Equivariant Neural Diffusion for Molecule Generation

François Cornet [1] **Grigory Bartosh** [2] **Mikkel N. Schmidt** [1] **Christian A. Naesseth** [2]

Abstract

We introduce Equivariant Neural Diffusion (END), a novel diffusion model for molecule generation in 3D that is equivariant to Euclidean transformations. Compared to current state-of-the-art equivariant diffusion models, the key innovation in END lies in its learnable forward process for enhanced generative modelling. Rather than pre-specified, the forward process is parameterized through a time- and data-dependent transformation that is equivariant to rigid transformations. Through a series of experiments on standard molecule generation benchmarks, we demonstrate that END improves on several strong baselines for both unconditional and conditional generation.

1. Introduction

The discovery of novel chemical compounds with relevant properties is critical to a number of scientific fields, such as drug discovery and materials design (Merchant et al., 2023). However, due to the large size and complex structure of the chemical space (Ruddigkeit et al., 2012), which combines continuous and discrete features, it is notably difficult to search. Additionally, *ab-initio* quantum mechanics methods for computing target properties are often computationally expensive, preventing brute-force enumeration. While some of these heavy computations can be amortized through learned surrogates, the need for innovative search methods remains, and generative models have recently emerged as a promising avenue (Anstine & Isayev, 2023). Generative models can learn complex data distributions, that, in turn, can be sampled from to obtain novel samples similar to the original data. Compared to other data modalities such as images or text, molecules have to adhere to strict chemical rules, and obey the symmetries of the 3D space.

[1]Technical University of Denmark [2]University of Amsterdam. Correspondence to: François Cornet <frjc@dtu.dk>, Grigory Bartosh <g.bartosh@uva.nl>, Mikkel N. Schmidt <mnsc@dtu.dk>, Christian A. Naesseth <c.a.naesseth@uva.nl>.

Accepted at the AI4Science Workshop of the 41^{st} International Conference on Machine Learning, Vienna, Austria. PMLR 235, 2024.

Currently, the most promising directions for molecule generation in 3D are either auto-regressive models (Gebauer et al., 2019; 2022; Luo & Ji, 2022; Daigavane et al., 2024), building molecules one atom at a time, or Diffusion Models (DM) (Hoogeboom et al., 2022; Vignac et al., 2023; Le et al., 2024) that learn to revert a corruption mechanism that transforms the data distribution into noise. As both approaches directly operate in 3D space, they can leverage the numerous architectures designed for machine learned force fields (Unke et al., 2021), which were carefully developed to encode the symmetries inherent to the data (Schütt et al., 2017; 2021; Batzner et al., 2022; Batatia et al., 2022).

DM have not only been successful at molecule generation, but also on a variety of other data modalities (Yang et al., 2023). Nevertheless, most existing DM pre-specify the forward process, forcing the reverse process to comply with it. A recent line of work has sought to overcome that limitation and improve generation by replacing the fixed forward process with a learnable one (Bartosh et al., 2023; Nielsen et al., 2024; Bartosh et al., 2024).

Contributions In this paper, we present Equivariant Neural Diffusion (END), a novel diffusion model for molecule generation in 3D that (1) is equivariant to Euclidean transformations, and (2) features a learnable forward process. We demonstrate competitive performance in unconditional molecule generation on the QM9 and GEOM-Drugs benchmarks. For conditional generation driven by composition and substructure constraints, our approach exhibits a substantial performance gain compared to existing equivariant diffusion models, underscoring the utility of a learned forward model for effective conditional molecule generation.

2. Background

We begin by establishing the necessary background for generative modeling of geometric graphs. We first introduce the data representation and its inherent symmetries. We then discuss Diffusion Models (DM), and more specifically Equivariant Diffusion Model (EDM) (Hoogeboom et al., 2022). Finally, we present the Neural Flow Diffusion Models (NFDM) framework (Bartosh et al., 2024).

2.1. Equivariance

Molecules as geometric graphs in E(3) We consider geometric graphs embedded in 3-dimensional Euclidean space that represent molecules. Formally, each atomistic system can be described by a tuple $\boldsymbol{x} = (\boldsymbol{r}, \boldsymbol{h})$, where $\boldsymbol{r} = (\boldsymbol{r}_1, ..., \boldsymbol{r}_M) \in \mathbb{R}^{M\times 3}$ form a collection of vectors in 3D representing the coordinates of the atoms, and $\boldsymbol{h} = (\boldsymbol{h}_1, ..., \boldsymbol{h}_M) \in \mathbb{R}^{M\times D}$ are the associated scalar features (e.g. atomic types or charges).

When dealing with molecules, we are particularly interested in $E(3)$, the Euclidean group in 3 dimensions, generated by translations, rotations and reflections. Each group element in $E(3)$ can be represented as a combination of a translation vector $\mathbf{t} \in \mathbb{R}^3$ and an orthogonal matrix $\mathbf{R} \in O(3)$ encoding rotation or reflection. While scalar features $\boldsymbol{h}$ remain invariant, coordinates $\boldsymbol{r}$ transform under translation, rotation and reflection as $\mathbf{R}\boldsymbol{r} + \mathbf{t} = (\mathbf{R}\boldsymbol{r}_1 + \mathbf{t}, ..., \mathbf{R}\boldsymbol{r}_M + \mathbf{t})$.

Equivariant functions A function $f : \mathcal{X} \to \mathcal{Y}$ is said to be equivariant to the action of a group G, or G-equivariant, if $g \cdot f(\boldsymbol{x}) = f(g \cdot \boldsymbol{x}), \forall g \in G$. It is said to be G-invariant, if $f(\boldsymbol{x}) = f(g \cdot \boldsymbol{x}), \forall g \in G$. In the case of a function $f : (\mathbb{R}^{M\times 3} \times \mathbb{R}^{M\times D}) \to (\mathbb{R}^{M\times 3} \times \mathbb{R}^{M\times D})$ operating on geometric graphs, the function is said to be $E(3)$-equivariant if,

$$\mathbf{R}\boldsymbol{y}^{(r)} + \mathbf{t}, \boldsymbol{y}^{(h)} = f\Big(\mathbf{R}\boldsymbol{r} + \mathbf{t}, \boldsymbol{h}\Big),$$

$\forall\, \mathbf{R} \in O(3)$ and $\mathbf{t} \in \mathbb{R}^3$, where $\boldsymbol{y}^{(r)}$ and $\boldsymbol{y}^{(h)}$ denote the output related to $\boldsymbol{r}$ and $\boldsymbol{h}$ respectively. There exists a large variety of graph neural network architectures designed to be equivariant to the Euclidean group (Schütt et al., 2017; 2021; Batzner et al., 2022; Batatia et al., 2022).

Equivariant distributions A conditional distribution $p(\boldsymbol{y}|\boldsymbol{x})$ is equivariant to rotations and reflections when $p(\boldsymbol{y}|\boldsymbol{x}) = p(\mathbf{R}\boldsymbol{y}|\mathbf{R}\boldsymbol{x}), \forall\, \mathbf{R} \in O(3)$, while a distribution is said to be invariant when $p(\boldsymbol{x}) = p(\mathbf{R}\boldsymbol{x}), \forall\, \mathbf{R} \in O(3)$. Regarding translation, it is not possible to have a translation-invariant non-zero distribution, as it would require that $p(\boldsymbol{x}) = p(\boldsymbol{x} + \mathbf{t}), \forall \mathbf{t} \in \mathbb{R}^3, \boldsymbol{x} \in \mathbb{R}^{M\times 3}$, which would mean that $p(\boldsymbol{x})$ cannot integrate to 1 (Garcia Satorras et al., 2021). However, a translation invariant distribution can be constructed in the linear subspace R where the centre of gravity is fixed to $\mathbf{0}$ (i.e. zero CoG subspace): $R = \{\boldsymbol{r} \in \mathbb{R}^{M\times 3} : \frac{1}{M}\sum_{i=1}^{M} \boldsymbol{r}_i = \mathbf{0}\}$ (Xu et al., 2022). As R can be shown to be intrinsically equivalent to $\mathbb{R}^{(M-1)\times 3}$ (Bao et al., 2023), we will consider in what follows that $\boldsymbol{r}$ is defined in $\mathbb{R}^{(M-1)\times 3}$ for ease of notation.

2.2. Equivariant Diffusion Models

Diffusion Models (DM) (Sohl-Dickstein et al., 2015; Ho et al., 2020) are generative models that learn distributions through a hierarchy of latent variables, corresponding to perturbed versions of the data at increasing noise scales. DM consist of a forward and a reverse (or generative) process. The Equivariant Diffusion Model (EDM) (Hoogeboom et al., 2022) is a particular instance of a DM, where the learned marginal $p_\theta(\boldsymbol{x})$ is made invariant to the action of translations, rotations and reflections by construction. Intuitively, this means that the likelihood of a given molecule does not depend on its orientation.

Forward process The forward process perturbs samples from the data distribution, $\boldsymbol{x} \sim q(\boldsymbol{x})$, over time through noise injection, resulting in a trajectory of latent variables $(\boldsymbol{z}_t)_{t\in[0,1]}$, conditional on $\boldsymbol{x}$. The conditional distribution for $(\boldsymbol{z}_t)_{t\in[0,1]}$ given $\boldsymbol{x}$, can be described by an initial distribution $q(\boldsymbol{z}_0|\boldsymbol{x})$ and a Stochastic Differential Equation (SDE),

$$d\{\boldsymbol{z}_t^{(r)}, \boldsymbol{z}_t^{(h)}\} = f(t)\big[\boldsymbol{z}_t^{(r)}, \boldsymbol{z}_t^{(h)}\big]\, dt + g(t)\, d\{\boldsymbol{w}^{(r)}, \boldsymbol{w}^{(h)}\},$$

where the drift $f(t)$ and volatility $g(t)$ are scalar functions of time, and $\boldsymbol{w}^{(r)}$ and $\boldsymbol{w}^{(h)}$ are two independent standard Wiener processes defined in $\mathbb{R}^{(M-1)\times 3}$ and $\mathbb{R}^{M\times D}$ respectively. Specifically, EDM implements the Variance-Preserving SDE (VP-SDE) scheme (Song et al., 2020), with $f(t) = -\frac{1}{2}\beta(t)$ and $g(t) = \sqrt{\beta(t)}$ for a fixed schedule $\beta(t)$. Due to the linearity of the drift term, the conditional marginal distribution can be reconstructed as

$$\begin{aligned} q\big([\boldsymbol{z}_t^{(r)}, \boldsymbol{z}_t^{(h)}]\big|[\boldsymbol{r}, \boldsymbol{h}]\big) &= q(\boldsymbol{z}_t^{(r)}|\boldsymbol{r})q(\boldsymbol{z}_t^{(h)}|\boldsymbol{h}), \\ &= \mathcal{N}(\boldsymbol{z}_t^{(r)}, |\alpha_t\boldsymbol{r}, \sigma_t^2\mathbb{I}) \cdot \mathcal{N}(\boldsymbol{z}_t^{(h)}, |\alpha_t\boldsymbol{h}, \sigma_t^2\mathbb{I}), \end{aligned}$$

where $\alpha_t = \exp\big(-\frac{1}{2}\int_0^t \beta(s)\, ds\big)$ and $\sigma_t = 1 - \exp\big(-\frac{1}{2}\int_0^t \beta(s)\, ds\big)$. The conditional distribution evolves from a low-variance Gaussian centered around the data $q(\boldsymbol{z}_0|\boldsymbol{x}) \approx \mathcal{N}(\boldsymbol{z}_0|\boldsymbol{x}, \delta\mathbb{I})$ to an uninformative prior distribution (that contains no information about the data distribution), i.e. a unit Gaussian $q(\boldsymbol{z}_1|\boldsymbol{x}) \approx \mathcal{N}(\boldsymbol{z}_1|\mathbf{0}, \mathbb{I})$.

Reverse (generative) process Starting from the prior $[\boldsymbol{z}_1^{(r)}, \boldsymbol{z}_1^{(h)}] \sim \mathcal{N}(\boldsymbol{z}_t^{(r)}|\mathbf{0}, \mathbb{I}) \cdot \mathcal{N}(\boldsymbol{z}_t^{(h)}|\mathbf{0}, \mathbb{I})$, samples from $q(\boldsymbol{x})$ can be generated by reversing the forward process. This can be done by following the reverse-time SDE (Anderson, 1982),

$$\begin{aligned} d\{\boldsymbol{z}_t^{(r)}, \boldsymbol{z}_t^{(h)}\} &= f^B(\boldsymbol{z}_t, t)\, dt + g(t)\, d\{\bar{\boldsymbol{w}}^{(r)}, \bar{\boldsymbol{w}}^{(h)}\} \\ &= \Big[f(t)\boldsymbol{z}_t - g^2(t)\nabla_{\boldsymbol{z}_t} \log q(\boldsymbol{z}_t)\Big]\, dt + g(t)\, d\{\bar{\boldsymbol{w}}^{(r)}\bar{\boldsymbol{w}}^{(h)}\}, \end{aligned}$$

where $\boldsymbol{z}_t = \big[\boldsymbol{z}_t^{(r)}, \boldsymbol{z}_t^{(h)}\big]$, $\bar{\boldsymbol{w}}^{(r)}$ and $\bar{\boldsymbol{w}}^{(h)}$ are independent standard Wiener processes defined in $\mathbb{R}^{(M-1)\times 3}$ and $\mathbb{R}^{M\times D}$, respectively, with time flowing backwards. DM approximate the reverse process by learning an approximation of the score function $\nabla_{\boldsymbol{z}_t} \log q(\boldsymbol{z}_t)$ parameterized by a neural network $s_\theta(\boldsymbol{z}_t, t)$. With the learned score function $s_\theta(\boldsymbol{z}_t, t)$, a sample $\boldsymbol{z}_0 \sim p_\theta(\boldsymbol{z}_0) \approx q(\boldsymbol{z}_0) \approx q(\boldsymbol{x})$ can

be obtained by first sampling from the prior $[\boldsymbol{z}_1^{(r)}, \boldsymbol{z}_1^{(h)}] \sim \mathcal{N}(\boldsymbol{z}_t^{(r)}|\mathbf{0}, \mathbb{I}) \cdot \mathcal{N}(\boldsymbol{z}_t^{(h)}|\mathbf{0}, \mathbb{I})$, and then simulating the reverse SDE,

$$d\boldsymbol{z}_t = \Big[f(t)\boldsymbol{z}_t - g^2(t)s_\theta(\boldsymbol{z}_t, t)\Big]\, dt + g(t)\, d\{\bar{\boldsymbol{w}}^{(r)}, \bar{\boldsymbol{w}}^{(h)}\},$$

where the true score function has been replaced by its approximation $s_\theta(\boldsymbol{z}_t, t)$. In EDM, the approximate score is parameterized through an equivariant function: $s_\theta(\boldsymbol{z}_t, t) = \big[s_\theta^{(r)}(\boldsymbol{z}_t, t), s_\theta^{(h)}(\boldsymbol{z}_t, t)\big]$ such that $s_\theta([\mathbf{R}\boldsymbol{z}_t^{(r)}, \boldsymbol{z}_t^{(h)}], t) = \big[\mathbf{R}s_\theta^{(r)}(\boldsymbol{z}_t, t), s_\theta^{(h)}(\boldsymbol{z}_t, t)\big], \forall\ \mathbf{R} \in O(3)$. This is realised through the specific parameterization,

$$s_\theta(\boldsymbol{z}_t, t) = \frac{\alpha_t \hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t) - \boldsymbol{z}_t}{\sigma_t^2},$$

where the data point predictor $\hat{\boldsymbol{x}}_\theta$ is implemented by an equivariant neural network.

Optimization The data point predictor $\hat{\boldsymbol{x}}_\theta$, or s_θ, is trained by optimizing the denoising score matching loss

$$\mathcal{L}_{\text{DSM}} = \mathbb{E}_{\substack{u(t) \\ q(\boldsymbol{x}, \boldsymbol{z}_t)}} \Big[\lambda(t)\big|\big|s_\theta(\boldsymbol{z}_t, t) - \nabla_{\boldsymbol{z}_t} \log q(\boldsymbol{z}_t|\boldsymbol{x})\big|\big|_2^2\Big],$$

where $\lambda(t)$ is a positive weighting function, and $u(t)$ is a uniform distribution over the interval $[0, 1]$.

2.3. Neural Flow Diffusion Models

Neural Flow Diffusion Models (NFDM) (Bartosh et al., 2024) are based on the observation that latent variables, $\boldsymbol{z}_t$, in conventional DM are inferred through a pre-specified transformation. This potentially limits the flexibility of the latent space, and make the learning of the reverse (generative) process more challenging.

Forward process In contrast to conventional DMs, NFDM defines the forward process implicitly through a learnable transformation $F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x})$ of injected noise $\boldsymbol{\varepsilon}$, time t, and data point $\boldsymbol{x}$. The latent variables $\boldsymbol{z}_t$ are obtained by transforming noise samples $\boldsymbol{\varepsilon}$, conditional on data point $\boldsymbol{x}$ and time step t: $\boldsymbol{z}_t = F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x})$. If F_φ is differentiable with respect to $\boldsymbol{\varepsilon}$ and t, and invertible with respect to $\boldsymbol{\varepsilon}$, then, for fixed $\boldsymbol{x}$ and $\boldsymbol{\varepsilon}$, samples from $q_\varphi(\boldsymbol{z}_t|\boldsymbol{x})$ can be obtained by solving the following conditional Ordinary Differential Equation (ODE) until time t,

$$d\boldsymbol{z}_t = f_\varphi(\boldsymbol{z}_t, t, \boldsymbol{x})\, dt, \tag{1}$$

where $f_\varphi(\boldsymbol{z}_t, t, \boldsymbol{x}) = \frac{\partial F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x})}{\partial t}\Big|_{\boldsymbol{\varepsilon} = F_\varphi^{-1}(\boldsymbol{z}_t, t, \boldsymbol{x})}$, and with $\boldsymbol{z}_0 \sim q(\boldsymbol{z}_0|\boldsymbol{x})$. While F_φ and $q(\boldsymbol{\varepsilon})$ define the conditional marginal distribution $q_\varphi(\boldsymbol{z}_t|\boldsymbol{x})$, we need a distribution over the trajectories $(\boldsymbol{z}_t)_{t\in[0,1]}$. NFDM obtains this through the introduction of a conditional SDE starting from $\boldsymbol{z}_0$ and running forward in time. Given access to the ODE in Equation (1) and the score function $\nabla_{\boldsymbol{z}_t} \log q_\varphi(\boldsymbol{z}_t|\boldsymbol{x})$, a conditional SDE with conditional marginal distribution $q_\varphi(\boldsymbol{z}_t|\boldsymbol{x})$ is given by

$$d\boldsymbol{z}_t = f_\varphi^F(\boldsymbol{z}_t, t, \boldsymbol{x})\, dt + g_\varphi(t)\, d\boldsymbol{w}, \tag{2}$$

where $f_\varphi^F(\boldsymbol{z}_t, t, \boldsymbol{x}) = f_\varphi(\boldsymbol{z}_t, t, \boldsymbol{x}) + \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{z}_t} \log q_\varphi(\boldsymbol{z}_t|\boldsymbol{x})$. The score function of $q_\varphi(\boldsymbol{z}_t|\boldsymbol{x})$ is

$$\nabla_{\boldsymbol{z}_t} \log q_\varphi(\boldsymbol{z}_t|\boldsymbol{x}) = \nabla_{\boldsymbol{z}_t}\big[\log q(\boldsymbol{\varepsilon}) + \log\big|J_F^{-1}\big|\big], \tag{3}$$

with $\boldsymbol{\varepsilon} = F_\varphi^{-1}(\boldsymbol{z}_t, t, \boldsymbol{x})$, and $J_F^{-1} = \frac{\partial F_\varphi^{-1}(\boldsymbol{z}_t, t, \boldsymbol{x})}{\partial \boldsymbol{z}_t}$.

Reverse (generative) process A conditional reverse SDE that starts from $\boldsymbol{z}_1 \sim q(\boldsymbol{z}_1)$, runs backward in time, and reverses the conditional forward SDE from Equation (2) can be defined as

$$\begin{aligned} d\boldsymbol{z}_t &= f_\varphi^B(\boldsymbol{z}_t, t, \boldsymbol{x}) + g_\varphi(t)\, d\bar{\boldsymbol{w}} \\ &= \Big[f_\varphi(\boldsymbol{z}_t, t, \boldsymbol{x}) - \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{z}_t} \log q_\varphi(\boldsymbol{z}_t|\boldsymbol{x})\Big]\, dt + g_\varphi(t)\, d\bar{\boldsymbol{w}}\,. \end{aligned}$$

As we do not have access to $\boldsymbol{x}$ when generating samples, we can rewrite Equation (2), with the prediction of $\boldsymbol{x}$,

$$d\boldsymbol{z}_t = \hat{f}_{\theta,\varphi}(\boldsymbol{z}_t, t)\, dt + g_\varphi(t)\, d\bar{\boldsymbol{w}}, \tag{4}$$

where $\hat{f}_{\theta,\varphi}(\boldsymbol{z}_t, t) = f_\varphi^B\big(\boldsymbol{z}_t, t, \hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t)\big)$, and $\hat{\boldsymbol{x}}_\theta$ is a function that predicts the data point $\boldsymbol{x}$. Provided that the reconstruction distribution $q(\boldsymbol{z}_0|\boldsymbol{x})$ and the prior distribution $q(\boldsymbol{z}_1)$ are defined, this fully specifies the reverse process.

Optimization The forward and reverse processes can be optimized jointly by matching the drift terms of the true and approximate conditional reverse SDEs,

$$\mathcal{L}_{\text{NFDM}} = \mathbb{E}_{\substack{u(t) \\ q_\varphi(\boldsymbol{x}, \boldsymbol{z}_t)}} \Big[\frac{1}{2g_\varphi^2(t)}\big|\big|f_\varphi^B(\boldsymbol{z}_t, t, \boldsymbol{x}) - \hat{f}_{\theta,\varphi}(\boldsymbol{z}_t, t)\big|\big|_2^2\Big]. \tag{5}$$

3. Equivariant Neural Diffusion

Equivariant Neural Diffusion (END) generalizes the Equivariant Diffusion Model (EDM) (Hoogeboom et al., 2022), by defining the forward process through a learnable transformation. Our approach is a synthesis of NFDM introduced in Section 2.3, and leverages ideas of EDM outlined in Section 2.2 to maintain the desired invariance of the learned marginal distribution $p_{\theta,\varphi}(\boldsymbol{z}_0)$. By providing an equivariant learnable transformation F_φ and an equivariant data point predictor $\hat{\boldsymbol{x}}_\theta$, we show that it is possible to obtain a generative model with the desired properties. Finally, we propose a simple yet flexible parameterization meeting the requirements.

3.1. Formulation

The key innovation in END lies in its forward process, which is also leveraged in the reverse (generative) process. The forward process is defined through a learnable time- and data-dependent transformation $F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x})$, such that the latent $\boldsymbol{z}_t$ transforms covariantly with the injected noise $\boldsymbol{\varepsilon}$ (i.e. a collection of random vectors) and the data point $\boldsymbol{x}$,

$$F_\varphi(\mathbf{R}\boldsymbol{\varepsilon}, t, \mathbf{R}\boldsymbol{x}) = \mathbf{R}F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x}) = \mathbf{R}\boldsymbol{z}_t.$$

We then define $\hat{\boldsymbol{x}}_\theta$ as another learnable equivariant function, implying that the predicted data point transforms covariantly with the latent $\boldsymbol{z}_t$, i.e. $\hat{\boldsymbol{x}}_\theta(\mathbf{R}\boldsymbol{z}_t, t) = \mathbf{R}\hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t)$. Finally, we choose the noise distribution $p(\boldsymbol{\varepsilon})$, and the prior distribution $p(\boldsymbol{z}_1)$ to be invariant to the considered symmetry group.

Invariance of the learned distribution With the following choices: (1) $p(\boldsymbol{z}_1)$ an invariant distribution, (2) F_φ an equivariant function that satisfies $F_\varphi(\mathbf{R}\boldsymbol{\varepsilon}, t, \mathbf{R}\boldsymbol{x}) = \mathbf{R}F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x})$, and (3) $\hat{\boldsymbol{x}}_\theta$ an equivariant function, we have that the learned marginal $p_{\theta,\varphi}(\boldsymbol{z}_0)$ is invariant as desired. This can be shown by demonstrating that the reverse SDE is equivariant. We start by noting that the reverse SDE in END is given by

$$d\boldsymbol{z}_t = \hat{f}_{\theta,\varphi}(\boldsymbol{z}_t, t)\, dt + g_\varphi(t)\, d\bar{\boldsymbol{w}}\,.$$

As the Wiener process is isotropic, this boils down to showing that the drift term, $\hat{f}_{\theta,\varphi}, (\boldsymbol{z}_t, t)$ is equivariant, i.e. $\hat{f}_{\theta,\varphi}(\mathbf{R}\boldsymbol{z}_t, t) = \mathbf{R}\hat{f}_{\theta,\varphi}(\boldsymbol{z}_t, t)$. As the drift is expressed as a sum of two terms, we inspect each of them separately. The first term is

$$f_\varphi\big(\boldsymbol{z}_t, t, \hat{\boldsymbol{x}}_\theta\big) = \frac{\partial F_\varphi\big(\boldsymbol{\varepsilon}, t, \hat{\boldsymbol{x}}_\theta\big)}{\partial t}\Big|_{\boldsymbol{\varepsilon}=F_\varphi^{-1}(\boldsymbol{z}_t, t, \hat{\boldsymbol{x}}_\theta)},$$

where $\hat{\boldsymbol{x}}_\theta = \hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t)$.

If F_φ is equivariant, then so is its time-derivative (see Appendix A.1). The same holds for its inverse with respect to $\boldsymbol{\varepsilon}$ (see Appendix A.2), such that we have $F_\varphi^{-1}(\mathbf{R}\boldsymbol{z}_t, t, \mathbf{R}\boldsymbol{x}) = \mathbf{R}F_\varphi^{-1}(\boldsymbol{z}_t, t, \boldsymbol{x}) = \mathbf{R}\boldsymbol{\varepsilon}$. We additionally have that $\hat{\boldsymbol{x}}_\theta$ is equivariant by definition.

As the equivariance of F_φ implies the equivariance of q_φ, when looking at the second term of the drift, we can see that, for $\boldsymbol{y}_t = \mathbf{R}\boldsymbol{z}_t$, we have

$$\nabla_{\boldsymbol{y}_t} \log q_\varphi\big(\boldsymbol{y}_t | \hat{\boldsymbol{x}}_\theta(\boldsymbol{y}_t, t)\big) = \mathbf{R}\nabla_{\boldsymbol{z}_t} \log q_\varphi\big(\boldsymbol{z}_t | \hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t)\big).$$

In summary, in addition to an invariant prior, an equivariant F_φ and an equivariant $\hat{\boldsymbol{x}}_\theta$ ensure the equivariance of the reverse process, and hence the invariance of the learned distribution. In Appendix A.3, we additionally show that the objective function in Equation (5) is invariant, i.e. $\mathcal{L}_{\text{END}}(\mathbf{R}\boldsymbol{x}) = \mathcal{L}_{\text{END}}(\boldsymbol{x}), \forall \boldsymbol{R} \in O(3)$.

3.2. Parameterization

We now introduce a simple parameterization of F_φ that meets the requirements outlined above:

$$F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x}) = \mu_\varphi(\boldsymbol{x}, t) + U_\varphi(\boldsymbol{x}, t)\boldsymbol{\varepsilon}, \tag{6}$$

where, due to the geometric nature of $\boldsymbol{x}$, $U_\varphi(\boldsymbol{x}, t) \in \mathbb{R}^{(M-1)\times 3\times 3}$ is structured as a block-diagonal matrix where each block is a 3×3 matrix. This is the equivalent to a diagonal parameterization in the case of scalar features. Similarly to EDM, our parametrization of F_φ leads to a conditional marginal $q_\varphi(\boldsymbol{z}_t|\boldsymbol{x})$ that is a conditional Gaussian with (block-) diagonal covariance, with the notable difference that the mean and covariance are now learnable through F_φ,

$$q_\varphi(\boldsymbol{z}_t|\boldsymbol{x}) = \mathcal{N}\big(\boldsymbol{z}_t | \mu_\varphi(\boldsymbol{x}, t), \Sigma_\varphi(\boldsymbol{x}, t)\big), \tag{7}$$

where $\Sigma_\varphi(\boldsymbol{x}, t) = U_\varphi(\boldsymbol{x}, t)U_\varphi^\top(\boldsymbol{x}, t)$ such that $\Sigma_\varphi(\boldsymbol{x}, t)$ is also block-diagonal.

As F_φ is linear in $\boldsymbol{\varepsilon}$, both μ_φ and U_φ must be equivariant functions whose outputs transform covariantly with $\boldsymbol{x}$, to ensure the equivariance of $F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x})$,

$$\begin{aligned} F_\varphi(\mathbf{R}\boldsymbol{\varepsilon}, t, \mathbf{R}\boldsymbol{x}) &= \mu_\varphi(\mathbf{R}\boldsymbol{x}, t) + U_\varphi(\mathbf{R}\boldsymbol{x}, t)\mathbf{R}\boldsymbol{\varepsilon}, \\ &= \mathbf{R}\mu_\varphi(\boldsymbol{x}, t) + \mathbf{R}U_\varphi(\boldsymbol{x}, t)\mathbf{R}\boldsymbol{\varepsilon} \\ &= \mathbf{R}F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x}), \end{aligned}$$

as desired. We can then readily check that q_φ is equivariant, as $\forall\ \mathbf{R} \in O(3)$, we have that

$$\begin{aligned} q_\varphi(\boldsymbol{z}_t|\boldsymbol{x}) &= \mathcal{N}\big(\boldsymbol{z}_t | \mu_\varphi(\boldsymbol{x}, t), \Sigma_\varphi(\boldsymbol{x}, t)\big) \\ &= \mathcal{N}\big(\mathbf{R}\boldsymbol{z}_t | \mathbf{R}\mu_\varphi(\boldsymbol{x}, t), \mathbf{R}\Sigma_\varphi(\boldsymbol{x}, t)\mathbf{R}^\top\big) \\ &= q_\varphi(\mathbf{R}\boldsymbol{z}_t | \mathbf{R}\boldsymbol{x}). \end{aligned}$$

Prior and Reconstruction While not strictly required, it can be advantageous to parameterize F_φ such that the prior and reconstruction losses do not need to be computed. To do so, we need to make sure that $F_\varphi(\boldsymbol{\varepsilon}, \boldsymbol{x}, t)$ is such that (i) $q(\boldsymbol{z}_0|\boldsymbol{x}) \approx \mathcal{N}(\boldsymbol{z}_0|\boldsymbol{x}, \delta^2\mathbb{I})$, and (ii) $q(\boldsymbol{z}_1|\boldsymbol{x}) \approx \mathcal{N}(\mathbf{0}, \mathbb{I})$. We parameterize the mean function as

$$\mu_\varphi(\boldsymbol{x}, t) = (1-t)\boldsymbol{x} + t\big(1-t\big)\bar{\mu}_\varphi(\boldsymbol{x}, t), \tag{8}$$

which ensures that $\mu_\varphi(\boldsymbol{x}, 0) = \boldsymbol{x}$, and $\mu_\varphi(\boldsymbol{x}, 1) = \mathbf{0}$; whereas for $U_\varphi(\boldsymbol{x}, t)$, we use the following

$$\begin{aligned} U_\varphi(\boldsymbol{x}, t) = \big(\delta^{1-t}\bar{\sigma}_\varphi(\boldsymbol{x}, t)^{t(1-t)}\big)\mathbb{I} \\ + t\big(1-t\big)\bar{U}_\varphi(\boldsymbol{x}, t), \end{aligned} \tag{9}$$

which ensures that $\Sigma_\varphi(\boldsymbol{x}, 0) = \delta^2\mathbb{I}$, and $\Sigma_\varphi(\boldsymbol{x}, 1) = \mathbb{I}$, while being unconstrained for $t \in]0, 1[$. We give additional details about F_φ in Appendix A.4.

Implementation In practice, F_φ is implemented as a neural network with an architecture similar to that of the data point predictor $\hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t)$, but with a specific readout layer that produces $[\bar{\mu}_\varphi(\boldsymbol{x}, t), \bar{\sigma}_\varphi, \bar{U}_\varphi(\boldsymbol{x}, t)]$. The mean output $\bar{\mu}_\varphi(\boldsymbol{x}, t)$ is similar to that of $\hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t)$. For $U_\varphi(\boldsymbol{x}, t)$, we need to ensure that $\Sigma_\varphi(\boldsymbol{x}, t) = U_\varphi(\boldsymbol{x}, t)U_\varphi^\top(\boldsymbol{x}, t)$ rotates properly: $\bar{\sigma}_\varphi(\boldsymbol{x}, t)$ is a positive invariant scalar, while $\bar{U}_\varphi(\boldsymbol{x}, t)$ is constructed as matrix whose columns are vectors that transform covariantly with $\boldsymbol{x}$.

For ease of notation, we introduced all notations in the linear subspace R, however in practice we work in the ambient space, i.e. $\boldsymbol{r} \in \mathbb{R}^{M\times 3}$. We detail in Appendix A.4.1, how working in ambient space is possible.

The training and sampling procedures are detailed in Algorithms 1 and 2 in the appendix.

3.3. Conditional Generation

While unconditional generation is a required stepping stone, many real-life applications require some form of controllability. As other generative models, DM can model conditional distributions $p(\boldsymbol{x}|\boldsymbol{c})$, where $\boldsymbol{c}$ is a given condition. While different methods exist for sampling from the conditional distribution, e.g. (Wu et al., 2024), the simplest approach consists in training a conditional model on pairs $(\boldsymbol{x}, \boldsymbol{c})$. In that setting, F_φ and $\hat{x}_\theta$ are simply provided with an extra input $\boldsymbol{c}$ representing the conditional information, such that they respectively become $F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x}, \boldsymbol{c})$, and $\hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t, \boldsymbol{c})$. It is important to note that, compared to conventional DM, the forward process of END is now condition-dependent.

4. Experiments

In this section, we demonstrate the benefits of END with a comprehensive set of experiments. In Section 4.1, we first display the advantages of END for unconditional generation on 2 standard benchmarks, namely QM9 (Ramakrishnan et al., 2014) and GEOM-DRUGS (Axelrod & Gomez-Bombarelli, 2022), including an ablation study on QM9. Then, in Section 4.2, we perform conditional generation in 2 distinct settings on QM9.

4.1. Unconditional Generation

Datasets The QM9 dataset (Ramakrishnan et al., 2014) contains 134 thousand small- and medium-sized organic molecules with up to 9 heavy atoms, and up to 29 when counting hydrogen atoms. GEOM-DRUGS (Axelrod & Gomez-Bombarelli, 2022) contains 430 thousand medium- and large-sized drug-like molecules with 44 atoms on average, and up to maximum 181 atoms. We follow the same setup as previous work (Hoogeboom et al., 2022).

Task and Evaluation For each method, we sample 10 000 molecules using the stochastic sampling procedure detailed in Algorithm 2, with the number of integration steps varying from 50 to 1000. We repeat each sampling for 3 seeds, and report averages along with standard deviations for each metric.

On QM9, we follow previous work (Hoogeboom et al., 2022; Xu et al., 2023), and first evaluate the chemical quality of the generated samples in terms of stability, validity, and uniqueness. We additionally evaluate the ability of the model to learn the atom and bond types distributions by measuring the total variation between the dataset's and generated distributions. Finally, we evaluate the geometry of the generated molecules, by computing the MMD (Gretton et al., 2012) between the generated and true bond distances distributions for the 10 most common bonds. We provide additional details about the evaluation procedure in Appendix A.6.1.

On GEOM-DRUGS, in addition to atom stability and validity as commonly reported in previous work (Hoogeboom et al., 2022; Xu et al., 2023), we also compute connectivity and total variation for atom types. Connectivity accounts for the fact that validity can easily be increased by generating several disconnected fragments (where only the largest counts towards validity), while the total variation ensures that the model properly samples all atom types.

Baselines We compare END to several relevant baselines from the literature: the original EDM (Hoogeboom et al., 2022), EDM-BRIDGE (Wu et al., 2022) an improved version of EDM that adds a physics-inspired force guidance in the reverse process, and GEOLDM (Xu et al., 2023) an equivariant latent DM. For a fair comparison, we additionally implement our own EDM (Hoogeboom et al., 2022), and denote it EDM*. It features the exact same architecture as END, as well as the same amount of learnable parameters. In EDM*, $\hat{\boldsymbol{x}}_\theta$ is made of 10 layers, while END comprises a 5-layer $\hat{\boldsymbol{x}}_\theta$ and a 5-layer F_φ. We provide additional details in Appendix A.6.

Results on QM9 Our results on the QM9 dataset are summarized in Table 1, and a few illustrative samples are shown in Figure 1. In addition to comparing with baselines from the literature, we conduct an ablation study, where EDM* + γ is similar to EDM*, but with a learned SNR (Kingma et al., 2021) for each data modality (i.e. atomic types and coordinates), and END (μ_φ only) where only the mean of the conditional marginal is learned. Details about the compared models are provided in Table 5.

We observe that the addition of a learnable forward allows for improved generative modeling, as the two variants of END are shown to perform better than (or be on par with) all baselines across all metrics. Most notably, with as few

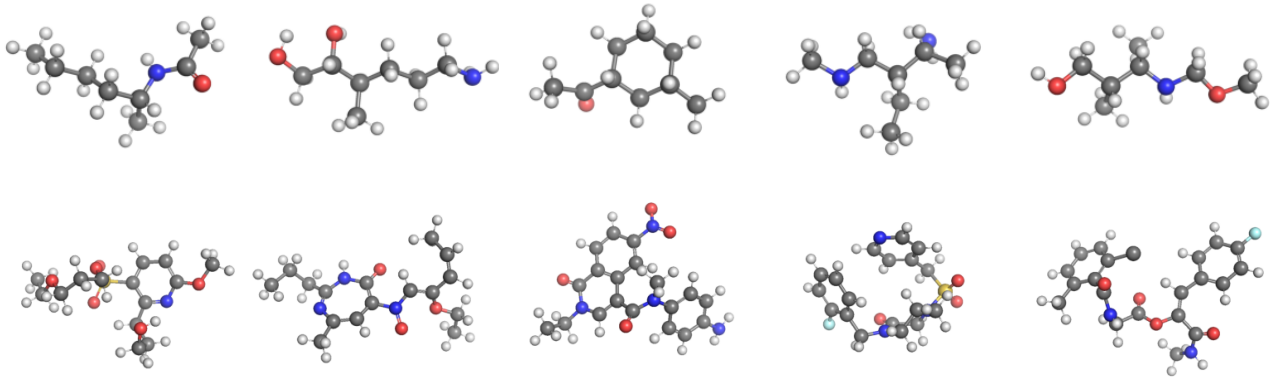

Figure 1: Representative samples generated by END on QM9 (top row), and GEOM-DRUGS (bottom row).

Table 1: Results on QM9. Metrics are obtained over 10000 samples, with mean/standard deviation across 3 sampling runs. For baselines from the literature, results are extracted from the respective papers. The two variants of END compare favorably to baselines across metrics, while offering competitive performance for reduced number of sampling steps.

		Stability (↑)		Validity / Uniqueness (↑)		Total Variation (↓)		MMD (↓)
		A [%]	M [%]	V [%]	V×U [%]	A [10^{-2}]	B [10^{-3}]	[10^{-1}]
Model	Steps							
Data		99.0	95.2	97.7	97.7			
EDM (Hoogeboom et al., 2022)	1000	98.7	82.0	91.9	90.7			
EDM-BRIDGE (Wu et al., 2022)	1000	98.8	84.6	92.0	90.7			
GEOLDM (Xu et al., 2023)	1000	$98.9_{\pm.1}$	$89.4_{\pm.5}$	$93.8_{\pm.4}$	$92.7_{\pm.5}$			
EDM*	50	$97.6_{\pm.0}$	$77.6_{\pm.5}$	$90.2_{\pm.2}$	$89.2_{\pm.2}$	$4.6_{\pm.1}$	$1.7_{\pm.5}$	$1.91_{\pm.03}$
	100	$98.1_{\pm.0}$	$81.9_{\pm.4}$	$92.1_{\pm.2}$	$90.9_{\pm.2}$	$3.5_{\pm.1}$	$1.4_{\pm.3}$	$1.67_{\pm.02}$
	250	$98.3_{\pm.0}$	$84.3_{\pm.1}$	$93.2_{\pm.4}$	$91.7_{\pm.3}$	$2.8_{\pm.2}$	$1.3_{\pm.4}$	$1.52_{\pm.02}$
	500	$98.4_{\pm.0}$	$85.2_{\pm.5}$	$93.5_{\pm.2}$	$92.2_{\pm.3}$	$2.6_{\pm.2}$	$1.3_{\pm.4}$	$1.50_{\pm.04}$
	1000	$98.4_{\pm.0}$	$85.3_{\pm.3}$	$93.5_{\pm.1}$	$91.9_{\pm.1}$	$2.5_{\pm.1}$	$1.4_{\pm.4}$	$1.51_{\pm.02}$
EDM* + γ_φ	50	$97.7_{\pm.0}$	$77.4_{\pm.3}$	$91.1_{\pm.4}$	$90.2_{\pm.4}$	$4.3_{\pm.1}$	$1.5_{\pm.2}$	$2.04_{\pm.02}$
	100	$98.2_{\pm.0}$	$82.6_{\pm.2}$	$92.9_{\pm.2}$	$91.6_{\pm.2}$	$3.2_{\pm.1}$	$1.2_{\pm.2}$	$1.66_{\pm.01}$
	250	$98.5_{\pm.0}$	$85.3_{\pm.3}$	$93.9_{\pm.1}$	$92.4_{\pm.1}$	$2.5_{\pm.1}$	$1.0_{\pm.1}$	$1.54_{\pm.02}$
	500	$98.5_{\pm.1}$	$86.1_{\pm.4}$	$94.1_{\pm.2}$	$92.5_{\pm.2}$	$2.2_{\pm.1}$	$1.0_{\pm.3}$	$1.50_{\pm.02}$
	1000	$98.5_{\pm.0}$	$86.1_{\pm.3}$	$94.1_{\pm.2}$	$92.4_{\pm.2}$	$2.1_{\pm.1}$	$1.1_{\pm.1}$	$1.45_{\pm.03}$
END (μ_φ only)	50	$98.5_{\pm.0}$	$83.9_{\pm.2}$	$95.2_{\pm.2}$	$93.8_{\pm.3}$	$1.4_{\pm.1}$	$1.9_{\pm.4}$	$2.80_{\pm.06}$
	100	$98.7_{\pm.0}$	$87.0_{\pm.3}$	$95.5_{\pm.2}$	$93.6_{\pm.2}$	$1.1_{\pm.0}$	$1.5_{\pm.2}$	$1.97_{\pm.05}$
	250	$98.9_{\pm.0}$	$89.0_{\pm.2}$	$95.8_{\pm.2}$	$93.8_{\pm.2}$	$1.0_{\pm.0}$	$0.5_{\pm.1}$	$1.48_{\pm.03}$
	500	$98.9_{\pm.0}$	$88.6_{\pm.2}$	$95.6_{\pm.1}$	$93.5_{\pm.1}$	$0.9_{\pm.0}$	$0.7_{\pm.1}$	$1.36_{\pm.03}$
	1000	$98.9_{\pm.0}$	$89.2_{\pm.3}$	$95.6_{\pm.1}$	$93.5_{\pm.1}$	$0.9_{\pm.2}$	$1.0_{\pm.1}$	$1.36_{\pm.02}$
END	50	$98.6_{\pm.0}$	$84.6_{\pm.1}$	$92.7_{\pm.1}$	$91.4_{\pm.1}$	$1.5_{\pm.1}$	$1.9_{\pm.4}$	$1.91_{\pm.00}$
	100	$98.8_{\pm.0}$	$87.4_{\pm.2}$	$94.1_{\pm.0}$	$92.3_{\pm.2}$	$1.3_{\pm.0}$	$1.8_{\pm.3}$	$1.63_{\pm.02}$
	250	$98.9_{\pm.1}$	$88.8_{\pm.5}$	$94.7_{\pm.2}$	$92.6_{\pm.1}$	$1.2_{\pm.1}$	$0.8_{\pm.2}$	$1.44_{\pm.04}$
	500	$98.9_{\pm.0}$	$88.8_{\pm.4}$	$94.8_{\pm.2}$	$92.8_{\pm.2}$	$1.2_{\pm.1}$	$0.8_{\pm.5}$	$1.41_{\pm.01}$
	1000	$98.9_{\pm.0}$	$89.1_{\pm.1}$	$94.8_{\pm.1}$	$92.6_{\pm.2}$	$1.2_{\pm.1}$	$0.8_{\pm.5}$	$1.37_{\pm.04}$

Table 2: Results on GEOM-DRUGS. Metrics are obtained over 10000 samples, with mean/standard deviation across 3 sampling runs. For baselines from the literature, results are extracted from the respective papers. Most notably, END generates more connected samples.

Model	Steps	Stability (↑) A [%]	Val. / Conn. (↑) V [%]	V×C [%]	TV (↓) A [10^{-2}]
Data		86.5	99.0		
EDM (Hoogeboom et al., 2022)	1000	81.3	92.6	–	–
EDM-BRIDGE (Wu et al., 2022)	1000	82.4	–	–	–
GEOLDM (Xu et al., 2023)	1000	84.4	99.3	–	–
EDM*	50	$84.7_{\pm.0}$	$93.6_{\pm.2}$	$46.6_{\pm.3}$	$10.5_{\pm.1}$
	100	$85.2_{\pm.1}$	$93.8_{\pm.3}$	$56.2_{\pm.4}$	$8.0_{\pm.1}$
	250	$85.4_{\pm.0}$	$94.2_{\pm.1}$	$61.4_{\pm.6}$	$6.7_{\pm.1}$
	500	$85.4_{\pm.0}$	$94.3_{\pm.2}$	$63.4_{\pm.1}$	$6.4_{\pm.1}$
	1000	$85.3_{\pm.1}$	$94.4_{\pm.1}$	$64.2_{\pm.6}$	$6.2_{\pm.0}$
END	50	$87.8_{\pm.0}$	$89.8_{\pm.2}$	$68.2_{\pm.9}$	$5.7_{\pm.1}$
	100	$87.6_{\pm.1}$	$91.5_{\pm.1}$	$76.0_{\pm.3}$	$4.6_{\pm.2}$
	250	$87.2_{\pm.0}$	$92.4_{\pm.4}$	$80.0_{\pm.3}$	$3.5_{\pm.3}$
	500	$87.1_{\pm.0}$	$92.8_{\pm.3}$	$81.1_{\pm.5}$	$3.3_{\pm.2}$
	1000	$87.0_{\pm.0}$	$92.9_{\pm.3}$	$82.2_{\pm.2}$	$3.0_{\pm.3}$

as 100 integration steps, END is able to generate samples that are qualitatively better than those generated by most baselines in 1000 steps.

Results on GEOM-DRUGS Our results on the GEOM-DRUGS dataset are presented in Table 2, and a few samples are displayed in Figure 1. We observe that END outperforms all baselines across all metrics but validity. This is due to the fact that the validity metric, as computed by (Hoogeboom et al., 2022; Xu et al., 2023), is obtained by only considering the largest fragment that can be extracted from a given sample. A method generating smaller disconnected fragments, can easily exploit and artificially increase this metric. When discarding disconnected samples, we observe that END does much better than the baseline, with an increase of around 20% on average.

4.2. Controllable Generation

Dataset and Setup We perform our experiments on the QM9 dataset, on 2 different tasks: composition-conditioned generation, and substructure-conditioned generation. Both tasks allow for direct validation with ground-truth properties without requiring expensive quantum mechanics calculations, or approximations with surrogate models. In each case, we train a conditional diffusion model as described in Section 3.3, i.e. where F_φ and $\hat{x}_\theta$ are provided with an extra input corresponding to the condition. Additional details are provided in Appendix A.6.4.

Task 1: composition-conditioned generation The model is tasked to generate a compound with a predefined composition, i.e. structural isomers of a given formula. The condition is specified as a vector $\boldsymbol{c} = (c_1, ..., c_D) \in \mathbb{Z}^D$, where c_d denotes the number of atoms of type d that the sample should contain. To evaluate the model, we generate 10 samples per target formula, and compute the proportion of samples that match the provided composition. Our results are provided in Table 3, where we observe that conditional END significantly outperforms the baseline, and offers nearly perfectly controllable composition generation. Additionally, we can also see that reducing the number of sampling steps has a very limited impact on the controllability.

Table 3: Results on composition-conditioned generation, where CEND offers nearly perfect composition controllability. Matching refers to the proportion of samples featuring the prompted composition.

Model	Steps	Matching [%] (↑)
CEDM*	50	$69.6_{\pm.6}$
	100	$73.0_{\pm.6}$
	250	$74.1_{\pm1.4}$
	500	$76.2_{\pm.6}$
	1000	$75.5_{\pm.5}$
CEND	50	$89.2_{\pm0.8}$
	100	$90.1_{\pm1.0}$
	250	$91.2_{\pm0.8}$
	500	$91.5_{\pm0.8}$
	1000	$91.0_{\pm0.9}$

Task 2: substructure-conditioned generation We follow the same setup as (Bao et al., 2023) and train a con-

Table 4: Results on substructure-conditional sampling. For baselines from the literature, results are borrowed from the corresponding paper (Bao et al., 2023). cEND shows competitive performance, even surpassing EEGSDE that leverages an additional property predictor.

Model	Steps	Tanimoto Sim. (↑)
cEDM (Bao et al., 2023)	1000	$0.671_{\pm.004}$
EEGSDE (Bao et al., 2023)	1000	$0.750_{\pm.003}$
cEDM*	50	$0.601_{\pm.000}$
	100	$0.640_{\pm.002}$
	250	$0.663_{\pm.002}$
	500	$0.669_{\pm.001}$
	1000	$0.673_{\pm.002}$
cEND	50	$0.783_{\pm.001}$
	100	$0.807_{\pm.001}$
	250	$0.819_{\pm.001}$
	500	$0.825_{\pm.001}$
	1000	$0.828_{\pm.001}$

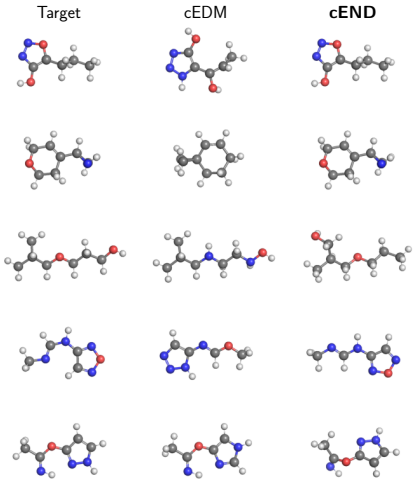

Figure 2: Excerpt of substructure-conditioned samples., where cEND can be seen to match the provided substructure better (in terms of compositions and local patterns).

ditional END, where the condition is a molecular fingerprint encoding structural information about the molecule. A fingerprint is a binary vector $\boldsymbol{c} = (c_1, ..., c_F) \in \{0, 1\}^F$, where c_f is set to 1 if substructure f is present in the molecule, or to 0 if not. Fingerprints are obtained using OpenBabel (O'Boyle et al., 2011). To evaluate the ability of the compared models to leverage the provided structural information, we evaluate them by conditioning on unseen fingerprints (taken from test set) at sampling time. We then compute the similarity between the fingerprints computed on the generated samples and the fingerprints provided as inputs. We compare cEND to EEGSDE (Bao et al., 2023), an improved version of EDM (Hoogeboom et al., 2022), that performs conditional generation by combining a conditional diffusion model and regressor guidance. Our results are presented in Table 4, along with a handful of samples in Figure 2. cEND offers better controllability than the compared baselines, as highlighted by the higher similarity.

5. Related Work

The main approaches to molecule generation in 3D are autoregressive models (Gebauer et al., 2019; Simm et al., 2020; Gebauer et al., 2022; Luo & Ji, 2022; Daigavane et al., 2024), flow-based models (Garcia Satorras et al., 2021), and diffusion models (Hoogeboom et al., 2022; Igashov et al., 2024). A notable exception to the geometric graph representation of 3D molecules are voxels (Skalic et al., 2019; Ragoza et al., 2022; O Pinheiro et al., 2024), from which the 3D graph is extracted using some post-processing procedure. The closest work to ours is GeoLDM (Xu et al., 2023), a geometric latent diffusion model that performs diffusion in the latent space of an equivariant Variational Auto-Encoder (VAE). It corresponds to a particular instance of END with $F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x}) = \alpha_t E_\varphi(\boldsymbol{x}) + \boldsymbol{\epsilon}\sigma_t$, where $E_\varphi(\boldsymbol{x})$ corresponds to the encoder of the VAE. Recently, several works have shown that leveraging 2D connectivity information can lead to improved results (Peng et al., 2023; Vignac et al., 2023; Le et al., 2024). While not incompatible with END, we perform our experiments without modeling that auxiliary information, and therefore do not compare to these approaches directly. Other generative frameworks have also been tailored to molecule generation, such as Flow Matching (Lipman et al., 2022; Song et al., 2023) or Bayesian Flow Networks (Graves et al., 2024; Song et al., 2024).

6. Conclusion

In this work, we have presented Equivariant Neural Diffusion (END), a novel diffusion model that is equivariant to Euclidean transformations. The key innovation in END lies in the forward process that is specified by a learnable data- and time-dependent transformation. Experimental results demonstrate the benefits of our method. In the unconditional setting, we show that END yields competitive generative performance across two different benchmarks. In the conditional setting, END offers better controllability, when conditioning on composition and substructure. Avenues for future work are numerous. In particular, leveraging the flexible framework of NFDM (Bartosh et al., 2024) to constrain the generative trajectories, e.g. to be straight and enable faster sampling, modelling bond information, or extending the conditional setting to other types of conditioning information, e.g. other point cloud or target property, are all promising directions.

References

Anderson, B. D. Reverse-time diffusion equation models. *Stochastic Processes and their Applications*, 12(3):313–326, 1982.

Anstine, D. M. and Isayev, O. Generative models as an emerging paradigm in the chemical sciences. *Journal of the American Chemical Society*, 145(16):8736–8750, 2023.

Axelrod, S. and Gomez-Bombarelli, R. Geom, energy-annotated molecular conformations for property prediction and molecular generation. *Scientific Data*, 9(1):185, 2022.

Bao, F., Zhao, M., Hao, Z., Li, P., Li, C., and Zhu, J. Equivariant energy-guided SDE for inverse molecular design. In *The Eleventh International Conference on Learning Representations*, 2023. URL https://openreview.net/forum?id=r0otLtOwYW.

Bartosh, G., Vetrov, D., and Naesseth, C. A. Neural diffusion models. *arXiv preprint arXiv:2310.08337*, 2023.

Bartosh, G., Vetrov, D., and Naesseth, C. A. Neural flow diffusion models: Learnable forward process for improved diffusion modelling. *arXiv preprint arXiv:2404.12940*, 2024.

Batatia, I., Kovacs, D. P., Simm, G., Ortner, C., and Csányi, G. Mace: Higher order equivariant message passing neural networks for fast and accurate force fields. *Advances in Neural Information Processing Systems*, 35: 11423–11436, 2022.

Batzner, S., Musaelian, A., Sun, L., Geiger, M., Mailoa, J. P., Kornbluth, M., Molinari, N., Smidt, T. E., and Kozinsky, B. E (3)-equivariant graph neural networks for data-efficient and accurate interatomic potentials. *Nature communications*, 13(1):2453, 2022.

Daigavane, A., Kim, S. E., Geiger, M., and Smidt, T. Symphony: Symmetry-equivariant point-centered spherical harmonics for 3d molecule generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=MIEnYtlGyv.

Garcia Satorras, V., Hoogeboom, E., Fuchs, F., Posner, I., and Welling, M. E (n) equivariant normalizing flows. *Advances in Neural Information Processing Systems*, 34: 4181–4192, 2021.

Gebauer, N., Gastegger, M., and Schütt, K. Symmetry-adapted generation of 3d point sets for the targeted discovery of molecules. *Advances in neural information processing systems*, 32, 2019.

Gebauer, N. W., Gastegger, M., Hessmann, S. S., Müller, K.-R., and Schütt, K. T. Inverse design of 3d molecular structures with conditional generative neural networks. *Nature communications*, 13(1):973, 2022.

Graves, A., Srivastava, R. K., Atkinson, T., and Gomez, F. Bayesian flow networks, 2024.

Gretton, A., Borgwardt, K. M., Rasch, M. J., Schölkopf, B., and Smola, A. A kernel two-sample test. *The Journal of Machine Learning Research*, 13(1):723–773, 2012.

Ho, J., Jain, A., and Abbeel, P. Denoising diffusion probabilistic models. *Advances in neural information processing systems*, 33:6840–6851, 2020.

Hoogeboom, E., Satorras, V. G., Vignac, C., and Welling, M. Equivariant diffusion for molecule generation in 3d. In *International conference on machine learning*, pp. 8867–8887. PMLR, 2022.

Igashov, I., Stärk, H., Vignac, C., Schneuing, A., Satorras, V. G., Frossard, P., Welling, M., Bronstein, M., and Correia, B. Equivariant 3d-conditional diffusion model for molecular linker design. *Nature Machine Intelligence*, pp. 1–11, 2024.

Kingma, D., Salimans, T., Poole, B., and Ho, J. Variational diffusion models. *Advances in neural information processing systems*, 34:21696–21707, 2021.

Landrum, G. et al. Rdkit: A software suite for cheminformatics, computational chemistry, and predictive modeling. *Greg Landrum*, 8:31, 2013.

Le, T., Noe, F., and Clevert, D.-A. Representation learning on biomolecular structures using equivariant graph attention. In *The First Learning on Graphs Conference*, 2022. URL https://openreview.net/forum?id=kv4xUo5Pu6.

Le, T., Cremer, J., Noe, F., Clevert, D.-A., and Schütt, K. T. Navigating the design space of equivariant diffusion-based generative models for de novo 3d molecule generation. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=kzGuiRXZrQ.

Lipman, Y., Chen, R. T., Ben-Hamu, H., Nickel, M., and Le, M. Flow matching for generative modeling. In *The Eleventh International Conference on Learning Representations*, 2022.

Luo, Y. and Ji, S. An autoregressive flow model for 3d molecular geometry generation from scratch. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=C03Ajc-NS5W.

Merchant, A., Batzner, S., Schoenholz, S. S., Aykol, M., Cheon, G., and Cubuk, E. D. Scaling deep learning for materials discovery. *Nature*, 624(7990):80–85, 2023.

Nielsen, B. M. G., Christensen, A., Dittadi, A., and Winther, O. Diffenc: Variational diffusion with a learned encoder. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=8nxy1bQWTG.

O Pinheiro, P. O., Rackers, J., Kleinhenz, J., Maser, M., Mahmood, O., Watkins, A., Ra, S., Sresht, V., and Saremi, S. 3d molecule generation by denoising voxel grids. *Advances in Neural Information Processing Systems*, 36, 2024.

O'Boyle, N. M., Banck, M., James, C. A., Morley, C., Vandermeersch, T., and Hutchison, G. R. Open babel: An open chemical toolbox. *Journal of cheminformatics*, 3: 1–14, 2011.

Peng, X., Guan, J., Liu, Q., and Ma, J. Moldiff: Addressing the atom-bond inconsistency problem in 3d molecule diffusion generation. In *International Conference on Machine Learning*, pp. 27611–27629. PMLR, 2023.

Ragoza, M., Masuda, T., and Koes, D. R. Generating 3d molecules conditional on receptor binding sites with deep generative models. *Chemical science*, 13(9):2701–2713, 2022.

Ramakrishnan, R., Dral, P. O., Rupp, M., and Von Lilienfeld, O. A. Quantum chemistry structures and properties of 134 kilo molecules. *Scientific data*, 1(1):1–7, 2014.

Ruddigkeit, L., Van Deursen, R., Blum, L. C., and Reymond, J.-L. Enumeration of 166 billion organic small molecules in the chemical universe database gdb-17. *Journal of chemical information and modeling*, 52(11):2864–2875, 2012.

Schütt, K., Kindermans, P.-J., Sauceda Felix, H. E., Chmiela, S., Tkatchenko, A., and Müller, K.-R. Schnet: A continuous-filter convolutional neural network for modeling quantum interactions. *Advances in neural information processing systems*, 30, 2017.

Schütt, K., Unke, O., and Gastegger, M. Equivariant message passing for the prediction of tensorial properties and molecular spectra. In *International Conference on Machine Learning*, pp. 9377–9388. PMLR, 2021.

Simm, G., Pinsler, R., and Hernández-Lobato, J. M. Reinforcement learning for molecular design guided by quantum mechanics. In *International Conference on Machine Learning*, pp. 8959–8969. PMLR, 2020.

Skalic, M., Jiménez, J., Sabbadin, D., and De Fabritiis, G. Shape-based generative modeling for de novo drug design. *Journal of chemical information and modeling*, 59(3):1205–1214, 2019.

Sohl-Dickstein, J., Weiss, E., Maheswaranathan, N., and Ganguli, S. Deep unsupervised learning using nonequilibrium thermodynamics. In Bach, F. and Blei, D. (eds.), *Proceedings of the 32nd International Conference on Machine Learning*, volume 37 of *Proceedings of Machine Learning Research*, pp. 2256–2265, Lille, France, 07–09 Jul 2015. PMLR. URL https://proceedings.mlr.press/v37/sohl-dickstein15.html.

Song, Y., Sohl-Dickstein, J., Kingma, D. P., Kumar, A., Ermon, S., and Poole, B. Score-based generative modeling through stochastic differential equations. In *International Conference on Learning Representations*, 2020.

Song, Y., Gong, J., Xu, M., Cao, Z., Lan, Y., Ermon, S., Zhou, H., and Ma, W.-Y. Equivariant flow matching with hybrid probability transport for 3d molecule generation. In Oh, A., Neumann, T., Globerson, A., Saenko, K., Hardt, M., and Levine, S. (eds.), *Advances in Neural Information Processing Systems*, volume 36, pp. 549–568. Curran Associates, Inc., 2023. URL https://proceedings.neurips.cc/paper_files/paper/2023/file/01d64478381c33e29ed611f1719f5a37-Paper-Conference.pdf.

Song, Y., Gong, J., Zhou, H., Zheng, M., Liu, J., and Ma, W.-Y. Unified generative modeling of 3d molecules with bayesian flow networks. In *The Twelfth International Conference on Learning Representations*, 2024. URL https://openreview.net/forum?id=NSVtmmzeRB.

Unke, O. T., Chmiela, S., Sauceda, H. E., Gastegger, M., Poltavsky, I., Schuett, K. T., Tkatchenko, A., and Mueller, K.-R. Machine learning force fields. *Chemical Reviews*, 121(16):10142–10186, 2021.

Vignac, C., Osman, N., Toni, L., and Frossard, P. Midi: Mixed graph and 3d denoising diffusion for molecule generation. In *Joint European Conference on Machine Learning and Knowledge Discovery in Databases*, pp. 560–576. Springer, 2023.

Weininger, D. Smiles, a chemical language and information system. 1. introduction to methodology and encoding rules. *Journal of chemical information and computer sciences*, 28(1):31–36, 1988.

Wu, L., Gong, C., Liu, X., Ye, M., and qiang liu. Diffusion-based molecule generation with informative prior bridges. In Oh, A. H., Agarwal, A., Belgrave, D., and Cho,

K. (eds.), *Advances in Neural Information Processing Systems*, 2022. URL https://openreview.net/forum?id=TJUNtiZiTKE.

Wu, L., Trippe, B., Naesseth, C., Blei, D., and Cunningham, J. P. Practical and asymptotically exact conditional sampling in diffusion models. *Advances in Neural Information Processing Systems*, 36, 2024.

Xu, M., Yu, L., Song, Y., Shi, C., Ermon, S., and Tang, J. Geodiff: A geometric diffusion model for molecular conformation generation. In *International Conference on Learning Representations*, 2022. URL https://openreview.net/forum?id=PzcvxEMzvQC.

Xu, M., Powers, A. S., Dror, R. O., Ermon, S., and Leskovec, J. Geometric latent diffusion models for 3d molecule generation. In *International Conference on Machine Learning*, pp. 38592–38610. PMLR, 2023.

Yang, L., Zhang, Z., Song, Y., Hong, S., Xu, R., Zhao, Y., Zhang, W., Cui, B., and Yang, M.-H. Diffusion models: A comprehensive survey of methods and applications. *ACM Computing Surveys*, 56(4):1–39, 2023.

A. Appendix / supplemental material

A.1. Time-derivative of an O(3)-equivariant function

Let $f : \mathcal{X} \times [0, 1] \to \mathcal{Y}$ be a function that is equivariant to actions of the group $O(3)$, such that $\boldsymbol{R} \cdot f(\boldsymbol{x}, t) = f(\boldsymbol{R} \cdot \boldsymbol{x}, t), \forall \boldsymbol{R} \in O(3)$.

Proof sketch We need to show that $\frac{\partial}{\partial t}\big(f(\boldsymbol{R} \cdot \boldsymbol{x}, t)\big) = \boldsymbol{R} \cdot \frac{\partial}{\partial t}\big(f(\boldsymbol{x}, t)\big)$, $\forall \boldsymbol{R} \in O(3)$ and $\forall \boldsymbol{x} \in \mathcal{X}$.

$$\frac{\partial}{\partial t}\big(f(\boldsymbol{R} \cdot \boldsymbol{x}, t)\big) = \frac{\partial}{\partial t}\big(\boldsymbol{R} \cdot f(\boldsymbol{x}, t)\big), \tag{10}$$

$$= \boldsymbol{R} \cdot \frac{\partial}{\partial t}\big(f(\boldsymbol{x}, t)\big) \tag{11}$$

where the last equality follows by linearity.

A.2. Inverse of an O(3)-equivariant function

Let $f : \mathcal{X} \to \mathcal{Y}$ be a function that (1) is equivariant to the action of the group $O(3)$, and (2) admits an inverse $f^{-1} : \mathcal{Y} \to \mathcal{X}$, then f^{-1} is also equivariant to the action of $O(3)$.

Proof sketch We need to show that $f^{-1}\big(\boldsymbol{R} \cdot \boldsymbol{y}\big) = \boldsymbol{R} \cdot f^{-1}(\boldsymbol{y})$, $\forall \boldsymbol{R} \in O(3)$ and $\forall \boldsymbol{y} \in \mathcal{Y}$.

Since f is invertible, we have that to any $\boldsymbol{y} \in \mathcal{Y}$ corresponds a unique $\boldsymbol{x} \in \mathcal{X}$, such that $\boldsymbol{y} = f(\boldsymbol{x})$, and that $f^{-1}(\boldsymbol{y}) = f^{-1}(f(\boldsymbol{x})) = \boldsymbol{x}$. As f is equivariant to the action of $O(3)$, we have that $\boldsymbol{R} \cdot f(\boldsymbol{x}) = f(\boldsymbol{R} \cdot \boldsymbol{x}), \forall \boldsymbol{R} \in O(3)$:

$$f^{-1}\big(\boldsymbol{R} \cdot \boldsymbol{y}\big) = f^{-1}\big(\boldsymbol{R} \cdot f(\boldsymbol{x})\big), \tag{12}$$

$$= f^{-1}\big(f(\boldsymbol{R} \cdot \boldsymbol{x})\big), \tag{13}$$

$$= \boldsymbol{R} \cdot \boldsymbol{x}, \tag{14}$$

$$= \boldsymbol{R} \cdot f^{-1}(\boldsymbol{y}). \tag{15}$$

A.3. O(3)-invariance of the objective function

In this section, we show that the objective function is invariant under the action of $O(3)$: $\mathcal{L}_{\text{END}}(\mathbf{R}\boldsymbol{x}) = \mathcal{L}_{\text{END}}(\boldsymbol{x}), \forall \boldsymbol{R} \in O(3)$, provided that F_φ and $\hat{x}_\theta$.

$$
\begin{aligned}
\mathcal{L}_{\text{END}}(\mathbf{R}\boldsymbol{x}) &= \mathbb{E}_{u(t),q_\varphi(\boldsymbol{z}_t|\mathbf{R}\boldsymbol{x})q(\mathbf{R}\boldsymbol{x})}\Big[\frac{1}{2g_\varphi^2(t)}\big|\big|f_\varphi(\boldsymbol{z}_t,t,\mathbf{R}\boldsymbol{x}) - \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{z}_t}\log q_\varphi(\boldsymbol{z}_t|\mathbf{R}\boldsymbol{x}) \\
&\qquad - f_\varphi(\boldsymbol{z}_t,t,\hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t,t)) + \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{z}_t}\log q_\varphi(\boldsymbol{z}_t|\hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t,t))\big|\big|_2^2\Big], \\
&= \mathbb{E}_{u(t),q_\varphi(\boldsymbol{z}_t|\mathbf{R}\boldsymbol{x})q(\mathbf{R}\boldsymbol{x})}\Big[\frac{1}{2g_\varphi^2(t)}\big|\big|f_\varphi(\mathbf{R}\mathbf{R}^{-1}\boldsymbol{z}_t,t,\mathbf{R}\boldsymbol{x}) - \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{z}_t}\log q_\varphi(\mathbf{R}\mathbf{R}^{-1}\boldsymbol{z}_t|\mathbf{R}\boldsymbol{x}) \\
&\qquad - f_\varphi(\mathbf{R}\mathbf{R}^{-1}\boldsymbol{z}_t,t,\hat{\boldsymbol{x}}_\theta(\mathbf{R}\mathbf{R}^{-1}\boldsymbol{z}_t,t)) + \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{z}_t}\log q_\varphi(\mathbf{R}\mathbf{R}^{-1}\boldsymbol{z}_t|\hat{\boldsymbol{x}}_\theta(\mathbf{R}\mathbf{R}^{-1}\boldsymbol{z}_t,t))\big|\big|_2^2\Big], \\
&= \mathbb{E}_{u(t),q_\varphi(\boldsymbol{z}_t|\mathbf{R}\boldsymbol{x})q(\mathbf{R}\boldsymbol{x})}\Big[\frac{1}{2g_\varphi^2(t)}\big|\big|\mathbf{R}f_\varphi(\mathbf{R}^{-1}\boldsymbol{z}_t,t,\boldsymbol{x}) - \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{z}_t}\log q_\varphi(\mathbf{R}^{-1}\boldsymbol{z}_t|\boldsymbol{x}) \\
&\qquad - \mathbf{R}f_\varphi(\mathbf{R}^{-1}\boldsymbol{z}_t,t,\hat{\boldsymbol{x}}_\theta(\mathbf{R}^{-1}\boldsymbol{z}_t,t)) + \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{z}_t}\log q_\varphi(\mathbf{R}^{-1}\boldsymbol{z}_t|\hat{\boldsymbol{x}}_\theta(\mathbf{R}^{-1}\boldsymbol{z}_t,t))\big|\big|_2^2\Big], \\
&= \mathbb{E}_{u(t),q_\varphi(\mathbf{R}\boldsymbol{y}_t|\mathbf{R}\boldsymbol{x})q(\mathbf{R}\boldsymbol{x})}\Big[\frac{1}{2g_\varphi^2(t)}\big|\big|\mathbf{R}f_\varphi(\boldsymbol{y}_t,t,\boldsymbol{x}) - \frac{g_\varphi^2(t)}{2}\mathbf{R}\nabla_{\boldsymbol{y}_t}\log q_\varphi(\boldsymbol{y}_t|\boldsymbol{x}) \\
&\qquad - \mathbf{R}f_\varphi(\boldsymbol{y}_t,t,\hat{\boldsymbol{x}}_\theta(\boldsymbol{y}_t,t)) + \frac{g_\varphi^2(t)}{2}\mathbf{R}\nabla_{\boldsymbol{y}_t}\log q_\varphi(\boldsymbol{y}_t|\hat{\boldsymbol{x}}_\theta(\boldsymbol{y}_t,t))\big|\big|_2^2\Big], \\
&= \mathbb{E}_{u(t),q_\varphi(\boldsymbol{y}_t|\boldsymbol{x})q(\boldsymbol{x})}\Big[\frac{1}{2g_\varphi^2(t)}\big|\big|f_\varphi(\boldsymbol{y}_t,t,\boldsymbol{x}) - \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{y}_t}\log q_\varphi(\boldsymbol{y}_t|\boldsymbol{x}) \\
&\qquad - f_\varphi(\boldsymbol{y}_t,t,\hat{\boldsymbol{x}}_\theta(\boldsymbol{y}_t,t)) + \frac{g_\varphi^2(t)}{2}\nabla_{\boldsymbol{y}_t}\log q_\varphi(\boldsymbol{y}_t|\hat{\boldsymbol{x}}_\theta(\boldsymbol{y}_t,t))\big|\big|_2^2\Big], \\
&= \mathcal{L}_{\text{END}}(\boldsymbol{x})
\end{aligned}
$$

The first equality is obtained by replacing $\boldsymbol{x}$ by $\mathbf{R}\boldsymbol{x}$ in the definition of the objective function Equation (5). The second is obtained by multiplying by $\mathbf{R}\mathbf{R}^{-1} = \mathbb{I}$. The third equality by leveraging that f_φ, q_φ and $\hat{x}_\theta$ are equivariant. We then perform a change of variable $\boldsymbol{y}_t = \mathbf{R}^{-1}\boldsymbol{z}_t$. As rotation does preserve distances, we obtain the last equality.

A.4. Details about F_φ

A.4.1. Working in ambient space

In practice, we would like F_φ and $\hat{x}_\theta$ to operate in ambient space, i.e. on $\boldsymbol{z}_t^{(r)} \in \mathbb{R}^{M\times 3}$.

Garcia Satorras et al. (2021) showed that the Jacobian of the transformation can be directly computed in ambient space for $\boldsymbol{z}_t^{(r)}$ and $\boldsymbol{\varepsilon}^{(r)}$ that live in the linear subspace R, provided that the transformation F_φ leaves the center of mass unchanged. Additionally, F_φ is required to be invertible with respect to $\boldsymbol{\varepsilon}$.

If we consider a flat representation of $\boldsymbol{z}_t^{(r)} \in \mathbb{R}^{M\cdot d}$, such transformation can be written as,

$$
U_\varphi(\boldsymbol{x},t) = (\mathbb{I}_{M\cdot d} - \frac{1}{M}TT^\top)\tilde{U}_\varphi(\boldsymbol{x},t) + \frac{1}{M}TT^\top, \tag{16}
$$

where $T \in \mathbb{R}^{M\cdot d\times d} = [\mathbb{I}_d, \mathbb{I}_d, ...]^\top$, and $\frac{1}{M}TT^\top$ corresponds to the linear operator computing the center of mass. Intuitively, the first term corresponds to a linear transformation followed by a projection to the 0-CoM subspace, while the second term translates the system back to the initial center of mass.

Computing $\boldsymbol{z}_t^{(r)}$ from $\boldsymbol{\varepsilon}^{(r)}$ Given that $\boldsymbol{\varepsilon}^{(r)} \in R$, obtaining $\boldsymbol{z}_t^{(r)} \in R$ from $\boldsymbol{\varepsilon}^{(r)}$ simply amounts to (1) computing $\tilde{\boldsymbol{z}}_t^{(r)} = \tilde{U}_\varphi(\boldsymbol{x},t)\boldsymbol{\varepsilon}^{(r)}$, and then (2) removing the center of mass from $\tilde{\boldsymbol{z}}_t^{(r)}$.

Computing $|J_F|$ and F^{-1} In what follows, we shorten the notation, and denote $U_\varphi(\boldsymbol{x},t)$ by U and $\tilde{U}_\varphi(\boldsymbol{x},t)$ by $\tilde{U}$. We

start by reorganizing Equation (16), as

$$U = \tilde{U} + \frac{1}{M} T T^\top (\mathbb{I}_{M\cdot d} - \tilde{U}). \tag{17}$$

Computing $\det U$ can be done by leveraging the Matrix Determinant Lemma,

$$\begin{aligned}
\det U &= \det \tilde{U} \cdot \det \big(\mathbb{I}_d + \frac{1}{M} T^\top (\mathbb{I}_{M\cdot d} - \tilde{U}) \tilde{U}^{-1} T\big), && (18)\\
&= \det \tilde{U} \cdot \det \big(\mathbb{I}_d + \frac{1}{M} T^\top (\tilde{U}^{-1} - \mathbb{I}_{M\cdot d}) T\big), && (19)\\
&= \det \tilde{U} \cdot \det \big(\frac{1}{M} \sum_{m=1}^{M} (\tilde{U}^m)^{-1}\big), && (20)\\
&= \det \tilde{U} \cdot \det V, && (21)\\
&= \prod_{m=1}^{M} \det \tilde{U}^m \cdot \det V && (22)
\end{aligned}$$

where $V = \frac{1}{M} \sum_{m=1}^{M} (\tilde{U}^m)^{-1}$ is a $d \times d$-matrix, and $\tilde{U}^m$ denotes the m-th $d \times d$-block in $\tilde{U}$.

The inverse can be obtained by leveraging the Woodbury matrix identity,

$$\begin{aligned}
U^{-1} &= \big(\tilde{U} + \frac{1}{M} T T^\top (\mathbb{I}_{M\cdot d} - \tilde{U})\big)^{-1} && (23)\\
&= \tilde{U}^{-1} - \frac{1}{M} \tilde{U}^{-1} T V^{-1} T^\top (\tilde{U}^{-1} - \mathbb{I}_{M\cdot d}), && (24)\\
&= \tilde{U}^{-1} (\mathbb{I} - C) && (25)
\end{aligned}$$

where $V = \frac{1}{M} \sum_{m=1}^{M} (\tilde{U}^m)^{-1}$, as previously defined, and $C = \frac{1}{M} T V^{-1} T^\top (\tilde{U}^{-1} - \mathbb{I}_{M\cdot d})$.

In practice, we do not need the inverse itself, but rather $\boldsymbol{\varepsilon}^{(r)}$ given $\boldsymbol{z}_t^{(r)}$,

$$\begin{aligned}
\boldsymbol{\varepsilon}^{(r)} &= U^{-1} \boldsymbol{z}_t^{(r)}, && (26)\\
&= \tilde{U}^{-1} (\mathbb{I} - C) \boldsymbol{z}_t^{(r)}, && (27)\\
&= \tilde{U}^{-1} \big[\boldsymbol{z}_t^{(r)} - \boldsymbol{c}\big], && (28)
\end{aligned}$$

where $\boldsymbol{c}$ is a translation vector that can be obtained without constructing the full matrix C.

A.4.2. Invariant features

For simplicity, we omitted in Section 3.1 that molecules are described as tuples: $\boldsymbol{x} = (\boldsymbol{r}, \boldsymbol{h})$, as only $\boldsymbol{r}$ transform under Euclidean transformations. For the invariant features $\boldsymbol{h}$, we use the following parameterization

$$\mu_\varphi^{(h)}(\boldsymbol{x}, t) = (1-t)\boldsymbol{h} + t\big(1-t\big)\bar{\mu}_\varphi^{(h)}(\boldsymbol{x}, t), \tag{29}$$

which ensures that $\mu_\varphi^{(h)}(\boldsymbol{x}, 0) = \boldsymbol{h}$, and $\mu_\varphi^{(h)}(\boldsymbol{x}, 1) = \mathbf{0}$; whereas for $\sigma_\varphi^{(h)}(\boldsymbol{x}, t)$, we use the following

$$\sigma_\varphi^{(h)}(\boldsymbol{x}, t) = \delta^{1-t} \bar{\sigma}_\varphi(\boldsymbol{x}, t)^{t(1-t)}. \tag{30}$$

Implementation F_φ is implemented as a neural network with an architecture similar to that of the data point predictor $\hat{\boldsymbol{x}}_\theta(\boldsymbol{z}_t, t)$, but with a specific readout layer that produces the outputs related to $\boldsymbol{r}$ ($[\bar{\mu}_\varphi(\boldsymbol{x}, t), \bar{\sigma}_\varphi, \bar{U}_\varphi(\boldsymbol{x}, t)]$), as described in the main text in Section 3.1. Additionally, it produces $\bar{\mu}_\varphi^{(h)}(\boldsymbol{x}, t)$ and $\bar{\sigma}_\varphi^{(h)}(\boldsymbol{x}, t)$ as invariant outputs.

Inverse transformation The logarithm of the determinant of the inverse transformation $\log |J_F^{-1}|$ writes

$$\log |J_F^{-1}| = -\log |J_F| = -\underbrace{\sum_{i=1}^{M\times D} \sigma_\varphi^{(h),i}(\boldsymbol{x}, t)}_{\text{invariant features}} - \underbrace{\sum_{i=m}^{M} \log \big|\det(U_\varphi^m(\boldsymbol{x}, t))\big| - \log\det|V|}_{\text{vectorial features}}, \tag{31}$$

where V is defined as in Equation (22).

A.5. Algorithms

Algorithm 1 Training algorithm of END

Require: $q(\boldsymbol{x})$, F_φ, $\hat{\boldsymbol{x}}_\theta$
  for training iterations **do**
    $\boldsymbol{x} \sim q(\boldsymbol{x})$, $t \sim u(t)$, $\boldsymbol{\varepsilon} \sim p(\boldsymbol{\varepsilon})$
    $\boldsymbol{z}_t \leftarrow \mu_\varphi(\boldsymbol{x}, t) + U_\varphi(\boldsymbol{x}, t)\boldsymbol{\varepsilon}$
    $\mathcal{L} = \frac{1}{2g_\varphi^2(t)} \left|\left| f_\varphi^B(\boldsymbol{z}_t, t, \boldsymbol{x}) - \hat{f}_{\theta,\varphi}(\boldsymbol{z}_t, t) \right|\right|_2^2$
    Gradient step on θ and φ
  end for

Algorithm 2 Stochastic sampling from END

Require: F_φ, $\hat{\boldsymbol{x}}_\theta$, integration steps T, empirical distribution of number of atoms $p(N)$
  $\Delta t = \frac{1}{T}$
  $N \sim p(N)$
  $\boldsymbol{z}_1 \sim p(\boldsymbol{z}_1)$
  for $t = 1, ..., \frac{1}{T}$ **do**
    $\bar{\boldsymbol{w}} \sim \mathcal{N}(\mathbf{0}, \mathbb{I})$
    $\boldsymbol{z}_{t-\Delta t} \leftarrow \boldsymbol{z}_t - \hat{f}_{\theta,\varphi}(\boldsymbol{z}_t, t)\Delta t + g_\varphi(t)\bar{\boldsymbol{w}}\sqrt{\Delta t}$
  end for
  $\boldsymbol{x} \sim p(\boldsymbol{x}|\boldsymbol{z}_0)$

A.6. Experimental details

A.6.1. Evaluation metrics

Stability An atom is deemed stable if it has a charge of 0, whereas a molecule is stable if all its atoms have 0 charge. We reuse the lookup table from (Hoogeboom et al., 2022) to infer bond types from pairwise distances.

Validity Validity corresponds to the percentage of samples that can be parsed and sanitized by RDKit (Landrum et al., 2013), after inference of the bonds using a lookup table (Hoogeboom et al., 2022). It should be noted that the metric does not penalize fragmented samples as long as each individual fragment appears valid. This can be problematic when running evaluation on larger compounds, as models tend to generated disconnected structures.

Uniqueness It is the proportion of samples that are valid and have a unique SMILES string (Weininger, 1988).

Total variation The total variation is computed as the MAE between the (discrete) marginal obtained on the training data and on the generated samples. For bond types, we compute the ground truth and generated distributions using the lookup table mechanism.

MMD We follow the procedure of (Daigavane et al., 2024), and compute the MMD between true and generated pairwise distances distributions for the 10 most common bonds in QM9: ["C-H:1.0", "C-C:1.0", "C-O:1.0", "C-N:1.0", "H-N:1.0", "C-O:2.0", "C-N:1.5", "H-O:1.0", "C-C:1.5", "C-N:2.0"]. Here, we infer bonds using RDKit's `rdDetermineBonds.DetermineBonds` with `charge=0`.

A.6.2. Architecture

Our forward transformation F_φ and data point predictor $\hat{x}_\theta$ share a common neural network architecture that we detail here. The architecture is similar to that of EQCAT (Le et al., 2022), and updates a collection of invariant and equivariant features for each node in the graph. We chose that architecture because it allows for an easy construction of $\bar{U}_\varphi(\boldsymbol{x}, t)$ by linear projection of the final equivariant layer.

We follow previous work (Hoogeboom et al., 2022) and consider fully-connected graphs. We initially featurize pairwise distances through Gaussian Radial Basis functions, with dataset-specific cutoff taken large enough to ensure full connectivity. In opposition to (Hoogeboom et al., 2022), we do not update positions in the message-passing phase, but instead obtain

the positions prediction through a linear projection of the final equivariant hidden states. The predictions for the invariant features are obtained by reading out the final invariant hidden states.

Optimization For all model variants, we employ Adam with a learning rate of 10^{-4}. We perform gradient clipping (norm) with a value of 10 on QM9, and a value of 1 on GEOM-DRUGS.

A.6.3. UNCONDITIONAL GENERATION

We reuse the data setup from previous work (Hoogeboom et al., 2022; Xu et al., 2023).

QM9 On QM9, we use 10 layers of message passing for EDM*, while the variants of END feature 5 layers of message-passing in F_φ and 5 layers in $\hat{x}_\theta$. For all models, we use 256 invariant and 256 equivariant hidden features, along with an RBF expansion of dimension 64 with a cutoff of 10Å for pairwise distances. This ensures that the compared models have the same number of learnable parameters, i.e. 9.4M each. We train all models for 1000 epochs with a batch size of 64.

GEOM-Drugs On GEOM-DRUGS, we use 10 layers of message passing for EDM*, while the variants of END feature 5 layers of message-passing in F_φ and 5 in $\hat{x}_\theta$. The hidden size of the invariant and equivariant features is set to 192, along with an RBF expansion of dimension 64 with a cutoff of 30Å for pairwise distances. Each model features 5.4M learnable parameters. We train all models for 10 epochs with an effective batch size of 64.

A.6.4. CONDITIONAL GENERATION

We use 10 layers of message passing for EDM*, while the variants of END feature 5 layers of message-passing in F_φ and 5 in $\hat{x}_\theta$. The hidden size of the invariant and equivariant features is set to 192 , along with an RBF expansion of dimension 64 with a cutoff of 10Å for pairwise distances. We train all models for 1000 epochs with a batch size of 64.

After an initial encoding, the conditional information is introduced at the end of each message passing step, and alters the scalar hidden states through a one-layer MLP, that shares the same dimension as the hidden scalar state.

Composition-conditioned generation The encoding of the condition follows that of (Gebauer et al., 2022). Each atom type gets its own embedding, weighted by the proportion it represents in the provided formula. The weighted embeddings of all atom types are then concatenated and flattened, and the obtained vector is processed through a 2-layer MLP with 64 hidden units. The composition used at sampling time are extracted from the validation and test set. For each unique formula, the model gets to generate 10 samples.

Substructure-conditioned generation The encoding of the condition follows that of (Bao et al., 2023). The 1024-dimensional fingerprint is simply processed by a $2-$layer MLP with hidden dimensions $[512, 256]$, and a final linear projection to 192, i.e. the hidden size of the invariant features.

A.7. Compared models

In Table 5, we detail the compared all models in terms of their transformation F_φ.

Table 5: Compared models.

	$F_\varphi(\boldsymbol{\varepsilon}, t, \boldsymbol{x})$	Comment
EDM (Hoogeboom et al., 2022) / EDM*	$\alpha_t \boldsymbol{x} + \sigma_t \boldsymbol{\varepsilon}$	$\alpha_t = \exp\big(-\frac{1}{2}\int_0^t \beta(s)\,ds\big)$ $\sigma_t = 1 - \exp\big(-\frac{1}{2}\int_0^t \beta(s)\,ds\big)$
GEOLDM (Xu et al., 2023)	$\alpha_t E_\varphi(\boldsymbol{x}) + \sigma_t \boldsymbol{\varepsilon}$	α_t and σ_t similar to EDM $p(\boldsymbol{x}\|\boldsymbol{z}_0) = \mathcal{N}\big(\boldsymbol{x}\|D_\varphi(\boldsymbol{z}_0), \delta^2\mathbb{I}\big)$
EDM* + γ_φ	$\alpha_\varphi(t)\boldsymbol{x} + \sigma_\varphi(t)\boldsymbol{\varepsilon}$	learned γ_φ with 2 separate outputs (for $\boldsymbol{r}$ and $\boldsymbol{h}$)
END (μ_φ only)	$\mu_\varphi(\boldsymbol{x}, t) + \sigma_t \boldsymbol{\varepsilon}$	σ_t similar to EDM
END	$\mu_\varphi(\boldsymbol{x}, t) + U_\varphi(\boldsymbol{x}, t)\boldsymbol{\varepsilon}$	as introduced in Equation (6)

A.8. Compute resources

All experiments were run on a single GPU. The experiments on QM9 were run on a NVIDIA SM3090 with 24 GB of memory. The experiments on GEOM-DRUGS were run on NVIDIA A100 with 40 GB. Training took up to 7 days.