs denoising in a continuous space; GraBFN [Song et al., 2025] is a Bayesian-flow-network-based generative model; SID [Boget, 2025] is a non-Markovian discrete diffusion method; and DeFoG [Qin et al., 2025] is a discrete flow-matching model. In general, we report the numbers from the original papers, except for the `ZINC250K` results of DiGress, for which we use the values provided by Jo et al. [2024], as DiGress does not report results on this dataset. Finally, we also report metrics computed on random samples of the training set, each containing the same number of graphs as the generated samples (Train.).

For the `Planar` and `SBM` datasets, we reran the experiments for baselines that reported results from a *single* sampling run. Below, we motivate this choice and describe our reproduction procedure.

E.7.1 Motivation for Multiple Sampling Runs

On unattributed graph datasets such as `Planar` and `SBM`, the test sets are relatively small, and sampling-based metrics exhibit high variance across runs. Moreover, evaluating many models, checkpoints, or seeds increases the risk of overfitting to the test set. Multiple sampling runs are therefore strongly recommended, and the corresponding variability should be assessed via standard deviations.

Several baselines report results from a *single* sampling run, including GruM, DiGress, and GraBFN. These reported results even outperform samples drawn directly from the training set, which suggests overfitting to the test set. For this reason, we reran the experiments for GruM and DiGress. Unfortunately, the official GraBFN repository is currently empty, which prevented us from reproducing this baseline; we therefore do not report it on `Planar` and `SBM`.

E.7.2 Procedure for Result Reproduction

We were unable to reproduce the published results of either GruM or DiGress. Below, we describe the exact procedure we followed.

GruM. We used the pretrained models available in the official repository: `https://github.com/harryjo97/GruM/tree/master/GruM_2D`. We performed five sampling runs, varying only the sampling seed, and computed metrics using the code provided by the authors. We note that the default seed yields substantially better performance than the others.

DiGress. Since no pretrained models are currently available, we retrained DiGress using the official repository (`https://github.com/cvignac/DiGress/`) and the configurations specified in the corresponding YAML files. As specified in these configurations, we evaluated the model every 400 epochs and computed metrics on the validation set. For stability, we increased the number of graphs sampled at each evaluation during training to 40. We trained for 100,000 epochs on `Planar` and 20,000 epochs on `SBM`, and selected the checkpoint with the highest validation validity for final evaluation. Note that we fixed a minor bug in `src/datasets/spectre_dataset.py`, where preprocessed graphs were added twice to their respective sets (lines 93 and 100).

F Additional Results

In this section, we provide the following additional results:

- model sizes and sampling times (Appendix F.1);
- results on QM9H (Appendix F.2);
- ablations on the NFE, the noise schedule, and the prior (Appendix F.3);
- guidance (Appendix F.4);
- detailed results with standard deviations on Planar, SBM, ZINC250K, and Enzymes (Appendix F.5).

F.1 Model Sizes and Sampling Times

We report the number of parameters of each model, as well as the wall-clock time required for 100 sampling iterations when generating 40 Planar graphs and 40 SBM graphs. All experiments were conducted on a single server with one GPU (NVIDIA GeForce RTX 3090) and 128 CPU cores.

Our model is the fastest on both datasets. Since Dirichlet flow matching (DIRIFM) and UNSID share the same denoiser architecture, differences in sampling speed arise solely from the sampling procedure itself. Likewise, our implementation of SID (NM-DD) uses the same architecture as UNSID; the additional overhead of NM-DD is primarily due to the computation of the extra structural features required by the discrete denoiser.

Table 5: Model sizes and sampling times (in seconds).

	PLANAR		**SBM**		**QM9H**	**ZINC250K**
	TIME↓	PARAMS	TIME↓	PARAMS	PARAMS	PARAMS
DIGRESS	16.9 ± 0.0	8.9M	96.9 ± 0.1	7.1M	3.6M	8.2M
GRUM	29.3 ± 0.0	7.1M	84.2 ± 0.0	7.1M	-	8.2M
NM-DD	9.9 ± 0.4	5.4M	84.1 ± 0.3	6.2M	6.2M	6.2M
DIRIFM	9.4 ± 0.1	5.4M	89.0 ± 0.3	6.2M	6.2M	6.2M
UNSID	6.3 ± 0.0	5.4M	72.6 ± 0.5	6.2M	6.2M	6.2M

For DIGRESS on ZINC250K, whose results we take from Jo et al. [2024], we assume that Jo et al. [2024] used the same architecture as for GRUM.

F.2 Results on QM9H

UNSID outperforms all baselines on all metrics except uniqueness, where it remains competitive. For DIGRESS, we report the results from the original paper, which does not report FCD and NSPDK.

Table 6: Results on QM9H. NSPDK is reported $\times 10^3$.

	VALID (%)↑	FCD↓	NSPDK↓	UNIQUE (%)↑
TRAIN.	98.90 ± 0.05	0.062 ± 0.002	0.121 ± 0.016	99.81 ± 0.04
DIGRESS	95.4 ± 1.1	-	-	97.6 ± 0.4
DISCDIFF	22.29 ± 0.62	4.246 ± 0.454	41.932 ± 0.741	73.72 ± 0.56
NM-DD	97.97 ± 0.07	0.366 ± 0.021	1.149 ± 0.047	95.23 ± 0.17
NM-CD	31.29 ± 0.15	1.688 ± 0.031	22.675 ± 0.099	91.45 ± 0.27
DIRIFM	92.20 ± 0.19	0.356 ± 0.029	0.495 ± 0.010	97.53 ± 0.13
UNSID	98.87 ± 0.07	0.152 ± 0.011	0.487 ± 0.068	96.33 ± 0.14

F.3 Ablations

We conduct all ablation studies on ZINC250K, as it is by far the largest dataset and has the largest test set, which yields more precise and robust evaluation metrics. This dataset is therefore best suited to detecting small effects.

F.3.1 NFE

When ablating the NFE, we observe that our model achieves state-of-the-art validity with as few as 64 denoising steps. FCD and NSPDK keep improving with more steps: our model surpasses the best baseline FCD from 512 steps on, and the best baseline NSPDK at 1024 steps. This confirms the efficiency and overall effectiveness of our approach. Mismatch with the main table comes from the single run evaluation used for this ablation.

Table 7: Effect of the NFE on ZINC250K. NSPDK is reported $\times 10^3$.

NFE	INVALID (%)↓	FCD↓	NSPDK↓
16	5.71	6.54	7.97
32	1.05	4.32	4.00
64	0.36	3.26	2.79
128	0.08	2.63	2.08
256	0.09	2.17	1.59
512	0.01	1.92	1.34
1024	0.02	1.71	0.87

F.3.2 Noise Schedule

We assess the effect of the noise schedule $\alpha_t = -a \log(1-t)$, which parameterizes our probability path $\mathrm{Dir}(\mathbf{x}_t; \mathbf{1} + \alpha_t \mathbf{x}_1)$, by evaluating our model across a range of values of the hyperparameter a.

Figure 4 shows the corresponding Voronoi probabilities for each tested value of a. Since the Voronoi probability depends on the number of categories K, we report it separately for node attributes ($K = 9$) and edge attributes ($K = 4$).

Figure 4 shows that low values of a (in particular $a = 1$) induce a probability path that becomes too sharp near the end of the trajectory (i.e., as $t \to 1$), which makes the denoising task overly difficult in the final steps. Conversely, for large values of a, the probability path becomes too flat toward the end of the process, leaving the denoiser with little useful signal to correct.

The results in Table 8 align with these expectations. When $a = 1$, performance degrades substantially. For large values of a, performance also decreases, but only marginally. Overall, the model is robust to variations in a: all tested values $a \geq 2$ outperform the baselines in FCD and NSPDK.

We draw two main conclusions from this experiment: (1) the Voronoi-probability analysis provides a reliable tool for calibrating the noise schedule, and (2) our model is robust to a wide range of schedules.

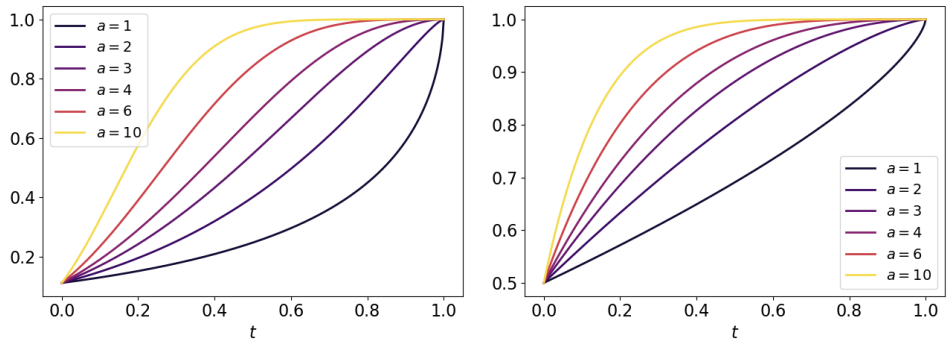


Figure 4: Voronoi probabilities for various values of a in the schedule $\alpha_t = -a\log(1-t)$, for $K = 9$ (left) and $K = 2$ (right).

Table 8: Results on ZINC250K for various values of the schedule hyperparameter a. NSPDK is reported $\times 10^3$.

a	VALID (%)↑	FCD↓	NSPDK↓	UNIQUE (%)↑	NOVEL (%)↑
1	98.87 ± 0.16	4.13 ± 0.08	2.61 ± 0.11	99.86 ± 0.01	99.99 ± 0.01
2	99.89 ± 0.03	1.89 ± 0.01	0.98 ± 0.06	99.98 ± 0.00	99.96 ± 0.01
3	99.98 ± 0.01	1.79 ± 0.03	1.00 ± 0.06	100.00 ± 0.00	99.97 ± 0.01
4	99.94 ± 0.02	1.82 ± 0.05	1.05 ± 0.04	99.99 ± 0.00	99.98 ± 0.01
6	99.92 ± 0.02	1.66 ± 0.03	0.82 ± 0.03	99.99 ± 0.01	99.97 ± 0.02
10	99.91 ± 0.00	1.95 ± 0.03	1.25 ± 0.07	99.99 ± 0.01	99.99 ± 0.00

F.3.3 Prior Choice

Compared with the uniform prior, the marginal-weighted mixture of Dirichlet distributions slightly but consistently improves sampling performance across all metrics (Figure 5).

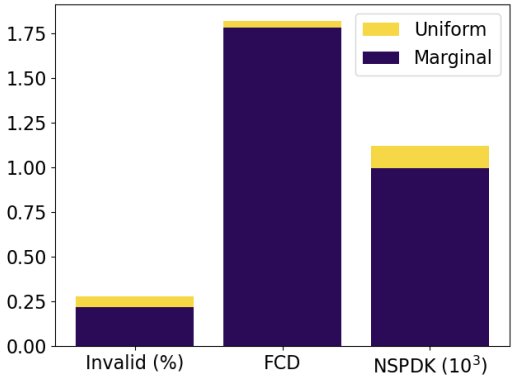


Figure 5: Comparison between the uniform prior $\mathrm{Dir}(\mathbf{1})$ and the marginal-weighted mixture of Dirichlet distributions q_0^{marg} ($\kappa = 2$) on ZINC250K.

F.4 Guidance

The sole purpose of this experiment is to show that our model supports classifier guidance without affecting its ability to generate valid molecules. As described in Section 5, we target standardized QED values on QM9H and compute the MAE in standardized units.

Table 9: Effect of classifier guidance on QM9H. The MAE between target and generated QED values is computed in standardized units.

	VALID (%)↑	MAE↓
UNCONDITIONAL	99.6	1.06
CLASSIFIER GUIDANCE	99.5	0.42

F.5 Detailed Results

We report means and standard deviations over five sampling runs on Planar (Table 10), SBM (Table 11), ZINC250K (Table 12), and Enzymes (Table 13).

Table 10: Results on Planar. MMDs are reported $\times 10^3$.

	VALID (%)↑	DEG.↓	CLUST.↓	ORBIT↓	SPECT.↓
TRAIN.	100.0 ± 0.0	0.25 ± 0.18	36.6 ± 3.5	0.78 ± 0.37	6.62 ± 1.34
DIGRESS	45.5 ± 7.0	0.71 ± 0.40	45.4 ± 13.6	1.31 ± 0.78	8.40 ± 1.16
GRUM	91.0 ± 2.6	0.38 ± 0.15	40.7 ± 4.8	6.42 ± 1.96	7.87 ± 1.17
SID	91.3 ± 4.1	5.93 ± 1.26	163.4 ± 31.9	19.1 ± 4.14	7.62 ± 1.34
DEFOG	99.5 ± 1.0	0.5 ± 0.2	50.1 ± 14.9	0.6 ± 0.4	7.2 ± 1.1
DISCDIFF	0.0 ± 0.0	56.1 ± 3.70	294.0 ± 3.4	1410.0 ± 44.5	85.1 ± 1.6
NM-DD	98.0 ± 1.0	14.1 ± 2.58	363.0 ± 45.6	27.2 ± 5.6	6.9 ± 0.8
NM-CD	0.0 ± 0.0	1.23 ± 0.50	226.3 ± 8.4	45.77 ± 6.20	21.96 ± 0.74
DIRIFM	0.0 ± 0.0	6.44 ± 1.42	196.6 ± 22.5	45.04 ± 5.83	21.76 ± 1.30
UNSID	100.0 ± 0.0	0.36 ± 0.24	39.9 ± 8.7	0.78 ± 0.49	7.12 ± 0.45

Table 11: Results on SBM. MMDs are reported $\times 10^3$.

	VALID (%)↑	DEG.↓	CLUST.↓	ORBIT↓	SPECT.↓	NOVEL (%)↑
TRAIN.	93.50 ± 2.00	1.57 ± 0.55	50.11 ± 0.51	37.0 ± 10.7	4.58 ± 0.49	100
DIGRESS	51.00 ± 9.62	1.28 ± 0.48	51.49 ± 1.31	39.6 ± 7.7	5.04 ± 0.63	100.0 ± 0.0
GRUM	67.54 ± 2.98	2.20 ± 0.76	49.88 ± 0.62	40.4 ± 5.6	5.06 ± 0.72	100.0 ± 0.0
SID	63.5 ± 3.7	11.5 ± 2.7	51.4 ± 1.5	123.1 ± 5.4	5.93 ± 1.18	100.0 ± 0.0
DEFOG	90.0 ± 5.2	0.6 ± 2.3	51.7 ± 1.2	55.6 ± 73.9	5.40 ± 1.20	90.0 ± 5.1
DISCDIFF	0.00 ± 0.00	1.82 ± 0.88	86.38 ± 6.37	125.7 ± 3.3	11.28 ± 1.06	100.0 ± 0.0
NM-DD	60.50 ± 4.30	4.38 ± 1.65	50.92 ± 0.89	52.9 ± 6.3	5.70 ± 0.30	100.0 ± 0.0
NM-CD	57.50 ± 5.40	1.89 ± 0.50	50.29 ± 0.45	41.08 ± 1.00	5.38 ± 1.30	100.0 ± 0.0
DIRIFM	46.00 ± 4.36	4.26 ± 0.54	53.04 ± 1.05	53.0 ± 8.9	5.02 ± 1.14	100.0 ± 0.0
UNSID	78.50 ± 4.64	1.74 ± 0.60	49.94 ± 1.07	52.1 ± 1.1	5.90 ± 1.05	100.0 ± 0.0

Table 12: Results on ZINC250K. NSPDK is reported $\times 10^3$.

	VALID (%)↑	FCD↓	NSPDK↓	UNIQUE (%)↑	NOVEL (%)↑
TRAIN.	100.00 ± 0.00	1.13 ± 0.01	0.10 ± 0.00	99.97 ± 0.02	-
DIGRESS	94.99	3.48	2.1	-	-
GRUM	98.65	2.26	1.5	-	-
GRABFN	99.22	2.12	1.3	-	-
SID	99.50 ± 0.06	2.06 ± 0.05	2.01 ± 0.01	99.84 ± 0.02	99.97 ± 0.00
DISCDIFF	74.17 ± 0.65	4.78 ± 0.14	4.08 ± 0.13	100.00 ± 0.00	100.00 ± 0.00
NM-DD	99.92 ± 0.01	2.65 ± 0.06	3.48 ± 0.06	99.88 ± 0.04	99.95 ± 0.03
NM-CD	57.96 ± 0.16	9.51 ± 0.04	13.65 ± 0.07	99.92 ± 0.02	99.99 ± 0.00
DIRIFM	97.32 ± 0.12	2.79 ± 0.01	1.92 ± 0.07	99.99 ± 0.01	99.99 ± 0.01
UNSID	99.98 ± 0.01	1.79 ± 0.03	1.00 ± 0.06	100.00 ± 0.00	99.97 ± 0.01

Table 13: Results on `Enzymes`. MMDs are reported $\times 10^3$.

	DEG.↓	CLUST.↓	ORBIT↓	SPECT.↓
TRAIN.	3.3 ± 0.3	19.2 ± 4.2	0.9 ± 0.3	18.9 ± 0.1
GRAPHVAE	1369 ± 20	629 ± 5	191 ± 20	-
GRAPHRNN	17 ± 7	62 ± 20	46 ± 31	-
GDSS	26 ± 8	61 ± 10	9 ± 5	-
DISCDIFF	212.7 ± 11.8	631.1 ± 45.7	515.5 ± 54.1	19.6 ± 0.2
NM-DD	13.6 ± 4.1	39.9 ± 5.2	10.4 ± 0.9	17.4 ± 0.0
NM-CD	84.5 ± 14.8	307.9 ± 29.9	125.4 ± 16.0	20.2 ± 0.2
DIRIFM	7.4 ± 2.4	100.6 ± 17.5	6.5 ± 2.7	19.8 ± 0.2
UNSID	10.2 ± 4.2	52.0 ± 15.9	2.7 ± 1.1	19.5 ± 0.0

G Visualizations

We present here some visualizations of the generated and reference graphs.

Figure 6: `Qm9H`

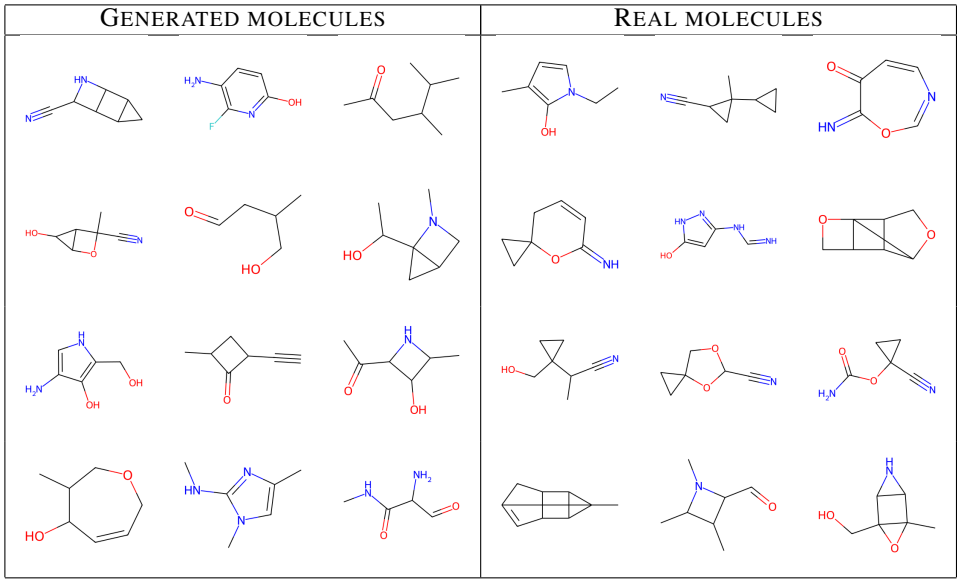


Figure 7: ZINC250K

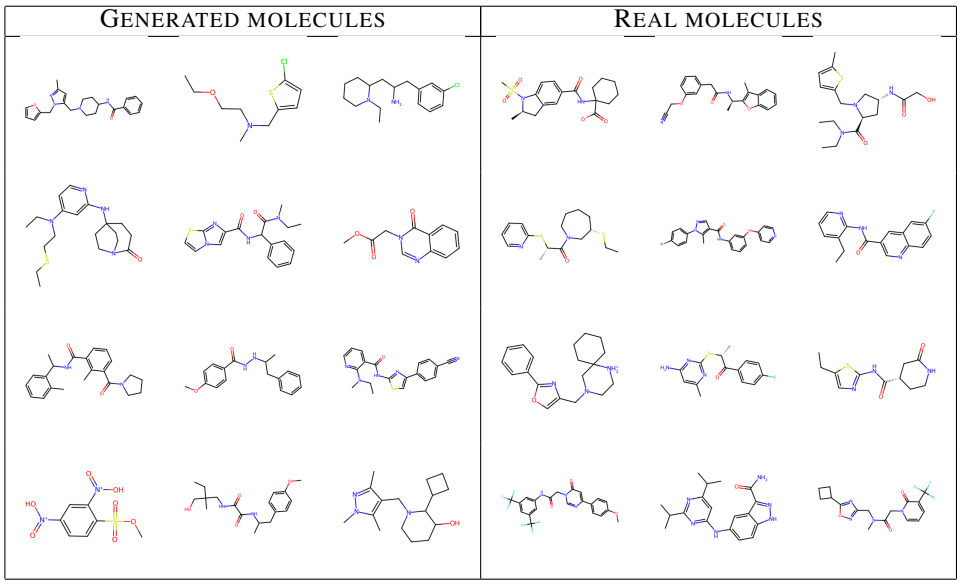
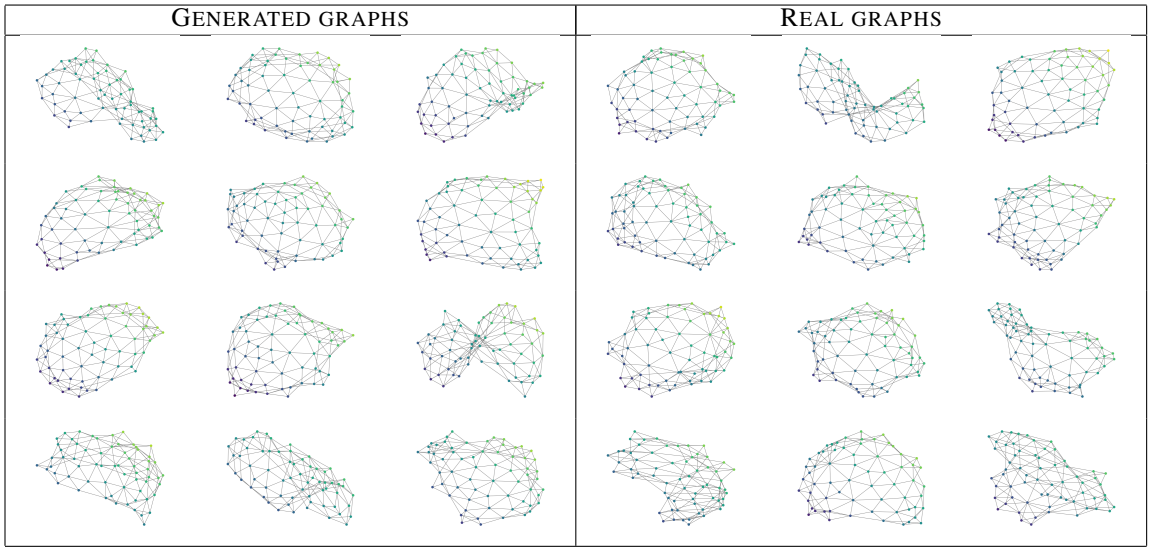


Figure 8: Planar



NeurIPS Paper Checklist

1. **Claims**

   Question: Do the main claims made in the abstract and introduction accurately reflect the paper's contributions and scope?

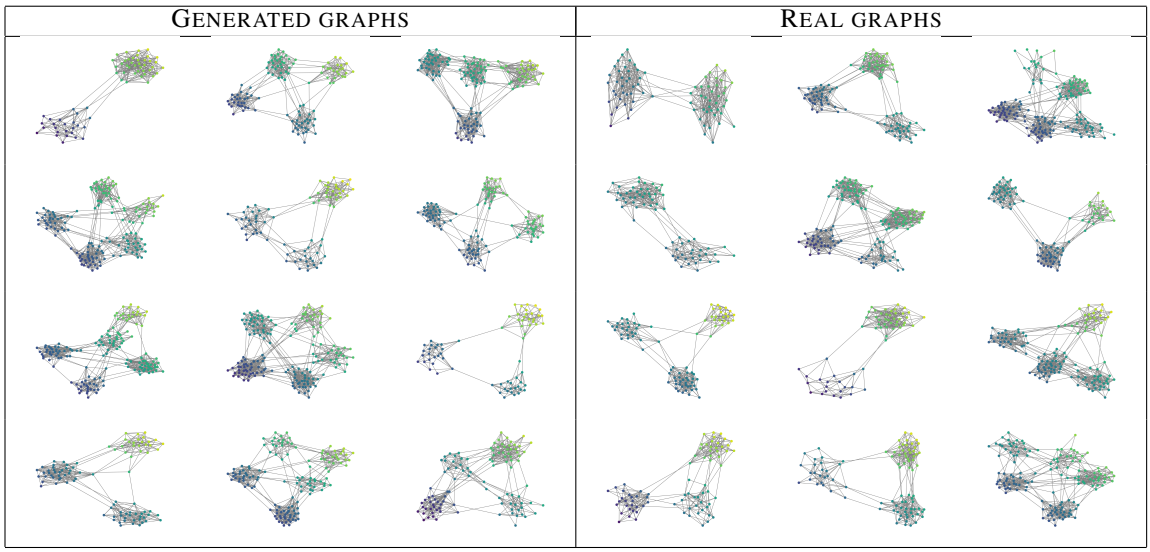
   Answer: [Yes]

   Justification: Specifically we states that we evaluate our method only for graph generation. Furhter research and evaluations are needed to extend the proposed model to other data modalities.

   Guidelines:

   - The answer [N/A] means that the abstract and introduction do not include the claims made in the paper.

Figure 9: `Stochstic Block Model`



- The abstract and/or introduction should clearly state the claims made, including the contributions made in the paper and important assumptions and limitations. A [No] or [N/A] answer to this question will not be perceived well by the reviewers.
- The claims made should match theoretical and experimental results, and reflect how much the results can be expected to generalize to other settings.
- It is fine to include aspirational goals as motivation as long as it is clear that these goals are not attained by the paper.

2. **Limitations**

   Question: Does the paper discuss the limitations of the work performed by the authors?

   Answer:[Yes]

   Justification: See limitations in conclusion.

   Guidelines:

   - The answer [N/A] means that the paper has no limitation while the answer [No] means that the paper has limitations, but those are not discussed in the paper.
   - The authors are encouraged to create a separate "Limitations" section in their paper.
   - The paper should point out any strong assumptions and how robust the results are to violations of these assumptions (e.g., independence assumptions, noiseless settings, model well-specification, asymptotic approximations only holding locally). The authors should reflect on how these assumptions might be violated in practice and what the implications would be.
   - The authors should reflect on the scope of the claims made, e.g., if the approach was only tested on a few datasets or with a few runs. In general, empirical results often depend on implicit assumptions, which should be articulated.
   - The authors should reflect on the factors that influence the performance of the approach. For example, a facial recognition algorithm may perform poorly when image resolution is low or images are taken in low lighting. Or a speech-to-text system might not be used reliably to provide closed captions for online lectures because it fails to handle technical jargon.
   - The authors should discuss the computational efficiency of the proposed algorithms and how they scale with dataset size.
   - If applicable, the authors should discuss possible limitations of their approach to address problems of privacy and fairness.
   - While the authors might fear that complete honesty about limitations might be used by reviewers as grounds for rejection, a worse outcome might be that reviewers discover

limitations that aren't acknowledged in the paper. The authors should use their best judgment and recognize that individual actions in favor of transparency play an important role in developing norms that preserve the integrity of the community. Reviewers will be specifically instructed to not penalize honesty concerning limitations.

3. **Theory assumptions and proofs**

   Question: For each theoretical result, does the paper provide the full set of assumptions and a complete (and correct) proof?

   Answer: [Yes]

   Justification: See Appendix A

   Guidelines:

   - The answer [N/A] means that the paper does not include theoretical results.
   - All the theorems, formulas, and proofs in the paper should be numbered and cross-referenced.
   - All assumptions should be clearly stated or referenced in the statement of any theorems.
   - The proofs can either appear in the main paper or the supplemental material, but if they appear in the supplemental material, the authors are encouraged to provide a short proof sketch to provide intuition.
   - Inversely, any informal proof provided in the core of the paper should be complemented by formal proofs provided in appendix or supplemental material.
   - Theorems and Lemmas that the proof relies upon should be properly referenced.

4. **Experimental result reproducibility**

   Question: Does the paper fully disclose all the information needed to reproduce the main experimental results of the paper to the extent that it affects the main claims and/or conclusions of the paper (regardless of whether the code and data are provided or not)?

   Answer: [Yes]

   Justification: We provide the code, the hyperparameters and additional technical details in Appendix E

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - If the paper includes experiments, a [No] answer to this question will not be perceived well by the reviewers: Making the paper reproducible is important, regardless of whether the code and data are provided or not.
   - If the contribution is a dataset and/or model, the authors should describe the steps taken to make their results reproducible or verifiable.
   - Depending on the contribution, reproducibility can be accomplished in various ways. For example, if the contribution is a novel architecture, describing the architecture fully might suffice, or if the contribution is a specific model and empirical evaluation, it may be necessary to either make it possible for others to replicate the model with the same dataset, or provide access to the model. In general. releasing code and data is often one good way to accomplish this, but reproducibility can also be provided via detailed instructions for how to replicate the results, access to a hosted model (e.g., in the case of a large language model), releasing of a model checkpoint, or other means that are appropriate to the research performed.
   - While NeurIPS does not require releasing code, the conference does require all submissions to provide some reasonable avenue for reproducibility, which may depend on the nature of the contribution. For example
     (a) If the contribution is primarily a new algorithm, the paper should make it clear how to reproduce that algorithm.
     (b) If the contribution is primarily a new model architecture, the paper should describe the architecture clearly and fully.
     (c) If the contribution is a new model (e.g., a large language model), then there should either be a way to access this model for reproducing the results or a way to reproduce the model (e.g., with an open-source dataset or instructions for how to construct the dataset).

(d) We recognize that reproducibility may be tricky in some cases, in which case authors are welcome to describe the particular way they provide for reproducibility. In the case of closed-source models, it may be that access to the model is limited in some way (e.g., to registered users), but it should be possible for other researchers to have some path to reproducing or verifying the results.

5. **Open access to data and code**

   Question: Does the paper provide open access to the data and code, with sufficient instructions to faithfully reproduce the main experimental results, as described in supplemental material?

   Answer: [Yes]

   Justification: All datasets are public and the code with instruction is released.

   Guidelines:

   - The answer [N/A] means that paper does not include experiments requiring code.
   - Please see the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - While we encourage the release of code and data, we understand that this might not be possible, so [No] is an acceptable answer. Papers cannot be rejected simply for not including code, unless this is central to the contribution (e.g., for a new open-source benchmark).
   - The instructions should contain the exact command and environment needed to run to reproduce the results. See the NeurIPS code and data submission guidelines (`https://neurips.cc/public/guides/CodeSubmissionPolicy`) for more details.
   - The authors should provide instructions on data access and preparation, including how to access the raw data, preprocessed data, intermediate data, and generated data, etc.
   - The authors should provide scripts to reproduce all experimental results for the new proposed method and baselines. If only a subset of experiments are reproducible, they should state which ones are omitted from the script and why.
   - At submission time, to preserve anonymity, the authors should release anonymized versions (if applicable).
   - Providing as much information as possible in supplemental material (appended to the paper) is recommended, but including URLs to data and code is permitted.

6. **Experimental setting/details**

   Question: Does the paper specify all the training and test details (e.g., data splits, hyperparameters, how they were chosen, type of optimizer) necessary to understand the results?

   Answer: [Yes],

   Justification: Please see the appendix and more specifically Appendix E

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The experimental setting should be presented in the core of the paper to a level of detail that is necessary to appreciate the results and make sense of them.
   - The full details can be provided either with the code, in appendix, or as supplemental material.

7. **Experiment statistical significance**

   Question: Does the paper report error bars suitably and correctly defined or other appropriate information about the statistical significance of the experiments?

   Answer: [Yes]

   Justification: We provide standard deviation over multiple run for all our experiments following standard practice in the field.

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.

- The authors should answer [Yes] if the results are accompanied by error bars, confidence intervals, or statistical significance tests, at least for the experiments that support the main claims of the paper.
- The factors of variability that the error bars are capturing should be clearly stated (for example, train/test split, initialization, random drawing of some parameter, or overall run with given experimental conditions).
- The method for calculating the error bars should be explained (closed form formula, call to a library function, bootstrap, etc.)
- The assumptions made should be given (e.g., Normally distributed errors).
- It should be clear whether the error bar is the standard deviation or the standard error of the mean.
- It is OK to report 1-sigma error bars, but one should state it. The authors should preferably report a 2-sigma error bar than state that they have a 96% CI, if the hypothesis of Normality of errors is not verified.
- For asymmetric distributions, the authors should be careful not to show in tables or figures symmetric error bars that would yield results that are out of range (e.g., negative error rates).
- If error bars are reported in tables or plots, the authors should explain in the text how they were calculated and reference the corresponding figures or tables in the text.

8. **Experiments compute resources**

   Question: For each experiment, does the paper provide sufficient information on the computer resources (type of compute workers, memory, time of execution) needed to reproduce the experiments?

   Answer: [Yes]

   Justification: Please see Appendix E

   Guidelines:

   - The answer [N/A] means that the paper does not include experiments.
   - The paper should indicate the type of compute workers CPU or GPU, internal cluster, or cloud provider, including relevant memory and storage.
   - The paper should provide the amount of compute required for each of the individual experimental runs as well as estimate the total compute.
   - The paper should disclose whether the full research project required more compute than the experiments reported in the paper (e.g., preliminary or failed experiments that didn't make it into the paper).

9. **Code of ethics**

   Question: Does the research conducted in the paper conform, in every respect, with the NeurIPS Code of Ethics `https://neurips.cc/public/EthicsGuidelines`?

   Answer: [Yes]

   Justification: We do.

   Guidelines:

   - The answer [N/A] means that the authors have not reviewed the NeurIPS Code of Ethics.
   - If the authors answer [No], they should explain the special circumstances that require a deviation from the Code of Ethics.
   - The authors should make sure to preserve anonymity (e.g., if there is a special consideration due to laws or regulations in their jurisdiction).

10. **Broader impacts**

    Question: Does the paper discuss both potential positive societal impacts and negative societal impacts of the work performed?

    Answer: [N/A].

Justification: This paper presents work whose goal is to advance the field of Machine Learning. There are many potential societal consequences of our work, none which we feel must be specifically highlighted here.

Guidelines:

- The answer [N/A] means that there is no societal impact of the work performed.
- If the authors answer [N/A] or [No], they should explain why their work has no societal impact or why the paper does not address societal impact.
- Examples of negative societal impacts include potential malicious or unintended uses (e.g., disinformation, generating fake profiles, surveillance), fairness considerations (e.g., deployment of technologies that could make decisions that unfairly impact specific groups), privacy considerations, and security considerations.
- The conference expects that many papers will be foundational research and not tied to particular applications, let alone deployments. However, if there is a direct path to any negative applications, the authors should point it out. For example, it is legitimate to point out that an improvement in the quality of generative models could be used to generate Deepfakes for disinformation. On the other hand, it is not needed to point out that a generic algorithm for optimizing neural networks could enable people to train models that generate Deepfakes faster.
- The authors should consider possible harms that could arise when the technology is being used as intended and functioning correctly, harms that could arise when the technology is being used as intended but gives incorrect results, and harms following from (intentional or unintentional) misuse of the technology.
- If there are negative societal impacts, the authors could also discuss possible mitigation strategies (e.g., gated release of models, providing defenses in addition to attacks, mechanisms for monitoring misuse, mechanisms to monitor how a system learns from feedback over time, improving the efficiency and accessibility of ML).

11. **Safeguards**

    Question: Does the paper describe safeguards that have been put in place for responsible release of data or models that have a high risk for misuse (e.g., pre-trained language models, image generators, or scraped datasets)?

    Answer: [N/A]

    Justification: No risk

    Guidelines:

    - The answer [N/A] means that the paper poses no such risks.
    - Released models that have a high risk for misuse or dual-use should be released with necessary safeguards to allow for controlled use of the model, for example by requiring that users adhere to usage guidelines or restrictions to access the model or implementing safety filters.
    - Datasets that have been scraped from the Internet could pose safety risks. The authors should describe how they avoided releasing unsafe images.
    - We recognize that providing effective safeguards is challenging, and many papers do not require this, but we encourage authors to take this into account and make a best faith effort.

12. **Licenses for existing assets**

    Question: Are the creators or original owners of assets (e.g., code, data, models), used in the paper, properly credited and are the license and terms of use explicitly mentioned and properly respected?

    Answer: [N/A]v

    Justification: No asset

    Guidelines:

    - The answer [N/A] means that the paper does not use existing assets.
    - The authors should cite the original paper that produced the code package or dataset.

- The authors should state which version of the asset is used and, if possible, include a URL.
- The name of the license (e.g., CC-BY 4.0) should be included for each asset.
- For scraped data from a particular source (e.g., website), the copyright and terms of service of that source should be provided.
- If assets are released, the license, copyright information, and terms of use in the package should be provided. For popular datasets, `paperswithcode.com/datasets` has curated licenses for some datasets. Their licensing guide can help determine the license of a dataset.
- For existing datasets that are re-packaged, both the original license and the license of the derived asset (if it has changed) should be provided.
- If this information is not available online, the authors are encouraged to reach out to the asset's creators.

13. **New assets**

    Question: Are new assets introduced in the paper well documented and is the documentation provided alongside the assets?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not release new assets.
    - Researchers should communicate the details of the dataset/code/model as part of their submissions via structured templates. This includes details about training, license, limitations, etc.
    - The paper should discuss whether and how consent was obtained from people whose asset is used.
    - At submission time, remember to anonymize your assets (if applicable). You can either create an anonymized URL or include an anonymized zip file.

14. **Crowdsourcing and research with human subjects**

    Question: For crowdsourcing experiments and research with human subjects, does the paper include the full text of instructions given to participants and screenshots, if applicable, as well as details about compensation (if any)?

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
    - Including this information in the supplemental material is fine, but if the main contribution of the paper involves human subjects, then as much detail as possible should be included in the main paper.
    - According to the NeurIPS Code of Ethics, workers involved in data collection, curation, or other labor should be paid at least the minimum wage in the country of the data collector.

15. **Institutional review board (IRB) approvals or equivalent for research with human subjects**

    Question: Does the paper describe potential risks incurred by study participants, whether such risks were disclosed to the subjects, and whether Institutional Review Board (IRB) approvals (or an equivalent approval/review based on the requirements of your country or institution) were obtained?

    Answer: [N/A]

    Justification:

    Guidelines:

- The answer [N/A] means that the paper does not involve crowdsourcing nor research with human subjects.
- Depending on the country in which research is conducted, IRB approval (or equivalent) may be required for any human subjects research. If you obtained IRB approval, you should clearly state this in the paper.
- We recognize that the procedures for this may vary significantly between institutions and locations, and we expect authors to adhere to the NeurIPS Code of Ethics and the guidelines for their institution.
- For initial submissions, do not include any information that would break anonymity (if applicable), such as the institution conducting the review.

16. **Declaration of LLM usage**

    Question: Does the paper describe the usage of LLMs if it is an important, original, or non-standard component of the core methods in this research? Note that if the LLM is used only for writing, editing, or formatting purposes and does *not* impact the core methodology, scientific rigor, or originality of the research, declaration is not required.

    Answer: [N/A]

    Justification:

    Guidelines:

    - The answer [N/A] means that the core method development in this research does not involve LLMs as any important, original, or non-standard components.
    - Please refer to our LLM policy in the NeurIPS handbook for what should or should not be described.